

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский национальный исследовательский технологический университет»**

**(ФГБОУ ВО «КНИТУ»)**

Кафедра\_Интеллектуальных систем и управления информационными ресурсами\_

Направление\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_02.03.03\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4311-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Уровень образования\_\_\_\_\_\_\_\_\_\_\_\_бакалавр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(бакалавр, специалист, магистр)

Вид ВКР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_проектный\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( проектный, исследовательский, комбинированный)

Тема Разработка приложения для студии дизайна интерьера\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Д.А. Ахметшин, доцент)

Нормоконтролер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (М.Г. Лаптева, старший преподаватель )

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Д.Л. Егоров, доцент)

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Ю.И. Ахмадеев)

Консультанты:

по библиографии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Т.В. Толок)

2025 г.

Содержание

Введение……………………………………………………………………….…3

Цель проекта……………………………………………………………………..4

1. Изучение предметной области………………………………………………..5

1.1. Аналоги. Плюсы и минусы…………………………………………………5

1.2. Область применения приложения……………………………………….....6

1.3. Требования к приложению………………………………………………….7

2. Проектирование приложения……………………………………………...…8

2.1. Архитектура приложения ……………………………………………...…..8

2.2. Структура приложения……………………………………………………..9

2.3. Макет приложения………………………………………………………....10

2.4. База данных……………………………………………………………...…14

3. Реализация практической части…………………………………………….17

3.1. Клиентская часть………………………………………………………...…17

3.2. Серверная часть………………………………………………………….....26

3.3. Создание физической БД………………………………………………..…29

3.4. Показ страниц приложения……………………………………………..…32

4. Возможности для улучшения приложения……………………………..…..39

Завершение……………………………………………………………….……..40

Список литературы……………………………………………………………..41

Введение

В рамках дипломной работы разработано приложение для студии дизайна интерьера Cascabel. Приложение представляет собой платформу для взаимодействия клиентов и дизайнеров, предлагающих свои услуги.

Приложение может предоставить ряд функций:

1) управление заказами (оформление, прием);

2) портфолио работ, которые загружены дизайнерами и доступны пользователям для ознакомления;

3) система обмена сообщений;

4) система отзывов;

5) система рейтингов;

6) система геймификации, которая была внедрена с целью повышения вовлеченности как клиентов, так и дизайнеров.

Перед началом работы, были изучены материалы с описанием принципов проектирования пользовательского интерфейса, методов защиты данных и т.д.

Приложение разработано с использованием языков программирования и инструментов: HTML, CSS и JavaScript, Node.js, Express, PostgreSQL.

Цель проекта

Целью проекта является создание веб-приложения для студии дизайна интерьера “Cascabel”, которое сможет обеспечить интуитивное и понятное взаимодействие между клиентами и дизайнерами, а также повысить эффективность работы всей студии.

1) Разработка удобного и понятного интерфейса

2) Добавление системы регистрации как для клиентов, так и для дизайнеров

3) Реализация систем управления заказами, портфолио, профилем.

4) Добавление системы геймификации (уровни, ранги, достижения)

5) Обеспечение безопасности данных пользователей.

Основные задачи:

1) Разработка клиентской и серверной части приложения.

2) Разработка архитектуры приложения с использование модели Model-View-Presenter (MVP).

3) Проектирование и разработка базы данных PostgreSQL. С хранением данных о пользователях, сообщениях, заказах и т.д.

1. Изучение предметной области.

1.1. Аналоги. Плюсы и минусы.

На рынке существует множество платформ, представляющих услуги в сфере дизайна интерьеров. Многие из них схожи в той или иной степени, будем рассматривать три аналога:

Houzz.ru. Платформа обеспечивает пользователей большой базой дизайнеров и работ. Практически каждый найдет для себя что-то нужное или интересное.

Плюсы: пользовательская база, мобильное приложение, интерфейс.

Минусы: нет автоматизации управления заказами, персонализация.

RoomPlanner. Приложения для самостоятельного проектирования интерьера. Есть 3D-планировщик, можно создавать планы помещений. Подходит для пользователей кто использует другие работы как ссылку, чтобы сделать что-то свое.

Плюсы: удобный инструмент, много вариантов использования.

Минусы: нет взаимодействия с дизайнерами.

Inmyroom.ru. Платформа, где можно поделиться различными идеями, товарами и большой базой дизайнеров. Можно найти готовые решения дизайнов.

Плюсы: каждый найдет что-то свое, огромное количество различной информации (статьи, изображения).

Минусы: отсутствует автоматизация процессов.

1.2. Область применения приложения

Данное веб-приложение предназначено для автоматизации в студии дизайна интерьера и непосредственное клиентов, и сторонних дизайнеров.

Пользователь может зайти на платформу, ознакомиться с различными работами и заказами других пользователей. Приложение построено так, что пользователя сразу знакомят с различными интерьерами во всевозможных стилях. Это сделано для того, чтобы сразу заинтересовать пользователя.

Система геймификации была внедрена по нескольким причинам: во-первых, это вовлечение пользователей, создание конкуренции между ними; во-вторых, это приносит как прибыль, так и повышает репутацию. За различные достижения можно получать «монеты», которые пользователь может потратить либо на «премиум», либо выставить одну из своих работ на видное место.

Область применения самого приложения можно определить так:

Студии дизайна интерьера - используют платформу для демонстрации своих работ и коммуникации с клиентами

Сторонние дизайнеры - используют платформу для собственного продвижения.

Клиенты - используют как место поиска опытного мастера для выполнения заказов, просмотра портфолио дизайнеров, отслеживания процесса.

1.3. Требования к приложению

Реализация основных функций:

Регистрация и авторизация: регистрация клиента/дизайнера. Авторизация по логину и паролю;

Управление профилем: редактирование личных данных (почта, пароль, аватар). Просмотр различной информации (уровень, заказы, сообщения);

Управление заказами: Клиенты могут создавать заказы в соответствии с их требованиями. Дизайнеры могут отслеживать доступные заказы, принимать их.

Портфолио работ: Дизайнеры могут загружать свои работы, а клиенты просматривать их.

Система отзывов: Клиенты могут оставить отзыв после получения заказа.

Система обмена сообщениями: возможность обмена сообщениями между пользователями.

Система рейтинга: рейтинги отображаются на главной странице, отдельное окно с полным списком рейтинга.

Система геймификации: Система уровней (1-30), рангов (Bronze, Silver, Gold, Platinum) на основе опыта. Система достижений (достижение определенного уровня, выполнение заказов). Опыт дается за ежедневную активность пользователя, выполнение заказов, получения достижений.

Требования к приложению:

Удобный и понятный интерфейс: дизайн с анимациями, адаптивностью для разных устройств

Безопасность данных: шифрование паролей, сообщений

2. Проектирование приложения

Используемые технологии для разработки клиентской и серверной частей приложения.

1) HTML5, CSS3, JavaScript.

2) Node.js, Express, PostgreSQL.

2.1. Архитектура приложения

Серверная и клиентская части приложения разделены:

1) Фронтенд часть приложения на JavaScript, построена по модели MVP (Model-View-Presenter), где:

- Model отвечает за данные, логику, через API взаимодействует с сервером

- View характеризует пользовательский интерфейс, ввод и вывод данных.

- Presenter объединяет Model и View, обрабатывает события.

2) Серверная часть построена на Node.js с использование фреймворка Express, беря архитектуру Rest Api как пример.

Ключевые компоненты серверной части:

- Маршруты

- Контроллеры

Главные части архитектуры:

Клиентская часть отвечает за отображение контента и взаимодействие с пользователями.

Серверная часть отвечает за обработку запросов со стороны пользователей. Работает вместе с БД через библиотеку pg.

База данных содержит в себе информацию о пользователях, заказах и других сущностях.

2.2 Структура приложения:

В корневой папке приложения (cascabel) расположены папки страниц (pages), папка с css файлом (css), серверная часть приложения (server), папка с файлами JavaScript - js (js) и папка с изображениями (img).

Папка js в свою очередь содержит папки и файлы:

1) mains - с файлами \*main.js каждой страницы.

2) models – с файлами \*Model.js каждой страницы.

3) views – с файлами \*View.js каждой страницы.

4) presenters – с файлами \*Presenter.js каждой страницы.

5) файл common.js. – содержит утилитные функции для работы с api и другие.

Папка server содержит папки и файлы:

1) config – содержит с файлом конфигурации db.js.

2) routes – содержит файлы с маршрутами.

3) node\_modules – содержит модули node.js.

4) файл server – файл для работы с сервером

5) файл .env – содержит данные для подключения к БД PostgreSQL. (логин, пароль, порт, хост).

6) package.json – содержит скрипты для работы с node.js.

7) package-lock.json.

2.3. Макет приложения

Веб-приложение построено на языках HTML, CSS, JavaScript. HTML страницы содержат контент приложения, CSS отвечает за стилизацию и анимацию, JS отвечает за обработку действий пользователя.

1) Страницы приложения:

«Главная страница» (index.html): отображает последние заказы, рейтинг пользователей, информацию о студии.

«Работы» (works.html): страница для просмотра портфолио дизайнеров.

«Оформить заказ» (order.html): содержит форму для оформления заказа.

«Список заказов» (orderList.html): на странице показаны активные, доступные и завершенные заказы.

«Личный кабинет» (profile.html): страница с информацией пользователя. Просмотре заказов, сообщений, достижений.

«Отзывы» (reviews.html): страница с формой для написания отзыва. Так же возможен просмотр чужих отзывов.

«Авторизация» (auth.html): страница с формой регистрации и авторизации.

«Панель администратора» (admin.html).

2) Алгоритмы основных функций прописаны в JavaScript:

Регистрация пользователя:

-Проверка введенных данных (имя, почта, пароль) на валидность

-Вставка записи в таблицу Users

Создание заказа:

-Проверка роли пользователя (только клиенты могут создавать заказы)

-Валидность данных (требования, размер, цена)

-Вставка записи в таблицу Orders со статусом Open

-Возврат данных заказа клиенту

-Подтверждение заказа дизайнером:

-Проверка роли пользователя (только дизайнеры могут принимать заказы), принятие заказа (designer\_id)

-Проверка текущего статуса заказа («In progress»)

-Обновление статуса на «Under Review» и определение времени

-Уведомление клиента

-Начисление опыта и достижения

-После завершение заказа начислить опыт клиенту и дизайнеру

-Проверить, достиг пользователь нового уровня или нет

-Проверить, достиг ли пользователь необходимого количества выполненных заказов для получения достижения

-Если количество достигнуто, добавить запись в таблицу.

3) Алгоритм действий пользователя

Сценарий для пользователя «Клиент»

Регистрация / вход: перейти на страницу «Авторизация», выбрать регистрацию, указать роль «Клиент», заполнить данных. Войти с логином и паролем.

Просмотр загруженных работ и рейтинга: на странице работ, просматривать различные загруженные проекты дизайнеров. На главной странице ознакомимся с рейтингом дизайнеров. Выберем подходящего нам дизайнера.

Создание заказа: перейти на страницу оформления заказа. Заполнить форму (требования, параметры, бюджет). Подтвердить создание заказа.

Отслеживание статуса заказа: перейти в профиль, нажимаем заказы. Проверяем статус заказа. При получении уведомления о выполнении дизайнером заказа, подтвердить получение.

Сценарий для пользователя «Дизайнер»

Регистрация / вход: перейти на страницу «Авторизация», выбрать регистрацию, указать роль «Клиент», заполнить данных. Войти с логином и паролем.

Загрузка работ: загрузка своих работ на платформу.

Просмотр заказов: ознакомление со списком заказов

Принятие заказа: принимаем заинтересовавший пользователя заказ

Общение с клиентом: заходим в профиль, далее почта, пишем сообщение клиенту. Уточняем вопросы по поводу заказа.

Выполнение заказа: После выполнение заказа, переходим в профиль, далее заказы, найдем в списке нужный нам заказ и подтверждаем выполнение. После чего загружаем работу, ждем ответа от «клиента».

Для наглядности, построена UML-схема возможных действий пользователей. На рис. 1 схема действий «клиента», на рис. 2 – «дизайнера».

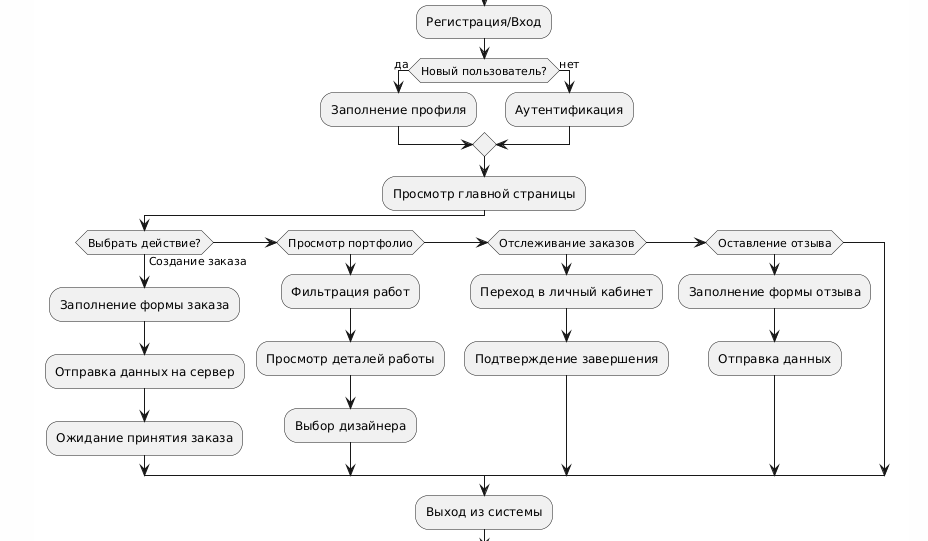


Рис. 1

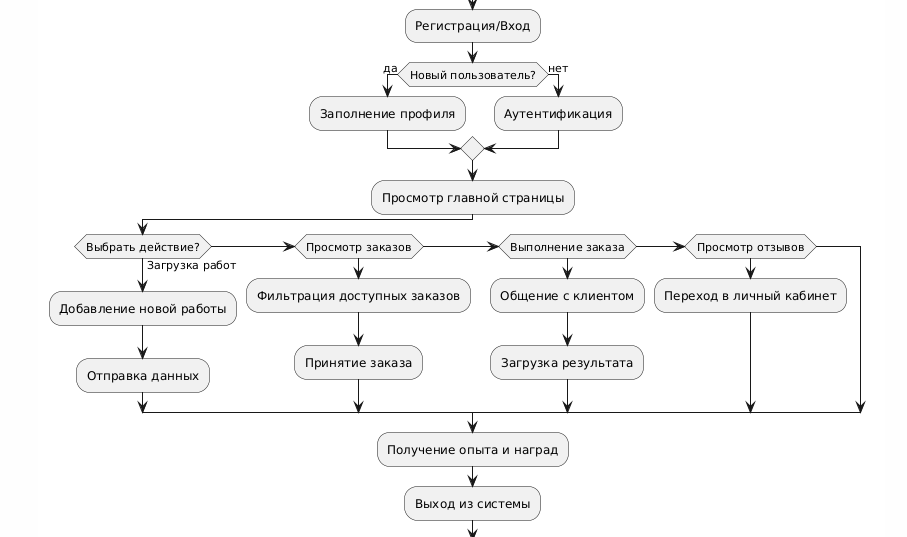


Рис. 2

Пример регистрации пользователя:

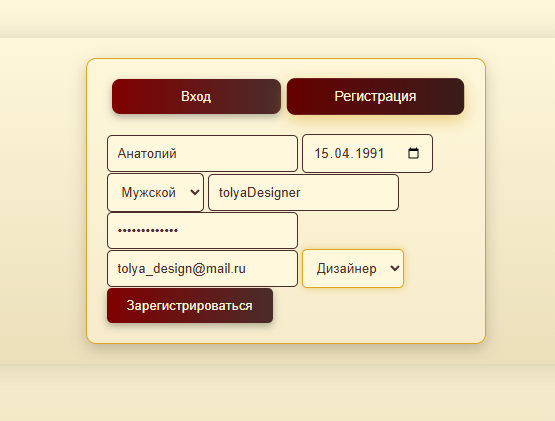


Рис. 3

Пример авторизации пользователя:

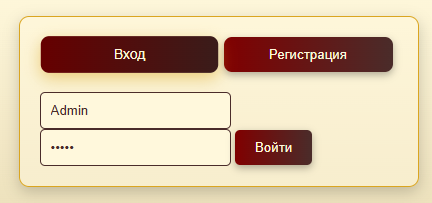


Рис. 4

После успешной авторизации переносит на главную страницу:

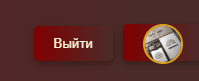


Рис. 5

2.4. База данных

Для хранения данных приложения использована база данных PostgreSQL. Разрабатывается база данных использованием postgre4.

1) Концептуальное проектирование

Базовые сущности: (описать каждую сущность).

Пользователи (users) – содержит данные о всех пользователях.

Заказы (orders) – данные о заказе, статус, личности дизайнера и клиента.

Работы (works) - портфолио дизайнеров.

Отзывы (reviews) – оценки и комментарии клиентов

Сообщения (messages) – содержит данные о сообщениях между пользователями

Стили (styles) – данные о стилях интерьера

Услуги (services) – содержит информацию о типах услуг.

Атрибуты для каждой сущности:

Пользователи:

id – уникальный id. (первичный ключ);

name – имя пользователя, уникальное;

email – почта пользователя, контактные данные, уникальная;

password – ключ к профилю;

role – дизайнер или клиент, также записан администратор;

level – уровень пользователя, от 1 до 30 уровня;

exp – текущее количество опыта;

rank – звание пользователя: «Bronze, Silver, Gold, Platinum»;

avatar – ссылка на изображение профиля пользователя;

current\_orders – текущие заказы, массив;

completed\_orders – завершенные заказы, массив.

Заказы:

id – уникальный id. (первичный ключ);

client – имя клиента (внешний ключ на user.name);

requirements – требования пользователя относительно заказа;

size – размер места ремонта;

price – цена;

status – состояние заказа, выполнен, выполняется, на рассмотрении или доступен;

designer – имя дизайнера (внешний ключ на user.name);

created\_at – время создания;

taken\_at – время получения заказа;

under\_at\_review\_at – время отправки клиенту на проверку;

completed\_at – время завершения заказа.

Отзывы:

id – уникальный id (первичный ключ);

author – имя автора (внешний ключ на user.name);

text – текст отзыва;

date – дата написания;

avatar – изображение аватара автора;

order\_id – id заказа;

rating – оценка пользователями работы дизайнеров, от 1 до 5 звезд;

target – имя дизайнера (внешний ключ на user.name).

Работы:

id – уникальный id (первичный ключ);

title – название работы;

image – ссылка на изображение работы;

style – стиль работы;

type – тип работы;

uploader – имя дизайнера, автор (внешний ключ на user.name).

Стили:

id – уникальный id (первичный ключ);

name – название стиля;

description – текст с описанием стиля;

image – ссылка на изображение.

Услуги:

id – уникальный id (первичный ключ);

type – тип услуги;

styles – массив стилей;

sub\_services – массив под-услуг.

2) Логическое проектирование

Модель сущность-связь.

Один пользователь может создавать множество заказов как клиент (1:М через client в orders);

Один пользователь может принять множество заказов как дизайнер (1:М через designer в orders);

Один дизайнер может загрузить много работ (1:N через uploader) ;

Один заказ может иметь только один отзыв (reviews) (1:1 через order\_id).

Назначение ключей

Первичные ключи: id для всех таблиц.

Внешние ключи: ссылки на пользователя в таблицах: отзывы, заказы, работы, сообщения.

3. Реализация практической части

3.1. Клиентская часть

1) Написание HTML кода для страниц

Код главной страницы:

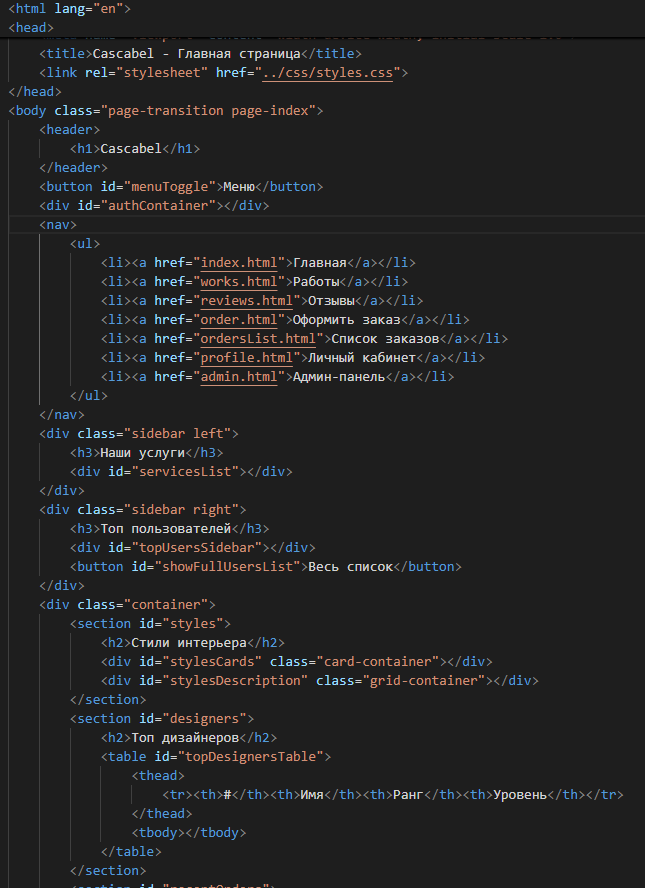


Рис. 6

 Рис. 7

На главной странице, помимо общих со всеми страницами header, footer и кнопки «Меню», расположена информация об услугах, карточки с различными стилями дизайна, рейтинг пользователей, последние выполненные заказы.

Footer содержит контактную информацию. Кнопка «меню» позволяет убирать навигационную панель, когда она больше не нужна. Весь контент будет загружаться из базы данных, необходимый код прописан в homeView.js страницы.

Далее идет страница авторизации.

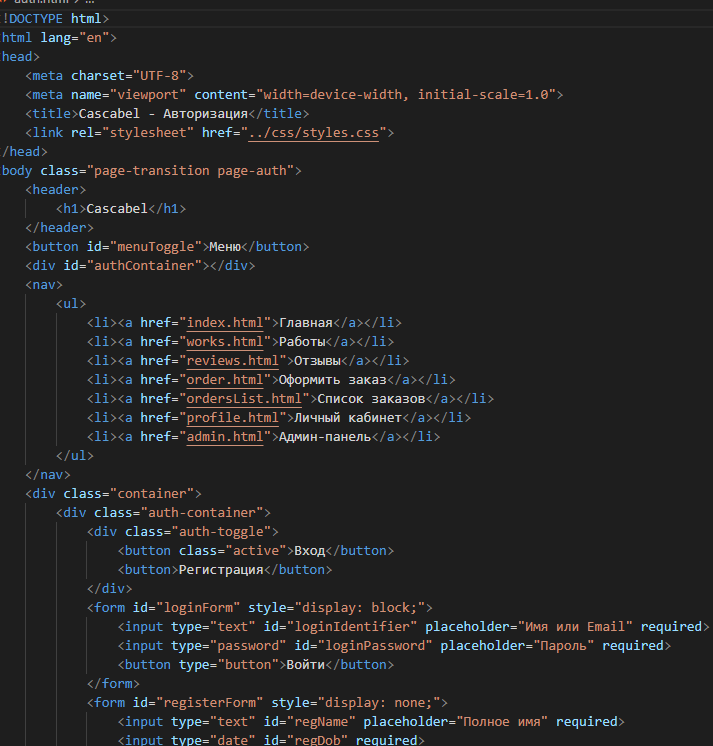


Рис. 8

Как и во всех файлах страниц, у них общая часть header, навигационная панель с кнопками переходов на другие страницы, кнопка «Меню», footer и кнопка «Вверх» позволяющая быстро перемещаться в верхнюю часть экрана при необходимости.

Страница оформления заказов:

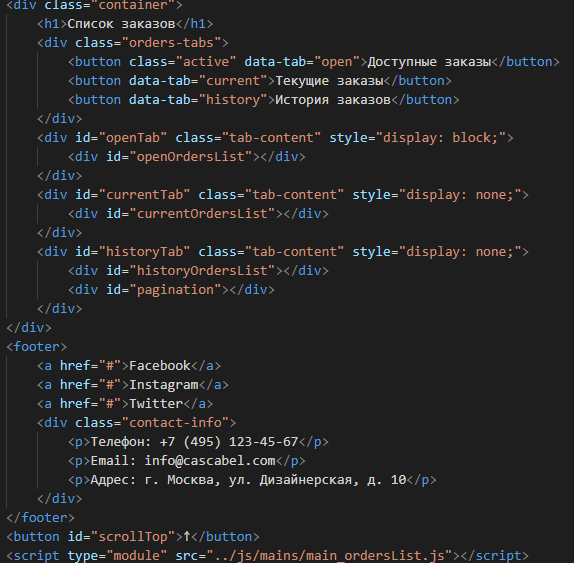


Рис. 9

Страница со списком заказов:

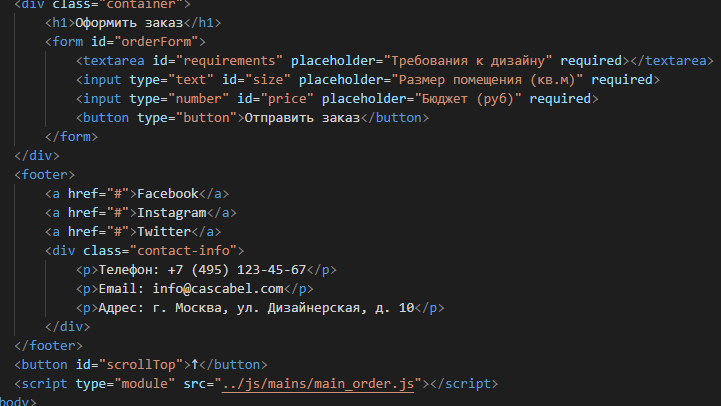


Рис. 10

Страница с работами (портфолио):

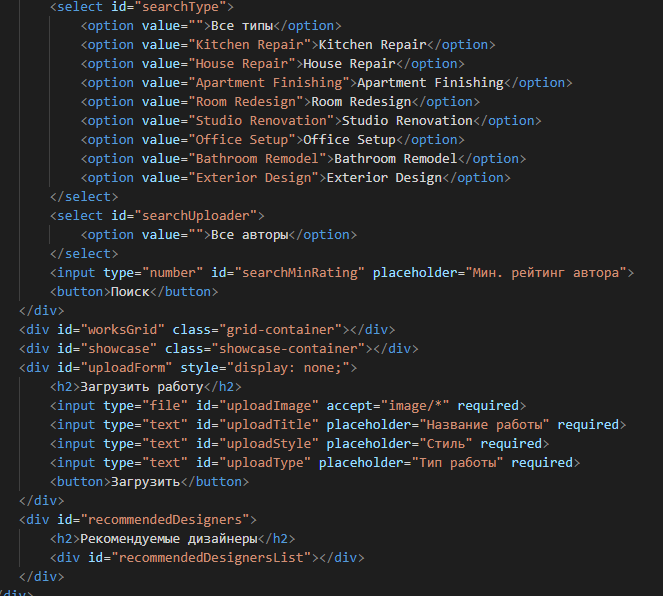


Рис. 11

Страница личного кабинета:



Рис. 12

Продолжение страницы личного кабинета:



Рис. 13

Страница с отзывами:

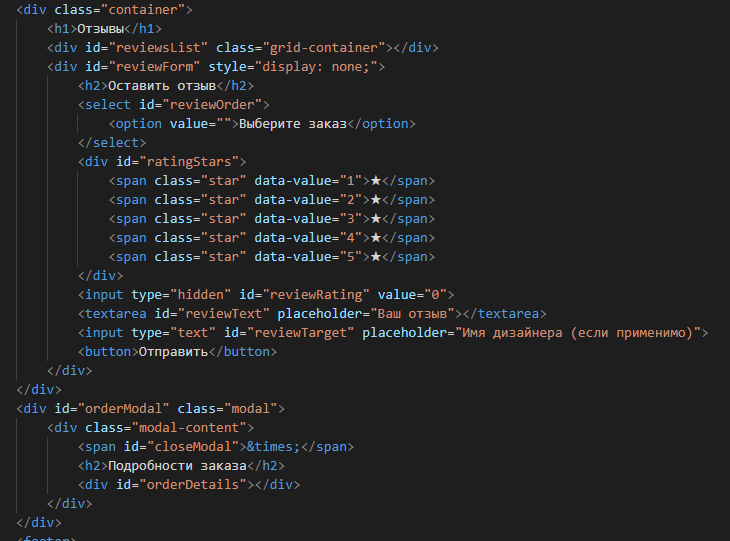


Рис. 14

2) Код CSS

Styles.css содержит в себе стили и анимации всего приложения.

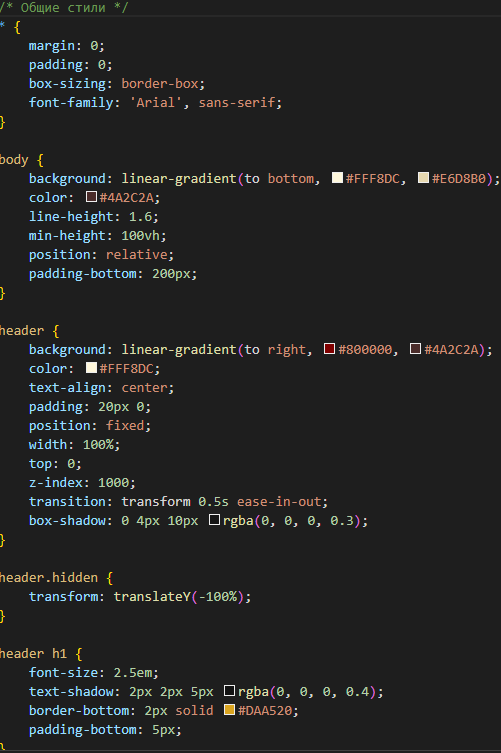
Фрагмент кода css:

Рис. 15

Здесь показаны стили для всего приложения: отступы, цвета, анимации.

В первом блоке обнуляются все отступы, устанавливается шрифт. В следующих блоках прописан изначальный код для всех body и header контейнеров: устанавливаются цвета, расположение блока, оформление текста и анимация с задержкой в 0.5 секунд.

3) Код JavaScript

У каждой страницы есть свой main.js файл, через который подключаются все необходимые функции. Следуя модели MVP, каждая страница имеет файл model.js, view.js, presenter.js, которые в свою очередь подключены к main.js своей страницы.

Фрагмент кода homeModel.js:



Рис. 16

Фрагмент кода homeModel.js:



Рис. 17

Фрагмент кода common.js, где прописана функция, возвращающая результат повышения уровня пользователя:

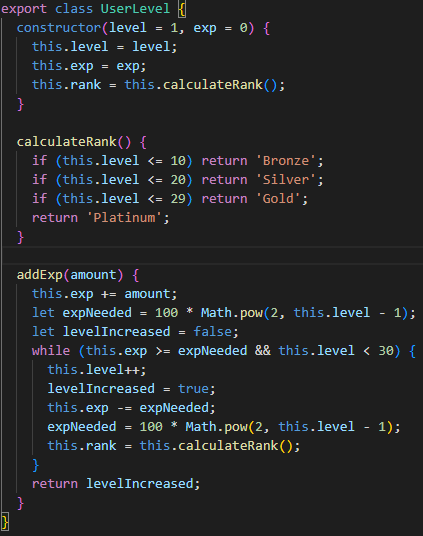


Рис. 18

Один из main.js файлов, главная страница:

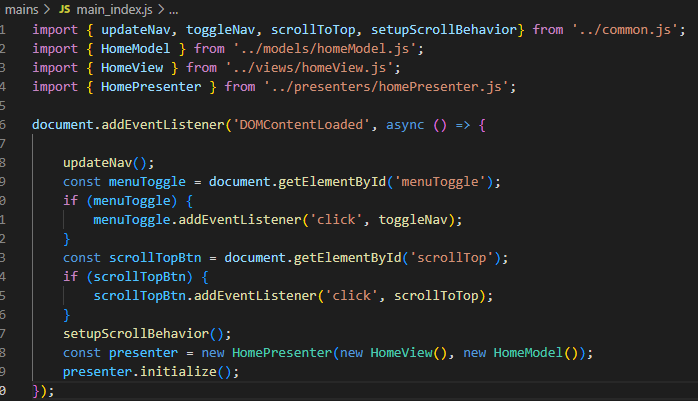


Рис. 19

Здесь подключены все три файла модели MVP, а также общие функции приложения из файла common.js.

3.2. Серверная часть

Серверная часть разрабатывается на Node.js.

В корневой папке приложения, создаем папку server. Далее через терминал переходим в папку server, подключаем node.js и подключаем нужные модули, при помощи команд «npm init -y» и «npm install express pg cors dotenv».

После установления необходимых файлов, создаем файл «.env» в папке сервера, где будут храниться данные о сервере и базе данных PostgreSQL (рис. 9).

Далее создадим файл server.js, с которого и будет запускаться сервер.

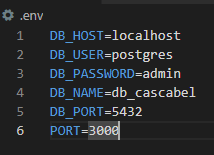


Рис. 20

Создадим файл конфигурации db.js, и поместим его в новую папку config в папке сервера.

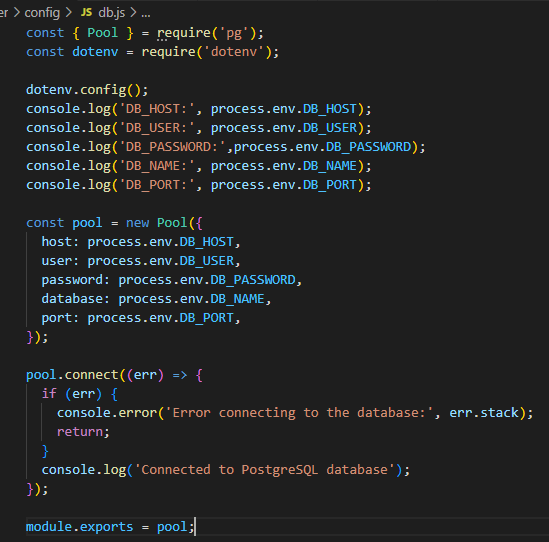


Рис. 21

Чтобы запросы со стороны клиента обрабатывались, создадим маршруты.

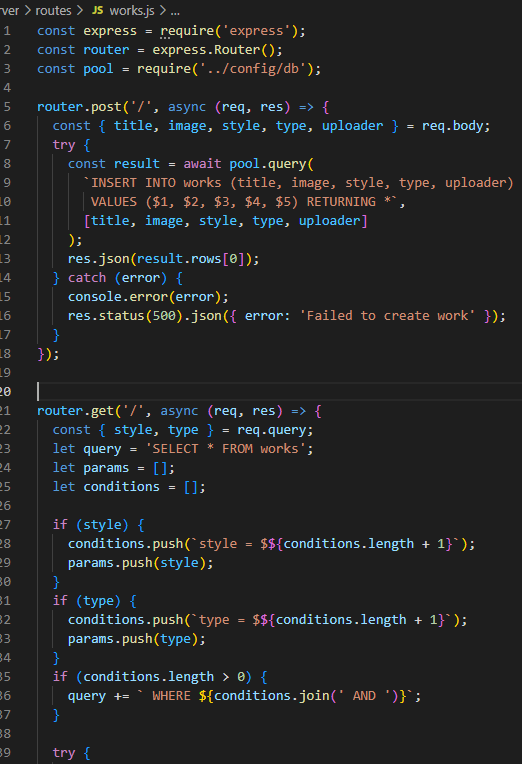


Рис. 22



Рис. 23

Можно получить обработку данных: загрузка новой работы, получение списка всех работ, определение определенной работы по id.

В файле server.js пропишем код для подключения маршрутов:



Рис. 24

Запуск сервера делается через терминал. Сначала переходим из корневой папки в папку сервера:



Рис. 25

Далее запускаем сервер и подключаемся:



Рис. 26

Теперь у нас запущены сервер и база данных:

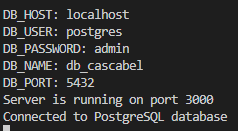


Рис.27

3.3. Создание физической БД

Создание физической БД

База данных будет спроектирована на языке SQL, и использоваться инструмент postgre4 для реализации.

Перед тем как приступить к работе с базой данных, необходимо создать сервер, запомнить логин и пароль, так как далее будем подключаться к базе данных именно через логин и пароль. Помимо логина и пароля есть хост и порт. После создания сервера, создаем базу данных. После ее создания, начинаем создавать таблицы сущностей. (Рис. 27 – Рис. 29).

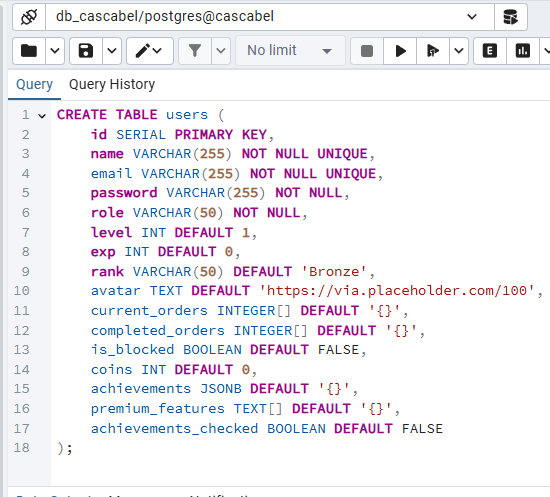


Рис. 28

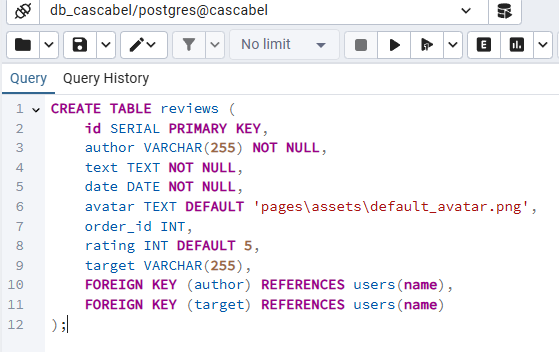


Рис. 29

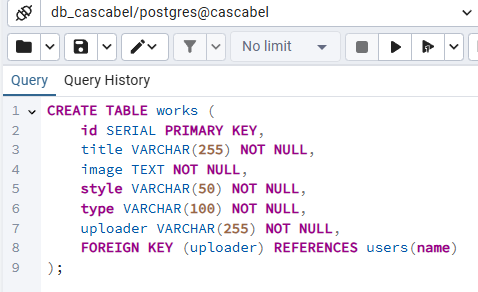


Рис. 30

Схожим способом созданы и остальные таблицы: заказы, стили, услуги, сообщения.

Следующий шаг, наполнить таблицы данными через команду INSERT INTO. (Рис. 30 – Рис. 32).

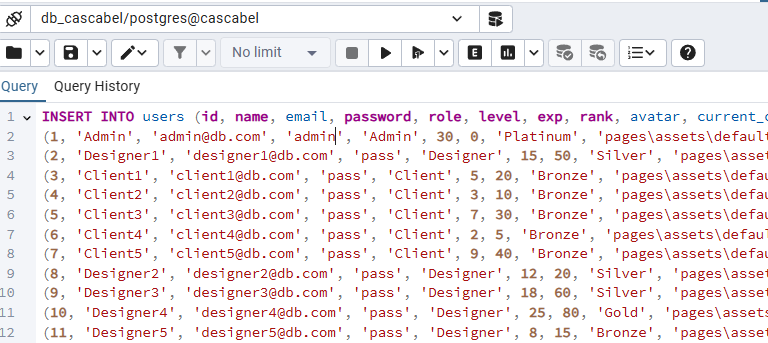


Рис. 31

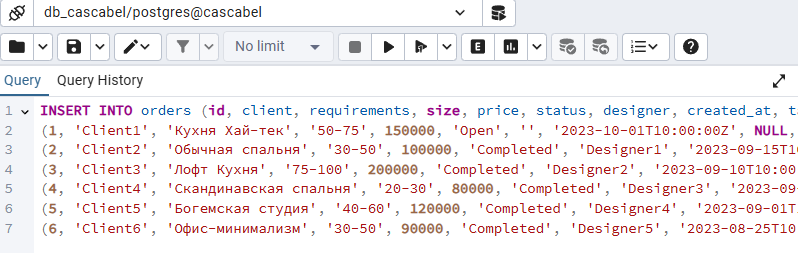


Рис. 32

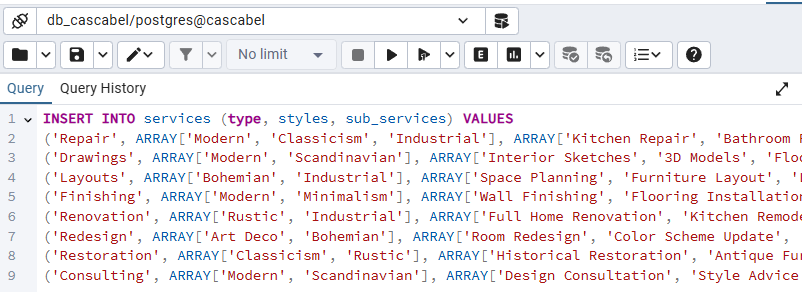


Рис. 33

Таким же образом заполняем остальные таблицы сущностей.

3.4. Демонстрация страниц приложения.

1) Главная страница. (Рис. 34 – Рис. 37).



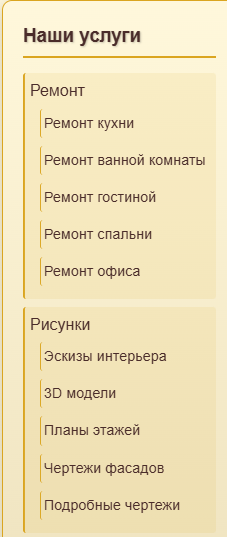
Рис. 34

Рис. 35 Рис. 36

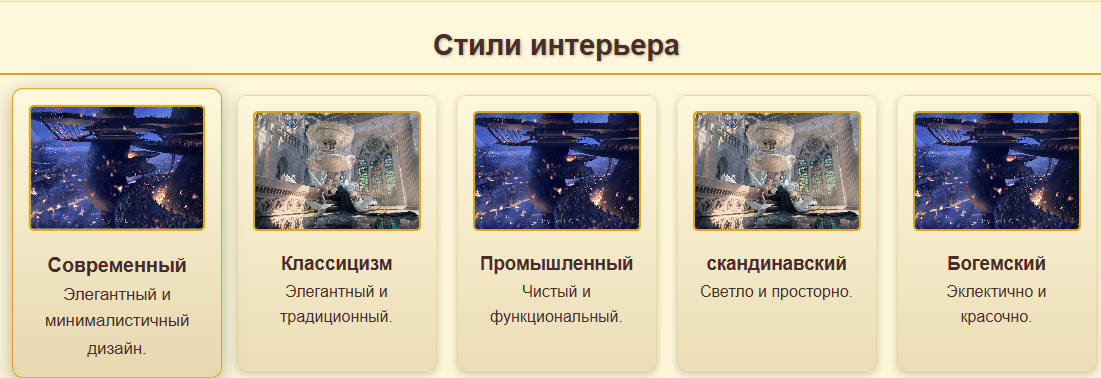


Рис. 37

На этих изображениях показаны части главной страницы приложения. Рейтинговые таблицы, Стили интерьера, а также услуги, которые дизайнеры платформы могут предоставить.

Рейтинговая таблица дизайнеров на рис. 34, показывает пользователей с самым высоким уровнем. Реализованы модальные окна, на которых выводится информация о пользователях при нажатии на имя в списке.

На рис. 35, показана часть боковой панели со списком услуг, которые клиент может получить на платформе. От ремонта крыльца и подвала, до проектировки и строительства дома.

Рейтинговая таблица на рис. 36 также показывает пользователей с самым высоким уровнем, вдобавок к этому, выделяет 3х лучших окантовкой и статусом «премиум». При нажатии на кнопку весь список, всплывает модальное окно со списком всех пользователей на сайте (Рис. 38), на каждой странице по 10 пользователей.

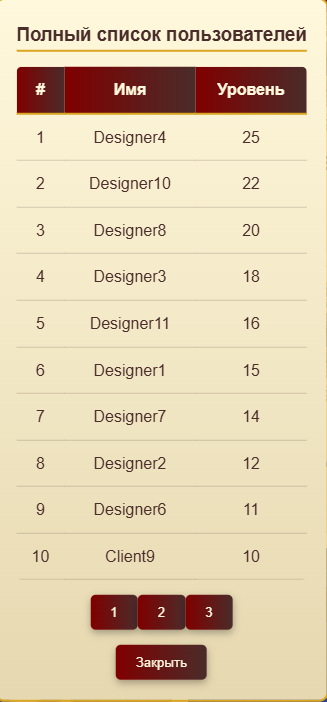


Рис. 38

На рис. 37 показан блок со стилями дизайна. При обновлении страницы эти стили случайным образом отображаются на странице.

2) Страница «Работы», с портфолио дизайнеров (рис. 39 – Рис. 41).

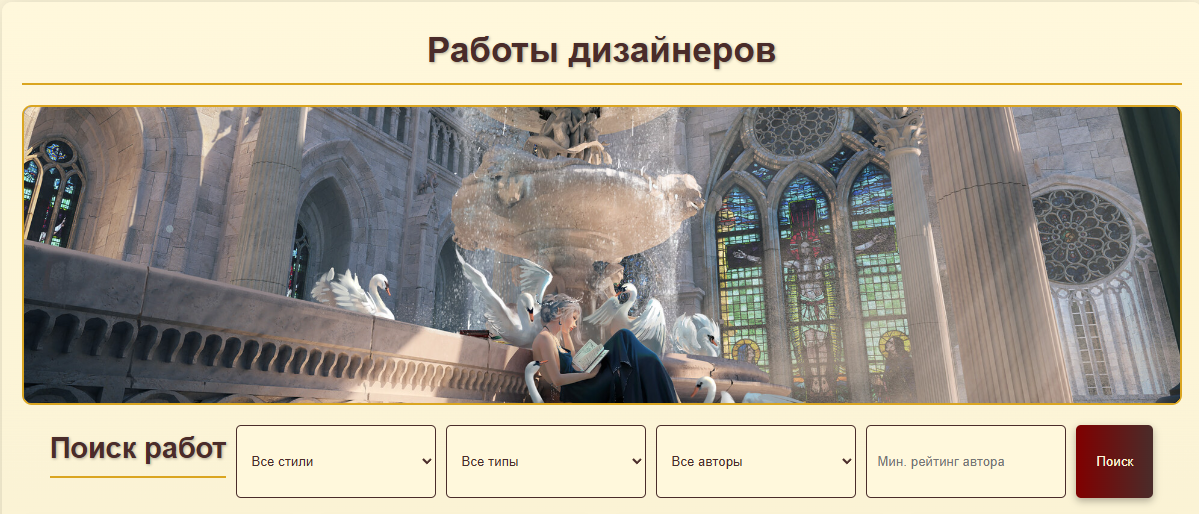


Рис. 39

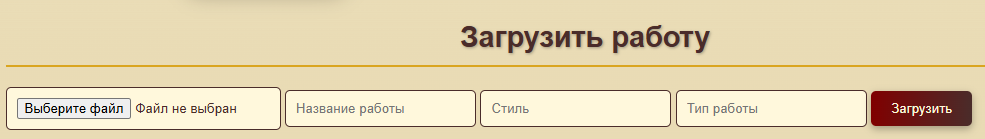


Рис. 40

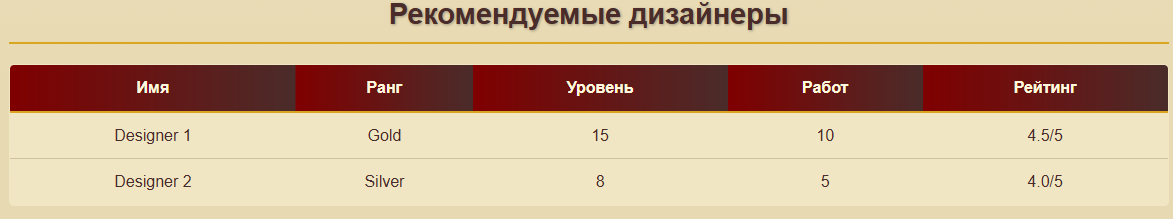


Рис. 41

На рис. 39 показан блок, в котором размещена форма поиска. Есть возможность делать поиск сортируя по стилям, типам, а также выбрать автора, работы которого, возможно заинтересовали пользователя.

На рис. 40 форма загрузки файла. После загрузки изображения, оно будет доступно к просмотру только самому пользователю (дизайнеру), т.к. после загрузки оно должно пройти проверку.

На рис. 41 контейнер с рекомендуемыми дизайнерами. В этот список может попасть любой. Так как, чтобы попасть туда нужен либо «премиум», высокий рейтинг или большое количество загруженных и отобранных работ.

3) Страница «Личный кабинет» (Рис. 42 – Рис. 45).

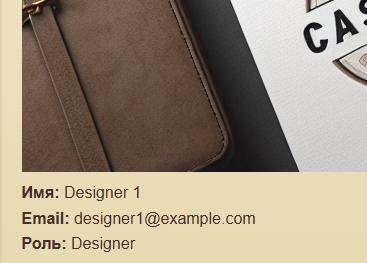


Рис. 42

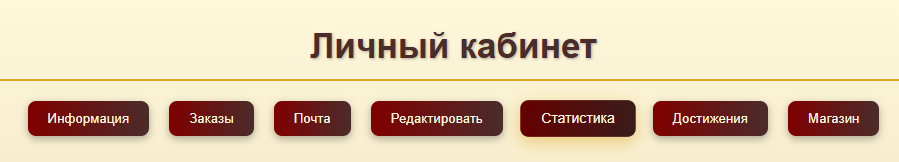


Рис. 43

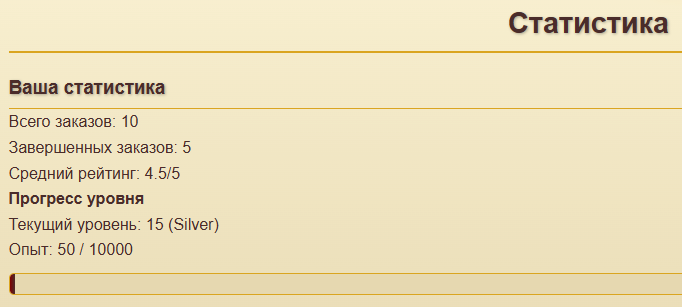


Рис. 44

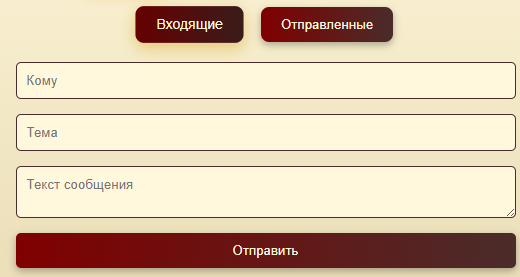


Рис. 45

На рис. 42 показан профиль пользователя, который выводит пользователю общую информацию о нем.

На рис. 43 отображена панель с кнопками: Информация, Заказы, Почта, Редактирование профиля (изменение пароля, аватара), Статистика, Достижения, Магазин. Кнопка «Информация» приведет нас к главному окну личного кабинета, показанному на рис. 42; кнопка «Заказы» показывает пользователю список заказов, которые он принял или выполнил; кнопка почта, откроет страницу, где можно просмотреть входящие сообщения или же отправить его самому (рис. 45); кнопка редактирования позволяет перейти на страницу, где можно изменить некоторые данные о себе; кнопка статистика, показывает более подробную информацию, чем на странице «Информация»; при переходе на страницу «Достижения», будет показан список достижений, если достижений нет, то и список будет пуст. Кнопка магазин, в котором продаются различные товары, которые добавят визуальный эффект на главный профиль пользователя, или изменит цветовую гамму приложения.

На рис. 44 показана страница «Статистика», куда выводится подробная информация о пользователе, о том сколько заказов было выполнено, или сколько осталось опыта до следующего повышения уровня.

На рис. 45 показана форма отправки сообщения. Для того, чтобы отправить сообщение, нужно ввести тему и текст сообщения, а также имя(логин) пользователя, которому пользователь хочет написать.

4) Страница «Оформление заказов»

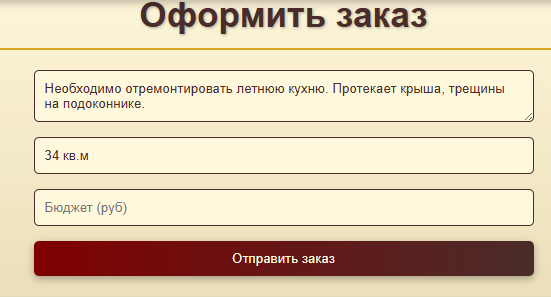


Рис. 46

На странице помимо формы оформления заказа больше ничего нет, но она одна из самых важных частей всего приложения. Требования со своей сторону лучше раскрыть более подробно, т.к. чем понятнее будет дизайнеру что нужно сделать, тем быстрее за ваш заказ примется дизайнер. Необходимо корректно рассчитать бюджет и размер помещения.

5) Страница «Отзывы»

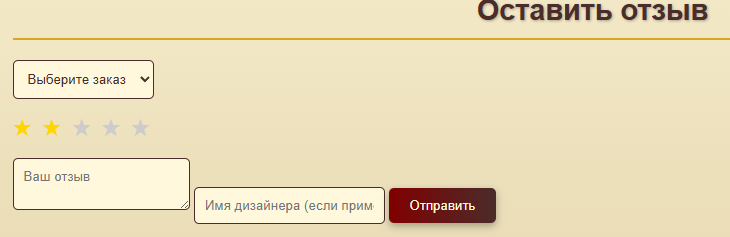


Рис. 47

Чтобы оставить отзыв, необходимо быть «клиентом», завершить хотя бы 1 заказ, т.к. для того чтобы оставить отзыв необходимо выбрать к какому именно заказу адресуется отзыв. Помимо формы заполнения отзыва, есть так же блок с отзывами других пользователей.

6) Страница «Авторизация» (Рис. 48 – Рис. 49).

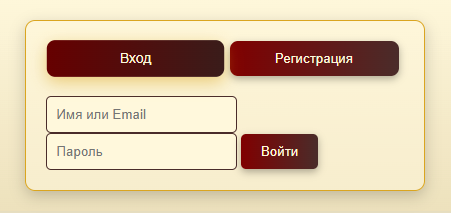


Рис. 48

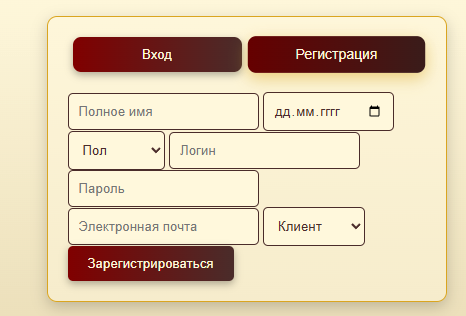


Рис. 49

На странице авторизации находится форма с регистрацией / авторизацией. При нажатии на соответствующую кнопку, форма изменится с авторизации на регистрацию, и наоборот. После входа в профиль, пользователя отправляют на главную страницу. После регистрации пользователь переходит на страницу личного кабинета. Если пользователь специально, или же непреднамеренно попадет на страницы с ограниченным уровнем доступа: как панель администратора, форма оформления заказа (доступна только клиентам) или же как страница списка заказов (доступна только дизайнерам), то его сразу отправят на страницу авторизации.

4. Возможности для улучшения приложения.

Данное приложение имеет большой потенциал для развития:

1) Добавить платежную систему прямо в приложение.

2) Разработать мобильное приложение.

3) Если база клиентов и дизайнеров будет достаточно большой, платформу можно превратить в полноценную гильдию.

4) Подключение инструментов для проведения аналитики, что сделало бы отслеживание за данными более удобным.

5) Расширить функции приложения.

6) Увеличить количество элементов системы геймификации.

7) Улучшить систему шифрования.

Заключение

В ходе выполнения дипломной работы было разработано веб-приложение для студии дизайна интерьера «Cascabel». Разработана как клиентская, так и серверная часть.

В ходе работы все поставленные задачи были выполнены, а разработанное приложение полностью соответствует целям разработки.

Приложение представляет из себя удобный инструмент для автоматизации процессов взаимодействий клиентов и дизайнеров, а также для привлечения новых клиентов и товарищей.

Список литературы

1. Джон Дакетт: HTML и CSS: Разработка и дизайн веб-сайтов — 2020 год — 480 страниц.

2. Марейн Хавербеке: Выразительный JavaScript — 2019 год — 480 страниц.

3. Кристофер Шмитт: CSS рецепты программирования — 2011 год — 672 страницы.

4. Дуглас Крокфорд: JavaScript: cильные стороны — 2012 год — 163 страницы.

5. Бен Фрэйн HTML5 и CSS3: Разработка сайтов для любых браузеров и устройств — 2014 год — 273 страницы.

6. Аарон Густафсон: Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement — 2015 год — 264 страницы.

7. Дэвид Флэнаган: JavaScript. Подробное руководство — 2013 год — 386 страниц.

8. Дэвид Сойер Макфарланд: JavaScript и jQuery. Исчерпывающее руководство — 2015 год — 880 страниц.

9. Эрик Фримен, Элизабет Фримен: изучаем HTML, XHTML и CSS — 2014 год — 720 страниц.

10. Кит Грант: CSS для профи — 2019 год — 496 страниц.