Министерство образования Омской области

бюджетное профессиональное образовательное учреждение Омской области

«ОМСКИЙ АВТОТРАНСПОРТНЫЙ КОЛЛЕДЖ»

(БПОУ «Омский АТК»)

**Курсовой проект**

Тема**: *Проектирование и разработка сайта для фрилансеров***

|  |  |
| --- | --- |
| Студент | **Н.В. Глотов** |
| Руководитель курсового  Проектирования | **Д.Е. Таран** |
|  |  |
| Заведующий отделением | **И.В. Субботина** |

Омск 2025

СОДЕРЖАНИЕ

[Введение 4](#_Toc137633792)

[1 Исследовательский раздел 6](#_Toc137633793)

[1.1 Обоснование разработки сайта, целевая аудитория 6](#_Toc137633794)

[1.2Обзор существующих аналогов 8](#_Toc137633795)

[1.3 Техническое задание на проект 11](#_Toc137633796)

[1.4 Технические и программные средства для разработки продукта 22](#_Toc137633797)

[2 Техногологический раздел 26](#_Toc137633798)

[2.1 Проектирование фриланс сайта 26](#_Toc137633800)

[2.2 Технология разработки веб-приложения 30](#_Toc137633801)

[2.3 Система контроля версий 34](#_Toc137633802)

[3 Организационный раздел 40](#_Toc137633804)

[3.1 Описание назначения веб-сайта 40](#_Toc137633805)

[3.2 Руководство пользователя 40](#_Toc137633806)

[Заключение 45](#_Toc137633807)

[Список используемых источников 46](#_Toc137633808)

# ВВЕДЕНИЕ

В последние годы всё больше специалистов в сфере информационных технологий переходят на фриланс. Многие уходят из офисов, чтобы работать над собственными проектами и самостоятельно выбирать себе заказчиков. Сейчас фриланс — это не только веб-дизайн или программирование: сюда входят переводы, копирайтинг, маркетинг и множество других направлений. Поэтому неудивительно, что появляются самые разные онлайн-площадки, которые помогают таким специалистам и их потенциальным заказчикам найти друг друга.

Однако, как показывает практика, не все существующие сайты для фрилансеров достаточно удобны и безопасны. Некоторые платформы «грешат» сложным интерфейсом, другие — высокими комиссиями или отсутствием должной защиты сделок. Именно поэтому я выбрал тему «Проектирование и разработка сайта для фрилансеров» в качестве курсового проекта: мне захотелось разобраться, как создать современный, функциональный и комфортный сервис, который будет отвечать запросам как исполнителей, так и заказчиков.

Цель данной работы — спроектировать и разработать сайт, который позволит фрилансерам и их клиентам эффективно сотрудничать, гарантировать безопасность сделок и упростить весь процесс — от размещения проектов и резюме до обсуждения условий и оплаты.

В ходе выполнения проекта планируется решить несколько основных задач:

1. Исследовать предметную область;
2. Разработать структуру системы;
3. Разработать техническое задание;
4. Разработать макет пользовательского интерфейса;
5. Разработать серверную часть;
6. Разработать пользовательский интерфейс;
7. Реализовать автоматизацию в работе базы данных;

Практическая ценность такого сайта состоит в том, что его запуск может упростить поиск квалифицированных специалистов для самых разных задач, а фрилансерам — помочь быстрее находить клиентов и выстраивать долгосрочные партнёрские отношения. Кроме того, подобный проект будет стимулировать конкуренцию на рынке, а значит, повышать качество оказываемых услуг.

Структура моей курсовой работы включает в себя как теоретический обзор существующих площадок и принципов проектирования, так и практические аспекты создания сайта: рассмотрение архитектуры базы данных, проработка клиент-серверного взаимодействия, дизайн и тестирование.

# 1 **ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**

## 1.1 Обоснование разработки сайта, целевая аудитория

## Развитие фриланс-сегмента в последние годы продолжает набирать обороты, и всё больше людей выбирают свободный график работы, сотрудничая с различными заказчиками в дистанционном формате. Этот формат позволяет исполнителям самостоятельно регулировать рабочую нагрузку, а заказчикам — подбирать специалистов из любой точки мира. Однако, несмотря на большой выбор площадок для фрилансеров, многие из них либо не обладают интуитивно понятным интерфейсом, либо не дают достаточных гарантий безопасности сделок, либо взимают высокие комиссии. Также некоторые платформы предлагают узконаправленный функционал, который охватывает не все аспекты взаимодействия (например, сложность поиска и фильтрации проектов, отсутствие качественной системы отзывов и т. д.).

## Таким образом, необходимость в разработке нового сайта для фрилансеров обусловлена несколькими факторами. Во-первых, это стремление предоставить пользователям (как исполнителям, так и заказчикам) более удобный сервис с понятным интерфейсом и дополнительными возможностями для самопрезентации и поиска проектов. Во-вторых, актуальным остаётся вопрос безопасности: система должна обеспечивать не только надёжные способы оплаты, но и чёткую модерацию контента, а также механизмы разрешения споров. В-третьих, современный рынок требует комплексного подхода к продвижению личного бренда: наличие портфолио, отзывов, рейтинга, а также интеграции с социальными сетями может существенно повысить шансы фрилансеров на получение новых заказов.

## При разработке такого сайта важно чётко определить целевую аудиторию, чтобы ориентироваться на её потребности и задачи. В данном случае основными группами пользователей являются:

## 1. Исполнители (фрилансеры):

## - Начинающие специалисты (18-25 лет), которые хотят найти первых клиентов, представить своё портфолио и набрать рейтинг.

## - Опытные профессионалы (25-40 лет), которые стремятся расширить клиентскую базу и выйти на более высокий уровень дохода, а также укрепить свою репутацию.

## - Узкопрофильные эксперты (возраст варьируется 25+), которые ищут проекты в своей нише и нуждаются в удобном инструменте для продвижения.

## По половой принадлежности фрилансеры могут быть как мужчинами так и женщинами.

## 2. Заказчики (компании и частные лица):

## - Малый и средний бизнес (25-45 лет), который часто нуждается в разовых проектах (создание сайта, реклама, дизайн бренда) и предпочитает не держать подобных специалистов в штате на постоянной основе.

## - Стартапы (20-35 лет), которым важно быстро найти необходимых экспертов и оперативно выполнить поставленные задачи.

## - Индивидуальные предприниматели или частные лица (возраст может быть самым разным от 18 до 40 лет), которые ищут фрилансеров для разовых заказов (написание статей, настройка контекстной рекламы, переводы и т. д.).

## 3. Администрация площадки:

## - Команда (20-35 лет), обеспечивающая разработку, поддержку и модерирование ресурса. Её интересы связаны с поддержанием стабильной работы сайта, защитой пользователей от мошенничества и улучшением функционала на основе обратной связи.

## Учитывая столь разнообразную аудиторию, одной из ключевых целей разработки сайта является обеспечение удобства и эффективности взаимодействия между всеми группами пользователей. Кроме того, проект предполагает решение актуальных проблем существующих платформ: снижение риска мошенничества, упрощение механизма поиска и фильтрации проектов, прозрачная система рейтингов и отзывов. За счёт комплексного подхода к проектированию сайта планируется добиться высокой лояльности пользователей и повысить конкурентоспособность данного ресурса на рынке фриланс-услуг.

## Таким образом, обоснование разработки нового сайта для фрилансеров базируется на сочетании растущего спроса на дистанционные услуги, потребности в безопасных и удобных онлайн-сервисах, а также необходимости создания функционала, ориентированного на нужды разных категорий пользователей. В дальнейшем исследовании будут детально рассмотрены способы реализации этих идей, а также раскрыта структура и технологический стек, выбранные для разработки проекта.

## 1.2Обзор существующих аналогов

Проведя поиск и проанализировав уже существующие игровые сайта, были выделены следующие системы:

* FL.ru;

Одна из самых известных русскоязычных бирж, позволяющая работать в широком спектре направлений (IT, дизайн, копирайтинг и т. д.).

* Kwork;

Сервис, ориентированный на формат «микроуслуг»; исполнители предлагают готовые пакеты услуг — «кворки».

* freelance.habr.com;

Freelansim — это биржа, связанная с популярным техническим ресурсом «Хабр», что делает её особенно популярной среди разработчиков, дизайнеров и других специалистов в сфере IT.

На рис. 1.1 изображена главная страница сайта «FL.ru».

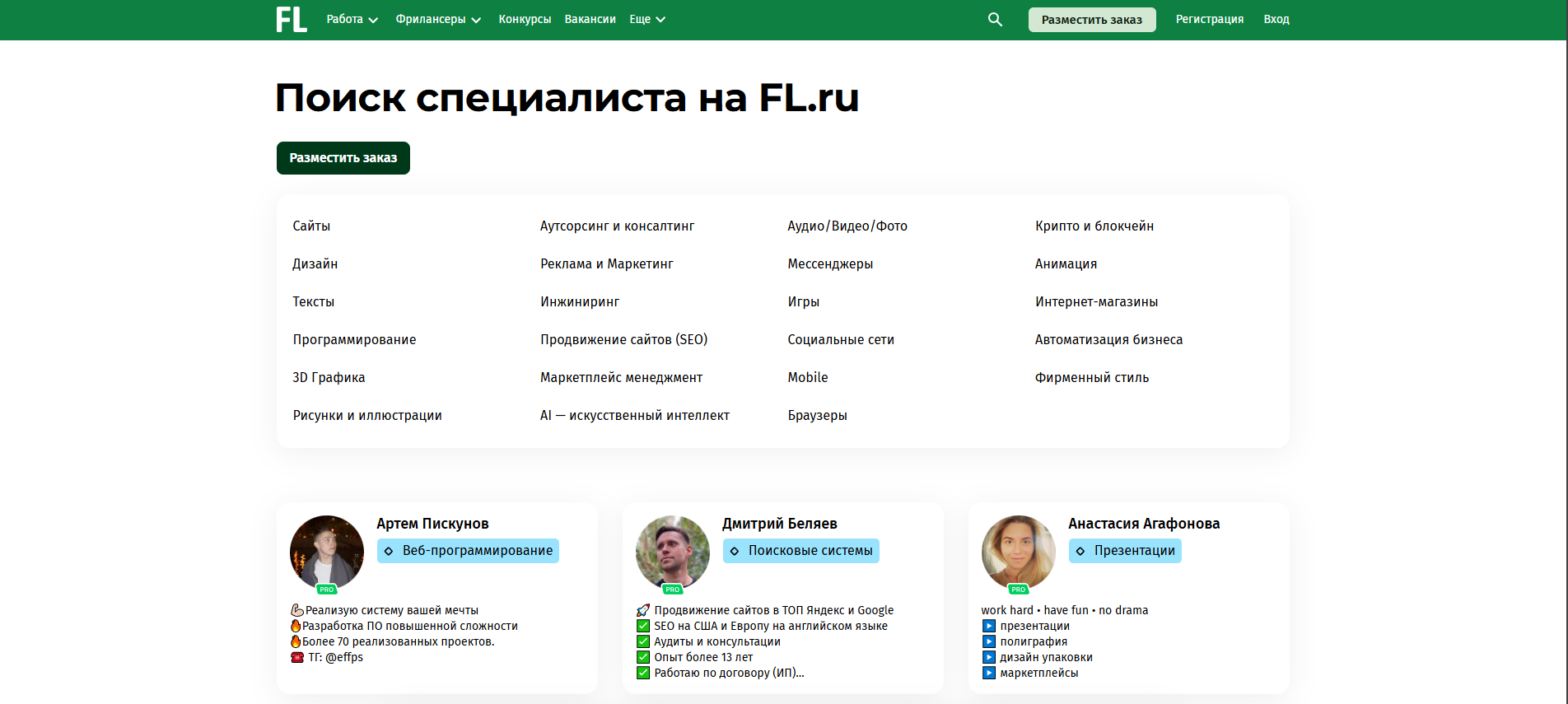


Рисунок 1.1 – Главная страница сайта «FL.ru».

 **Особенности**:

* Одна из старейших и наиболее известных бирж фриланса в русскоязычном сегменте.
* Широкий спектр специализаций: IT-разработка, дизайн, копирайтинг, маркетинг и т. д.
* Предусмотрена платная подписка Pro, которая даёт исполнителям повышенный лимит откликов и возможность выгружать больше проектов в портфолио.

 **Недостатки**:

* Высокая конкуренция: новичкам бывает сложно найти первых заказчиков среди большого количества откликов.
* Нередко пользователи жалуются на высокий уровень спама и «мёртвые» заказы (когда заказчик перестаёт отвечать).
* Механизмы безопасной сделки присутствуют, но требуют внимательного изучения правил и комиссий.

На рис. 1.2 изображена главная страница сайта «Kwork»



Рисунок 1.2 – Главная страница «Kwork»

 **Особенности**:

* Сервис, основанный на концепции «микроуслуг»: исполнители предлагают готовые пакеты услуг (кворки) по фиксированной цене (от 500 рублей и выше).
* Удобен для начинающих специалистов и заказчиков, которые хотят быстро получить небольшой объём работы (логотипы, тексты, переводы и т. п.).
* Имеет встроенную систему безопасной сделки (средства замораживаются у сервиса до выполнения заказа).

 **Недостатки**:

* При более крупных или сложных проектах «микроформат» может быть неудобен, так как требует разбивки заказа на несколько «кворков».
* Сильная конкуренция в популярных категориях: чтобы выделиться, нужно нарабатывать рейтинг и отзывы.
* Определённая комиссия сервиса может казаться высокой для исполнителей, особенно на начальном этапе.

На рис. 1.3 изображена главная страница сайта «Freelansim».

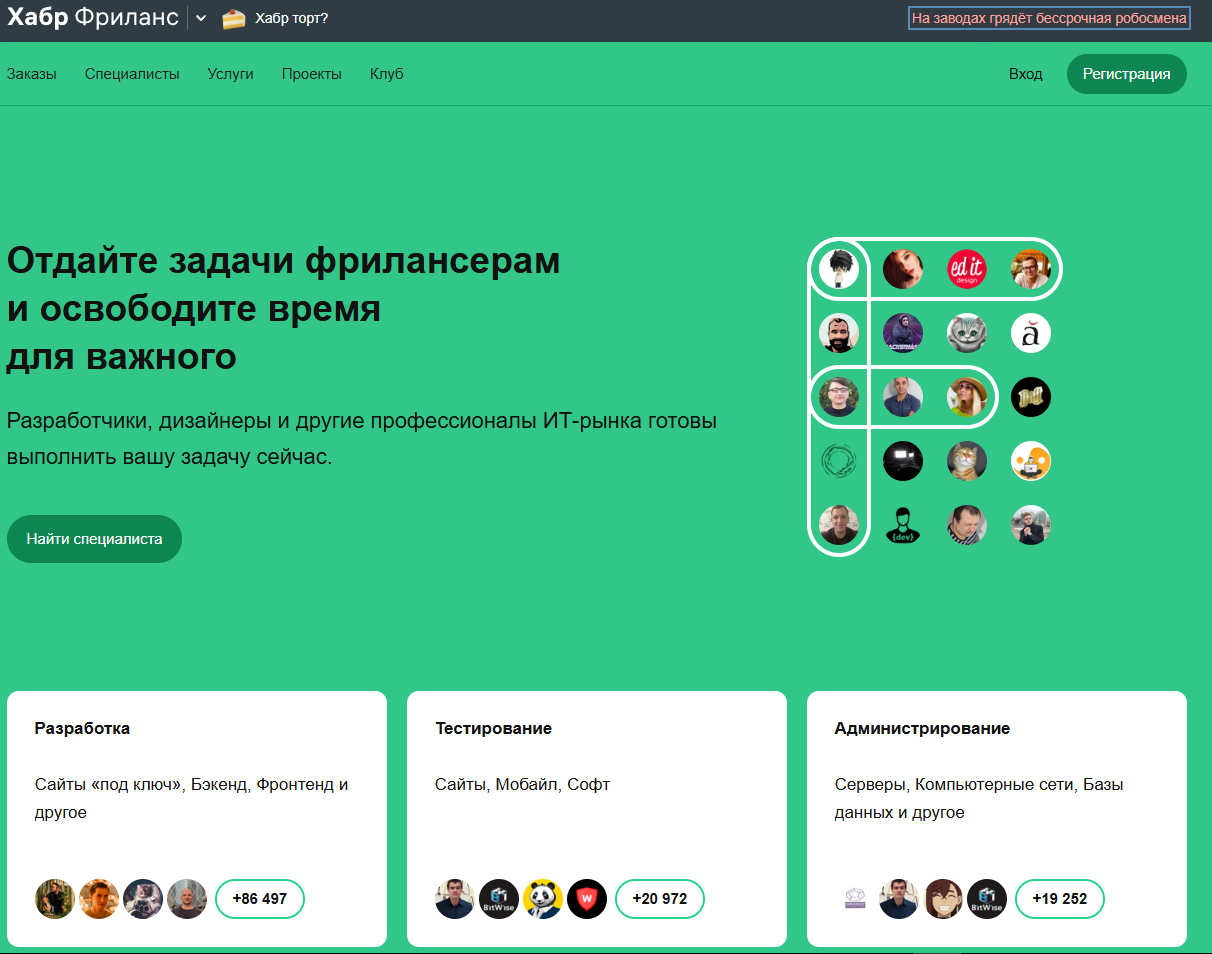


Рисунок 1.3 – Главная страница «Freelansim»

 **Особенности**:

* Главная специализация — IT-сфера: программирование, разработка сайтов и приложений, дизайн интерфейсов.
* Благодаря интеграции с «Хабром» платформа привлекает профессионалов высокого уровня, а также компании, ищущие качественные решения.
* Интерфейс относительно прост и ориентирован на опытных пользователей, которые знакомы с платформами Хабра.
* Предусмотрены удобные категории для быстрого поиска проектов.

 **Недостатки**:

* Платформа сильно ориентирована на технические проекты; для специалистов из других областей (например, копирайтинг, маркетинг) выбор заказов ограничен.
* Новичкам может быть сложно освоиться из-за ориентации на профессионалов.
* Платные услуги, такие как поднятие профиля или проектов, могут быть дорогими для начинающих.

## 1.3 Техническое задание на проект

Данное веб-приложение для фриланс-платформы позволяет заказчикам и исполнителям взаимодействовать с использованием удобной и интуитивно понятной системы.

Структура фрилансерского сайта изображена на рис. 1.4.

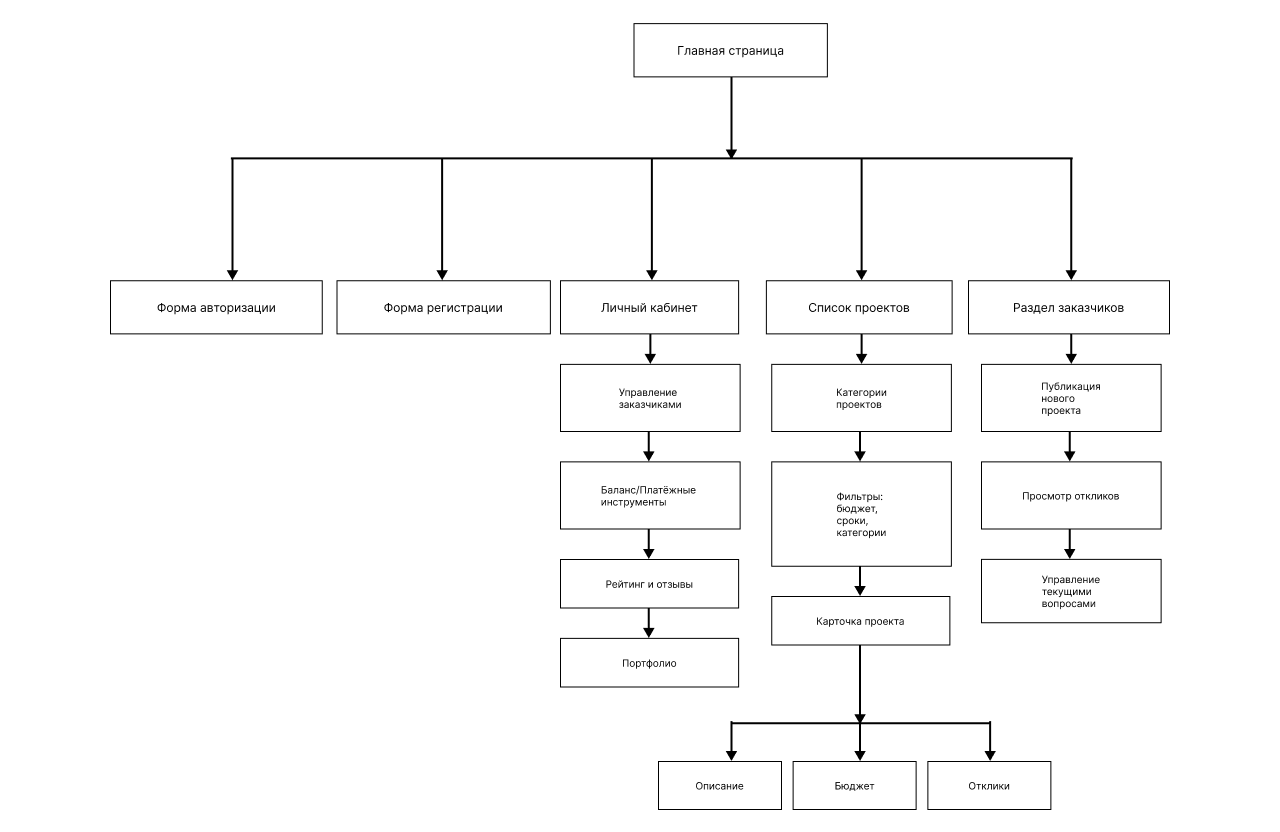


Рисунок 1.4 – Структура фрилансерского сайта

**Цель разработки веб-приложения** – создать платформу, которая упростит процесс поиска заказчиков и исполнителей, а также обеспечит безопасное и комфортное сотрудничество. Разработка подобного проекта значительно облегчит процесс работы для обеих сторон, повышая их эффективность и удовлетворённость от взаимодействия.

**Главные задачи проекта**:

* привлечение новых пользователей (как исполнителей, так и заказчиков) к платформе;
* предоставление подробной информации о проектах и пользователях;
* обеспечение удобного интерфейса для выполнения ключевых задач: публикации проектов, откликов, управления заказами и финансами;
* создание системы обратной связи с пользователями для повышения качества работы платформы;
* внедрение механизмов безопасных сделок для защиты обеих сторон.

Проект должен учитывать современные принципы дизайна, обеспечивать быструю работу и надёжность всех функций.

1.3.1 Требования к функциональным характеристикам

**Функциональные требования** — это перечень функций, которые должна выполнять система, а также описание её поведения в определённых ситуациях и реакции на различные входные данные. В некоторых случаях указывается, что система не должна делать. Эти требования определяют поведение системы, предоставляемые ею сервисы и зависят от типа разрабатываемого веб-приложения и потребностей пользователей.

#### Основные функции веб-приложения:

* **Регистрация** пользователей (заказчиков и исполнителей).
* **Авторизация** с использованием логина и пароля.
* Возможность **публикации новых проектов** заказчиками.
* Возможность **отправки откликов** фрилансерами на опубликованные проекты.
* **Управление заказами**, включая их статусы (в работе, завершён).
* **Система безопасной сделки**, позволяющая проводить финансовые операции через платформу.
* Возможность оставлять **отзывы и оценки** для обеих сторон после завершения работы.
* **Поиск и фильтрация проектов** по категориям, бюджету и срокам.

#### Главное навигационное меню:

Для **неавторизованного пользователя** главное меню должно включать следующие пункты:

* Логотип компании (ссылку на главную страницу).
* Специалисты – страница с специалистами.
* Заказы – работа от заказчиков.
* Вакансии – предложения постоянной работы.
* Раздел **«Техническая поддержка»** — помощь и ответы на часто задаваемые вопросы.
* **Вход в личный кабинет** — кнопка для авторизации или регистрации.

Если пользователь **зарегистрирован и авторизовался в системе**, пункт **«Вход в личный кабинет»** заменяется на **«Личный кабинет»**, а также добавляется пункт **«Выйти»** для выхода из системы.

#### Меню в нижней части страницы (footer):

* Реквизиты компании.
* Пользовательское соглашение.
* Политика конфиденциальности.
* Название компании.

#### Главная страница:

На главной странице должны присутствовать следующие элементы:

* Меню навигации по сайту.
* Краткая информация о **возможностях платформы**.

#### Разделы платформы:

1. **«О платформе»**:  
   Этот раздел предоставляет информацию о назначении и возможностях платформы, включая её преимущества для заказчиков и фрилансеров.
2. **«Техническая поддержка»**:  
   Раздел, где пользователи могут найти ответы на часто задаваемые вопросы, инструкции по использованию платформы, а также обратиться за помощью.
3. **«Регистрация/Авторизация»**:  
   Необходимы для создания личного кабинета и разграничения доступа между различными пользователями. Авторизованные пользователи получают доступ к полному функционалу платформы.
4. **«Личный кабинет»**:  
   Личный кабинет доступен только авторизованным пользователям и включает:
   * **Управление заказами** — просмотр статусов и детализация проектов.
   * **Баланс и платёжные инструменты** — пополнение счёта, вывод средств, просмотр транзакций.
   * **Рейтинг и отзывы** — информация о репутации пользователя.
   * **Портфолио** — возможность добавления своих работ или достижений для фрилансеров.

1.3.2 Требования к графическому интерфейсу

Под графическим интерфейсом понимается вариант оформления главной страницы и графическая оболочка внутренних страниц, демонстрирующие общее визуальное (композиционное, цветовое, шрифтовое, навигационное) решение основных страниц сайта. При разработке сайта должны быть использованы преимущественно темные цвета. Основные разделы сайта должны быть доступны с первой страницы. В основном навигационном меню используются исключительно текстовые ссылки. При необходимости дополнительно можно использовать тематические картинки к разделам с жёстко прописанными адресами этих страниц. Так же, исключительно текстом. Желательно так же — название сайта, слоган, контактная информация и т.д. В идеале — текст должен быть текстом, а не графикой. Главное при разработке интерфейса – сделать интуитивно понятным для целевой аудитории.

В дизайне сайта не должны присутствовать:

* мелькающие баннеры;
* много сливающегося текста;
* ядовитые цвета (неоновый, лимонный и пр).

Далее идет показано, насколько интерфейс веб-приложения соответствует требованиям. На рис. 1.9 изображено навигационное меню веб-приложения.



Рисунок 1.9 – Навигационное меню

На рис. 1.10 изображено меню в нижней части страницы(footer).

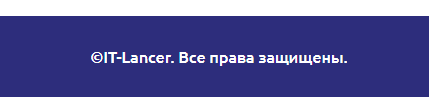


Рисунок 1.10 – Footer меню

Главная страница веб-приложения (рис. 1.11, 1.12, 1.13)

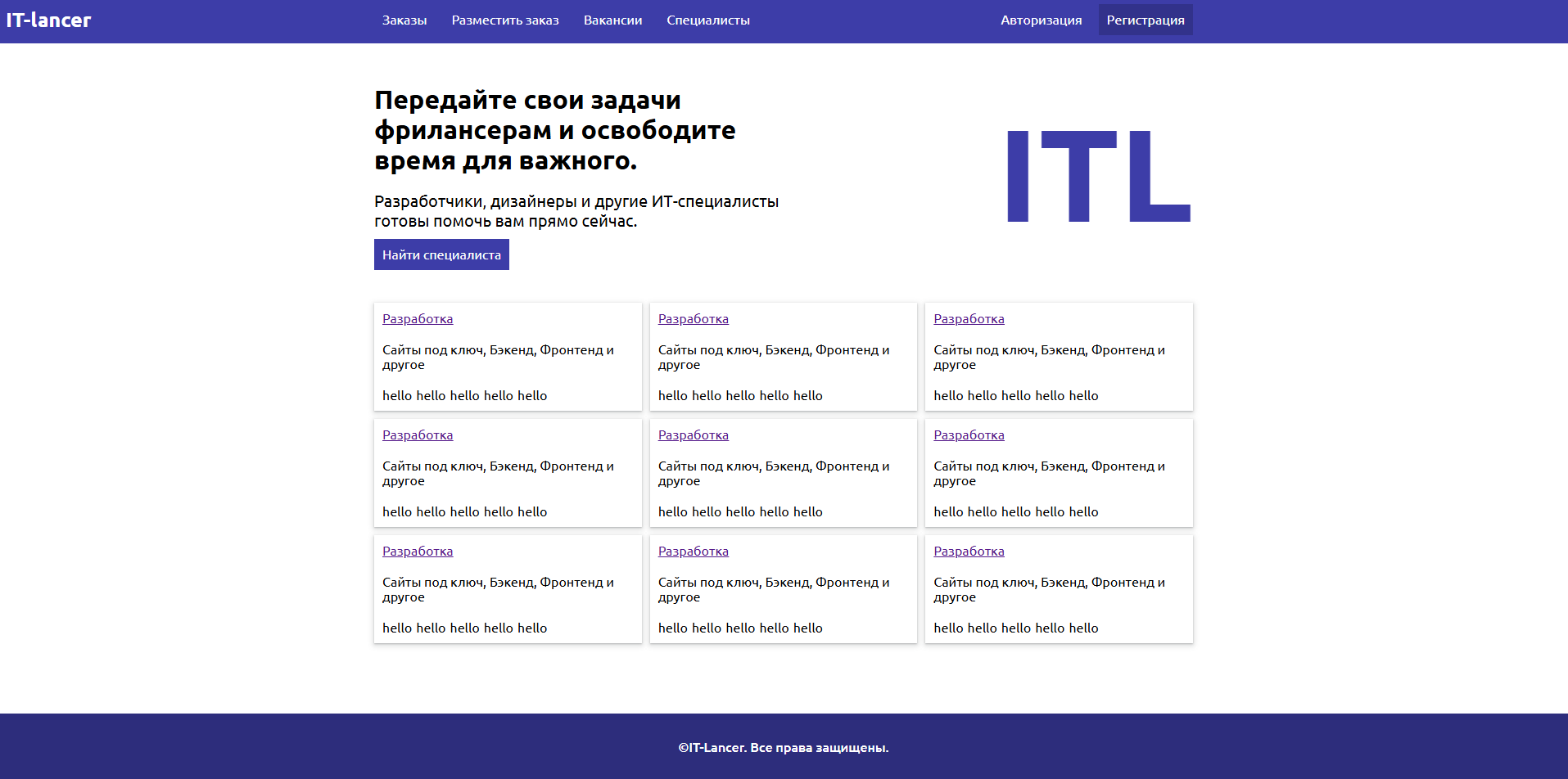


Рисунок 1.11 – Главная страница

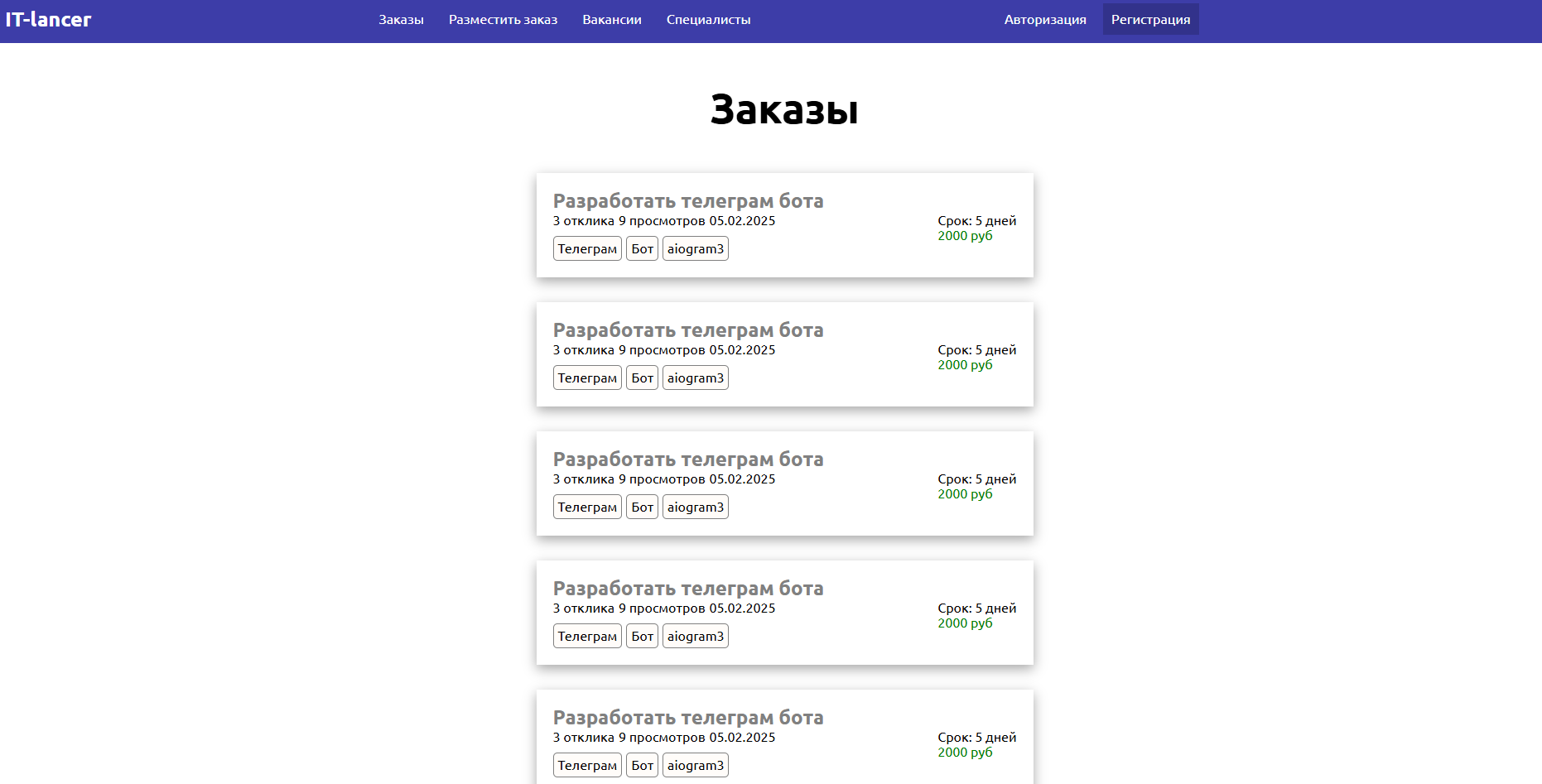


Рисунок 1.12 – Раздел заказы

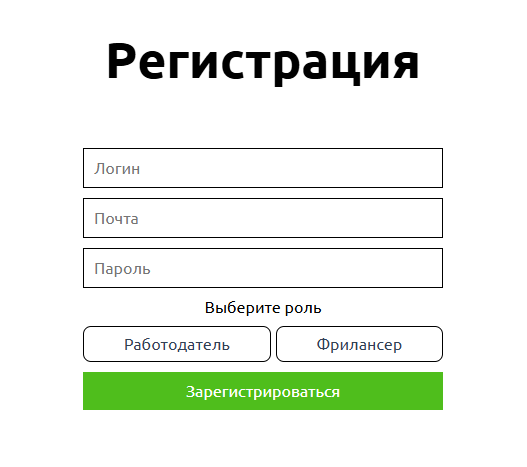


Рисунок 1.13 – Страница регистрации

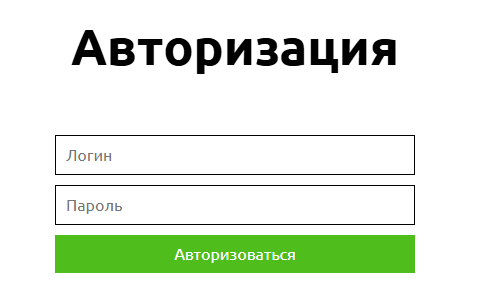


Рисунок 1.14 – Страница авторизации

На рис. 1.15 изображен личный кабинет.

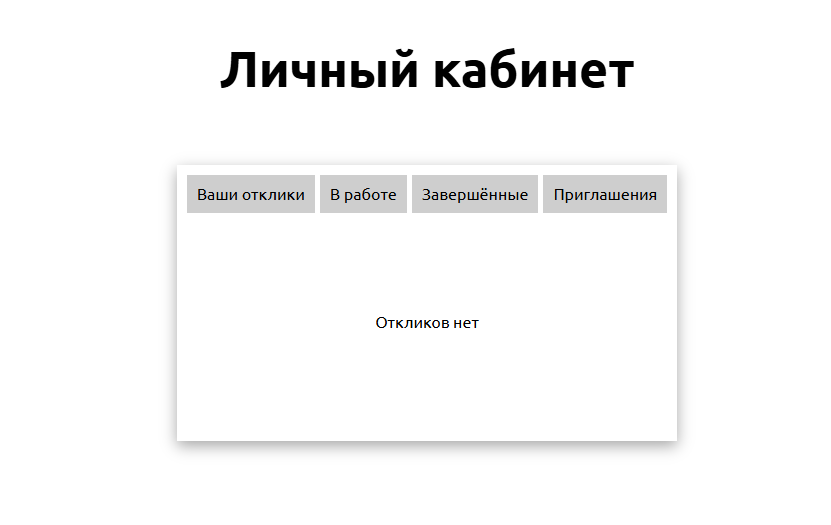


Рисунок 1.15 – Страница личного кабинета

1.3.3 Требования к техническим характеристикам

Для стабильной работы сайта будет использоваться облачный хостинг — объединённая система серверов, на которых располагаются клиентские сайты.

Таким образом, выделяемые для клиента мощности не ограничиваются одним сервером, а распределяются между несколькими серверами. Это обеспечивает бесперебойную работу платформы вне зависимости от выхода из строя какого-либо одного сервера.

**Основные преимущества облачного хостинга:**

1. **Гибкость** — для облачного хостинга можно подобрать оптимальную конфигурацию ресурсов под потребности платформы.
2. **Надёжность** — благодаря использованию сети серверов платформа будет работать стабильно. Один или несколько серверов предоставляют вычислительные ресурсы, другие — выполняют резервное копирование и восстановление данных при необходимости. Такая архитектура исключает заторможенность или недоступность сайта.
3. **Доступность** — платформа будет доступна 24 часа в сутки 7 дней в неделю. Хостинг не ограничивается одним сервером, а использует ресурсы целой сети серверов, объединяющей более 100 единиц. Для оптимальной работы используется распределение нагрузки (балансировка нагрузки) и разделение дискового пространства.
4. Для полноценного использования платформы пользователю подойдёт компьютер или мобильное устройство с современным браузером и стабильным интернет-соединением.

## 1.4 Технические и программные средства для разработки продукта

В ходе разработки использовались различные технологии, включая языки гипертекстовой разметки, таблицы стилей, объектно-ориентированные языки программирования, а также сервисы для проектирования системы, создания диаграмм, написания плана разработки и многое другое.

1. **Текстовый редактор**:

Для разработки веб-приложения использовался **Visual Studio Code**. Этот редактор позиционируется как "лёгкий" инструмент для кроссплатформенной разработки веб- и облачных приложений. Были также использованы плагины для упрощения и ускорения процесса разработки.

1. **Дизайн интерфейса**:

Для разработки макета сайта применялся **Figma**. Этот онлайн-сервис позволяет проектировать интерфейсы и прототипы, организовывать совместную работу в режиме реального времени.

1. **Языки программирования и разметки**:

* **HTML** — основа технологии гипертекста для структурирования содержимого веб-страниц.
* **CSS** — каскадные таблицы стилей, которые отвечают за визуальное оформление сайта, его дизайн, кроссбраузерность и кроссплатформенность.
* **JavaScript** — используется для реализации интерактивности и динамического контента. Например, обновление данных на странице без её перезагрузки.
* **jQuery** — библиотека JavaScript, упрощающая манипуляции с DOM-элементами и добавляющая готовые функции для ускорения разработки.

1. **Проектирование диаграмм и схем**:

**Draw.io** — инструмент для построения диаграмм и UML-моделей, создания блок-схем и других графических элементов. Главным преимуществом является бесплатность и простота использования.

1. **Локальный сервер**:

**OpenServer** — локальный сервер для разработки, отладки и тестирования веб-приложений. Благодаря простоте настройки он широко используется начинающими и опытными веб-разработчиками. Этот инструмент позволяет быстро развернуть рабочую среду без сложных манипуляций.

Эти средства обеспечили реализацию всех этапов разработки веб-приложения, начиная с проектирования и заканчивая тестированием и развёртыванием продукта.

# ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 2.1 Проектирование фриланс сайта

Проектирование системы — это один из первых этапов разработки сайта. На этом этапе разрабатывается архитектура будущего сайта, с учетом специфики бизнеса и целей веб-проекта.

Проектирование сайта – ключевой этап создания сайта, в процессе которого определяются цели и задачи создания ресурса, способы достижения целей и как сайт в итоге будет выглядеть и функционировать.

Преимущества проектирования: высокая гарантия достижения результатов, экономия времени и средств, эффективное разделение работы. Исправить ошибку на этапе проектирования гораздо дешевле, чем на этапе дизайна, вёрстки или программирования.

Чтобы отобразить сущность взаимодействия пользователя с сайтом разработана Диаграмма прецедентов.

Проектирование разрабатываемой системы. Подразумевает под собой составления диаграмм взаимодействия, между собой разных сущностей проекта. Разработана схема взаимодействия пользователя с сайтом.

На рис. 2.2 изображена диаграмма деятельности.

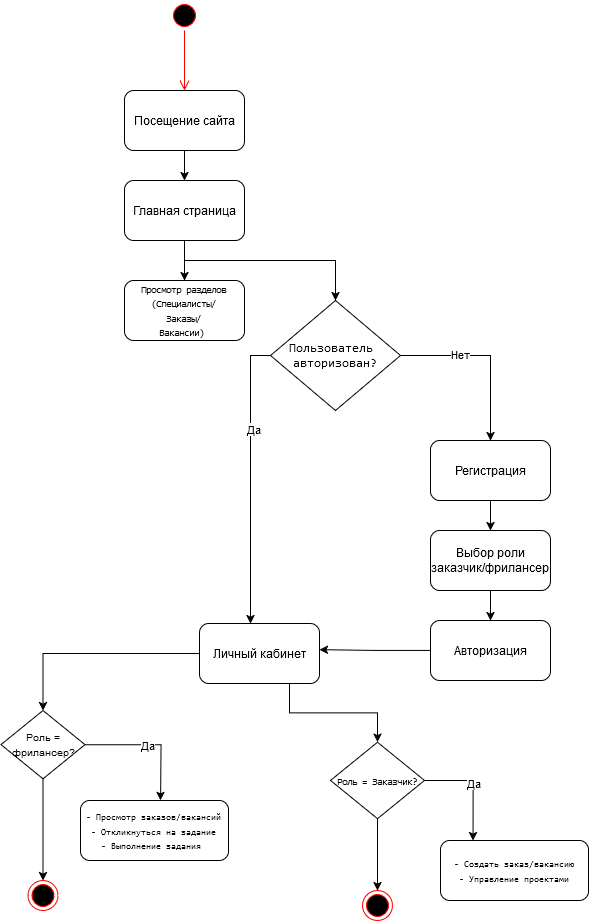


Рисунок 2.2 – Диаграмма деятельности

Диаграмма деятельности – это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Результат может привести к изменению состояния системы или возвращению некоторого значения.

На рис. 2.3 изображена диаграмма прецедентов.

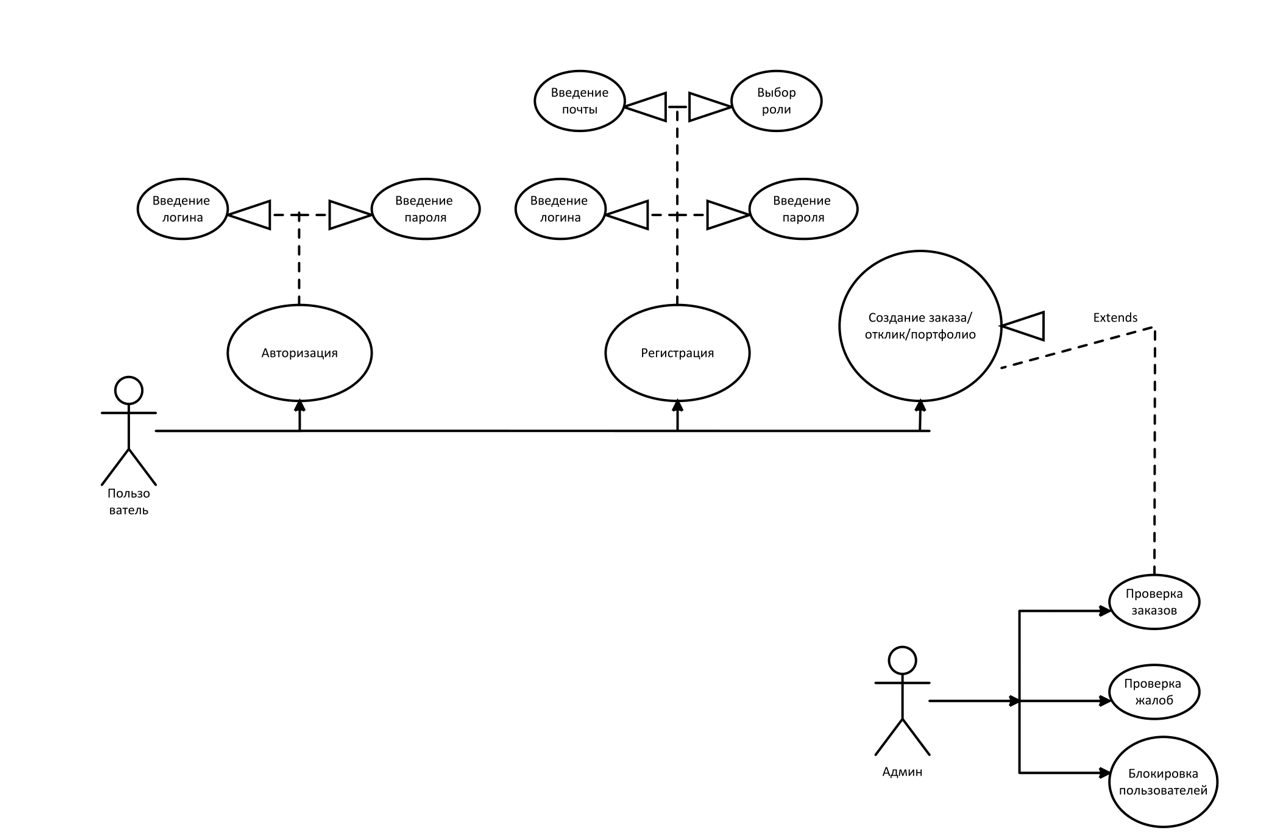


Рисунок 2.3 – Диаграмма прецедентов

Диаграмма прецедентов – это графическое представление всех или части актеров, прецедентов и взаимодействий в системе. В каждой системе обычно есть главная диаграмма прецедентов, которая отображает границы системы (актеров) и основное функциональное поведение системы (прецеденты). Другие диаграммы прецедентов могут создаваться при необходимости.На диаграмме отражены отношения между пользователями и прецедентами, которые являются составной частью модели прецедентов.

На рис. 2.4 изображена ERмодель данных

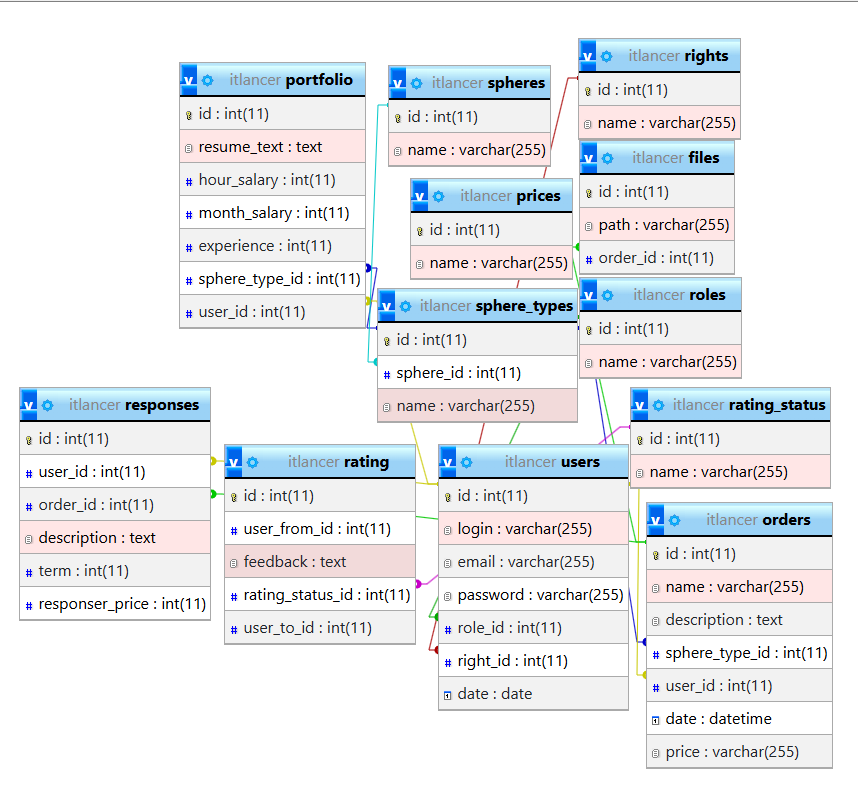


Рисунок 2.4 – ERмодель данных

Диаграмма отношений сущностей — это визуальное представление базы данных, которое показывает, как связаны элементы внутри. Диаграмма ER состоит из двух типов объектов — сущностей и отношений. Сущность в этом контексте — это компонент данных из набора данных, отображаемый в виде фигуры на холсте. Отношения между сущностями представлены в виде строк, которые имеют специальные окончания строк, называемые кардинальностями, которые описывают, как два элемента базы данных взаимодействуют друг с другом.

## 2.3 Технология разработки веб-приложения

Для упрощения разработки и оптимизации кода, некоторые части приложения, которые используются почти на каждой странице, были вынесены в отдельные компоненты:

1. Dropzone – скрипт для создания зоны загрузки файлов с предпросмотром и валидацией. (рис. 2.5).

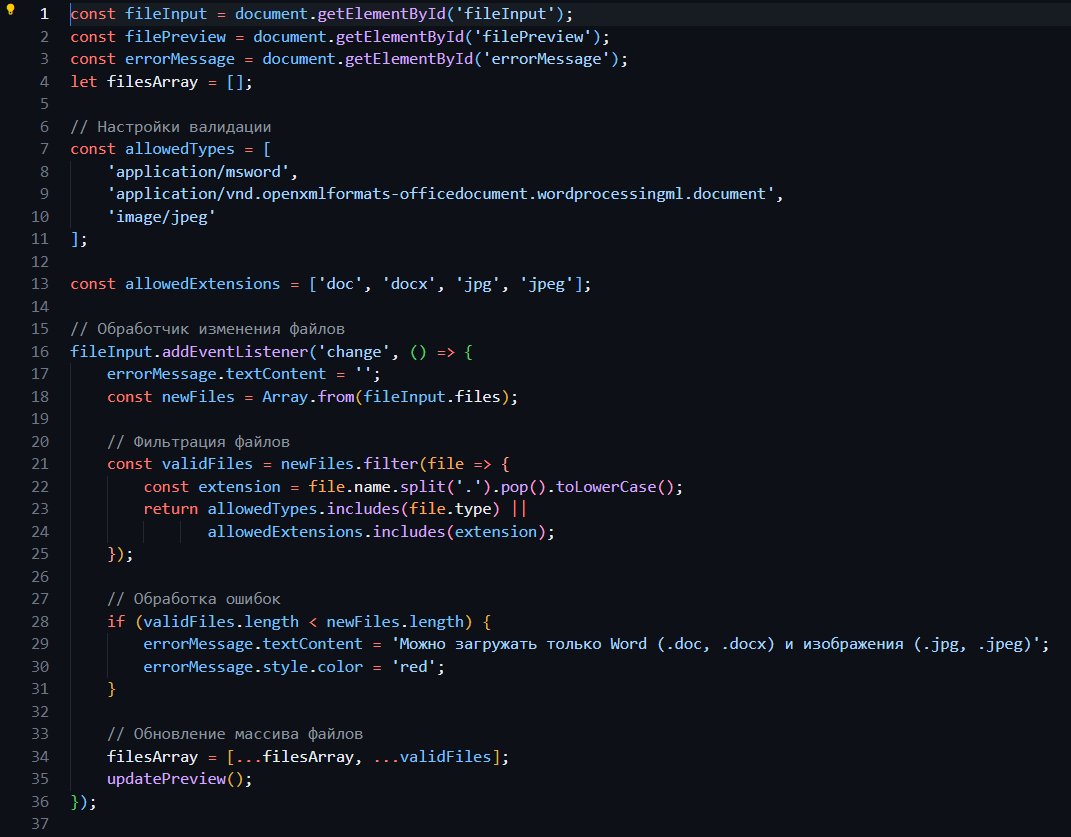


Рисунок 2.5 – Код DropZone

Выделяют три вида веб-приложений, которые определяют подход к разработке:

[SPA или Single Page Application](https://www.azoft.ru/blog/spa-mpa-pwa/) – это одностраничное веб-приложение, которое загружается на одну HTML-страницу. Пользователям не нужно перезагружать или подгружать дополнительные страницы благодаря динамическому обновлению с использованием JavaScript. То есть пользователи видят в браузере основной контент, а когда прокручивают страницу или переходят на другие, нужные элементы просто подгружаются.

MPA или Multi Page Application – это многостраничные приложения, где при каждом изменении данных или загрузке новой информации страница обновляется.

[PWA или Progressive Web App](https://www.azoft.ru/blog/progressive-web-apps/" \t "_blank) – это своего рода гибрид сайта с мобильным приложением. PWA может устанавливаться на главный экран смартфона, отправлять push-уведомления и работать в офлайн режиме.

Была выбрана технология веб-разработки MPA, потому что в нём можно оптимизировать каждую из страниц приложения под нужные ключевые запросы, даёт привычность для пользователя за счёт просто интерфейса и классической навигации.

## Система контроля версий

Система управления версиями (от[англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA" \o "Английский язык)VersionControlSystem, VCS или RevisionControlSystem) - [программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Программное обеспечение) для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Такие системы наиболее широко используются при разработке программного обеспечения для хранения [исходных кодов](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4" \o "Исходный код) разрабатываемой программы. Однако системы могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов. В частности, системы управления версиями применяются в [САПР](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BD%D1%8B%D1%85_%D1%80%D0%B0%D0%B1%D0%BE%D1%82" \o "Система автоматизации проектных работ), обычно в составе систем управления данными об изделии ([PDM](https://ru.wikipedia.org/wiki/PDM" \o "PDM)). Управление версиями используется в инструментах [конфигурационного управления](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5" \o "Конфигурационное управление) (Software Configuration Management Tools).

GitHub, представляет собой веб-приложение, в котором размещаются репозитории Git. GitHub позволяет легко обмениваться кодом между несколькими компьютерами и разработчиками. GitHub стал инструментом централизованной организации сообщества opensource и, в свою очередь, используется тысячами компаний и команд. У некоторых пользователей GitHub есть один репозиторий, с которым работают. Пример созданного репозитория с моим сайтом рис 2.10.

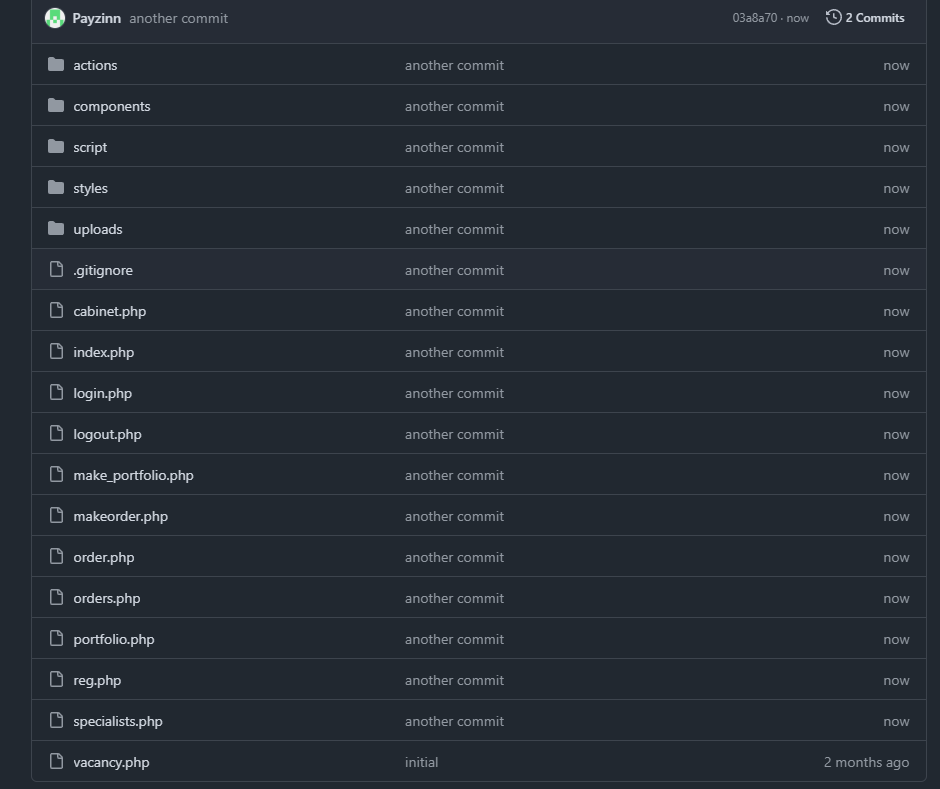


Рисунок 2.6 – Репозиторий с проектом

Наиболее важные инструменты слоев GitHub поверх Git включают в себя:

* запросы на извлечение, которые позволяют разработчикам предлагать изменения и запрашивать отзывы и обсуждения от других разработчиков (так называемый CodeReview);
* проблемы, которые разработчики используют для отслеживания ошибок, улучшений или других запросов, связанных с данным репозиторием.

Преимущества:

* распределенная VCS, которую можно использовать в автономном режиме;
* git настолько вездесущ, что большинство других разработчиков знакомы с ним, многие проекты с открытым исходным кодом размещаются на GitHub, поэтому это отличное место, чтобы найти и внести вклад в эти проекты;
* github также включает сторонние интеграции чтобы упростить рабочие процессы.

**Недостатки:**

Основным недостатком как Git, так и GitHub является то, что у них обоих довольно трудно обучиться.

Репозиторий или хранилище – это каталог, в котором хранятся файлы вашего проекта. Репозиторий может быть расположен в хранилище GitHub или в локальном хранилище на вашем компьютере. Пользователь может хранить файлы кодов, изображения, аудио или всё, что связано с проектом, в хранилище.

Удалённый репозиторий – это модификации проекта, которые хранятся в интернете или ещё где-то в сети. Удалённых репозиториев может быть несколько, как правило доступен для вас либо только на чтение, либо на чтение и запись.

# **ОРГАНИЩАЦИОННЫЙ РАЗДЕЛ**

## 3.1 **Описание назначения веб-сайта**

## 3.1 Описание назначения веб-сайта

## Разрабатываемый проект представляет собой онлайн-платформу, предназначенную для создания эффективного взаимодействия между заказчиками и фрилансерами в digital-сфере. Основная цель веб-сайта — упростить поиск исполнителей для проектов различной сложности, а также предоставить специалистам возможность демонстрировать свои навыки и находить заказы. Платформа будет реализована в формате многофункционального сервиса с интуитивным интерфейсом, ориентированным на быстрое решение задач обеих сторон.

## Ключевым элементом главной страницы станет блок «Актуальные проекты», который позволит пользователям оперативно знакомиться с новыми заказами, фильтруя их по категориям (дизайн, программирование, копирайтинг и др.). Для повышения вовлеченности будет реализован раздел «Рейтинги и отзывы», обеспечивающий прозрачность репутации исполнителей и заказчиков.

## Проект решает проблему географической и временной ограниченности в поиске работы и сотрудников. Благодаря онлайн-формату пользователи смогут размещать вакансии, подавать заявки на проекты, а также заключать безопасные сделки через встроенную платежную систему без необходимости личных встреч. Это сократит временные затраты на поиск и формализацию сотрудничества.

## Внедрение веб-приложения позволит автоматизировать процессы подбора кандидатов, уведомления о новых проектах, контроля сроков выполнения задач и проведения финансовых расчетов. В результате платформа оптимизирует workflow как для фрилансеров, стремящихся к постоянной загрузке, так и для заказчиков, нуждающихся в оперативном закрытии бизнес-задач.

## 3.2Руководство пользователя

Если пользователь регистрируется на данном сайте у него появляется возможность не только просматривать свежие новости, но и попробовать демо версию игры. Для того чтобы посмотреть новостную ленту нужно нажать на кнопку «Новости» и выбрать соответствующую новость (рис.3.1 и рис. 3.2).

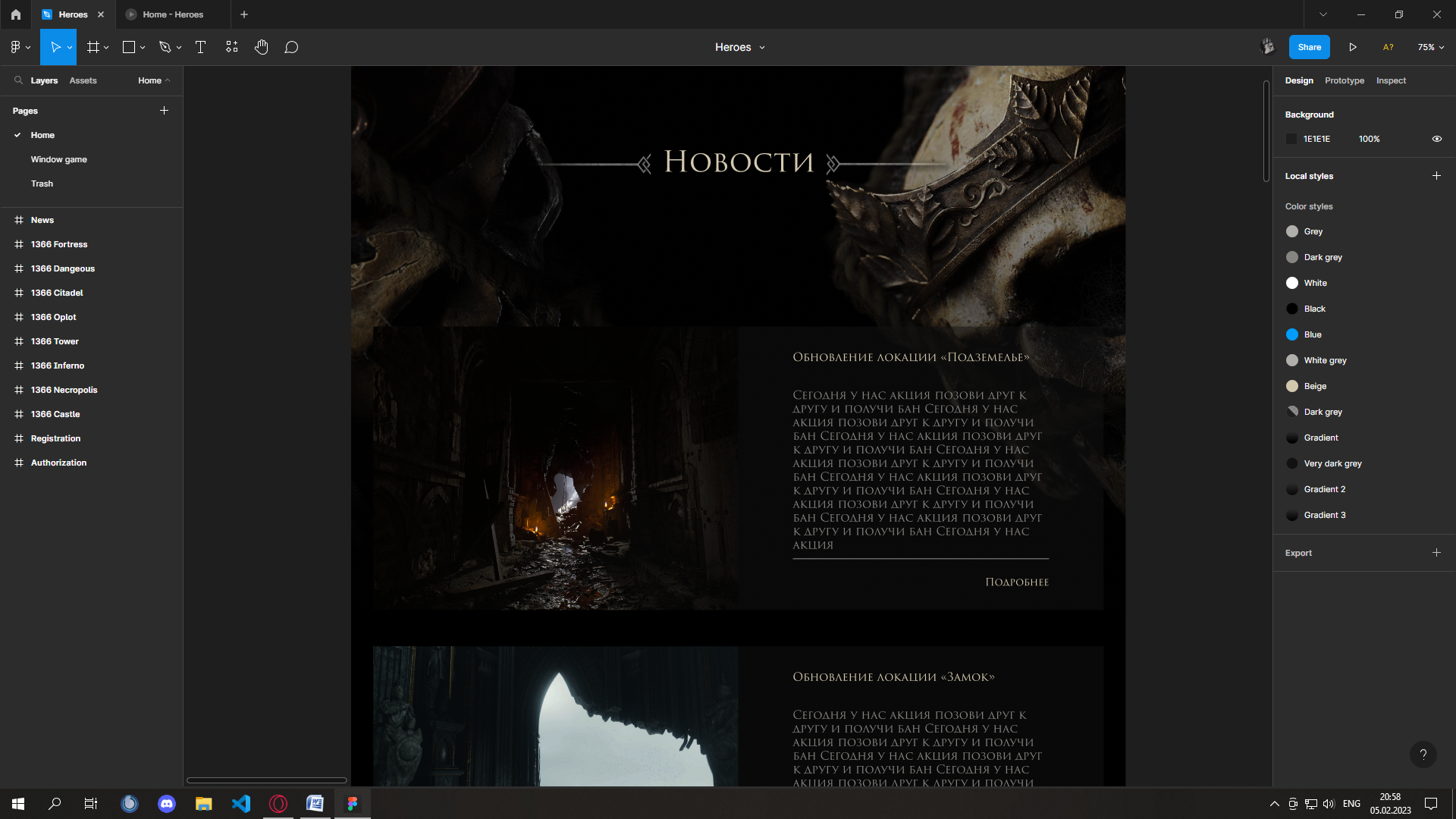


Рисунок 3.1 – Страница «Новости»

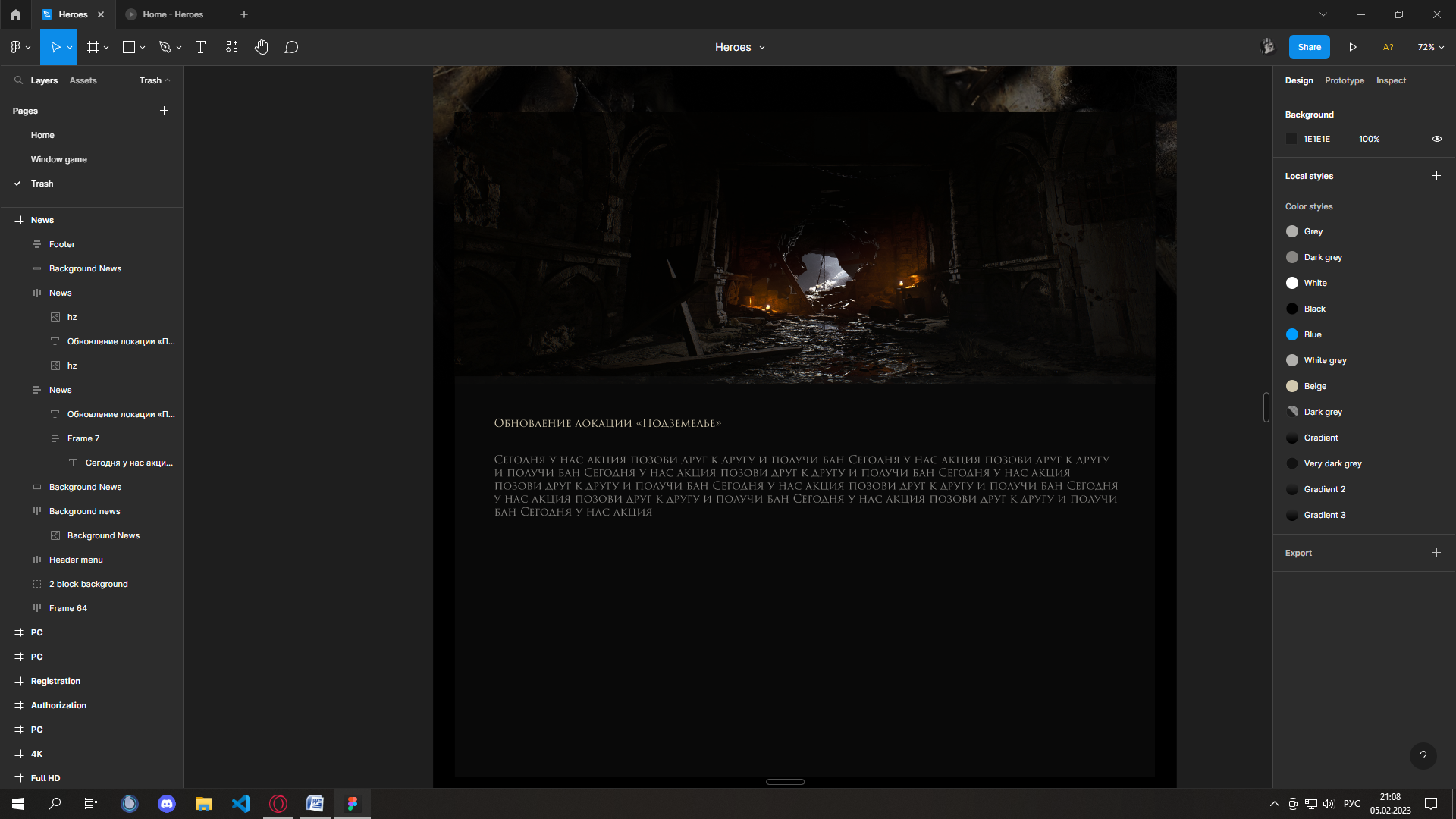


Рисунок 3.2 – Страница блока новостей

Также сайте можно поиграть в демо версию (рис. 3.3).

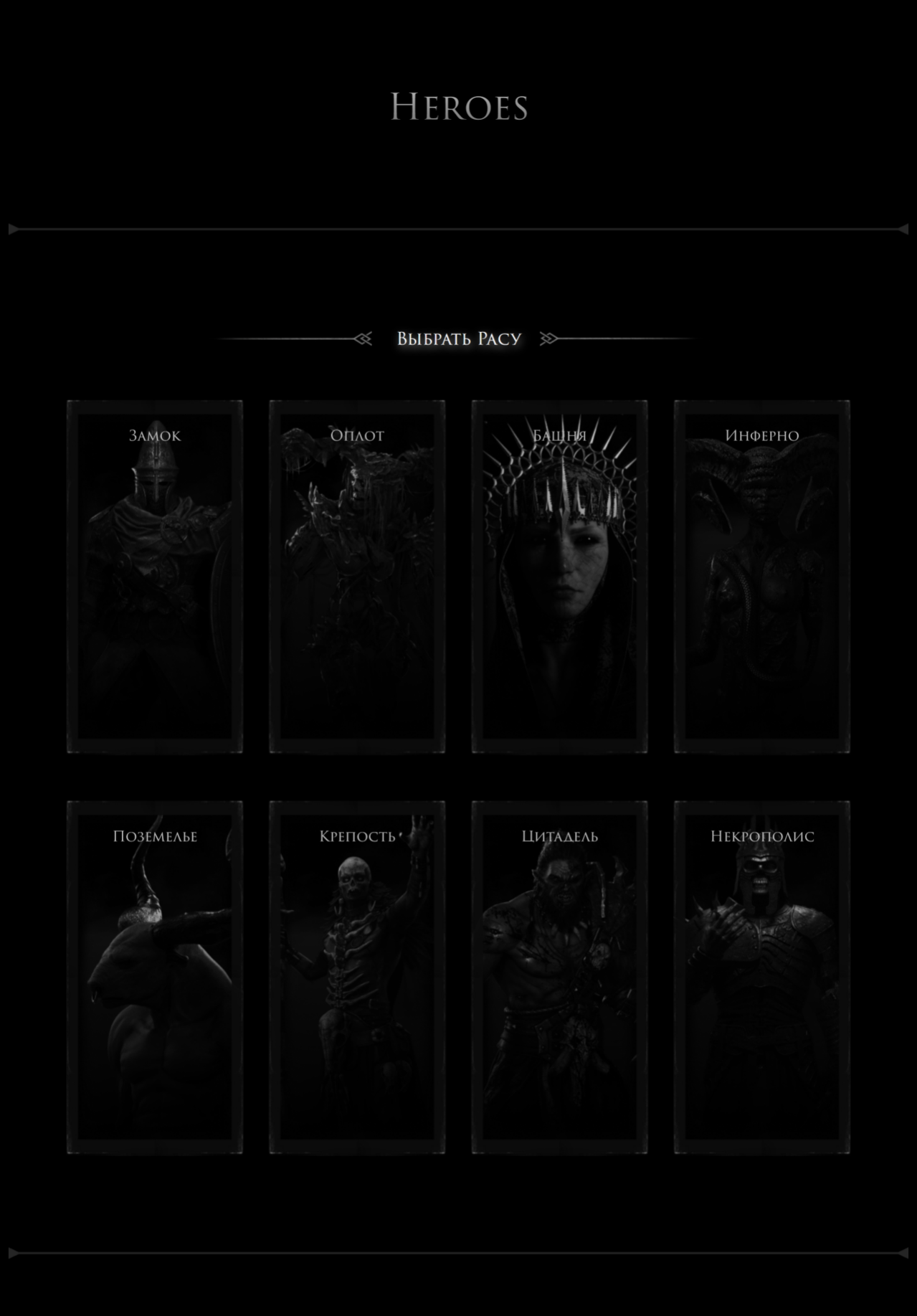


Рисунок 3.3 – Страница выбора фракций

Также у зарегистрированного пользователя есть возможность посмотреть об игре. Для этого на главной странице находится кнопка «Об игре». При нажатии на которую происходит переход страницу с информацией о фракциях, картах (рис. 3.4 и рис. 3.5).

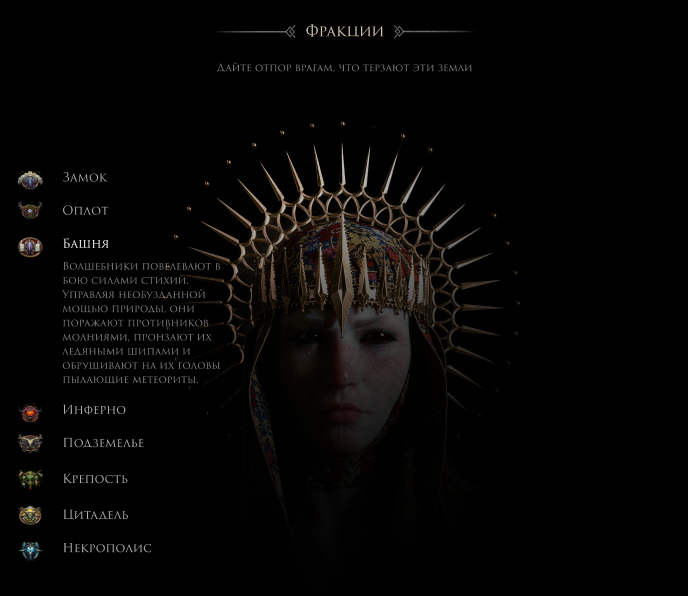


Рисунок 3.4 – Блок «Фракции»



Рисунок 3.5 – Блок «Локации»

Если пользователю необходимо сменить логин, e-mail или пароль, в таком случае в меню есть вкладка «Учётная запись», при переходе на эту страницу необходимо изменить данные (рис. 3.6).

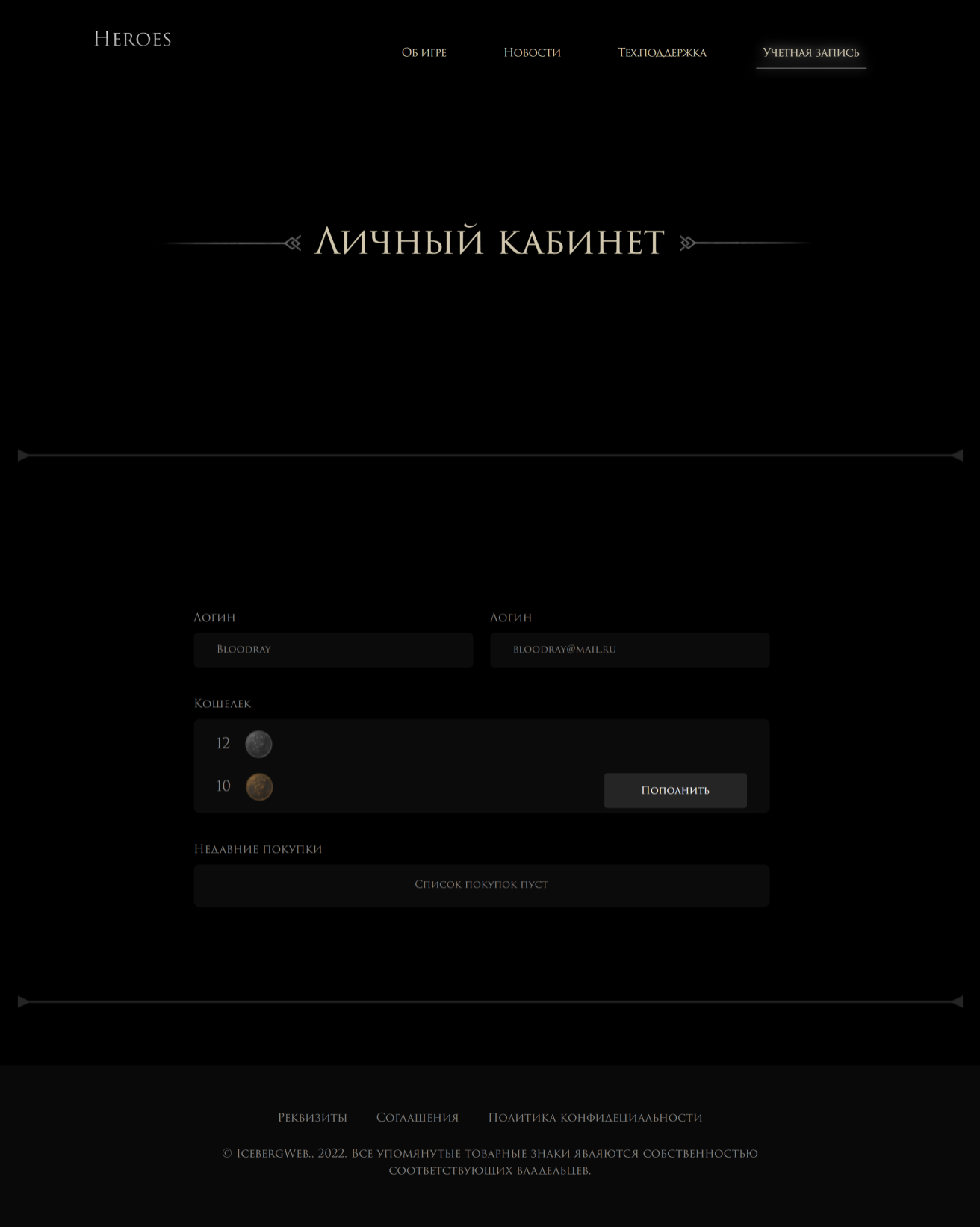


Рисунок 3.6 – Страница «Личный кабинет»

# ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта на тему: «Проектирование и разработка игрового онлайн проекта Heroes» (комплексный проект) был разработан веб сайт для свободно времяпровождения. При разработке портала были выполнены следующие задачи:

* исследована предметная область, выявлена целевая аудитория;
* разработана структура системы для успешного продвижения, удобство использования для пользователя;
* разработано техническое задание, в котором были зафиксированы все требования для сайта;
* разработанмакет пользовательского интерфейса соответствующий техническому заданию;
* разработан пользовательский интерфейс на сайте в соответствии с макетом;
* разработана серверная часть для формирования контента под каждого пользователя;
* реализована автоматизация в работе базы данных для обеспечения удобства доступа к контенту, безопасности и высокой эффективности работы пользователей с огромным объёмом информации;
* проведено тестирование системы в результате чего, были устранены возможные ошибки на сайте.

При создании системы были задействованы такие языки программирования как: Javascript, CSS3, HTML5, Bootstrap 5, Sass, jQuery. Была проанализирована логика система, недостатки с которыми можно было бы столкнуться и разработан макет самого сервиса.

Портал понравится всем интересующимся игровымжанром RPG!

Задачи дипломного проекта решена, цель достигнута.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

Стандарты

1. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартамот 18 декабря 1978 г. N 3351: пересмотр: 01.01.2010: введен впервые: дата введения 1980-01-01 /Единая система программной документации: Сборник национальных стандартов. – Москва: Стандартинформ, 2010. – 4 с.
2. П-ОАТК-03.143-2023. СТАНДАРТ ПРЕДПРИЯТИЯ. Дипломные работы и проекты. Курсовые работы и проекты. Правила оформления и структура: свод правил: утвержден и введен в действие Приказом директора С.В. Евсеева БПОУ "Омский АТК" от 10.05.2023 года: пересмотрП-ОАТК-03.143-2018: дата введения 2023-05-10/ подготовленМинистерством образования Омской области бюджетным профессиональным образовательным учреждением Омской области «Омский автотранспортный колледж»//БПОУ «Омский АТК»: [сайт]. – URL: [https://xn--80aukr.xn--p1ai/students/kursovye-raboty-i-diplomnoe-proektirovanie/%D0%A1%D0%A2%D0%9F%20\_2018.pdf](https://xn--80aukr.xn--p1ai/students/kursovye-raboty-i-diplomnoe-proektirovanie/СТП%20_2018.pdf) (дата обращения: 10.05.2023).

Основные источники

1. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5: зарубежная компьютерная литература/ Р. Никсон. – 6-е изд. – Санкт-Петербуг: Питер, 2023. – 816 с.–ISBN978-5-4461-1970-7.
2. Флэнаган, Д.JavaScript. Полное руководство: справочник/ Д. Флэнаган. – Москва: Диалектика-Вильямс, 2021. – 720 с.–ISBN978-5-907203-79-2.
   * 1. Чем frontendотличается от backend’а? Объясняем на мемах: сайт / Российская компания в сфере онлайн образования. – Москва. – URL:<https://blog.skillfactory.ru/chem-frontend-otlichaetsya-ot-backenda-obyasnyaem-na-memah/>(дата обращения: 07.05.2023).
     2. Чем веб-приложение отличается от сайта: сайт/ Блог связанный с информационными технологиями, бизнесом и интернетом –URL: <https://itproger.com/news/chem-veb-prilozhenie-otlichaetsya-ot-sayta> (дата обращения: 07.05.2023).
     3. Web 3.0: Теория: сайт / Блог связанный с информационными технологиями, бизнесом и интернетом. –URL: <https://forklog.com/sp/web3-0/theory/> (дата обращения: 07.05.2023).
     4. Целевая аудитория: сайт / Блог связанный с информационными технологиями, бизнесом и интернетом.–URL: <https://ipipe.ru/info/celevaya-auditoriya> (дата обращения: 07.05.2023).
     5. Формирование требований и классификация требований: сайт / Блог связанный с информационными технологиями, бизнесом и интернетом.–URL: <https://analytics.infozone.pro/formation-requirements-and-classification-requirements/> (дата обращения: 07.05.2023).
     6. ORMили как забыть о проектировании БД: сайт / Блог связанный с информационными технологиями, бизнесом и интернетом.–URL: <https://habr.com/ru/articles/237889/> (дата обращения: 07.05.2023).
     7. Зачем нужна структура проекта и как собрать её за семь шагов. Разбираем на примере: сайт / Российская компания в сфере онлайн образования. – Москва. – URL: <https://skillbox.ru/media/management/zachem-nuzhna-struktura-proekta-i-kak-sobrat-eye-za-sem-shagov-razbiraem-na-primere/> (07.05.2023).
     8. Одностраничные (spa)и многостраничные (pwa) веб-приложения: сайт / Блог связанный с информационными технологиями, бизнесом и интернетом. –URL: <https://vc.ru/seo/108149-odnostranichnye-spa-i-mnogostranichnye-pwa-veb-prilozheniya> (дата обращения: 07.05.2023).
     9. Что такое RESTfulAPI: сайт / Американская компания в сфере информационных технологий. – Сиэтл. – URL: <https://aws.amazon.com/ru/what-is/restful-api/> (дата обращения: 07.05.2023).
     10. Model-View-Controller: сайт / Свободная энциклопедия. – Сан-Франциско. – URL: <https://ru.wikipedia.org/wiki/Model-View-Controller> (датаобращения: 06.05.2023).
     11. Model-View-ViewModel: сайт / Свободная энциклопедия. – Сан-Франциско. – URL: <https://ru.wikipedia.org/wiki/Model-View-ViewModel> (дата обращения: 06.05.2023).
     12. Model-View-Presenter: сайт / Свободнаяэнциклопедия. –Сан-Франциско. – URL: <https://ru.wikipedia.org/wiki/Model-View-Presenter> (дата обращения: 06.05.2023).
     13. Микросервисная архитектура: сайт / Свободная энциклопедия. –Сан-Франциско. – URL:<https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0> (дата обращения: 06.05.2023).
     14. Git: сайт / Свободная энциклопедия. – Сан-Франциско. – URL: <https://ru.wikipedia.org/wiki/Git> (дата обращения: 07.05.2023).
     15. GitHub: сайт / Свободная энциклопедия. – Сан-Франциско. – URL: <https://ru.wikipedia.org/wiki/GitHub> (дата обращения: 07.05.2023).
     16. Что Такое GitHub И Для Чего Именно Он Нужен: сайт / Литовская компания провайдер веб-хостинга. –Каунас. – URL: <https://www.hostinger.com.ua/rukovodstva/chto-takoje-github/> (дата обращения: 07.05.2023).
     17. Работа с удалёнными репозиториями: сайт / Пользовательская документация по работе с git. – URL: <https://yunwuxin1.gitbooks.io/git/content/ru/e67d21b34cfd876b83d0216b91c95cbf/ddfaf49c3da628dc46caa4f67f584d79.html> (дата обращения: 07.05.2023)
     18. Тестирование сайтов: основные этапы: сайт. / Украинская компания онлайн образования.– Киев. – URL: <https://qalabs.com.ua/testirovanie-saitov-osnovnye-etapy.html> (дата обращения: 07.05.2023).
     19. VDS – Что это такое: сайт. / Российский облачный провайдер. –Санкт-Петербург. – URL: <https://1cloud.ru/blog/chto-takoe-vds> (дата обращения: 07.05.2023).
     20. Drawio: сайт / Программное обеспечение для рисования графиков. – Торонто. – URL: <https://app.diagrams.net/>(дата обращения: 02.05.2023).
     21. ОсновыJavaScript: сайт / Справочный материалJavaScript. – Торонто. –URL: <https://learn.javascript.ru/intro>(дата обращения: 05.05.2023).
     22. Основы MySQL: сайт / Проект о компьютерных технологиях. – Москва. –URL: <https://www.opennet.ru/docs/RUS/sql/> (дата обращения: 05.05.2023).