



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Приложение «Магазин компьютерных комплектующих»

Студент: Чибиток Даниил Сергеевич

Группа: ИКБО-24-21

Работа представлена к защите _____ (дата) _____ / Чибиток Д.С. /
(подпись и ф.и.о. студента)

Руководитель: старший преподаватель кафедры ИиППО Рачков Андрей Владимирович

Работа допущена к защите _____ (дата) _____ / Рачков А.В. /
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/ _____ /
_____/ _____ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине: Шаблоны программных платформ языка Джава
по профилю: Разработка программных продуктов и проектирование информационных систем
направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Чибиток Даниил Сергеевич

Группа: ИКБО-24-21

Срок представления к защите: 11.05.2023 г.

Руководитель: старший преподаватель Рачков Андрей Владимирович

Тема: Приложение «Магазин компьютерных комплектующих»

Исходные данные: индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java (версия не ниже 8); инструменты и технологии: JDK (не ниже 8), создание Spring MVC web-приложений, RESTful web-сервисов, Spring ORM и Spring DAO, Gradle, gitHub, IntelliJIDEA. ГОСТ 7.32-2017. Отчет о научно-исследовательской работе. Структура и правила оформления, Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

Перечень вопросов, подлежащих разработке, и обязательного графического материала:

1. Провести анализ предметной области и формирование основных требований к приложению, 2. Обосновать выбор средств ведения разработки. 3. Разработать приложение с использованием фреймворка Spring, выбранной технологии и инструментария. 4. Провести тестирование приложения. 5. Оформить пояснительную записку по курсовой работе в соответствии с ГОСТ 7.32-2017. 6. Провести анализ текста на антиплагиат 7. Создать презентацию по выполненной курсовой работе.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: [подпись] /Р. Г. Болбаков/, « 20 » 02 2023 г.

Задание на КР выдал: [подпись] / А.В. Рачков/, « 20 » 02 2023 г.

Задание на КР получил: [подпись] /Д. С. Чибиток/, « 20 » 02 2023 г.

РЕФЕРАТ

Курсовой проект содержит 35 страницы отчета, 31 иллюстрацию, 1 таблицу и 18 использованных источников.

ПРИЛОЖЕНИЕ, ВЕБ-ПРИЛОЖЕНИЕ, МАГАЗИН КОМПЬЮТЕРНЫХ КОМПЛЕКТУЮЩИХ, JAVA, SPRING FRAMEWORK, POSTGRESQL

Объектом разработки является приложение магазина компьютерных комплектующих.

Цель работы – создание веб-приложения.

В процессе выполнения работы проводился анализ сайтов-аналогов, предоставляющих услуги по продаже компьютерных комплектующих.

В результате разработки создано приложение по теме «Магазин компьютерных комплектующих», удовлетворяющий современным требованиям и решающий актуальные проблемы.

Область применения результатов – информационные ресурсы.

REPORT

The course project contains 35 pages of the report, 31 illustrations, 1 table and 18 sources used.

APPLICATION, WEB APPLICATION, COMPUTER COMPONENTS STORE, JAVA, SPRING FRAMEWORK, POSTGRESQL

The object of development is the application of the computer components store.

The purpose of the work is to create a web application.

In the course of the work, an analysis of analog sites providing services for the sale of computer components was carried out.

As a result of the development, an application was created on the topic "Computer Components Store" that meets modern requirements and solves current problems.

The scope of the results is information resources.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ	7
1 ОБЩИЕ СВЕДЕНИЯ	8
1.1 Наименование клиентской части интернет-ресурса	8
1.2 Функциональное назначение интернет-ресурса	8
1.3 Прикладное ПО, используемое для разработки веб-страниц.....	13
2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	13
2.1 Анализ предметной области.....	13
2.2 Выбор технологий разработки	16
2.3 Разработка приложения	16
2.3.1 Структура проекта.....	16
2.3.2 База данных.....	18
2.3.3 Авторизация и регистрация	33
2.3.4 Работа с запросами	35
2.3.5 JWT токен	36
3 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ	37
3.1 Тестирование доступности страниц	37
3.2 Тестирование запросов	41
ЗАКЛЮЧЕНИЕ	33
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	35

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В настоящем отчете применяют следующие сокращения с соответствующими определениями:

API	– Программный интерфейс некоторой системы. Используется для взаимодействия с другими системами через программный код
Backend	– Разработка бизнес-логики продукта, отвечает за взаимодействие пользователя с внутренними данными, которые потом отображает frontend
CSS	– Cascading Style Sheets (формальный язык описания внешнего вида документа, написанного с использованием языка разметки)
Frontend	– Разработка пользовательского интерфейса и функций, которые работают на клиентской стороне веб-сайта или приложения
Java	– Строго типизированный объектно-ориентированный язык программирования общего назначения
HTML	– HyperText Markup Language (стандартизированный язык разметки документов для просмотра веб-страниц в браузере)
React	– JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов
Spring	– Универсальный фреймворк с открытым исходным кодом для Java-платформы, который используют для создания корпоративных приложений

TypeScript	– Язык программирования, позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript
UI	– Дизайн пользовательского интерфейса, то есть создание визуальной части приложения или сайта: экранов, кнопок, иконок
URL	– Система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса

ВВЕДЕНИЕ

За прошедшее время тенденция развития информационных технологий и внедрение их в повседневную жизнь человека перешла на новые границы. То, что раньше считалось фантастикой, в данный момент является неотъемлемой реальностью нашего мира. Этот процесс уже необратим, у людей появляются все более новые и новые потребности, а большие компании находят способы им соответствовать. Этим они подталкивает данную область на дальнейшее развитие.

Сузим круг рассматриваемых информационных технологий на таких приложениях как интернет-магазины. Можно проследить ярко-выраженный темп развития данной области. Данный «бешенный» темп очень легко объясним, так как именно данная область позволяют превратить рутинное занятие поиска и покупки вещей во что-то быстрое, привлекательное и интересное. Если говорить более скованно, то упрощать жизнь всех людей, без исключений, ведь всем нам так или иначе нужно ходить в магазин за продуктами и тому подобное.

Целью данного проекта курсовой работы является разработка приложения на тему «Магазин компьютерных комплектующих» с использованием основных современных технологий HTML5 [1], CSS3 [2], React [3], Java [4], TS [5] и Spring [6].

Для достижения цели необходимо решить несколько задач:

- 1) провести анализ предметной области разрабатываемого приложения и сформировать основные требования к приложению;
- 2) с учетом проведенного анализа разработать приложение с использованием фреймворка Spring, выбранной технологии и инструментария;
- 3) провести тестирование приложения.

В ходе выполнения курсовой работы используются методы наблюдения, сравнения, анализа, описания, и моделирования.

Информационной базой в данной работе являются знания, полученные в ходе выполнения практических работ в течение курса «Шаблоны программных платформ языка Джава», а также интернет-источники.

Структура курсовой работы имеет следующий вид:

- 1) введение;
- 2) три основных раздела, среди которых: общие сведения, описание логической структуры, тестирование приложения;
- 3) заключение;
- 4) приложения.

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование клиентской части интернет-ресурса

Интернет-ресурс называется «CCS», что означает – COMPUTER COMPONENTS STORE, в переводе с английского «Магазин компьютерных комплектующих». Это напрямую связано с тематикой сайта.

1.2 Функциональное назначение интернет-ресурса

Интернет-ресурс «CCS» создан с целью продажи компьютерных ресурсов в удобном формате онлайн.

Главная страница содержит карточки товаров комплектующих. При нажатии на любую из карточек открывается расширенная информация о данном товаре. Товар можно добавить в корзину как с главной страницы, так и при переходе на расширенную карточку.

Шапка сайта имеет несколько страниц: главная, о ресурсе, доставка, обратная связь. Также расположены кнопки для быстрой регистрации и входа пользователя, иконка корзины и поисковая строка.

Пользователь может добавлять товары в корзину, даже если он не зарегистрирован, но оформить заказ сможет только после регистрации или, если учетная запись уже существует, после входа в аккаунт. При переходе по иконке корзины пользователь попадает на страницу с добавленными в корзину товарами. Товары отображаются в виде списка с изображением,

описанием и количеством товаров. Также отображается общее количество позиций и сумма выкупа. В корзине можно оформить заказ, нажав на кнопку «Перейти к оформлению заказа».

Страница входа включает в себя форму, в которую нужно ввести уникальное имя пользователя и пароль. Если пользователь введет корректные данные, то он будет перенаправлен на страницу личного кабинета, в противном случае он увидит ошибку.

Страница регистрации реализована с помощью слайдера, чтобы уменьшить количество визуального контента на странице. Регистрация разделяется на два типа: для покупателей и для поставщиков. Выбор типа регистрации осуществляется путем нажатия на checkbox. После успешной регистрации пользователь автоматически переходит на страницу личного кабинета.

Наполнение личного кабинета отличается и зависит от типа пользователя. Если пользователь является покупателем, то в личном кабинете находится один выпадающий список для выхода из системы и две вкладки: настройки и история. Во вкладке настройки пользователь может поменять свои данные, а во вкладке история он может посмотреть историю всех своих покупок в виде таблицы.

Выпадающий список позволяет пользователю выбрать один из сценариев выхода из системы: он может завершить текущую сессию или завершить все сессии.

Если пользователем будет является поставщик товаров, то у него так же будет выпадающий список для выхода из системы и вкладка настроек и еще 2 вкладки: товары и добавить товар. Во вкладке товаров представлены все товары, которые выставлены поставщиком, товары отображаются в виде таблицы, и каждая строчка товара имеет функционал удаления товара из магазина и изменения информации о товаре. Вкладка добавить товар содержит в себе форму для создание нового товара.

1.3 Прикладное ПО, используемое для разработки веб-страниц

В качестве среды разработки frontend была выбрана интегрированная среда разработки на TypeScript, CSS & HTML «WebStorm» [7]. Основными его преимуществами являются доступность, широкий функционал, удобный и понятный интерфейс.

В качестве среды разработки backend была выбрана интегрированная среда для разработки на Java «IntelliJ IDEA» [8]. Главными преимуществами высокая интеграция с языком программирования Java и высокий уровень интеграции с фреймворком Spring.

Для промежуточного тестирования обработки backend сервером запросов был выбран Postman [9]. Основными преимуществами являются простота и большой функционал.

Для просмотра веб-страниц используются браузеры Google Chrome и Microsoft Edge, поскольку они наиболее распространены среди пользователей и поддерживают больше функционала, чем другие браузеры, что делает их хорошими площадками для тестирования созданных страниц.

2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

2.1 Анализ предметной области

Анализ аналогов веб-приложений в категории «Магазин компьютерных комплектующих» проводился среди трех компаний: «RapidComp» [10], «LexComp» [11], и «PromPortal.su» [12]. Сравнение данных сайтов произведено по следующим параметрам:

- скорость загрузки сайта;
- удобство использования;
- адаптивный дизайн;
- светлая и темная тема;
- возможность регистрации клиентом;
- возможность добавления товара в корзину сразу, при наведении на карточку товара.

В качестве первого примера рассмотрим интернет-магазин фирмы «RapidComp» (рисунок 1). Страница загружается за одну – три секунды. Из явных минусов по критерию удобство использования можно выделить огромную шапку сайта, которая перекрывает большое количество контента, что сильно усложняет навигацию и поиск на сайте. Карточки товаров не занимают большую часть контента. Адаптивный дизайн присутствует. Поддерживается только светлая тема. Возможность регистрации клиента отсутствует. Возможность быстрого добавления товара в корзину также отсутствует. Из плюсов большой выбор товара, а также удобное расположение контактов, связи с магазином, и большое количество информации о качестве товаров, которые подтверждаются сертификатами.



Рисунок 1 – Сайт «RapidComp»

Далее рассмотрим интернет-магазин фирмы «LexComp» (рисунок 2). Страница загружается за семь – десять секунд, это в два раза дольше, чем предыдущий сайт. Сайт удобен в использовании. Карточки товаров не занимают большую часть контента. Адаптивный дизайн присутствует. Поддерживается только светлая тема. Возможность регистрации клиента и добавления товара в корзину присутствует.

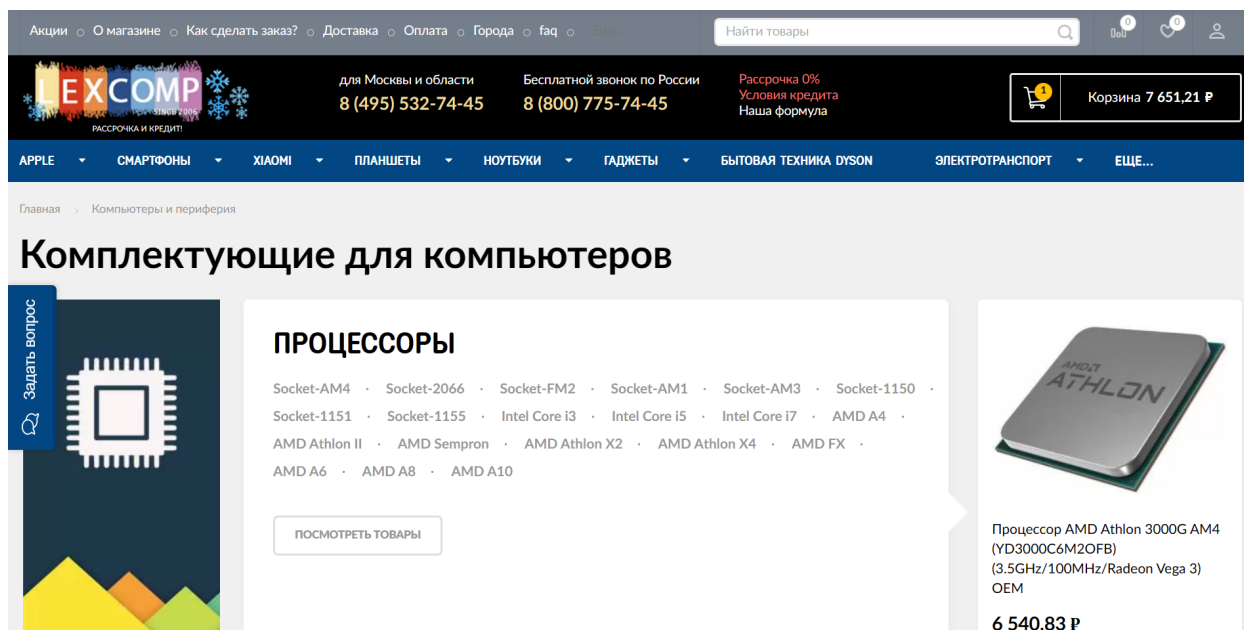


Рисунок 2 – Сайт «LexComp»

Последний интернет-магазин «PromPortal.su» (рисунок 3). Обратим внимание, что сайт предоставляет возможность не только покупки товаров, но и размещения компаниями товаров для продажи. Сайт загружается за одну-три секунды, также, как и первый сайт. На сайте отсутствуют кнопки быстрого добавления товара в корзину, а также адаптивный дизайн. Темной темы также нет. Регистрация клиента присутствует.

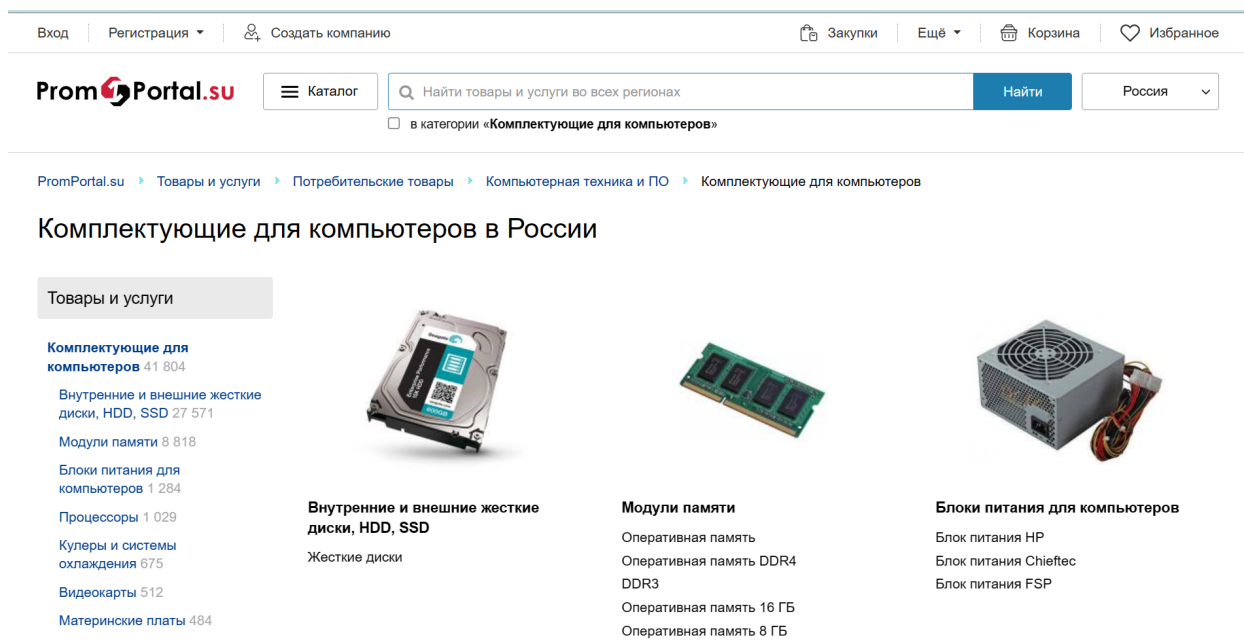


Рисунок 3 – Сайт «PromPortal.su»

Подводя итог, составим сравнительную таблицу (таблица 1).

Таблица 1 – Анализ сайтов-аналогов

Характеристика	«RapidComp»	«LexComp»	«PromPortal.su»
Скорость загрузки, сек	1-3	7-10	1-3
Удобство использования	-	+	-
Адаптивный дизайн	+	+	-
Светлая и темная тема	-	-	-
Регистрация клиента	-	+	+
Быстрое добавление товара в корзину	-	+	-

На основе проведенного анализа и таблицы 1 было принято решение о разработке приложения, который будет удобен в использовании пользователем и навигация по которому не будет вызывать затруднений. Для удобства пользования сайтом не только пользователем, но и поставщиками комплектующих, было принято решение разработать единый личный кабинет с возможностью регистрации как пользователя, так и поставщика. Также решение должно позволять быстро и интуитивно добавить товар в корзину, а также предоставлять современное решение приложения в виде светлой, темной темы сайта и адаптивного дизайна.

2.1 Выбор технологий разработки

HTML5 – это язык разметки текста, основа основ для проектирования любых проектов связанных с интернет-ресурсами. Благодаря ему мы можем делать сложно-структурированные конструкции из текста и картинок.

CSS3 – это язык стилей, который позволяет разным HTML-элементам задавать разные свойства и использовать анимации, что позволяет сделать интернет-ресурс красивым и понятно структурированным.

TypeScript – удобный язык программирования для веб-ресурсов который компилируется в JavaScript, так как все браузеры имеют его поддержку и сам язык изначально разрабатывался для взаимодействия с HTML-элементами.

React – удобный инструмент для создания UI интерфейса для frontend приложения, данная библиотека придерживается компонентного подхода,

что позволяет создавать переиспользуемые компоненты. Это уменьшает написания количество кода и повышает его качество и читаемость.

MobX [13] – библиотека для управления состояниями объектов. Данная библиотека помогает упростить написание `fronted` приложения так как разработчику не приходится задумываться о том, как отобразить актуальное состояние объекта в компоненте. Это за него делает библиотека.

GraphQL [14] – это язык запросов для API. В отличии от REST API, GraphQL позволяет обращаться к одному URL адресу и делать выборку нужных данных. Это позволяет упростить написания `frontend` и облегчить написание `backend` сервиса.

Язык программирования Java – это объектно-ориентированный язык программирования, который широко используется для создания веб-приложений, особенно для крупных проектов, благодаря своей модульной и расширяемой структуре кода.

Система сборки Maven [15]– это инструмент для автоматизации сборки проектов, который облегчает управление зависимостями и конфигурацией проекта.

Spring – это фреймворк, который упрощает разработку веб-приложений на Java и предоставляет разработчикам множество возможностей, включая управление транзакциями, обработку запросов, безопасность и так далее.

Spring Data JPA [16] – это модуль Spring, который упрощает работу с базами данных с помощью JPA, предоставляя абстракцию для работы с данными.

Spring Security [17] – это модуль Spring, который упрощает работу с базами данных с помощью JPA, предоставляя абстракцию для работы с данными.

PostgreSQL [18] – это модуль Spring, который упрощает работу с базами данных с помощью JPA, предоставляя абстракцию для работы с данными.

IntelliJ IDEA – это интегрированная среда разработки для языка программирования Java, которая предоставляет множество инструментов для

упрощения процесса разработки, включая поддержку системы сборки Maven, интеграцию с фреймворком Spring, а также инструменты для работы с базами данных, включая PostgreSQL.

Интегрированная среда разработки на TypeScript, CSS & HTML «WebStorm» - включает в себя с самого начала все вышеописанные технологии, и помогает подсказками при написании кода, что облегчает процесс создания веб-ресурса.

Браузеры Google Chrome и Microsoft Edge выбраны как самые популярные, их использование даст возможность объективно судить о успешности созданного приложения.

2.1 Разработка приложения

Все страницы frontend приложения построены по структуре «шапка – контентная часть – подвал» и основаны на компонентном подходе, который использует React. Весь backend сервис будет предоставлять API для frontend приложения.

2.1.1 Структура проекта

В директории проекта содержатся файлы, отвечающие за работу backend и frontend части приложения.

Backend часть приложения состоит из файла системы сборки Maven, а именно pom.xml, в котором прописаны все зависимости, необходимые для работы приложения. Также имеется директория security, которая содержит в себе файлы конфигурации и классы, необходимые для обеспечения безопасности приложения, построенного с использованием Spring Framework.

Также находятся классы, которые реализуют пользовательские сервисы аутентификации и авторизации, используемые для проверки подлинности пользователей и управления доступом и классы реализующие кастомные фильтры безопасности, которые можно использовать для обеспечения

дополнительного уровня защиты для конкретных участков приложения.
(рисунок 4, 5).

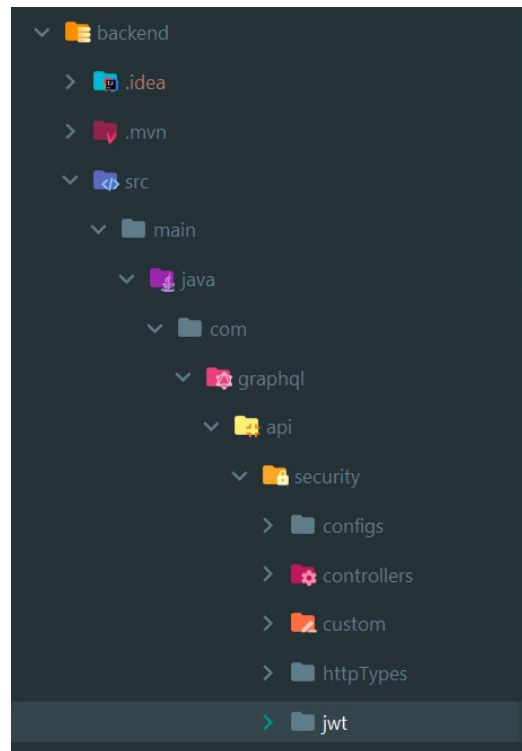


Рисунок 4 – Структура backend части приложения

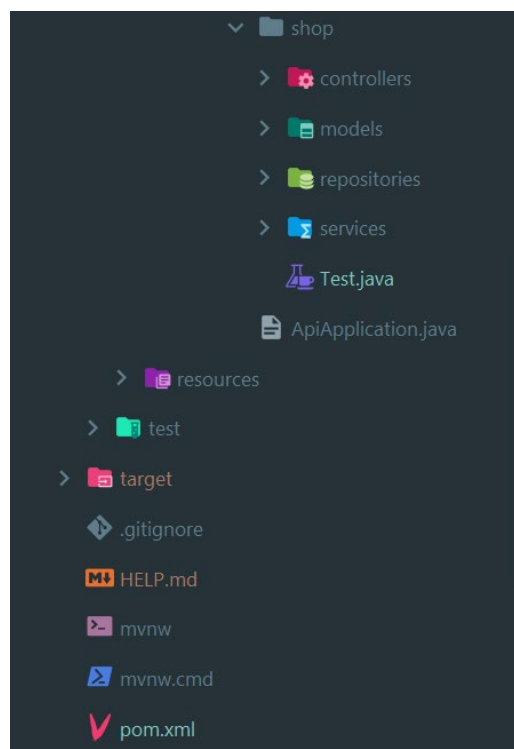


Рисунок 5 – Структура backend части приложения

Frontend часть приложения состоит из React, Webpack, TypeScript и MobX, включает в себя компоненты пользовательского интерфейса (UI),

модули для управления состоянием и сборщик модулей (рисунок 6, 7). Они позволяют создавать быстрое и надежное приложение с легко поддерживаемым кодом и удобным пользовательским интерфейсом.

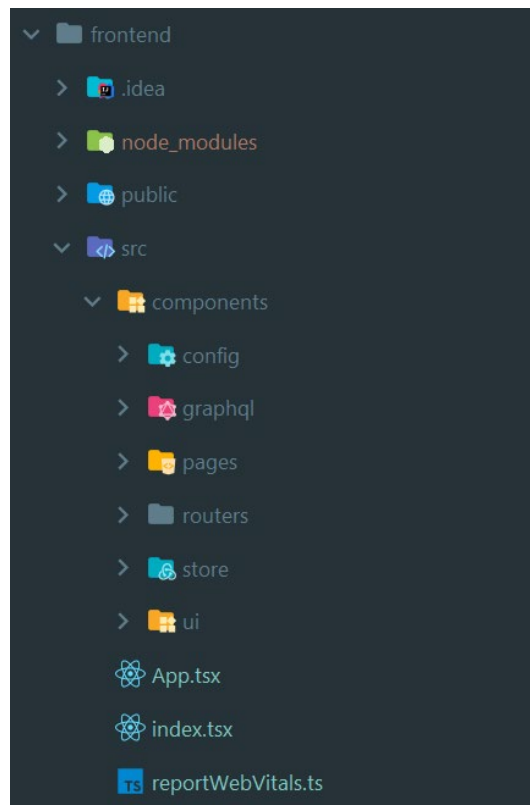


Рисунок 6 – Структура frontend части приложения

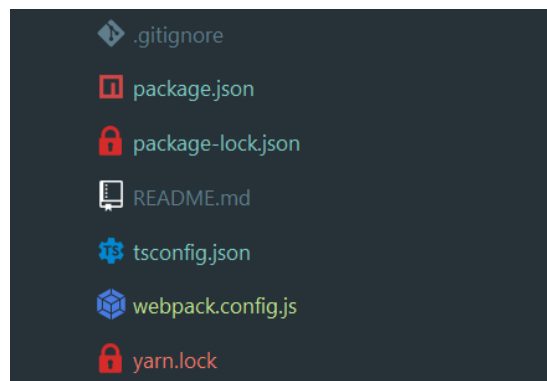


Рисунок 7 – Структура frontend части приложения

2.1.1 База данных

База данных разрабатываемого приложения состоит из 15 таблиц (рисунок 8):

- category – категория товара;
- computer_component – компьютерное комплектующее;

- `computer_component_categories` – таблица связи между категориями и компьютерным комплектующими;
- `customer` – покупатель;
- `image` – изображения товаров;
- `jwt_model` – JS web токен;
- `orders` – корзина;
- `publisher` – поставщик;
- `relation_orders_to_computer_components` – таблица связи между корзиной и компьютерными комплектующими;
- `roles` – роли пользователей (администратор, покупатель и т.д.);
- `state_token` – состояние токена (удален/не удален);
- `token` – токен;
- `type_token` – тип токена;
- `users` – пользователь;
- `users_roles` – таблица связи между пользователями и ролью пользователя.

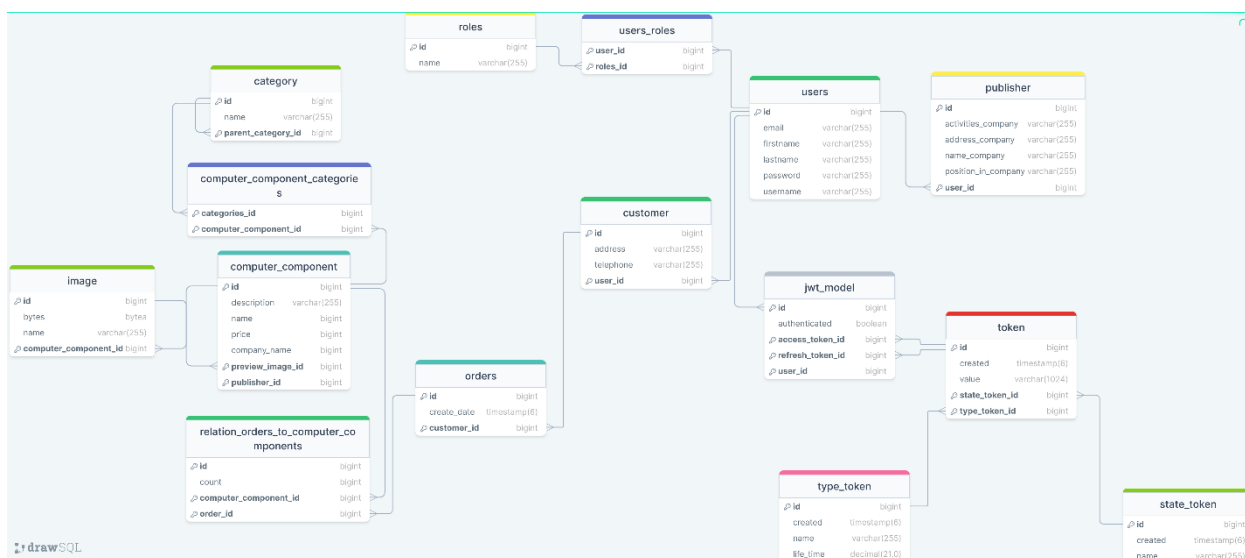


Рисунок 8 – ERD-диаграмма

В каждой таблице присутствует уникальный идентификатор (ID), который является primary key (PK), что означает однозначность идентификации записи. Остальные ID, которые используются в таблицах

являются foreign key (FK), то есть внешними ключами, которые ссылаются на первичный ключ другой таблицы.

Таблица roles хранит в себе название роли и уникальный id. Связана с таблицей users по отношению многие ко многим. Данная связь реализуется через смежную таблицу users_roles, которая хранит id пользователя и id роли для этого пользователя.

Таблица publisher связана с таблицей users по отношению один к одному и хранит в себе данные о поставщике.

Таблица customer связана с таблицей users по отношению один к одному и хранит в себе данные о покупателе.

Таблица jwt_model хранит в себе id пользователя, id двух токенов из таблицы token и логическое значение о авторизации пользователя.

Таблица token связана с таблицами type_token и state_token по отношению многие к одному. Данная таблица содержит значение токена и время его создания.

Таблица state_token содержит время создания записи в бд и название состояния для токена.

Таблица type_token содержит время жизни для токена, название токена и время создания записи в бд.

Таблица category содержит в себе название категории и id на родителя категории из таблицы category.

Таблица computer_component_categories это таблица реализующая связь многие ко многим для категорий и для компьютерных комплектующих.

Таблица image хранит в себе картинку и название для картинки и связана с таблицей computer_component по отношению многие к одному.

Таблица computer_component связана таблицей image по отношению один ко одному и содержит информацию о компьютерных комплектующих.

Таблица relation_orders_to_computer_components это связывающая таблица для таблицы order и таблицы computer_component и содержит поле для количества товара.

Таблица order это таблица корзины покупателя она связана с таблицей customer по отношению один к одному и содержит время создания.

В файле application.yml прописаны настройки для подключения к базе данных PostgreSQL к приложению (рисунок 9).

```
1  spring:
2    datasource:
3      url: jdbc:postgresql://localhost:5432/javaGraphQLApi
4      username: admin
5      password: ss261645
6      driver-class-name: org.postgresql.Driver
7    jpa:
8      hibernate:
9        ddl-auto: create-drop
10     show-sql: false
11     properties:
12       hibernate:
13         format_sql: true
14     database: postgresql
15     database-platform: org.hibernate.dialect.PostgreSQLDialect
```

Рисунок 9 – Файл application.yml

2.1.1 Авторизация и регистрация

Для авторизации и регистрации используются кастомные классы. Авторизацию осуществляется по JWT.

Для ограничения доступа к некоторым адресам и для использования кастомного фильтра авторизации был написан конфигурационный файл (рисунок 10) для Spring Security. Помимо этого, в данном файле добавлен алгоритм для хеширования паролей пользователей.

```

64         no usages   Даниил
        @Autowired
65     public void configureGlobal(@NotNull AuthenticationManagerBuilder auth) throws Exception {
66         auth.userDetailsService(userDetailsService).passwordEncoder(passwordEncoder);
67     }
68
69     no usages   Даниил
70     @Bean
71     public UserDetailsService userDetailsService() { return userDetailsService; }
72
73
74     no usages   Даниил *
75     @Bean
76     public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
77         http
78             .csrf(AbstractHttpConfigurer::disable)
79             .authorizeHttpRequests(authorize -> authorize
80                 .requestMatchers(...patterns: "/graphql").permitAll()
81                 .requestMatchers(...patterns: "/graphql").permitAll()
82                 .requestMatchers(...patterns: "/images").permitAll()
83                 .requestMatchers(...patterns: "/images/**").permitAll()
84                 .anyRequest().authenticated()
85                 .and().addFilterAfter(jwtFilter, UsernamePasswordAuthenticationFilter.class)
86             ).httpBasic().disable();
87         return http.build();
88     }

```

Рисунок 10 – Файл конфигурации для Spring Security

Для регистрации пользователей был создан отдельный контроллер, который содержит метод для создания нового пользователя (рисунок 11). Контроллер обрабатывает запрос, полученный по обращению через GraphQL в качестве мутации.

```

11     no usages   Даниил *
12     @Controller
13     @CrossOrigin
14     public class UserController {
15
16         2 usages
17         private final UserService userService;
18
19         no usages   Даниил
20         @Autowired
21         public UserController(UserService userService) { this.userService = userService; }
22
23         no usages   Даниил
24         @MutationMapping
25         public User createUser(@Argument User user) { return userService.saveUser(user); }

```

Рисунок 11 – Контроллер для регистрации

Для авторизации и выхода из системы так же создан отдельный контроллер (рисунок 12), который содержит методы для входа и выхода пользователя из системы. Когда пользователь входит в систему и указывает

валидные данные, он получает в ответ от сервера 2 токена для дальнейшего обращения к защищенным методам.

```
14 @Controller
15 @CrossOrigin
16 public class AuthController {
17
18     5 usages
19     private final AuthService authService;
20
21     no usages Даниил
22     @Autowired
23     public AuthController(AuthService authService) { this.authService = authService; }
24
25     no usages Даниил
26     @MutationMapping
27     public JwtResponse login(@Argument JwtRequest jwtRequest) throws AuthException {
28         return authService.getAccessAndRefreshTokens(jwtRequest);
29     }
30
31     no usages Даниил
32     @MutationMapping
33     public JwtResponse refreshAccessToken(@Argument JwtRefreshRequest request) throws AuthException {
34         return authService.getAccessToken(request.getRefreshToken());
35     }
36
37     no usages new *
38     @MutationMapping
39     public String logout(@Argument String refreshToken) throws AuthException {
40         authService.disableJwtByRefreshToken(refreshToken);
41         return "disabled this session";
42     }
43
44     no usages new *
45     @MutationMapping
46     public String logoutAllSessions(@Argument String refreshToken) throws AuthException {
47         authService.disableAllJwtByRefreshToken(refreshToken);
48         return "disabled all sessions";
49     }
50 }
```

Рисунок 12 – Контроллер авторизации

2.1.1 Работа с запросами

Согласно подходу Spring MVC, модели взаимодействуют с базами данных, а контроллеры получают запросы.

Для взаимодействия с базой данных требуется репозиторий, который представляет собой интерфейс, наследующий встроенный в SpringDataJpa интерфейс JpaRepository. Данный интерфейс упрощает работу с запросами, так как он уже содержит реализацию практически всех необходимых запросов.

Сервис выступает в роли прослойки, которая разделяет логику контроллера и репозитория, и где хранится вся бизнес-логика.

2.1.2 JWT токен

Для более удобной работы с защитой и аутентификацией, было принято решение использовать JWT. Для более удобной и гибкой настройки токенов все параметры были вынесены в файл `application.yml` (рисунок 13).

```
30 jwt:
31   type: Bearer
32   refresh:
33     timelife:
34       days: 20
35       minutes: 0
36     key: zL1HB3Pch05Avfynovxrf/kpF902m4NCWKJUjEp27s9J2jEG3if1KCGylaZ8fDeo0NSTJP/wAzKawB8F9r0MNg==
37   access:
38     timelife:
39       days: 0
40       minutes: 5
41     key: qBTmv4oXFFR2GwjexDJ4t6fsIUIUhhXqlktXjXdkcyygs8nPVEwMfo29VDRRepYDVV5IkIxBMzr70EHXEhd37w==
42
```

Рисунок 13 – Настройки для JWT

Для работы с JWT токеном был написан класс (рисунок 14), который содержит утилиты для создания, валидации и получения данных их токена.

```
49 public String generateAccessToken(@NotNull User user) {
50     return Jwts
51         .builder()
52         .setSubject(user.getUsername())
53         .setExpiration(Date
54             .from(LocalDateTime.now().plusSeconds(lifeAccess.getSeconds()) LocalDateTime
55                 .atZone(ZoneId.systemDefault()) ZonedDateTime
56                 .toInstant()))
57         .signWith(accessKey)
58         .claim(s: "roles", user.getRoles())
59         .claim(s: "email", user.getEmail())
60         .compact();
61 }
62
63 public String generateRefreshToken(@NotNull User user) {
64     return Jwts
65         .builder()
66         .setSubject(user.getUsername())
67         .setExpiration(Date
68             .from(LocalDateTime.now().plusSeconds(lifeRefresh.getSeconds()) LocalDateTime
69                 .atZone(ZoneId.systemDefault()) ZonedDateTime
70                 .toInstant()))
71         .signWith(refreshKey)
72         .compact();
73 }
74
```

Рисунок 14 – Класс с утилитами для работы с JWT

3 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

3.1 Тестирование доступности страниц

В первом тест-кейсе будем использовать программу Postman для тестирования публичных и ограниченных API по протоколу GraphQL и RestAPI. Под публичным API подразумевается получение каких-либо данных без предоставления прав доступа. Под ограниченным API подразумевается получение данных с предоставлением прав доступа, в нашем случае это роли зарегистрированных пользователей: администратор, покупатель, поставщик.

Для проверки роли у пользователей будем использовать аннотацию @PreAuthorize.

В качестве публичного API будем использовать данный GraphQL запрос по `http://localhost:8080/graphql`. Данный запрос позволяет зарегистрировать пользователя и получить данные, которые были указаны в запросе GraphQL. Запрос представлен на рисунке 15.

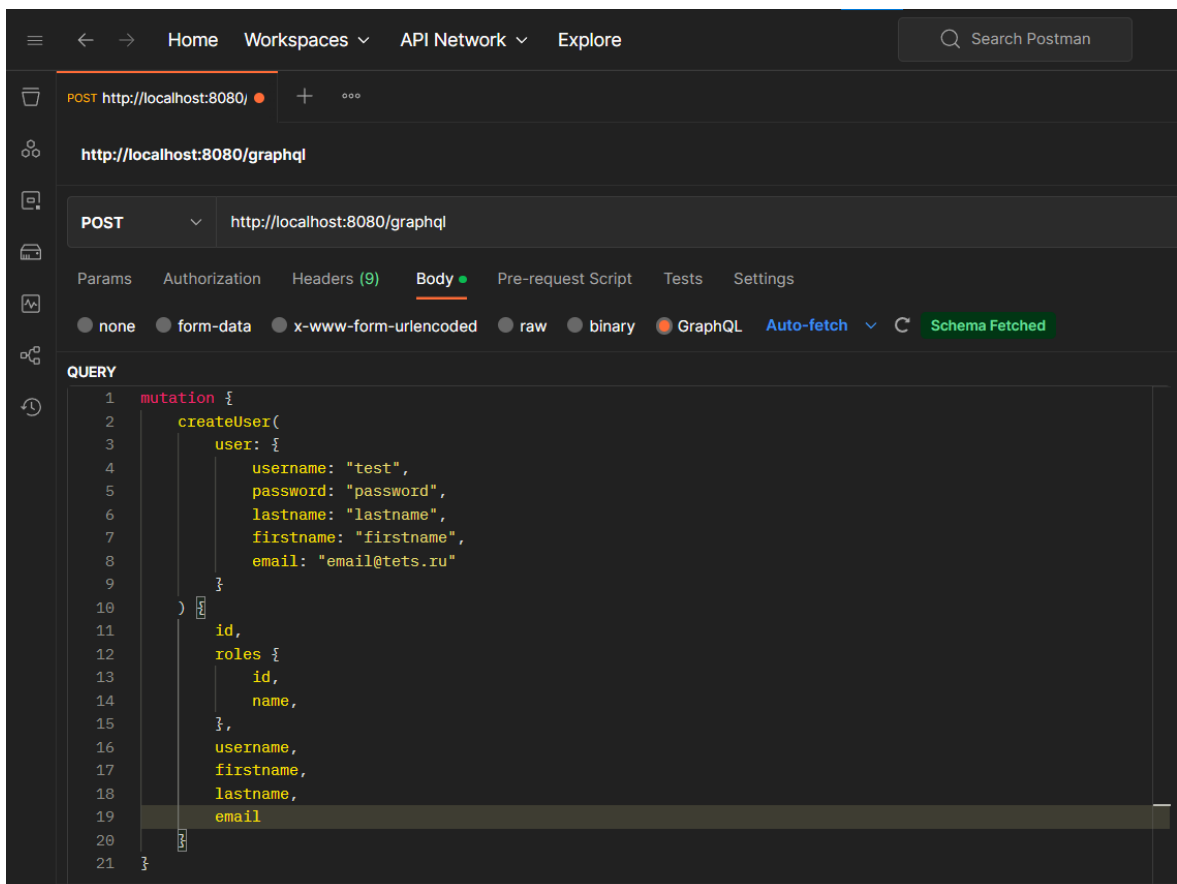


Рисунок 15 – Запрос на регистрацию пользователя

Ответ представлен на рисунке 16. Все работает корректно.

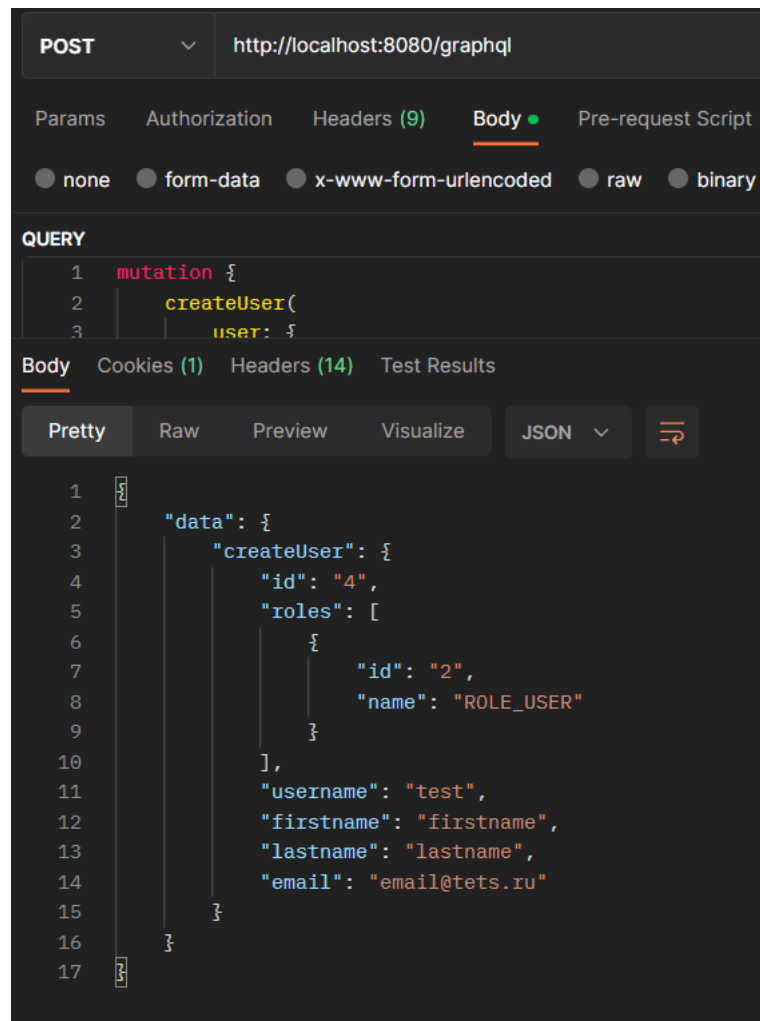


Рисунок 16 – Ответ на регистрацию пользователя

В качестве публичного API будем использовать второй GraphQL запрос по <http://localhost:8080/graphql>. Данный запрос позволяет получить access и refresh токены, с помощью которых пользователь может авторизоваться. Запрос представлен на рисунке 17.

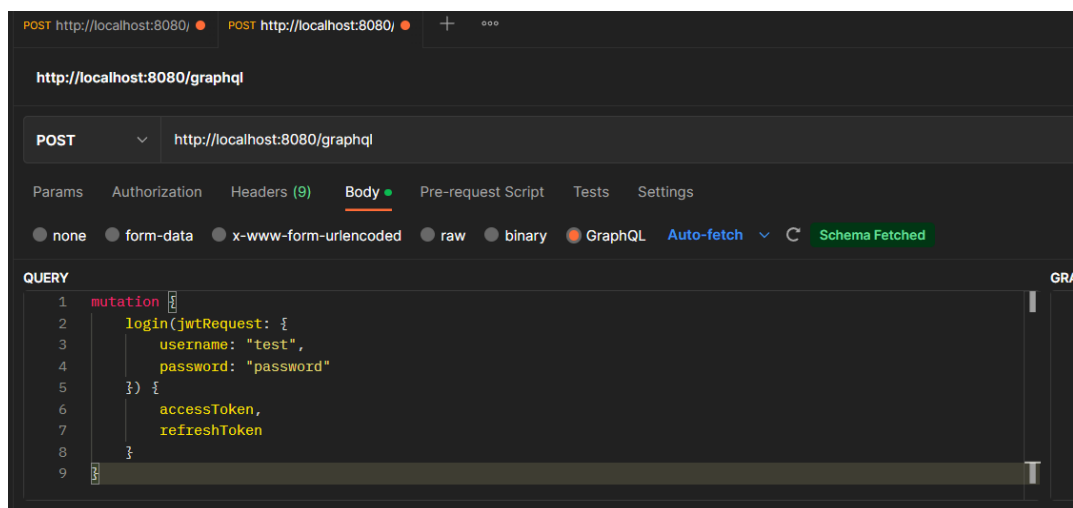


Рисунок 17 – Запрос на получение токенов

Ответ представлен на рисунке 18. Все работает корректно.

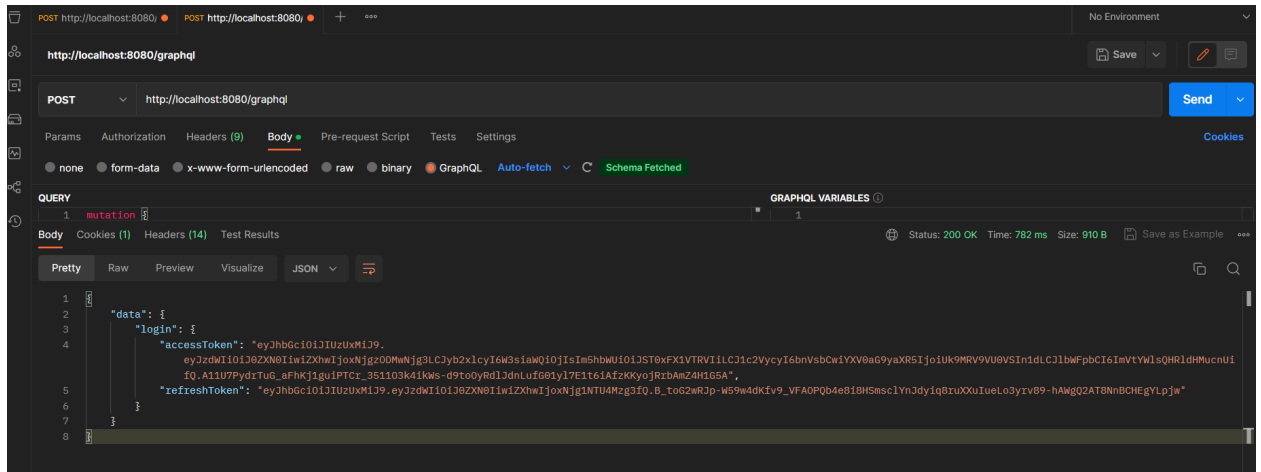


Рисунок 18 – Ответ на получение токенов

В качестве публичного API будем использовать третий GraphQL запрос по `http://localhost:8080/graphql`, который позволяет зарегистрировать поставщика и в дальнейшем, опираясь на него, протестируем ограниченный API. Запрос представлен на рисунке 19.

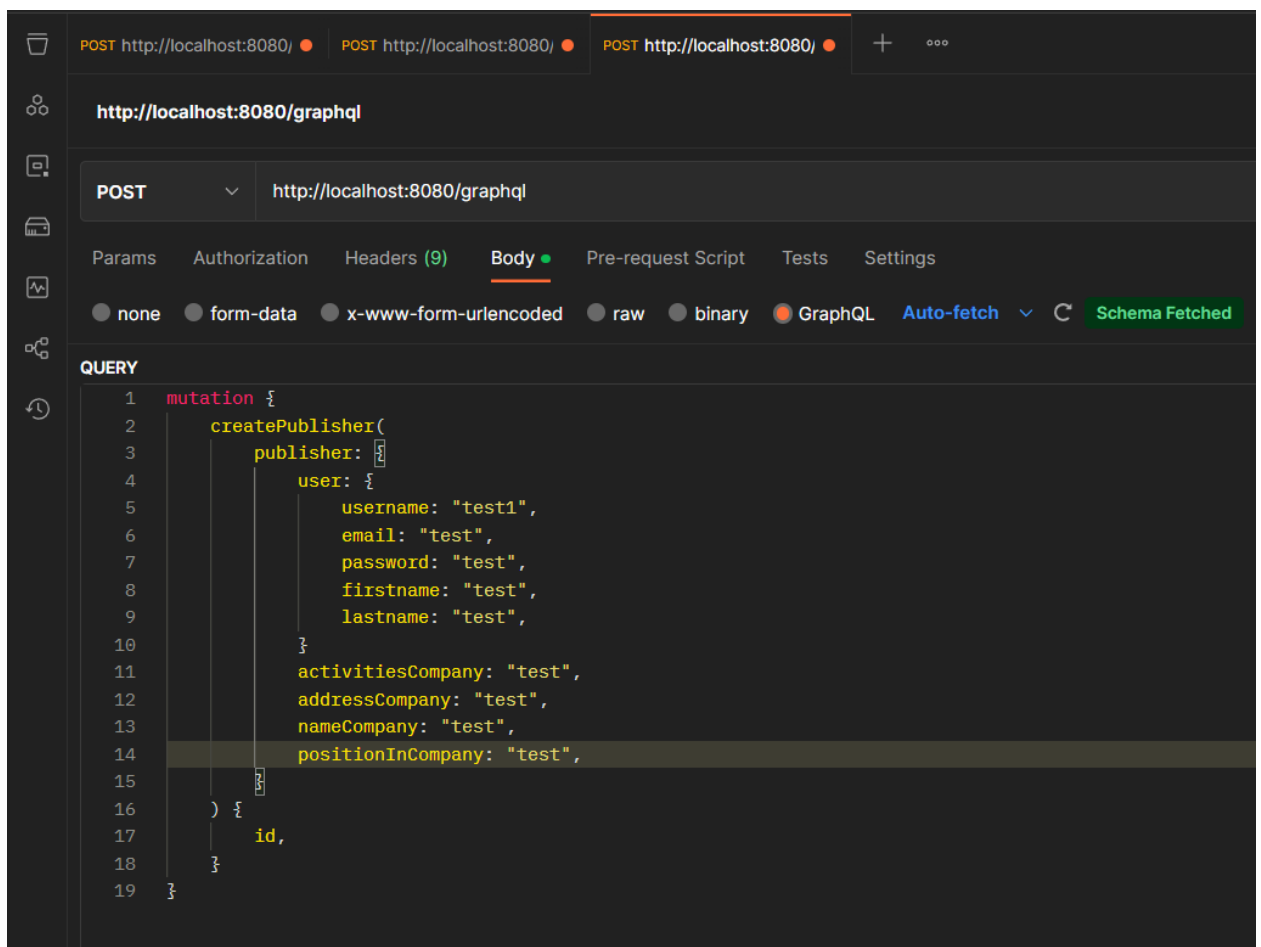


Рисунок 19 – Запрос на регистрацию поставщика

Ответ представлен на рисунке 20. Все работает корректно.

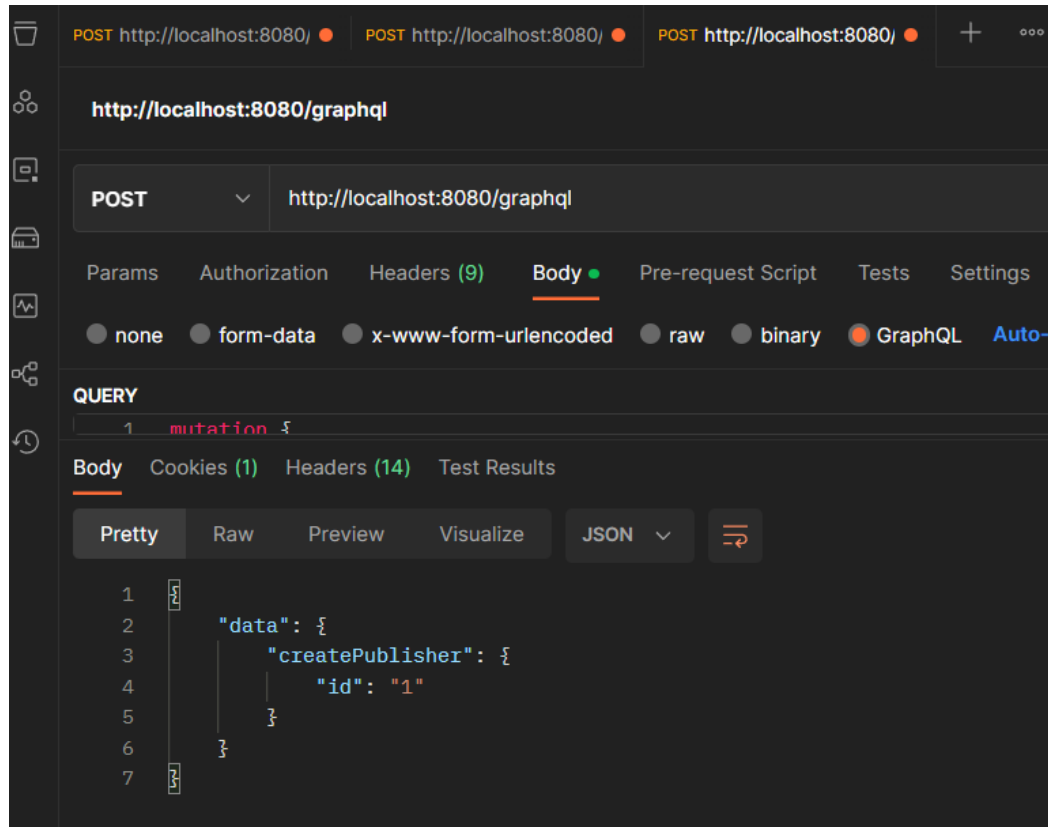


Рисунок 20 – Ответ на регистрацию поставщика

Затем протестируем вход в качестве поставщика, тем самым тестируя ограниченный API. Вход выполнен успешно, об этом говорит статус 200 (рисунок 21).

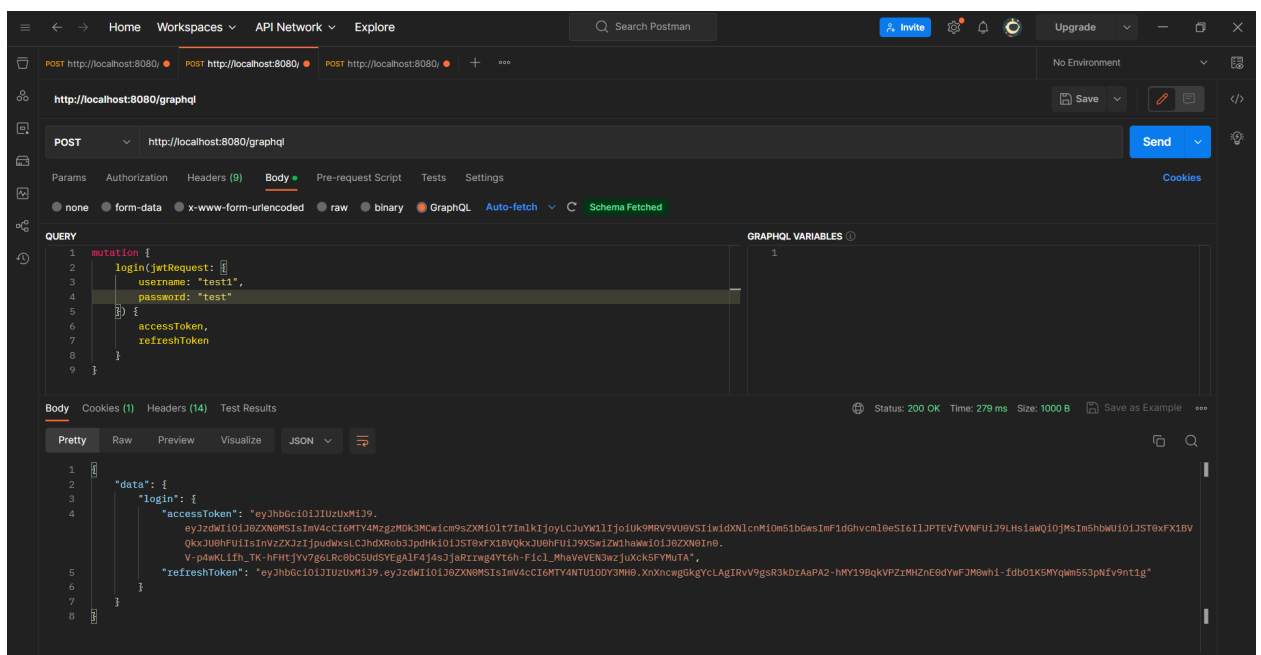


Рисунок 21 – Вход поставщика

Право загружать изображения на сайт имеет только поставщик, поэтому проверим, что при попытке обычным пользователем загрузить изображение вернется ошибка (рисунок 22).

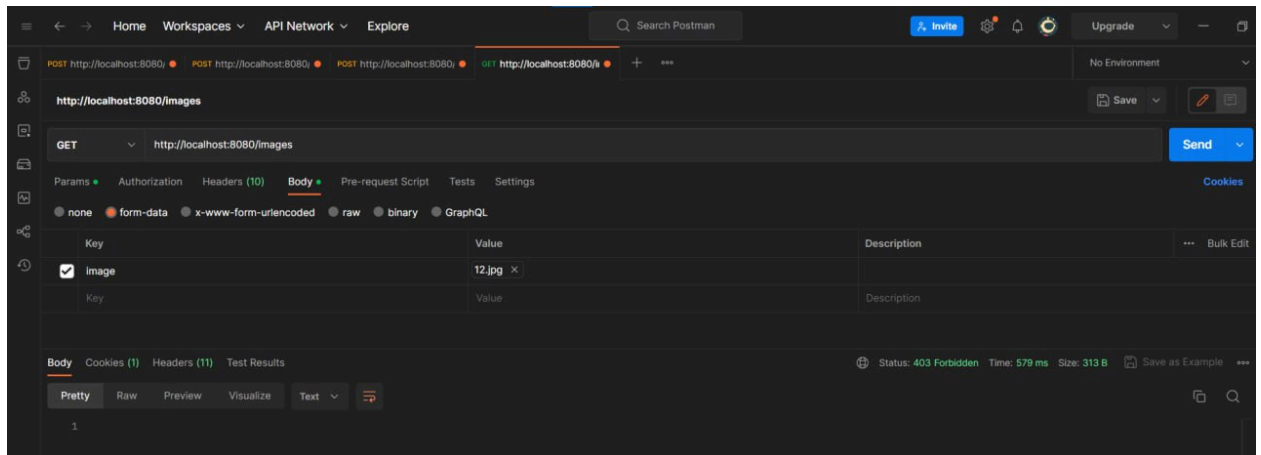


Рисунок 22 – Загрузка изображения пользователем

Исходя из рисунка 22 можно заметить, что вернулся статус 403, что означает, что у пользователя недостаточно прав, следовательно, все работает корректно.

Проверим работоспособность загрузки изображения через поставщика (рисунок 23).

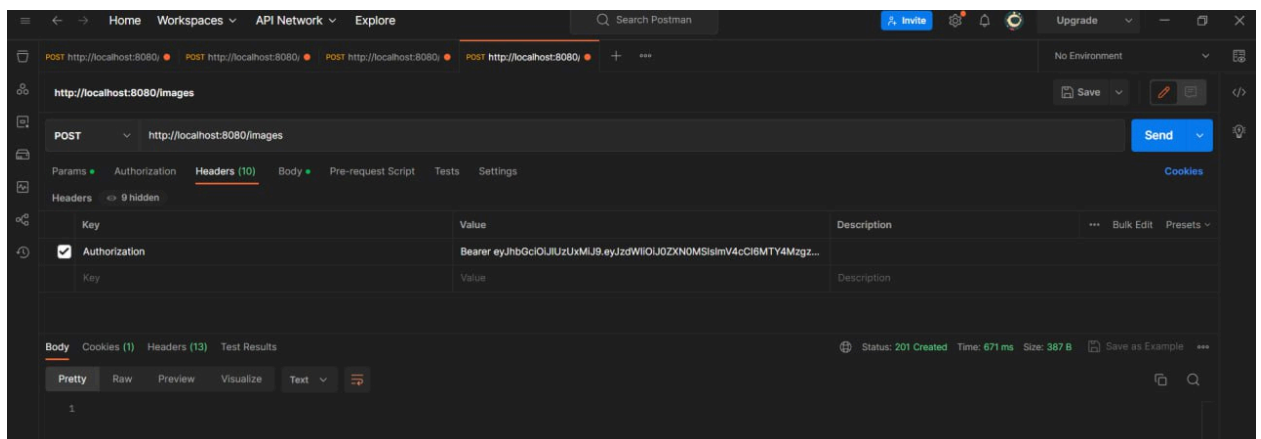


Рисунок 23 – Загрузка изображения поставщиком

Исходя из рисунка 23 можно заметить, что вернулся статус 201, что означает, что файл создан и загружен успешно, следовательно, все работает корректно.

3.2 Тестирование запросов

Проведем тестирование приложения через браузер.

При переходе на страницу авторизации и попытке ввести некорректные данные возникает ошибка, которая информирует об ошибочных данных и выделяется в виде красного прямоугольника с указанием ошибки (рисунок 24).

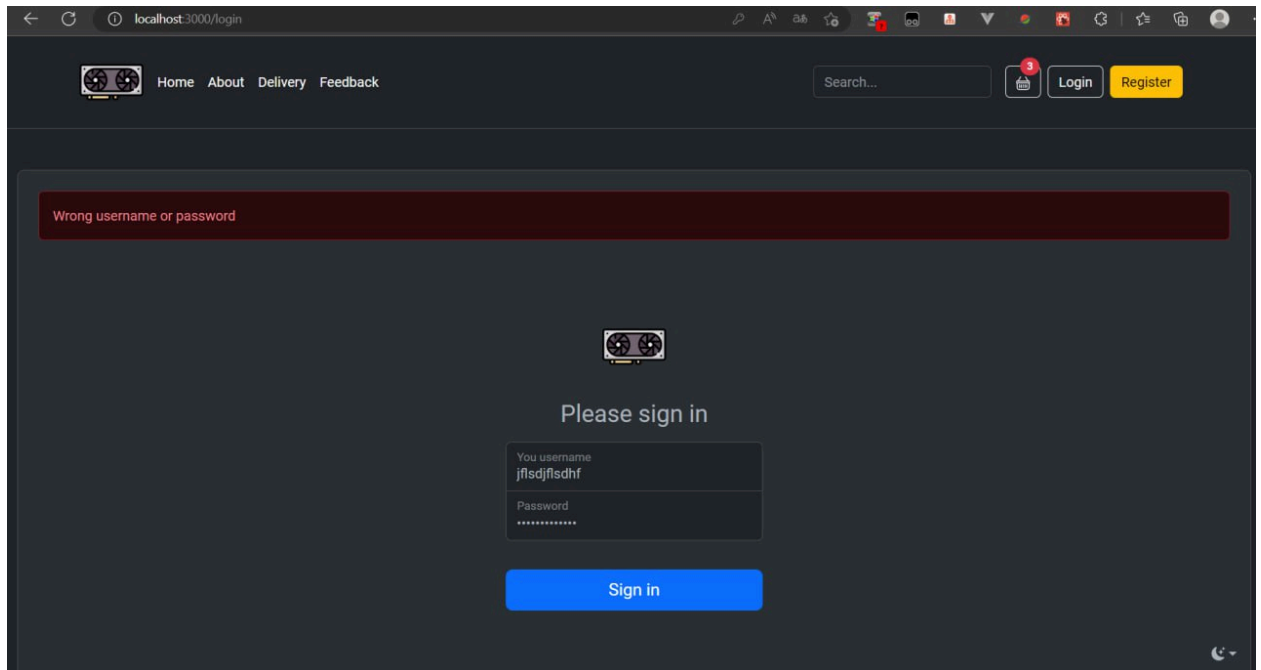


Рисунок 24 – Ошибка авторизации

Исходя из рисунка 24, тест-кейс работает успешно.

Если ошибка будет не связана с данными, а с тем, что frontend не смог отправить запрос на backend, то также получим уведомление на указание данной ошибки (рисунок 25).

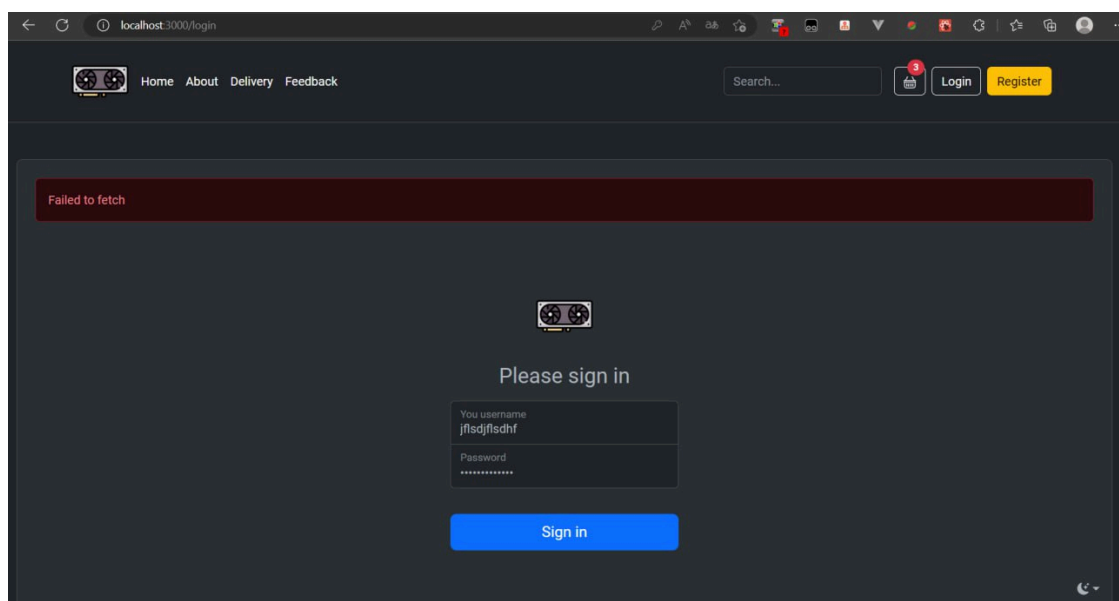


Рисунок 25 – Ошибка запроса в backend

Исходя из рисунка 25, тест-кейс работает успешно.

Если пользователь введет валидные данные, то он окажется в личном кабинете (рисунок 26).

Рисунок 26 – Личный кабинет

Исходя из рисунка 26, тест-кейс работает успешно.

При нажатии на выпадающий список выхода у пользователя будет на выбор два сценария: закончить текущий сеанс или завершить все сеансы (рисунок 27).

Рисунок 27 – Выход из личного кабинета


У данного пользователя три активных сеанса. Данную информацию проверим через pgAdmin (рисунок 28). В таблице всего шесть записей токенов

по парам (токен доступа и токен обновления) и у всех токенов статус один, что означает их активность.


	id [PK] bigint	created timestamp without time zone	value character varying (1024)	state_token_id bigint	type_token_id bigint
1	1	2023-05-11 22:18:22.285	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDcwMn0.DMveR9MoJ40UE...	1	1
2	2	2023-05-11 22:18:22.342	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4MzgzMzAwMiwicm9sZXMiOiI7Imkl...	1	2
3	3	2023-05-11 22:20:03.782	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDgwM30.MlipFi1ONv1Qn9I8...	1	1
4	4	2023-05-11 22:20:03.796	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4MzgzMzEwMywicm9sZXMiOiI7Imkl...	1	2
5	5	2023-05-11 22:20:15.158	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDgxNX0.s5WhwzEfisMOTIA...	1	1
6	6	2023-05-11 22:20:15.171	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4MzgzMzExNSwicm9sZXMiOiI7Imklkj...	1	2


Рисунок 28 – Таблица токенов со статусом «один»

После нажатия на кнопку выхода из всех сеансов, пользователь выходит из личного кабинета (рисунок 29) и попадает на главную страницу (рисунок 30). Тем временем в PgAdmin у всех токенов, связанных с пользователем, статус изменился на два, что означает их деактивацию (рисунок 31).

[Home](#)[About](#)[Delivery](#)[Feedback](#)

3





settings

items

add item

Logout

logout this session

logout all session

First name

Last name

Username

Email

Address

Telephone

Last password

New password

Save changes

Рисунок 29 – Выход из всех сессий

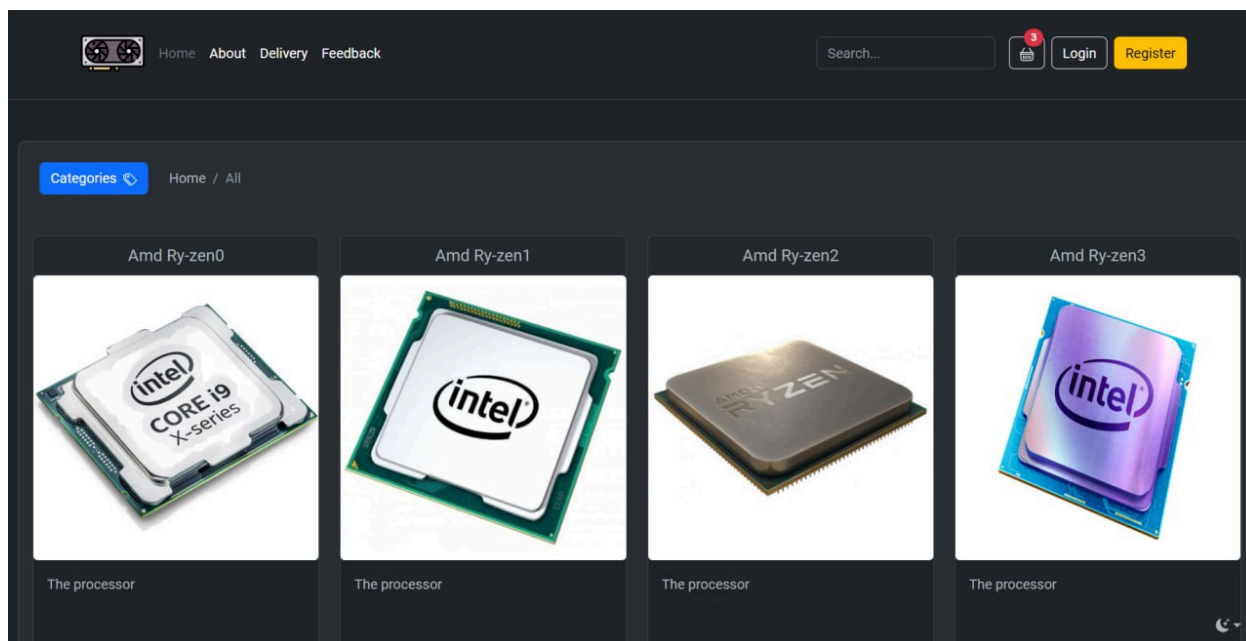


Рисунок 30 – Главная страница после выхода из сессии

17:59:17						
Результат План выполнения Сообщения Notifications Query Editor						
	id [PK] bigint	created timestamp without time zone	value character varying (1024)	state_token_id bigint	type_token_id bigint	
1	1	2023-05-11 22:18:22.285	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDc...	2	1	
2	2	2023-05-11 22:18:22.342	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4Mzg2MzE...	2	2	
3	3	2023-05-11 22:20:03.782	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDg...	2	1	
4	4	2023-05-11 22:20:03.796	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4Mzg2MzE...	2	2	
5	5	2023-05-11 22:20:15.158	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4NTU2MDg...	2	1	
6	6	2023-05-11 22:20:15.171	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbGlzImV4cCI6MTY4Mzg2MzE...	2	2	

Рисунок 31 – Таблица токенов со статусом «два»

Исходя из рисунков 29, 30, 31, можно сделать вывод, что тестирование тест-кейса проведено успешно.

ЗАКЛЮЧЕНИЕ

Результатом выполнения курсовой работы является веб-приложение на тему «Магазин компьютерных комплектующих», созданный с использованием технологий HTML5, CSS3, Java, Java Spring Framework, PostgreSQL и навыков, полученных в ходе курса «Шаблоны программных платформ языка Джава».

В ходе выполнения работы была проведен анализ предметной области для определения недостатков аналогичных сайтов и их устранения при реализации собственного интернет-ресурса. Приложение содержит соответствующий теме контент и функционал.

Ссылка на репозиторий GitHub с проектом –
<https://github.com/S0IG0/JavaCourseWork/issues>

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Э. Фримен, Э. Робсон. Изучаем HTML, XHTML и CSS 2-е изд. – М.: Питер, 2013 г. — 720 с.
2. Д. Макфарланд. Новая большая книга CSS. – М.: Питер, 2018 г. — 720 с.
3. Д. Фридман. React и Redux: Функциональная веб-разработка. – М.: ООО «И.Д. Вильямс», 2020 г. – 352 с.
4. Г. Шилдт. Java: Полное руководство, 10-е изд. – М.: Диалектика-Вильямс, 2018 г. – 1376 с.
5. А. Нейл, М. Гриффитс. TypeScript: быстрый старт. – М.: ООО «И.Д. Вильямс», 2020 г. – 352 с.
6. К. Уоллис, Spring в действии, 5-е изд. – М.: ДМК Пресс, 2021 г. – 784 с.
7. J. Herman, WebStorm Essentials. Бирмингем, Великобритания: Packt Publishing, 2015. 166 с.
8. Дмитриев, С. Владение IntelliJ IDEA. Бирмингем, Великобритания: Packt Publishing, 2017. 422 с.
9. А. Кришна, Краткое руководство по Postman. Бирмингем, Великобритания: Packt Publishing, 2018. 114 с.
10. Магазин компьютерных комплектующих «RapidComp» [Электронный ресурс]. – URL: <https://rapidcomp.ru/>, (дата обращения: 1.05.2023).
11. Магазин компьютерных комплектующих «LexComp» [Электронный ресурс]. – URL: <https://www.lexcomp.ru/>, (дата обращения: 1.05.2023).
12. Магазин компьютерных комплектующих «PromPortal.su» [Электронный ресурс]. – URL: <https://promportal.su/g/1819/komplektuyuschie-dlya-kompyuterov/>, (дата обращения: 1.05.2023).

13. V. Horgus, Реактивное программирование с использованием MobX. Санкт-Петербург: Питер, 2018. 192 с.
14. Kramer, H., N. Pratap, and A. Sharp. Изучение GraphQL. Сан-Франциско, США: O'Reilly Media, 2018. 372 с.
15. Косарев, А. Система сборки Maven. СПб.: БХВ-Петербург, 2018. 352 с.
16. Шлаер, Б., Марк, К. Spring Data JPA. М.: ДМК Пресс, 2019. 320 с.
17. Молграби, М., Касеем, М. Spring Security. М.: ДМК Пресс, 2020. 384 с.
18. Фарлей, Д. PostgreSQL: Надежное и масштабируемое развертывание баз данных. М.: ДМК Пресс, 2021. 464 с.