|  |
| --- |
|  |
| 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: Разработка серверных частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: «Разработка панели администратора для сервиса генерации онлайн таблиц»

Студент: Горшенёв Леонид Александрович

Группа: ИКБО-16-19

Работа представлена к защите 23.11.21 / Горшенёв Л.А./

(подпись и ф.и.о. студента)

Руководитель: Лобанов Александр Анатольевич, доцент

Работа допущена к защите (дата) / Лобанов А.А./

(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты:

/ /

/ /

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

РТУ МИРЭА. 2021 г.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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: Разработка серверных частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Горшенёв Леонид Александрович

Группа: ИКБО-16-19

Срок представления к защите: 23.11.2021

Руководитель: Лобанов Александр Анатольевич, доцент

**Тема:** «Разработка панели администратора для сервиса генерации онлайн таблиц»

**Исходные данные:** используемые технологии: HTML5, CSS3, Java/Kotlin/PHP (на выбор), Jetbrains Intellij IDEA/Jetbrains PHPStorm/Eclipse (на выбор), SQL/noSQL СУБД (на выбор), наличие: межстраничной навигации, внешнего вида страниц, соответствующего современным стандартам веб-разработки, использование паттерна проектирования (MVC, Clear Architecture, DDD). 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 разрабатываемого веб-приложения. 2. Обосновать выбор технологий разработки веб-приложения. 3. Разработать архитектуру веб-приложения на основе выбранного паттерна проектирования. 4. Реализовать слой серверной логики веб-приложения с применением выбранной технологии. 5. Реализовать слой логики базы данных. 6. Разработать слой клиентского представления веб-приложения 7. Создать презентацию по выполненной курсовой работе.

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

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_/Р. Г. Болбаков/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.А. Лобанов/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_/Л.А. Горшенёв/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

# Глоссарий

1. Apache Ant — утилита для автоматизации процесса сборки программного продукта.
2. Apache Maven — фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах на языке POM, являющемся подмножеством XML.
3. API (Application Programming Interface) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
4. BLoC — это акроним от «Business Logic Component» (компонент бизнес-логики). Как следует из названия, это класс, отделяющий бизнес-логику приложения от пользовательского интерфейса.
5. Clean Architecture – паттерн проектирования веб-приложений, обеспечивающий возможность тестировать модули, независимый от UI и независимый от БД, внешних фреймворков и библиотек.
6. Cubit – реализация паттерна BLoC, которая доступна из пакета flutter\_bloc
7. Dart — язык программирования, созданный Google. Dart позиционируется в качестве замены/альтернативы JavaScript.
8. DDD (Domain-Driven Design) – это набор принципов и схем, направленных на создание оптимальных систем объектов. Сводится к созданию программных абстракций, которые называются моделями предметных областей. В эти модели входит бизнес-логика, устанавливающая связь между реальными условиями области применения продукта и кодом.
9. Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.
10. Flutter — комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, а также веб-приложений с использованием языка программирования Dart, разработанный и развиваемый корпорацией Google.
11. Google — американская транснациональная корпорация в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.
12. Google Docs — бесплатный онлайн-офис, разрабатываемый компанией Google.
13. Google Forms — это программа для администрирования опросов, входящая в состав бесплатного веб-пакета редакторов документов Google, предлагаемого Google.
14. Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven
15. IDE – комплекс программных средств, используемый программистами для разработки программного обеспечения.
16. IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.
17. Java — строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems.
18. LibreOffice — кроссплатформенный, свободно распространяемый офисный пакет с открытым исходным кодом, созданный как ответвление OpenOffice в 2010 году.
19. Lombok — это плагин компилятора, который добавляет в Java новые «ключевые слова» и превращает аннотации в Java-код, уменьшая усилия на разработку и обеспечивая некоторую дополнительную функциональность.
20. Material Design — стиль графического дизайна интерфейсов программного обеспечения и приложений, разработанный компанией Google.
21. Microsoft Excel — программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone.
22. MongoDB — документоориентированная система управления базами данных, не требующая описания схемы таблиц.
23. MVC (Model-View-Controller) – схема разделения данных приложения, и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо;
24. NoSQL – нереляционная база данных – база, которая хранит данные без четких связей друг с другом и не имеет четкой структуры.
25. REST — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
26. RowShare – корпоративная онлайн-платформа, позволяющая создавать и взаимодействовать с таблицами.
27. SOAP — протокол обмена структурированными сообщениями в распределённой вычислительной среде.
28. Spring Framework — универсальный фреймворк с открытым исходным кодом для Java-платформы.
29. Spyder – свободная и кроссплатформенная интерактивная IDE для научных расчетов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части.
30. SQL – реляционная база данных – база, где данные хранятся в формате таблиц, строго структурированных и связанных друг с другом.
31. XML-RPC — стандарт/протокол вызова удалённых процедур, использующий XML для кодирования своих сообщений и HTTP в качестве транспортного механизма.
32. Бэкенд – программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части.
33. Джава – транскрипция слова Java
34. Кубит – простая реализация паттерна BLoC.
35. Неоморфизм — это свежий (относительно skeuo/flat/material) тренд в дизайне. Он использует размытие, угол и интенсивность тени объекта для его выделения.
36. ПО (Програ́ммное обеспе́чение) — программа или множество программ, используемых для управления компьютером.
37. Фронтенд – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Содержание

[Глоссарий 3](#_Toc88414838)

[Введение 8](#_Toc88414839)

[1. Основная часть 9](#_Toc88414840)

[1.1. Анализ существующих средств решения проблемы 9](#_Toc88414841)

[1.2. Выбор и обоснование технологий 11](#_Toc88414842)

[1.3. Разработка архитектуры приложения на основе выбранного паттерна 13](#_Toc88414843)

[1.4. Разработка серверной части интернет-ресурса 17](#_Toc88414844)

[1.5. Разработка клиентской части интернет-ресурса 19](#_Toc88414845)

[1.6. Тестирование 23](#_Toc88414846)

[Заключение 24](#_Toc88414847)

[2. Список литературы 25](#_Toc88414848)

[3. Приложение 28](#_Toc88414849)

# Введение

Чтобы определить цель данной курсовой работы, нужно рассмотреть проблему: непонятно, как удобно хранить статистику в онлайн формате с возможностью коллаборации с коллегами. Поэтому целью данной курсовой работы является разработка сервиса для собственных нужд по причине того, что существующие решения не позволяют реализовать желаемое.

Актуальность данной исследовательской и практической работы неоспорима, поскольку появление новых сервисов на рынке цифровых и информационных технологий это естественная вещь, которая позволяет за счет конкуренции выявлять слабые места у имеющихся продуктов, а значит, открывать новые границы для развития и совершенствования.

Объектом и предметом исследования данной курсовой работы является веб-приложение, обладающее возможностью удобного использования формул, создания кастомных функций, обеспечивающее интуитивную работу сводных таблиц.

Предполагаемым результатом является отечественный сервис, обладающий технической базой для расширения и полной кастомизации под нужды пользователя, вне зависимости от глобальной ситуации и возможных санкций. Также важно упомянуть про необходимость наличия четко разграниченных ролей и уровней доступа, защищенное хранение конфиденциальной информации должно быть обеспечено с целью предотвращения получения доступа к аккаунтам и данным пользователей при утечке содержимого баз данных.

1. Основная часть
   1. Анализ существующих средств решения проблемы

Самое популярное решение – это Excel от компании Microsoft, но у него присутствует два серьёзных недостатка: данный сервис платный и в нем отсутствует возможность совместной работы посредством сети Интернет.

Есть возможность отказаться от покупки лицензии, воспользовавшись аналогичным, но беслпатным ПО Libre Office, однако тогда будет по-прежнему отсутствовать возможность коллаборации с другими людьми.

Существует менее популярные корпоративные решения такие как RowShare, но они не подходят для решения нашей задачи по причине дороговизны.

Самым популярным решением, близким к решению задачи, является Google Docs. Однако, инструментарий данного сервиса весьма узок. Существующий функционал сводных таблиц не позволяет реализовать сложные пользовательские функции. Нет возможности подгрузки данных из внешнего источника, кроме Google Forms, однако они, в свою очередь, не позволяют реализовать произвольные типы данных, что сильно сокращает возможности аналитики. Кроме того, в Google Docs отсутствует возможность сложной сортировки данных (например, сортировки по нескольким параметрам или с динамическим отсевом данных). Наконец, не ясно, как экспортировать данные в удобном для конкретной задачи формате.

В связи с тем, что решение проблемы отсутствует, было принято решение сделать собственный веб-сервис, который удовлетворит следующим требованиям: он должен иметь собственный инструментарий для генерации таблиц, включающий в себя редактирование ячеек и форматирование, и собственную реализацию хранения данных, которая позволить загружать, хранить и экспортировать статистику в удобном нам формате. Таким образом, будет создана платформа, на которой станет возможным создание функционала, недоступного в конкурирующих сервисах.

**Выводы к первой главе**

Был проведен анализ предметной области, рассмотрены преимущества и недостатки готовых решений, представленных на рынке цифровых технологий, сформированы основные требования к создаваемому в рамках данной курсовой работы программному продукту.

* 1. Выбор и обоснование технологий

Среди различных паттернов проектирования наиболее популярными являются MVC, DDD и Clear Architecture. Второй вариант не подходит из-за отсутствия сложной бизнес-логики, а третий из-за отсутствия необходимости в комплексных unit-тестах для решения данной задачи, поэтому был выбран первый вариант, поскольку он идеально подходит для небольших проектов, одним из который и является данная исследовательская работа.

Выбор между реляционными и нереляционными базами данных в конечном итоге пал на нереляционные (NoSQL), поскольку разрабатываемый сервис в теории имеет возможность для хранения данных любого типа и добавления новых в процессе работы, что является решающим фактором при выборе архитектуры хранения данных. Также не стоит забывать про преимущества в производительности и масштабируемости избранного вида баз данных, что позволяет быстрее читать данные в распределенной среде и хорошо масштабировать БД горизонтально.

Поскольку разрабатывается web-сервис, то стоит упомянуть про архитектуру REST для взаимодействия с данными, отправляемыми или получаемыми в запросах. Данная технология очень проста в плане использования, а также позволяет передавать данные без применения дополнительных слоев и обертывания данных в различные форматы (в отличие от XML-RPC или SOAP), позволяет серьезно упростить поставленные в рамках данного проекта задачи.

В качестве языка для написания контроллеров для серверной части веб-ресурса был выбран вышедший в 1995 году язык Java как наиболее удобный, надежный и по причине того, что именно он изучается на дисциплине «Архитектуры клиентских частей интернет-ресурсов» в этом семестре.

Вместе с Java используется фреймворк Spring Boot для построения API структуры. Чтобы автоматически и наиболее эффективно собирать приложение на Джаве, используется Gradle – система автосборки, построенная на принципах Apache Ant и Apache Maven, но усовершенствованная в сравнении с исходными технологиями. Ради сокращения кода за счет использования аннотаций к функциям и классам и упрощения написания кода в целом используется библиотека Lombok.

Для контейнеризации и возможности успешного развертывания веб-приложения на различных системах и в различных средах используется ПО Docker.

Управлением базами данных заведует документно-ориентированная система MongoDB, поскольку не требует описания схемы таблиц и является наиболее предпочтительной и даже классической среди NoSQL-систем.

Разработка клиентской части всецело основана на технологии Flutter, к причинам выбора именно этой технологии можно причислить: в Flutter не используются нативные компоненты, поэтому не требуется писать дополнительные прослойки для коммуникации с ними; Flutter не использует JavaScript ни в каком виде, что сказывается на скорости выполнения операций, для языка программирования фреймворка Flutter был выбран Dart.

Полный процесс разработки происходил в интегрированной среде разработки ПО IntelliJ IDEA от компании JetBrains, выбор сделан из личных предпочтений, надежности, удобства и быстродействия среды.

**Выводы ко второй главе**

В результаты работы по данной главе был проведен подробный анализ возможных технологий, паттернов проектирования и обоснован выбор технологий и того подхода, который будет наиболее оптимален для заданных условий.

* 1. Разработка архитектуры приложения на основе выбранного паттерна

В данной работе используется паттерн BLoC, который помогает хранить текущее состояния и отслеживать изменения этих состояний посредством контроллеров. Для хранения некоторых состояний используется Cubit – реализация паттерна BLoC. Принцип работы паттерна BLoC проиллюстрирован на рисунке 1.

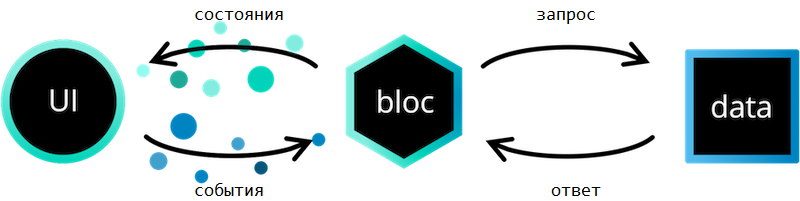
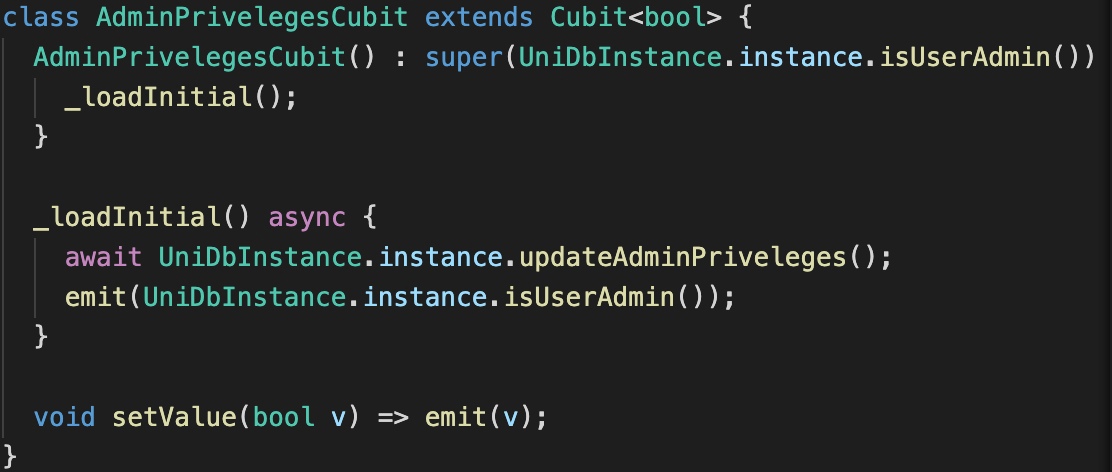


Рисунок 1 – Скриншот принципа работы паттерна BLoC

Для хранения состояния авторизации реализован Cubit прав администратора, что можно увидеть на рисунке 2. Он хранит информацию об авторизации и может независимо от остальной логики приложения проверить наличие прав администратора.

Рисунок 2 – Скриншот кода кубита хранения информации о таблице

Бекенд построен на архитектуре RESTful API. Фреймворк spring boot её реализует в том числе благодаря выделения контроллеров, которые представляют методы API. На представленном ниже рисунке 3 можно видеть код класса, запускающий Spring Boot приложение.

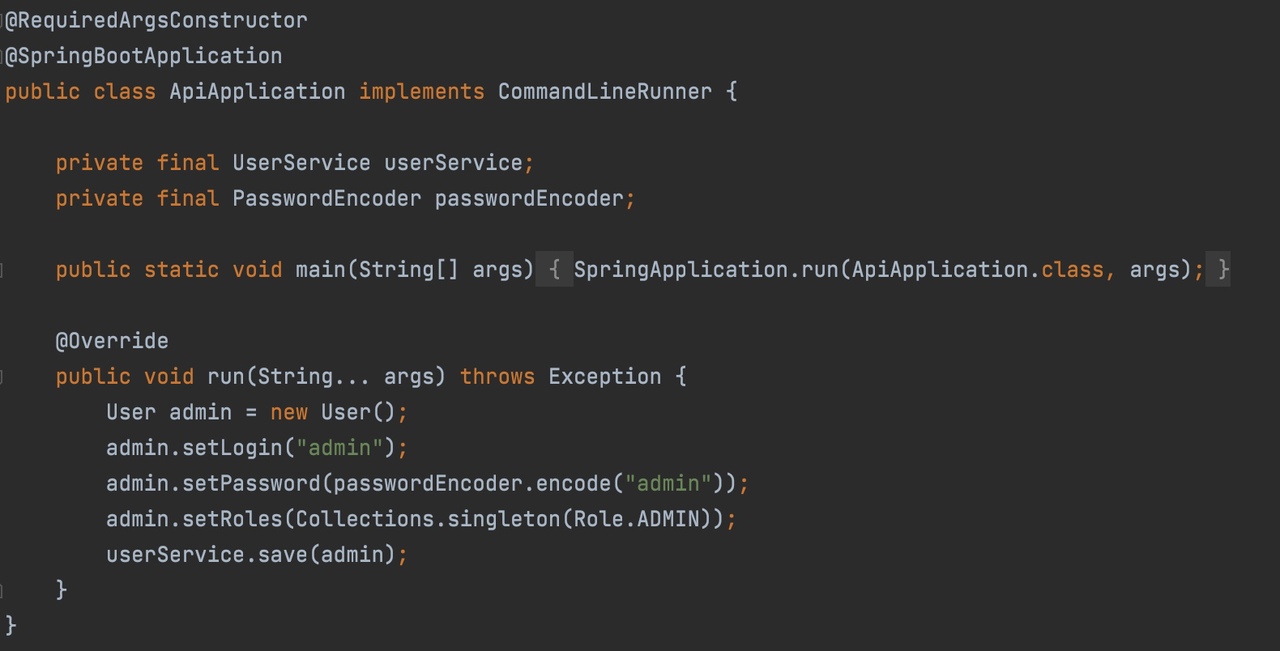


Рисунок 3 – Скриншот кода класса Spring Boot приложения

MVC – это архитектура отделения внешних модулей от рабочих механизмов, разделения задач и ролей программных блоков. Код на Flutter реализует её. На рисунке 4 можно видеть структуру директорий проекта, организованную по выбранной архитектуре.

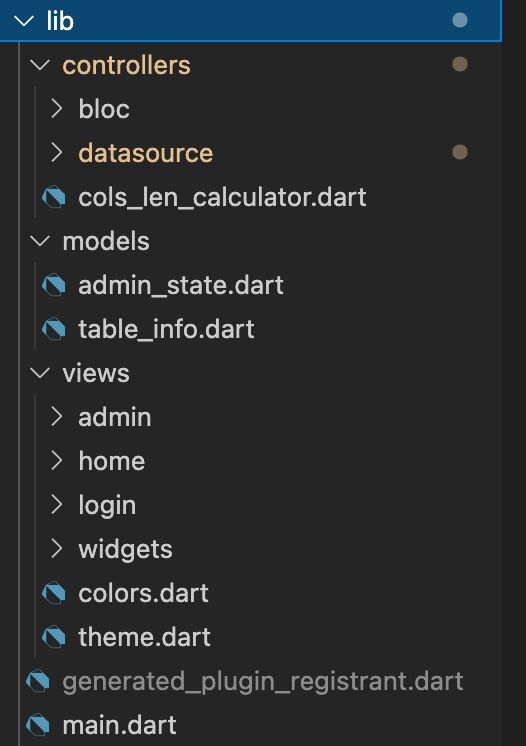


Рисунок 4 – Скриншот директория проекта, структурно отражающий архитектуру MVC

Конфигурация подразумевает связку с базой данных MongoDB, для этого в файл Docker-compose прописаны соответствующие настройки, отображенные на рисунке 5.

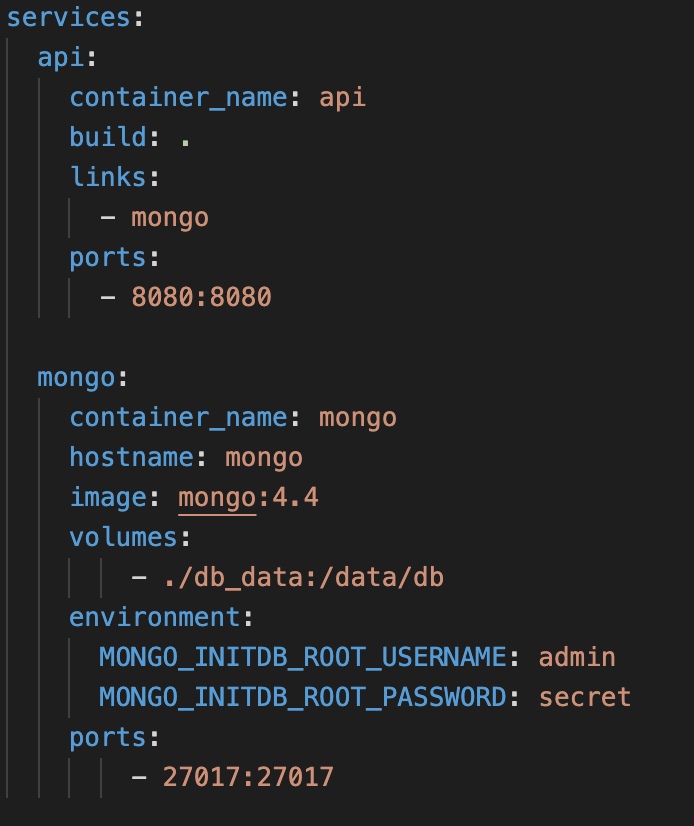


Рисунок 5 – Скриншот содержимого файла Docker-compose

**Выводы к третьей главе**

Было детально описано применение выбранного паттерна проектирования и других ключевых моментов для разработки архитектуры создаваемого программного продукта.

* 1. Разработка серверной части интернет-ресурса

Как уже упоминалось выше, бекенд реализован на фреймворке Spring Boot. Созданы контроллеры, репозитории и сервисы для работы с пользователями и таблицами. Контроллеры представляют методы API. Для каждой сущности реализована группа CRUD, то есть методы добавления, чтения, обновления и удаления записей.

В частности, для работы c таблицами был создан следующий контроллер, исходный код которого приведён на рисунке 6.

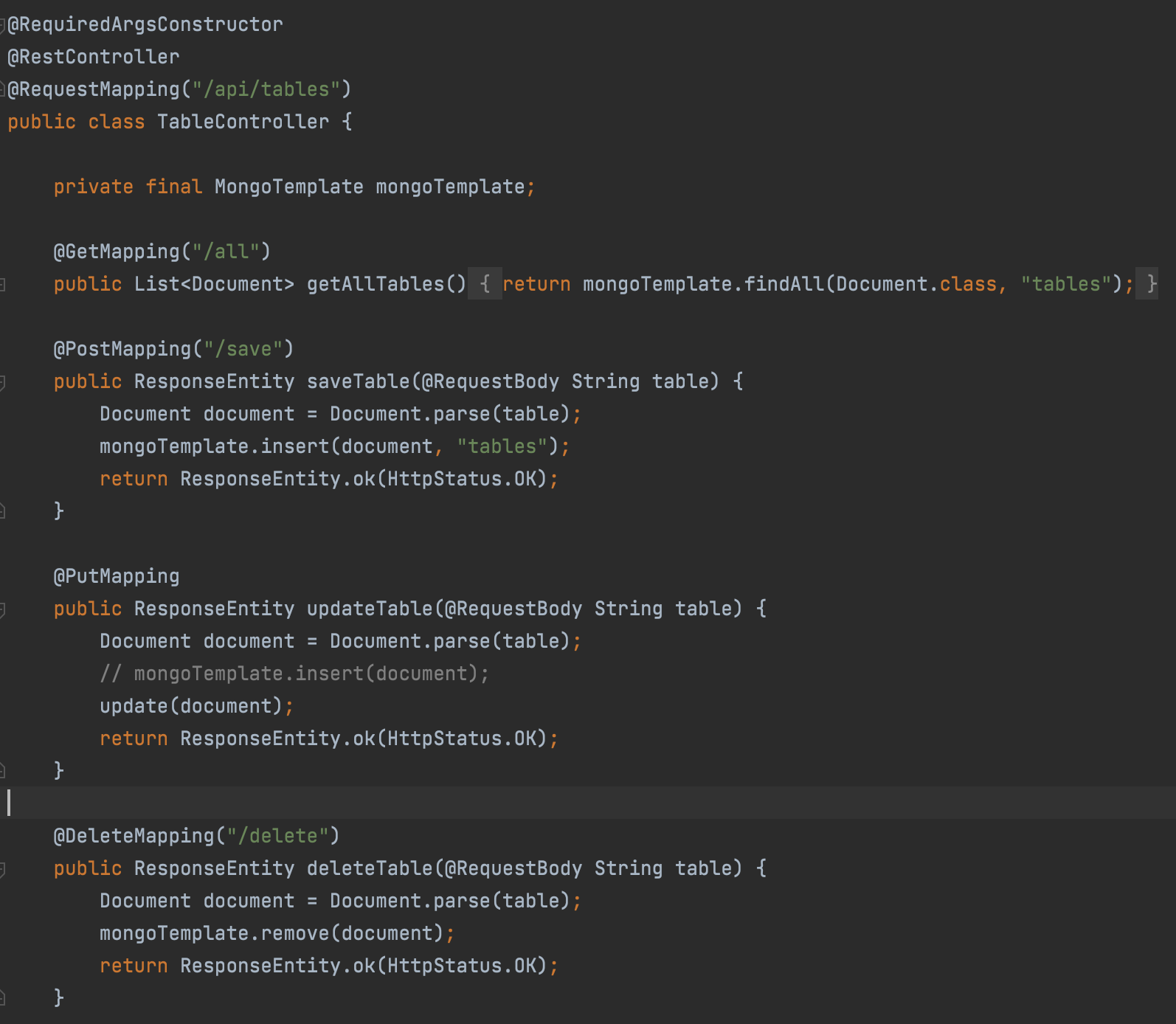


Рисунок 6 - Скриншот кода сервиса поиска пользователя в БД

Для защиты API от несанкционированного доступа были выполнены соответствующие настройки, запрещающие доступ к запросам данных без авторизации. Исходный код настроек приведён на рисунке 7. Хранение паролей пользователей в открытом виде противоречит базовым понятиям о безопасности данных, поэтому предусмотрено шифрование паролей, механизм которого приведён на рисунке 8.

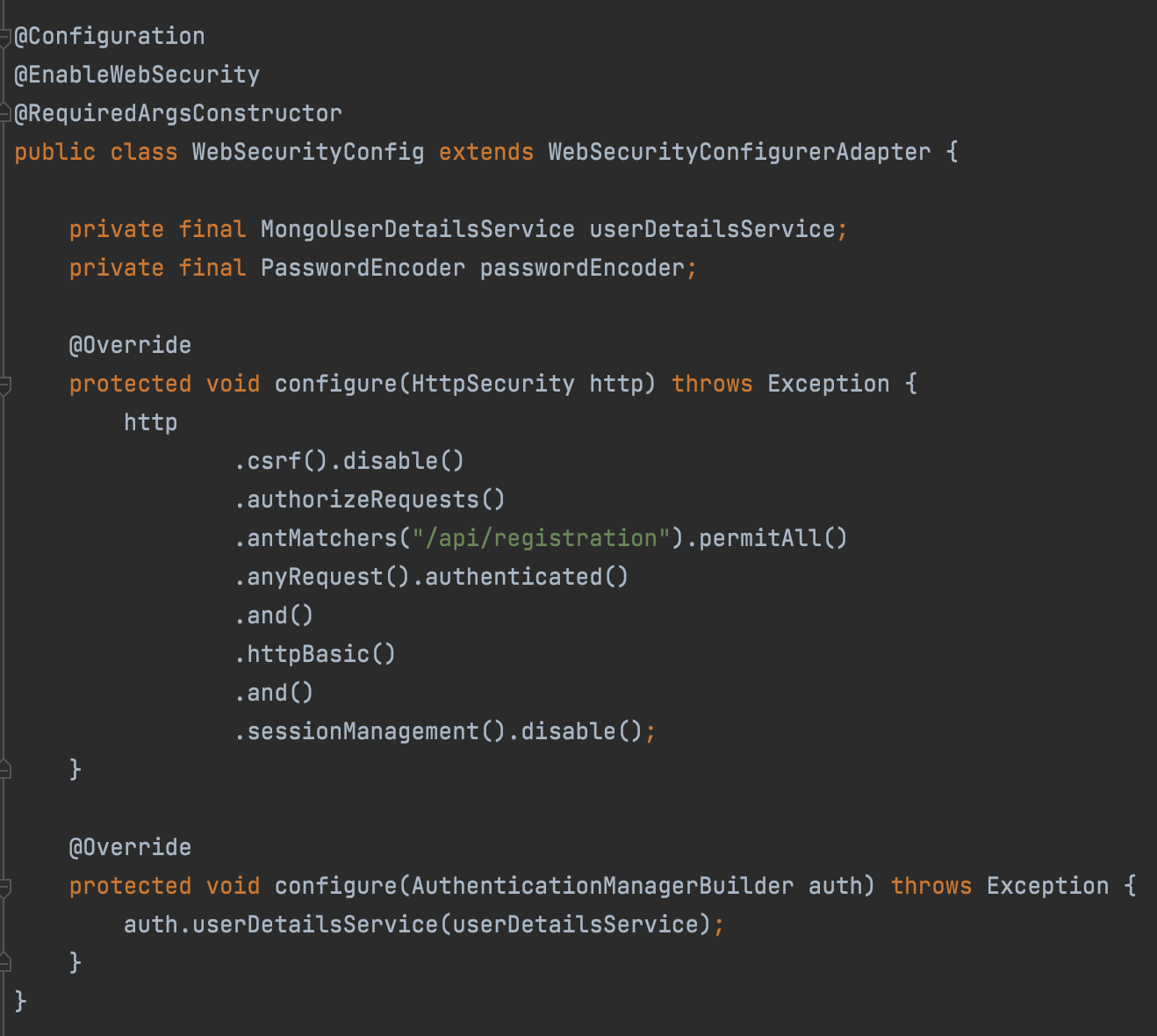
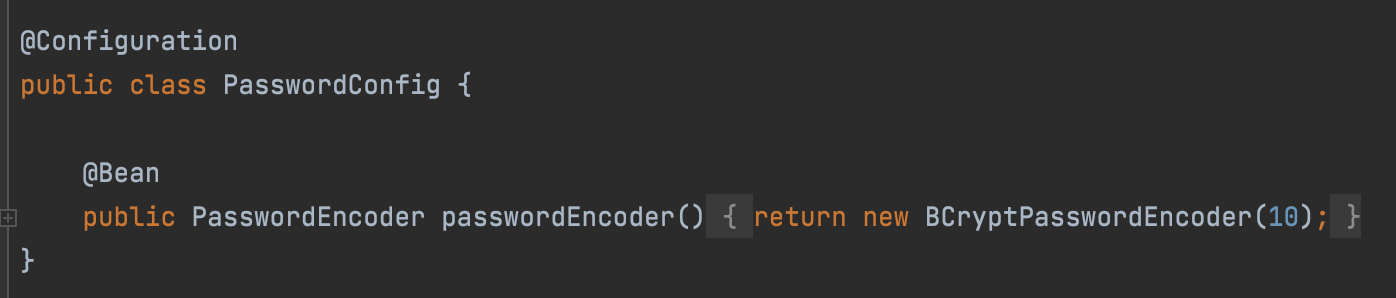


Рисунок 7 – Скриншот настроек безопасности

Рисунок 8 – Скриншот настроек шифрования паролей пользователей

Вся конфигурация может быть удобно развёрнута благодаря настройке Docker контейнера, представленного на рисунке 9.

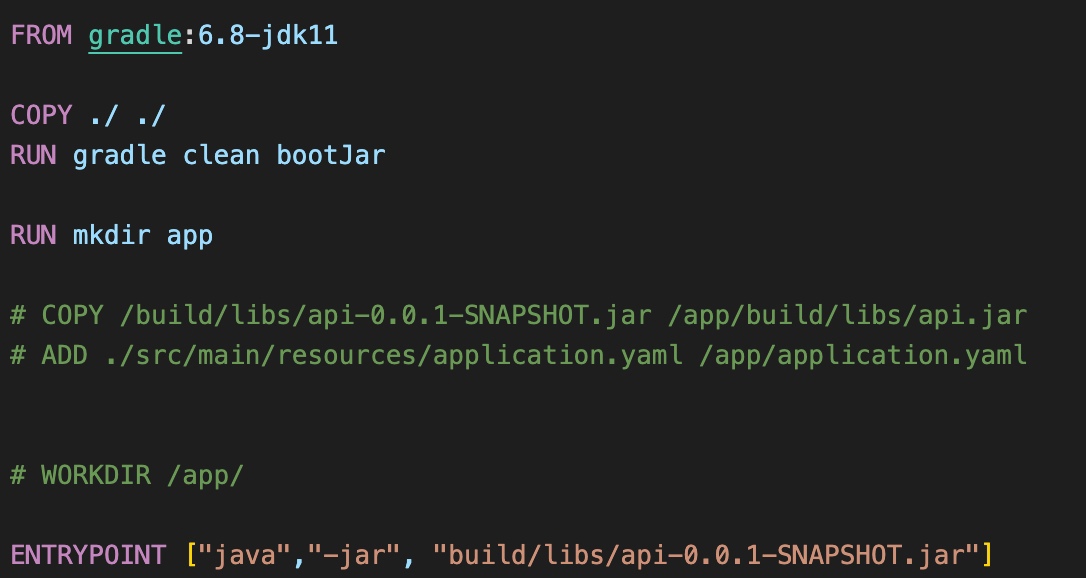


Рисунок 9 – Скриншот кода Dockerfile для развертывания Java приложения

**Выводы к четвертой главе**

По выполнении четвертой главы были описаны детали разработки серверной части разрабатываемого интернет-ресурса, представлены наиболее значимые листинги для более четкого, изложены важные сведения касательно бэкенд составляющей программного продукта.

* 1. Разработка клиентской части интернет-ресурса

Были реализованы страницы просмотра таблиц, логина и их редактирования. Использованы обучающие материалы Material Design от Google и стиль неоморфизма.

Все страницы имеют функциональную панель навигации, позволяющую переключить режим, обновить данные из БД и сменить тему, панель навигации показана на рисунке 10.

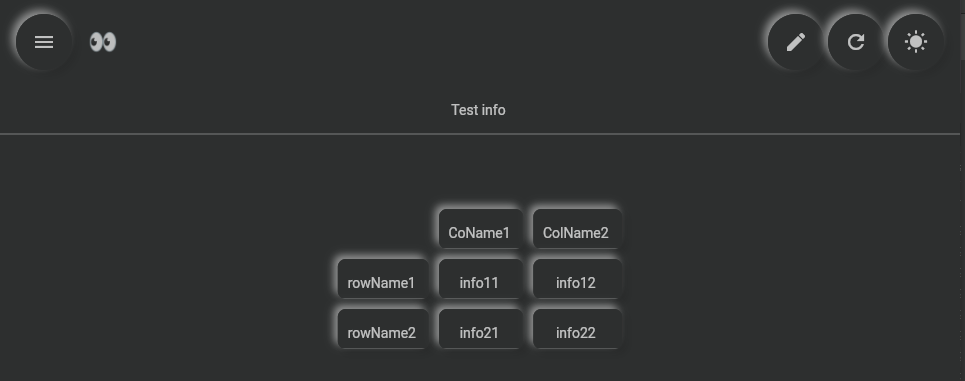


Рисунок 10 – Скриншот навигационной панели в режиме простого пользователя

Страница просмотра, представленная на рисунке 10, позволяет видеть список скачанных таблиц. Есть возможность вызвать боковое меню, показанное на рисунке 11, чтобы выбрать другую таблицу для просмотра.

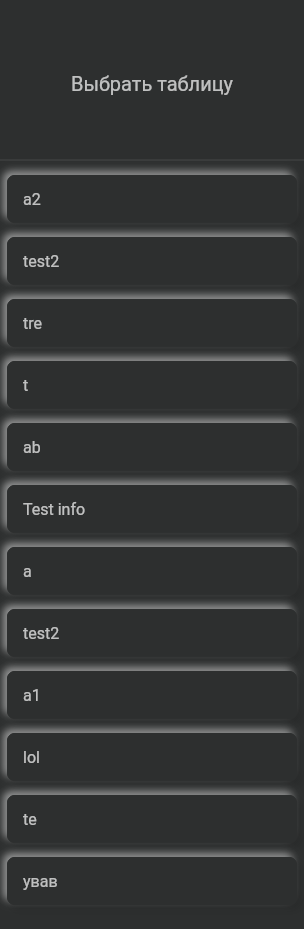


Рисунок 11 – Скриншот боковой панели просмотра скачанных таблиц

При нажатии на кнопку смены режима приложение проверит авторизацию пользователя. Если он не авторизован, то будет отправлен на страницу авторизации, показанную на рисунке 12, которая содержит стандартную форму входа. Предусмотрена возможность логина и регистрации. В случае успешной авторизации пользователь попадает на экран редактирования таблиц, представленный на рисунке 13. Он отличается тем, что появляются соответствующие кнопки для создания новых строк, столбцов, а также меню управления таблицей из- 3-х кнопок:

1. удалить изменения,
2. автоматически форматировать ширину столбцов,
3. сохранить изменения.

Кроме того, появляется возможность изменить имя таблицы. Если у текущего пользователя есть права администратора, он сможет сохранить изменения в БД, в противном случае сохранения изменений не произойдет.

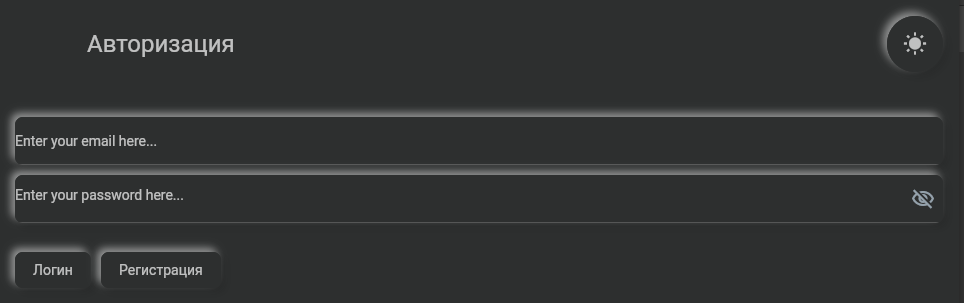


Рисунок 12 – Скриншот окна авторизации и регистрации



Рисунок 13 – Скриншот экрана редактирования таблиц

**Выводы к пятой главе**

По выполнении данной главы были описаны все важные сведения касательно пользовательской стороны разрабатываемого портала, возможности и имеющиеся элементы интерфейса, приведено описание содержимого страниц с соответствующими иллюстрациями, показывающими внешнюю составляющую сервиса.

* 1. Тестирование

С целью проверки работоспособности и верного поведения создаваемого приложения был использован Postman. Данный инструмент позволяет отправить запросы со стороны клиента на сервер и получить ответ от сервера обратно клиенту в различном текстовом представлении на выбор тестировщика.

Проверка запроса на регистрацию представлена на рисунке 14.

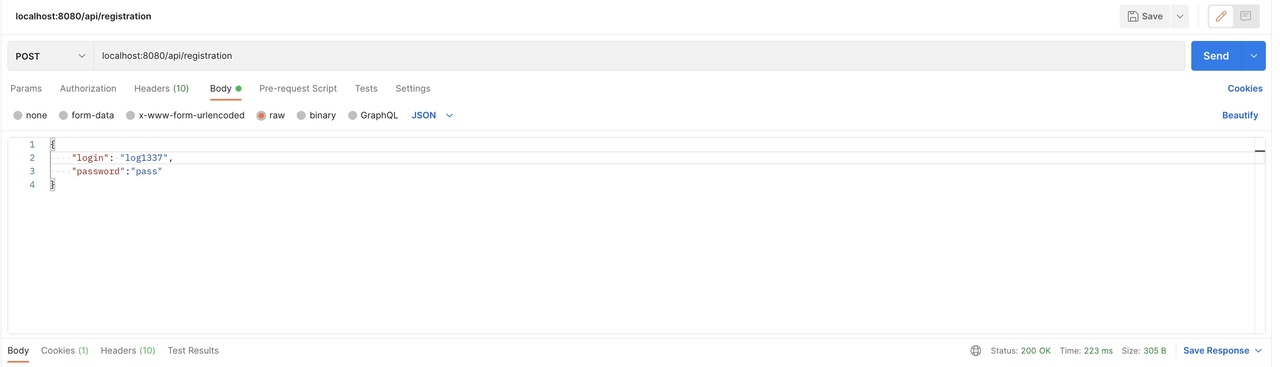
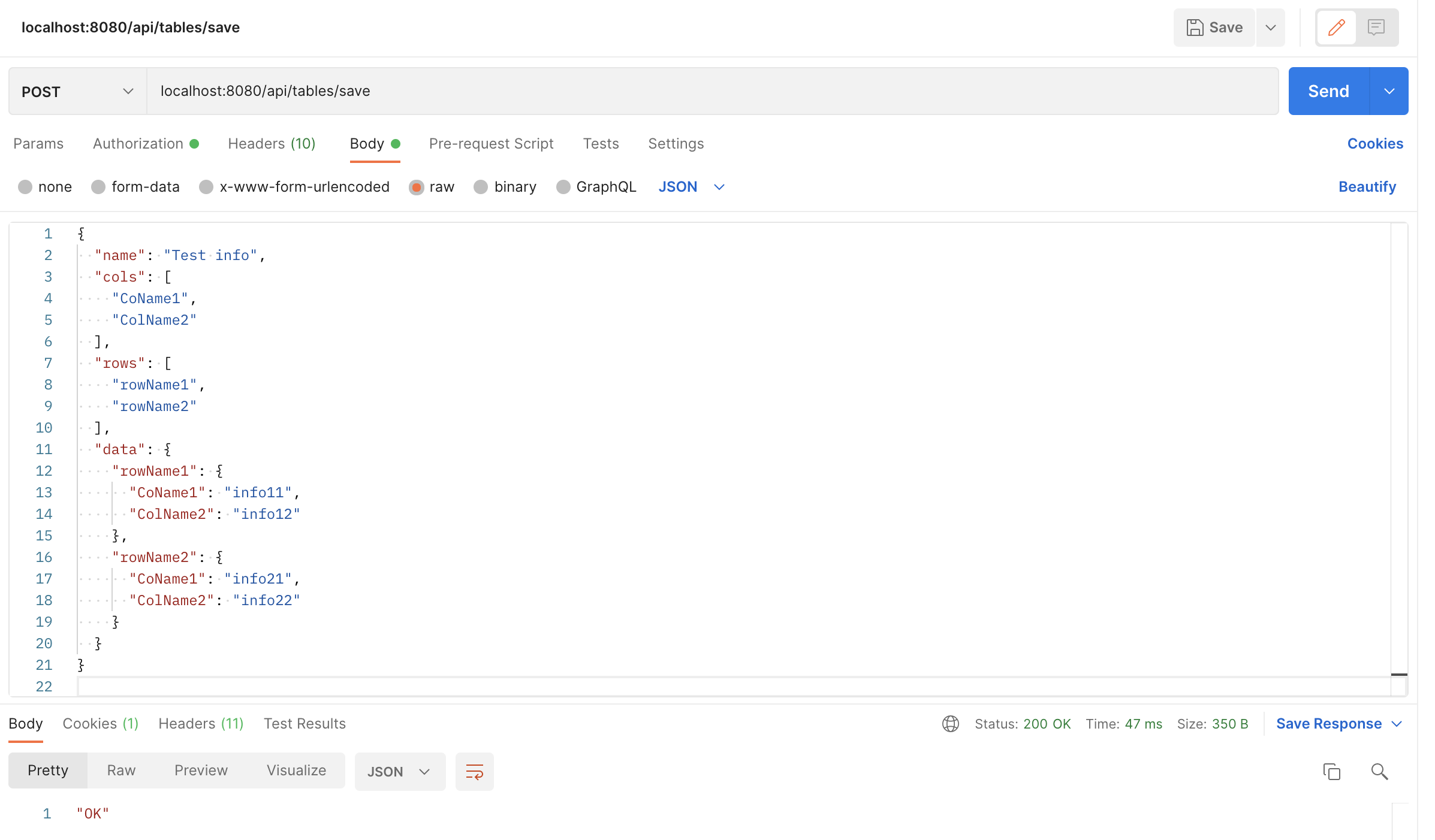
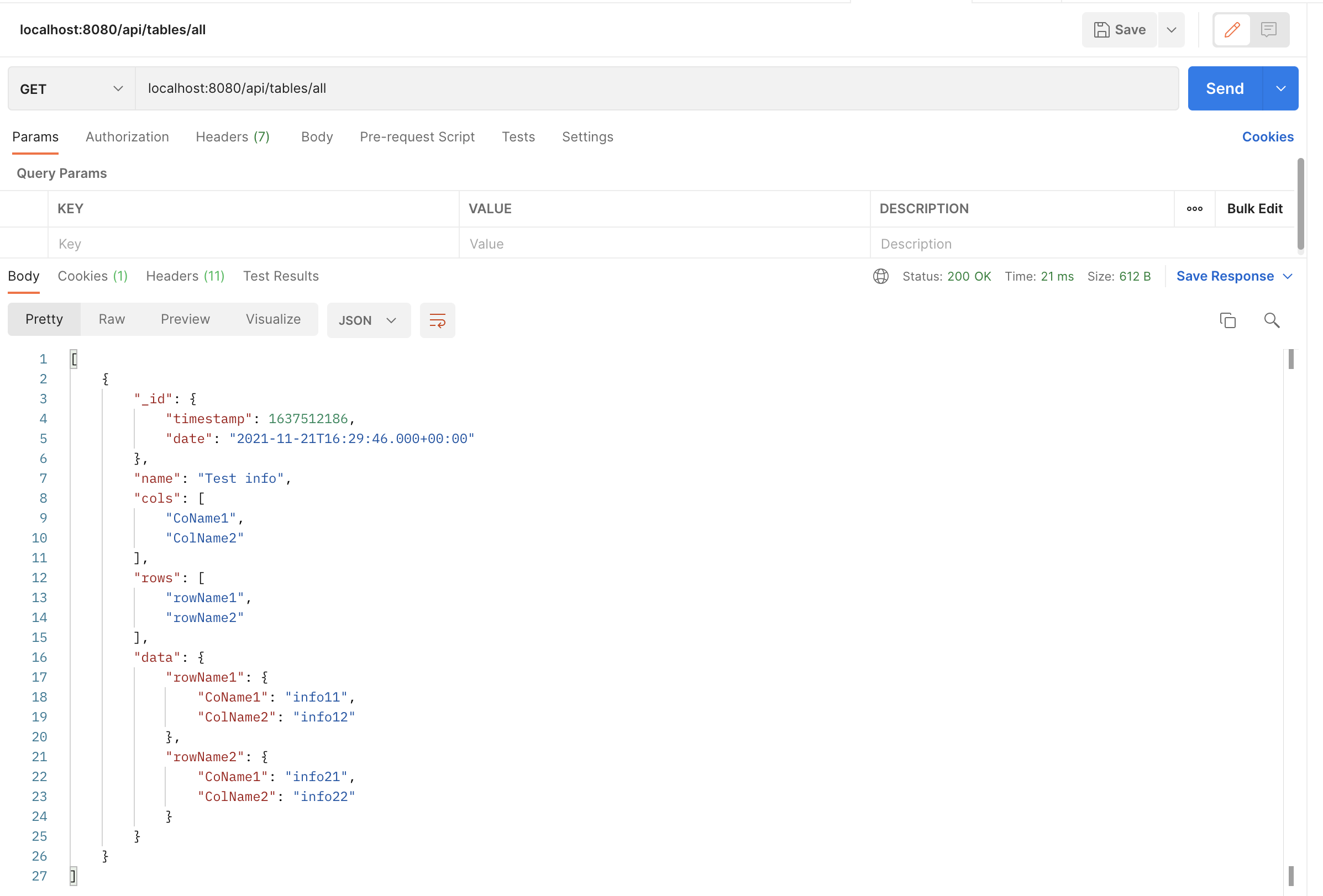


Рисунок 14 – Скриншот успешной проверки отправки запроса на регистрацию

Проверка запроса на создание таблицы представлена на рисунке 15.

Рисунок 15 – Скриншот успешной проверки запроса на создание таблицы

Проверка запроса на получение таблиц представлена на рисунке 16.

Рисунок 16 – Скриншот успешной проверки запроса на получение таблиц

**Выводы к шестой главе**

Разрабатываемое приложение было успешно проверено с помощью Postman, в результате чего можно сделать вывод о качественной и надежной совместной работе и связи клиентской и серверной частей веб-приложения, возможности использования данного приложения реальными пользователями без их страха встретить ошибки и потерять личные данные.

Заключение

В итоге выполнения данной исследовательской работы было получено функционирующее веб-приложение, решающее поставленную в начале работы проблему и позволяющее хранить статистику в онлайн формате с возможностью совместного редактирования с коллегами. Несомненно, инструментарий разработанного приложения пока не так полон, как у имеющихся конкурентов, но зато имеет место быть возможность расширения функционала посредством установки дополнительных модулей и программных обновлений, расширяющих функционал. Кроме того, был получен опыт разработки веб-приложений, получено много теоретических и практических знаний в данной области, что, несомненно, является огромным плюсом для будущего разработчика продуктов на рынке IT.

1. Список литературы
2. Алексеев, В. М. Язык программирования HTML5 : учебно-методическое пособие / В. М. Алексеев. — Москва : РУТ (МИИТ), 2019. — 159 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/175604 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
3. Альтман, Е. А. Платформа IntelliJ IDEA для разработки программ : учебно-методическое пособие / Е. А. Альтман, А. В. Александров, Т. В. Васеева. — Омск : ОмГУПС, 2020. — 37 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/165620 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
4. Баккет, К. Dart в действии : учебное пособие / К. Баккет. — Москва : ДМК Пресс, 2013. — 528 с. — ISBN 978-5-94074-918-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/73063 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
5. Беленькая, М. Н. Администрирование в информационных системах : учебное пособие / М. Н. Беленькая, С. Т. Малиновский, Н. В. Яковенко. — 2-е изд., испр. и доп. — Москва : Горячая линия-Телеком, 2018. — 408 с. — ISBN 978-5-9912-0418-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/176120 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
6. Григорьев, Ю. А. Реляционные базы данных и системы NoSQL : учебное пособие / Ю. А. Григорьев, А. Д. Плутенко, О. Ю. Плужникова. — Благовещенск : АмГУ, 2018. — 424 с. — ISBN 978-5-93493-308-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/156492 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
7. Даева, С. Г. Основы системного администрирования и администрирования СУБД : учебно-методическое пособие / С. Г. Даева. — Москва : РТУ МИРЭА, 2021. — 75 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/171547 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
8. Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2019. — 188 с. — ISBN 978-5-8114-3822-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/122174 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
9. Дэвид, Х. Java EE 7 и сервер приложений GlassFish 4 : учебное пособие / Х. Дэвид ; перевод с английского А. Н. Киселев. — Москва : ДМК Пресс, 2016. — 332 с. — ISBN 978-5-97060-332-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/69962 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
10. Заметти, Ф. Flutter на практике : руководство / Ф. Заметти ; перевод с английского А. С. Тищенко. — Москва : ДМК Пресс, 2020. — 328 с. — ISBN 978-5-97060-808-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179464 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
11. Ларина, Т. Б. Администрирование операционных систем. Управление системой : учебное пособие / Т. Б. Ларина. — Москва : РУТ (МИИТ), 2020. — 71 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/175980 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
12. Лоре, А. Проектирование веб-API : руководство / А. Лоре ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 440 с. — ISBN 978-5-97060-861-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179498 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
13. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва : СОЛОН-Пресс, 2007. — 336 с. — ISBN 5-93455-134-5 . — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/13633 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
14. Садыков, А. М. Методы разработки веб-приложений : учебно-методическое пособие / А. М. Садыков. — Иваново : ИГЭУ, 2019. — 72 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/154584 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
15. Эделман, Д. Автоматизация программируемых сетей : руководство / Д. Эделман, С. С. Лоу, М. Осуолт ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2019. — 616 с. — ISBN 978-5-97060-699-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123708 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
16. Эрик, Р. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Р. Эрик, Р. У. Джим. ; под редакцией Ж. Картер ; перевод с английского А. А. Слинкин. — Москва : ДМК Пресс, 2013. — 384 с. — ISBN 978-5-94074-866-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/58690 (дата обращения: 17.11.2021). — Режим доступа: для авториз. пользователей.
17. Приложение