

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский Государственный Университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Рецензент

\_\_\_\_\_ 2019 г.  
«\_\_» \_\_\_\_\_

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2019 г.

Веб-приложение планирования для дизайна ландшафта садового участка

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. «Электронные  
вычислительные машины»

\_\_\_\_\_ И.Л. Надточий  
«\_\_» \_\_\_\_\_ 2019 г.

Автор работы,  
студент группы КЭ-452

\_\_\_\_\_ А.А. Угрюмова  
«\_\_» \_\_\_\_\_ 2019 г.

Нормоконтролер,  
ст. преп. каф. «Электронные  
вычислительные машины»

\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2019 г.

Челябинск-2019

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский Государственный Университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»  
Направление «Информатика и вычислительная техника»

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г.И. Радченко  
«\_\_\_» \_\_\_\_\_ 2019 г.

**ЗАДАНИЕ**  
**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-452  
Угрюмовой Александры Александровны  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

- 1. Тема работы:** «Веб-приложение планирования для дизайна ландшафта садового участка» утверждена приказом по университету от 25 апреля 2019 г. №899
- 2. Срок сдачи студентом законченной работы:** 19 июня 2019 г.
- 3. Исходные данные к работе:**  
Обеспечить основной функционал веб-приложения:
  - просмотр статей посетителями и пользователями сайта;
  - публикация статей пользователями сайта;
  - редактирование статей пользователями сайта;
  - комментирование статей пользователями;
  - редактирование комментариев пользователями;
  - создание чертежа ландшафта садового участка с помощью графического онлайн-редактора веб-приложения.

Предусмотреть наличие ролей: администратор, модератор, обычный пользователь.

Предусмотреть систему регистрации учетных записей пользователей.

Предусмотреть отображение страниц для неавторизованных пользователей.

Предусмотреть систему изменения профиля пользователей.

Предусмотреть систему изменения параметров учетной записи пользователей.

#### **4. Перечень подлежащих разработке вопросов:**

- рассмотрение существующих аналогов на рынке по выбранной тематике;
- формирование требований к веб-приложению;
- анализ современных программных технологий для создания веб-приложения публикации статей;
- анализ современных программных технологий для создания графического онлайн-редактора в веб-браузере;
- проектирование веб-приложения;
- разработка собственного веб-приложения просмотра, публикации, редактирования статей, создания чертежа ландшафта садового участка;
- тестирование работоспособности разработанного веб-приложения.

#### **5. Дата выдачи задания: 1 декабря 2018 г.**

Руководитель работы \_\_\_\_\_ /И.Л. Надточий/

Студент \_\_\_\_\_ /А.А. Угрюмова /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2019	
Разработка модели, проектирование	01.04.2019	
Реализация системы	01.05.2019	
Тестирование, отладка, эксперименты	15.05.2019	
Компоновка текста работы и сдача на нормоконтроль	13.06.2019	
Подготовка презентации и доклада	15.06.2019	
Приемка приложения при комиссии высшего учебного заведения	19.06.2019	

Руководитель работы \_\_\_\_\_ */И.Л. Надточий/*

Студент \_\_\_\_\_ */А.А. Угрюмова/*

## АННОТАЦИЯ

А.А. Угрюмова. Веб-приложение планирования для дизайна ландшафта садового участка. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭиКН; 2019, 140 с., 19 табл., 36 ил., библиогр. список –14 наим.

В рамках выпускной квалификационной работы производится разработка веб-приложения для дизайна ландшафта садового участка. Целью разработки данного веб-приложения является оказание помощи посетителям и пользователям веб-приложения в формировании представления будущего дизайна садового участка. Помощь заключается в разработке веб-приложения, где все пользователи могут публиковать/редактировать свои статьи на тематику, так или иначе связанную с дизайном садового участка, а также с его уходом, защитой и пр. Организуется:

- анализ существующих аналогов на интернет-рынке, их преимущества и недостатки;
- проектирование веб-приложения;
- анализ бесплатных систем управления содержимым сайта, на базе которых разрабатываются веб-приложения;
- разработка веб-приложения на базе системы управления содержимым сайта “WordPress” для просмотра статей незарегистрированными пользователями, публикации и редактирования статей зарегистрированными пользователями, модерирования статей и отзывов пользователей к ним модераторами и администрацией, предоставления специального графического онлайн-редактора для создания чертежа ландшафта садового участка.

Производится оценка, выборка и анализ результатов работы разработанного веб-приложения в процессе его тестирования. В работе рассматриваются преимущества и недостатки разработанного веб-приложения.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	9
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	11
1.1. ОБЗОР АНАЛОГОВ.....	11
1.1.1. ОБЗОР АНАЛОГОВ СРЕДИ ВЕБ-СЕРВИСОВ.....	11
1.1.2. ОБЗОР АНАЛОГОВ СРЕДИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	13
1.2. АНАЛИЗ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	14
1.2.1. АНАЛИЗ СИСТЕМ УПРАВЛЕНИЯ СОДЕРЖИМЫМ САЙТА .....	14
1.2.2. ЯЗЫК ПРОГРАММИРОВАНИЯ БАЗЫ ДАННЫХ.....	17
1.2.3. ЯЗЫК ПРОГРАММИРОВАНИЯ СЕРВЕРНОЙ ЧАСТИ ВЕБ- ПРИЛОЖЕНИЯ .....	19
1.2.4. ЯЗЫК ПРОГРАММИРОВАНИЯ КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ .....	22
1.2.5. JAVASCRIPT-БИБЛИОТЕКА ДЛЯ РЕАЛИЗАЦИИ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА.....	22
1.3. ВЫВОД.....	24
2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К ВЕБ-ПРИЛОЖЕНИЮ .....	25
2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	25
2.1.1. В СОСТАВЕ ВЕБ-ПРИЛОЖЕНИЯ РЕАЛИЗОВАНЫ СЛЕДУЮЩИЕ ПОДСИСТЕМЫ .....	25
2.1.2. РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЯ ПРЕДОСТАВЛЯЕТ .....	25
2.1.3. ПУБЛИКАЦИЯ ТЕМАТИЧЕСКИХ МАТЕРИАЛОВ АВТОРИЗОВАННЫМИ ПОЛЬЗОВАТЕЛЯМИ ПРЕДОСТАВЛЯЕТ.....	26
2.1.4. СОЗДАНИЕ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА С ПОМОЩЬЮ СПЕЦИАЛЬНОГО РЕДАКТОРА ПРЕДОСТАВЛЯЕТ .....	26
2.1.5. СОЗДАНИЕ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА ПРЕДОСТАВЛЯЕТ .....	27
2.1.6. В ПРОЦЕССЕ СОЗДАНИЯ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА ПОЛЬЗОВАТЕЛЮ ДОСТУПНЫ.....	27

2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	28
2.2.1. ТРЕБОВАНИЕ К КВАЛИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ .....	28
2.2.2. ТРЕБОВАНИЯ К ОБЩЕСИСТЕМНОМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ .....	29
3. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ .....	30
3.1. АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЯ .....	30
3.2. ОПИСАНИЕ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ .....	31
3.2.1. ВХОДНЫЕ ДАННЫЕ .....	31
3.2.2. ВЫХОДНЫЕ ДАННЫЕ .....	33
3.2.3. ОПИСАНИЕ БАЗЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ .....	34
4. РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ .....	45
4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ .....	45
4.2. ВИД СТРАНИЦ ВЕБ-ПРИЛОЖЕНИЯ .....	53
5. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ВЕБ-ПРИЛОЖЕНИЯ .....	54
5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ .....	54
5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ .....	54
5.2.1. РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЯ .....	54
5.2.2. ПУБЛИКАЦИЯ ЗАПИСИ ПОЛЬЗОВАТЕЛЕМ .....	57
5.2.3. КОММЕНТИРОВАНИЕ ЗАПИСИ ПОЛЬЗОВАТЕЛЕМ .....	60
5.2.4. ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА .....	60
ЗАКЛЮЧЕНИЕ .....	61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	63
ПРИЛОЖЕНИЕ А .....	65
СНИМКИ СТРАНИЦ ВЕБ-ПРИЛОЖЕНИЯ .....	65
ПРИЛОЖЕНИЕ Б .....	79
ПРИМЕР ИСПОЛЬЗОВАНИЯ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА ВЕБ- ПРИЛОЖЕНИЯ .....	79
ПРИЛОЖЕНИЕ В .....	91
ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА UTIL .....	91

ПРИЛОЖЕНИЕ Г .....	103
ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА BACKEND .....	103
ПРИЛОЖЕНИЕ Д .....	105
ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА FORM.....	105
ПРИЛОЖЕНИЕ Е .....	107
ИСПОДНЫЙ КОД JAVASCRIPT-ФАЙЛА TOOLS .....	107
ПРИЛОЖЕНИЕ Ж .....	113
ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА HISTORY.....	113
ПРИЛОЖЕНИЕ З.....	115
ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА CANVAS .....	115
ПРИЛОЖЕНИЕ И .....	125
ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-DATA .....	125
ПРИЛОЖЕНИЕ К .....	135
ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-OBJECT .....	135
ПРИЛОЖЕНИЕ Л .....	136
ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-SELECT-OPTIONS .....	136

## **ВВЕДЕНИЕ**

Актуальность темы, связанной с благоустройством участка, заключается в том, что с помощью современных технологий, а также с помощью функционального оборудования и аппаратуры строительные дизайнеры могут создавать по-настоящему удивительные и неповторимые художественные проекты, которые станут отличным украшением для любого частного коттеджа. Стоит отметить, что создание таких проектов требует от специалиста достаточно серьезного внимания и особенно тщательного подхода, поскольку грамотно расположить все детали такого дизайнерского произведения невероятно сложно.

Перед созданием ландшафтных построек строительные дизайнеры, прежде всего, разрабатывают подробный проект и макет, благодаря которому они могут наглядно посмотреть и просчитать все особенности будущих работ. Для проведения более точных расчетов и изучений многие дизайнерские организации такой категории используют в своей работе специализированную строительную технику, а также множество компьютерных и программных приложений, которые позволяют им значительно ускорить и упростить рабочий процесс. Разумеется, в системе многих искусственных водопадов используются дорогостоящие и редкие породы дерева и камней, которые позволяют сделать ландшафтный проект по-настоящему уникальным. Как правило, все подобные детали обрабатываются специальными веществами, которые позволяют им достаточно долго сохранять свою целостность вне зависимости от разнообразных климатических изменений. Помимо искусственных водопадов на садовом участке также устанавливается множество архитектурных и скульптурных ансамблей, которые как раз и дополняют общую экстерьерную композицию.

Целью представленной выпускной квалификационной работы является разработка веб-приложения, обеспечивающего просмотр опубликованных статей посетителям ресурса, публикацию и редактирование статей для зарегистрированных пользователей, создание чертежа ландшафта садового участка с помощью графического онлайн-редактора веб-приложения.

Для достижения поставленной цели, необходимо решить следующие поставленные задачи:

- рассмотреть существующие аналоги, предоставляющие услуги просмотра, публикации, редактирования статей по теме благоустройства земельного участка;
- рассмотреть существующие аналоги, предоставляющие услуги по созданию чертежа ландшафта садового участка;
- провести анализ современных систем управления содержимым сайта для разработки веб-приложения просмотра, публикации, редактирования статей, создания чертежа ландшафта садового участка;
- разработать собственное веб-приложение на базе системы управления содержимым сайта “WordPress” и представить конечную ее реализацию;
- оценить работоспособность разработанного веб-приложения в различных условиях;
- произвести выборку результатов и проанализировать работу веб-приложения.

# **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Перед созданием ландшафтных построек строительные дизайнеры, прежде всего, разрабатывают подробный проект и макет, благодаря которому они могут наглядно посмотреть и просчитать все особенности будущих работ. Для проведения более точных расчетов и изучений многие дизайнерские организации такой категории используют в своей работе специализированную строительную технику, а также множество компьютерных и программных приложений, которые позволяют им значительно ускорить и упростить рабочий процесс. Разумеется, в системе многих искусственных водопадов используются дорогостоящие и редкие породы дерева и камней, которые позволяют сделать ландшафтный проект по-настоящему уникальным. Как правило, все подобные детали обрабатываются специальными веществами, которые позволяют им достаточно долго сохранять свою целостность вне зависимости от разнообразных климатических изменений. Помимо искусственных водопадов на садовом участке также устанавливается множество архитектурных и скульптурных ансамблей, которые как раз и дополняют общую экстерьерную композицию [1].

## **1.1. ОБЗОР АНАЛОГОВ**

Для создания дизайна ландшафта садового участка пользователь может использовать в качестве ресурсов какие-либо публикации на веб-ресурсах на тему дизайна и оформления участка, а также специальные программы, позволяющие проектировать чертеж ландшафта садового участка.

### **1.1.1. ОБЗОР АНАЛОГОВ СРЕДИ ВЕБ-СЕРВИСОВ**

Большое количество веб-сервисов, предоставляющий информацию о дизайне ландшафта садового участка это форумы, одностраничные веб-сервисы, либо веб-сервисы предоставляют далеко не полную информацию. В качестве аналогов были рассмотрены лишь некоторые из веб-сервисов.

Веб-сервис «Ландшафтный дизайн» [2] предоставляет широкий диапазон различных тематических публикаций на такие темы, как оформление участка, руководства по уходу за участком/насаждениями и пр.

В самих публикациях присутствуют некоторые фотографии общего вида участка, небольшие списки с элементами декора, которые можно приобрести в интернет-магазинах.

Однако стоит заметить, что, несмотря на подробное описание, в публикациях присутствуют далеко не все изображения объектов декора, на которые стоит обращать внимание при изучении веб-ресурсов, из-за чего пользователю приходится искать дополнительную информацию в Интернете. Также в списках предполагаемых объектов декора присутствуют не все объекты, из-за чего пользователю также приходится искать дополнительную информацию в Интернете.

Веб-сервис «Дизайн интерьера от НВ-Студио» [3] предоставляет услуги по проектированию ландшафта садового участка и располагает небольшим количеством публикаций на такие темы как «Ландшафтный дизайн», «Дизайн-проект интерьера» и др., которые не предоставляют особо полезной информации, которая необходима при проектировании ландшафта участка.

На веб-сервисе присутствуют примеры завершенных проектов с разрешения клиентов. Специалисты-дизайнеры предлагают проработку индивидуальных решений для дизайн-проектов участка по запросу заказчика. Также предлагают свои услуги по невысокой цене.

Однако стоит заметить, что сроки выполнения таких услуг – одна неделя. Также веб-сервис, в целом, предоставляет только услуги по проектированию. У обычного пользователя нет возможности посмотреть какие-либо публикации, в которых можно найти информацию, необходимую при проектировании ландшафта участка.

Преимущества и недостатки рассматриваемых аналогов среди веб-сервисов представлены в таблице 1.1.

Таблица 1.1 – Сравнительный анализ веб-сервисов-аналогов

<b>Веб-сервис</b>	<b>Преимущества</b>	<b>Недостатки</b>
«Ландшафтный дизайн»	Списки примерных объектов декора	Списки объектов декора неполные
	Подробное описание обустройства участка	Фотографии общего плана
«Дизайн ландшафта от НВ-Студио»	Примеры завершенных дизайн-проектов	Большой срок выполнения проекта
	Индивидуальные решения по просьбе заказчика	Отсутствие тематических публикации по проектированию дизайна участка
	Низкая стоимость оплаты услуг	Предоставляет только услуги по проектированию

### **1.1.2. ОБЗОР АНАЛОГОВ СРЕДИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Большое количество программного обеспечения для проектирования ландшафта является комплексным и позволяет, помимо проектирования ландшафта участка, проектировать интерьер помещений и внешний вид построек, и являются довольно сложными для понимания обычным пользователем. В качестве аналогов было выбрано программное обеспечение, которое позволяет спроектировать чертеж ландшафта садового участка.

Программное обеспечение “Garden Planner” [4] является простым для понимания обычным пользователем. Программа предоставляет на выбор более чем 1000 объектов-зеленых насаждений для размещения на чертеже, позволяет вносить коррективы с учетом климата, освещенности, температуры.

Однако данное программное обеспечение является условно-бесплатным и является англоязычным продуктом, что не делает его простым в использовании пользователями, которые не знают английский язык.

Итоговые преимущества и недостатки программного обеспечения “Garden Planner” перечислены в таблице 1.2.

Таблица 1.2 – Преимущества и недостатки программного обеспечения “Garden Planner”

<b>Программное обеспечение</b>	<b>Преимущества</b>	<b>Недостатки</b>
“Garden Planner”	Простота использования	Условно-бесплатный продукт
	Большая база зеленых насаждений	Англоязычный продукт
	Возможность выбора объектов под конкретные условия	

## **1.2. АНАЛИЗ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ**

### **1.2.1. АНАЛИЗ СИСТЕМ УПРАВЛЕНИЯ СОДЕРЖИМЫМ САЙТА**

Публикация каких-либо материалов на сайте подразумевает собой добавление содержимого, а его изменение/удаление – управление содержимым. На основании этого решено использовать в разработке веб-приложения систему управления содержимым сайта.

Выпускная квалифицированная работа является личным проектом студента и дальнейшее распространение в сети Интернет изначально не планируется, поэтому в качестве систем управления содержимым сайта выбираются для рассмотрения лишь те системы, которые соответствуют следующим критериям:

- свободное использование с любой целью;
- доступность исходного текста программы;
- свободное распространение копий программы.

Все критерии, названные выше, являются описанием свободно-распространяемых программ, которые именуются «Open-Source».

Система управления содержимым сайта “WordPress” является самой популярной свободной системой управления сайтом по версии интернет-ресурса «Рейтинг Рунета» [5]. Количество проектов, учитываемые рейтингом, и разработанные на данной системе – 13 557.

Преимущества:

- простая установка и настройка;
- простое управление содержимым сайта для пользователя, исполняющему функции администратора;
- легкая реализация сложных решений;
- поддержка RSS, trackback, Atom, pingback;
- поддержка веб-стандартов XHTML и CSS;
- подключаемые модули с простым взаимодействием с кодом;
- наличие русских официальных и неофициальных переводов;
- большая и доступная документация;
- большое активное сообщество.

Недостатки:

- нагрузка на сервер;
- дублирование изображений и страниц.

Система управления содержимым сайта “Drupal” является второй по популярности свободной системой управления сайтом по версии ресурса «Рейтинг Рунета» [5]. Количество проектов, учитываемые рейтингом, и разработанные на данной системе – 4 753.

Преимущества:

- простая установка и настройка;
- подключение модулей, которые постоянно обновляются разработчиками;
- большое активное сообщество.

Недостатки:

- сложное управление содержимым сайта для пользователя, исполняющему функции администратора;
- очень большая нагрузка на сервер;
- большинство модулей имеют недоработки.

Система управления содержимым сайта “Joomla!” является третьей по популярности свободной системой управления сайтом по версии ресурса

«Рейтинг Рунета» [5]. Количество проектов, учитываемые рейтингом, и разработанные на данной системе – 9 320.

Преимущества:

- простая установка и настройка;
- множество готовых решений для множества ситуаций.

Недостатки:

- очень большая нагрузка на сервер ввиду большого количества кода;
- невозможность оптимизации при проектировании больших веб-сервисов.

На основе рассмотренных систем управления содержимым сайта, в качестве базы для разработки в выпускной квалифицированной работы веб-сервиса была выбрана система «WordPress» ввиду большой и доступной документации, большого сообщества, а также ввиду простоты управления содержимым сайта пользователями, не имеющих отношения к разработке (администратор, модератор и пр.).

“WordPress” построен на базе архитектуры EDA (Event Driven Architecture – Событийно-ориентированная архитектура) и модели EAV (Entity Attribute Value – Модель «Сущность-Атрибут-значение»).

EDA обеспечивает взаимодействие между слабосвязанными между собой частями системы и дает гибкость в разработке [6]. С помощью этого паттерна реализованы так называемые фильтры (filters) и действия (actions), которые в “WordPress” принято называть «хуки» (hooks). Разработчик может создавать собственные точки срабатывания хуков и взаимодействовать с хуками определенными в ядре или плагинах, прикрепляя к ним произвольный код, таким образом влиять на поведение отдельных частей системы без необходимости их прямой модификации.

Модель EAV регулирует принцип хранения информации в БД, обеспечивая неизменность структуры таблиц при изменении структуры данных [7]. Все записи WP хранит в таблице “wp\_posts”, а дополнительные поля в таблице “wp\_postmeta” в виде: “meta\_id”, “post\_id”, “meta\_key”, meta\_value. И, если требуется добавить новый параметр к какому-либо типу поста, например размер к товару в интернет-

магазине, то нет необходимости создавать новые колонки в таблице, достаточно добавить строку в “wp\_postmeta” с соответствующими данными. Такой подход дает возможность быстрого расширения разнообразия данных, но работает несколько медленнее по сравнению с подходом с «широкими таблицами», где каждый параметр находится в своей собственной колонке. Однако скоростными характеристиками можно пренебречь при небольшом количестве хранимых строк или применить дополнительную оптимизацию если информации становится много.

### **1.2.2. ЯЗЫК ПРОГРАММИРОВАНИЯ БАЗЫ ДАННЫХ**

SQL (structured query language – «язык структурированных запросов») – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных [8].

Основные достоинства языка SQL заключаются в следующем:

- использование языка SQL в программах стандартизировано международными организациями;
- все распространенные СУБД используют SQL, т.к. реляционную базу данных можно перенести с одной СУБД на другую с минимальными доработками;
- СУБД может быть ориентирована на различные вычислительные системы, однако приложения, созданные с помощью SQL, допускают использование как для локальных БД, так и для крупных многопользовательских систем;
- SQL является языком реляционных БД, поэтому он стал популярным тогда, когда получила широкое распространение реляционная модель представления данных. Табличная структура реляционной БД хорошо понятна, а потому язык SQL прост для изучения;

- SQL обеспечивает пользователям немедленный доступ к данным, при этом в интерактивном режиме можно получить результат запроса за очень короткое время без написания сложной программы;
- язык SQL легко использовать в приложениях, которым необходимо обращаться к базам данных. Одни и те же операторы SQL употребляются как для интерактивного, так и программного доступа, поэтому части программ, содержащие обращение к БД, можно вначале проверить в интерактивном режиме, а затем встраивать в программу;
- с помощью SQL можно представить такую структуру данных, что тот или иной пользователь будет видеть различные их представления. Кроме того, данные из разных частей БД могут быть скомбинированы и представлены в виде одной простой таблицы, а значит, представления пригодны для усиления защиты БД и ее настройки под конкретные требования отдельных пользователей;
- язык SQL позволяет манипулировать структурой БД, тем самым обеспечивая гибкость с точки зрения приспособленности БД к изменяющимся требованиям предметной области;
- SQL – одно из лучших средств для реализации приложений на платформе клиент-сервер. SQL служит связующим звеном между взаимодействующей с пользователем клиентской системой и серверной системой, управляющей БД, позволяя каждой из них сосредоточиться на выполнении своих функций.

Язык SQL – первый и пока единственный стандартный язык для работы с базами данных, который получил достаточно широкое распространение. Практически все крупнейшие разработчики СУБД в настоящее время создают свои продукты с использованием языка SQL либо с SQL-интерфейсом. В него сделаны огромные инвестиции как со стороны разработчиков, так и со стороны пользователей. Он стал частью архитектуры приложений, является стратегическим выбором многих крупных и влиятельных организаций.

### 1.2.3. ЯЗЫК ПРОГРАММИРОВАНИЯ СЕРВЕРНОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЯ

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений [9]. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

Преимущества:

- является свободным программным обеспечением, распространяемым под особой лицензией (PHP license);
- легок в освоении на всех этапах;
- поддерживается большим сообществом пользователей и разработчиков;
- имеет развитую поддержку баз данных;
- имеется огромное количество библиотек и расширений языка;
- может использоваться в изолированной среде;
- предлагает нативные средства организации веб-сессий, программный интерфейс расширений;
- является довольно полной заменой проприетарной среды ASP (*Active Server Pages*) от Microsoft;
- может быть развернут почти на любом сервере;
- портирован под большое количество аппаратных платформ и операционных систем.

Недостатки:

- не подходит для создания десктопных приложений или системных компонентов;
- имеет слабые средства для работы с исключениями;
- глобальные параметры конфигурации влияют на базовый синтаксис языка, что затрудняет настройку сервера и разворачивание приложений;

- веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью.

Ruby — динамический, рефлексивный, интерпретируемый высокоуровневый язык программирования [10]. Язык обладает независимой от операционной системы реализацией многопоточности, сильной динамической типизацией, сборщиком мусора и многими другими возможностями.

Преимущества:

- открытая разработка;
- работает на многих платформах;
- может внедряться в HTML-разметку;
- относится к языкам программирования сверхвысокого уровня (VHLL), то есть обладает высоким уровнем абстракции и предметным подходом в реализации алгоритмов;
- реализует концептуально чистую объектно-ориентированную парадигму;
- предоставляет продвинутые методы манипуляции строками и текстом;
- легко интегрирует в свои программы высокопроизводительные серверы баз данных (DB2, MySQL, Oracle и Sybase);
- благодаря VHLL программы на Ruby хорошо масштабируются и легко сопровождаются;
- имеется простой программный интерфейс для создания многопоточных приложений;
- имеет продвинутые средства для работы с массивами;
- зарезервированные слова могут являться идентификаторами, если это не создаёт неоднозначности для парсера;
- дополнительные возможности для обеспечения безопасности;
- встроенный отладчик.

Недостатки:

- информационных ресурсов, посвящённых Ruby, недостаточно;
- Ruby менее производителен по сравнению со многими другими языками, применяемыми в веб-разработке;

- Ruby относительно медленно разрабатывается и развивается.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [11]. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Преимущества:

- открытая разработка;
- особенности синтаксиса стимулируют программиста писать хорошо читаемый код;
- предоставляет средства быстрого прототипирования и динамической семантики;
- имеет большое сообщество, позитивно настроенное по отношению к новичкам;
- множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
- механизмы модульности хорошо продуманы и могут быть легко использованы;
- абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту.

Недостатки:

- не слишком удачная поддержка многопоточности;
- на Python создано не так уж много качественных программных проектов по сравнению с другими универсальными языками программирования;
- изначальная ограниченность средств для работы с базами данных.

В качестве языка программирования на стороне backend был выбран язык PHP, так как выпускная квалифицированная работа является личным проектом студента и дальнейшее распространение в сети Интернет изначально не планируется, а также в качестве системы управления содержимым сайта была выбрана “WordPress”, которая написана на языке программирования PHP.

#### **1.2.4. ЯЗЫК ПРОГРАММИРОВАНИЯ КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ**

Язык JavaScript – это бесплатный язык сценариев, исполняемых на стороне клиента, который позволяет создавать интерактивные HTML-страницы [12]. «На стороне клиента» (client-side) означает, что JavaScript запускается в веб-браузере и не используется на стороне сервера. Сценарии на стороне клиента позволяют пользователю интерактивно взаимодействовать с веб-страницей после того, как она была обработана сервером и загружена веб-браузером.

JavaScript обеспечивает взаимодействие пользователя с веб-страницей после её загрузки. Исходя из этого, он обычно используется для решения следующих задач:

- динамическое добавление, редактирование и удаление HTML-элементов и их значений;
- проверка содержимого веб-форм перед отправкой на сервер;
- создание на компьютере клиента cookie-файлов для сохранения и получения данных при последующих визитах.

#### **1.2.5. JAVASCRIPT-БИБЛИОТЕКА ДЛЯ РЕАЛИЗАЦИИ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА**

Чертеж ландшафта садового участка представляет собой «плоский» 2D-рисунок с видом сверху, где показаны все размещенные объекты, формирующие дизайн ландшафта садового участка. Пример чертежа представлен на рисунке 1.1.

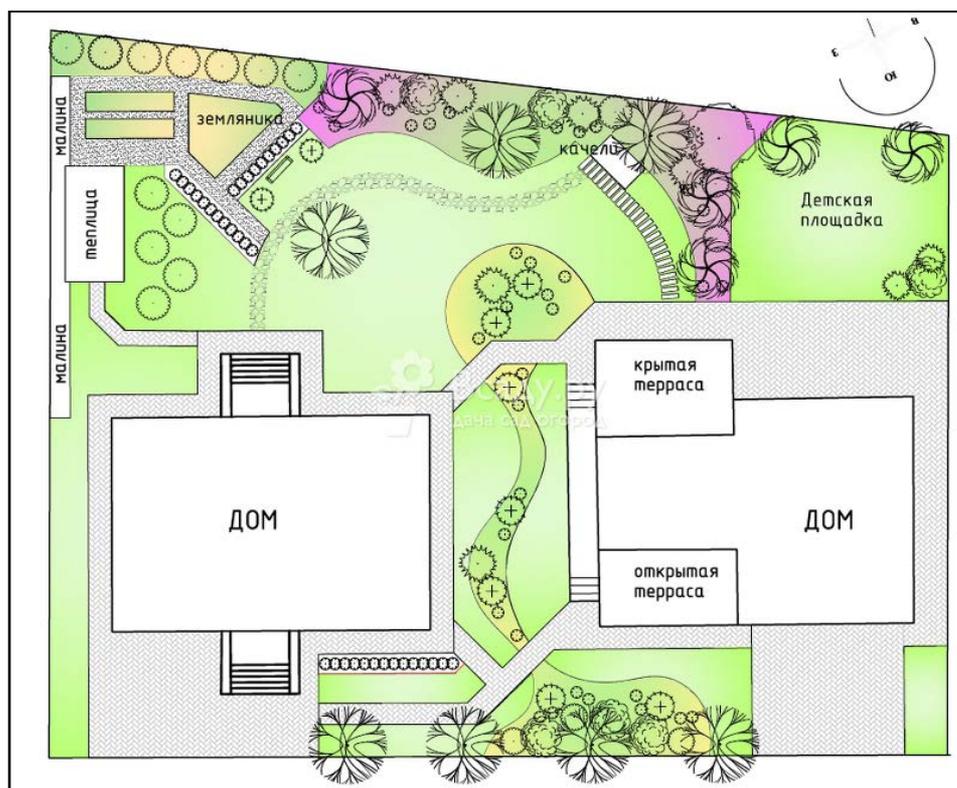


Рисунок 1.1 – Чертеж ландшафта садового участка

Стандартными средствами JavaScript HTML-элемент “canvas” используется для рисования простейшей графики. Любые динамические изменения на области отображения графической информации предусматривают собой стирание изображение и рисование нового. Редактор чертежа ландшафта садового участка предполагает помимо простейшей графики интерактивность объектов на холсте (перетаскивание/масштабирование объектов и пр.). На основании этого решено использовать JavaScript-библиотеку, позволяющую пользователю свободно изменять вид и состояние объектов на холсте и самого холста в целом.

Fabric.js – достаточно мощная JavaScript-библиотека для разработки веб-приложений, использующих такой элемент HTML5, как <canvas>. Как было сказано выше, стандартный API Canvas низкоуровневый и не позволяет создавать интерактивные графические веб-приложения. Использование Adobe Flash в данном случае нецелесообразно, так как это совершенно другая технология.

Fabric.js использует объектную модель, что означает, что все на холсте – объект и с ним можно взаимодействовать, не затрагивая весь холст глобально. К примеру, в интернет-магазине одежды можно добавить функцию редактирования

рисунка на заказываемой футболке. Fabric.js предоставляет данный функционал [13].

### **1.3. ВЫВОД**

Анализ рынка на наличие аналогов показало, что ни один существующий сервис не предоставляет полного функционала, позволяющего пользователям сформировать представление о дизайне ландшафта садового участка.

Для реализации веб-приложения просмотра/публикации тематических материалов, была выбрана система управления содержимым сайта, позволяющая разрабатывать веб-приложения с данным функционалом. Важным преимуществом системы управления сайтом является простота использования пользователями, не участвующих в разработке, а в управлении содержимым (например, модератор, который проверяет публикуемую статью на корректность). Также, помимо использования данной системы, в веб-приложении будет располагаться графический онлайн-редактор, позволяющий создавать чертеж ландшафта садового участка с учетом условий, заданных пользователем.

Для реализации серверной части приложения был выбран язык программирования PHP, для клиентской части – JavaScript, а также библиотека Fabric.js и технология AJAX для динамической отправки/получения данных с сервера без перезагрузки страницы. Помимо JavaScript для реализации пользовательского интерфейса используются HTML и CSS.

## **2. ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К ВЕБ-ПРИЛОЖЕНИЮ**

Для реализации веб-приложения используется следующий набор подсистем:

- HTTP-сервер Apache с веб-интерфейсом в виде набора PHP-скриптов. С помощью веб-сайта обеспечивается общее управление системой, создание, конфигурация, управление и получение результатов заданий и регистрация новых пользователей;
- MySQL база данных, обеспечивающая хранение данных о пользователях и других;
- система управления содержимым сайта, обеспечивающая разработку веб-приложения с необходимым функционалом;
- веб-браузер, позволяющий тестировать разрабатываемый веб-сервис.

### **2.1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

#### **2.1.1. В СОСТАВЕ ВЕБ-ПРИЛОЖЕНИЯ РЕАЛИЗОВАНЫ СЛЕДУЮЩИЕ ПОДСИСТЕМЫ**

- регистрация и авторизация пользователя;
- создание чертежа ландшафта садового участка с помощью специального редактора;
- публикация тематических материалов авторизованными пользователями.

#### **2.1.2. РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЯ ПРЕДОСТАВЛЯЕТ**

- возможность пользователю зарегистрироваться;
- возможность пользователю авторизоваться в системе;
- возможность восстановления пароля от учетной записи пользователя через электронную почту в случае его утери;
- изменение параметров учетной записи пользователя;
- изменение параметров профиля пользователя.

### **2.1.3. ПУБЛИКАЦИЯ ТЕМАТИЧЕСКИХ МАТЕРИАЛОВ АВТОРИЗОВАННЫМИ ПОЛЬЗОВАТЕЛЯМИ ПРЕДОСТАВЛЯЕТ**

- публикацию тематических материалов от имени пользователя;
- редактирование опубликованного материала его автором;
- проверка нового или отредактированного публикуемого материала и/или комментария на корректность модератором;
- удаление опубликованного материала его автором и/или представителем администрации;
- комментирование опубликованного материала пользователями;
- изменение комментария пользователя самим пользователем и/или представителем администрации;
- удаление комментария пользователя самим пользователем и/или представителем администрации;
- оценивание опубликованных материалов пользователем;
- отображение определенного числа опубликованных материалов на определенных страницах;
- авторизованным пользователям и посетителям сайта просматривать тематические материалы.

### **2.1.4. СОЗДАНИЕ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА С ПОМОЩЬЮ СПЕЦИАЛЬНОГО РЕДАКТОРА ПРЕДОСТАВЛЯЕТ**

- создание чертежа ландшафта садового участка пользователем с помощью редактора создания чертежа ландшафта садового участка, предоставляемого веб-приложением;
- отображение всех/и/или определенного количества чертежей ландшафта садового участка на главной странице.

### **2.1.5. СОЗДАНИЕ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА ПРЕДОСТАВЛЯЕТ**

- доступ к процессу создания чертежа ландшафта садового участка в специальном редакторе при указании требуемых начальных условий:
  - климатический пояс, в котором располагается садовый участок,
  - длина участка;
  - ширина участка.
- доступ к процессу создания чертежа ландшафта садового участка в специальном редакторе при указании необязательных условий по желанию:
  - требовательность растений к солнечному свету;
  - требуемый полив растений;
  - выносливость растений;
  - температура воздуха;
  - влажность воздуха.

### **2.1.6. В ПРОЦЕССЕ СОЗДАНИЯ ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА ПОЛЬЗОВАТЕЛЮ ДОСТУПНЫ**

- панель инструментов, в состав которой входят следующие пункты, при выборе которых выводится список объектов, доступных для размещения на области чертежа и которые соответствуют указанным пользователем начальным условиям:
  - цветы;
  - растения, произрастающие в водоемах;
  - деревья;
  - кустарники;
  - водоемы;
  - список всех объектов.
- панель инструментов-фигур, позволяющих размещать на чертеже геометрические фигуры:
  - прямоугольник;

- прямоугольник с заполненной областью-изображением (газон, грунт и тп.);
  - окружность;
  - прямая линия;
  - кривая линия (кисть);
  - текстовое поле.
- панель инструментов-действий, позволяющих глобально управлять областью чертежа и объектами на нем:
- выделение объектов;
  - отменить выбор объектов;
  - дублирование объектов;
  - перемещение объекта на передний план;
  - перемещение объекта на задний план;
  - отображение теней объектов;
  - удаление теней объектов;
  - отображения источника солнечного света;
  - удаление источника солнечного света;
  - удалить один выделенный объект или несколько выделенных объектов;
  - отменить действие;
  - повторить действие;
  - стереть область;
  - сохранить область.

## **2.2. НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

### **2.2.1. ТРЕБОВАНИЕ К КВАЛИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ**

- начальные навыки работы с персональным компьютером.

## **2.2.2. ТРЕБОВАНИЯ К ОБЩЕСИСТЕМНОМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ**

- возможность работы с помощью персонального компьютера или мобильного устройства;
- обеспечить возможность доступа для десктопных версий следующих браузеров:
  - Internet Explorer версии 11.0 или выше;
  - Microsoft Edge версии 17.0 или выше;
  - Firefox версии 63.0 или выше;
  - Chrome версии 70.0 или выше;
  - Safari версии 12.0 или выше;
  - Opera версии 56.0 или выше;
  - Yandex Browser версии 18.10 или выше.
- обеспечить возможность доступа для мобильных версий следующих браузеров:
  - Chrome for Android версии 70.0 или выше;
  - iOS Safari версии 12.1 или выше.

## 3. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

### 3.1. АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЯ

Архитектура веб-приложения представлена на рисунке 3.1.

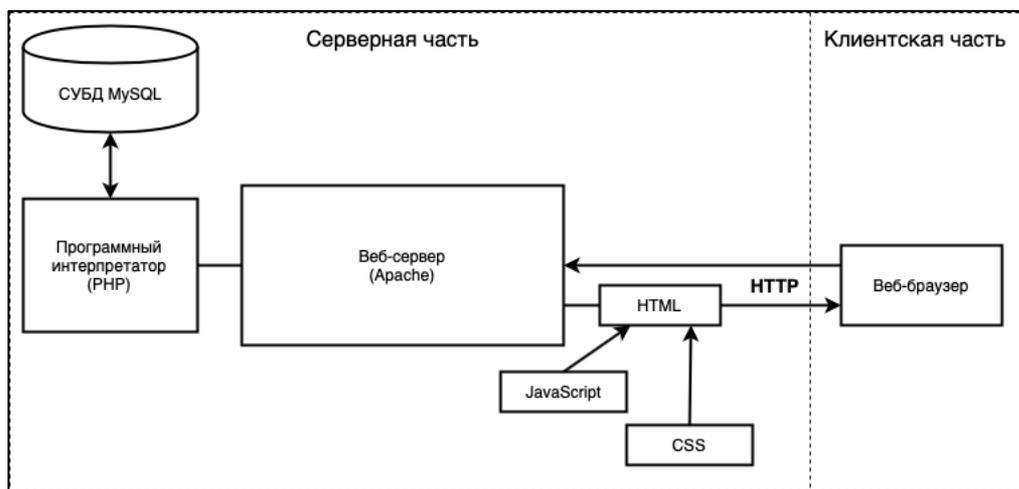


Рисунок 3.1 – Архитектура веб-приложения

Веб-приложение — это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети. Одним из достоинств такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами [14].

Структурно серверная часть веб-приложения состоит из трех крупных компонентов:

- веб-сервера (HTTP-сервера);
- сервера базы данных (БД);
- программного интерпретатора.

Веб-сервер является связующим звеном между веб-приложением и клиентом. Он принимает запросы от клиента в виде набора данных по протоколу HTTP и возвращает в ответ документы на языке HTML и файлы, которые требуются дополнительно (картинки, листы стилей, другие объекты).

Программный интерпретатор — это сердце системы. С помощью программного интерпретатора выполняется код, содержащий логику работы системы (алгоритмы). При разработке веб-приложения используются PHP. Интерпретируемые программы медленнее выполняются, но процесс их разработки значительно проще и быстрее. Учитывая постоянно повышающиеся требования бизнеса на скорость внедрения изменений в программные продукты (информационные системы), быстрый цикл разработки программ становится решающим фактором. Кроме того, мощность компьютеров постоянно возрастает, позволяя до определенного предела забывать о производительности приложений в угоду удобству и скорости разработки.

Сервер баз данных выполняет накопление структурированной информации, а также ее выдачу по запросам веб-приложения. От сервера баз данных зависят производительность приложения и функциональность с точки зрения данных. В качестве сервера баз данных в веб-приложении используется СУБД MySQL, имеющая серьезную репутацию в мире веб-разработок.

Центром клиентской части веб-приложения является браузер. Любой компьютер, подключенный к интернету, содержит программу-браузер, через которую можно начать работу с веб-приложением.

Со стороны клиентской части для придания интерактивности веб-страницам, помимо таблиц стилей CSS, к HTML-странице подключаются JavaScript-файлы.

## **3.2. ОПИСАНИЕ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ**

### **3.2.1. ВХОДНЫЕ ДАННЫЕ**

В качестве входных данных от пользователя в момент регистрации используются:

- имя пользователя;
- адрес электронной почты;
- пароль и его подтверждение.

В качестве входных данных от пользователя в момент авторизации или в случае утери пароля от учетной записи в системе используются:

- имя пользователя или адрес электронной почты;
- пароль.

В качестве входных данных от пользователя при изменении пароля от учетной записи в системе используются:

- текущий пароль;
- новый пароль и его подтверждение.

В качестве входных данных от пользователя во вкладке «безопасность» учетной записи в системе используются:

- указание, кто может видеть публичный профиль пользователя;
- указание, может ли отображаться профиль пользователя в списке пользователей.

В качестве входных данных от пользователя в случае удаления учетной записи пользователя из системы используется:

- пароль от учетной записи.

В качестве входных данных от пользователя при редактировании профиля используются:

- фотография обложки профиля;
- фотография-аватар профиля;
- небольшая информация о пользователе.

В качестве входных данных при создании и/или редактировании статьи от пользователя используются:

- заголовок статьи;
- рубрика статьи;
- фотография-миниатюра статьи;
- содержимое статьи. Используются текстовые и медиа-данные;
- ключевые слова.

В качестве входных данных при создании и/или редактировании комментария к записи от пользователя используется:

- текстовое сообщение.

В качестве входных данных при отправке формы обратной связи от пользователя или посетителя сайта используются:

- имя отправителя;
- адрес электронной почты отправителя;
- текстовое сообщение.

В качестве входных данных на странице с редактором чертежа ландшафта садового участка используются:

- указанный климатический пояс;
- указанное требование растений к солнечному свету;
- указанное требование к поливу растений;
- указанное требование к выносливости растений;
- температура;
- влажность воздуха.

### **3.2.2. ВЫХОДНЫЕ ДАННЫЕ**

В качестве выходных данных в веб-приложении используются HTML, CSS, JavaScript, PHP файлы, формирующие внешний вид страниц:

- главная страница;
- страница регистрации;
- страница восстановления пароля;
- страница входа в систему;
- страница архива публикаций;
- страница публикаций по рубрикам;
- страница создания публикации;
- страница редактирования публикации;
- страница учетной записи пользователя;
- страница профиля пользователя;
- страница редактора чертежа ландшафта садового участка.



Таблица 3.1 – Описание таблиц базы данных

Таблица	Данные
wp_termmeta	Метаданные терминов. Эта таблица дополняет таблицу “wp_terms”.
wp_terms	Элементы таксономии (термины) и базовая информация о них.
wp_term_taxonomy	Дополнительные данные об элементе таксономии, в частности важные из них - к какой таксономии относится термин (поле “taxonomy”) и с какой записью связан (поле “term_taxonomy_id”).
wp_term_relationships	Связи терминов с записями. Таблица показывает какая запись какому термину принадлежит. Только ID термина тут связывается через поле “term_taxonomy_id”.
wp_options	Основные опции и настройки сайта
wp_links	Ссылки сайта. На данный момент не используется
wp_users	Пользователи
wp_usermeta	Метаданные для каждого пользователя
wp_posts	Содержимое сайта
wp_postmeta	Данные о записях и страница сайта
wp_comments	Комментарии к записям сайта
wp_commentmeta	Метаданные комментариев к записям сайта
flower	Растения-цветы
bush	Растения-кустарники
tree	Растения-деревья
water_plant	Растения, произрастающие в водоемах
pound	Водоемы

Описание полей каждой таблицы представлено в таблицах 3.2 – 3.17.

Таблица 3.2 – Описание полей таблицы “wp\_termmeta”

Поле	Тип	Назначение	Индекс
meta_id	bigint(20)	ID метаполя	Основной
term_id	bigint(20)	ID термина	term_id (wp_terms)
meta_key	varchar(255)	ключ метаполя	meta_key
meta_value	longtext	Значение метаполя	Нет

Таблица 3.3 – Описание полей таблицы “wp\_terms”

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

term_id	bigint(20)	ID термина	Основной
name	varchar(200)	Название термина	name
slug	varchar(200)	Ярлык термина	slug
term_group	bigint(10)	Группа термина	Нет

Таблица 3.4 – Описание полей таблицы “wp\_term\_taxonomy”

Поле	Тип	Назначение	Индекс
term_taxonomy_id	bigint(20)	ID строки в таблице	Основной
term_id	bigint(20)	ID термина	term_id (wp_terms)
taxonomy	varchar(32)	Название таксономии	taxonomy
description	longtext	Описание термина	Нет
parent	bigint(20)	ID родительского термина (для древоводных таксономий)	Нет
count	bigint(20)	Количество записей в термине	Нет

Таблица 3.5 – Описание полей таблицы “wp\_term\_relationships”

Поле	Тип	Назначение	Индекс
object_id	bigint(20)	ID записи	Основной
term_taxonomy_id	bigint(20)	ID строки в таблице	term_taxonomy_id (wp_term_taxonomy)
term_order	int(11)	Порядок термина при прикреплении к записи	Нет

Таблица 3.6 – Описание полей таблицы “wp\_options”

Поле	Тип	Назначение	Индекс
option_id	bigint(20)	ID опции	Основной

Продолжение таблицы 3.6

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

option_name	varchar(191)	Название опции	option_name
option_value	longtext	Значение опции	Нет
autoload	varchar(20)	Автозагрузка	Нет

Таблица 3.7 – Описание полей таблицы “wp\_users”

Поле	Тип	Назначение	Индекс
ID	bigint(20)	ID пользователя	Основной
user_login	varchar(60)	Имя пользователя	user_login
user_pass	varchar(255)	Пароль пользователя	Нет
user_nicename	varchar(50)	Никнейм	user_nicename
user_email	varchar(100)	Адрес электронной почты	user_email
user_url	varchar(100)	Веб-сайт	Нет
user_registered	datetime	Дата регистрации	Нет
user_activation_key	varchar(255)	Ключ активации	Нет
user_status	int(11)	Статус	Нет
display_name	varchar(250)	Публичное имя	Нет

Таблица 3.8 – Описание полей таблицы “wp\_usermeta”

Поле	Тип	Назначение	Индекс
umeta_id	bigint(20)	ID метаполя	Основной
user_id	bigint(20)	ID пользователя	ID (wp_users)
meta_key	varchar(255)	Ключ метаполя	meta_key
meta_value	longtext	Значение метаполя	Нет

Таблица 3.9 – Описание таблицы “wp\_posts”

Поле	Тип	Назначение	Индекс
ID	bigint(20)	ID записи	Основной

Продолжение таблицы 3.9

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

post_author	bigint(20)	ID автора	ID (wp_users)
post_date	datetime	Дата создания	type_status_date
post_date_gmt	datetime	Дата создания	Нет
post_content	longtext	Содержимое	Нет
post_title	text	Название записи	Нет
post_excerpt	text	Анонс записи	Нет
post_status	varchar(20)	Статус записи (опубликовано, черновик и т.п.)	type_status_date
comment_status	varchar(20)	Разрешение комментариев	Нет
comment_status	varchar(20)	Разрешение комментариев	Нет
ping_status	varchar(20)	Статус разрешения отправки ping-запросов	Нет
post_password	varchar(20)	Пароль допуска	Нет
post_name	varchar(200)	Имя записи	Нет
to_ping	text	URL, на которые нужно отправлять ping при публикации	Нет
pinged	text	URL, куда ping был отправлен	Нет
post_modified	datetime	Дата последнего изменения	Нет
post_modified_gmt	datetime	Дата последнего изменения	Нет
post_content_filtered	longtext	Временное содержимое	Нет
post_parent	bigint(20)	ID родительской записи	post_parent

Продолжение таблицы 3.9

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

guid	varchar(255)	Уникальный идентификатор записи	Нет
menu_order	int(11)	Числовой порядок в меню	Нет
post_type	varchar(20)	Тип записи	type_status_date
post_mime_type	varchar(100)	МIME тип записи	Нет
comment-count	bigint(20)	Количество комментариев	Нет

Таблица 3.10 – Описание полей таблицы “wp\_postmeta”

Поле	Тип	Назначение	Индекс
meta_id	bigint(20)	ID метаданных	Основной
post_id	bigint(20)	ID записи	post_id (wp_posts)
meta_key	varchar(255)	Ключ метаполя	meta_key
meta_value	longtext	Значение метаполя	Нет

Таблица 3.11 – Описание полей таблицы “wp\_comments”

Поле	Тип	Назначение	Индекс
comment_ID	bigint(20)	ID комментария	Основной
comment_post_ID	bigint(20)	ID комментируемой записи	comment_post_id
comment_author	tinytext	Имя автора	Нет
comment_author_email	varchar(100)	Адрес электронной почты автора	comment_author_email
comment_author_url	varchar(200)	URL автора	Нет
comment_author_IP	varchar(100)	IP автора	Нет
comment_date	datetime	Дата комментирования	Нет
comment_date_gmt	datetime	Дата комментирования	comment_date_gmt
comment_post_ID	bigint(20)	ID комментируемой записи	comment_post_id

Продолжение таблицы 3.11

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

comment_content	text	Содержимое комментария	Нет
comment_karma	int(11)	Оценка комментария	Нет
comment_approved	varchar(20)	Статус комментария	comment_approved
comment_agent	varchar(255)	HTTP агент автора комментария	Нет
comment_type	varchar(20)	Тип комментария	comment_type
comment_parent	bigint(20)	ID родительского комментария	comment_parent
user_id	bigint(20)	ID автора комментария	ID (wp_users)

Таблица 3.12 – Описание полей таблицы “wp\_commentmeta”

Поле	Тип	Назначение	Индекс
meta_id	bigint(20)	ID метаполя	Основной
comment_id	bigint(20)	ID комментария	comment_id (wp_comments)
meta_key	varchar(255)	Ключ метаполя	meta_key
meta_value	longtext	Значение метаполя	Нет

Таблица 3.13 – Описание полей таблицы “flower”

Поле	Тип	Назначение	Индекс
flower_ID	int(255)	ID растения	Нет
name	varchar(255)	Наименование растения	Нет
picture	varchar(255)	Наименование изображения	Нет
icon	varchar(255)	Наименование иконки	Нет
climate	varchar(255)	Климат	Нет
ground	varchar(255)	Почва	Нет
endurance	varchar(255)	Выносливость	Нет
life_time	int(255)	Время жизни	Нет

Продолжение таблицы 3.13

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

air_humidity_min	int(255)	Минимальная влажность воздуха	Нет
air_humidity_max	int(255)	Максимальная влажность воздуха	Нет
light	varchar(255)	Потребность в солнечном свете	Нет
temperature_min	int(255)	Минимальная температура	Нет
temperature_max	int(255)	Максимальная температура	Нет
watering	varchar(255)	Потребность в поливе	Нет
plant_time	varchar(255)	Сезон посадки	Нет
pit_deep	int(255)	Глубина ямы посадки	Нет
description	varchar(1000)	Описание растения	Нет
remark	varchar(300)	Примечания	Нет

Таблица 3.14 – Описание полей таблицы “bush”

Поле	Тип	Назначение	Индекс
bush_ID	int(255)	ID растения	Нет
name	varchar(255)	Наименование растения	Нет
picture	varchar(255)	Наименование изображения	Нет
icon	varchar(255)	Наименование иконки	Нет
climate	varchar(255)	Климат	Нет
ground	varchar(255)	Почва	Нет
endurance	varchar(255)	Выносливость	Нет
life_time	int(255)	Время жизни	Нет
air_humidity_min	int(255)	Минимальная влажность воздуха	Нет
air_humidity_max	int(255)	Максимальная влажность воздуха	Нет

Продолжение таблицы 3.14

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

light	varchar(255)	Потребность в солнечном свете	Нет
temperature_min	int(255)	Минимальная температура	Нет
temperature_max	int(255)	Максимальная температура	Нет
watering	varchar(255)	Потребность в поливе	Нет
plant_time	varchar(255)	Сезон посадки	Нет
pit_deep	int(255)	Глубина ямы посадки	Нет
pit_radius	int(255)	Радиус ямы для посадки	Нет
description	varchar(1000)	Описание растения	Нет
remark	varchar(300)	Примечания	Нет

Таблица 3.15 – Описание полей таблицы “tree”

Поле	Тип	Назначение	Индекс
tree_ID	int(255)	ID растения	Нет
name	varchar(255)	Наименование растения	Нет
picture	varchar(255)	Наименование изображения	Нет
icon	varchar(255)	Наименование иконки	Нет
climate	varchar(255)	Климат	Нет
ground	varchar(255)	Почва	Нет
endurance	varchar(255)	Выносливость	Нет
life_time	int(255)	Время жизни	Нет
air_humidity_min	int(255)	Минимальная влажность воздуха	Нет
air_humidity_max	int(255)	Максимальная влажность воздуха	Нет
light	varchar(255)	Потребность в солнечном свете	Нет

Продолжение таблицы 3.15

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

temperature_min	int(255)	Минимальная температура	Нет
temperature_max	int(255)	Максимальная температура	Нет
watering	varchar(255)	Потребность в поливе	Нет
plant_time	varchar(255)	Сезон посадки	Нет
pit_deep	int(255)	Глубина ямы посадки	Нет
pit_radius	int(255)	Радиус ямы для посадки	Нет
distance	int(255)	Дистанция между соседним деревом	Нет
description	varchar(1000)	Описание растения	Нет
remark	varchar(300)	Примечания	Нет

Таблица 3.16 – Описание полей таблицы “water\_plant”

Поле	Тип	Назначение	Индекс
plant_ID	int(255)	ID растения	Нет
name	varchar(255)	Наименование растения	Нет
picture	varchar(255)	Наименование изображения	Нет
icon	varchar(255)	Наименование иконки	Нет
climate	varchar(255)	Климат	Нет
water_deep	int(255)	Глубина водоема	Нет
endurance	varchar(255)	Выносливость	Нет
life_time	int(255)	Время жизни	Нет
air_humidity_min	int(255)	Минимальная влажность воздуха	Нет
air_humidity_max	int(255)	Максимальная влажность воздуха	Нет

Продолжение таблицы 3.16

Поле	Тип	Назначение	Индекс
------	-----	------------	--------

light	varchar(255)	Потребность в солнечном свете	Нет
temperature_min	int(255)	Минимальная температура	Нет
temperature_max	int(255)	Максимальная температура	Нет
plant_time	varchar(255)	Сезон посадки	Нет
description	varchar(1000)	Описание растения	Нет
remark	varchar(300)	Примечания	Нет

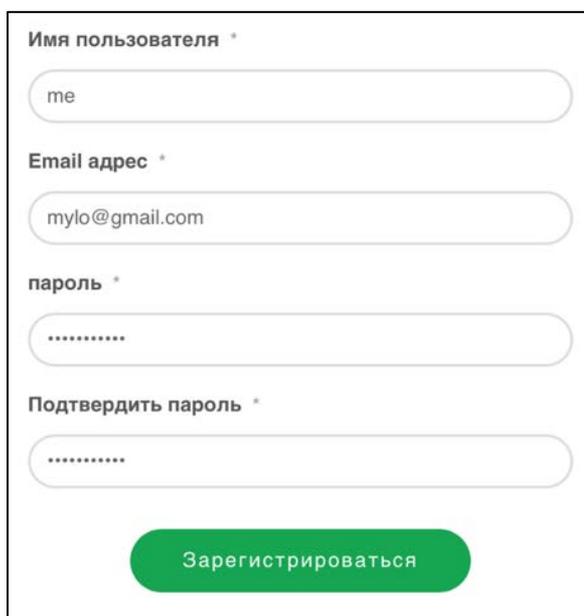
Таблица 3.17 – Описание полей таблицы “pound”

<b>Поле</b>	<b>Тип</b>	<b>Назначение</b>	<b>Индекс</b>
pound_ID	int(255)	ID водоема	Нет
name	varchar(255)	Наименование водоема	Нет
picture	varchar(255)	Наименование изображения	Нет
icon	varchar(255)	Наименование иконки	Нет
pound_deep	int(255)	Глубина водоема	Нет
pound_length	int(255)	Длина водоема	Нет
pound_width	int(255)	Ширина водоема	Нет
water_level	int(255)	Уровень воды	Нет
description	varchar(1000)	Описание	Нет
remark	varchar(300)	Примечание	Нет

## 4. РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ

### 4.1. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

Для того, чтобы пользователь мог публиковать/редактировать/комментировать записи, ему необходимо авторизоваться в системе. Если у пользователя нет учетной записи в системе, ему необходимо зарегистрироваться в системе. Экранная форма регистрации представлена на рисунке 4.1.



Имя пользователя \*

me

Email адрес \*

mylo@gmail.com

пароль \*

.....

Подтвердить пароль \*

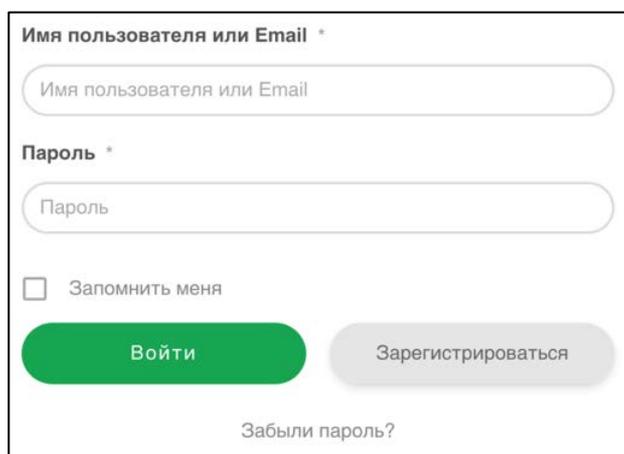
.....

Зарегистрироваться

Рисунок 4.1 – Экранная форма регистрации

Если все введенные данные корректны, система перенаправит пользователя на страницу с его профилем.

Экранная форма авторизации в системе приведена на рисунке 4.2.



Имя пользователя или Email \*

Имя пользователя или Email

Пароль \*

Пароль

Запомнить меня

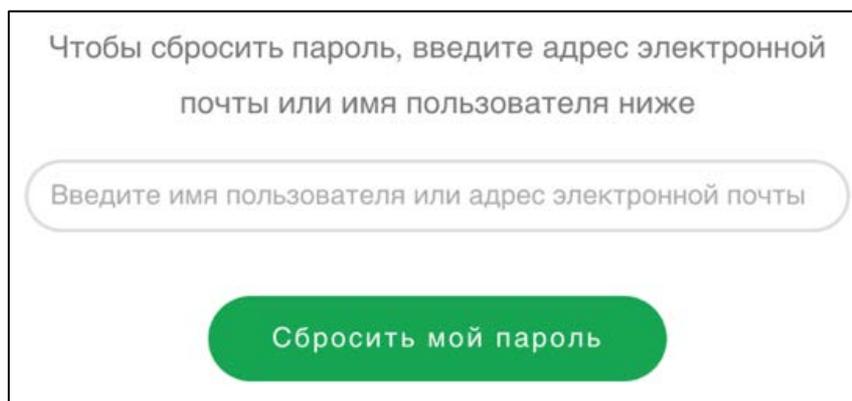
Войти

Зарегистрироваться

Забыли пароль?

Рисунок 4.2 – Экранная форма авторизации в системе

Если пользователь потерял/забыл пароль от своей учетной записи, он может указать адрес своей электронной почты, к которой привязана учетная запись, чтобы получить дальнейшие инструкции для сброса пароля. Экранная форма сброса пароля представлена на рисунке 4.3.



Чтобы сбросить пароль, введите адрес электронной почты или имя пользователя ниже

Введите имя пользователя или адрес электронной почты

Сбросить мой пароль

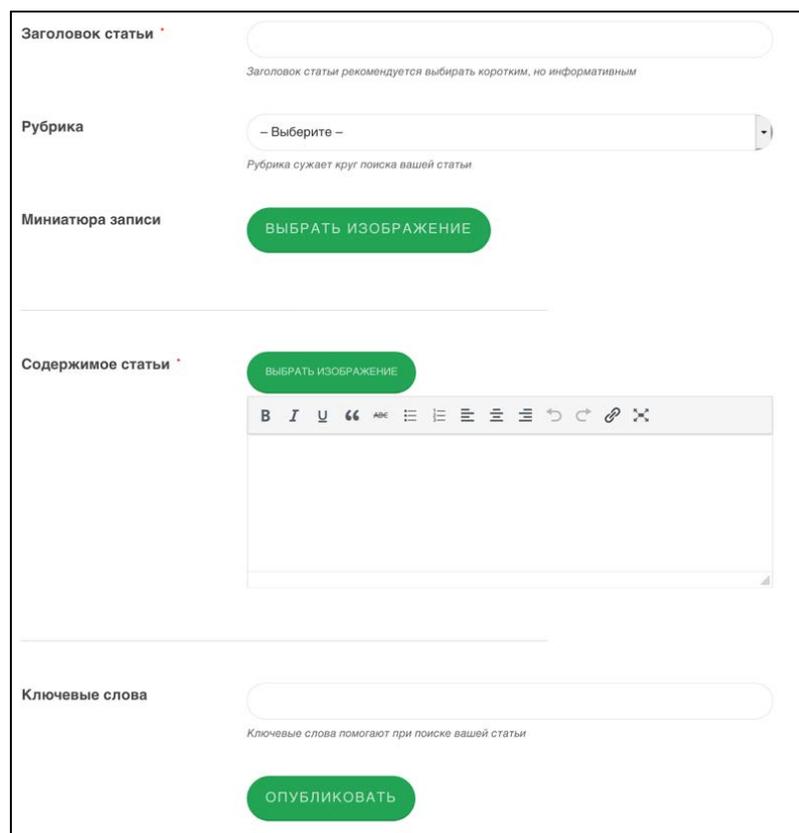
Рисунок 4.3 – Экранная форма восстановления пароля от учетной записи

Если неавторизованный пользователь попытается зайти на страницы, доступные только для авторизованных пользователей («Учетная запись», «Профиль» и пр.), то такие страницы будут для него недоступны. Пример вывода страницы «Опубликовать статью» для неавторизованного пользователя изображен на рисунке 4.4.

Страница недоступна. Необходимо войти в систему, чтобы опубликовать статью.

Рисунок 4.4 – Вывод страницы «Опубликовать статью» для неавторизованного пользователя

Для авторизованного пользователя экранная форма публикации статьи будет доступна в таком виде, как показано на рисунке 4.5.



Заголовок статьи \*  
Заголовок статьи рекомендуется выбирать коротким, но информативным

Рубрика  
– Выберите –  
Рубрика сужает круг поиска вашей статьи

Миниатюра записи  
ВЫБРАТЬ ИЗОБРАЖЕНИЕ

Содержимое статьи \*  
ВЫБРАТЬ ИЗОБРАЖЕНИЕ

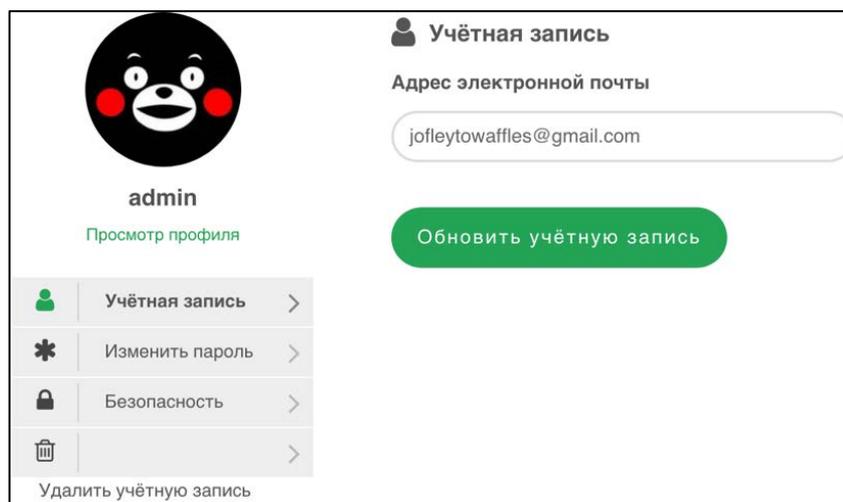
Ключевые слова  
Ключевые слова помогают при поиске вашей статьи

ОПУБЛИКОВАТЬ

Рисунок 4.5 – Экранная форма публикации статьи

Для публикации статьи необходимо заполнить поля «Заголовок статьи», «Содержимое статьи». После публикации статья будет находиться в базе данных со статусом «На одобрении». Модератор или администратор может изменить статус на «Опубликовано» в административной панели CMS “WordPress”.

На рисунках 4.6 – 4.9 представлены экранные формы изменения параметров учетной записи пользователя.



Учётная запись  
Адрес электронной почты  
jofleytowaffles@gmail.com  
Обновить учётную запись

admin  
Просмотр профиля

- Учётная запись >
- Изменить пароль >
- Безопасность >
- Удалить учётную запись

Рисунок 4.6 – Экранная форма изменения адреса электронной почты учетной записи

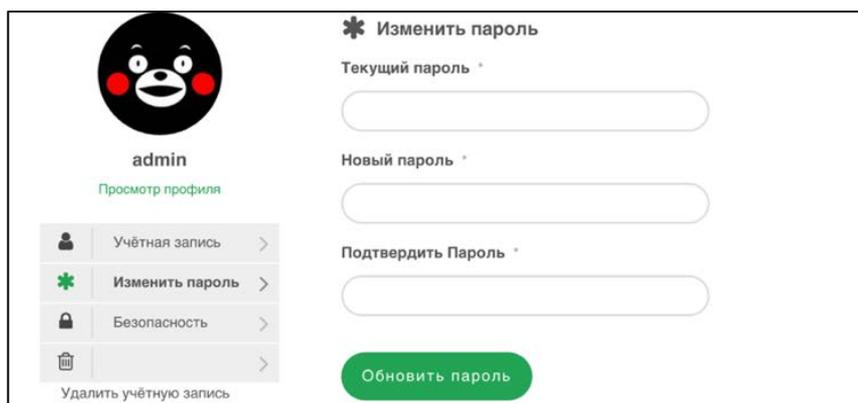


Рисунок 4.8 – Экранная форма изменения пароля учетной записи

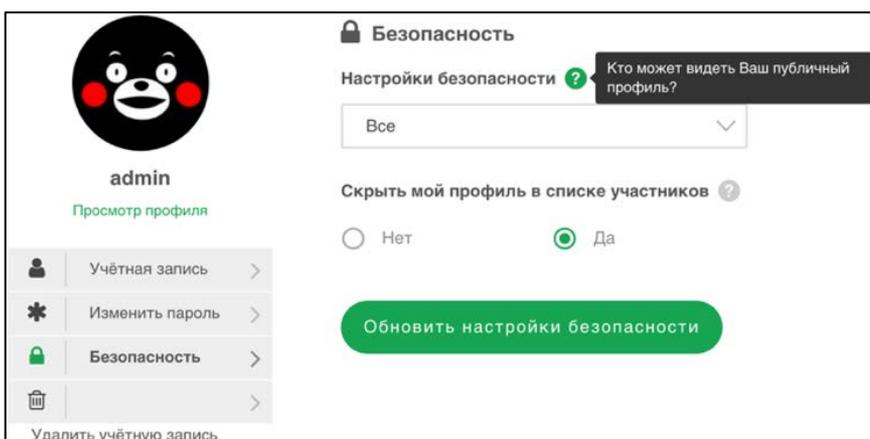


Рисунок 4.9 – Экранная форма изменения параметров отображения профиля в системе

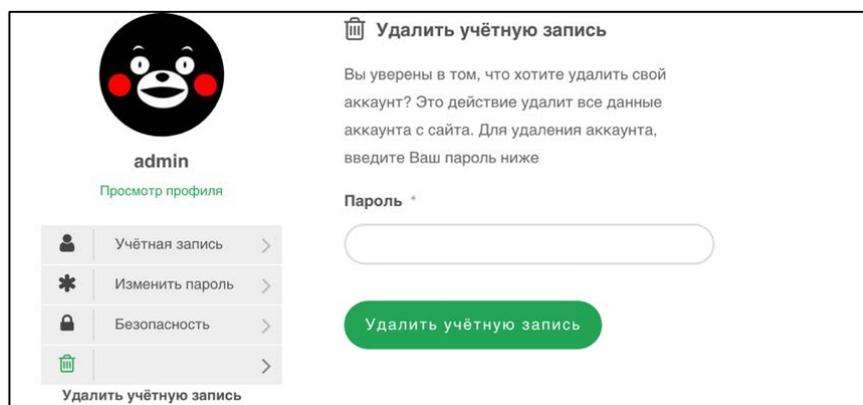


Рисунок 4.9 – Экранная форма удаления учетной записи

На рисунках 4.10 – 4.11 представлены снимки страницы профиля. Пользователь может изменить изображение обложки профиля, фотографию профиля, информацию о себе. Также в профиле отображаются все опубликованные им статьи, комментарии, их статус. Экранная форма публикации также доступна из профиля пользователя.

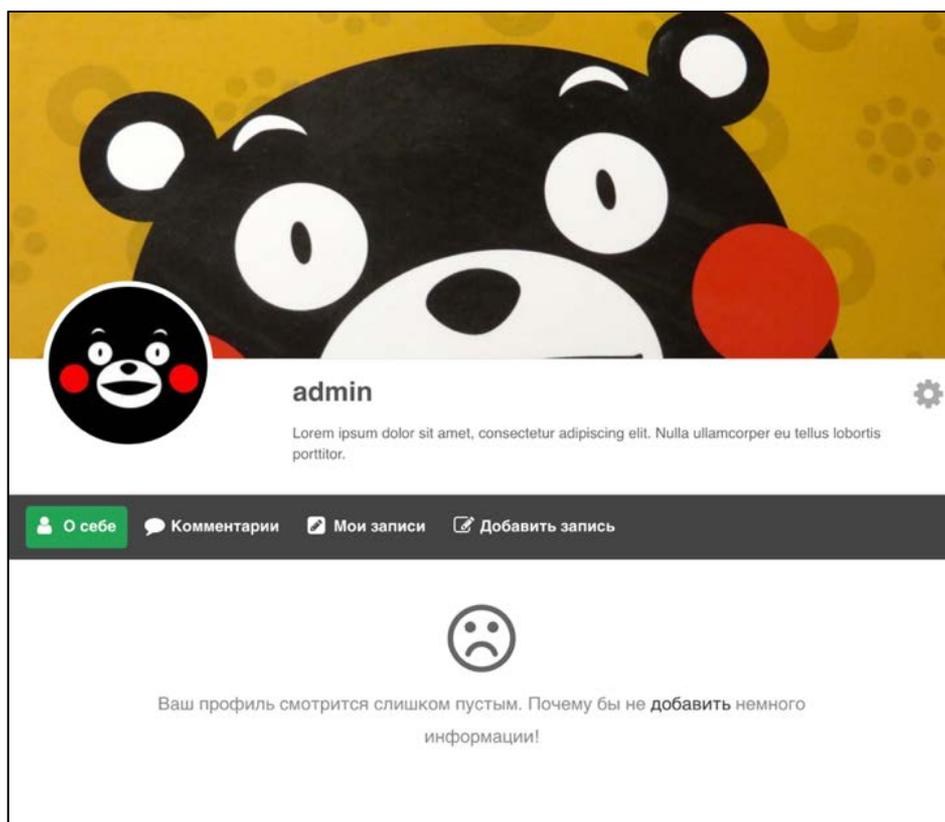


Рисунок 4.10 – Общий вид профиля пользователя

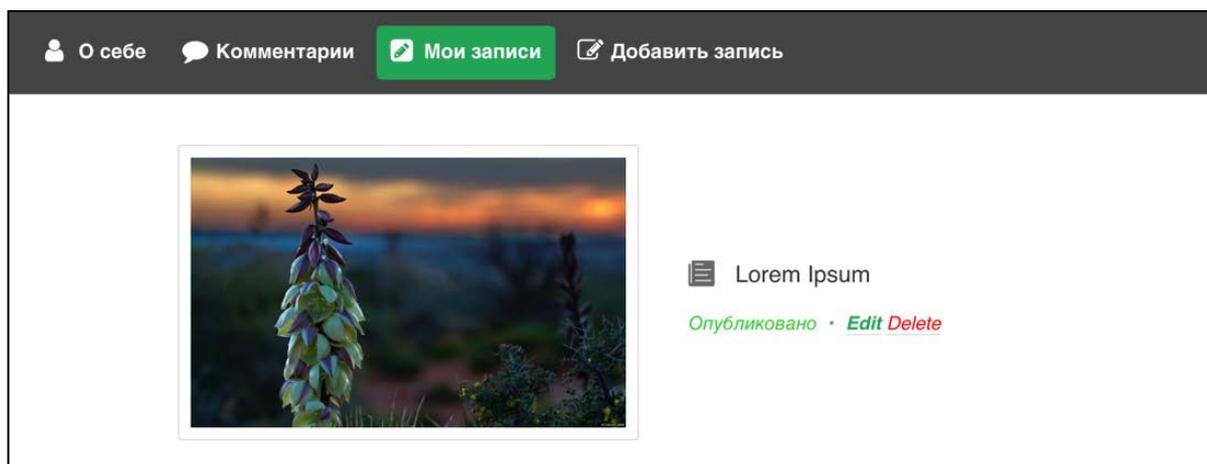


Рисунок 4.11 – Вкладка «Мои записи» в профиле пользователя

Все посетители и пользователи сайта могут создать свой собственный чертеж ландшафта садового участка с помощью специального графического редактора. Для начала работы пользователю необходимо отправить форму, которая задает параметры поиска всех «зеленых» объектов в базе данных, которые в результате добавляются в список.

Все варианты выбора для пунктов «Климат местности», «Солнечный свет», «Полив», «Выносливость растений» запрашиваются со стороны сервера.

Исходный код файла “form.js”, где показан запрос со стороны сервера, приведен в листинге Д.1 приложения Д. Исходный код файла “search-select-options.php”, где показана обработка запрашиваемых данных сервером, приведен в листинге Л.1 приложения Л. Исходный код файла “util.js”, где показана обработка полученных со стороны сервера данных, приведен в листинге В.1 приложения В.

Все данные, кроме «Длина участка», «Ширина участка», после отправки формы обрабатываются сервером и на их основе сервер возвращает веб-браузеру из базы данных все подходящие под условия объекты. Исходный код файла “form.js”, где показана отправка данных формы серверу, приведен в листинге Д.1 приложения Д. Исходный код файла “search-data.php”, где показана обработка данных формы сервером, приведен в листинге И.1 приложения И. Исходный код файла “util.js”, где показана обработка возвращенных данных со стороны сервера, приведен в листинге В.1 приложения В.

Вид экранной формы выбора начальных условий представлен на рисунке 4.12.

**Перед началом работы с редактором необходимо указать следующие параметры**

Основные параметры

Климат местности: Любой

Длина участка: 1000

Ширина участка: 500

Дополнительные параметры

Солнечный свет: Любой

Полив: Любой

Выносливость растений: Любой

Максимальная температура:

Максимальная влажность воздуха:

**НАЧАТЬ РЕДАКТИРОВАНИЕ** **СБРОСИТЬ ВСЕ**

Редактор доступен только в десктопной версии сайта

Рисунок 4.12 – Экранная форма выбора начальных условий

После отправки формы пользователю будет доступен графический редактор, вид которого представлен на рисунке 4.13.

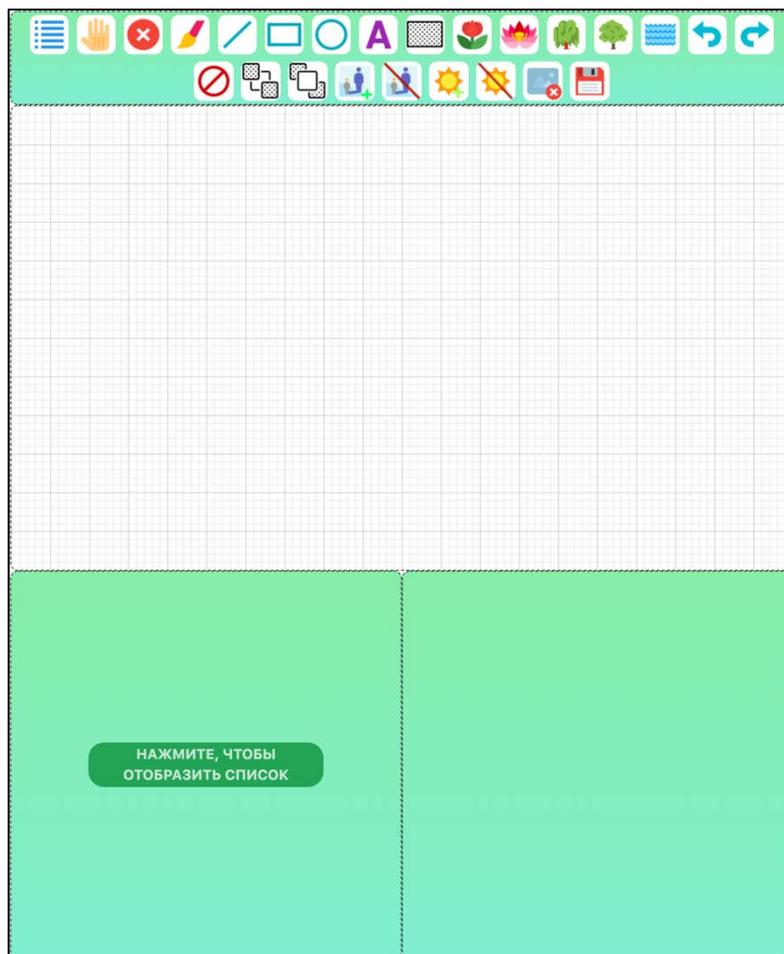


Рисунок 4.13 – Вид редактора чертежа ландшафта садового участка

На рисунке 4.14 представлен выводимый список объектов согласно сформированному из отправленной формы запросу.

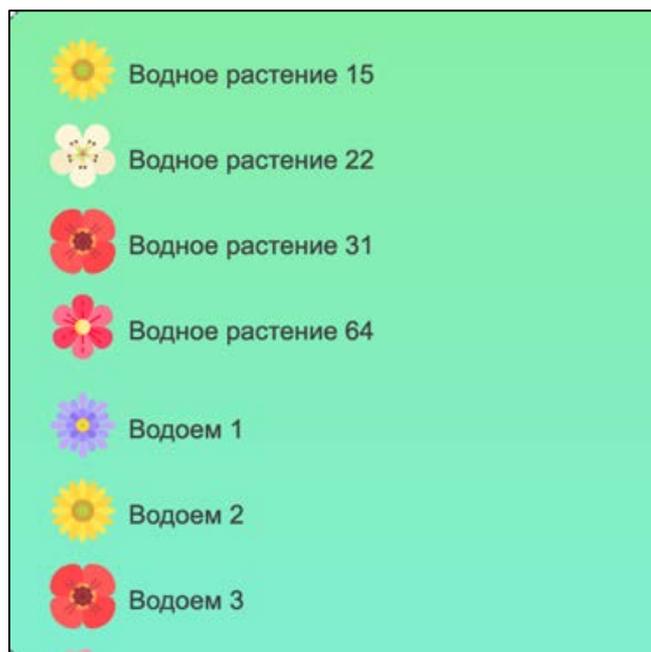


Рисунок 4.14 – Список выводимых объектов

Используя набор инструментов редактора, пользователь может создать изображение по своему усмотрению. Пользователь может любой добавленный на холст объект повернуть, изменить его размеры, изменить его масштаб, изменить его цвет контура/заливки/текстового поля, удалить объект(-ы), очистить холст, отменить действие по кнопке и по сочетанию клавиш, повторить отмененное действие по кнопке и по сочетанию клавиш, выбрать объекты на холсте, копировать объект(-ы), вывести список объектов, сгруппированных по типу «цветы»/«кустарники»/«деревья»/«водные растения»/«водоемы».

Исходный код файла “canvas.js”, где показан функционал инструментов редактора, приведен в листинге 3.1 приложения 3. Исходный код файла “tools.js”, где показаны вспомогательные функции редактора, приведен в листинге E.1 приложения E. Исходный код файла “history.js”, где показаны функции управления состоянием редактируемой области, приведен в листинге Ж.1 приложения Ж.

По клику на название объекта в списке пользователь может увидеть карточку объекта, где показаны наименование объекта, изображение объекта, его свойства (климат, грунт, время посадки и пр.), его описание и примечание, если таковое имеется.

Исходный код файла “canvas.js”, где показан запрос данных об объекте на сторону сервера, приведен в листинге 3.1 приложения 3. Исходный код файла “search-object.php”, где показана обработка запроса сервером, приведен в листинге K.1 приложения K. Исходный код файла “util.js”, где показана обработка полученных данных со стороны сервера, приведен в листинге B.1 приложения B.

Исходный код файла “backend.js”, где показаны все функции формирования запросов на сторону сервера, приведен в листинге Г.1 приложения Г.

Пример карточки с информацией об объекте представлен на рисунках 4.15 – 4.16.

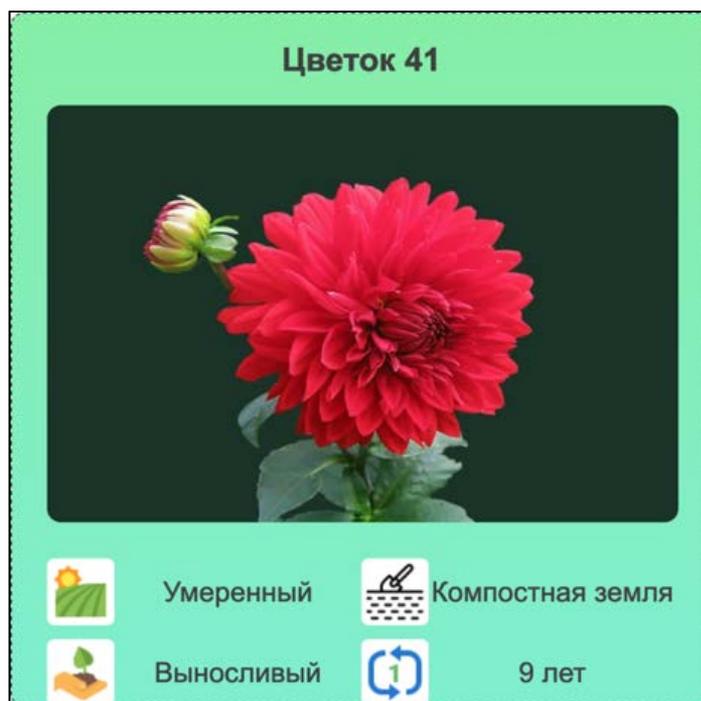


Рисунок 4.15 – Начало карточки с информацией об объекте

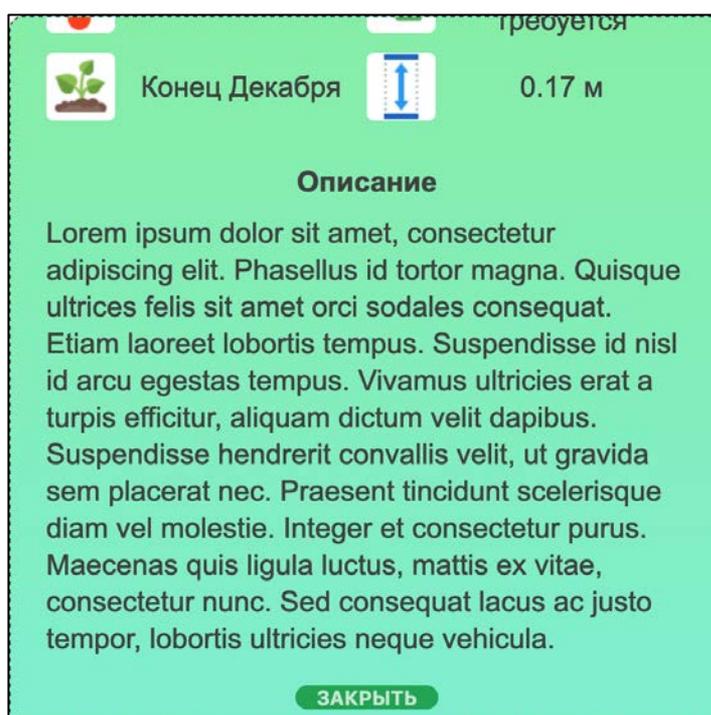


Рисунок 4.16 – Конец карточки с информацией об объекте

## 4.2. ВИД СТРАНИЦ ВЕБ-ПРИЛОЖЕНИЯ

Снимки страниц веб-приложения изображены на рисунках А.1 – А.14 приложения А.

## 5. ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ВЕБ-ПРИЛОЖЕНИЯ

### 5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ

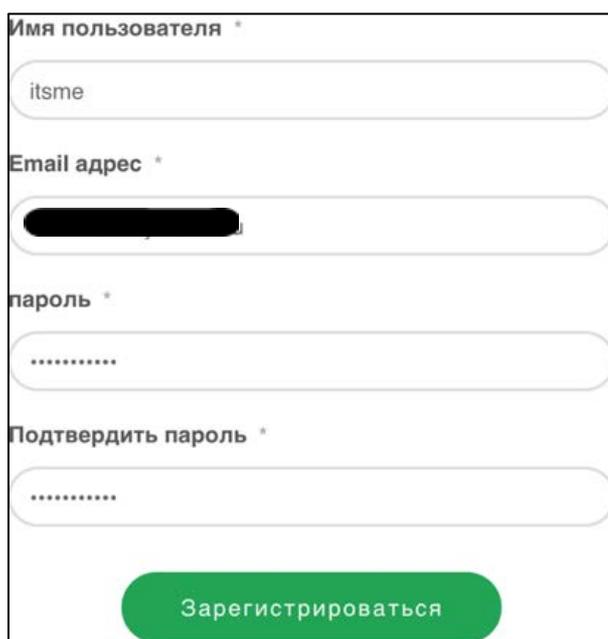
На текущий момент разработки веб-приложение прошло «альфа»-тестирование, то есть протестировано разработчиком, так как веб-приложение разрабатывается и находится на локальном компьютере.

Таким образом, производится системное тестирование веб-приложения.

### 5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ

#### 5.2.1. РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЯ

Заполнение экранной формы регистрации нового пользователя приведена на рисунке 5.1.



The image shows a registration form with the following fields and elements:

- Field: "Имя пользователя \*" (User name) with the value "itsme".
- Field: "Email адрес \*" (Email address) which is redacted with a black bar.
- Field: "пароль \*" (password) with masked characters ".....".
- Field: "Подтвердить пароль \*" (Confirm password) with masked characters ".....".
- Button: "Зарегистрироваться" (Register).

Рисунок 5.1 – Экранная форма регистрация нового пользователя

После успешной регистрации пользователь перенаправляется на страницу с его профилем.

Заполнение экранной формы изменения пароля от учетной записи представлено на рисунках 5.2. Уведомление об успешном изменении пароля представлено на рисунке 5.3.

Рисунок 5.2 – Заполнение экранной формы изменения пароля от учетной записи

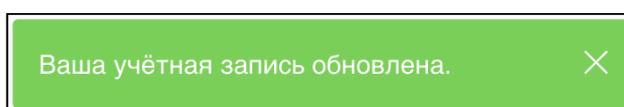
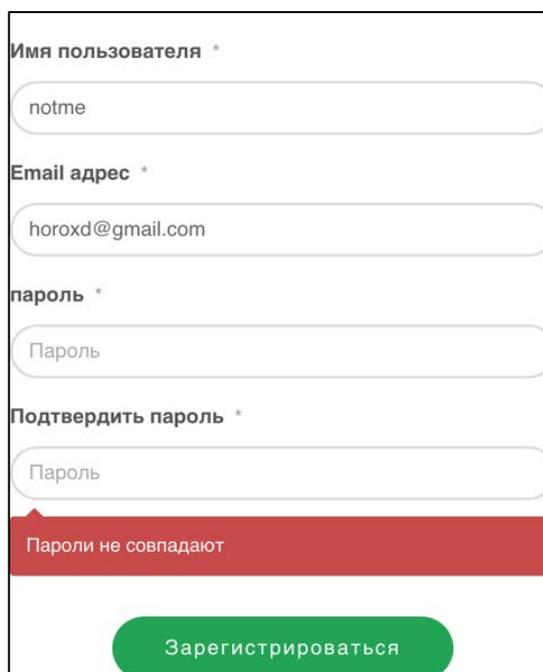


Рисунок 5.3 – Оповещение об изменении пароля

Посетитель сайта не может зарегистрироваться с уже использующейся электронной почтой. Пример заполнения экранной формы регистрации с уже использующейся электронной почтой представлен на рисунке 5.4.

Рисунок 5.4 – Заполнение экранной формы регистрации с уже использующейся электронной почтой

При вводе в поле «Подтвердите пароль» неправильного подтверждающего пароля, поля «Пароль» и «Подтвердите пароль» сбросятся и посетитель увидит сообщение, вид которого представлен на рисунке 5.5.



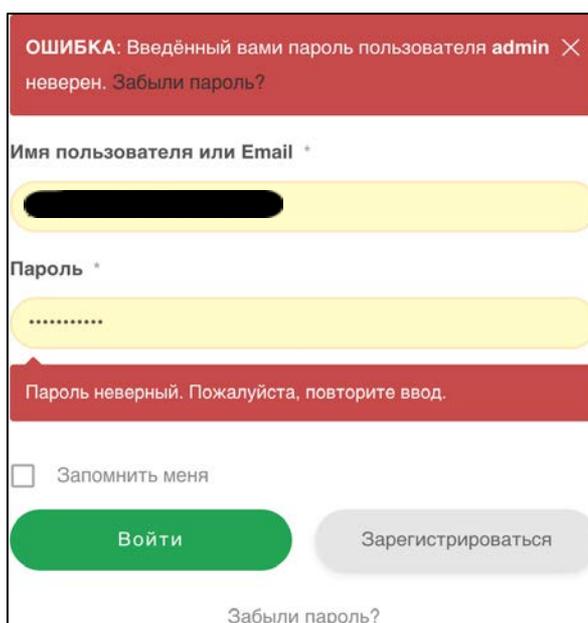
The screenshot shows a registration form with the following fields and values:

- Имя пользователя \*: notme
- Email адрес \*: horoxd@gmail.com
- пароль \*: Пароль
- Подтвердить пароль \*: Пароль

A red error message is displayed below the password fields: "Пароли не совпадают". At the bottom of the form is a green button labeled "Зарегистрироваться".

Рисунок 5.5 – Заполнение экранной формы регистрации с неправильным повтором пароля

При вводе неправильного пароля при авторизации в системе, пользователь увидит сообщение, вид которого представлен на рисунке 5.6.



The screenshot shows a login form with the following fields and values:

- Имя пользователя или Email \*: [Redacted]
- Пароль \*: [Redacted]

A red error message is displayed at the top: "ОШИБКА: Введённый вами пароль пользователя admin неверен. Забыли пароль?". Below the password field, another red message says: "Пароль неверный. Пожалуйста, повторите ввод.". At the bottom, there is a checkbox for "Запомнить меня", a green button "Войти", and a grey button "Зарегистрироваться". A link "Забыли пароль?" is located at the very bottom.

Рисунок 5.6 – Заполнение экранной формы авторизации с неправильным паролем

## 5.2.2. ПУБЛИКАЦИЯ ЗАПИСИ ПОЛЬЗОВАТЕЛЕМ

Процесс публикации статьи приведен на рисунках 5.7 – 5.11.

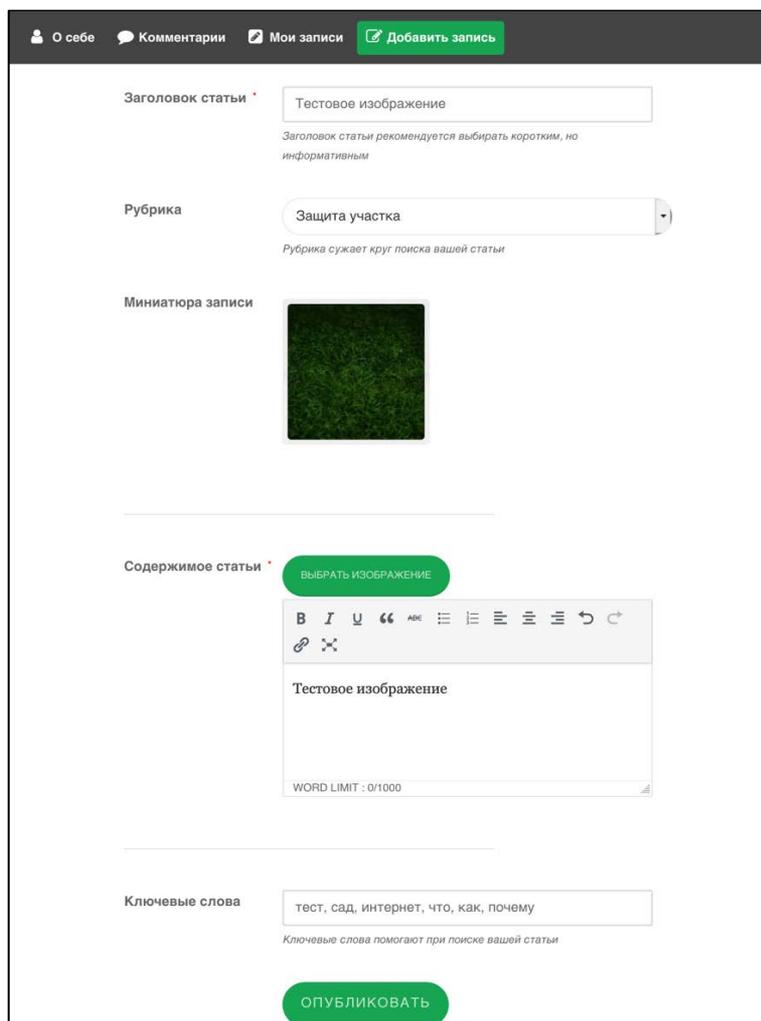


Рисунок 5.7 – Заполнение экранной формы публикации статьи



Рисунок 5.8 – Уведомление об успешной отправке экранной формы публикации статьи

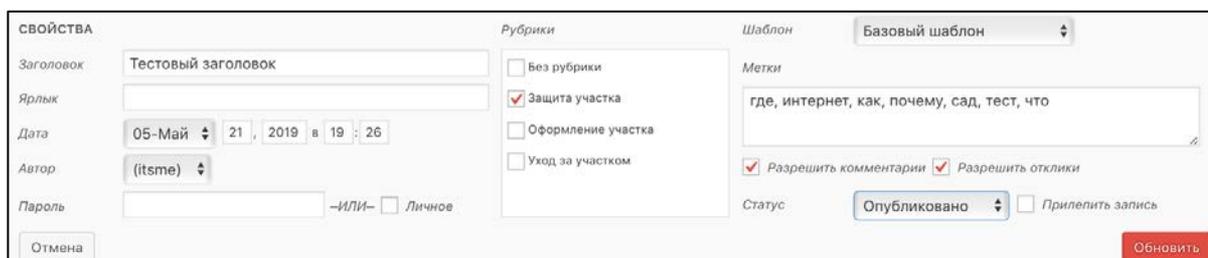


Рисунок 5.9 – Изменение статуса статьи модератором или администратором с помощью экранной формы «Свойства» в административной панели “WordPress”

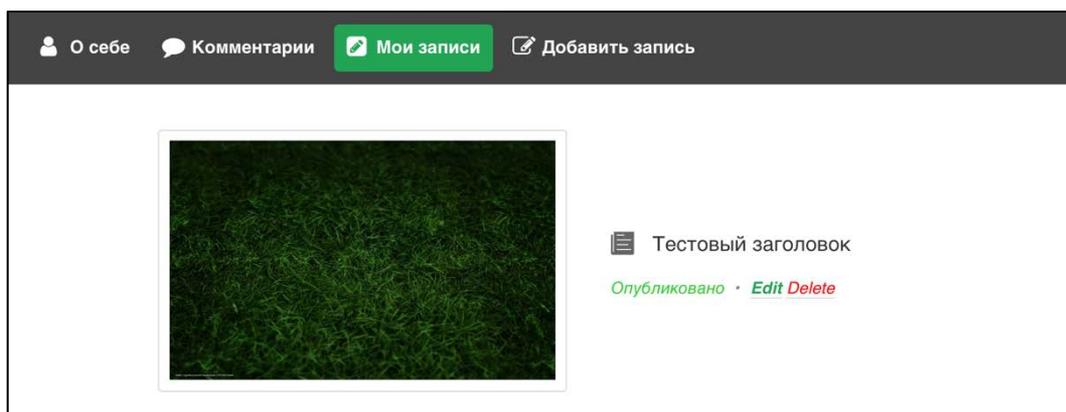


Рисунок 5.10 – Отображения статуса статьи в профиле пользователя

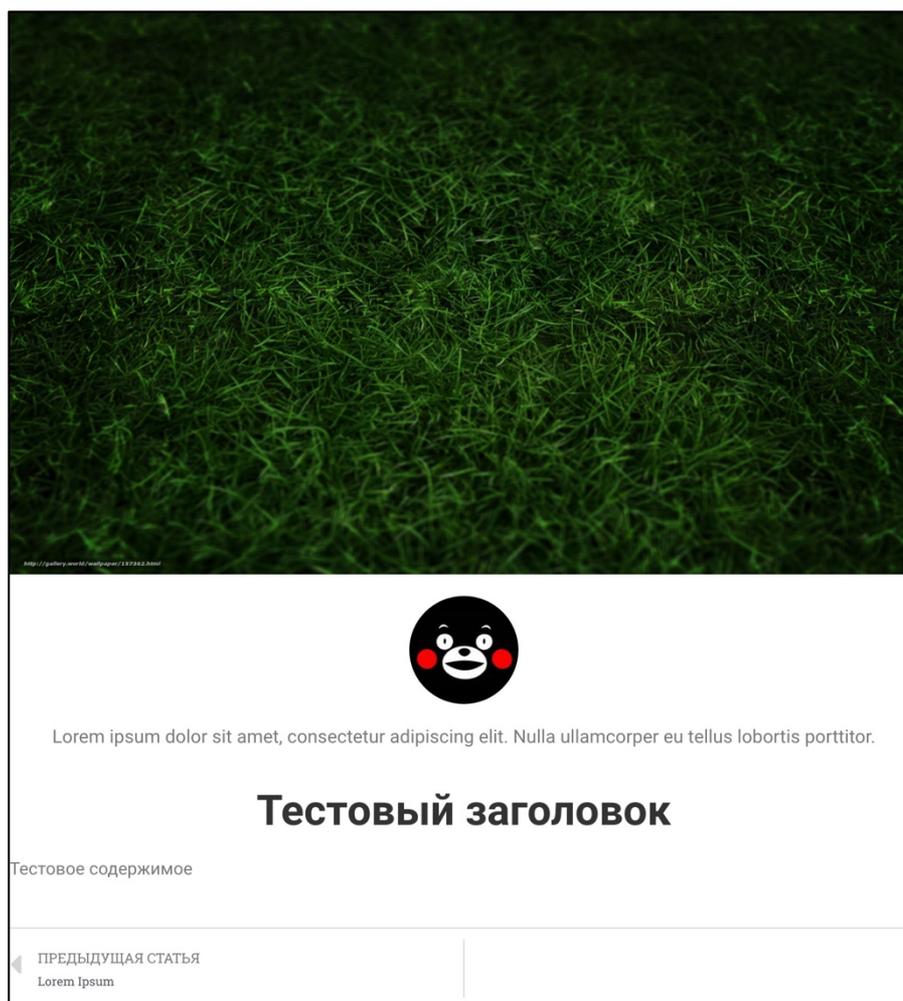


Рисунок 5.11 – Вид опубликованной статьи в системе

В экранной форме публикации статьи поля «Заголовок статьи» и «Содержимое статьи» являются обязательными и любое пустующее из них выведет пользователю ошибку, изображенную на рисунках 5.12 – 5.13.

The screenshot shows a web form for publishing an article. The fields and their states are as follows:

- Заголовок статьи:** A red error banner above the input field states "Заголовок статьи необходим" (Article title is required) and "Заголовок статьи рекомендуется выбирать коротким, но информативным" (Article title is recommended to be short but informative). The input field is empty.
- Рубрика:** A dropdown menu with the selected option "– Выберите –". Below it, a note says "Рубрика сужает круг поиска вашей статьи" (Category narrows the search for your article).
- Миниатюра записи:** A green button labeled "ВЫБРАТЬ ИЗОБРАЖЕНИЕ" (SELECT IMAGE).
- Содержимое статьи:** A red error banner above the text editor states "Содержимое статьи" (Article content). The text editor has a toolbar with icons for bold, italic, underline, quote, bulleted list, numbered list, link, and unlink. The text "Alo" is entered. Below the editor, a "WORD LIMIT" indicator shows "1/1000".
- Ключевые слова:** An input field containing the text "alo". Below it, a note says "Ключевые слова помогают при поиске вашей статьи" (Keywords help in searching for your article).
- Buttons:** A green "ОПУБЛИКОВАТЬ" (PUBLISH) button is located at the bottom of the form.
- Footer:** A red banner at the bottom of the form contains the text "Пожалуйста, исправьте некорректные данные" (Please correct the incorrect data).

Рисунок 5.12 – Экранная форма публикации статьи. Незаполненное поле «Заголовок статьи»

The screenshot shows the same article submission form, but with the content field highlighted in red to indicate an error:

- Заголовок статьи:** The input field now contains the text "Alo". The error banner is no longer present.
- Рубрика:** The dropdown menu remains "– Выберите –".
- Миниатюра записи:** The "ВЫБРАТЬ ИЗОБРАЖЕНИЕ" button is present.
- Содержимое статьи:** The text editor toolbar is visible. The text "Alo" is present. The "WORD LIMIT" indicator now shows "0/1000". The entire content area is highlighted with a red background.
- Ключевые слова:** The input field contains "alo".
- Buttons:** The "ОПУБЛИКОВАТЬ" button is present.
- Footer:** The red banner at the bottom still contains "Пожалуйста, исправьте некорректные данные".

Рисунок 5.13 – Экранная форма публикации статьи, незаполненное поле «Содержимое статьи»

### 5.2.3. КОММЕНТИРОВАНИЕ ЗАПИСИ ПОЛЬЗОВАТЕЛЕМ

Процесс комментирования статьи представлен на рисунках 5.14 – 5.17.

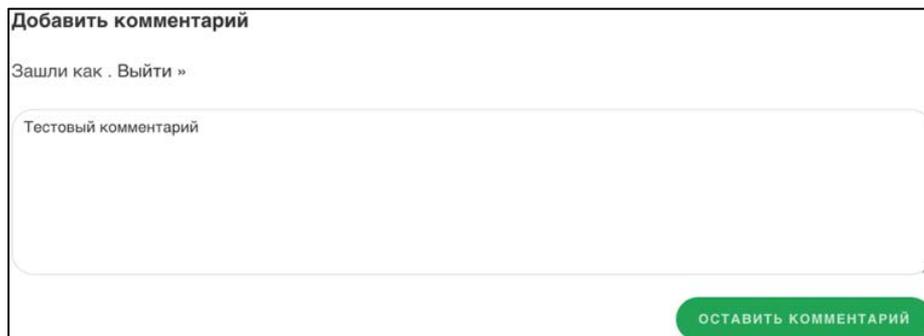


Рисунок 5.14 – Экранная форма комментирования статьи пользователем



Рисунок 5.15 – Оповещение об одобрении комментария пользователя



Рисунок 5.16 – Одобрение комментария пользователя модератором или администратором

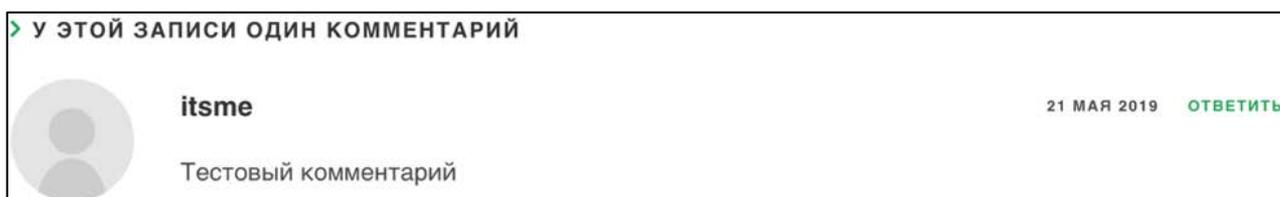


Рисунок 5.17 – Вид одобренного модератором комментария пользователя

### 5.2.4. ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА ЧЕРТЕЖА ЛАНДШАФТА САДОВОГО УЧАСТКА

Пример использования редактора графического онлайн-редактора чертежа ландшафта садового участка показан на рисунках Б.1 – Б.12 приложения Б.

## ЗАКЛЮЧЕНИЕ

В представленной работе были разобраны все основные темы, связанные с проектированием, разработкой и вводом в эксплуатацию веб-приложения для удовлетворения потребностей пользователей. Обозначены задачи, которые должны быть решены. Создано собственное веб-приложение, позволяющее:

- пользователям и посетителям, интересующимся садоводством или дизайном своего садового участка, публиковать/редактировать/комментировать свои статьи;
- создавать свой чертеж ландшафта садового участка, что позволяет им определиться с дизайном ландшафта своего садового участка.

На стадии разработки, внедрения и эксплуатации использовалась бесплатная система управления содержимым сайта “WordPress”. Озвучены причины, по которым производилась разработка веб-приложения на базе “WordPress”. Так же в ходе анализа “WordPress” выяснилось, что она обладает рядом неприятных особенностей. В частности, ее отрицательное влияние на скорость работы системы ввиду большого количества исходного кода. Помимо этого, данная система управления содержимым сайта генерирует дубликаты страниц, а также дубликаты изображений в разных размерах. На основании этого сайт необходимо размещать на качественном хостинге. Также необходимо учитывать, что несмотря на большую популярность данной системы управления содержимым сайта, поисковые системы относятся к сайтам на ее базе скептически.

По итогам работы можно сделать следующий вывод:

- создано веб-приложение, позволяющее пользователям создавать/редактировать статьи на определенную тематику:
  - оформление участка;
  - уход за участком;
  - защита участка.

- создано веб-приложение, позволяющее пользователям создавать/редактировать комментарии к статьям;
- создано веб-приложение, позволяющее пользователям изменять параметры своей учетной записи;
- создано веб-приложение, позволяющее пользователям изменять свой и просматривать чужой профиль учетной записи;
- создано веб-приложение, позволяющее пользователям, а также посетителям сайта, создавать свой чертеж ландшафта садового участка.

Созданное веб-приложение удовлетворяет потребностям пользователей в составлении своего представления дизайна ландшафта садового участка на основании опубликованных на сайте материалов, а также на основании использования графического онлайн-редактора чертежа ландшафта садового участка.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Современная актуальность ландшафтного дизайна. – Ландшафтный дизайн и проектирование, озеленение и благоустройство территории: Разработка стиля сада, создание ландшафтного проекта участка, полный комплекс ландшафтных работ, посадка деревьев, рулонный газон, крупномеры. – <http://palitra-land.ru/nashi-partnery/sovremenaja-aktualnost-landshaftnogo-dizaina.html>. – Дата обращения: 24.03.2019.
2. Ландшафтный дизайн. – <http://land.web-3.ru>. – Дата обращения: 24.03.2019.
3. Дизайн интерьера – опыт, профессионализм и низкие цены на дизайн интерьеров. – <http://www.nv-studio.ru>. – Дата обращения: 24.03.2019.
4. Garden Planner Online. – <http://www.smallblueprinter.com/garden/planner.html>. – Дата обращения: 24.03.2019.
5. Рейтинг CMS, популярные движки для сайтов. Рейтинг лучших платных и бесплатных CMS. – <http://www.ratingruneta.ru/cms/>. – Дата обращения: 24.03.2019.
6. НОУ ИНТУИТ | Лекция | Событийно-ориентированные архитектуры. – <https://www.intuit.ru/studies/courses/570/426/lecture/9700>. – Дата обращения: 19.05. 2019.
7. Модель «Сущность – атрибут – значение» (EAV. Entity-attribute-model) | Хостинг 7U3.RU. – <https://7u3.ru/blogs/admin/model-sushchnost-atribut-znachenie-eav-entity-attribute-model.html>. – Дата обращения: 19.05.2019.
8. SQL-Урок 1. Язык SQL. Основные понятия. – [http://moonexcel.com.ua/уроки-sql1-язык-основные-понятия\\_ru](http://moonexcel.com.ua/уроки-sql1-язык-основные-понятия_ru). – Дата обращения: 19.05.2019.
9. PHP: Что такое PHP? – Manual. – <https://www.php.net/manual/ru/intro-what-is.php>. – Дата обращения: 24.03.2019.
10. Язык программирования Ruby. – <https://www.ruby-lang.org/ru/>. – Дата обращения: 24.03.2019.

11. Python – Википедия. – <https://ru.wikipedia.org/wiki/Python>. – Дата обращения: 24.03.2019.
12. JavaScript – Словарь | MDN. – <https://developer.mozilla.org/ru/docs/Словарь/JavaScript>. – Дата обращения: 24.03.2019.
13. Знакомимся с Fabric.js. Часть 1-я – Fabric.js Canvas Library. – [http://fabricjs.com/fabric-intro-part-1\\_ru](http://fabricjs.com/fabric-intro-part-1_ru). – Дата обращения: 20.02.2019.
14. Разработка веб-приложений – что это такое, виды web-приложений и преимущества. – <https://infoshell.ru/blog/razrabotka-veb-prilozhenij/>. – Дата обращения 19.05.2019.

# ПРИЛОЖЕНИЕ А

## СНИМКИ СТРАНИЦ ВЕБ-ПРИЛОЖЕНИЯ

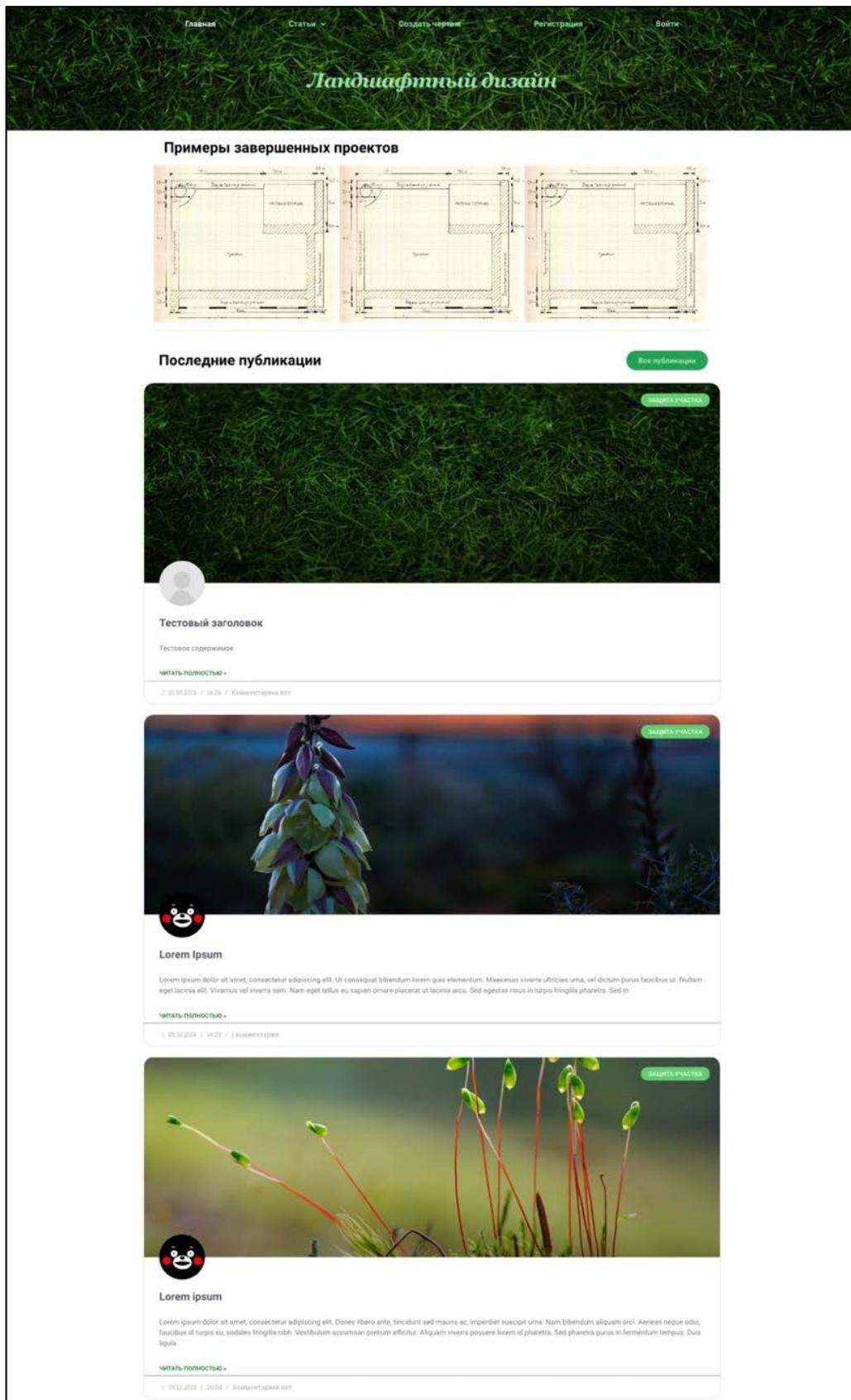


Рисунок А.1 – Главная страница веб-приложения

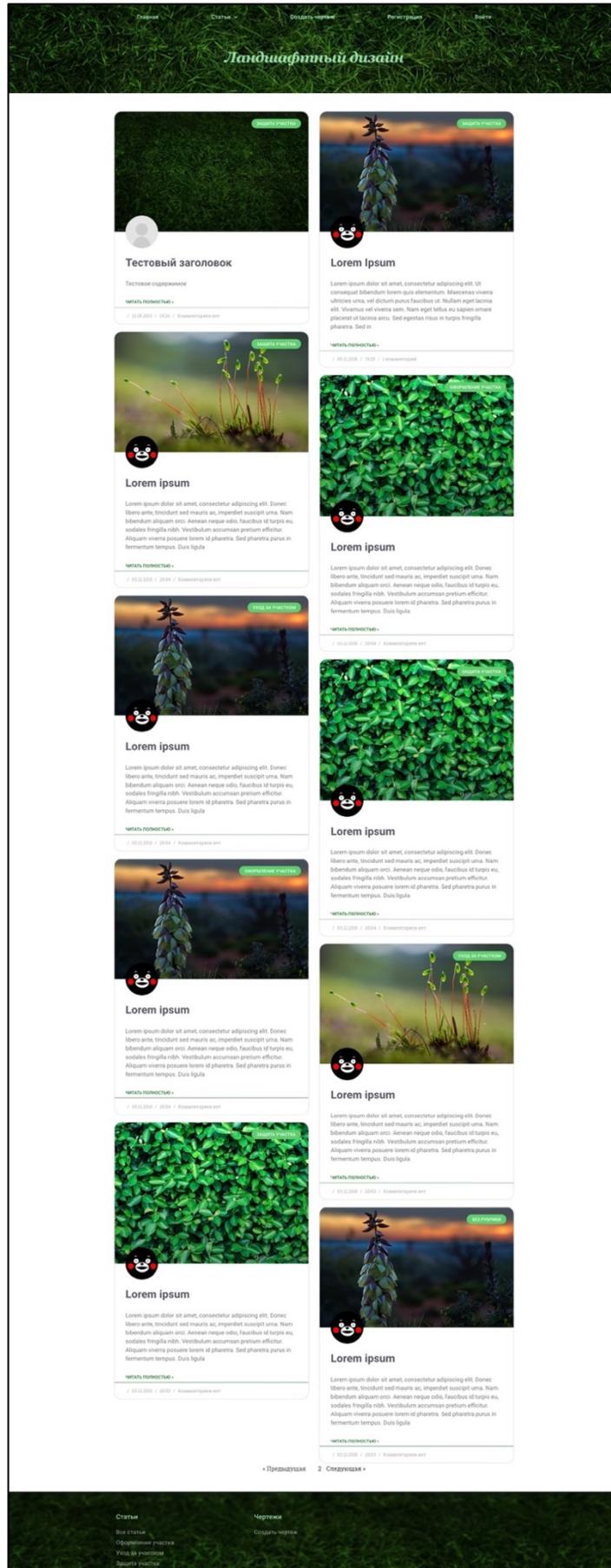


Рисунок А.2 – Страница «Все статьи» веб-приложения

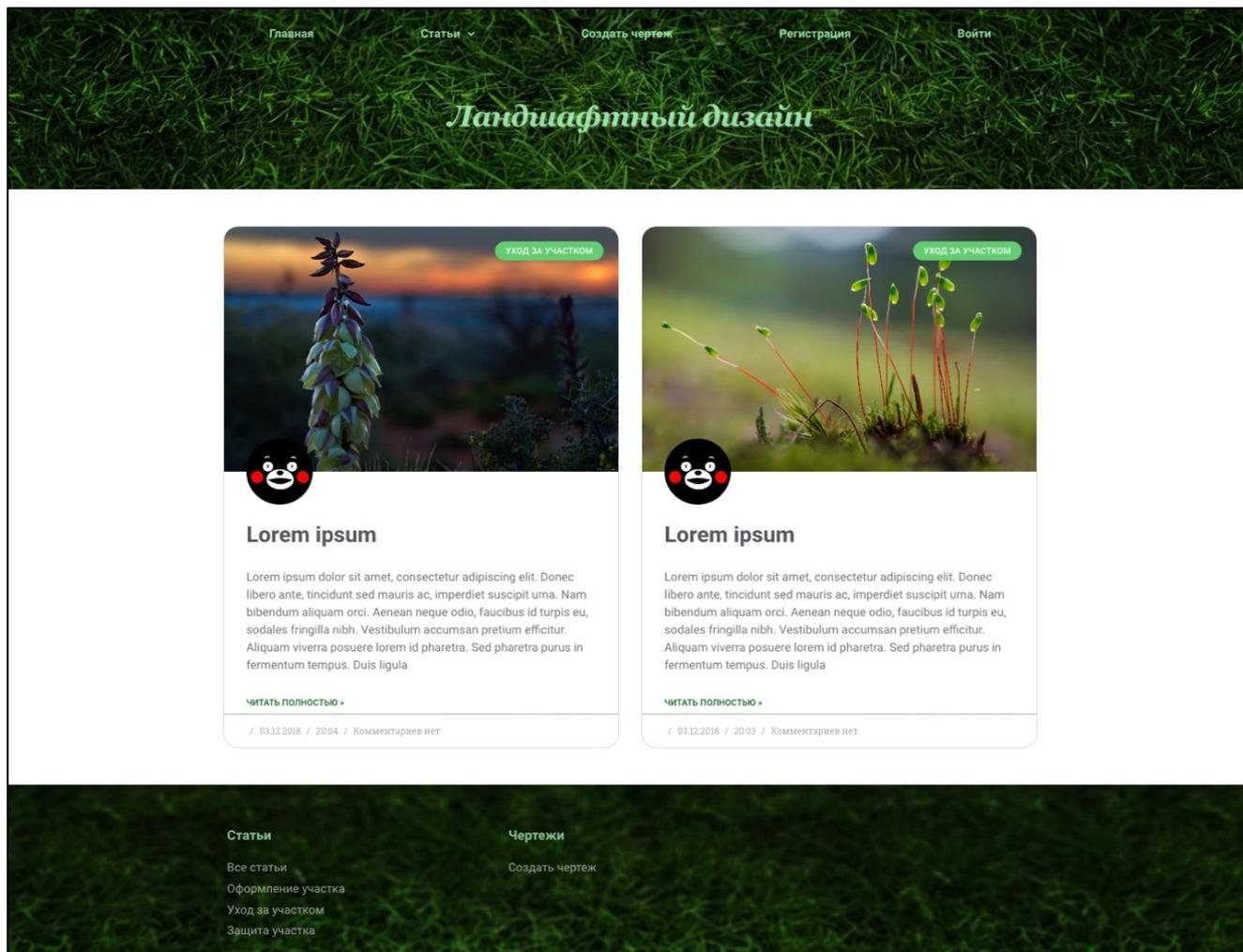


Рисунок А.3 – Страница «Уход за участком» веб-приложения

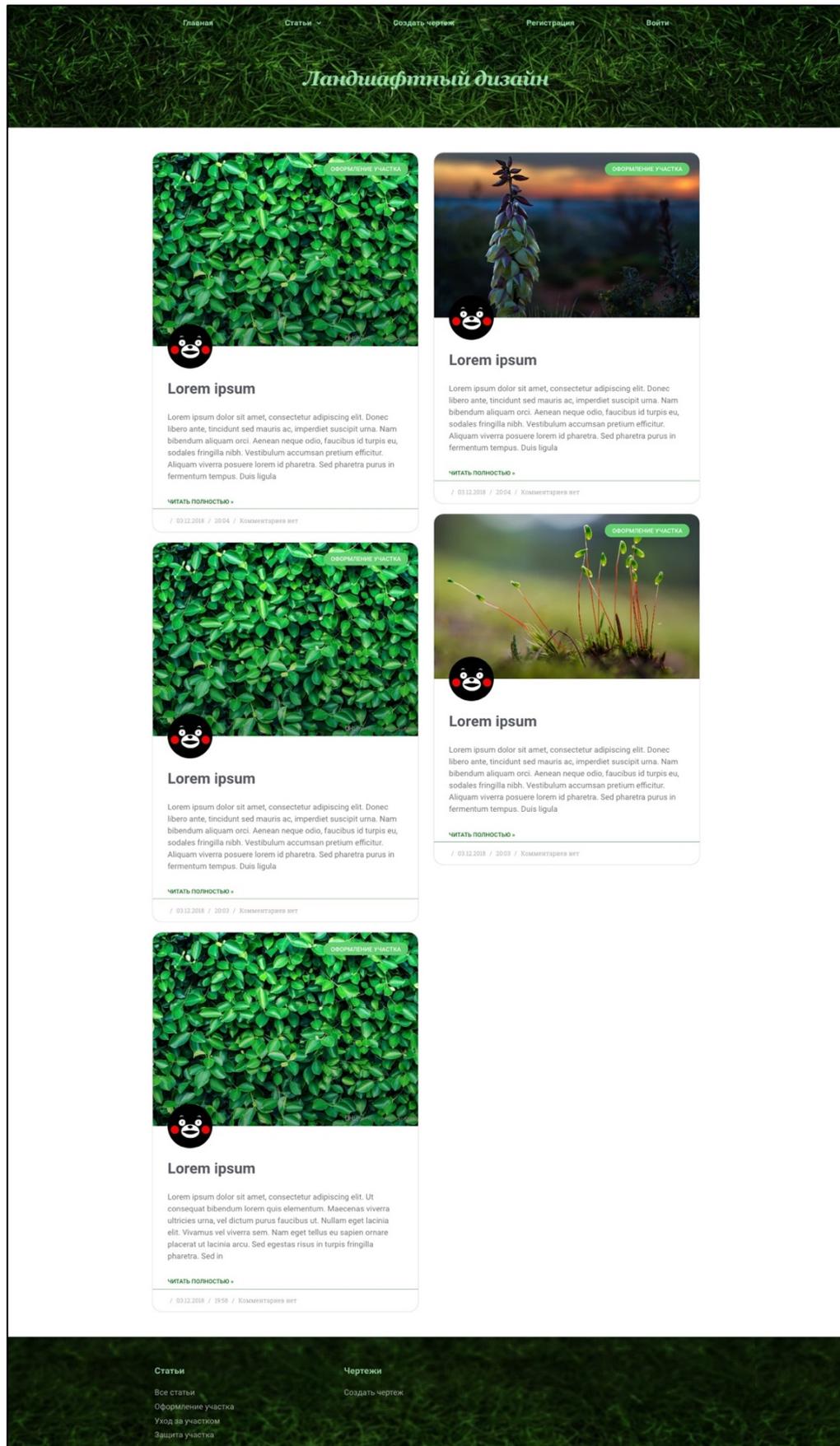


Рисунок А.4 – Страница «Оформление участка» веб-приложения

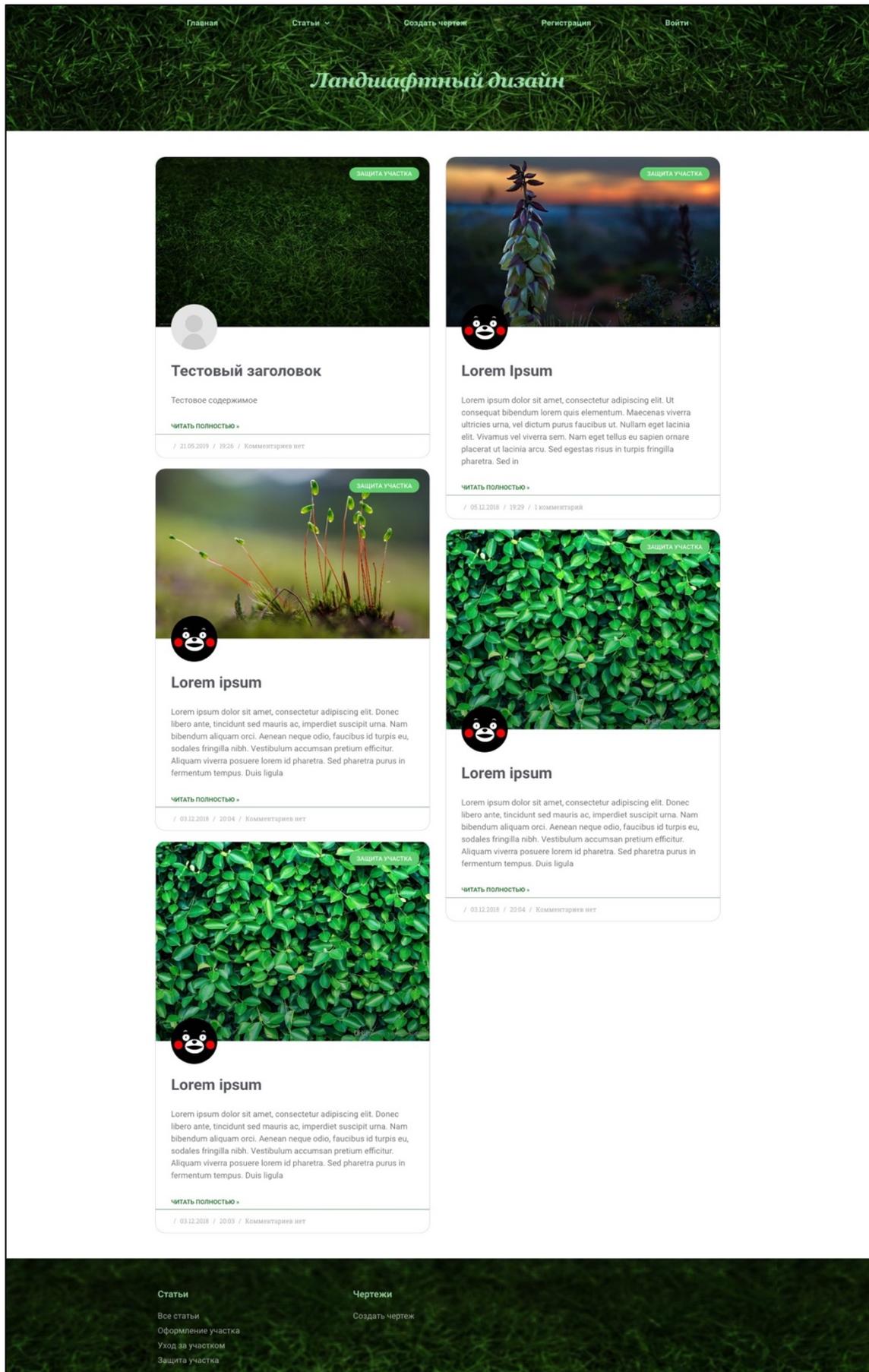


Рисунок А.5 – Страница «Защита участка» веб-приложения

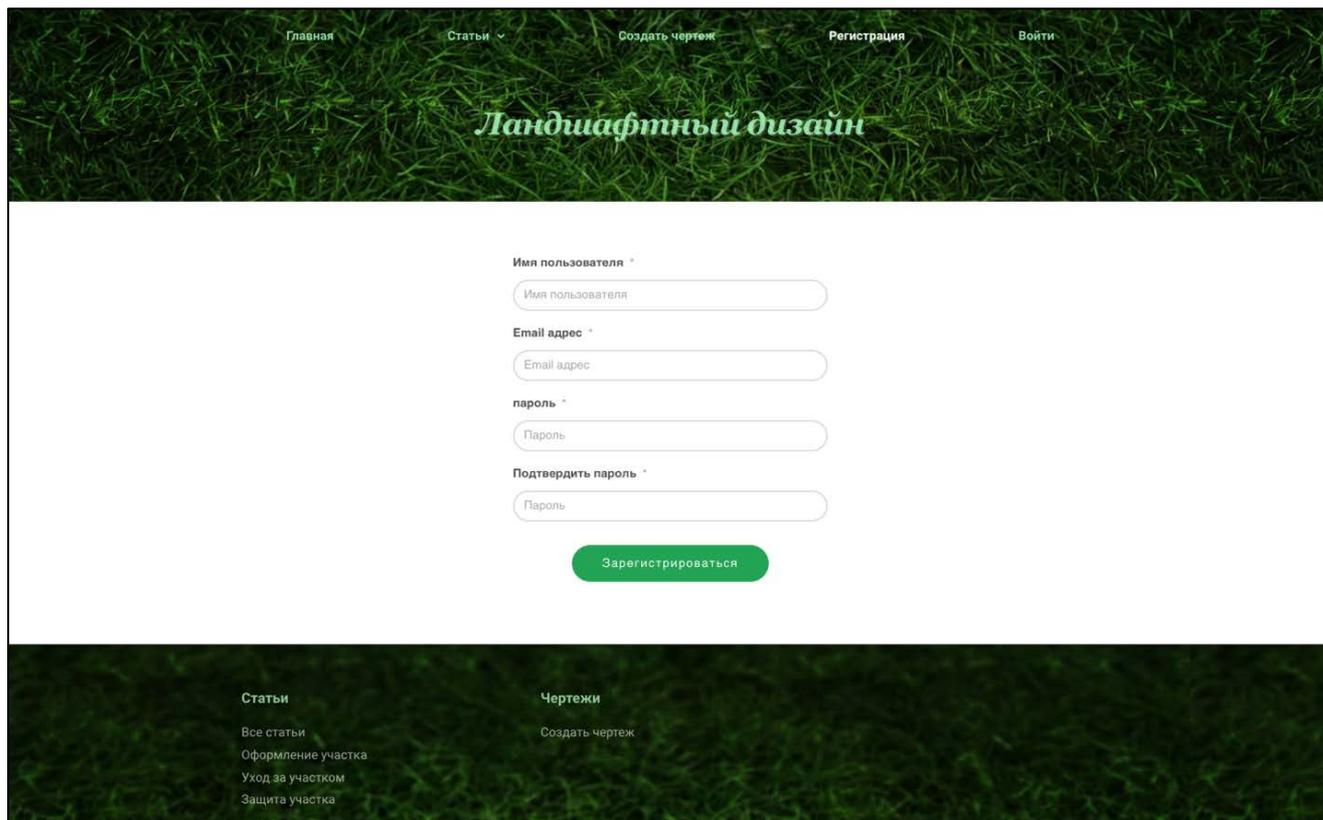


Рисунок А.6 – Страница «Регистрация» веб-приложения

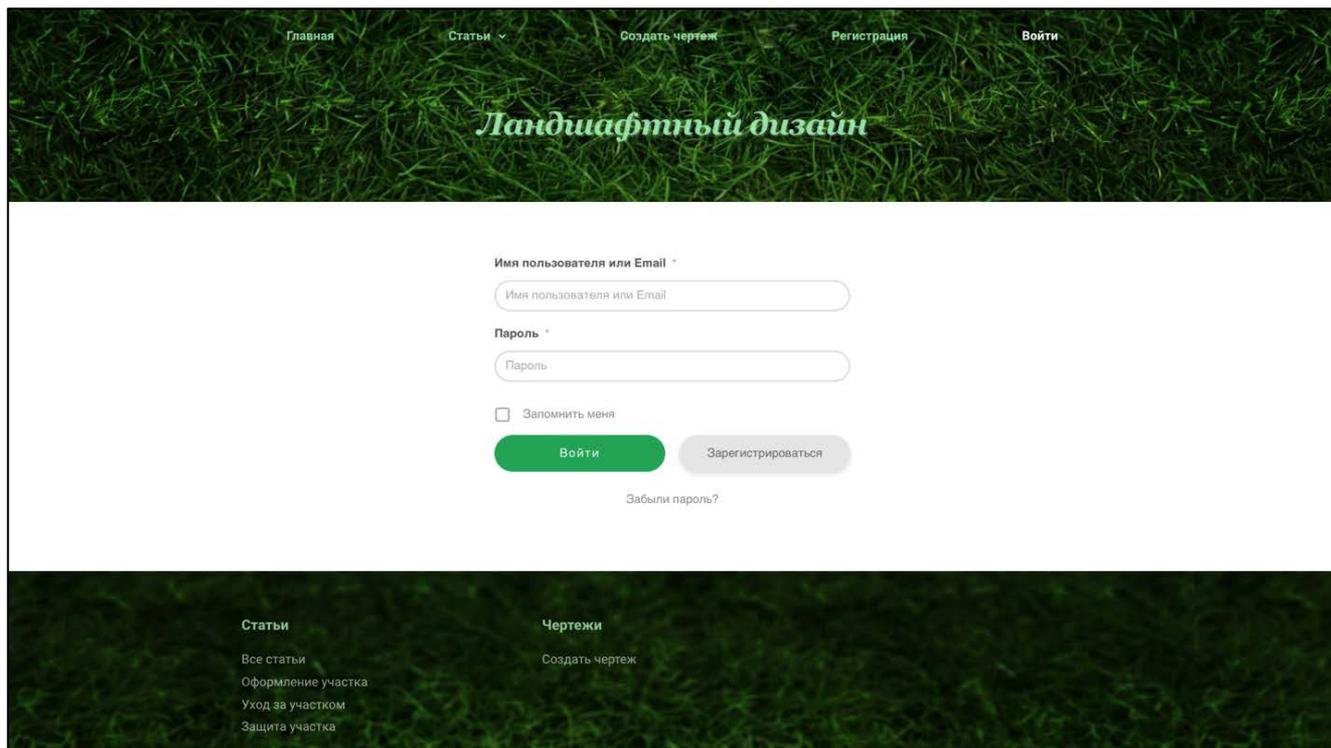


Рисунок А.7 – Страница «Войти» веб-приложения

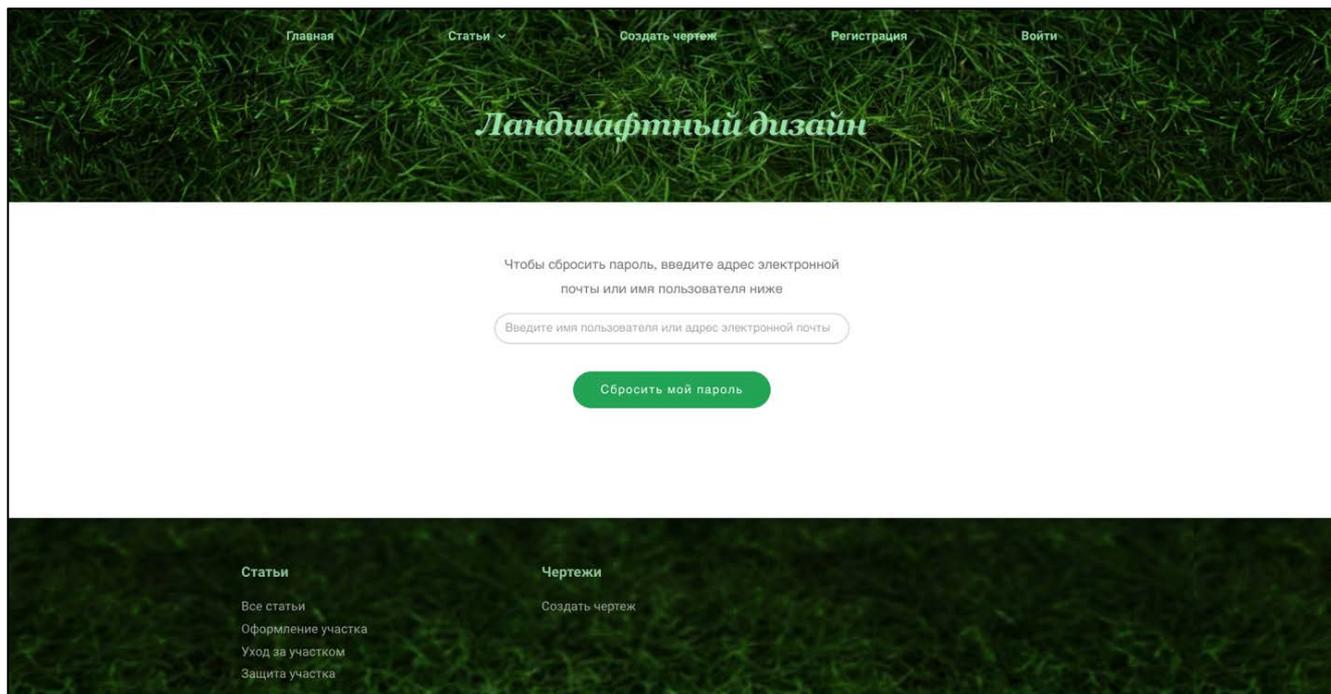


Рисунок А.8 – Страница «Восстановление пароля» веб-приложения

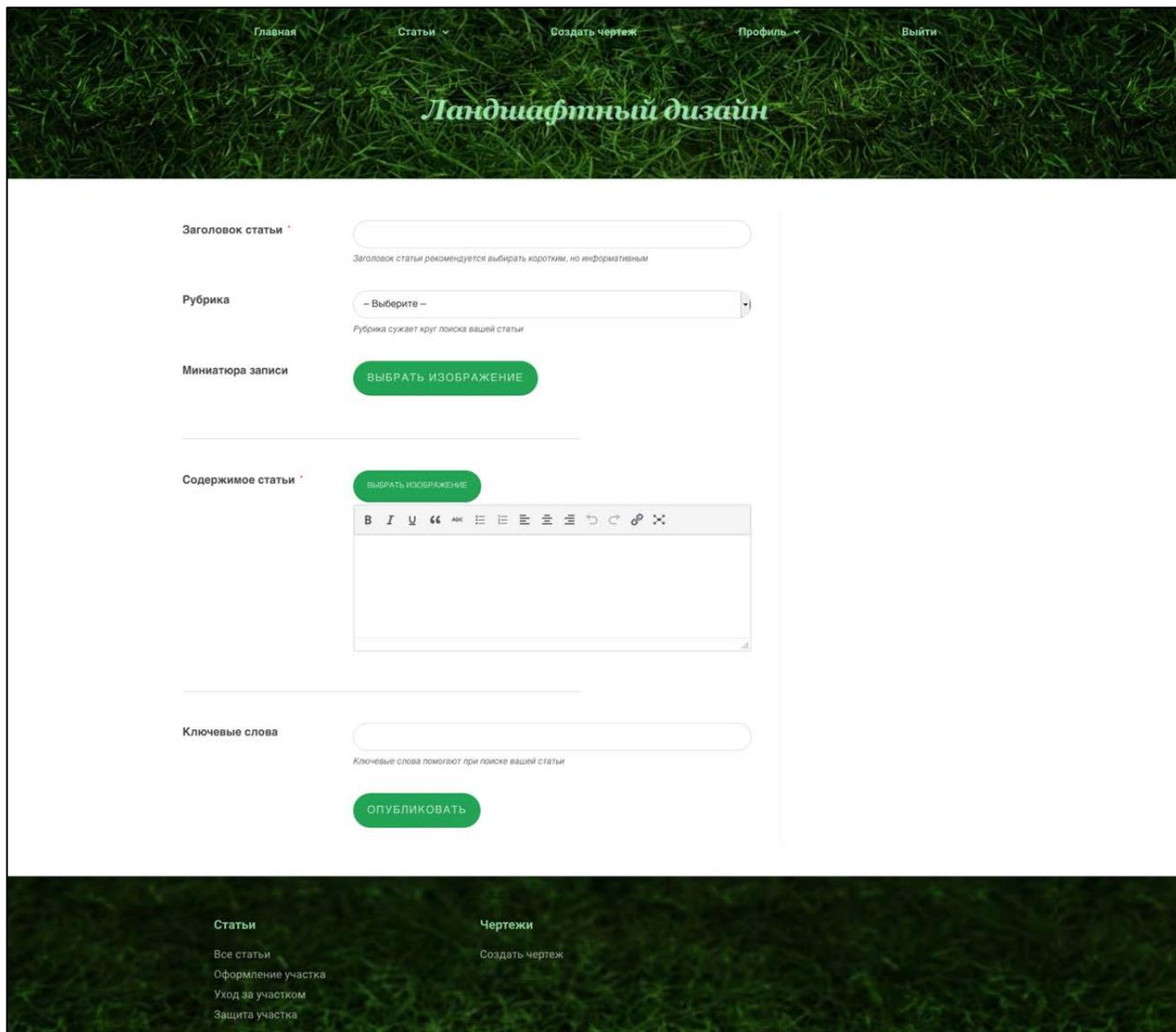


Рисунок А.9 – Страница «Создать статью» веб-приложения

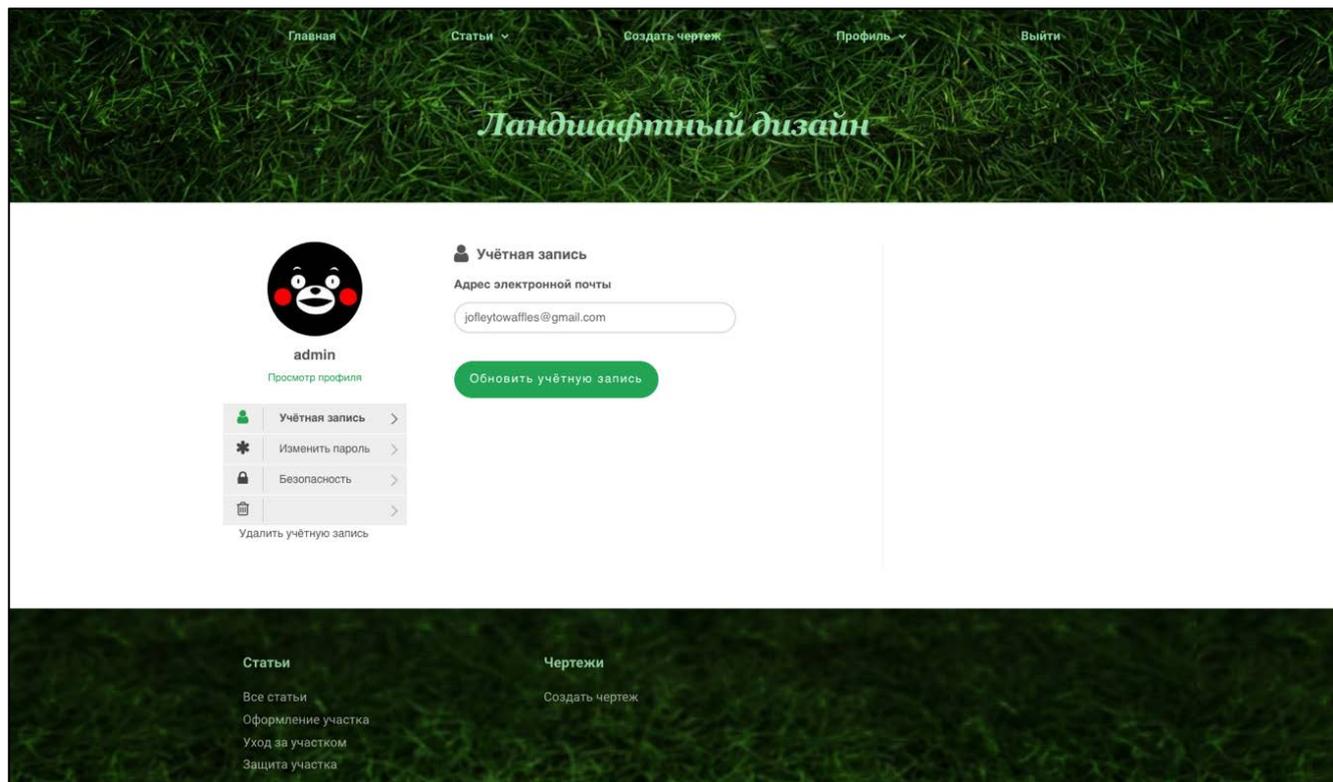


Рисунок А.10 – Страница «Учетная запись» веб-приложения

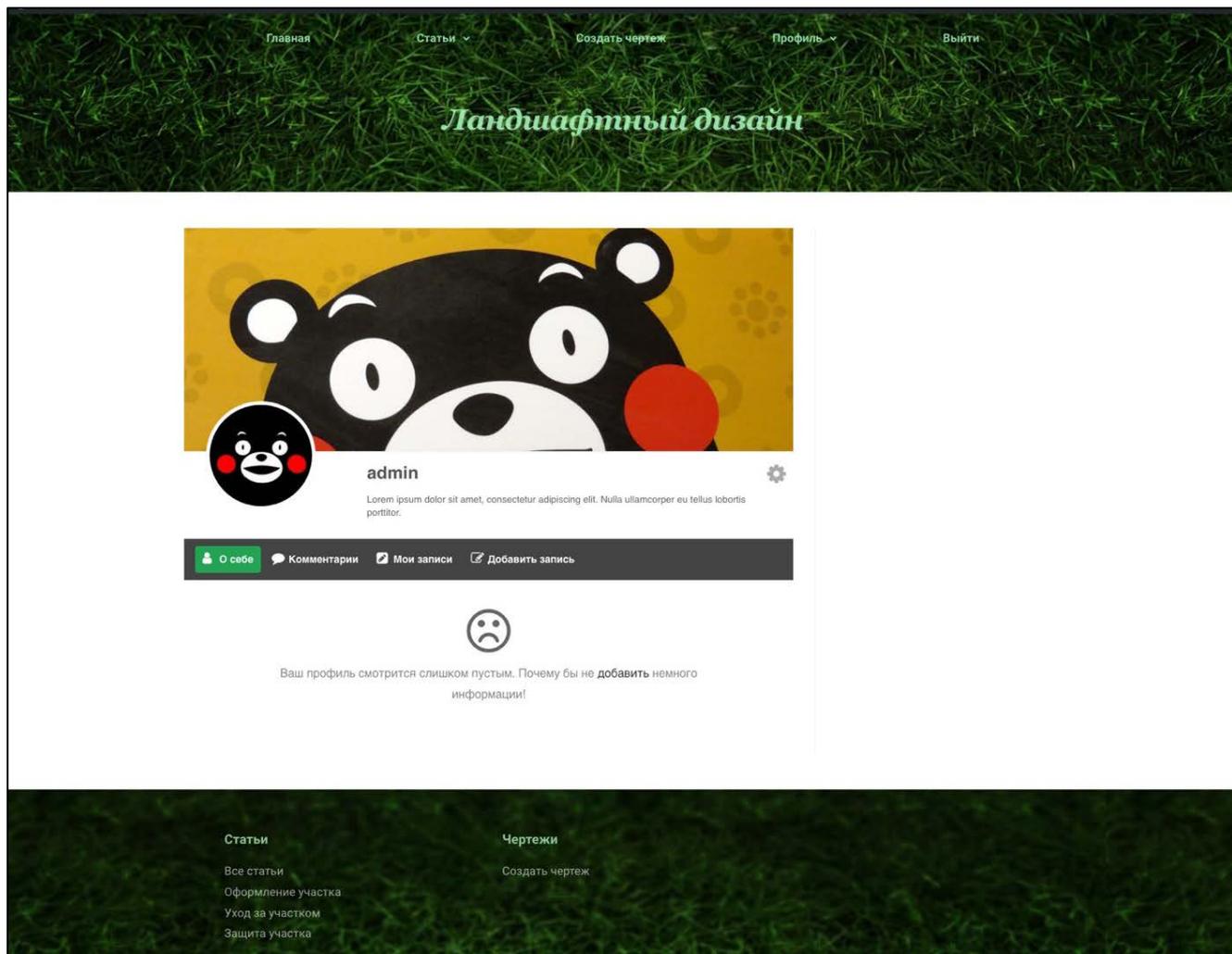


Рисунок А.11 – Страница «Профиль» веб-приложения

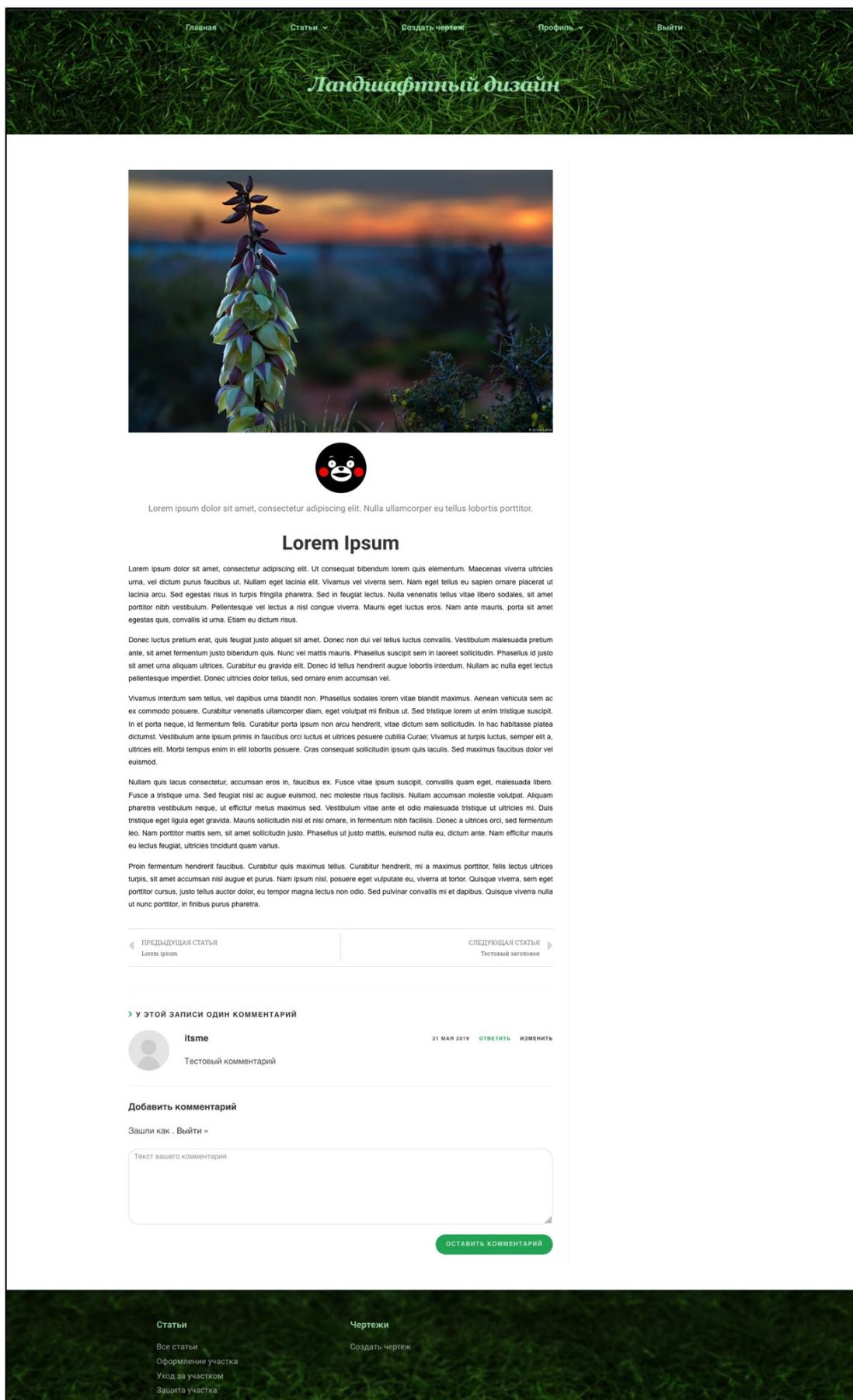


Рисунок А.12 – Страница статьи веб-приложения

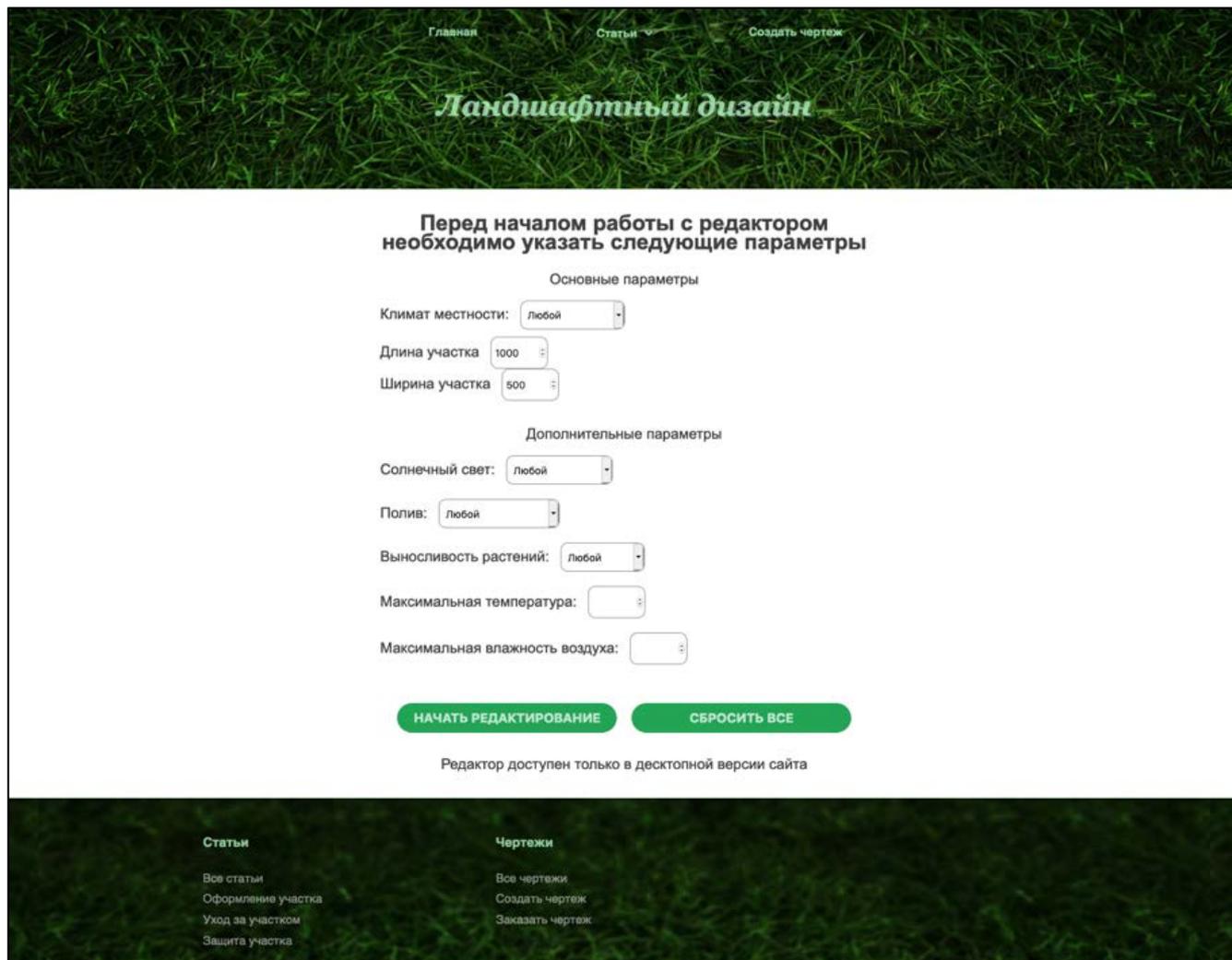


Рисунок А.13 – Страница «Редактор чертежа садового участка» веб-приложения.  
Экранная форма заполнения начальных условий

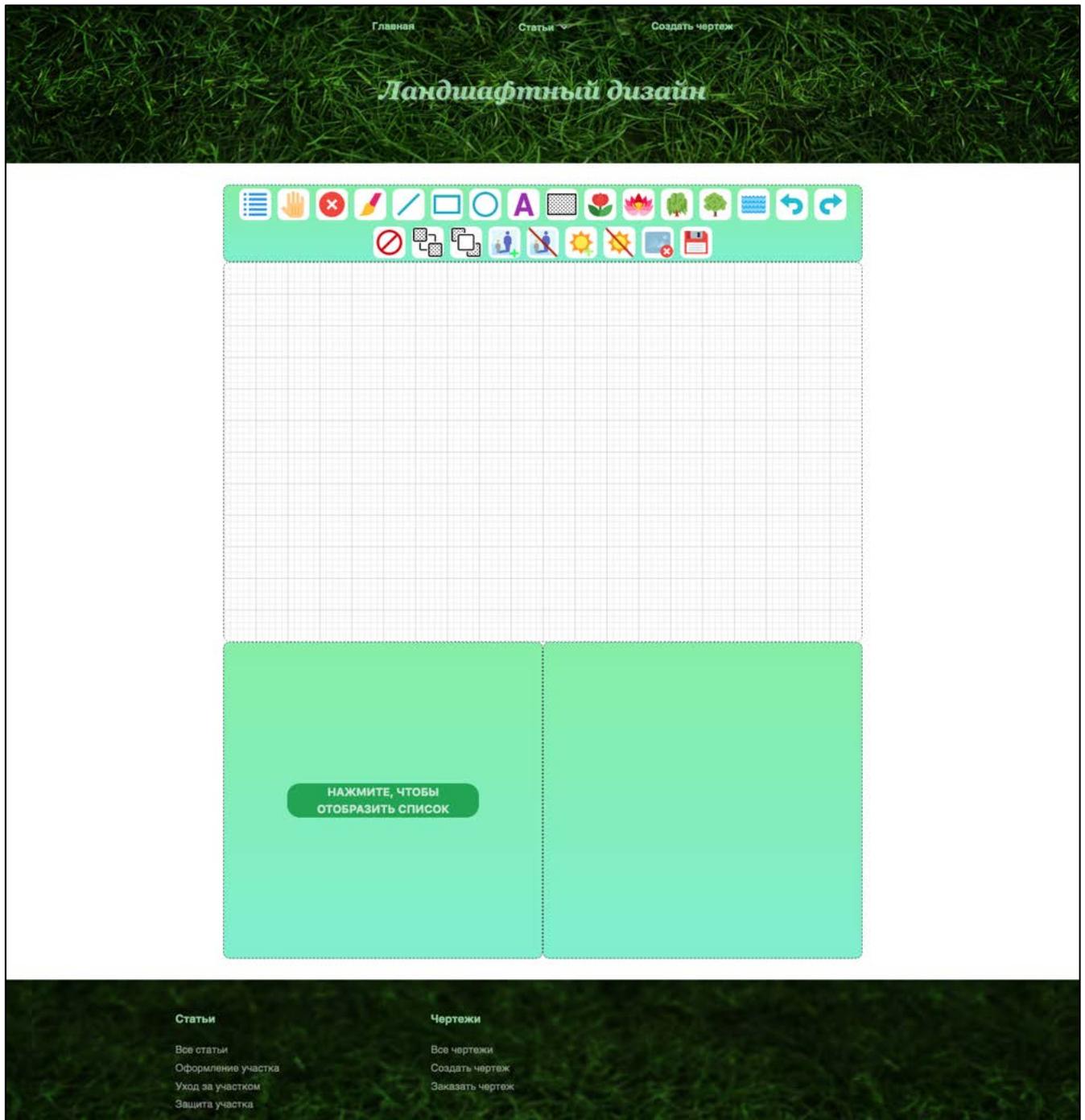


Рисунок А.14 – Страница «Редактор чертежа садового участка» веб-приложения.  
Графический онлайн-редактор

## ПРИЛОЖЕНИЕ Б

### ПРИМЕР ИСПОЛЬЗОВАНИЯ ГРАФИЧЕСКОГО ОНЛАЙН-РЕДАКТОРА ВЕБ-ПРИЛОЖЕНИЯ

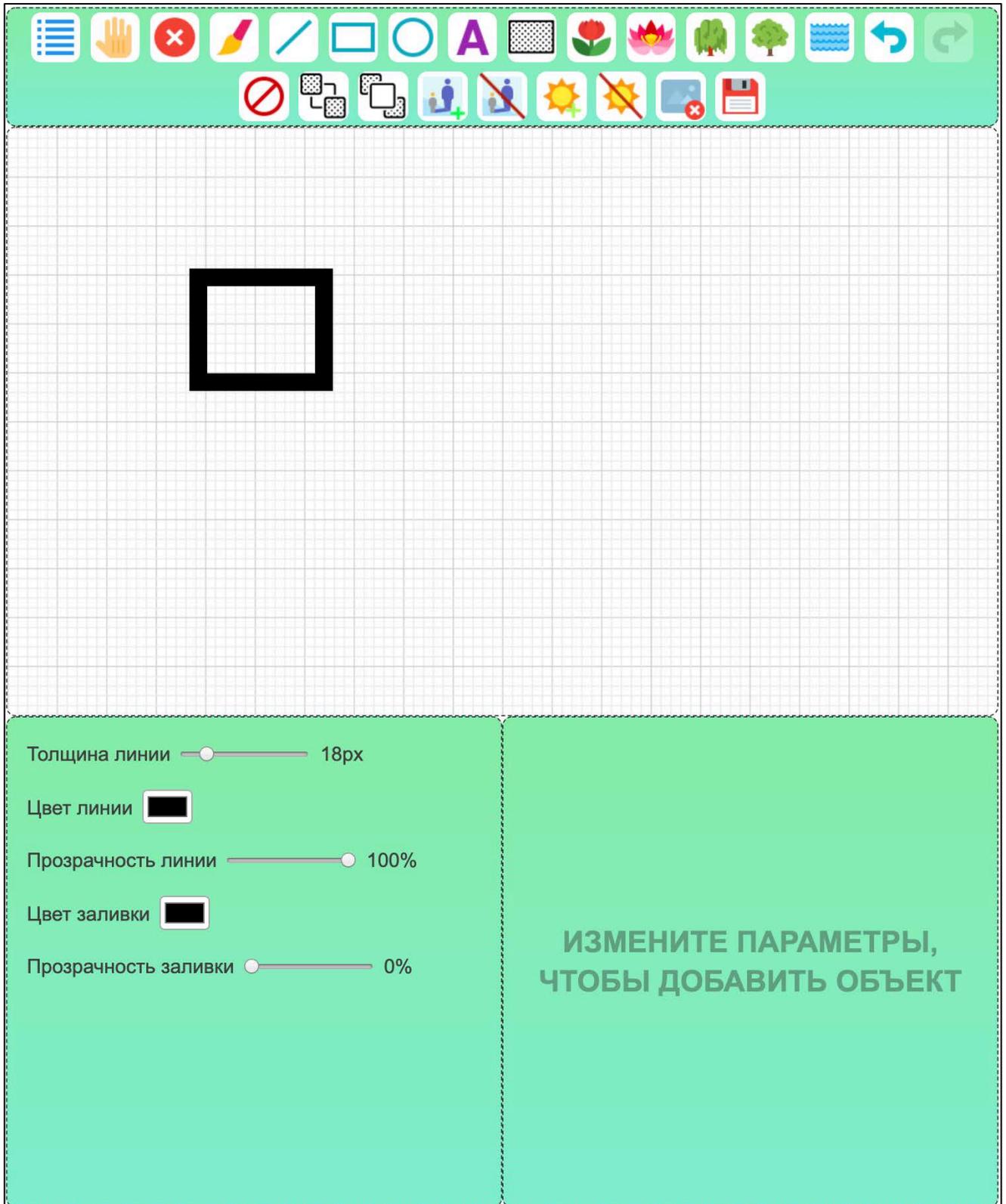


Рисунок Б.1 – Добавление фигуры

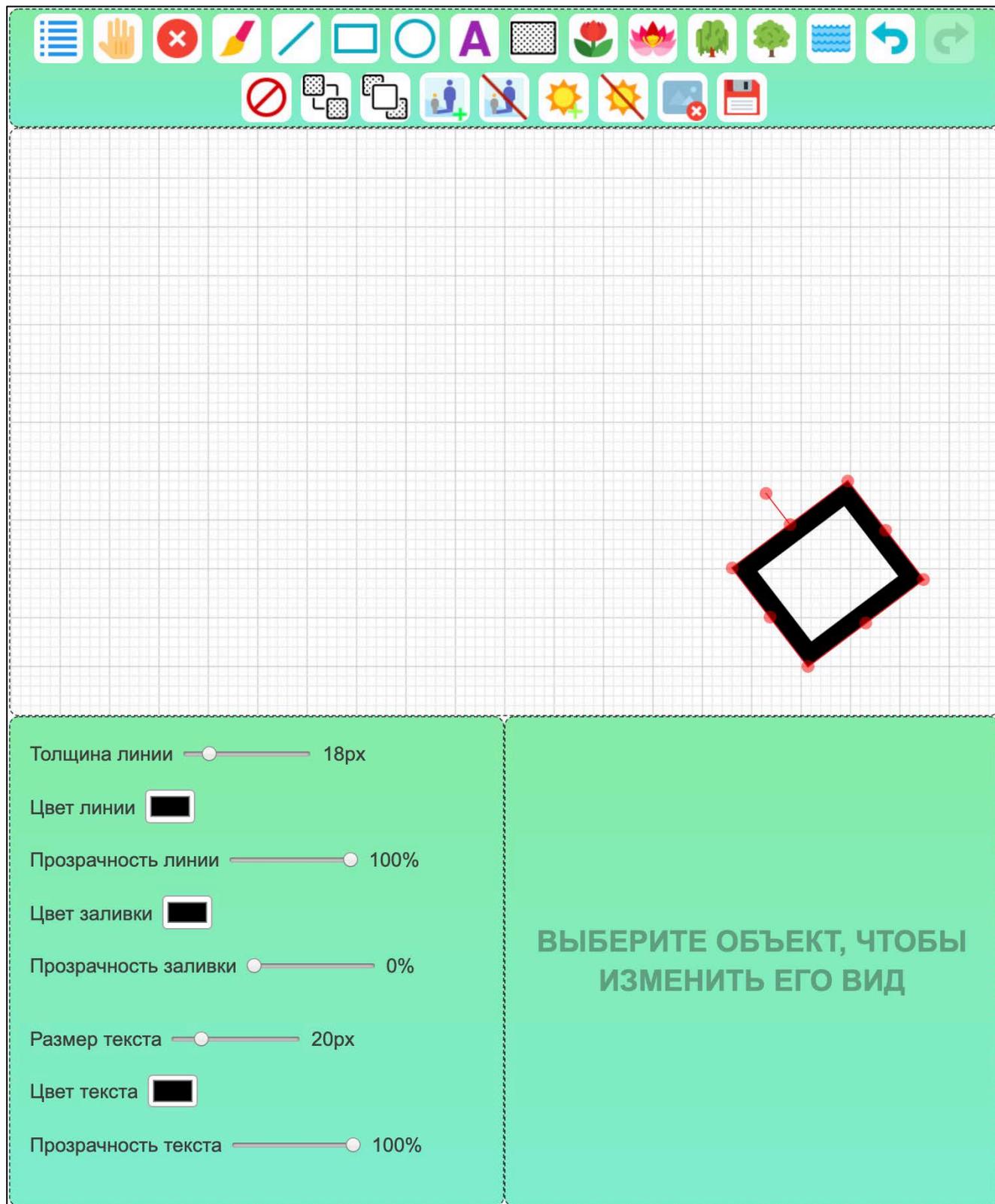


Рисунок Б.2 – Изменение размеров фигуры и ее наклона

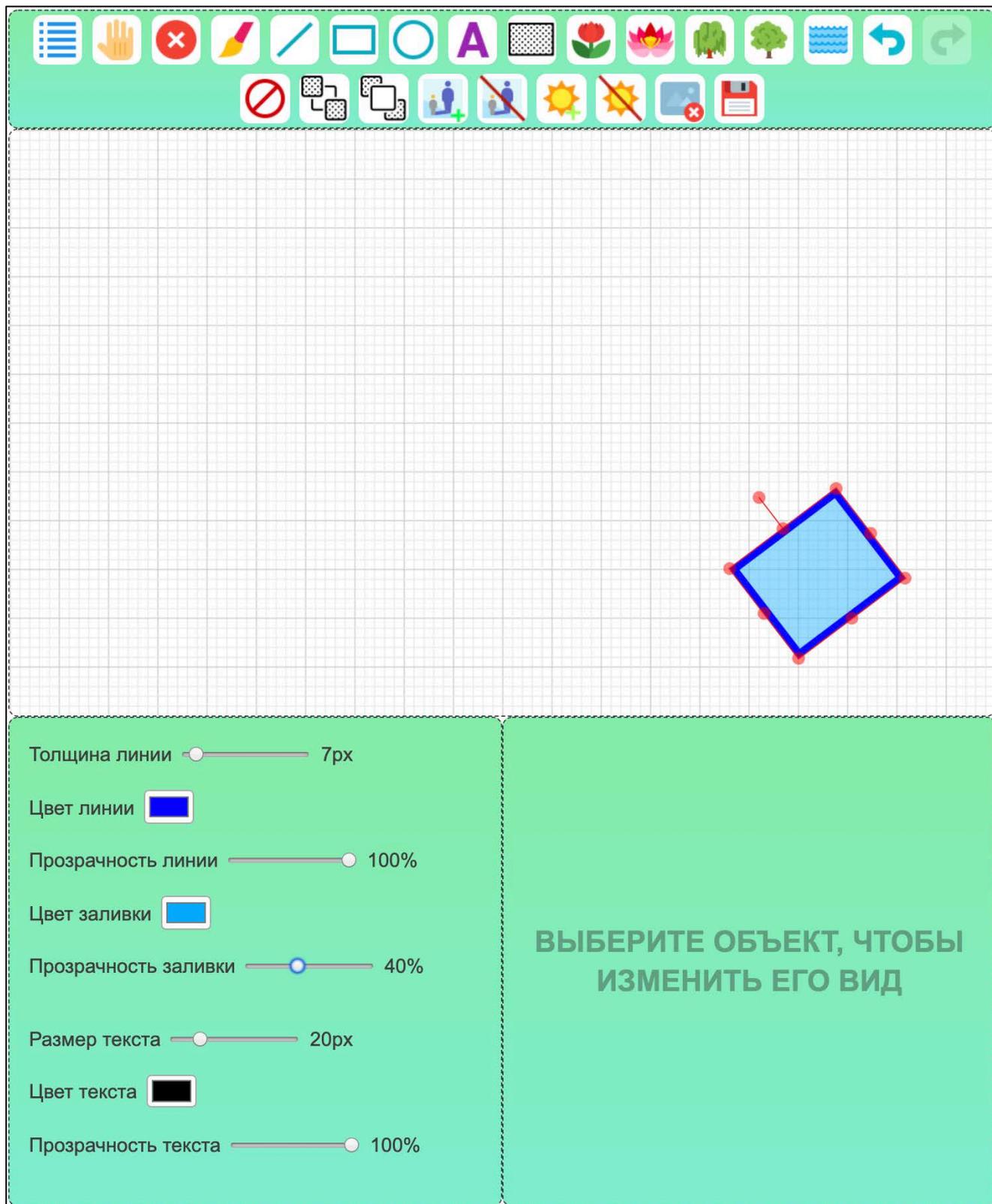


Рисунок Б.3 – Изменение толщины, цвета и прозрачности контура и заливки фигуры

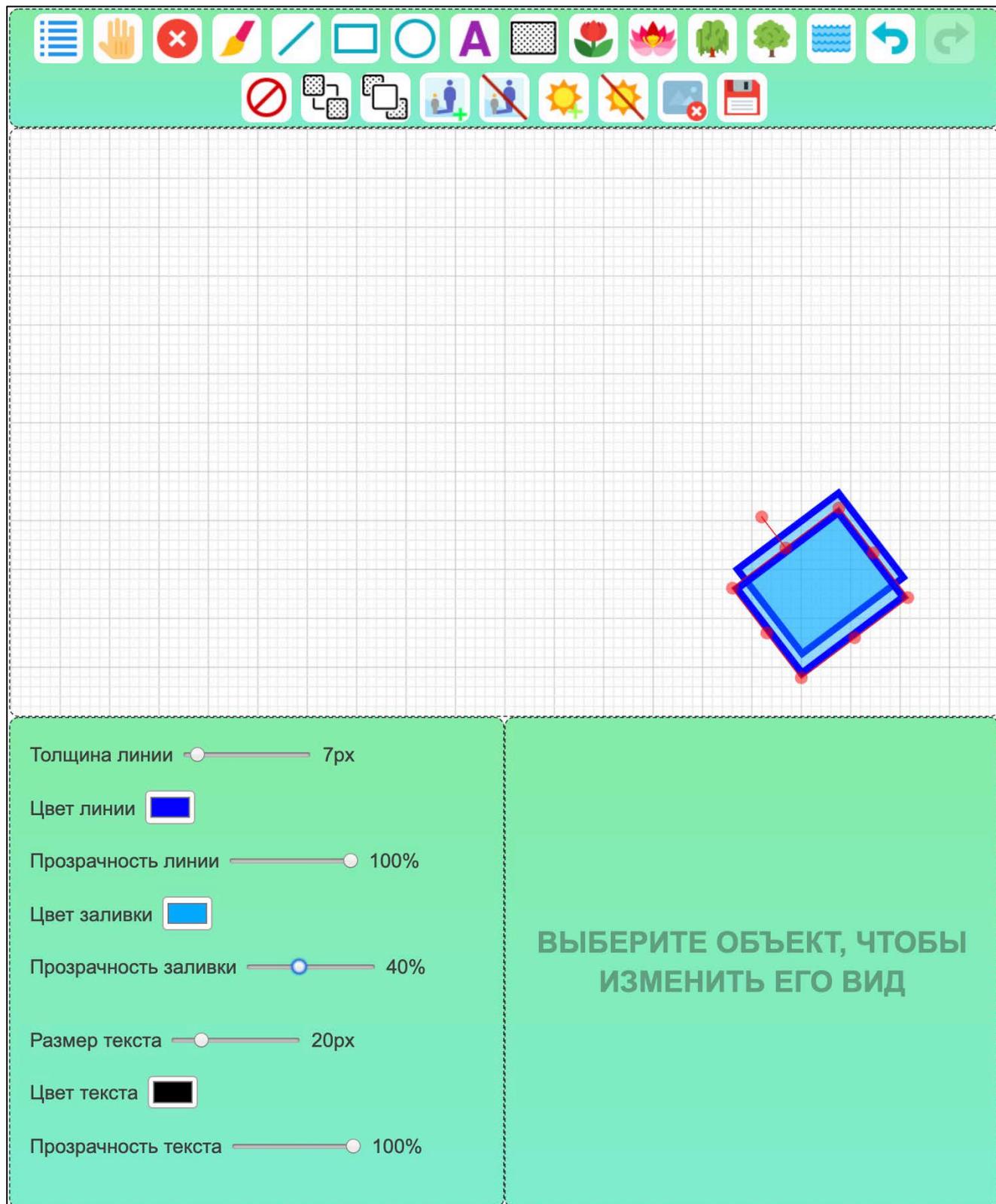


Рисунок Б.4 – Дублирование объекта

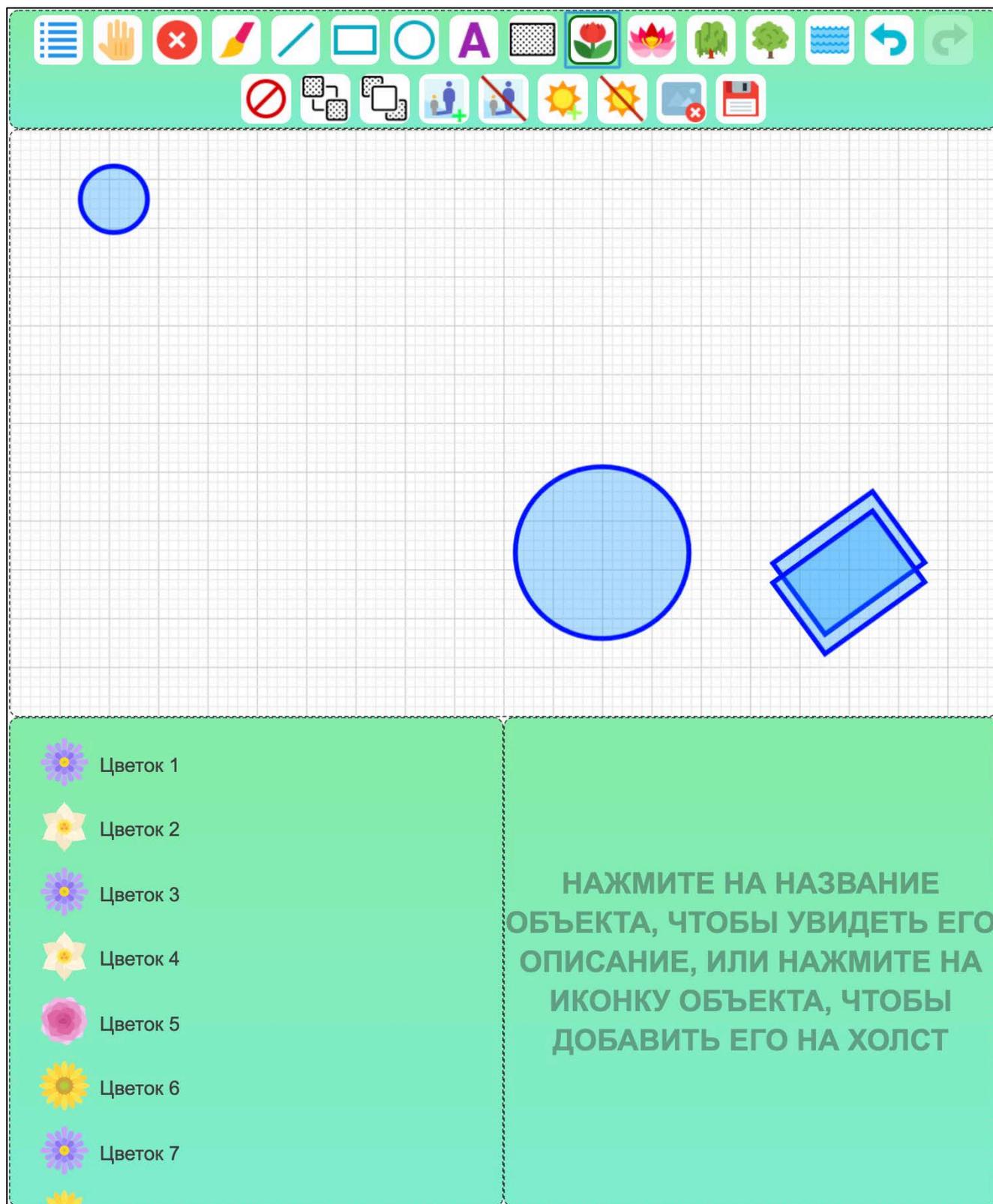


Рисунок Б.5 – Отображение списка всех цветов

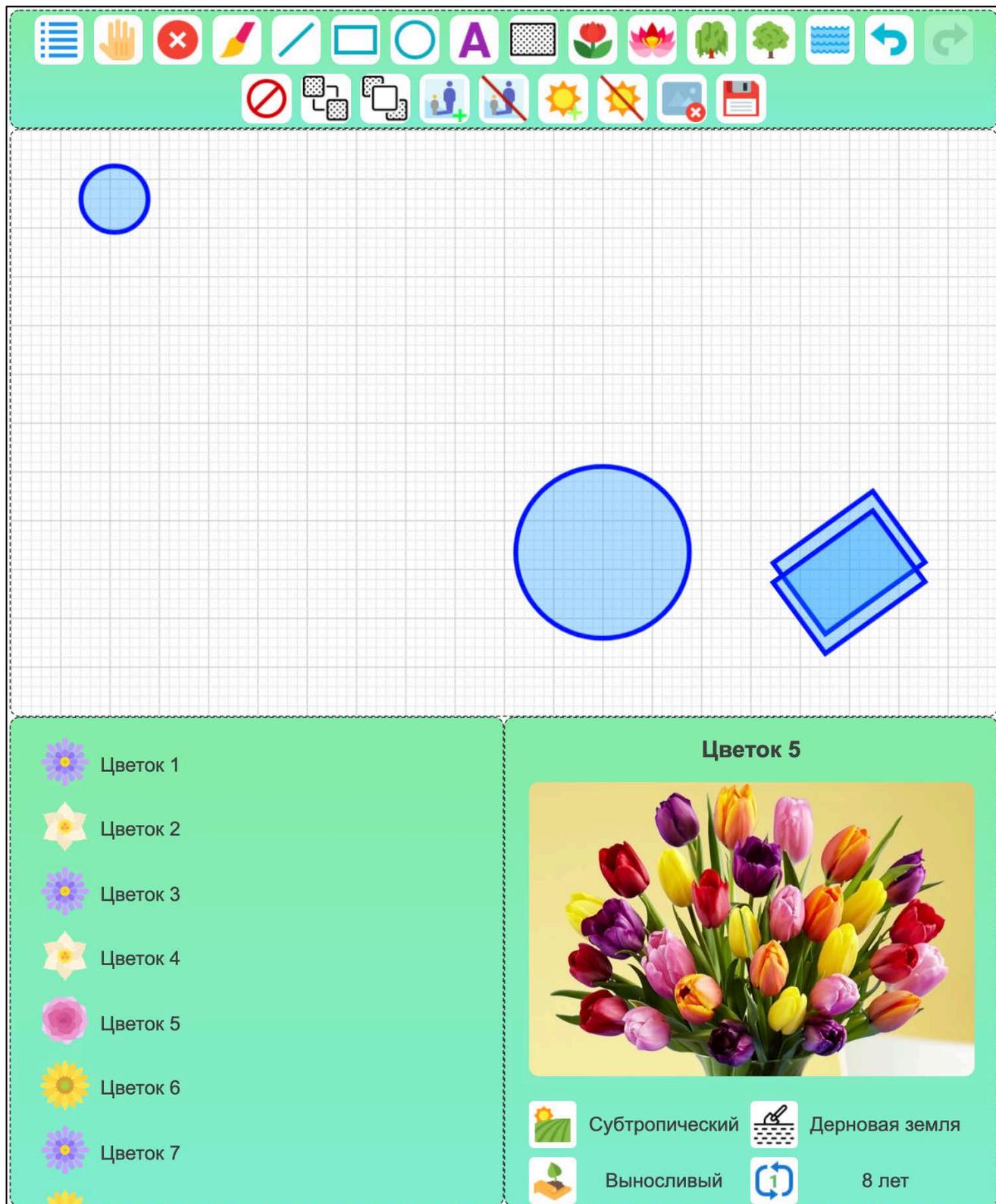


Рисунок Б.6 – Отображение подробной информации об объекте из списка цветов

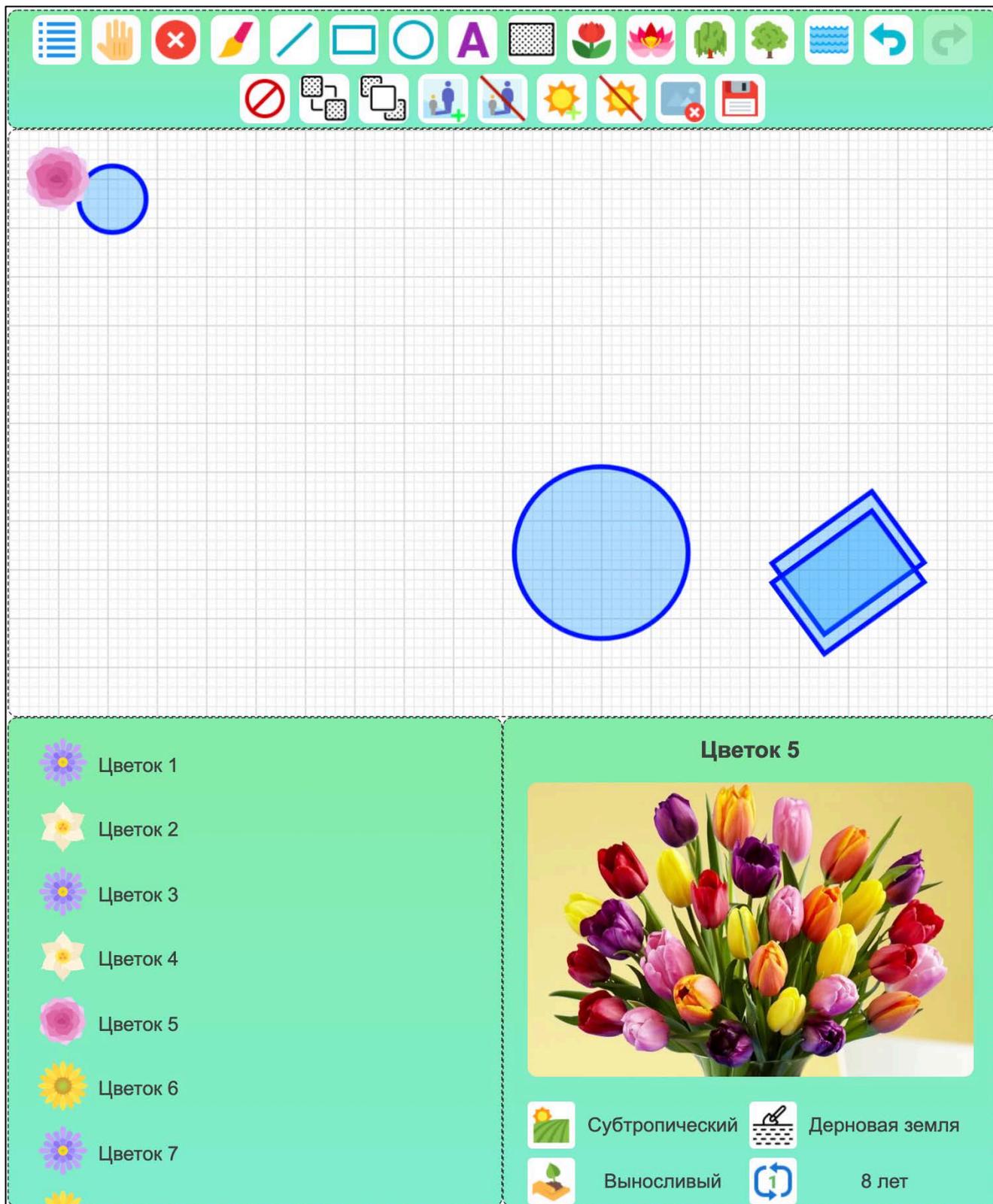


Рисунок Б.7 – Добавление иконки из списка цветов объекта на чертеж

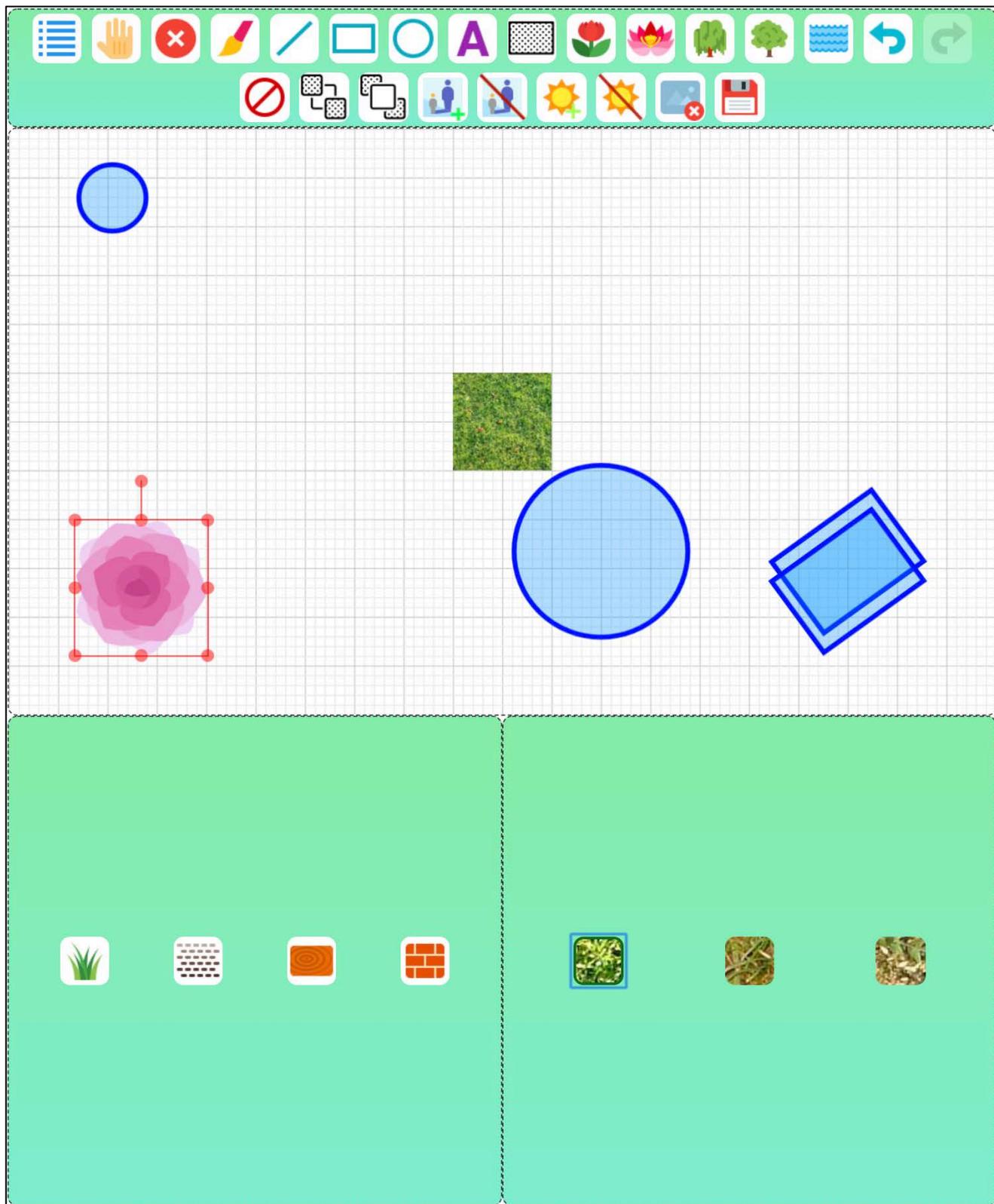


Рисунок Б.8 – Добавление фигуры «Прямоугольник-покрытие»

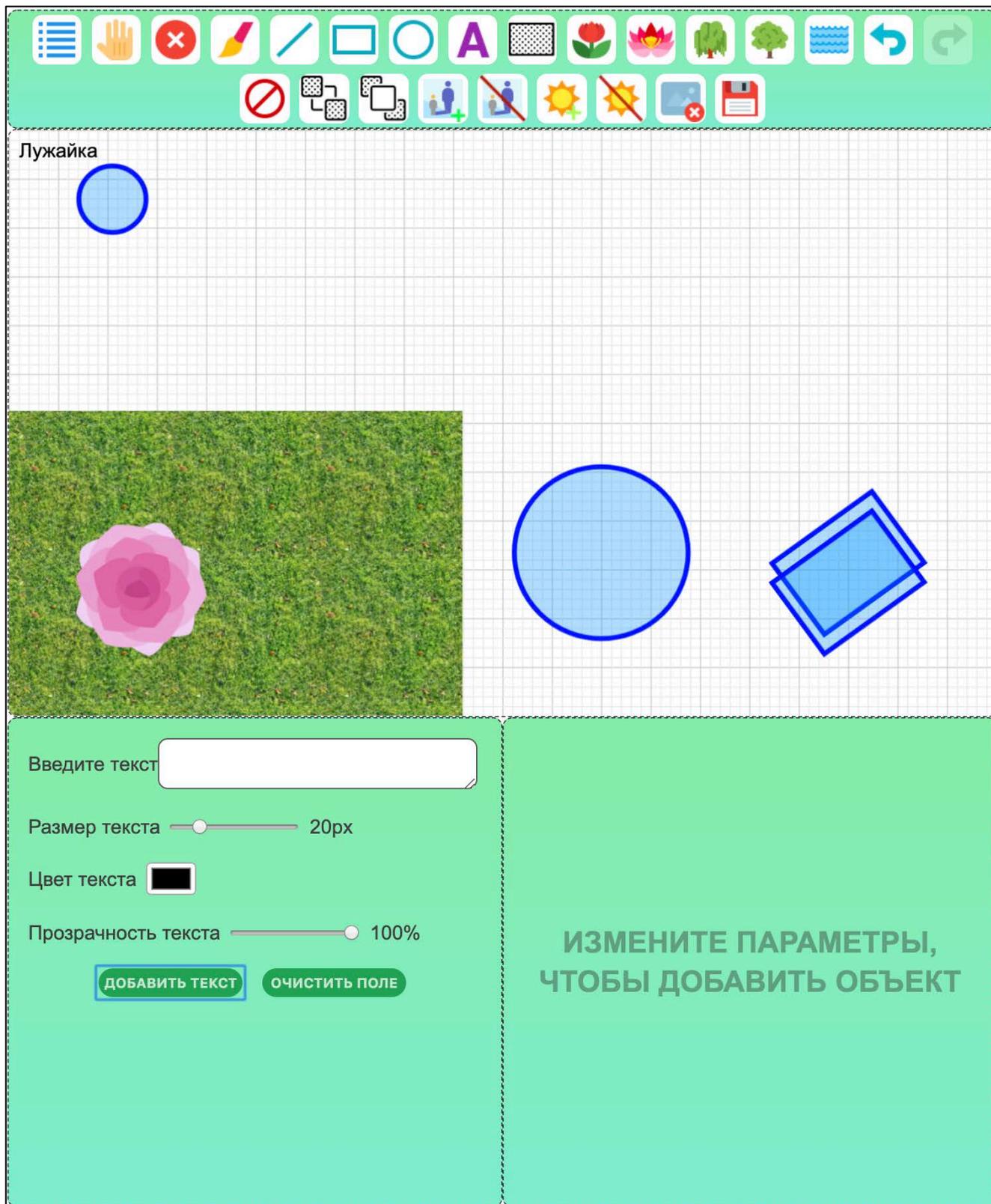


Рисунок Б.9 – Добавление текстового объекта на чертеж

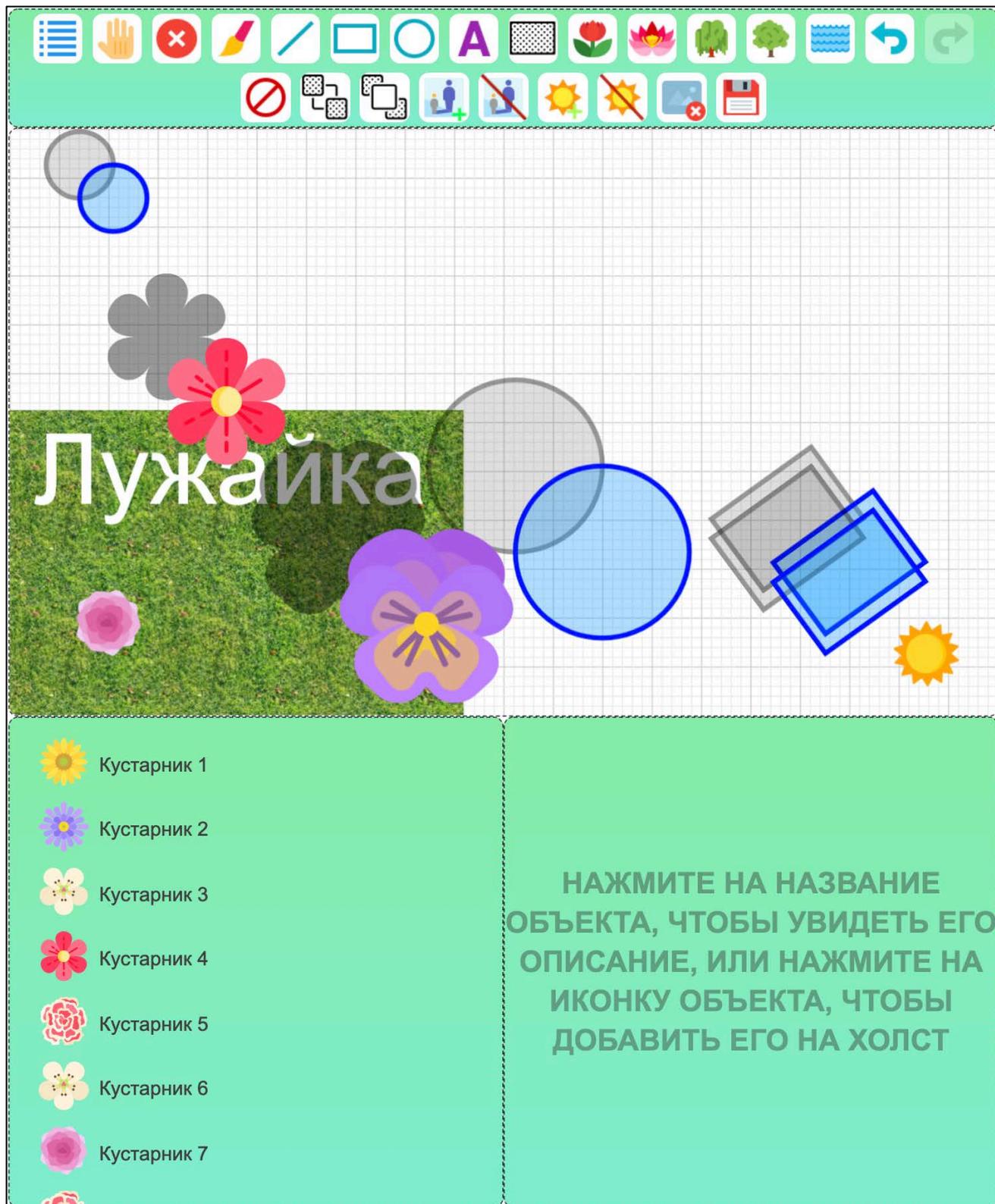


Рисунок Б.10 – Отображение теней фигур «Прямоугольник» и «Окружность», объектов типа «Кустарник» и «Дерево»

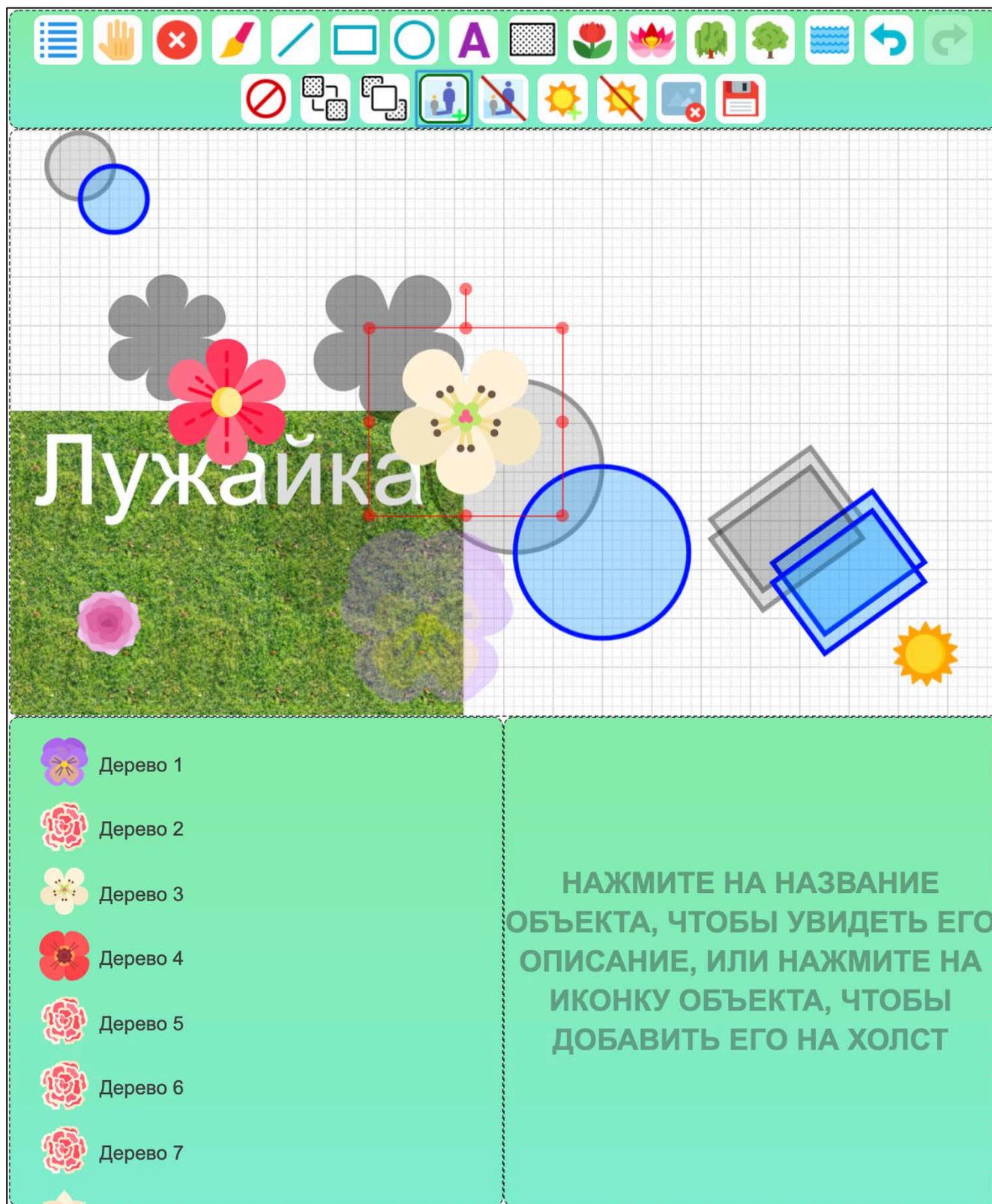


Рисунок Б.11 – Учет дистанции между объектами типа «Кустарник» и «Дерево». Понижение прозрачности объекта при пересечении с ним перемещаемого объекта

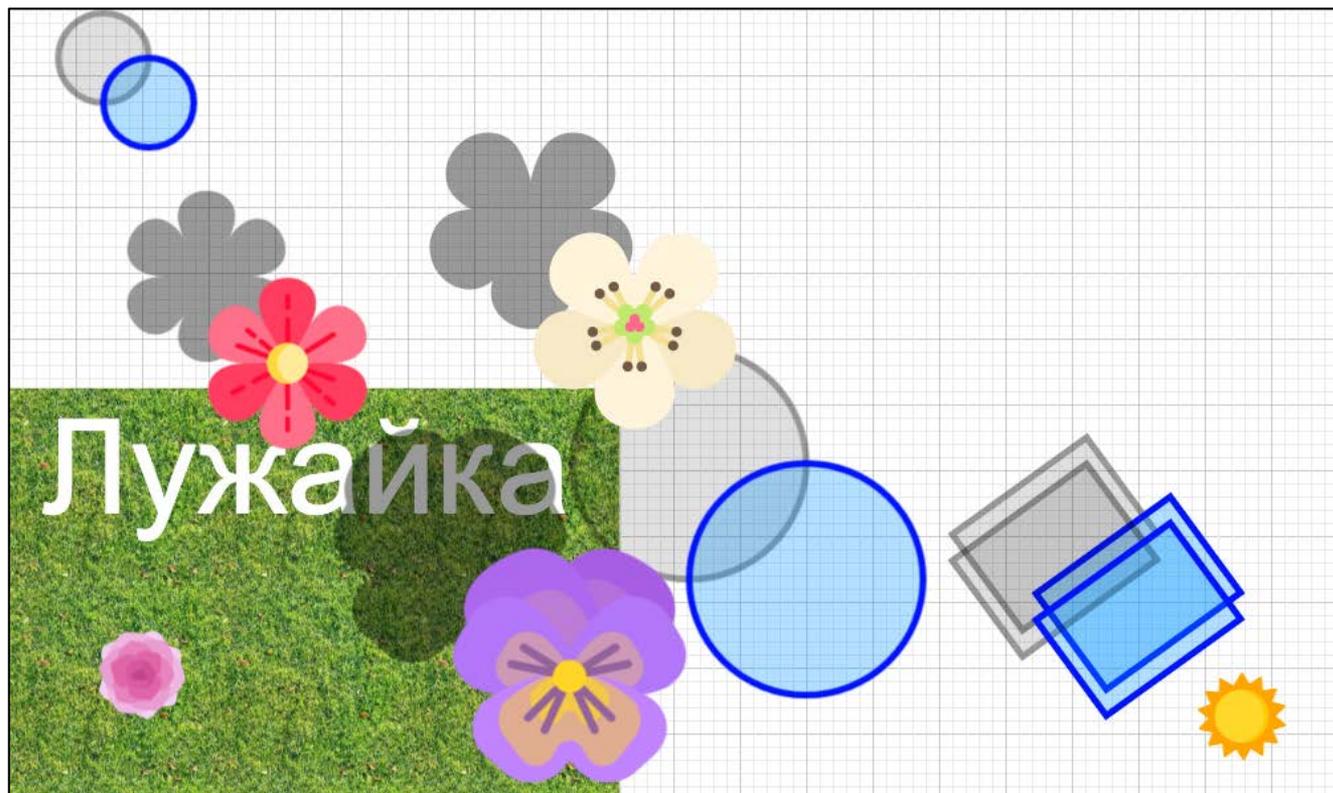


Рисунок Б.12 – Завершенный и скачанный на локальный компьютер пользователя чертеж ландшафта садового участка

## ПРИЛОЖЕНИЕ В

### ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА UTIL

#### Листинг В.1 – Исходный код файла “util.js”

```
'use strict';

/**
 * Немедленно выполняемая функция, предназначенная для работы с элементами HTML-страницы
 * Ничего не принимает
 * Возвращает набор функций для работы с элементами HTML-страницы
 */
window.util = (function () {
    let ICON_FOLDER = 'http://localhost/landscape-editor/objects_icons/',
        IMG_FOLDER = 'http://localhost/landscape-editor/object_imgs/';

    let listItems = document.querySelector(".list-items"),
        properties = document.querySelector(".properties"),
        actionBtn = document.querySelector(".action-btn"),
        editTip = document.querySelector('.edit-tip'),
        addTip = document.querySelector('.add-tip'),
        listTip = document.querySelector(".no-one-selected"),
        lineProperties = document.querySelector(".line-properties"),
        fillProperties = document.querySelector(".fill-properties"),
        textAreaProperties = document.querySelector(".text-area-input"),
        textArea = document.getElementById("text-area"),
        textAreaLabel = document.querySelector(".text-area-label"),
        textSubmit = document.querySelector(".add-text-area"),
        textErase = document.querySelector(".erase-text-area"),
        form = document.querySelector('form'),
        canvasContainer = document.querySelector('.canvas-block'),
        listFlowers = document.querySelector('.list-flowers'),
        listBushes = document.querySelector('.list-bushes'),
        listTrees = document.querySelector('.list-trees'),
        listWaterPlants = document.querySelector('.list-water-plants'),
        listPounds = document.querySelector('.list-pounds'),
        patternTypes = document.querySelector('.pattern-types');

    let objectCard = document.querySelector('.object-card'),
        objectName = objectCard.querySelector('h3'),
        objectPicture = objectCard.querySelector('.object-picture'),
        objectClimate = objectCard.querySelector('.climate'),
        objectGround = objectCard.querySelector('.ground'),
        objectWaterDeep = objectCard.querySelector('.water-deep'),
        objectEndurance = objectCard.querySelector('.endurance'),
        objectLifeTime = objectCard.querySelector('.lifetime'),
        objectAirHumidity = objectCard.querySelector('.air_humidity'),
        objectLight = objectCard.querySelector('.light'),
        objectTemperature = objectCard.querySelector('.temperature'),
        objectWatering = objectCard.querySelector('.watering'),
        objectPlantTime = objectCard.querySelector('.plant-time'),
        objectPitDeep = objectCard.querySelector('.pit-deep'),
        objectPitRadius = objectCard.querySelector('.pit-radius'),
        objectDistance = objectCard.querySelector('.distance'),
        poundDeep = objectCard.querySelector('.pound-deep'),
        poundLength = objectCard.querySelector('.pound-length'),
        poundWidth = objectCard.querySelector('.pound-width'),
        poundWaterLevel = objectCard.querySelector('.pound-water-level'),
        objectDescription = objectCard.querySelector('.description'),
        objectRemark = objectCard.querySelector('.remark');

    let grassPatterns = document.querySelector('.grass-patterns'),
        groundPatterns = document.querySelector('.ground-patterns'),
        woodPatterns = document.querySelector('.wood-patterns'),
        tilePatterns = document.querySelector('.tile-patterns');
```

```

return {
  /**
   * Функция вывода сообщения об ошибке в случае неудачной отправки HTTP-запроса
   * Принимает сообщение об ошибке
   * Ничего не возвращает
   */
  isError: function (errorMessage) {
    let elem = document.querySelector('.error-message');
    if (!elem) {
      let container = document.createElement('div');
      container.classList.add('error-message');
      container.textContent = errorMessage;
      document.body.insertBefore(container, document.body.children[0]);
    }
  },
  /**
   * Функция добавления класса HTML-элементу
   * Принимает HTML-элемент
   * Ничего не возвращает
   */
  hideElement: function (element) {
    if (!element.classList.contains("hidden")) {
      element.classList.add("hidden");
    }
  },
  /**
   * Функция удаления класса у HTML-элемента
   * Принимает HTML-элемент
   * Ничего не возвращает
   */
  showElement: function (element) {
    if (element.classList.contains("hidden")) {
      element.classList.remove("hidden");
    }
  },
  /**
   * Функция добавления вариантов выбора <option> в <select> со стороны сервера
   * Принимает JSON-объект
   * Ничего не возвращает
   */
  addSelectOptions: function (data) {
    for (let select in data) {
      if (select === 'climate') {
        for (let i = 0; i < data[select].length; i++) {
          let selectClimate =
document.getElementById('climate'),
          option = document.createElement('option');
          option.text = data[select][i];
          option.value = data[select][i];
          selectClimate.appendChild(option);
        }
      } else if (select === 'light') {
        for (let i = 0; i < data[select].length; i++) {
          let selectLight = document.getElementById('light'),
          option = document.createElement('option');
          option.text = data[select][i];
          option.value = data[select][i];
          selectLight.appendChild(option);
        }
      } else if (select === 'watering') {
        for (let i = 0; i < data[select].length; i++) {
          let selectWatering =
document.getElementById('watering'),
          option = document.createElement('option');
          option.text = data[select][i];
          option.value = data[select][i];
          selectWatering.appendChild(option);
        }
      }
    }
  }
}

```

## Продолжение приложения В

```

    } else if (select === 'endurance') {
        for (let i = 0; i < data[select].length; i++) {
            let selectEndurance =
document.getElementById('endurance'),
                option = document.createElement('option');
                option.text = data[select][i];
                option.value = data[select][i];
                selectEndurance.appendChild(option);
            }
        }
    },
    /**
     * Функция скрывает экранной формы начальных условий
     * Не принимает ничего
     * Ничего не возвращает
     */
    closeForm: function () {
        window.util.hideElement(form);
        window.util.showElement(canvasContainer);
    },
    /**
     * Функция заполнения списка полученными со стороны сервера данными объектов
     (иконка, наименование)
     * Принимает JSON-объект
     * Ничего не возвращает
     */
    addObject: function (data) {
        for (let type in data) {
            if (type === 'flower') {
                if (data[type].length > 0) {
                    let ul = document.createElement('ul');
                    for (let i = 0; i < data[type].length; i++) {
                        let li = document.createElement('li');
                        let p = document.createElement("p");
                        let img = document.createElement('img');
                        img.setAttribute('src', ICON_FOLDER +
data[type][i][3]);
                        img.setAttribute('alt', data[type][i][1]);
                        img.setAttribute('title', 'Нажмите, чтобы
добавить объект на холст');
                        p.innerHTML = data[type][i][1];
                        li.appendChild(img);
                        li.appendChild(p);
                        ul.appendChild(li);
                    }
                    listFlowers.appendChild(ul);
                }
            } else if (type === 'bush') {
                if (data[type].length > 0) {
                    let ul = document.createElement('ul');
                    for (let i = 0; i < data[type].length; i++) {
                        let li = document.createElement('li');
                        let p = document.createElement("p");
                        let img = document.createElement('img');
                        img.setAttribute('src', ICON_FOLDER +
data[type][i][3]);
                        img.setAttribute('alt', data[type][i][1]);
                        img.setAttribute('title', 'Нажмите, чтобы
добавить объект на холст');
                        p.innerHTML = data[type][i][1];
                        li.appendChild(img);
                        li.appendChild(p);
                        ul.appendChild(li);
                    }
                    listBushes.appendChild(ul);
                }
            } else if (type === 'tree') {
                if (data[type].length > 0) {

```

## Продолжение приложения В

```
let ul = document.createElement('ul');
for (let i = 0; i < data[type].length; i++) {
  let li = document.createElement('li');
  let p = document.createElement("p");
  let img = document.createElement('img');
  img.setAttribute('src', ICON_FOLDER +

data[type][i][3]);

  img.setAttribute('alt', data[type][i][1]);
  img.setAttribute('title', 'Нажмите, чтобы

добавить объект на холст');

  p.innerHTML = data[type][i][1];
  li.appendChild(img);
  li.appendChild(p);
  ul.appendChild(li);
}
listTrees.appendChild(ul);
}
} else if (type === 'water_plant') {
  if (data[type].length > 0) {
    let ul = document.createElement('ul');
    for (let i = 0; i < data[type].length; i++) {
      let li = document.createElement('li');
      let p = document.createElement("p");
      let img = document.createElement('img');
      img.setAttribute('src', ICON_FOLDER +

data[type][i][3]);

      img.setAttribute('alt', data[type][i][1]);
      img.setAttribute('title', 'Нажмите, чтобы

добавить объект на холст');

      p.innerHTML = data[type][i][1];
      li.appendChild(img);
      li.appendChild(p);
      ul.appendChild(li);
    }
    listWaterPlants.appendChild(ul);
  }
} else if (type === 'pound') {
  if (data[type].length > 0) {
    let ul = document.createElement('ul');
    for (let i = 0; i < data[type].length; i++) {
      let li = document.createElement('li');
      let p = document.createElement("p");
      let img = document.createElement('img');
      img.setAttribute('src', ICON_FOLDER +

data[type][i][3]);

      img.setAttribute('alt', data[type][i][1]);
      img.setAttribute('title', 'Нажмите, чтобы

добавить объект на холст');

      p.innerHTML = data[type][i][1];
      li.appendChild(img);
      li.appendChild(p);
      ul.appendChild(li);
    }
    listPounds.appendChild(ul);
  }
}
}
},
/**
 * Функция отображения карточки объекта путем заполнения HTML-элементов
 данными со стороны сервера
 * Принимает JSON-объект
 * Ничего не возвращает
 */
showCard: function(data) {
  if (data[0] === 'flower') {
    objectName.innerHTML = data[1][1];
    objectPicture.setAttribute('src', IMG_FOLDER + data[1][2]);
    objectPicture.setAttribute('alt', data[1][1]);
  }
}
```

## Продолжение приложения В

```
objectClimate.querySelector('p').innerHTML = data[1][4];
objectGround.querySelector('p').innerHTML = data[1][5];
objectEndurance.querySelector('p').innerHTML = data[1][6];
objectLifeTime.querySelector('p').innerHTML = data[1][7] + '
лет';
objectAirHumidity.querySelector('p').innerHTML = data[1][8] + '%'
- ' + data[1][9] + '%';
objectLight.querySelector('p').innerHTML = data[1][10];
objectTemperature.querySelector('p').innerHTML = data[1][11] +
'&deg;C - ' + data[1][12] + '&deg;C';
objectWatering.querySelector('p').innerHTML = data[1][13];
objectPlantTime.querySelector('p').innerHTML = data[1][14];
objectPitDeep.querySelector('p').innerHTML = data[1][15] / 100 +
' м';
objectDescription.querySelector('p').innerHTML = data[1][16];
if ((data[1][17] != null) || (data[1][17] !== ''))
objectRemark.querySelector('p').innerHTML = data[1][17];

this.showElement(objectClimate);
this.showElement(objectGround);
this.hideElement(objectWaterDeep);
this.showElement(objectEndurance);
this.showElement(objectLifeTime);
this.showElement(objectAirHumidity);
this.showElement(objectLight);
this.showElement(objectTemperature);
this.showElement(objectWatering);
this.showElement(objectPlantTime);
this.showElement(objectPitDeep);
this.hideElement(objectPitRadius);
this.hideElement(objectDistance);
this.hideElement(poundDeep);
this.hideElement(poundLength);
this.hideElement(poundWidth);
this.hideElement(poundWaterLevel);
if ((data[1][17] == null) || (data[1][17] === '')) {
    this.hideElement(objectRemark);
}
this.hideElement(listTip);
this.showElement(objectCard);
} else if (data[0] === 'water_plant') {
objectName.innerHTML = data[1][1];
objectPicture.setAttribute('src', IMG_FOLDER + data[1][2]);
objectPicture.setAttribute('alt', data[1][1]);
objectClimate.querySelector('p').innerHTML = data[1][4];
objectWaterDeep.querySelector('p').innerHTML = data[1][5] / 100 +
' м';
objectEndurance.querySelector('p').innerHTML = data[1][6];
objectLifeTime.querySelector('p').innerHTML = data[1][7] + '
лет';
objectAirHumidity.querySelector('p').innerHTML = data[1][8] + '%'
- ' + data[1][9] + '%';
objectLight.querySelector('p').innerHTML = data[1][10];
objectTemperature.querySelector('p').innerHTML = data[1][11] +
'&deg;C - ' + data[1][12] + '&deg;C';
objectPlantTime.querySelector('p').innerHTML = data[1][13];
objectDescription.querySelector('p').innerHTML = data[1][14];
if ((data[1][15] != null) || (data[1][15] !== ''))
objectRemark.querySelector('p').innerHTML = data[1][15];

this.showElement(objectClimate);
this.hideElement(objectGround);
this.showElement(objectWaterDeep);
this.showElement(objectEndurance);
this.showElement(objectLifeTime);
this.showElement(objectAirHumidity);
this.showElement(objectLight);
this.showElement(objectTemperature);
this.hideElement(objectWatering);
```

```

        this.showElement(objectPlantTime);
        this.hideElement(objectPitDeep);
        this.hideElement(objectPitRadius);
        this.hideElement(objectDistance);
        this.hideElement(poundDeep);
        this.hideElement(poundLength);
        this.hideElement(poundWidth);
        this.hideElement(poundWaterLevel);
        if ((data[1][15] == null) || (data[1][15] === '')) {
            this.hideElement(objectRemark);
        }
        this.hideElement(listTip);
        this.showElement(objectCard);
    } else if (data[0] === 'bush') {
        objectName.innerHTML = data[1][1];
        objectPicture.setAttribute('src', IMG_FOLDER + data[1][2]);
        objectPicture.setAttribute('alt', data[1][1]);
        objectClimate.querySelector('p').innerHTML = data[1][4];
        objectGround.querySelector('p').innerHTML = data[1][5];
        objectEndurance.querySelector('p').innerHTML = data[1][6];
        objectLifeTime.querySelector('p').innerHTML = data[1][7] + '
лет';
        objectAirHumidity.querySelector('p').innerHTML = data[1][8] + '%
- ' + data[1][9] + '%';
        objectLight.querySelector('p').innerHTML = data[1][10];
        objectTemperature.querySelector('p').innerHTML = data[1][11] +
'&deg;C - ' + data[1][12] + '&deg;C';
        objectWatering.querySelector('p').innerHTML = data[1][13];
        objectPlantTime.querySelector('p').innerHTML = data[1][14];
        objectPitDeep.querySelector('p').innerHTML = data[1][15] / 100 +
' м';
        objectPitRadius.querySelector('p').innerHTML = data[1][16] / 100
+ ' м';
        objectDescription.querySelector('p').innerHTML = data[1][17];
        if ((data[1][18] != null) || (data[1][18] !== ''))
objectRemark.querySelector('p').innerHTML = data[1][18];

        this.showElement(objectClimate);
        this.showElement(objectGround);
        this.hideElement(objectWaterDeep);
        this.showElement(objectEndurance);
        this.showElement(objectLifeTime);
        this.showElement(objectAirHumidity);
        this.showElement(objectLight);
        this.showElement(objectTemperature);
        this.showElement(objectWatering);
        this.showElement(objectPlantTime);
        this.showElement(objectPitDeep);
        this.showElement(objectPitRadius);
        this.hideElement(objectDistance);
        this.hideElement(poundDeep);
        this.hideElement(poundLength);
        this.hideElement(poundWidth);
        this.hideElement(poundWaterLevel);
        if ((data[1][18] == null) || (data[1][18] === '')) {
            this.hideElement(objectRemark);
        }
        this.hideElement(listTip);
        this.showElement(objectCard);
    } else if (data[0] === 'tree') {
        objectName.innerHTML = data[1][1];
        objectPicture.setAttribute('src', IMG_FOLDER + data[1][2]);
        objectPicture.setAttribute('alt', data[1][1]);
        objectClimate.querySelector('p').innerHTML = data[1][4];
        objectGround.querySelector('p').innerHTML = data[1][5];
        objectEndurance.querySelector('p').innerHTML = data[1][6];
        objectLifeTime.querySelector('p').innerHTML = data[1][7] + '
лет';

```

## Продолжение приложения В

```

- ' + data[1][9] + '%';
objectAirHumidity.querySelector('p').innerHTML = data[1][8] + '%
objectLight.querySelector('p').innerHTML = data[1][10];
objectTemperature.querySelector('p').innerHTML = data[1][11] +
'&deg;C - ' + data[1][12] + '&deg;C';
objectWatering.querySelector('p').innerHTML = data[1][13];
objectPlantTime.querySelector('p').innerHTML = data[1][14];
objectPitDeep.querySelector('p').innerHTML = data[1][15] / 100 +
' м';
objectPitRadius.querySelector('p').innerHTML = data[1][16] / 100
+ ' м';
objectDistance.querySelector('p').innerHTML = data[1][17] / 100 +
' м';
objectDescription.querySelector('p').innerHTML = data[1][18];
if ((data[1][19] != null) || (data[1][19] !== ''))
objectRemark.querySelector('p').innerHTML = data[1][19];

this.showElement(objectClimate);
this.showElement(objectGround);
this.hideElement(objectWaterDeep);
this.showElement(objectEndurance);
this.showElement(objectLifeTime);
this.showElement(objectAirHumidity);
this.showElement(objectLight);
this.showElement(objectTemperature);
this.showElement(objectWatering);
this.showElement(objectPlantTime);
this.showElement(objectPitDeep);
this.showElement(objectPitRadius);
this.showElement(objectDistance);
this.hideElement(poundDeep);
this.hideElement(poundLength);
this.hideElement(poundWidth);
this.hideElement(poundWaterLevel);
if ((data[1][19] == null) || (data[1][19] === '')) {
    this.hideElement(objectRemark);
}
this.hideElement(listTip);
this.showElement(objectCard);
} else if (data[0] === 'pound') {
objectName.innerHTML = data[1][1];
objectPicture.setAttribute('src', IMG_FOLDER + data[1][2]);
objectPicture.setAttribute('alt', data[1][1]);
poundDeep.querySelector('p').innerHTML = data[1][4] / 100 + ' м';
poundLength.querySelector('p').innerHTML = data[1][5] / 100 + '
м';
poundWidth.querySelector('p').innerHTML = data[1][6] / 100 + '
м';
poundWaterLevel.querySelector('p').innerHTML = data[1][7] / 100 +
' м';
objectDescription.querySelector('p').innerHTML = data[1][8];
if ((data[1][9] != null) || (data[1][9] !== ''))
objectRemark.querySelector('p').innerHTML = data[1][9];

this.hideElement(objectClimate);
this.hideElement(objectGround);
this.hideElement(objectWaterDeep);
this.hideElement(objectEndurance);
this.hideElement(objectLifeTime);
this.hideElement(objectAirHumidity);
this.hideElement(objectLight);
this.hideElement(objectTemperature);
this.hideElement(objectWatering);
this.hideElement(objectPlantTime);
this.hideElement(objectPitDeep);
this.hideElement(objectPitRadius);
this.hideElement(objectDistance);
this.showElement(poundDeep);
this.showElement(poundLength);

```

```

        this.showElement(poundWidth);
        this.showElement(poundWaterLevel);
        if ((data[1][9] == null) || (data[1][9] === '')) {
            this.hideElement(objectRemark);
        }
        this.hideElement(listTip);
        this.showElement(objectCard);
    }
},
/**
 * Функция отображения списка цветов
 * Ничего не принимает
 * Ничего не возвращает
 */
showFlowers: function () {
    this.showListElements();
    this.hideElement(objectCard);
    this.showElement(listFlowers);
    this.hideElement(listBushes);
    this.hideElement(listTrees);
    this.hideElement(listWaterPlants);
    this.hideElement(listPounds);
},
/**
 * Функция отображения списка кустарников
 * Ничего не принимает
 * Ничего не возвращает
 */
showBushes: function () {
    this.showListElements();
    this.hideElement(listFlowers);
    this.showElement(listBushes);
    this.hideElement(listTrees);
    this.hideElement(listWaterPlants);
    this.hideElement(listPounds);
},
/**
 * Функция отображения списка деревьев
 * Ничего не принимает
 * Ничего не возвращает
 */
showTrees: function () {
    this.showListElements();
    this.hideElement(listFlowers);
    this.hideElement(listBushes);
    this.showElement(listTrees);
    this.hideElement(listWaterPlants);
    this.hideElement(listPounds);
},
/**
 * Функция отображения списка водных растений
 * Ничего не принимает
 * Ничего не возвращает
 */
showWaterPlants: function () {
    this.showListElements();
    this.hideElement(listFlowers);
    this.hideElement(listBushes);
    this.hideElement(listTrees);
    this.showElement(listWaterPlants);
    this.hideElement(listPounds);
},
/**
 * Функция отображения списка водоемов
 * Ничего не принимает
 * Ничего не возвращает
 */
showPounds: function () {
    this.showListElements();

```

```

        this.hideElement(listFlowers);
        this.hideElement(listBushes);
        this.hideElement(listTrees);
        this.hideElement(listWaterPlants);
        this.showElement(listPounds);
    },
    /**
     * Функция отображения управляющих элементов для изменения свойств контура
     добавляемых фигур "Линия", "Кисть" (Толщина, цвет, прозрачность)
     * Ничего не принимает
     * Ничего не возвращает
     */
    showLineProperties: function () {
        this.hideElement(actionBtn);
        this.hideElement(listItems);
        this.hideElement(editTip);
        this.hideElement(listTip);
        this.showElement(addTip);
        this.showElement(properties);
        this.hideElement(patternTypes);
        this.hideElement(grassPatterns);
        this.hideElement(groundPatterns);
        this.hideElement(woodPatterns);
        this.hideElement(tilePatterns);
        this.showElement(lineProperties);
        this.hideElement(fillProperties);
        this.hideElement(textAreaProperties);
    },
    /**
     * Функция отображения управляющих элементов для изменения свойств контура и
     заливки добавляемых фигур "Окружность", "Прямоугольник" (Толщина контура, цвет контура,
     прозрачность контура, цвет заливки, прозрачность заливки)
     * Ничего не принимает
     * Ничего не возвращает
     */
    showLineAndFillProperties: function () {
        this.hideElement(actionBtn);
        this.hideElement(listItems);
        this.hideElement(listTip);
        this.showElement(addTip);
        this.hideElement(editTip);
        this.showElement(properties);
        this.hideElement(patternTypes);
        this.hideElement(grassPatterns);
        this.hideElement(groundPatterns);
        this.hideElement(woodPatterns);
        this.hideElement(tilePatterns);
        this.showElement(lineProperties);
        this.showElement(fillProperties);
        this.hideElement(textAreaProperties);
        this.showElement(textArea);
        this.showElement(textAreaLabel);
        this.showElement(textSubmit);
        this.showElement(textErase);
        this.hideElement(objectCard);
    },
    /**
     * Функция отображения управляющих элементов для изменения свойств текста
     добавляемого объекта "Текст" (Поле ввода текста, размер текста, цвет текста, прозрачность
     текста)
     * Ничего не принимает
     * Ничего не возвращает
     */
    showTextProperties: function () {
        this.hideElement(actionBtn);
        this.hideElement(listItems);
        this.hideElement(objectCard);
        this.hideElement(editTip);
        this.hideElement(listTip);
    }
}

```

```

        this.showElement( addTip );
        this.showElement( properties );
        this.hideElement( patternTypes );
        this.hideElement( grassPatterns );
        this.hideElement( groundPatterns );
        this.hideElement( woodPatterns );
        this.hideElement( tilePatterns );
        this.hideElement( lineProperties );
        this.hideElement( fillProperties );
        this.showElement( textAreaLabel );
        this.showElement( textArea );
        this.showElement( textAreaProperties );
        this.showElement( textSubmit );
        this.showElement( textErase );
    },
    /**
     * Функция отображения списка всех объектов
     * Ничего не принимает
     * Ничего не возвращает
     */
    showListElements: function () {
        this.hideElement( actionBtn );
        this.hideElement( properties );
        this.hideElement( patternTypes );
        this.hideElement( grassPatterns );
        this.hideElement( groundPatterns );
        this.hideElement( woodPatterns );
        this.hideElement( tilePatterns );
        this.showElement( listItems );
        this.showElement( listFlowers );
        this.showElement( listBushes );
        this.showElement( listTrees );
        this.showElement( listWaterPlants );
        this.showElement( listPounds );
        this.hideElement( editTip );
        this.showElement( listTip );
        this.hideElement( addTip );
        this.hideElement( objectCard );
    },
    /**
     * Функция отображения управляющих элементов для изменения свойств уже
     существующих фигур "Кисть", "Линия", "Прямоугольник", "Окружность", "Текст"
     * Ничего не принимает
     * Ничего не возвращает
     */
    showAllProperties: function () {
        this.hideElement( actionBtn );
        this.hideElement( listItems );
        this.showElement( editTip );
        this.hideElement( listTip );
        this.hideElement( addTip );
        this.hideElement( patternTypes );
        this.hideElement( grassPatterns );
        this.hideElement( groundPatterns );
        this.hideElement( woodPatterns );
        this.hideElement( tilePatterns );
        this.showElement( properties );
        this.showElement( lineProperties );
        this.showElement( fillProperties );
        this.showElement( textAreaProperties );
        this.hideElement( textArea );
        this.hideElement( textAreaLabel );
        this.hideElement( textSubmit );
        this.hideElement( textErase );
        this.hideElement( objectCard );
    },
    /**
     * Функция отображения всех типов покрытий
     * Ничего не принимает

```

```

    * Ничего не возвращает
    */
    showPatterns: function () {
        this.hideElement(actionBtn);
        this.hideElement(listItems);
        this.hideElement(properties);
        this.showElement(patternTypes);
    },
    /**
    * Функция отображения всех вариантов покрытий типа "Газон"
    * Ничего не принимает
    * Ничего не возвращает
    */
    showGrassPatterns: function () {
        this.showElement(grassPatterns);
        this.hideElement(groundPatterns);
        this.hideElement(woodPatterns);
        this.hideElement(tilePatterns);
        this.hideElement(listTip);
        this.hideElement(editTip);
        this.hideElement(addTip);
        this.hideElement(objectCard);
    },
    /**
    * Функция отображения всех вариантов покрытий типа "Грунт"
    * Ничего не принимает
    * Ничего не возвращает
    */
    showGroundPatterns: function () {
        this.hideElement(grassPatterns);
        this.showElement(groundPatterns);
        this.hideElement(woodPatterns);
        this.hideElement(tilePatterns);
        this.hideElement(listTip);
        this.hideElement(editTip);
        this.hideElement(addTip);
        this.hideElement(objectCard);
    },
    /**
    * Функция отображения всех вариантов покрытий типа "Деревянное покрытие"
    * Ничего не принимает
    * Ничего не возвращает
    */
    showWoodPatterns: function () {
        this.hideElement(grassPatterns);
        this.hideElement(groundPatterns);
        this.showElement(woodPatterns);
        this.hideElement(tilePatterns);
        this.hideElement(listTip);
        this.hideElement(editTip);
        this.hideElement(addTip);
        this.hideElement(objectCard);
    },
    /**
    * Функция отображения всех вариантов покрытий типа "Камень и плитка"
    * Ничего не принимает
    * Ничего не возвращает
    */
    showTilePatterns: function () {
        this.hideElement(grassPatterns);
        this.hideElement(groundPatterns);
        this.hideElement(woodPatterns);
        this.showElement(tilePatterns);
        this.hideElement(listTip);
        this.hideElement(editTip);
        this.hideElement(addTip);
        this.hideElement(objectCard);
    },
    /**

```

```
* Функция закрытия карточки объекта
* Ничего не принимает
* Ничего не возвращает
*/
closeCard: function () {
    this.hideElement(objectCard);
    this.showElement(listTip);
},
/**
 * Функция перевода типа объекта с английского на русский
 * Принимает тип объекта на английском языке
 * Возвращает тип объекта на русском языке
 */
checkObjectGroup: function (group) {
    switch (group) {
        case 'list-flowers':
            return 'flower';
        case 'list-bushes':
            return 'bush';
        case 'list-trees':
            return 'tree';
        case 'list-water-plants':
            return 'water_plant';
        case 'list-pounds':
            return 'pound';
    }
}
})();
```

## ПРИЛОЖЕНИЕ Г

### ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА BACKEND

#### Листинг Г.1 – Исходный код файла “backend.js”

```
'use strict';

/**
 * Немедленно выполняемая функция, предназначенная для отправки HTTP-запросов на сторону
 сервера
 * Ничего не принимает
 * Возвращает функции для отправки/получения данных со стороны сервера
 */
window.backend = (function () {
    let URL_FOR_GET_LIST = 'http://localhost/landscape-editor/search-data.php',
        URL_FOR_GET_OBJECT = 'http://localhost/landscape-editor/search-object.php',
        URL_FOR_GET_OPTIONS = 'http://localhost/landscape-editor/search-select-
options.php';

    return {
        /**
         * Функция получения вариантов выбора условий
         * Ничего не принимает
         * Ничего не возвращает
         */
        loadSelectOptions: function () {
            let xhr = new XMLHttpRequest();
            xhr.responseType = 'json';
            xhr.addEventListener('load', function () {
                if (xhr.status === 200) {
                    window.util.addSelectOptions(xhr.response);
                } else {
                    window.util.isError('Ошибка запроса: ' + xhr.status + ' / ' +
xhr.statusText);
                }
            });
            xhr.addEventListener('error', function () {
                window.util.isError('Ошибка соединения');
            });
            xhr.addEventListener('timeout', function () {
                window.util.isError('Превышено время ожидания');
            });
            xhr.timeout = 5000;
            xhr.open('GET', URL_FOR_GET_OPTIONS);
            xhr.send();
        },
        /**
         * Функция отправки выбранных условий на сторону сервера для получения объектов из
 базы данных
         * Принимает JSON-объект
         * Ничего не возвращает
         */
        saveSearchOptions: function (data) {
            let xhr = new XMLHttpRequest();
            xhr.responseType = 'json';
            xhr.addEventListener('load', function () {
                if (xhr.status === 200) {
                    window.util.addObjects(xhr.response);
                } else {
                    window.util.isError('Ошибка запроса: ' + xhr.status + ' / ' +
xhr.statusText);
                }
            });
            xhr.addEventListener('error', function () {
                window.util.isError('Ошибка соединения');
            });
        }
    };
});
```

```

    xhr.addEventListener('timeout', function () {
        window.util.isError('Превышено время ожидания');
    });
    xhr.timeout = 5000;
    xhr.open('POST', URL_FOR_GET_LIST);
    xhr.send(data);
},
/**
 * Функция получения полной информации об объекте
 * Принимает JSON-объект
 * Ничего не возвращает
 */
searchObject: function (data) {
    let xhr = new XMLHttpRequest();
    xhr.responseType = 'json';
    xhr.addEventListener('load', function () {
        if (xhr.status === 200) {
            window.util.showCard(xhr.response);
        } else {
            window.util.isError('Ошибка запроса: ' + xhr.status + ' / ' +
xhr.statusText);
        }
    });
    xhr.addEventListener('error', function () {
        window.util.isError('Ошибка соединения');
    });
    xhr.addEventListener('timeout', function () {
        window.util.isError('Превышено время ожидания');
    });
    xhr.timeout = 5000;
    xhr.open('POST', URL_FOR_GET_OBJECT);
    xhr.send(data);
}
};
})();

```

## ПРИЛОЖЕНИЕ Д

### ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА FORM

Листинг Д.1 – Исходный код файла “form.js”

```
'use strict';

/**
 * Немедленно выполняемая функция, предназначенная для обработки данных отправляемой формы
 * начальных условий
 * Ничего не принимает
 * Ничего не возвращает
 */
(function () {
    let form = document.querySelector('form');

    let climateSelect = document.getElementById('climate'),
        lightSelect = document.getElementById('light'),
        wateringSelect = document.getElementById('watering'),
        enduranceSelect = document.getElementById('endurance'),
        temperatureMaxInput = document.getElementById('temperature-max'),
        airHumidityMaxInput = document.getElementById('air-humidity-max'),
        areaWidth = document.getElementById('area-width'),
        areaHeight = document.getElementById('area-height');

    let climate,
        light,
        watering,
        endurance,
        temperature_max,
        air_humidity_max,
        area_width,
        area_height;

    window.backend.loadSelectOptions();

    form.addEventListener('submit', function (e) {
        e.preventDefault();

        climate = climateSelect.options[climateSelect.selectedIndex].value;
        light = lightSelect.options[lightSelect.selectedIndex].value;
        watering = wateringSelect.options[wateringSelect.selectedIndex].value;
        endurance = enduranceSelect.options[enduranceSelect.selectedIndex].value;
        temperature_max = temperatureMaxInput.value;
        air_humidity_max = airHumidityMaxInput.value;
        area_width = parseInt(areaWidth.value, 10) || 1000;
        area_height = parseInt(areaHeight.value, 10) || 500;

        if (temperature_max !== '') {
            temperature_max = parseInt(temperatureMaxInput.value, 10) || 0;
        } else temperature_max = 100;
        if (air_humidity_max !== '') {
            air_humidity_max = parseInt(airHumidityMaxInput.value, 10) || 0;
        } else air_humidity_max = 100;

        let json = {
            "climate": climate,
            "light": light,
            "watering": watering,
            "endurance": endurance,
            "temperature_max": temperature_max,
            "air_humidity_max": air_humidity_max
        };
        let data = JSON.stringify(json);

        window.backend.saveSearchOptions(data);
    });
});
```

```
let canvas = document.createElement('canvas');
canvas.id = 'this-canvas';
canvas.width = area_width;
canvas.height = area_height;
document.querySelector('.canvas-middle').appendChild(canvas);

let script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'js/canvas.js';
script.defer = 'defer';
document.querySelector('head').appendChild(script);

window.util.closeForm();
});
})();
```

## ПРИЛОЖЕНИЕ Е

### ИСПОДНЫЙ КОД JAVASCRIPT-ФАЙЛА TOOLS

#### Листинг Е.1 – Исходный код файла “tools.js”

```
'use strict';

/**
 * Немедленно выполняемая функция, предназначенная для работы с объектами редактора
 * Ничего не принимает
 * Возвращает функции управления состояниями объектов редактора (изменить, добавить,
 удалить)
 */
window.tools = (function () {
    return {
        /**
         * Функция изменения свойства объекта
         * Принимает холст, свойство, значение
         * Ничего не возвращает
         */
        setObjectProperty: function (canvas, property, value) {
            let object = canvas.getActiveObject();
            if (!object) return;
            if (property === 'fill') {
                if ((object.get('type') === 'line') ||
                    (object.get('type') === 'rect') ||
                    (object.get('type') === 'circle') ||
                    (object.get('type') === 'text')) {
                    object.set(property, value);
                    canvas.renderAll();
                }
            } else if ((property === 'stroke') || (property === 'strokeWidth')) {
                if ((object.get('type') === 'line') ||
                    (object.get('type') === 'rect') ||
                    (object.get('type') === 'circle') ||
                    (object.get('type') === 'path')) {
                    object.set(property, value);
                    canvas.renderAll();
                }
            } else if (property === 'fontSize') {
                if (object.get('type') === 'text') {
                    object.set('fontSize', value);
                    canvas.renderAll();
                }
            }
        },
        /**
         * Функция конвертирования цвета из формата HEX в RGBA
         * Принимает цвет в формате HEX, прозрачность
         * Возвращает цвет в формате RGBA
         */
        setColor: function (color, opacity) {
            return "rgba(" +
                parseInt(color.slice(-6, -4), 16) + ", " +
                parseInt(color.slice(-4, -2), 16) + ", " +
                parseInt(color.slice(-2), 16) + ", " +
                opacity + ")";
        },
        /**
         * Функция добавления фигуры с заливкой-покрытием на холст
         * Принимает URL покрытия, холст
         * Ничего не возвращает
         */
        addPatternObject: function (url, canvas) {
            fabric.Image.fromURL(url, function(img) {
                img.scaleToWidth(100);
            });
        }
    };
});
```

```

img.scaleToHeight(100);
let patternSourceCanvas = new fabric.StaticCanvas();
patternSourceCanvas.add(img);
patternSourceCanvas.setDimensions({
  width: img.getScaledWidth(),
  height: img.getScaledHeight()
});
patternSourceCanvas.renderAll();
let pattern = new fabric.Pattern({
  source: patternSourceCanvas.getElement(),
  repeat: 'repeat'
});
let rect = new fabric.Rect({
  width: 100,
  height: 100,
  fill: pattern,
  objectCaching: false,
  name: 'pattern',
  borderColor: 'red',
  cornerColor: 'rgba(255, 0, 0, .5)',
  cornerStyle: 'circle',
  transparentCorners: false,
});
canvas.add(rect);
rect.center().setCoords();
});
},
/**
 * Функция клонирования объекта
 * Принимает холст
 * Ничего не возвращает
 */
cloneObject: function (canvas) {
  let activeObject = canvas.getActiveObject();
  if (activeObject) {
    activeObject.clone(function (cloned) {
      canvas.discardActiveObject();
      cloned.set({
        top: cloned.top + 20,
        evented: true,
        borderColor: 'red',
        cornerColor: 'rgba(255, 0, 0, .5)',
        cornerStyle: 'circle',
        transparentCorners: false,
      });
    });
    if (cloned.type === 'activeSelection') {
      cloned.canvas = canvas;
      cloned.forEachObject(function (obj) {
        canvas.add(obj);
      });
      cloned.setCoords();
    } else {
      canvas.add(cloned);
    }
    canvas.setActiveObject(cloned);
    canvas.requestRenderAll();
  }
});
},
/**
 * Функция удаления выбранных объектов с холста
 * Принимает холст
 * Ничего не возвращает
 */
deleteSelectedObjects: function (canvas) {
  let selection = canvas.getActiveObject();
  if (!selection) {
    alert("Выберите объект!");
  } else {

```

```

        if (selection.type === 'activeSelection') {
            selection.forEachObject(function (element) {
                canvas.remove(element);
            });
        } else {
            canvas.remove(selection);
        }
        canvas.discardActiveObject();
        canvas.requestRenderAll();
    }
},
/**
 * Функция добавления текста на холст
 * Принимает холст, текст, размер текста, цвет текста
 * Ничего не возвращает
 */
addText: function (canvas, text, size, color) {
    let text_object = new fabric.Text(text, {
        left: 10,
        top: 10,
        name: 'Text',
        fontSize: parseInt(size, 10) || 20,
        fontFamily: "Arial",
        fill: color,
        selectable: true,
        borderColor: 'red',
        cornerColor: 'rgba(255, 0, 0, .5)',
        cornerStyle: 'circle',
        transparentCorners: false,
    });
    canvas.add(text_object);
    canvas.renderAll();
},
/**
 * Функция добавления иконки объекта на холст
 * Принимает холст, URL иконки, тип объекта
 * Ничего не возвращает
 */
addSVGIcon: function (canvas, url, type) {
    fabric.loadSVGFromURL(url, function(objects) {
        let image = fabric.util.groupSVGElements(objects, {
            name: type,
            left: 50,
            top: 50,
            originX: 'center',
            originY: 'center',
            lockRotating: true,
            lockRotation: true,
            borderColor: 'red',
            cornerColor: 'rgba(255, 0, 0, .5)',
            cornerStyle: 'circle',
            transparentCorners: false,
        });
        if (type === 'tree') {
            image.padding = 20;
        }
        canvas.add(image);
    });
},
/**
 * Функция скачивания изображения холста
 * Принимает холст
 * Ничего не возвращает
 */
downloadImg: function (canvas) {
    let a = document.createElement('a');
    a.href = canvas.toDataURL();
    a.download = "Чертеж ландшафта участка.png";
    a.click();
}

```

```

},
/**
 * Функция задания холсту заднего плана
 * Принимает холст
 * Ничего не возвращает
 */
addBackGroundImage: function (canvas) {
  fabric.Image.fromURL('img/grid.png', function(img) {
    canvas.setBackgroundImage(img, canvas.renderAll.bind(canvas), {
      width: img.width,
      height: img.height
    });
  });
},
/**
 * Функция перемещения выбранного объекта на передний план
 * Принимает холст
 * Ничего не возвращает
 */
toFront: function (canvas) {
  let obj = canvas.getActiveObject();
  if (!obj) return;
  if ((obj.get('type') === 'rect') || (obj.get('type') === 'circle')) {
    canvas.bringToFront(obj).discardActiveObject().renderAll();
  }
},
/**
 * Функция перемещения выбранного объекта на задний план
 * Принимает холст
 * Ничего не возвращает
 */
toBack: function (canvas) {
  let obj = canvas.getActiveObject();
  if (!obj) return;
  if ((obj.get('type') === 'rect') || (obj.get('type') === 'circle')) {
    canvas.sendToBack(obj).discardActiveObject().renderAll();
  }
},
/**
 * Функция обновления отображения теней объектов на холсте
 * Принимает логическое значение тени, холст, смещение солнца по оси Y, смещение
солнца по оси X
 * Ничего не возвращает
 */
updateShadow: function (isShadow, canvas, sunY, sunX) {
  if (isShadow) {
    canvas.getObjects().filter(function (object) {
      if (((object.get('type') === 'rect') && (object.get('name') !==
'pattern')) ||
      (object.get('type') === 'circle') ||
      ((object.get('type') === 'group') &&
      (object.get('name') === 'tree') || (object.get('name') ===
'bush')))) {

        let shadowX, shadowY;

        if (object.left > sunX) {
          shadowX = object.left - sunX;
          if (shadowX > (object.width / 2)) shadowX = object.width / 2;
        } else {
          shadowX = (sunX - object.left) * (-1);
          if (shadowX < (object.width / 2 * (-1))) shadowX =
(object.width / 2) * (-1);
        }

        if (object.top > sunY) {
          shadowY = object.top - sunY;

```

```

                if (shadowY > (object.height / 2)) shadowY = (object.height) /
2;
                } else {
                    shadowY = (sunY - object.top) * (-1);
                    if (shadowY < (object.height / 2 * (-1))) shadowY =
(object.height / 2) * (-1);
                }

                object.setShadow({
                    color: 'rgba(0, 0, 0, .4)',
                    blur: 0,
                    offsetX: shadowX,
                    offsetY: shadowY
                });
                canvas.renderAll();
            }
        });
    } else {
        canvas.getObjects().filter(object => {
            object.setShadow('none');
        });
        canvas.renderAll();
    }
},
/**
 * Функция отображения солнца на холсте
 * Принимает логическое значения солнца, холст, смещение солнца по оси X, смещение
солнца по оси Y
 * Ничего не возвращает
 */
updateSun: function (isSun, canvas, sunX, sunY) {
    if (isSun) {
        fabric.loadSVGFromURL('img/sunny.svg', function (objects) {
            let image = fabric.util.groupSVGElements(objects, {
                name: 'sun',
                top: sunY,
                left: sunX,
                lockUniScaling: true,
                lockRotation: true,
                lockScalingX: true,
                lockScalingY: true,
                hasBorders: false,
                transparentCorners: true,
                originX: 'center',
                originY: 'center',
                cornerColor: 'transparent',
            });
            canvas.add(image);
            canvas.bringToFront(image).discardActiveObject().renderAll();
        });
    } else {
        canvas.getObjects().filter(function (object) {
            if (object.get('name') === 'sun') {
                canvas.remove(object);
            }
        });
    }
},
/**
 * Функция проверки пересечения объектов
 * Принимает холст, объект события
 * Ничего не возвращает
 */
checkObjectIntersection: function (canvas, evt) {
    evt.target.setCoords();
    canvas.forEachObject(function (object) {
        if (object === evt.target) return;
        if ((object.get('name') === 'tree') || (object.get('name') === 'bush')) {

```

## Окончание приложения Е

```
1);
    object.set('opacity', evt.target.intersectsWithObject(object) ? 0.3 :
    });
  }
}
})();
```

## ПРИЛОЖЕНИЕ Ж

### ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА HISTORY

#### Листинг Ж.1 – Исходный код файла “history.js”

```
'use strict';

/**
 * Функция, необходимая для управления состоянием холста
 * Принимает холст
 * Возвращает функции для управления состоянием холста
 */
let History = function (_canvas) {
  let canvas = _canvas;
  let config = {
    canvasState           : [],
    currentStateIndex     : -1,
    undoStatus            : false,
    redoStatus            : false,
    undoFinishedStatus    : 1,
    redoFinishedStatus    : 1,
    undoButton            : document.querySelector('.undo'),
    redoButton           : document.querySelector('.redo'),
  };

  return {
    /**
     * Функция сохранения состояния холста в стек состояний
     * Ничего не принимает
     * Ничего не возвращает
     */
    save: function () {
      if ((config.undoStatus === false && config.redoStatus === false)) {
        let jsonData = canvas.toJSON(['name',
          'padding',
          'cornerColor',
          'borderColor',
          'cornerStyle',
          'transparentCorners',
          'hasBorders'
        ]);
        let canvasAsJson = JSON.stringify(jsonData);
        if (config.currentStateIndex < config.canvasState.length - 1) {
          let indexToBeInserted = config.currentStateIndex + 1;
          config.canvasState[indexToBeInserted] = canvasAsJson;
          let numberOfElementsToRetain = indexToBeInserted + 1;
          config.canvasState = config.canvasState.splice(0,
            numberOfElementsToRetain);
        } else {
          config.canvasState.push(canvasAsJson);
        }
        config.currentStateIndex = config.canvasState.length - 1;
        if ((config.currentStateIndex === config.canvasState.length - 1) &&
          config.currentStateIndex !== -1) {
          config.redoButton.disabled = "disabled";
        }
      }
    },
    /**
     * Функция получения предыдущего состояния холста из стека состояний
     * Ничего не принимает
     * Ничего не возвращает
     */
    undo: function () {
      if (config.undoFinishedStatus) {
        if (config.currentStateIndex === -1) {

```

```

        config.undoStatus = false;
    } else {
        if (config.canvasState.length >= 1) {
            config.undoFinishedStatus = 0;
            if (config.currentStateIndex !== 0) {
                config.undoStatus = true;
                canvas.loadFromJSON(config.canvasState[config.currentStateIndex
- 1], () => {
                    canvas.renderAll();
                    config.undoStatus = false;
                    config.currentStateIndex -= 1;
                    config.undoButton.removeAttribute("disabled");
                    if (config.currentStateIndex !== config.canvasState.length
- 1) {
                        config.redoButton.removeAttribute('disabled');
                    }
                    config.undoFinishedStatus = 1;
                });
            } else if (config.currentStateIndex === 0) {
                canvas.clear();
                config.undoFinishedStatus = 1;
                config.undoButton.disabled= "disabled";
                config.redoButton.removeAttribute('disabled');
                config.currentStateIndex -= 1;
            }
        }
    }
},
/**
 * Функция получения следующего состояния холста из стека состояний
 * Ничего не принимает
 * Ничего не возвращает
 */
redo: function () {
    if (config.redoFinishedStatus) {
        if ((config.currentStateIndex === config.canvasState.length -1) &&
            config.currentStateIndex !== -1) {
            config.redoButton.disabled= "disabled";
        } else {
            if (config.canvasState.length > config.currentStateIndex &&
                config.canvasState.length !== 0) {
                config.redoFinishedStatus = 0;
                config.redoStatus = true;
                canvas.loadFromJSON(config.canvasState[config.currentStateIndex +
1], () => {
                    canvas.renderAll();
                    config.redoStatus = false;
                    config.currentStateIndex += 1;
                    if (config.currentStateIndex !== -1) {
                        config.undoButton.removeAttribute('disabled');
                    }
                    config.redoFinishedStatus = 1;
                    if ((config.currentStateIndex === config.canvasState.length -
1) &&
                        config.currentStateIndex !== -1) {
                        config.redoButton.disabled= "disabled";
                    }
                });
            }
        }
    }
}
};

```

## ПРИЛОЖЕНИЕ 3

### ИСХОДНЫЙ КОД JAVASCRIPT-ФАЙЛА CANVAS

#### Листинг 3.1 – Исходный код файла “canvas.js”

```
'use strict';

/**
 * Немедленно выполняемая функция, обрабатывающая события в редакторе
 * Ничего не принимает
 * Ничего не возвращает
 */
window.canvas = (function () {
    let Z_KEY = 90,
        Y_KEY = 89,
        C_KEY = 67,
        BACKSPACE_KEY = 8,
        ESC_KEY = 27;

    let TEXTURE_PATH = '/landscape-editor/texture/';

    let canvas = new fabric.Canvas('this-canvas'),
        actionBtn = document.querySelector(".action-btn"),
        listBtn = document.querySelector(".list"),
        handBtn = document.querySelector(".hand"),
        delBtn = document.querySelector(".delete"),
        brshBtn = document.querySelector(".brush"),
        lineBtn = document.querySelector(".line"),
        recBtn = document.querySelector(".rectangle"),
        crclBtn = document.querySelector(".circle"),
        txtBtn = document.querySelector(".text"),
        rectTexture = document.querySelector('.rectangle-pattern'),
        flower = document.querySelector('.flower'),
        waterFlower = document.querySelector('.water-flower'),
        bush = document.querySelector('.bush'),
        tree = document.querySelector('.tree'),
        pound = document.querySelector('.pound'),
        undoBtn = document.querySelector(".undo"),
        redoBtn = document.querySelector(".redo"),
        unselectBtn = document.querySelector('.unselect'),
        toBackBtn = document.querySelector('.to-backward'),
        toFrontBtn = document.querySelector('.to-forward'),
        addShadowBtn = document.querySelector('.add-shadow'),
        removeShadowBtn = document.querySelector('.remove-shadow'),
        addSunBtn = document.querySelector('.add-sun'),
        removeSunBtn = document.querySelector('.remove-sun'),
        clrBtn = document.querySelector(".clear"),
        dwlndImg = document.querySelector(".save"),
        txtSubmitBtn = document.querySelector(".add-text-area"),
        txtEraseBtn = document.querySelector(".erase-text-area"),
        closeCardBtn = document.querySelector('.close-card');

    let lineWidthRange = document.getElementById("line-width"),
        lineWidthOutput = document.getElementById('line-width-val'),
        lineColorInput = document.getElementById("line-color"),
        lineOpacityRange = document.getElementById("line-color-opacity"),
        lineOpacityValue = document.getElementById("line-color-opacity-val"),
        fillColorInput = document.getElementById("fill-color"),
        fillOpacityRange = document.getElementById("fill-color-opacity"),
        fillOpacityValue = document.getElementById("fill-color-opacity-val"),
        textInput = document.getElementById("text-area"),
        textSizeRange = document.getElementById("text-size"),
        textSizeOutput = document.getElementById('text-size-val'),
        textColorInput = document.getElementById("text-color"),
        textOpacityInput = document.getElementById("text-color-opacity"),
        textOpacityValue = document.getElementById("text-color-opacity-val");
```

```

let itemList = document.querySelector('.list-items');
let grassTexture = document.querySelector('.grass-texture'),
    groundTexture = document.querySelector('.ground-texture'),
    woodTexture = document.querySelector('.wood-texture'),
    tileTexture = document.querySelector('.tile-texture');

let grassOne = document.querySelector('.grass.one'),
    grassTwo = document.querySelector('.grass.two'),
    grassThree = document.querySelector('.grass.three'),
    groundOne = document.querySelector('.ground.one'),
    groundTwo = document.querySelector('.ground.two'),
    groundThree = document.querySelector('.ground.three'),
    woodOne = document.querySelector('.wood.one'),
    woodTwo = document.querySelector('.wood.two'),
    woodThree = document.querySelector('.wood.three'),
    tileOne = document.querySelector('.tile.one'),
    tileTwo = document.querySelector('.tile.two'),
    tileThree = document.querySelector('.tile.three');

let isSelect = false,
    isDraw = false,
    isLine = false,
    isRectangle = false,
    isCircle = false,
    isShadow = false,
    isSun = false;

let pointer,
    points = [],
    mouse = {
        x: 0,
        y: 0
    },
    sun = {
        x: 50,
        y: 50
    };

let line_width = 1,
    line_style = "butt",
    line_color = "rgba(0, 0, 0, 1)",
    line_opacity = 1,
    fill_color = "rgba(0, 0, 0, 0)",
    fill_opacity = 0,
    text_size = 20 + "px",
    text_color = "rgba(0, 0, 0, 1)",
    text_opacity = 1;

let line_object,
    rectangle_object,
    circle_object;

window.tools.addBackgroundImage(canvas);

let history = new History(canvas);

canvas.on('mouse:down', evt => {
    if (!canvas.isDrawingMode) {
        isDraw = true;
        if (isLine) {
            pointer = canvas.getPointer(evt.e);
            points = [pointer.x, pointer.y, pointer.x, pointer.y];
            line_object = new fabric.Line(points, {
                strokeWidth: line_width,
                stroke: line_color,
                strokeLineCap: line_style,
                originX: 'center', originY: 'center',
                borderColor: 'red',
                cornerColor: 'rgba(255, 0, 0, .5)',
            });
        }
    }
});

```

```

        cornerStyle: 'circle',
        transparentCorners: false,
        selectable: true
    });
    canvas.add(line_object);
} else if (isRectangle) {
    pointer = canvas.getPointer(evt.e);
    mouse.x = pointer.x;
    mouse.y = pointer.y;
    rectangle_object = new fabric.Rect({
        left: mouse.x,
        top: mouse.y,
        strokeWidth: line_width,
        stroke: line_color,
        width: pointer.x - mouse.x,
        height: pointer.y - mouse.y,
        fill: fill_color,
        // originX: 'center', originY: 'center',
        borderColor: 'red',
        cornerColor: 'rgba(255, 0, 0, .5)',
        cornerStyle: 'circle',
        transparentCorners: false,
        selectable: true
    });
    canvas.add(rectangle_object);
} else if (isCircle) {
    pointer = canvas.getPointer(evt.e);
    mouse.x = pointer.x;
    mouse.y = pointer.y;
    circle_object = new fabric.Circle({
        left: pointer.x,
        top: pointer.y,
        radius: 1,
        strokeWidth: line_width,
        stroke: line_color,
        fill: fill_color,
        originX: 'center', originY: 'center',
        borderColor: 'red',
        cornerColor: 'rgba(255, 0, 0, .5)',
        cornerStyle: 'circle',
        transparentCorners: false,
        selectable: true
    });
    canvas.add(circle_object);
}
});
});

canvas.on('mouse:move', evt => {
    if (!isDraw) return;
    if (isLine) {
        pointer = canvas.getPointer(evt.e);
        line_object.set({
            x2: pointer.x,
            y2: pointer.y
        });
        canvas.renderAll();
    } else if (isRectangle) {
        pointer = canvas.getPointer(evt.e);
        rectangle_object.set('width', Math.abs(rectangle_object.left - pointer.x));
        rectangle_object.set('height', Math.abs(rectangle_object.top - pointer.y));
        canvas.renderAll();
    } else if (isCircle) {
        pointer = canvas.getPointer(evt.e);
        circle_object.set({ radius: Math.abs(mouse.x - pointer.x) });
        canvas.renderAll();
    }
});
});

```

```

canvas.on('mouse:up', () => {
  if (isLine || isRectangle || isCircle) {
    history.save();
  }
  isDraw = false;
  canvas.renderAll();
});

canvas.on('object:modified', () => { history.save(); });

canvas.on('object:scaling', evt => {
  if (evt.target.get('name') === 'pattern') {
    let obj = evt.target;
    let height = obj.height * obj.scaleY;
    let width = obj.width * obj.scaleX;
    obj.height = height;
    obj.width = width;
    obj.scaleX = 1;
    obj.scaleY = 1;
  } else if ((evt.target.get('name') === 'tree') || (evt.target.get('name') ===
'bush')) {
    window.tools.checkObjectIntersection(canvas, evt);
  }
});

canvas.on('object:moving', evt => {
  if (evt.target.get('name') === 'sun') {
    sun.x = evt.target.left;
    sun.y = evt.target.top;
    canvas.bringToFront(evt.target).discardActiveObject().renderAll();
    window.tools.updateShadow(isShadow, canvas, sun.y, sun.x);
  } else if ((evt.target.get('name') === 'tree') || (evt.target.get('name') ===
'bush')) {
    window.tools.checkObjectIntersection(canvas, evt);
  }
});

canvas.on('object:rotating', evt => {
  if ((evt.target.get('name') === 'tree') || (evt.target.get('name') === 'bush')) {
    window.tools.checkObjectIntersection(canvas, evt);
  }
});

canvas.on('object:moved', evt => {
  if (evt.target.get('name') === 'sun') {
    canvas.bringToFront(evt.target).discardActiveObject().renderAll();
  }
});

actionBtn.onclick = () => { window.util.showListElements(); };

listBtn.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  window.util.showListElements();
};

handBtn.onclick = () => {
  isSelect = true;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = true;
  canvas.isDrawingMode = false;
  window.util.showAllProperties();
};

```

```

delBtn.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = true;
  canvas.isDrawingMode = false;
  window.tools.deleteSelectedObjects(canvas);
  history.save();
};

brshBtn.onclick = () => {
  isSelect = false;
  isLine = true;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = true;
  window.util.showLineProperties();
};

lineBtn.onclick = () => {
  isSelect = false;
  isLine = true;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showLineProperties();
};

recBtn.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = true;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showLineAndFillProperties();
};

crclBtn.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = true;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showLineAndFillProperties();
};

txtBtn.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showTextProperties();
};

rectTexture.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showPatterns();
};

```

```

};

flower.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showFlowers();
};

waterFlower.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showWaterPlants();
};

bush.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showBushes();
};

tree.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showTrees();
};

pound.onclick = () => {
  isSelect = false;
  isLine = false;
  isRectangle = false;
  isCircle = false;
  canvas.selection = false;
  canvas.isDrawingMode = false;
  window.util.showPounds();
};

addShadowBtn.onclick = () => {
  isShadow = true;
  window.tools.updateShadow(isShadow, canvas, sun.y, sun.x);
  history.save();
};

removeShadowBtn.onclick = () => {
  isShadow = false;
  window.tools.updateShadow(isShadow, canvas, sun.y, sun.x);
  history.save();
};

addSunBtn.onclick = () => {
  isSun = true;
  window.tools.updateSun(isSun, canvas, sun.x, sun.y);
  history.save();
};

```

```

removeSunBtn.onclick = () => {
  isSun = false;
  window.tools.updateSun(isSun, canvas, sun.x, sun.y);
  history.save();
};

lineWidthRange.oninput = () => {
  line_width = parseInt(lineWidthRange.value);
  lineWidthOutput.value = line_width + "px";
  canvas.freeDrawingBrush.color = line_color;

  if (isSelect) window.tools.setObjectProperty(canvas, 'strokeWidth', line_width);
};

lineWidthRange.onmouseup = () => { if (isSelect) history.save(); };

lineColorInput.oninput = () => {
  let color = lineColorInput.value;
  line_opacity = lineOpacityRange.value;
  line_color = window.tools.setColor(color, line_opacity);
  canvas.freeDrawingBrush.color = line_color;

  if (isSelect) window.tools.setObjectProperty(canvas, 'stroke', line_color);
};

lineColorInput.onmouseup = () => { if (isSelect) history.save(); };

lineOpacityRange.oninput = () => {
  let color = lineColorInput.value;
  line_opacity = lineOpacityRange.value;
  line_color = window.tools.setColor(color, line_opacity);
  lineOpacityValue.value = lineOpacityRange.value * 100 + "%";
  canvas.freeDrawingBrush.color = line_color;

  if (isSelect) window.tools.setObjectProperty(canvas, 'stroke', line_color);
};

lineOpacityRange.onmouseup = () => { if (isSelect) history.save(); };

fillColorInput.oninput = () => {
  let color = fillColorInput.value;
  fill_opacity = fillOpacityRange.value;
  fill_color = window.tools.setColor(color, fill_opacity);

  if (isSelect) window.tools.setObjectProperty(canvas, 'fill', fill_color);
};

fillColorInput.onmouseup = () => { if (isSelect) history.save(); };

fillOpacityRange.oninput = () => {
  let color = fillColorInput.value;
  fill_opacity = fillOpacityRange.value;
  fill_color = window.tools.setColor(color, fill_opacity);
  fillOpacityValue.value = fillOpacityRange.value * 100 + "%";

  if (isSelect) window.tools.setObjectProperty(canvas, 'fill', fill_color);
};

fillOpacityRange.onmouseup = () => { if (isSelect) history.save(); };

textSizeRange.oninput = () => {
  text_size = parseInt(textSizeRange.value);
  textSizeOutput.value = text_size + "px";

  if (isSelect) window.tools.setObjectProperty(canvas, 'fontSize', text_size);
};

textSizeRange.onmouseup = () => { if (isSelect) history.save(); };

```

```

textColorInput.oninput = () => {
    let color = textColorInput.value;
    text_opacity = parseInt(textOpacityInput.value, 10) || 1;
    text_color = window.tools.setColor(color, text_opacity);
    textOpacityValue.value = textOpacityInput.value * 100 + "%";
    textInput.setAttribute("style", "color: " + text_color);

    if (isSelect) window.tools.setObjectProperty(canvas, 'fill', text_color);
};

textColorInput.onmouseup = () => { if (isSelect) history.save(); };

textOpacityInput.oninput = () => {
    let color = textColorInput.value;
    text_opacity = textOpacityInput.value;
    text_color = window.tools.setColor(color, text_opacity);
    textOpacityValue.value = textOpacityInput.value * 100 + "%";
    textInput.setAttribute("style", "color: " + text_color);

    if (isSelect) window.tools.setObjectProperty(canvas, 'fill', text_color);
};

textOpacityInput.onmouseup = () => { if (isSelect) history.save(); };

txtSubmitBtn.onclick = () => {
    if (textInput.value === "") {
        alert("Поле пустое!");
        textInput.focus();
    } else {
        canvas.selection = false;
        canvas.isDrawingMode = false;
        window.tools.addText(canvas, textInput.value, text_size, text_color);
        textInput.value = "";
        history.save();
    }
};

txtEraseBtn.onclick = () => { textInput.value = ""; };

grassTexture.onclick = () => { window.util.showGrassPatterns(); };
groundTexture.onclick = () => { window.util.showGroundPatterns(); };
woodTexture.onclick = () => { window.util.showWoodPatterns(); };
tileTexture.onclick = () => { window.util.showTilePatterns(); };

grassOne.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'grass-1.jpg', canvas);
    history.save();
};
grassTwo.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'grass-2.jpg', canvas);
    history.save();
};
grassThree.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'grass-3.jpg', canvas);
    history.save();
};
groundOne.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'ground-1.jpg', canvas);
    history.save();
};
groundTwo.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'ground-2.jpg', canvas);
    history.save();
};
groundThree.onclick = () => {
    window.tools.addPatternObject(TEXTURE_PATH + 'ground-3.jpg', canvas);
    history.save();
};
woodOne.onclick = () => {

```

```

        window.tools.addPatternObject(TEXTURE_PATH + 'wood-1.jpg', canvas);
        history.save();
    };
    woodTwo.onclick = () => {
        window.tools.addPatternObject(TEXTURE_PATH + 'wood-2.jpg', canvas);
        history.save();
    };
    woodThree.onclick = () => {
        window.tools.addPatternObject(TEXTURE_PATH + 'wood-3.jpg', canvas);
        history.save();
    };
    tileOne.onclick = () => {
        window.tools.addPatternObject(TEXTURE_PATH + 'tile-1.jpg', canvas);
        history.save();
    };
    tileTwo.onclick = () => {
        window.tools.addPatternObject(TEXTURE_PATH + 'tile-2.jpg', canvas);
        history.save();
    };
    tileThree.onclick = () => {
        window.tools.addPatternObject(TEXTURE_PATH + 'tile-3.jpg', canvas);
        history.save();
    };

    undoBtn.onclick = () => { history.undo(); };
    redoBtn.onclick = () => { history.redo(); };

    unselectBtn.onclick = () => { canvas.discardActiveObject().renderAll(); };

    toBackBtn.onclick = () => {
        window.tools.toBack(canvas);
        history.save();
    };
    toFrontBtn.onclick = () => {
        window.tools.toFront(canvas);
        history.save();
    };

    clrBtn.onclick = function () {
        canvas.clear();
        window.tools.addBackGroundImage(canvas);
        window.tools.updateSun(canvas);
        history.save();
    };

    dwnldImg.onclick = () => { window.tools.downloadImg(canvas); };

    itemList.onclick = evt => {
        isSelect = false;
        isLine = false;
        isRectangle = false;
        isCircle = false;
        canvas.selection = false;
        canvas.isDrawingMode = false;
        let target = evt.target || window.event.srcElement;
        let object_type;
        object_type =
window.util.checkObjectGroup(target.parentElement.parentElement.parentElement.className);
        if (target.tagName === 'P') {
            let json = {
                "type": object_type,
                "name": target.innerHTML
            };
            let data = JSON.stringify(json);
            window.backend.searchObject(data, window.util.isError);
        } else if (target.tagName === 'IMG') {
            window.tools.addSVGIcon(
                canvas,
                target.getAttribute('src'),

```

```
        object_type
    );
    }
};

closeCardBtn.onclick = () => { window.util.closeCard(); };

document.onkeydown = (evt) => {
    if (evt.which === Z_KEY && evt.ctrlKey) {
        history.undo();
    } else if (evt.which === Y_KEY && evt.ctrlKey) {
        history.redo();
    } else if (evt.which === C_KEY && evt.ctrlKey) {
        window.tools.cloneObject(canvas);
        history.save();
    } else if (evt.which === BACKSPACE_KEY) {
        window.tools.deleteSelectedObjects(canvas);
        history.save();
    } else if (evt.which === ESC_KEY) {
        canvas.discardActiveObject().renderAll();
    }
};
})();
```

# ПРИЛОЖЕНИЕ И

## ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-DATA

### Листинг И.1 – Исходный код файла “search-data.php”

```
<?php
header("Content-Type: application/json");
$json = json_decode(stripslashes(file_get_contents("php://input")));

$user = 'root';
$password = 'root';
$db = 'landscape_editor';
$host = 'localhost';
$port = 3306;
$link = mysqli_connect($host, $user, $password, $db) or die ("Ошибка: " .
mysqli_error($link));

if ($json->climate === 'Любой') {
    $query_flower = "SELECT * FROM flower";
    $query_bush = "SELECT * FROM bush";
    $query_tree = "SELECT * FROM tree";
    $query_water_plant = "SELECT * FROM water_plant";
    if ($json->light === 'Любой') {
        if ($json->watering === 'Любой') {
            if ($json->endurance === 'Любой') {
                if ($json->temperature_max == '') {
                    if ($json->air_humidity_max == '') {
                        } else {
                            $query_flower .= " WHERE air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_bush .= " WHERE air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_tree .= " WHERE air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_water_plant .= " WHERE air_humidity_max < '{ $json-
>air_humidity_max}'";
                        }
                    } else {
                        $query_flower .= " WHERE temperature_max < '{ $json->temperature_max}'";
                        $query_bush .= " WHERE temperature_max < '{ $json->temperature_max}'";
                        $query_tree .= " WHERE temperature_max < '{ $json->temperature_max}'";
                        $query_water_plant .= " WHERE temperature_max < '{ $json-
>temperature_max}'";
                    if ($json->air_humidity_max == '') {
                        } else {
                            $query_flower .= " AND air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_bush .= " AND air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_tree .= " AND air_humidity_max < '{ $json-
>air_humidity_max}'";
                            $query_water_plant .= " AND air_humidity_max < '{ $json-
>air_humidity_max}'";
                        }
                    }
                } else {
                    $query_flower .= " WHERE endurance = '{ $json->endurance}'";
                    $query_bush .= " WHERE endurance = '{ $json->endurance}'";
                    $query_tree .= " WHERE endurance = '{ $json->endurance}'";
                    $query_water_plant .= " WHERE endurance = '{ $json->endurance}'";
                    if ($json->temperature_max == '') {
                        if ($json->air_humidity_max == '') {
                            } else {
                                $query_flower .= " AND air_humidity_max < '{ $json-
>air_humidity_max}'";
```

```

$query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
$query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
$query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
    }
    } else {
        $query_flower .= " AND temperature_max < '{$json->temperature_max}''";
        $query_bush .= " AND temperature_max < '{$json->temperature_max}''";
        $query_tree .= " AND temperature_max < '{$json->temperature_max}''";
        $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}''";
        if ($json->air_humidity_max == '') {
        } else {
            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
        }
    }
    } else {
        $query_flower .= " WHERE watering = '{$json->watering}''";
        $query_bush .= " WHERE watering = '{$json->watering}''";
        $query_tree .= " WHERE watering = '{$json->watering}''";
        if ($json->endurance === 'любой') {
            if ($json->temperature_max == '') {
                if ($json->air_humidity_max == '') {
                } else {
                    $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_water_plant .= " WHERE air_humidity_max < '{$json-
>air_humidity_max}''";
                }
            } else {
                $query_flower .= " AND temperature_max < '{$json->temperature_max}''";
                $query_bush .= " AND temperature_max < '{$json->temperature_max}''";
                $query_tree .= " AND temperature_max < '{$json->temperature_max}''";
                $query_water_plant .= " WHERE temperature_max < '{$json-
>temperature_max}''";
                if ($json->air_humidity_max == '') {
                } else {
                    $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                    $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
                }
            }
        } else {
            $query_flower .= " AND endurance = '{$json->endurance}''";
            $query_bush .= " AND endurance = '{$json->endurance}''";
            $query_tree .= " AND endurance = '{$json->endurance}''";
            $query_water_plant .= " WHERE endurance = '{$json->endurance}''";
            if ($json->temperature_max == '') {
                if ($json->air_humidity_max == '') {
                } else {

```

```

$query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
$query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
$query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
$query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
    }
  } else {
    $query_flower .= " AND temperature_max < '{$json->temperature_max}''";
    $query_bush .= " AND temperature_max < '{$json->temperature_max}''";
    $query_tree .= " AND temperature_max < '{$json->temperature_max}''";
    $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}''";
    if ($json->air_humidity_max == '') {
    } else {
      $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
      $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
      $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
      $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
    }
  }
}
} else {
  $query_flower .= " WHERE light = '{$json->light}''";
  $query_bush .= " WHERE light = '{$json->light}''";
  $query_tree .= " WHERE light = '{$json->light}''";
  $query_water_plant .= " WHERE light = '{$json->light}''";
  if ($json->watering === 'Любой') {
    if ($json->endurance === 'Любой') {
      if ($json->temperature_max == '') {
        if ($json->air_humidity_max == '') {
          $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
          $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
          $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
          $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
        } else {
          $query_flower .= " AND temperature_max < '{$json->temperature_max}''";
          $query_bush .= " AND temperature_max < '{$json->temperature_max}''";
          $query_tree .= " AND temperature_max < '{$json->temperature_max}''";
          $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}''";
          if ($json->air_humidity_max == '') {
          } else {
            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}''";
          }
        }
      } else {
        $query_flower .= " AND endurance = '{$json->endurance}''";
        $query_bush .= " AND endurance = '{$json->endurance}''";

```

```

$query_tree .= " AND endurance = '{$json->endurance}'";
$query_water_plant .= " AND endurance = '{$json->endurance}'";
if ($json->temperature_max == '') {
    if ($json->air_humidity_max == '') {
        } else {
            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";

            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        }
    } else {
        $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
        $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
        $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
        $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
        if ($json->air_humidity_max == '') {
            } else {
                $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            }
        }
    } else {
        $query_flower .= " AND watering = '{$json->watering}'";
        $query_bush .= " AND watering = '{$json->watering}'";
        $query_tree .= " AND watering = '{$json->watering}'";
        if ($json->endurance === 'любой') {
            if ($json->temperature_max == '') {
                if ($json->air_humidity_max == '') {
                    } else {
                        $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                    }
                } else {
                    $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
                    $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
                    $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
                    $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
                    if ($json->air_humidity_max == '') {
                        } else {
                            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        }
                    }
                }
            }
        }
    }
}

```

```

    } else {
        $query_flower .= " AND endurance = '{$json->endurance}'";
        $query_bush .= " AND endurance = '{$json->endurance}'";
        $query_tree .= " AND endurance = '{$json->endurance}'";
        $query_water_plant .= " AND endurance = '{$json->endurance}'";
        if ($json->temperature_max == '') {
            if ($json->air_humidity_max == '') {
                } else {
                    $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                    $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                    $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                    $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                }
            } else {
                $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
                $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
                $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
                $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
                if ($json->air_humidity_max == '') {
                    } else {
                        $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                    }
                }
            }
        }
    }
} else {
    $query_flower = "SELECT * FROM flower WHERE climate = '{$json->climate}'";
    $query_bush = "SELECT * FROM bush WHERE climate = '{$json->climate}'";
    $query_tree = "SELECT * FROM tree WHERE climate = '{$json->climate}'";
    $query_water_plant = "SELECT * FROM water_plant WHERE climate = '{$json->climate}'";
    if ($json->light === 'Любой') {
        if ($json->watering === 'Любой') {
            if ($json->endurance === 'Любой') {
                if ($json->temperature_max == '') {
                    if ($json->air_humidity_max == '') {
                        } else {
                            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                        }
                    } else {
                        $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
                        $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
                        $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
                        $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
                        if ($json->air_humidity_max == '') {
                            } else {
                                $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";

```

```

$query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
$query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
$query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
    }
} else {
    $query_flower .= " AND endurance = '{$json->endurance}";
    $query_bush .= " AND endurance = '{$json->endurance}";
    $query_tree .= " AND endurance = '{$json->endurance}";
    $query_water_plant .= " AND endurance = '{$json->endurance}";
    if ($json->temperature_max == '') {
        if ($json->air_humidity_max == '') {
            } else {
                $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
            }
        } else {
            $query_flower .= " AND temperature_max < '{$json->temperature_max}";
            $query_bush .= " AND temperature_max < '{$json->temperature_max}";
            $query_tree .= " AND temperature_max < '{$json->temperature_max}";
            $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}";
            if ($json->air_humidity_max == '') {
                } else {
                    $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                    $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                    $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                    $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                }
            }
        } else {
            $query_flower .= " AND watering = '{$json->watering}";
            $query_bush .= " AND watering = '{$json->watering}";
            $query_tree .= " AND watering = '{$json->watering}";
            if ($json->endurance === 'любой') {
                if ($json->temperature_max == '') {
                    if ($json->air_humidity_max == '') {
                        } else {
                            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}";
                        }
                    } else {
                        $query_flower .= " AND temperature_max < '{$json->temperature_max}";
                        $query_bush .= " AND temperature_max < '{$json->temperature_max}";
                        $query_tree .= " AND temperature_max < '{$json->temperature_max}";
                        $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}";
                    }
                }
            }
        }
    }
}

```

```

        if ($json->air_humidity_max == '') {
        } else {
            $query_flower .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
            $query_bush .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
            $query_tree .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
            $query_water_plant .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
        }
    } else {
        $query_flower .= " AND endurance = '{$json->endurance}'";
        $query_bush .= " AND endurance = '{$json->endurance}'";
        $query_tree .= " AND endurance = '{$json->endurance}'";
        $query_water_plant .= " AND endurance = '{$json->endurance}'";
        if ($json->temperature_max == '') {
            if ($json->air_humidity_max == '') {
            } else {
                $query_flower .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_bush .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_tree .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_water_plant .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
            }
        } else {
            $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
            $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
            $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
            $query_water_plant .= " AND temperature_max < '{$json->
>temperature_max}'";
            if ($json->air_humidity_max == '') {
            } else {
                $query_flower .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_bush .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_tree .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                $query_water_plant .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
            }
        }
    }
} else {
    $query_flower .= " AND light = '{$json->light}'";
    $query_bush .= " AND light = '{$json->light}'";
    $query_tree .= " AND light = '{$json->light}'";
    $query_water_plant .= " AND light = '{$json->light}'";
    if ($json->watering === 'любой') {
        if ($json->endurance === 'любой') {
            if ($json->temperature_max == '') {
                if ($json->air_humidity_max == '') {
                } else {
                    $query_flower .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                    $query_bush .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                    $query_tree .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                    $query_water_plant .= " AND air_humidity_max < '{$json->
>air_humidity_max}'";
                }
            } else {

```

```

$query_flower .= " AND temperature_max < '{$json->temperature_max}'";
$query_bush .= " AND temperature_max < '{$json->temperature_max}'";
$query_tree .= " AND temperature_max < '{$json->temperature_max}'";
$query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
    if ($json->air_humidity_max == '') {
    } else {
        $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
    }
} else {
    $query_flower .= " AND endurance = '{$json->endurance}'";
    $query_bush .= " AND endurance = '{$json->endurance}'";
    $query_tree .= " AND endurance = '{$json->endurance}'";
    $query_water_plant .= " AND endurance = '{$json->endurance}'";
    if ($json->temperature_max == '') {
        if ($json->air_humidity_max == '') {
        } else {
            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        }
    } else {
        $query_flower .= " AND temperature_max < '{$json->temperature_max}'";
        $query_bush .= " AND temperature_max < '{$json->temperature_max}'";
        $query_tree .= " AND temperature_max < '{$json->temperature_max}'";
        $query_water_plant .= " AND temperature_max < '{$json-
>temperature_max}'";
        if ($json->air_humidity_max == '') {
        } else {
            $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            $query_water_plant .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
        }
    }
}
} else {
    $query_flower .= " AND watering = '{$json->watering}'";
    $query_bush .= " AND watering = '{$json->watering}'";
    $query_tree .= " AND watering = '{$json->watering}'";
    if ($json->endurance === 'любой') {
        if ($json->temperature_max == '') {
            if ($json->air_humidity_max == '') {
            } else {
                $query_flower .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                $query_bush .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
                $query_tree .= " AND air_humidity_max < '{$json-
>air_humidity_max}'";
            }
        }
    }
}

```



```

$objects = array(
    "flower" => array(),
    "water_plant" => array(),
    "bush" => array(),
    "tree" => array(),
    "pound" => array()
);

if ($result_flower) {
    $rows = mysqli_num_rows($result_flower);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result_flower);
        $objects["flower"][$i] = $row;
    }
    mysqli_free_result($result_flower);
}
if ($result_bush) {
    $rows = mysqli_num_rows($result_bush);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result_bush);
        $objects["bush"][$i] = $row;
    }
    mysqli_free_result($result_bush);
}
if ($result_tree) {
    $rows = mysqli_num_rows($result_tree);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result_tree);
        $objects["tree"][$i] = $row;
    }
    mysqli_free_result($result_tree);
}
if ($result_water_plant) {
    $rows = mysqli_num_rows($result_water_plant);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result_water_plant);
        $objects["water_plant"][$i] = $row;
    }
    mysqli_free_result($result_water_plant);
}
if ($result_pound) {
    $rows = mysqli_num_rows($result_pound);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result_pound);
        $objects["pound"][$i] = $row;
    }
    mysqli_free_result($result_pound);
}
mysqli_close($link);

$objects = json_encode($objects, JSON_UNESCAPED_UNICODE);
echo $objects;

```

## ПРИЛОЖЕНИЕ К

### ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-OBJECT

#### Листинг К.1 – Исходный код файла “search-object.php”

```
<?php

header("Content-Type: application/json");
$json = json_decode(stripslashes(file_get_contents("php://input")));

$user = 'root';
$password = 'root';
$db = 'landscape_editor';
$host = 'localhost';
$port = 3306;
$link = mysqli_connect($host, $user, $password, $db) or die ("Ошибка: " .
mysqli_error($link));

$object = array();

if ($json->type === 'flower') {
    $query = "SELECT * FROM flower WHERE name = '{$json->name}'";
    array_push($object, 'flower');
} else if ($json->type === 'bush') {
    $query = "SELECT * FROM bush WHERE name = '{$json->name}'";
    array_push($object, 'bush');
} else if ($json->type === 'tree') {
    $query = "SELECT * FROM tree WHERE name = '{$json->name}'";
    array_push($object, 'tree');
} else if ($json->type === 'water_plant') {
    $query = "SELECT * FROM water_plant WHERE name = '{$json->name}'";
    array_push($object, 'water_plant');
} else if ($json->type === 'pound') {
    $query = "SELECT * FROM pound WHERE name = '{$json->name}'";
    array_push($object, 'pound');
}

$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));

if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0; $i < $rows; ++$i) {
        $row = mysqli_fetch_row($result);
        array_push($object, $row);
    }
    mysqli_free_result($result);
}

mysqli_close($link);

$object = json_encode($object, JSON_UNESCAPED_UNICODE);
echo $object;
```

## ПРИЛОЖЕНИЕ Л

### ИСХОДНЫЙ КОД PHP-ФАЙЛА SEARCH-SELECT-OPTIONS

Листинг Л.1 – Исходный код файла “search-select-options.php”

```
<?php

$user = 'root';
$password = 'root';
$db = 'landscape_editor';
$host = 'localhost';
$port = 3306;
$link = mysqli_connect($host, $user, $password, $db) or die ("Ошибка: " .
mysqli_error($link));
$climate_query = array();
$light_query = array();
$watering_query = array();
$endurance_query = array();
$climate_result = array(
    'climate_flower' => array(),
    'climate_bush' => array(),
    'climate_tree' => array(),
    'climate_water_plant' => array(),
    'result' => array()
);
$light_result = array(
    'light_flower' => array(),
    'light_bush' => array(),
    'light_tree' => array(),
    'light_water_plant' => array(),
    'result' => array()
);
$watering_result = array(
    'watering_flower' => array(),
    'watering_bush' => array(),
    'watering_tree' => array(),
    'result' => array()
);
$endurance_result = array(
    'endurance_flower' => array(),
    'endurance_bush' => array(),
    'endurance_tree' => array(),
    'endurance_water_plant' => array(),
    'result' => array()
);
$json = array(
    'climate' => array(),
    'light' => array(),
    'watering' => array(),
    'endurance' => array()
);

if ($link) {
    $query = "SELECT climate FROM flower GROUP BY 1 HAVING COUNT(*) > 1";
    $result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
    if ($result) {
        $rows = mysqli_num_rows($result);
        for ($i = 0 ; $i < $rows ; ++$i) {
            $row = mysqli_fetch_row($result);
            $climate_query = $row;
            array_push($climate_result['climate_flower'], $climate_query[0]);
        }
    }
    mysqli_free_result($result);
    $query = "SELECT climate FROM bush GROUP BY 1 HAVING COUNT(*) > 1";
    $result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
```

```

if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $climate_query = $row;
        array_push($climate_result['climate_bush'], $climate_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT climate FROM tree GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $climate_query = $row;
        array_push($climate_result['climate_tree'], $climate_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT climate FROM water_plant GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $climate_query = $row;
        array_push($climate_result['climate_water_plant'], $climate_query[0]);
    }
}
mysqli_free_result($result);
foreach ($climate_result['climate_flower'] as $k => $v) {
    if (!in_array($v, $climate_result['result'])) {
        $climate_result['result'][$k] = $v;
    }
}
foreach ($climate_result['climate_bush'] as $k => $v) {
    if (!in_array($v, $climate_result['result'])) {
        $climate_result['result'][$k] = $v;
    }
}
foreach ($climate_result['climate_tree'] as $k => $v) {
    if (!in_array($v, $climate_result['result'])) {
        $climate_result['result'][$k] = $v;
    }
}
foreach ($climate_result['climate_water_plant'] as $k => $v) {
    if (!in_array($v, $climate_result['result'])) {
        $climate_result['result'][$k] = $v;
    }
}
$query = "SELECT light FROM flower GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $light_query = $row;
        array_push($light_result['light_flower'], $light_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT light FROM bush GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $light_query = $row;
    }
}

```

```

        array_push($light_result['light_bush'], $light_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT light FROM tree GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $light_query = $row;
        array_push($light_result['light_tree'], $light_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT light FROM water_plant GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $light_query = $row;
        array_push($light_result['light_water_plant'], $light_query[0]);
    }
}
mysqli_free_result($result);
foreach ($light_result['light_flower'] as $k => $v) {
    if (!in_array($v, $light_result['result'])) {
        $light_result['result'][$k] = $v;
    }
}
foreach ($light_result['light_bush'] as $k => $v) {
    if (!in_array($v, $light_result['result'])) {
        $light_result['result'][$k] = $v;
    }
}
foreach ($light_result['light_tree'] as $k => $v) {
    if (!in_array($v, $light_result['result'])) {
        $light_result['result'][$k] = $v;
    }
}
foreach ($light_result['light_water_plant'] as $k => $v) {
    if (!in_array($v, $light_result['result'])) {
        $light_result['result'][$k] = $v;
    }
}

$query = "SELECT watering FROM flower GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $watering_query = $row;
        array_push($watering_result['watering_flower'], $watering_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT watering FROM bush GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $watering_query = $row;
        array_push($watering_result['watering_bush'], $watering_query[0]);
    }
}
mysqli_free_result($result);

```

```

$query = "SELECT watering FROM tree GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $watering_query = $row;
        array_push($watering_result['watering_tree'], $watering_query[0]);
    }
}
mysqli_free_result($result);
foreach ($watering_result['watering_flower'] as $k => $v) {
    if (!in_array($v, $watering_result['result'])) {
        $watering_result['result'][$k] = $v;
    }
}
foreach ($watering_result['watering_bush'] as $k => $v) {
    if (!in_array($v, $watering_result['result'])) {
        $watering_result['result'][$k] = $v;
    }
}
foreach ($watering_result['watering_tree'] as $k => $v) {
    if (!in_array($v, $watering_result['result'])) {
        $watering_result['result'][$k] = $v;
    }
}
}
$query = "SELECT endurance FROM flower GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $endurance_query = $row;
        array_push($endurance_result['endurance_flower'], $endurance_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT endurance FROM bush GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $endurance_query = $row;
        array_push($endurance_result['endurance_bush'], $endurance_query[0]);
    }
}
mysqli_free_result($result);
$query = "SELECT endurance FROM tree GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $endurance_query = $row;
        array_push($endurance_result['endurance_tree'], $endurance_query[0]);
    }
}
}
mysqli_free_result($result);
$query = "SELECT endurance FROM water_plant GROUP BY 1 HAVING COUNT(*) > 1";
$result = mysqli_query($link, $query) or die ("Ошибка " . mysqli_error($link));
if ($result) {
    $rows = mysqli_num_rows($result);
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        $endurance_query = $row;
        array_push($endurance_result['endurance_water_plant'], $endurance_query[0]);
    }
}
}

```

```

mysqli_free_result($result);
foreach ($endurance_result['endurance_flower'] as $k => $v) {
    if (!in_array($v, $endurance_result['result'])) {
        $endurance_result['result'][$k] = $v;
    }
}
foreach ($endurance_result['endurance_bush'] as $k => $v) {
    if (!in_array($v, $endurance_result['result'])) {
        $endurance_result['result'][$k] = $v;
    }
}
foreach ($endurance_result['endurance_tree'] as $k => $v) {
    if (!in_array($v, $endurance_result['result'])) {
        $endurance_result['result'][$k] = $v;
    }
}
foreach ($endurance_result['endurance_water_plant'] as $k => $v) {
    if (!in_array($v, $endurance_result['result'])) {
        $endurance_result['result'][$k] = $v;
    }
}
}
mysqli_close($link);
$json['climate'] = $climate_result['result'];
$json['light'] = $light_result['result'];
$json['watering'] = $watering_result['result'];
$json['endurance'] = $endurance_result['result'];

$json = json_encode($json, JSON_UNESCAPED_UNICODE);
echo $json;

```