Содержание

[Введение 3](#_Toc97032687)

[1 Сбор, анализ и формирование требований к программному продукту 4](#_Toc97032688)

[1.1 Сбор требований 4](#_Toc97032689)

[1.2 Анализ и формирование требований 5](#_Toc97032690)

[1.2.1 Общие требования 5](#_Toc97032691)

[1.2.2 Требования к функциональным характеристикам 6](#_Toc97032692)

[1.2.3 Требования к надежности 7](#_Toc97032693)

[1.2.4 Требования к квалификации и численности персонала 8](#_Toc97032694)

[2 Проектирование и разработка архитектуры программного продукта .. 9](#_Toc97032695)

[2.1 Построение диаграммы связей 9](#_Toc97032696)

[2.2 Разработка сценария использования 9](#_Toc97032697)

[2.3 Архитектура программного продукта 11](#_Toc97032698)

[2.4 Выбор СУБД и разработка базы данных 14](#_Toc97032699)

[2.5 Прототипирование и дизайн программного продукта 17](#_Toc97032700)

[3 Разработка программного продукта 18](#_Toc97032701)

[3.1 Инструментальные и программные средства разработки 18](#_Toc97032702)

[3.2 Календарный план разработки 22](#_Toc97032703)

[4 Тестирование программного продукта 23](#_Toc97032704)

[4.1 Выбор метода обеспечения качества 23](#_Toc97032705)

[4.2 Тестирование программного продукта 23](#_Toc97032706)

[5 Разработка документации на программный продукт 27](#_Toc97032707)

[Заключение 28](#_Toc97032708)

[Список литературы 29](#_Toc97032709)

[Приложение А 30](#_Toc97032710)

[Приложение Б 46](#_Toc97032711)

[Приложение В 47](#_Toc97032712)

# Введение

Целью курсовой работы является реализация web-проекта, которым в дальнейшем будут пользоваться абитуриенты колледжей и институтов.

Абитуриенты хотят читать книги в общем доступе, при этом имея доступ к регистрации личного кабинета пользователя, а также создания своих личных подборок книг, которыми они бы могли обмениваться друг с другом.

Задача команды в проекте была реализовать удобный web-сервис для того чтобы абитуриенты могли без труда найти любой их интересующий материал, который был бы скомпонован в одном месте и был бесплатным.

# 1 Сбор, анализ и формирование требований к программному продукту

## 1.1 Сбор требований

На этапе сбора требований необходимо составить вопросы для заказчика и сформировать их в таблицу-опросник.

Таблица 1 – Интервью с заказчиком

|  |  |
| --- | --- |
| Вопросы | Ответы |
| Какая информация должна быть размещена на главной странице? | Категории книг, книги, обратная связь. |
| Требования к дизайну | Не перегруженность, зрительная  восприимчивость представляемой  информации. |
| Функционал сайта | Личный кабинет, обратная связь, подборка книг, фильтр по категориям. |
| Кто будет пользоваться web-продуктом? | Абитуриенты, преподаватели, администратор |
| Кто будет иметь возможность редактировать пароли пользователей? | Только администратор (superUser) |
| Какие возможности должны быть реализованы на сайте для администратора? | Возможность редактировать и добавлять книги, а также полное редактирование контента. |
| Какими возможностями будут обладать посетители сайта? | Возможность прочтения книг, составлять подборки книг и обратится за помощью при не работоспособности сайта или отдельных функций. |
| Есть ли системные требования к оборудованию? | Конкретных системных требований к оборудованию нет. |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Вопросы | Ответы |
| Кто из сотрудников будет иметь доступ к админ-панели? | Frontend develop, backend developer, HR. |
| Какие данные должны редактироваться в админ панели? | книги и информацию о них, а также прилегающие к ним фото и дата выпуска книги. |

## 1.2 Анализ и формирование требований

### 1.2.1 Общие требования

Web-приложение «Студенческая библиотека» предназначено для компоновки и структуризации информации в одном месте которая будет направлена для прочтения книг пользователями. Деятельность библиотеки также предусматривает регистрации и авторизацию пользователей. При этом пользователи могу создавать и редактировать подборки книг; пользователи проекта могут делать предварительные просмотр подбора книг для создания уже полноценных подборок.

Web-приложение должно предоставлять интерфейс для занесения и обновления в базе данных информации о реализации конечного продукта , о его новостях, пользователях и администраторов, являющихся членами библиотеки.

В состав технических средств, поддерживающих работу программы, должен входить IВМ-совместимый персональный компьютер (ПЭВМ), включающий в себя:

* процессор от Pentium-3.0Hz, не менее;
* оперативную память объемом 4Гигабайт, не менее;
* HDD 40 Гигабайт, не менее;
* операционную систему Windows 7 и выше.

### 1.2.2 Требования к функциональным характеристикам

Web-приложение «Студенческая библиотека» должна обеспечивать возможность выполнения перечисленных ниже функций:

* предоставление продавцу возможности оформить продажу товара, его резервирование и прокат (возврат из проката) – каждая операция оформляется квитанцией;
* предоставление продавцу возможности оформить клиента членом видео-клуба;
* предоставление пользователю возможности быстрого поиска нужной книги (по названию, по категории);
* предоставление Администратору библиотеки полного доступа к хранящейся в базе данных информации с возможностью ее обновления и редактирования;
* предоставление Администратору возможности занесения в базу данных (в библиотеку) новых изданий;

В web-приложении необходимо предусмотреть обратную связи с администрацией, для того чтобы был быстрый отклик на ошибки или багги проекта.

### 1.2.3 Требования к надежности

#### 1.2.3.1 Требования к обеспечению надежного функционирования программы

Разрабатываемое web-приложение должно иметь:

* возможность самовосстановления после сбоев (отключения серверов, сбоев в хостинге и т.д.);
* при авторизации или регистрации должна быть валидация;
* ограничение несанкционированного доступа к данным;
* возможность резервного копирования информационной базы;
* разграничение пользовательских прав;
* исключение несанкционированного копирования (тиражирования) продукта.

Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя при работе с приложением.

#### 1.2.3.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) хостинга, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) сервера, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

### 1.2.4 Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор (менеджер) и продавец.

Системный администратор должен иметь высшее профильное образование. В перечень задач, выполняемых системным администратором, должны входить:

* задача поддержания работоспособности технических средств;
* задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;
* задача установки (инсталляции) программы;
* задача создания резервных копий базы данных.

# 2 Проектирование и разработка архитектуры программного продукта

## 2.1 Построение диаграммы связей

Диаграмма связей программного продукта представляет собой графическую схему взаимодействия объектов (модулей, страниц и т. д.) проектируемого ПП. Схема взаимодействия программных модулей показывает структуру программы, её составные части (модули), отражает связь между ними и показывает, какое место в этой связи занимает база данных. Схема отражает не только подчиненность модулей, но и порядок их вызова или функционирования программы.

## 2.2 Разработка сценария использования

Сценарий использования проектируемого ПП можно разработать с помощью диаграмм вариантов использования (Use-case) и деятельности UML. Диаграмма Use-case разработана с целью определения функционала Windows-приложения «Магазин видеопродукции» для его пользователей, и представлена на рисунке 2.

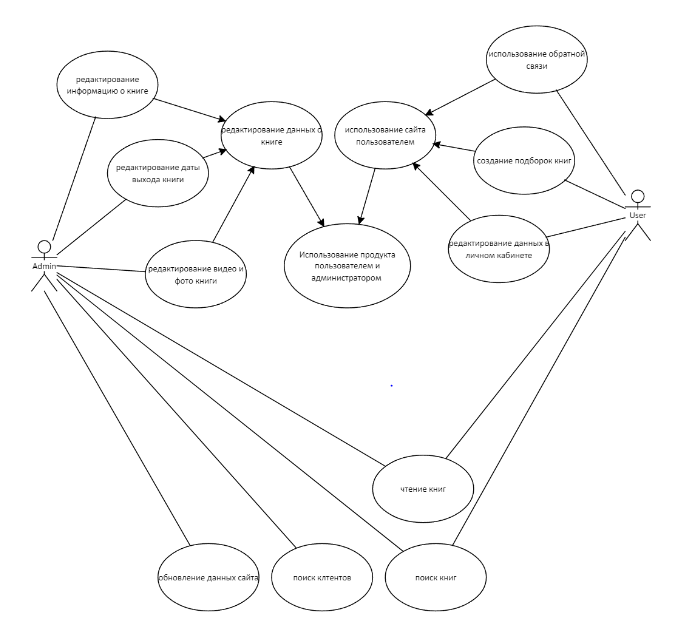


Рисунок 2 ‑ Диаграмма вариантов использования проектируемого ПП

Диаграмма деятельности (активности) UML позволяет более детально визуализировать конкретный вариант использования. Эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

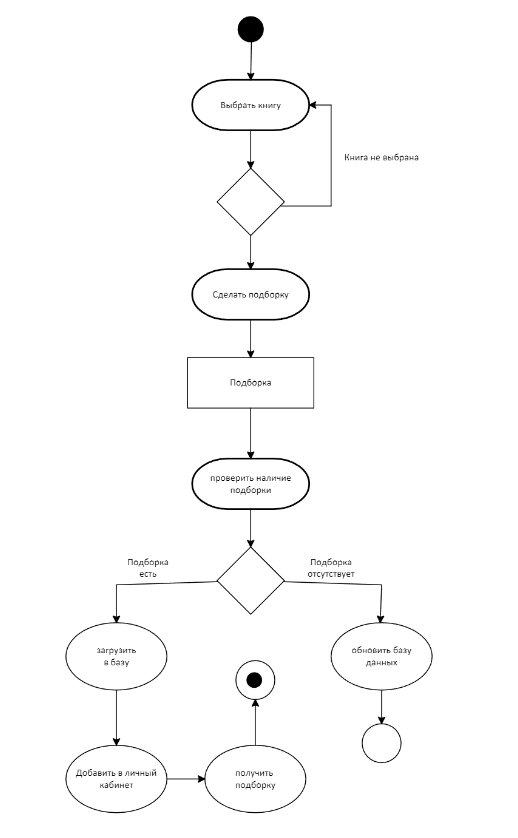


Рисунок 3 ‑ Диаграмма деятельности для функций редактирования справочников БД

## 2.3 Архитектура программного продукта

При проектировании Web-приложения «Студенческая библиотека» необходимо предусмотреть необходимость его распределенной работы на компьютерах, объединенных в локальную сеть и работающих по сети с единой базой данных. Для реализации данной системы предпочтительно использовать технологию ADO.NET и платформу .NET. Данная архитектура обладает возможностью масштабирования, адаптации к любому источнику данных. Архитектура проста в понимании разработчиками ИС и обладает гибким механизмом использования ресурсов.

Схематично архитектура ADO.NET представлена на рисунке 4. Взаимодействие с базой данных в ADO.NET осуществляется через провайдер данных. Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов:

* Connection (устанавливает подключение к источнику данных);
* Command (позволяет выполнять операции с данными из БД);
* DataReader (считывает полученные в результате запроса данные);
* DataSet (хранит данные из БД и позволяет работать с ними независимо от БД);
* DataAdapter (является посредником между DataSet и источником данных).



Рисунок 4 – Архитектура ADO.NET

Почти всё современное программирование построено на принципах ООП и для проектирования внутренней структуры объектно-ориентированного приложения используют диаграмму классов (class diagram), предназначенную для представления программы в виде классов и связей между ними. Диаграмма классов Web-приложения «Студенческая библиотека» представлена на рисунке 5.

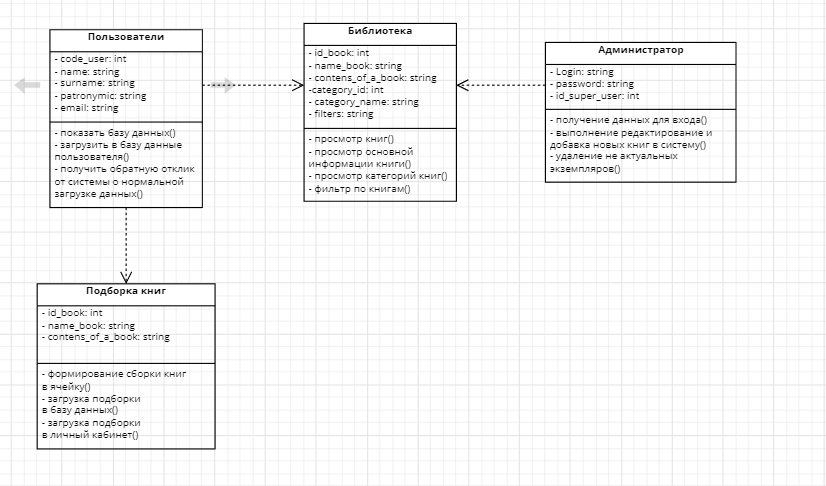


Рисунок 5 – Диаграмма классов проектируемого ПП

## 2.4 Выбор СУБД и разработка базы данных

Для выбора СУБД, в которой будет реализована база данных, необходимо, оценивая СУБД по поставщику, учесть следующие факторы:

* маркетинговую стратегию ‑ умение поставщика выбирать соответствующие целевые рынки, а также организовывать партнерство для расширения маркетинговых возможностей продукта;
* внедрение инноваций ‑ разработку новых технологий, вложения средств в научные исследования, влияние на развитие рынка, способность поставщика внедрить в СУБД новую функциональность;
* географическую стратегию ‑ способность компании использовать свои ресурсы в различных географических регионах, открывать филиалы и организовывать партнерство. Характеристики некоторых СУБД приводятся в таблице 2.

Таблица 2 – Характеристики СУБД

|  |  |
| --- | --- |
| СУБД | Краткая характеристика |
| Microsoft SQL Server | Реляционная СУБД для управления данными в масштабе предприятия, поддерживает технологии XML и Интернет, обладает встроенным средством анализа и извлечения данных, интегрированным с MS Office. ОС ‑ Windows. Используемый язык Transact-SQL, XML |
| Oracle | СУБД для масштабной обработки транзакций (OLTP), хранилищ данных с высокой интенсивностью потока запросов и ресурсоемких Интернет-приложений. ОС ‑ Unix, Windows и Linux. Последняя версия поддерживает Grid-вычисления. Используемые языки Java, Delphi PL/SQL, XML |
| MySQL | Компактная, быстродействующая реляционная СУБД для малых и средних предприятий. ОС ‑ Linux, Mac OS X, Unix и Windows |
| PostgreSQL | Реляционная СУБД, имеет многие возможности, которые реализованы в крупных коммерческих продуктах. ОС ‑ Unix, Windows и NetWare |

Основным минусом СУБД Oracle и Microsoft SQL Server является их высокая стоимость. Кроме того, они являются закрытыми, что ограничивает их возможность интеграции данных. Это не всегда подходит компаниям с высокими требованиями к безопасности. СУБД PostgreSQL поддерживает особенности и возможности традиционных приложений, в то время как СУБД MySQL сосредотачивается на более быстром выполнении веб-приложений.

Для интеграции данных в Web-приложение «Студенческая библиотека» была выбрана СУБД PostgreSQL. Структура спроектированной базы данных приведена в таблицах 3-6. Диаграмма «сущность-связь» (ER-диаграмма) базы данных с указанием сущностей, их атрибутов и связей между сущностями приведена на рисунке 6.

Таблица 3 – Подборка книг

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Код книги | Числовой | Внешний ключ |
| Название книги | Текстовый | Наименование издания |
| Информация | Текстовый | Информация о книге |
| Код пользователя | Числовой | Внешний ключ |
| Код подборки | Числовой | Ключевой ключ |

Таблица 4 – Пользователи

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Код пользователя | Числовой | Ключевой ключ |
| Имя | Текстовый | Наименование пользователя |
| Фамилия | Текстовый | ФИО контактного лица |
| Отчество | Текстовый | Адрес регистрации |
| Код библиотеки | Числовой | Внешний ключ |

Таблица 5 – Администратор

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Код администратора | Числовой | Ключевой ключ |
| Логин | Текстовый | Наименование для входа |
| Пароль | Текстовый | Пароль для входа |

Таблица 6 – Библиотека

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Код библиотеки | Числовой | Ключевой ключ |
| Код администратора | Числовой | Внешний ключ |
| Код книги | Числовой | Внешний ключ |
| Название книги | Текстовый | Наименование издания |
| Информация | Текстовый | Информация о книге |
| Код категории | Числовой | Внешний ключ |
| Фильтр книг | Текстовый | Фильтрация изданий |
| Название категории | Текстовый | Категории библиотеки |

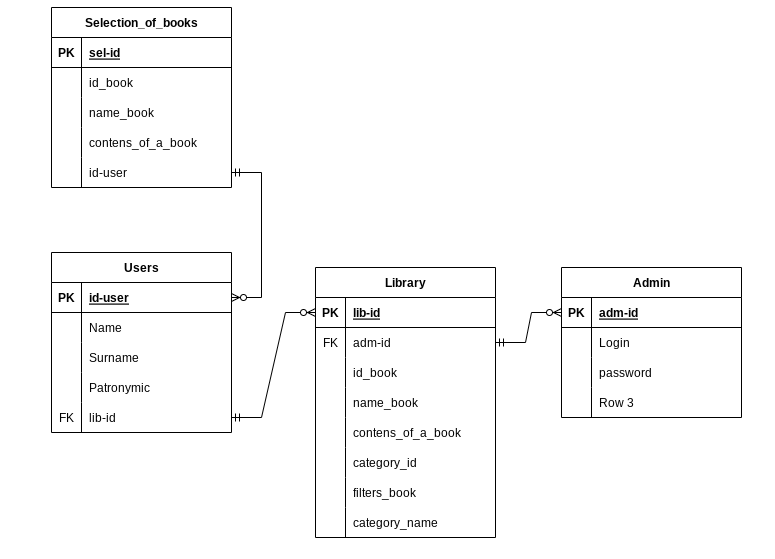


Рисунок 6 – ER-диаграмма базы данных проектируемого ПП

## 2.5 Прототипирование и дизайн программного продукта

Прототип — это набросок продукта, в котором заключены его внешний вид, логика работы и основная функциональность. Для презентации прототипа показывают, к чему и куда приводит взаимодействие будущего пользователя с элементами интерфейса. Связав элементы линиями с другими экранами (формами), на которые попадёт пользователь, получим пользовательские сценарии использования приложения, или user flow (карту навигации).

В будущем эта карта пригодится тестировщикам для сопоставления с рабочим приложением, чтобы проверить, не потерялось ли какое-то действие, не нарушена ли логика.

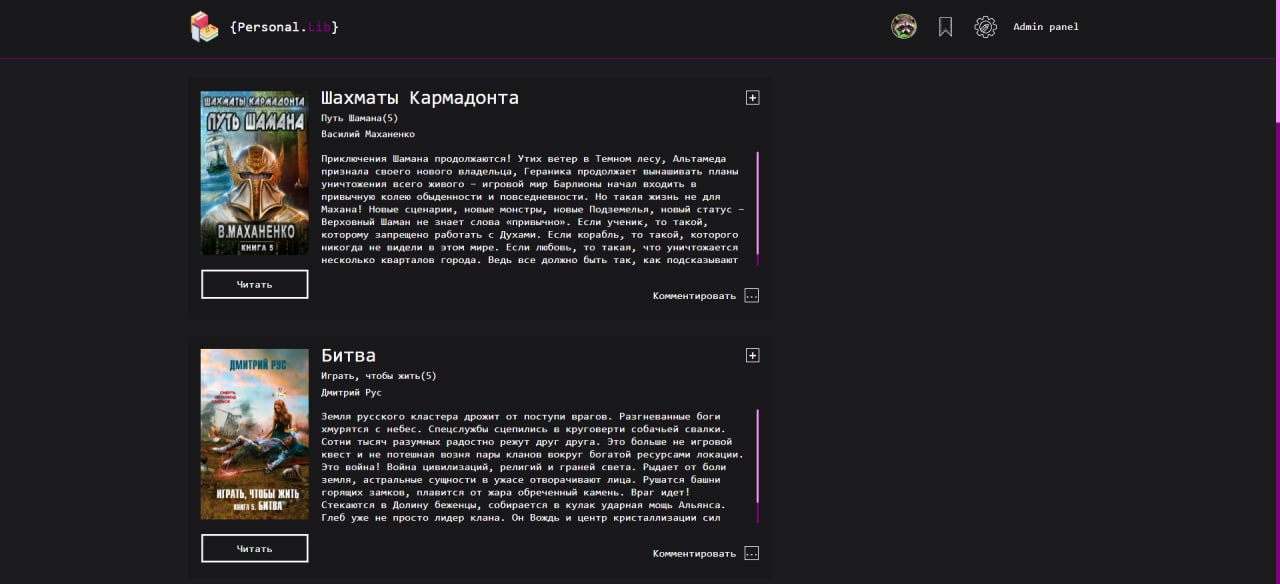


Рисунок 7 – Главная страница студенческой библиотеки.



Рисунок 8 – Панель Администратора для редактирования

изданий.

# 3 Разработка программного продукта

## 3.1 Инструментальные и программные средства разработки

Для разработки программного продукта были использованы следующие инструментальные и программные средства:

* Microsoft Visio ‑ интуитивный редактор для создания организационных UML-диаграмм, чертежей и комплексных блок-схем в нативном конструкторе, включающем в себя достаточно крупную библиотеку фигур, компонентов и визуальных объектов. Данный редактор позволяет составить сборку или срезку технического изделия или конструкции; описать бизнес-процесс, используемый в производственной, сервисной, логистической или любой другой бизнес-сфере; сформировать диаграмму программного решения или разработки с поэтапной детализацией всех итераций вашего проекта.
* Figma ‑ В фигме обычно создают прототипы сайтов и приложений, иллюстрации, векторную графику, рисуют элементы интерфейса. Ещё здесь создают макеты сайтов для тильды: есть возможность импортировать дизайн.

В фигме работают [дизайнеры](https://practicum.yandex.ru/blog/graphic-designer/), маркетологи, менеджеры продукта, разработчики. Рассмотрим, как использовать фигму.

* PostgreSQL — это мощная система объектно-реляционных баз данных с открытым исходным кодом. Она имеет проверенную архитектуру, которая заслужила хорошую репутацию за надежность, целостность данных и правильность. Существует в реализациях для множества UNIX-подобных платформ, а также для Microsoft Windows. PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта SQL:2011. Сильными сторонами PostgreSQL считаются:
* высокопроизводительные и надёжные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования;
* наследование;
* возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS;
* встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации;
* расширяемость (возможность создавать новые типы данных и др.).

В процессе создания Web-приложения «Студенческая библиотека» в среде psql была разработана база данных «Library student», структура (проекты) таблиц которой с указанием типов данных, а также первичных и внешних ключей (связей).

* Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и другие для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности, включая добавление поддержки систем контроля версий исходного кода и добавление новых наборов инструментов для различных аспектов процесса разработки программного обеспечения.

JavaScript  — [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Поддерживает [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [императивный](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [функциональный](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) стили. Является реализацией спецификации [ECMAScript](https://ru.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) (стандарт ECMA-262[[9]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-ECMA-262-10)).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0" \o "Веб-страница)[[10]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-11).

Основные архитектурные черты: [динамическая типизация](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [слабая типизация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%B0%D0%B1%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), [автоматическое управление памятью](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [прототипное программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5" \o "Прототипное программирование), функции как [объекты первого класса](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0).

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке[[~ 1]](https://ru.wikipedia.org/wiki/JavaScript" \l "cite_note-12)[[11]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-13).

Название «JavaScript» является зарегистрированным [товарным знаком](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%B2%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D0%B7%D0%BD%D0%B0%D0%BA) корпорации [Oracle](https://ru.wikipedia.org/wiki/Oracle" \o "Oracle) в США[[12]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-14).

В [1992 году](https://ru.wikipedia.org/wiki/1992_%D0%B3%D0%BE%D0%B4) компания *Nombas* (впоследствии приобретённая [Openwave](https://ru.wikipedia.org/w/index.php?title=Openwave&action=edit&redlink=1" \o "Openwave (страница отсутствует))[[en]](https://en.wikipedia.org/wiki/Openwave)) начала разработку встраиваемого скриптового языка *Cmm* (Си-минус-минус), который, по замыслу разработчиков, должен был стать достаточно мощным, чтобы заменить [макросы](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%BA%D1%80%D0%BE%D1%81), сохраняя при этом схожесть с Си, чтобы разработчикам не составляло труда изучить его[[13]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-nombasHistory-15). Главным отличием от Си была работа с памятью. В новом языке всё управление памятью осуществлялось автоматически: не было необходимости создавать [буфера](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D1%84%D0%B5%D1%80_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), объявлять переменные, осуществлять преобразование типов. В остальном языки сильно походили друг на друга: в частности, *Cmm* поддерживал стандартные функции и операторы Си[[14]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-16). *Cmm* был переименован в *ScriptEase*, поскольку исходное название звучало слишком негативно, а упоминание в нём Си «отпугивало» людей[[13]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-nombasHistory-15)[[15]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-17). На основе этого языка был создан [проприетарный продукт](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) *CEnvi*. В конце ноября [1995 года](https://ru.wikipedia.org/wiki/1995_%D0%B3%D0%BE%D0%B4) *Nombas* разработала версию *CEnvi*, внедряемую в веб-страницы. Страницы, которые можно было изменять с помощью скриптового языка, получили название *Espresso Pages* — они демонстрировали использование скриптового языка для создания игры, проверки пользовательского ввода в формы и создания анимации. *Espresso Pages* позиционировались как [демоверсия](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BC%D0%BE%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F), призванная помочь представить, что случится, если в браузер будет внедрён язык *Cmm*. Работали они только в 16-битовом [Netscape Navigator](https://ru.wikipedia.org/wiki/Netscape_Navigator" \o "Netscape Navigator) под управлением [Windows](https://ru.wikipedia.org/wiki/Windows" \o "Windows)[[16]](https://ru.wikipedia.org/wiki/JavaScript#cite_note-18).

## 3.2 Календарный план разработки

Для распределения временных сроков проекта, был создан календарный план разработки программного продукта. Он представлен на рисунке 7.

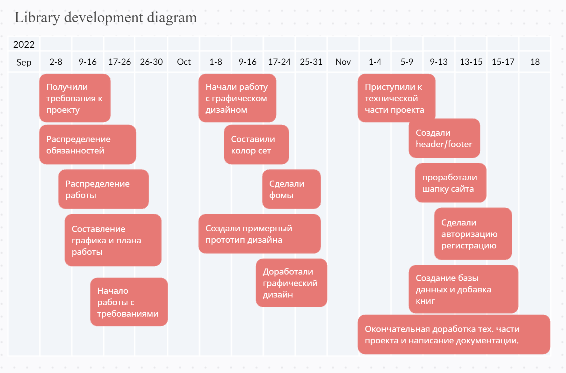


Рисунок 7 – Календарный план работы

# 

# 4 Тестирование программного продукта

## 4.1 Выбор метода обеспечения качества

Тестирование программного обеспечения ‑ это оценка разрабатываемого программного обеспечения/продукта, чтобы проверить его возможности, способности и соответствие ожидаемым результатам. Существуют различные типы методов, используемые в области тестирования и обеспечения качества.

Тестирование – процесс исследования и контроль качества, который состоит из планирования, проектирования, собственно проверки и анализа ее результатов.

## 4.2 Тестирование программного продукта

Для проведения тестирования программного продукта был составлен план проведения тестирования, который представлен в таблице 8.

Таблица 8 – План тестирования

|  |  |  |
| --- | --- | --- |
| Тест кейс, № | Название | Описание |
| 1 | Тест авторизации  пользователя Администратора | Тест авторизации в программе под Администратором |
| 2 | Тест авторизации пользователя | Тест авторизации в программе под пользователем |
| 3 | Тест занесения новых видов книг в  библиотеку | Тест возможности сохранения данных нового издания в базе данных |
| 4 | Тест открытия журнала опубликованных изданий | Тест загрузки информации из базы данных в табличный компонент формы приложения |

Тест-кейсы для проведения тестирования представлены в таблицах 9- 12.

Таблица 9 – Тест-кейс 1

|  |  |
| --- | --- |
| Идентификатор | 1 |
| Название | Тест авторизации Администратора |

Продолжение таблицы 9

|  |  |
| --- | --- |
| Приоритет | Высокий |
| Описание | Тестируется правильность соединения с базой данных и перехода на панель редактирования |
| Предусловие | Предусловие отсутствует |
| Шаги  тестирования | Запустить проект (форму авторизации)  Выбрать нужный логин в списке пользователей Ввести пароль  Нажать на кнопку «Вход» |
| Ожидаемый результат | При корректном вводе данных: переход на панель редактирования приложения с полностью  доступным  функционалом;  при некорректном вводе данных: сообщение об ошибке |
| Постусловие | Постусловие отсутствует |
| Фактический результат | При корректном вводе данных: происходит переход на главную панель редактирования с полностью  доступным функционалом;  при некорректном вводе данных: отображается сообщение об ошибке |
| Статус | Пройден |

Таблица 10 – Тест-кейс 2

|  |  |
| --- | --- |
| Идентификатор | 2 |
| Название | Тест авторизации пользователя |
| Приоритет | Средний |
| Описание | Тестируется правильность соединения с базой данных и перехода на главную форму приложения |
| Предусловие | Предусловие отсутствует |
| Шаги тестирования | Запустить проект (форму авторизации)  Выбрать нужный логин в списке пользователей Ввести пароль  Нажать на кнопку «Вход» |
| Ожидаемый результат | При корректном вводе данных: переход на главную форму приложения с активными для пользователя главная страница, контактная информация, личный кабинет;  при некорректном вводе данных: сообщение об ошибке |

Продолжение таблицы 10

|  |  |
| --- | --- |
| Постусловие | Постусловие отсутствует |
| Фактический результат | При корректном вводе данных: происходит переход на главную форму приложения где пользователю будет уже доступен его личный кабинет, создание подборок книг, и прочтение самих книг;  при некорректном вводе данных: отображается сообщение об ошибке |
| Статус | Пройден |

Таблица 11 – Тест-кейс 3

|  |  |
| --- | --- |
| Идентификатор | 3 |
| Название | Тест занесения новых видов книг в библиотеку |
| Приоритет | Высокий |
| Описание | Тестируется правильность внесения данных в БД при использовании приложения |
| Предусловие | Предусловие отсутствует |
| Шаги тестирования | Авторизоваться в приложении в роли Администратора. Открыть форму книги и выбрать пункт «Добавить книгу» из меню данные.  Ввести данные новой книги в очищенные текстовые поля формы.  Нажать кнопку «Сохранить данные» |
| Ожидаемый результат | При корректном вводе данных: поле с введенным кодом нового видеодиска на форме справочника становится недоступным к редактированию;  при некорректном вводе данных: сообщение об ошибке |
| Постусловие | Постусловие отсутствует |
| Фактический результат | При корректном вводе данных: поле с введенными данными новой книги на форме становится  недоступным к редактированию;  при некорректном вводе данных: сообщение об ошибке |
| Статус | Пройден |

Таблица 12 – Тест-кейс 4

|  |  |
| --- | --- |
| Идентификатор | 4 |
| Название | Тест открытия журнала опубликованных изданий |
| Приоритет | Высокий |
| Описание | Тестируется правильность загрузки информации из базы данных в табличный компонент формы при  использовании приложения |
| Предусловие | Предусловие отсутствует |
| Шаги тестирования | Авторизоваться в приложении в роли Администратора.  Открыть форму журнала книг выбором пункта  «Редактирование» из меню «Журналы» главной формыприложения |
| Ожидаемый результат | На форме журнала книги отображение таблицы из 4  столбцов («Номер», «Дата создание», «Категория»,  «Дата публикации»); |
| Постусловие | Постусловие отсутствует |
| Фактический результат | В таблице отображаются правильные названия столбцов |
| Статус | Пройден |

В процессе тестирования было выявлено не было ошибок:

* В ходе наших тестов в таблицах выше мы не выявили никаких ошибок, поэтому данный продукт полностью готов к эксплуатации пользователями. В тестировании были проверенны важные компоненты проекта: авторизация «Пользователя», авторизация «Администратора», редактирование и добавление новых изданий, а также отправка и получение данных с базы SQL.

# 5 Разработка документации на программный продукт

К видам документации, разрабатываемой на разных этапах жизненного цикла программного продукта, относятся:

1. технические требования;
2. технические спецификации;
3. сведения о выпуске;
4. руководства (например, по установке ПП, пользователя, администратора, программиста, по технической поддержке и т.д.);
5. описание проекта;
6. планы;
7. задания исполнителям (задание, распределённое между конкретными людьми или группами, участвующими в реализации проекта);
8. отчёты о ходе работ ‑ создаются менеджерами для контролирующих органов;
9. протоколы встреч и обсуждений;
10. отчёты о результатах активности;
11. журналы.

По своему назначению и ориентации на определенные задачи и группы пользователей, документацию ПП можно разбить на две группы:

1. программные документы (технологические документы управления разработкой ПП) ‑ документы, которые предназначены, прежде всего, для самих разработчиков и их начальства;
2. эксплуатационные программные документы (документы, входящие в состав программных систем) ‑ документы, предназначенные для конечных пользователей или же обслуживающего персонала, позволяющие им осваивать и квалифицированно применять эти системы для решения конкретных функциональных задач.

# Заключение

Результатом курсовой работы стал готовый проект к работе – web-приложение, пользователи могут читать книги, регистрировать аккаунты, а также создавать подборки книг и делится между пользователями. Оценивая проделанную работу, можно сделать следующие выводы:

- Сайт предоставляет пользователю информацию, а тот в свою очередь выбирает и просматривает то что ему нужно. Это может быть книга и её содержание.

- Дизайн сайта соответствует предпочтениям пользователей.

- Реализована навигация с помощью меню по страницам сайта.

- Реализована форма регистрации/авторизации пользователя.

- Реализована форма обратной связи

В дальнейшем будут реализованы такие функции как:

- Функция лайков

- Система личных сообщений

- Возможность для других пользователей вести свой личный блог

- Смена светлой темной/светлой темы на сайте

# Список литературы

1. Andy, Harris HTML, XHTML and CSS All–In–One For Dummies® / Andy Harris. - Москва: **Наука**, **2014**. - **173** c.
2. Ben, Henick HTML & CSS – The Good Parts / Ben Henick. - Москва: **СИНТЕГ**, **2013**. - 350 c.
3. Ed, Tittel HTML, XHTML & CSS For Dummies® / Ed Tittel. - Москва: **Гостехиздат**, **2012**. - 416 c.
4. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, **2014**. - 464 c.
5. Гудман, Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов / Д. Гудман. - М.: Питер, **2015**. - 523 c.
6. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM) / Джон Дакетт. - М.: Эксмо, 2013. - 480 c.
7. Дакетт, Джон Основы веб-программирования с использованием HTML, XHTML и CSS / Джон Дакетт. - М.: Эксмо, **2013**. - 768 c.
8. Дебольт HTML и CSS. Совместное использование / Дебольт, Вирджиния. - М.: НТ Пресс, **2013**. - 512 c.
9. Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2014. - **138** c.
10. Дронов, Владимир HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, **2013**. - 416 c

# Приложение А

Структура (проекта) представлена в виде функционала на рисунках А.1-А.10.

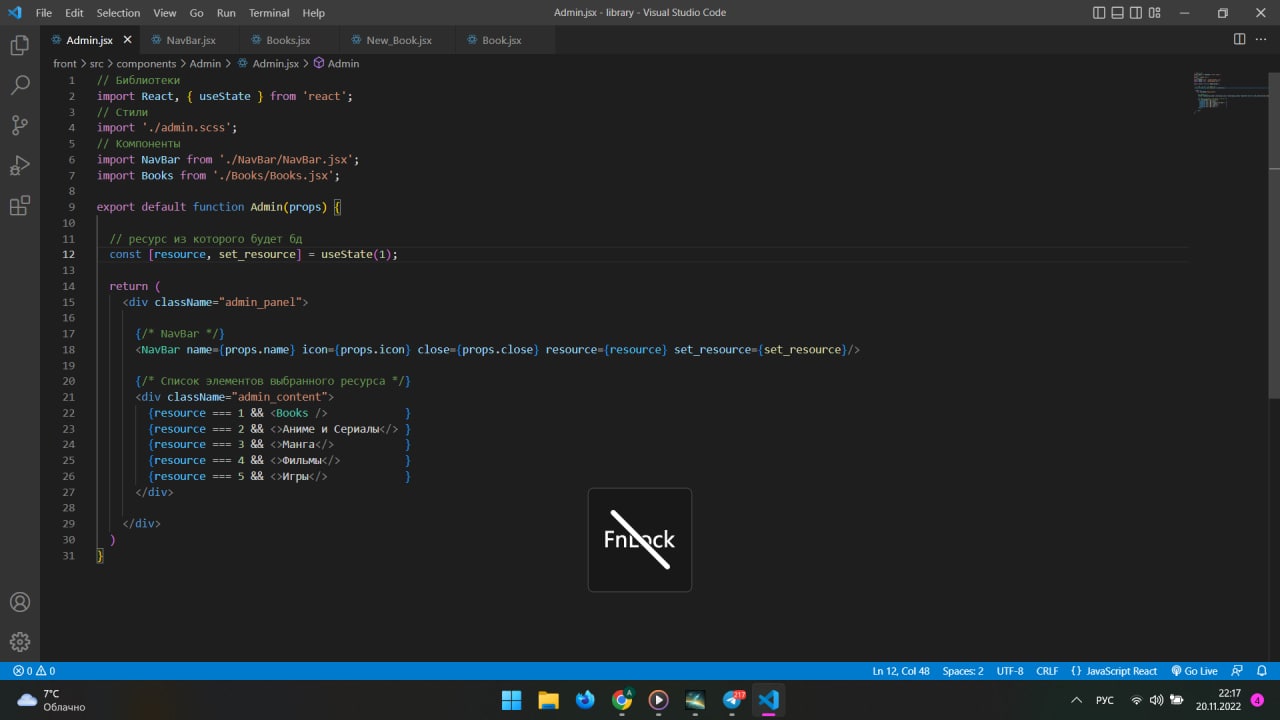
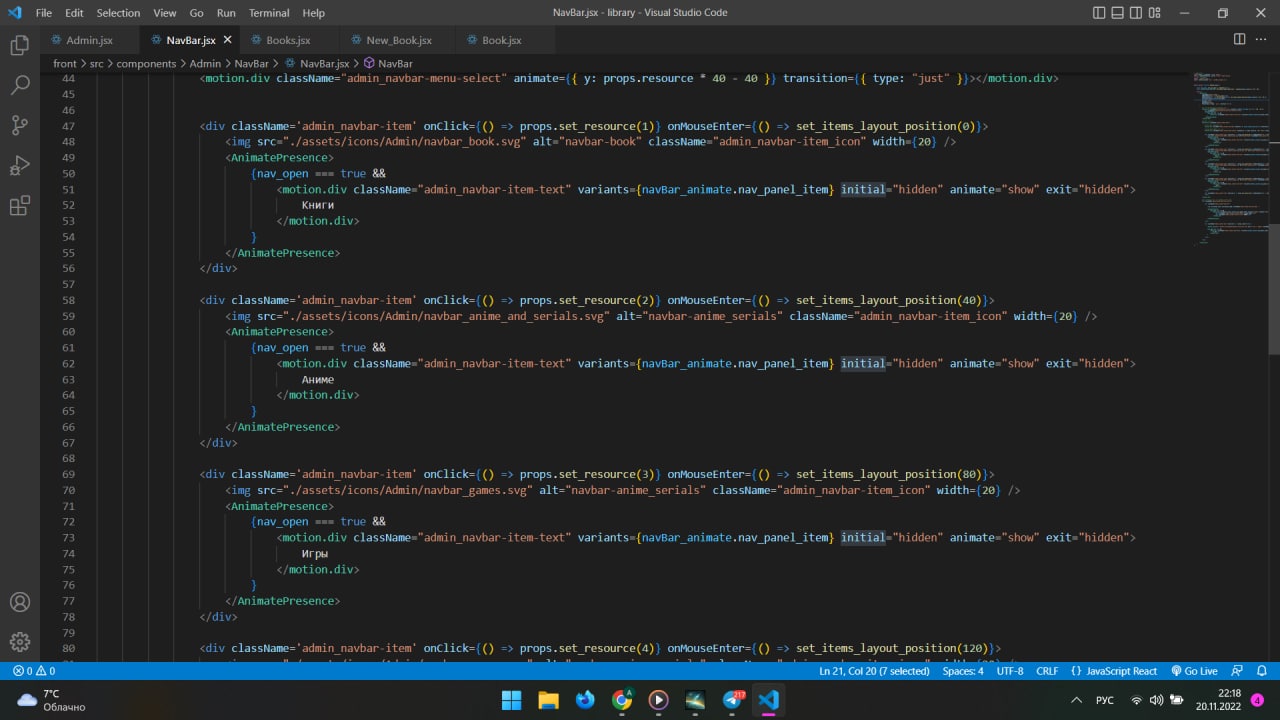


Рисунок А.1 – Админ панель



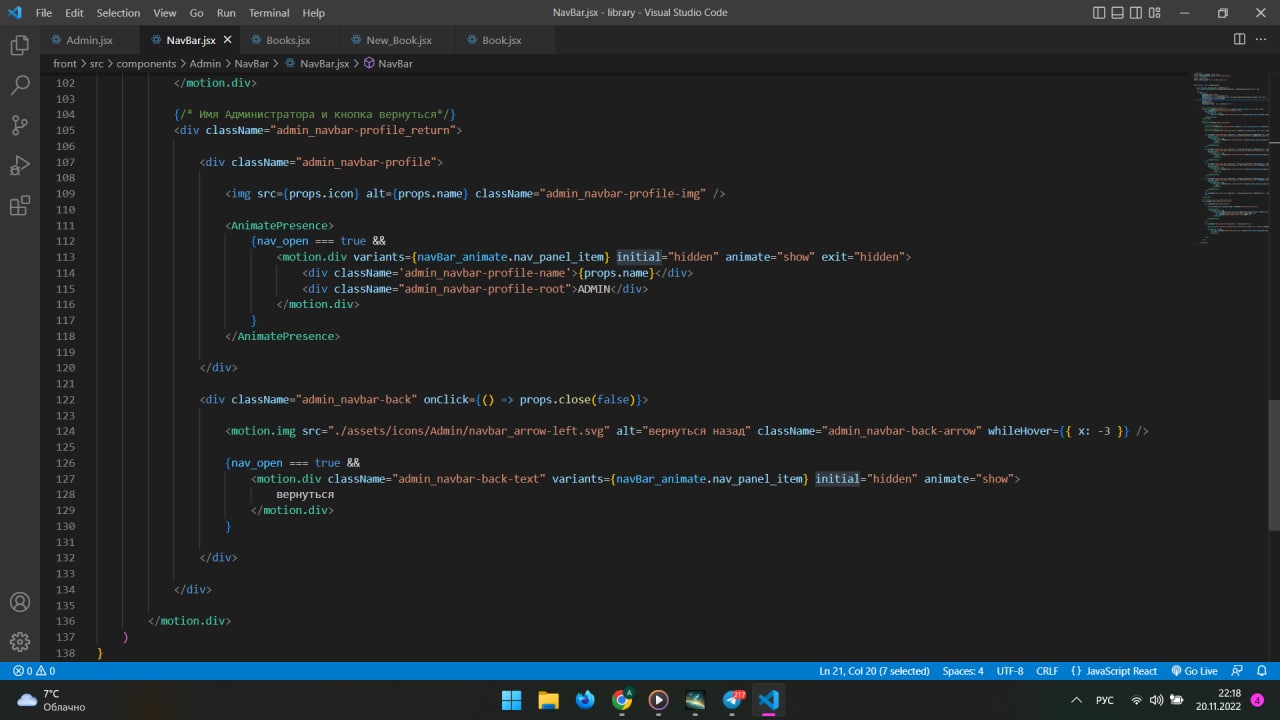
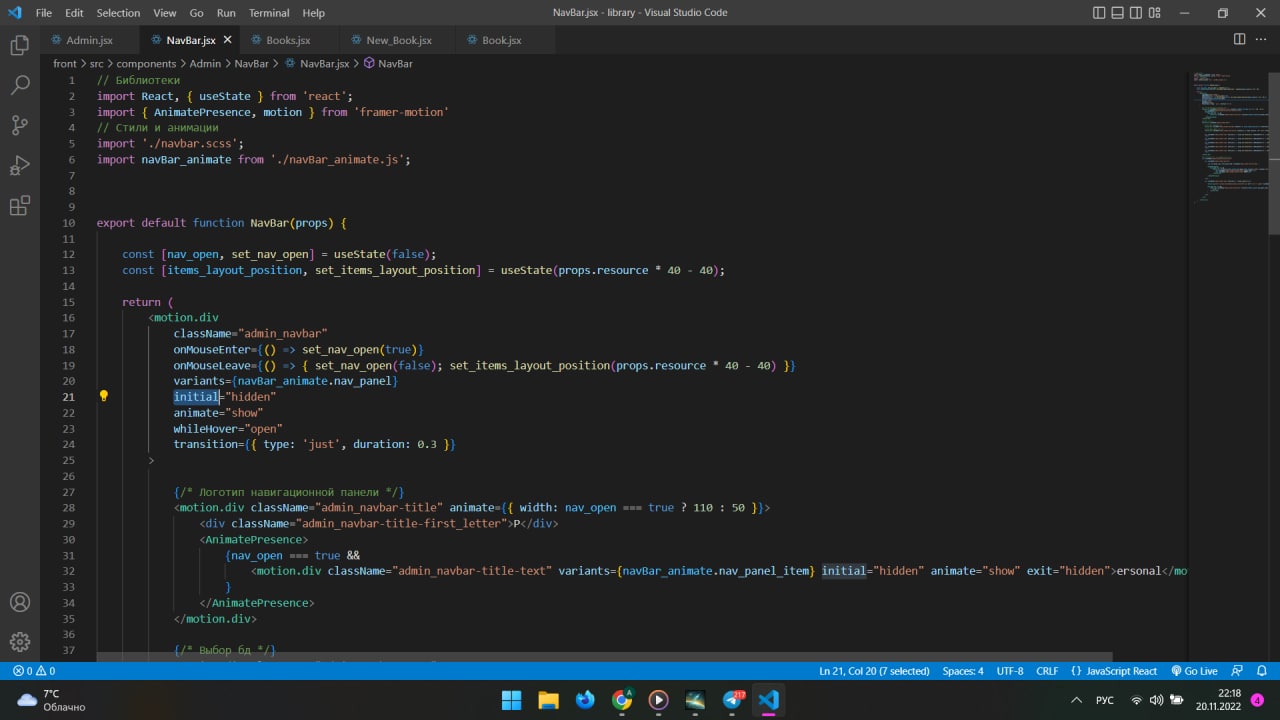


Рисунок А.2 – Меню административной панели

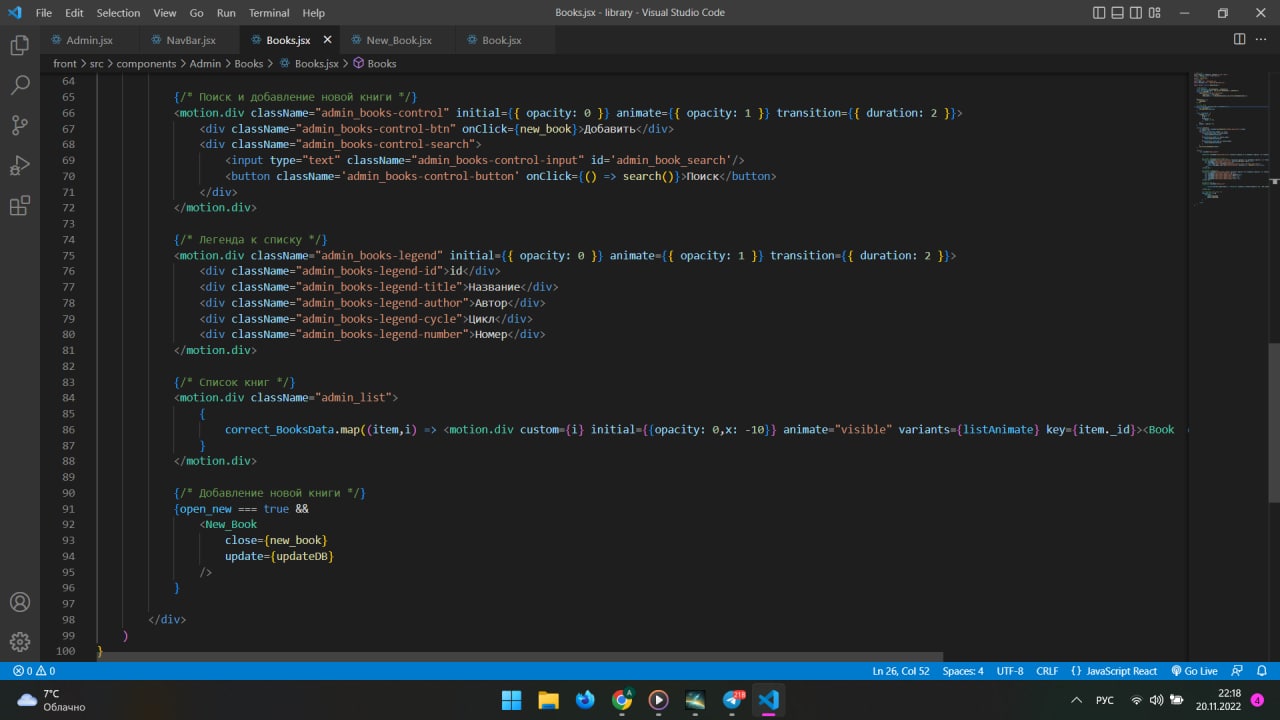
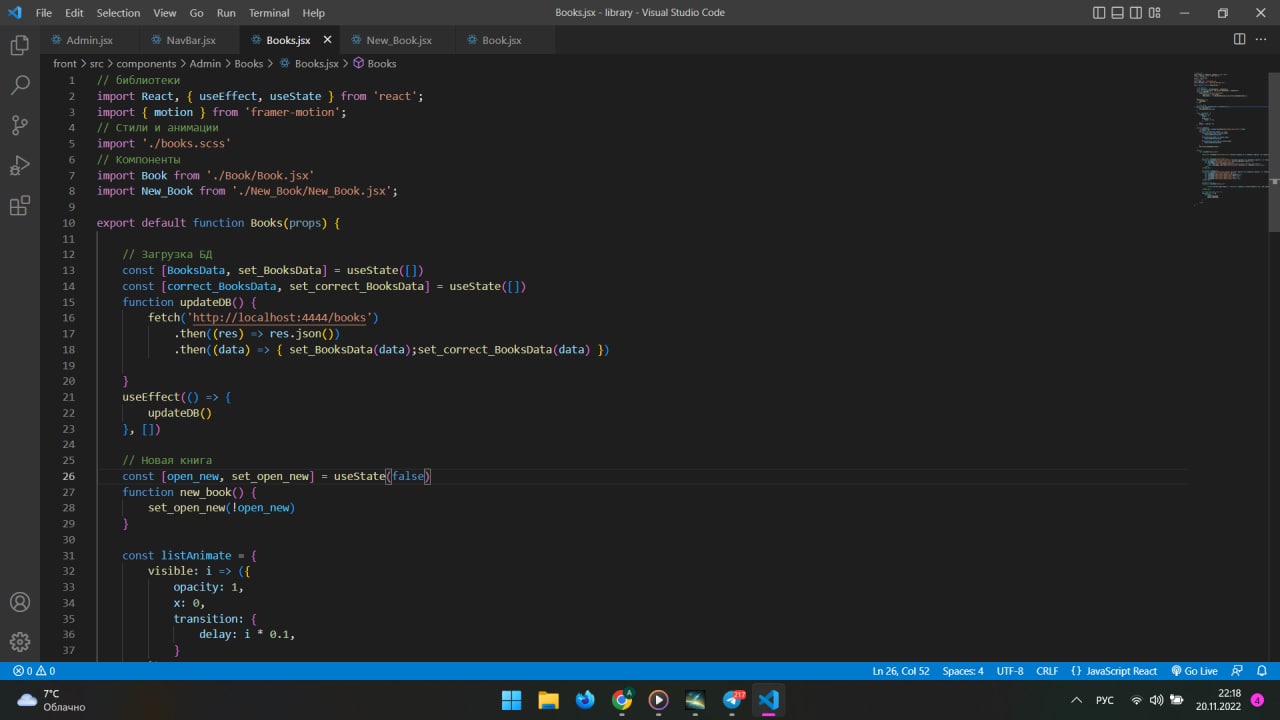
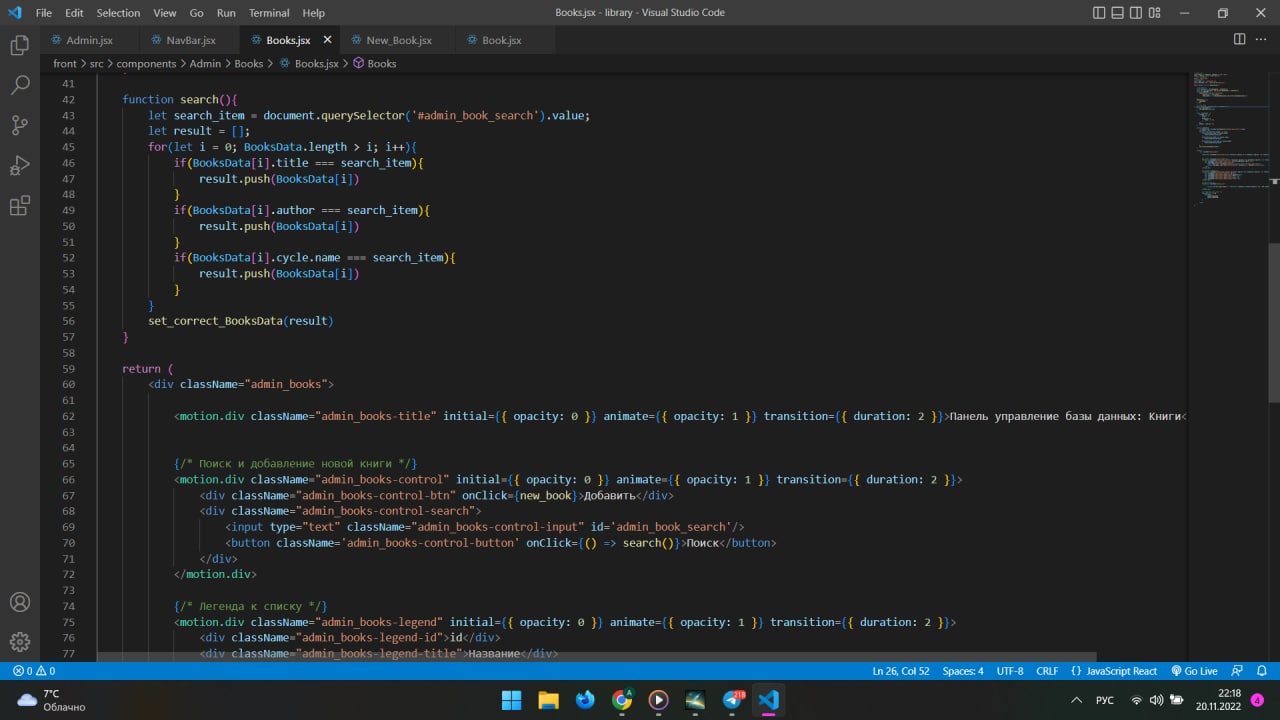
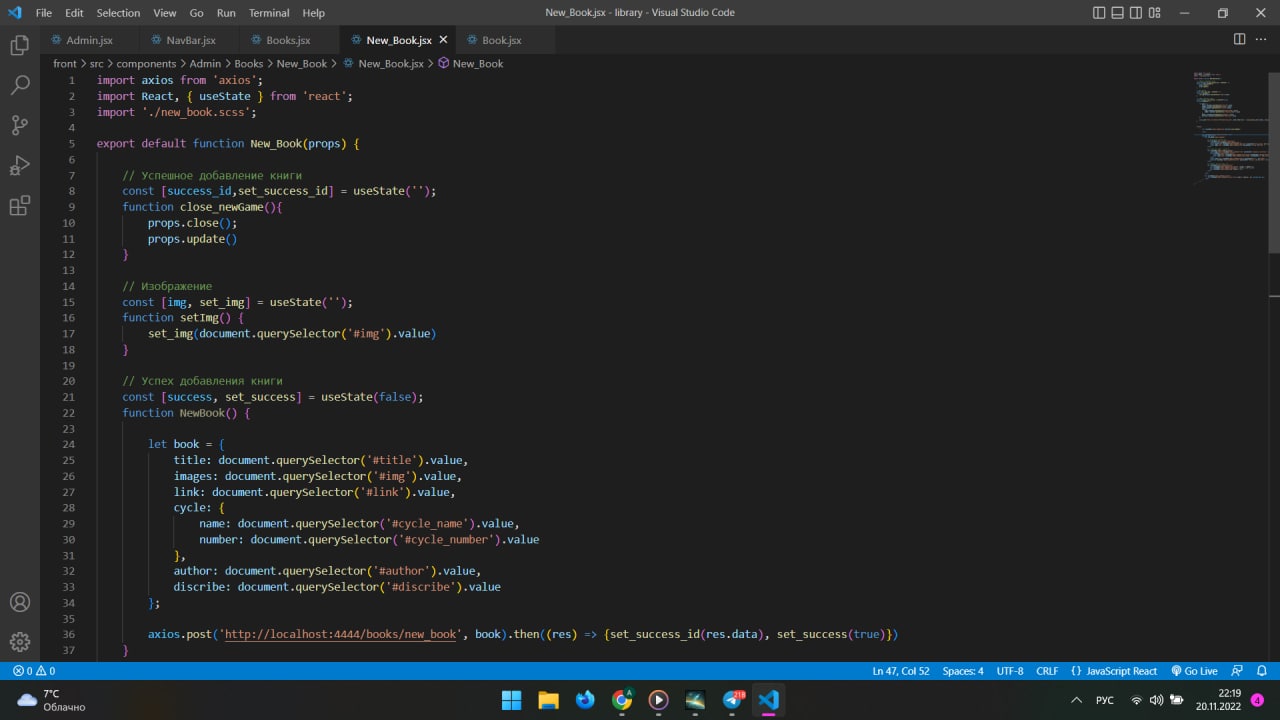


Рисунок А.3 – Книги в панеле администратора



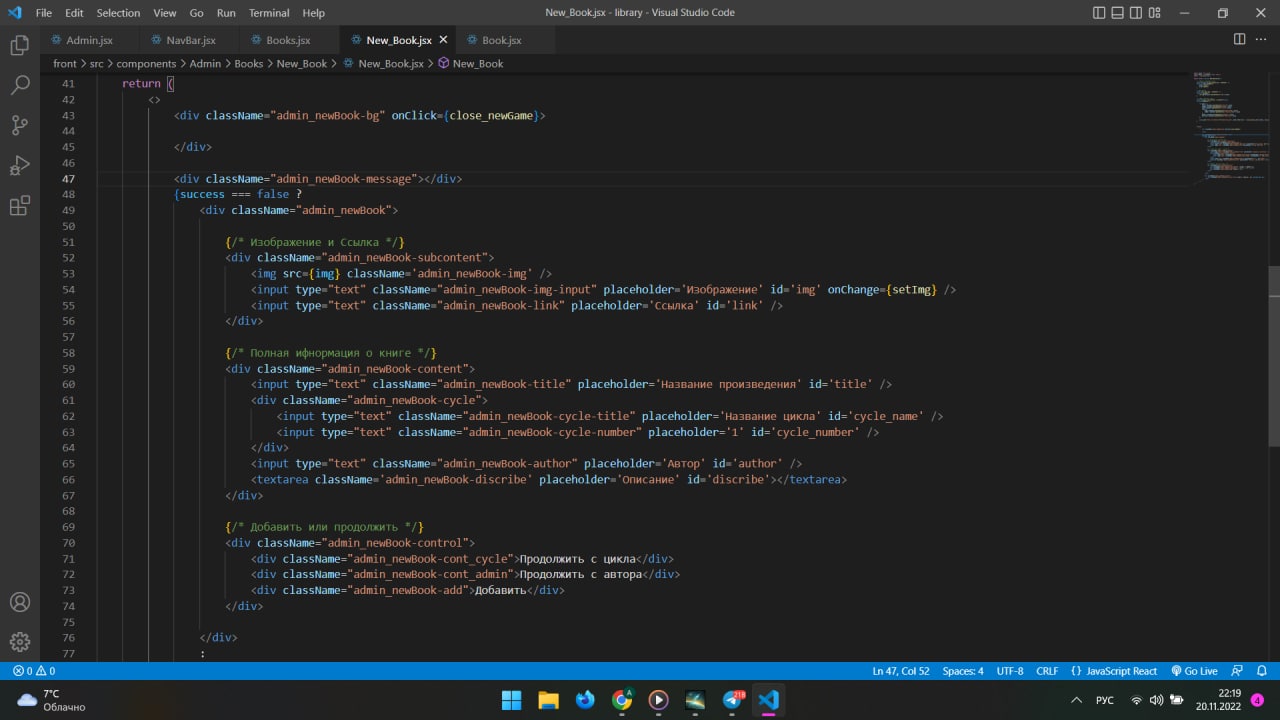
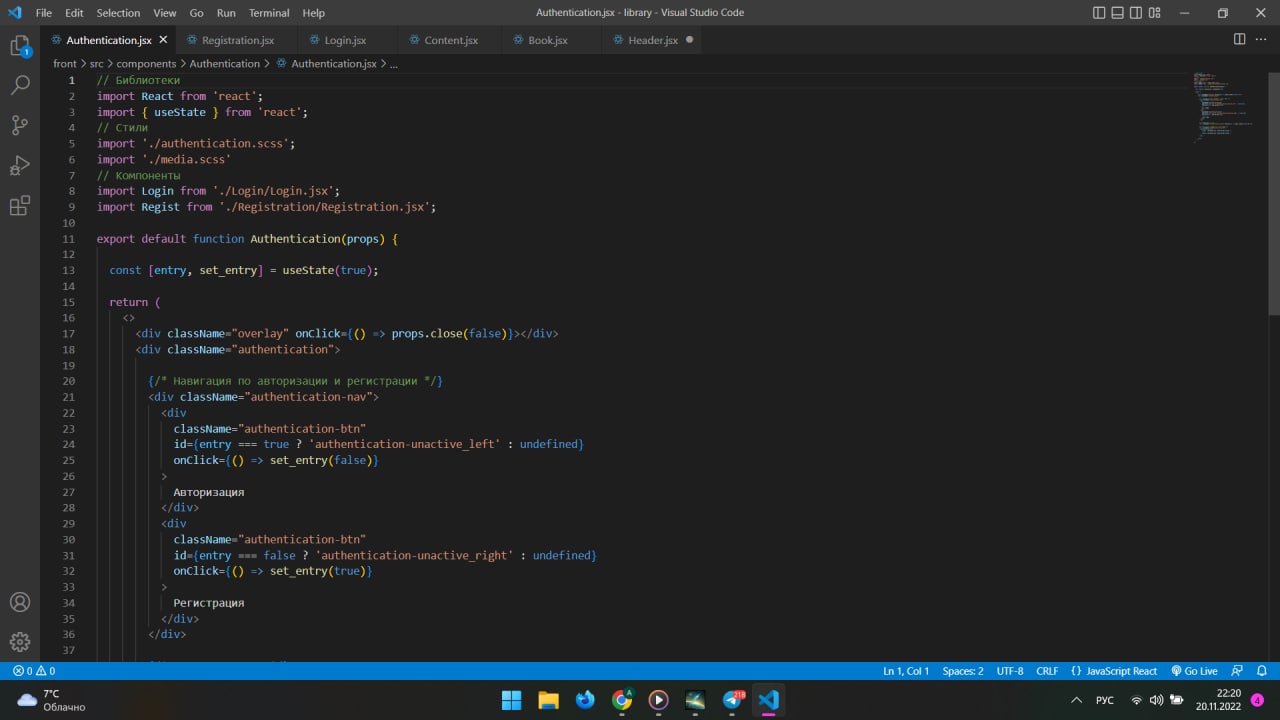


Рисунок А.4 – Содержание книги в панели Администратора



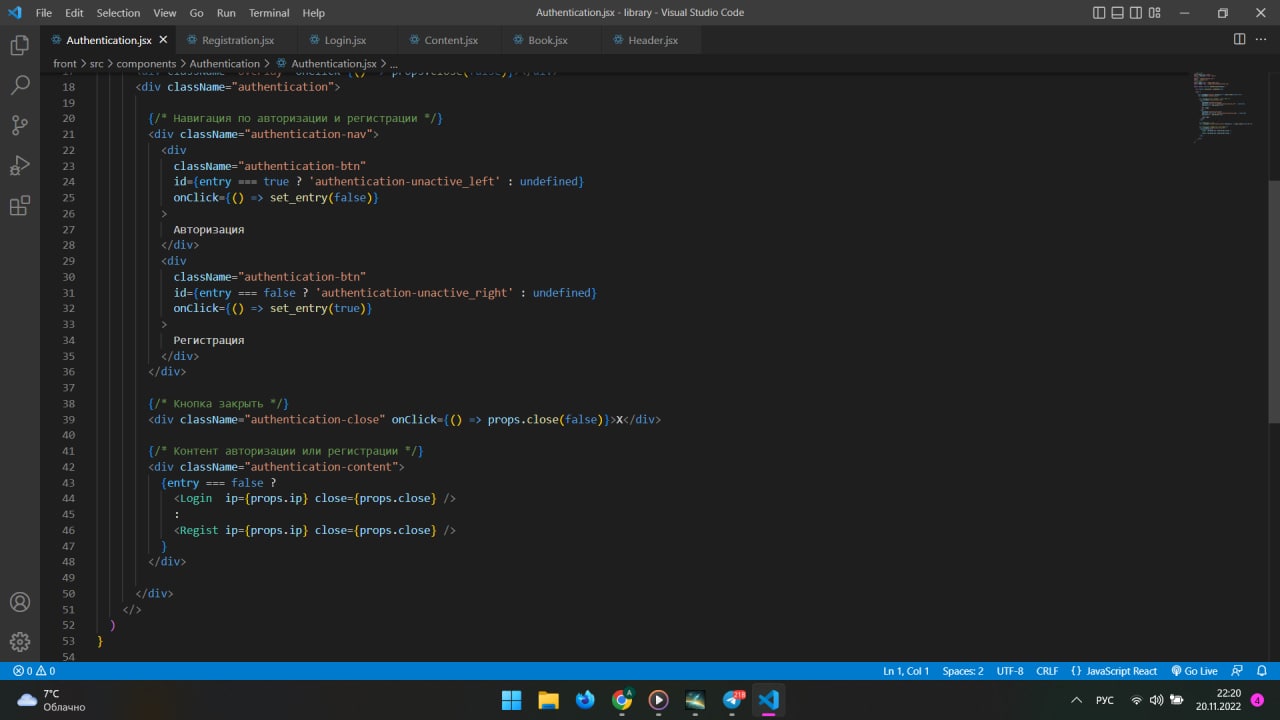
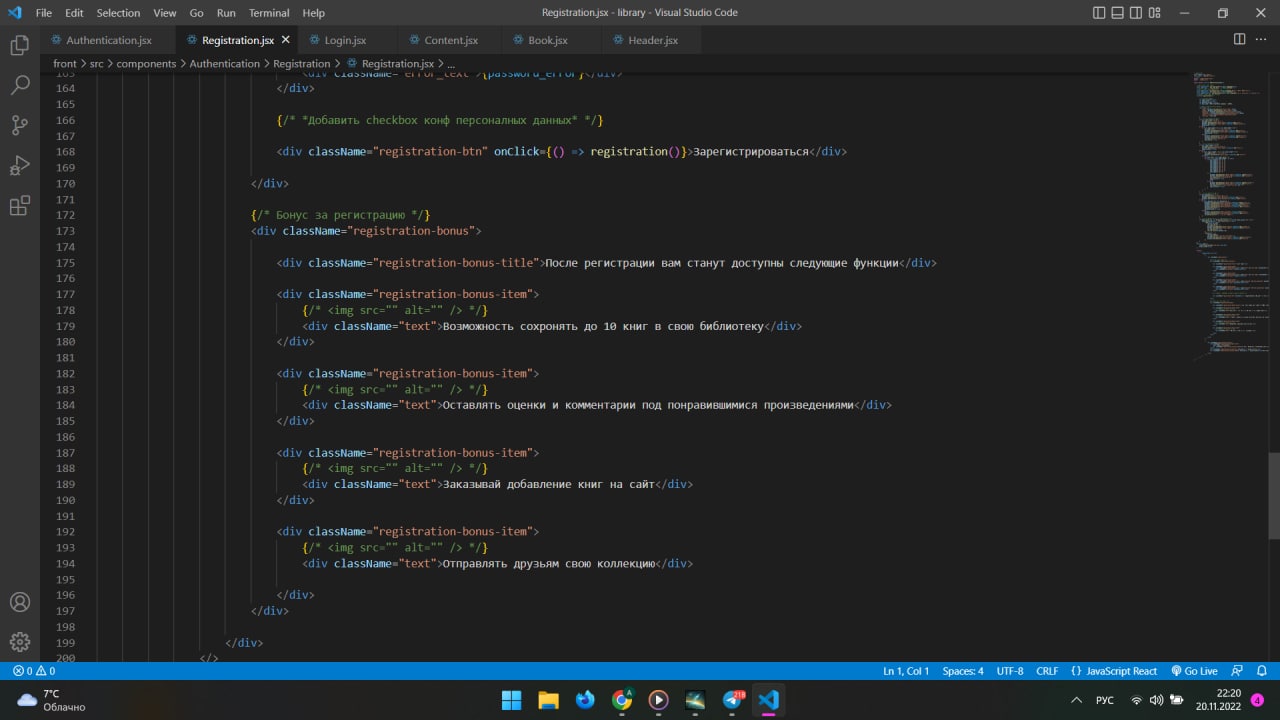


Рисунок А.5 – Аунтифекация



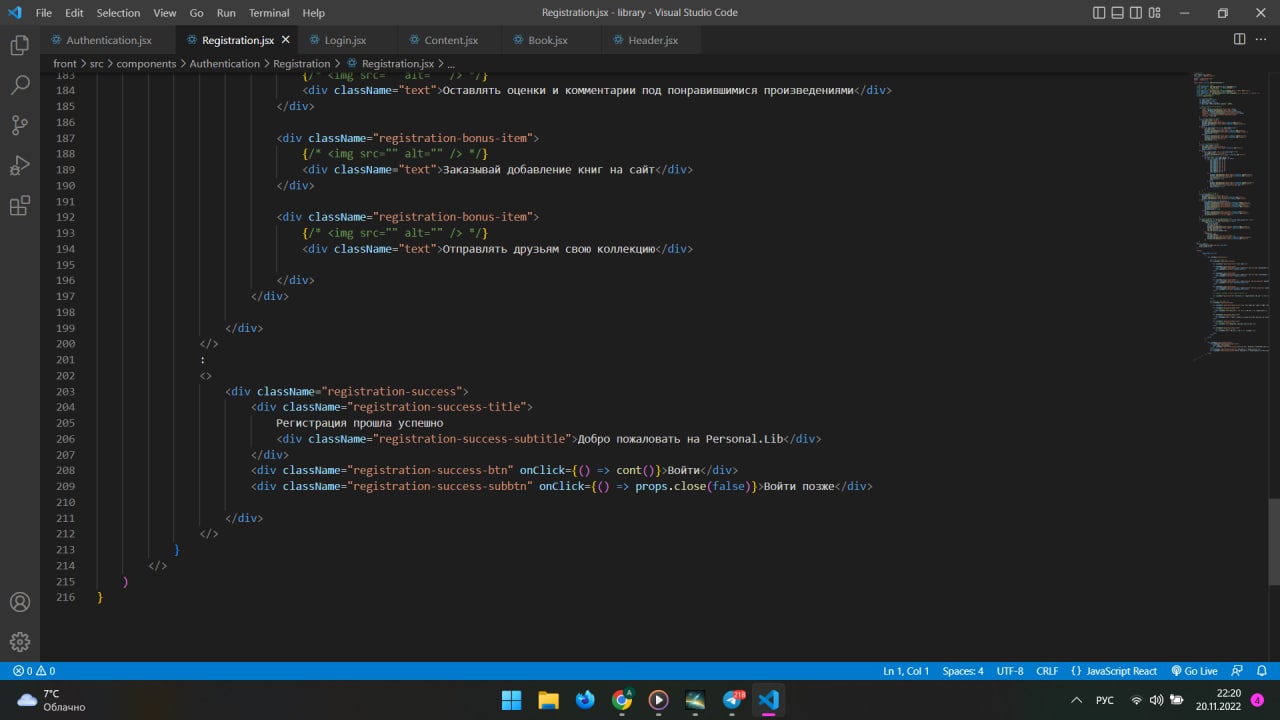
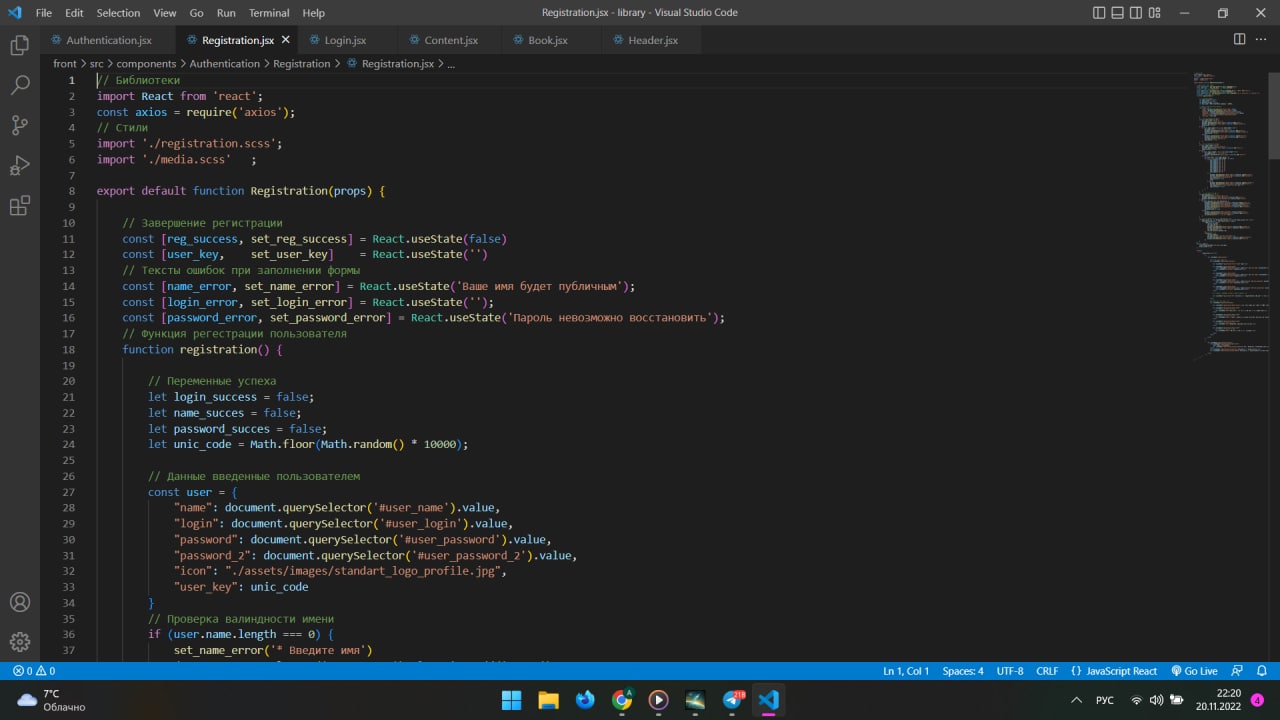
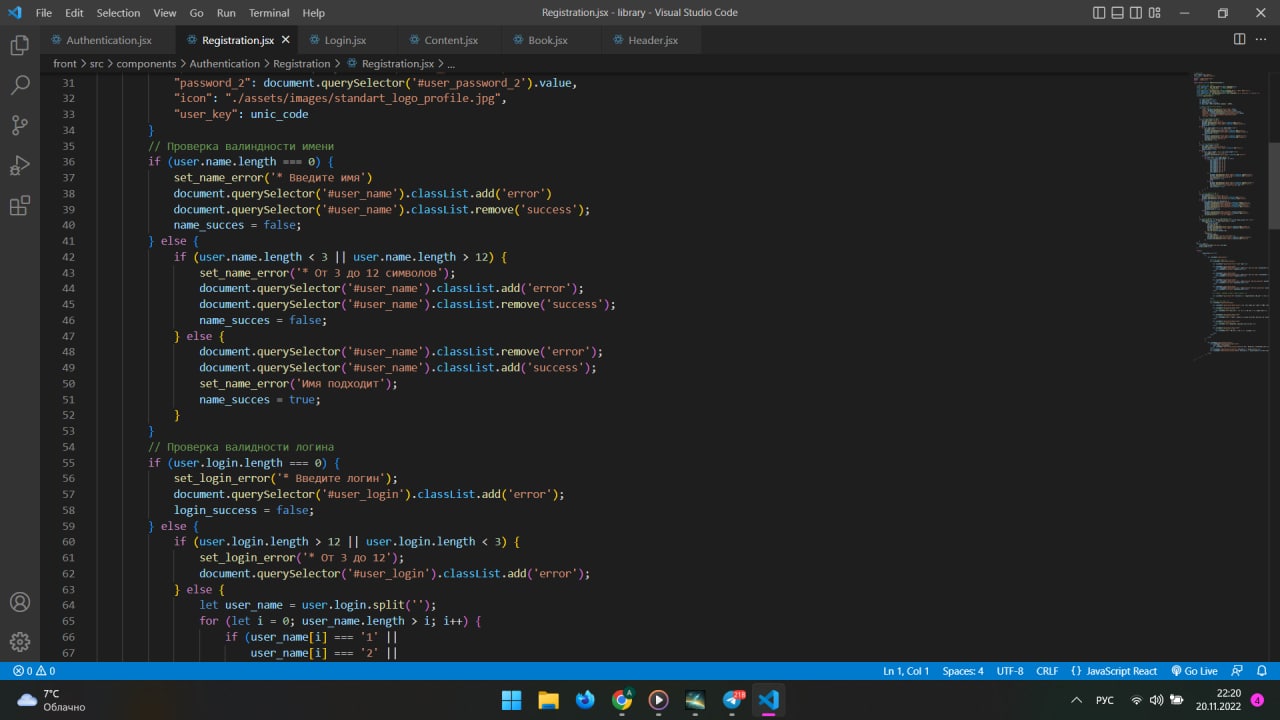
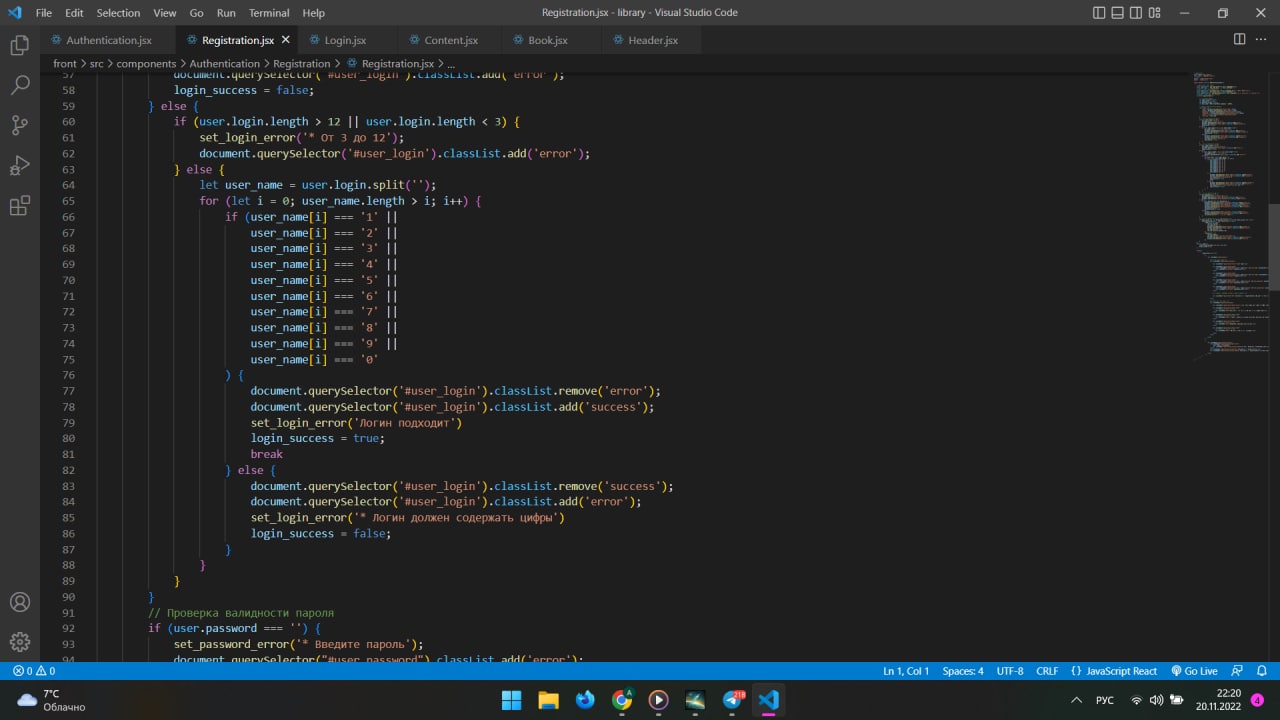
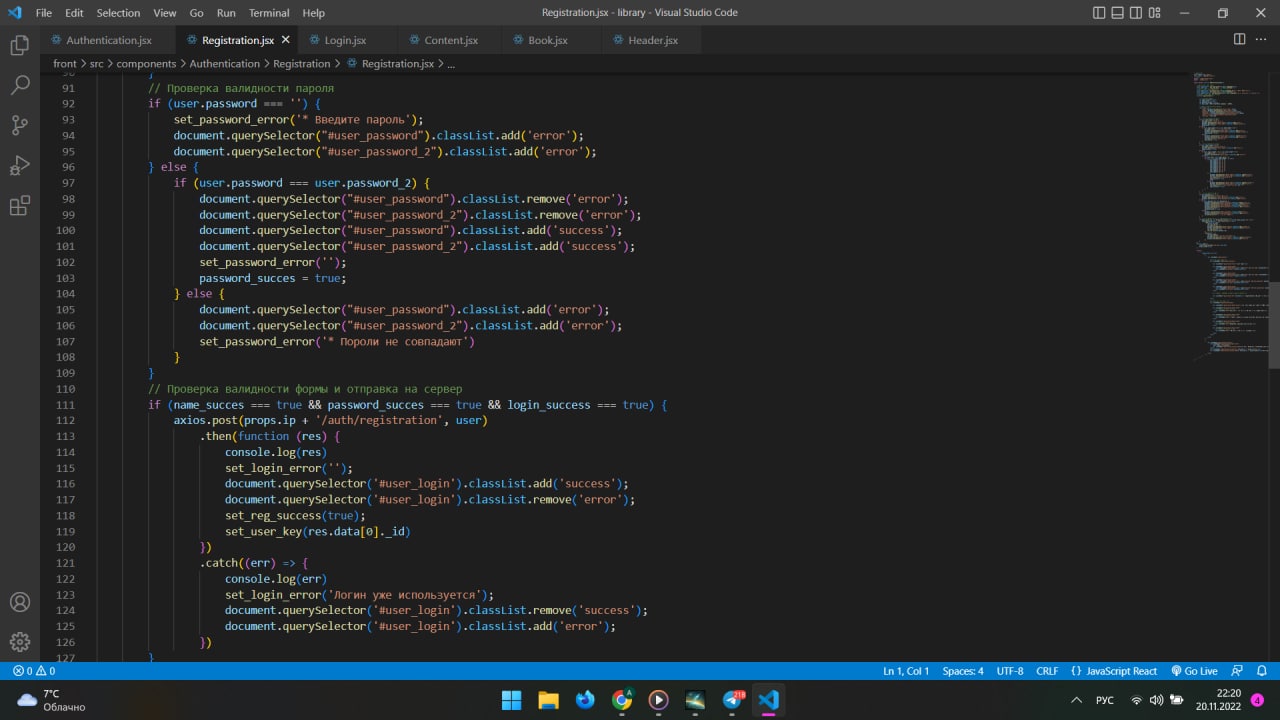
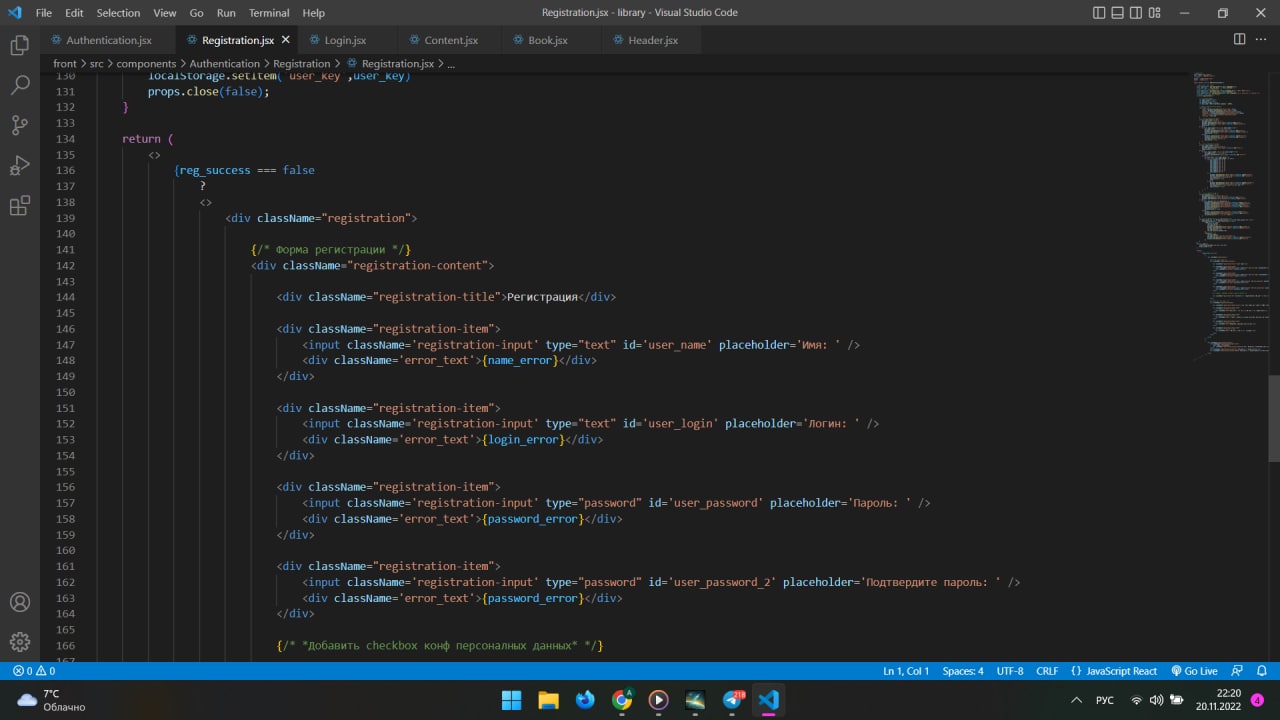
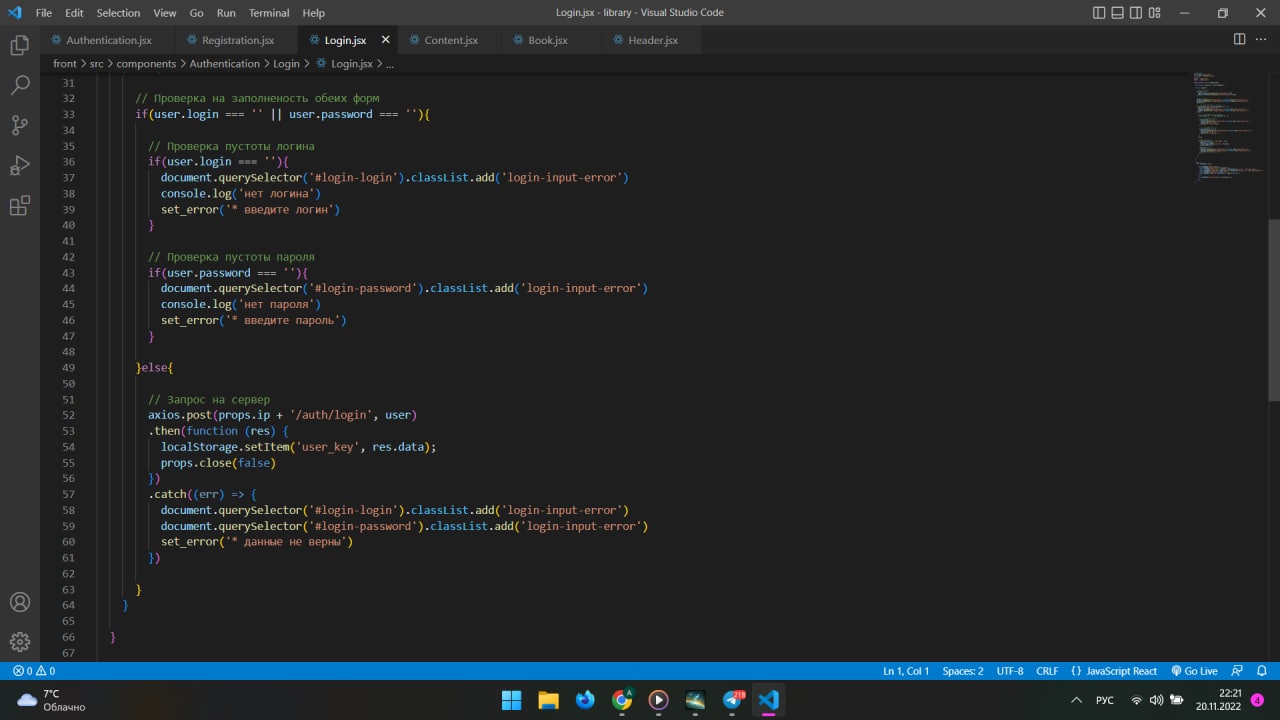
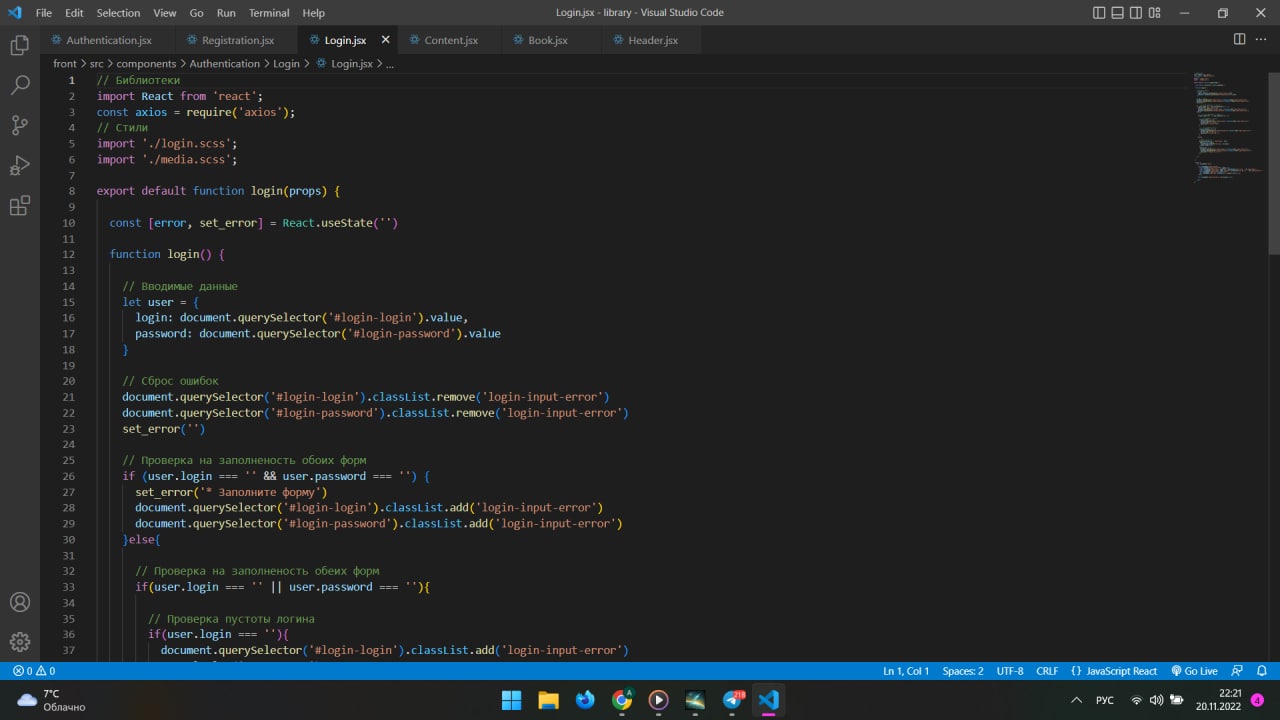


Рисунок А.6 – Регистрация



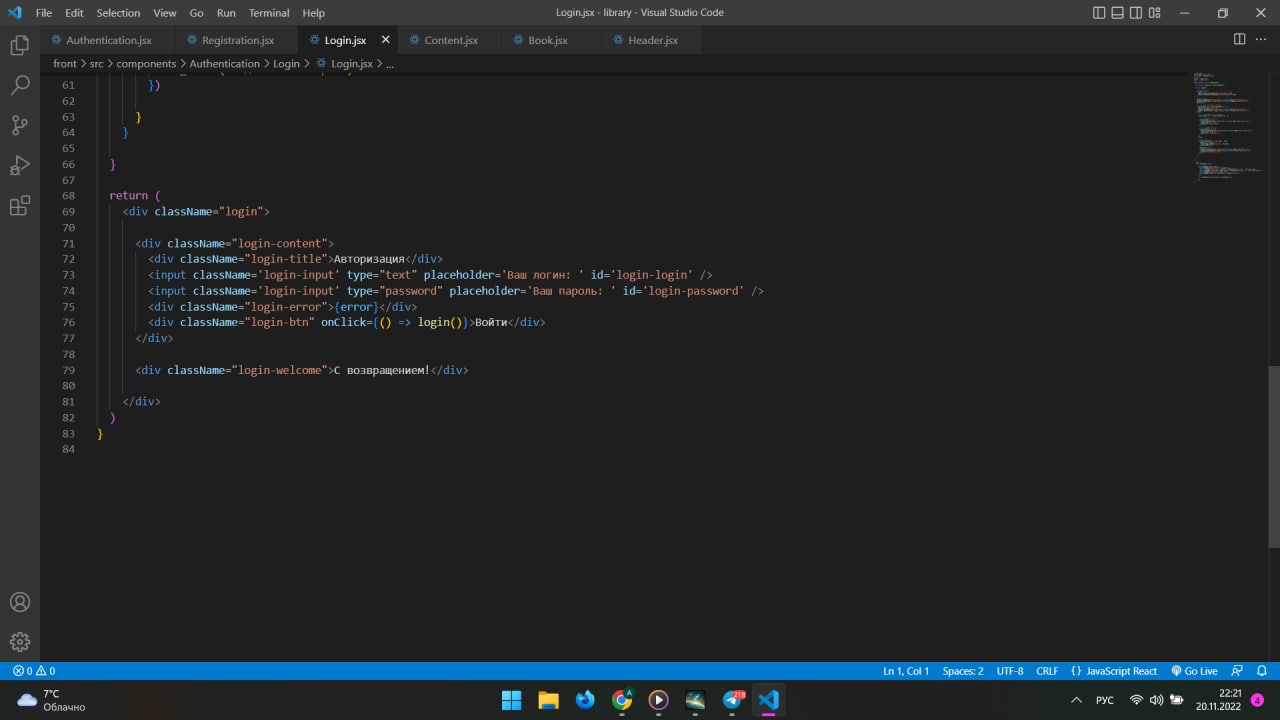
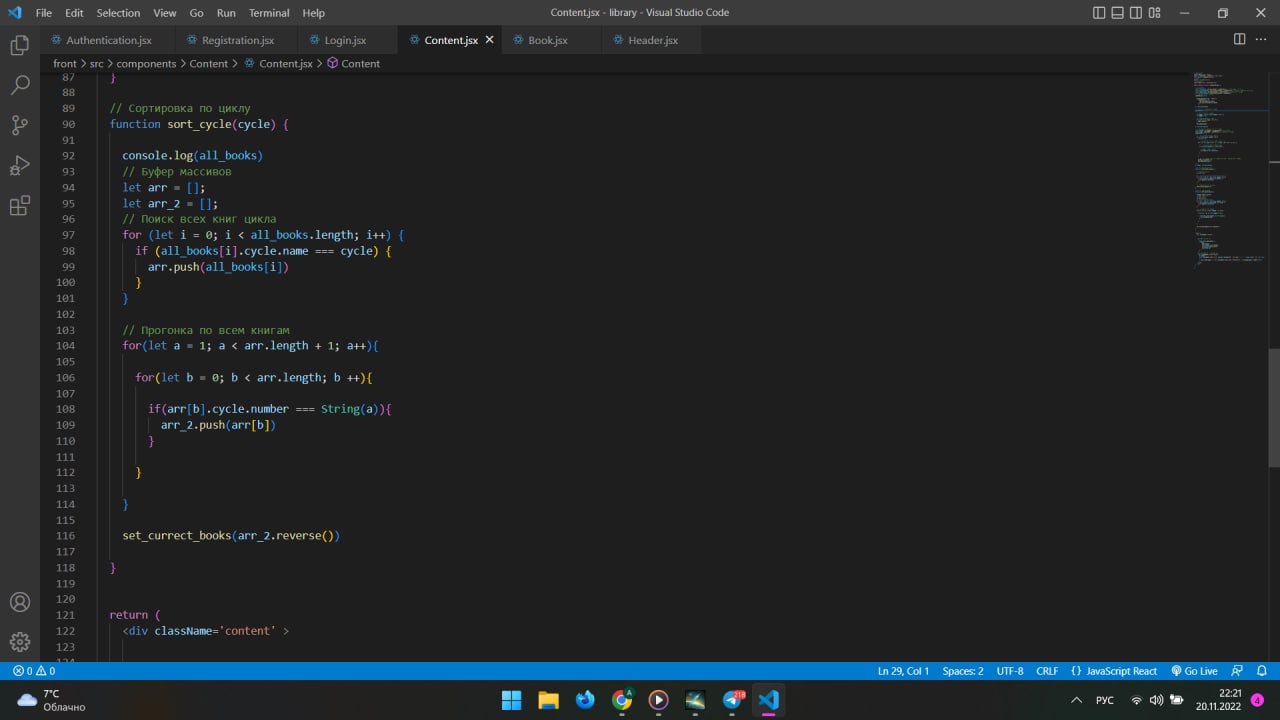


Рисунок А.7 – Авторизация



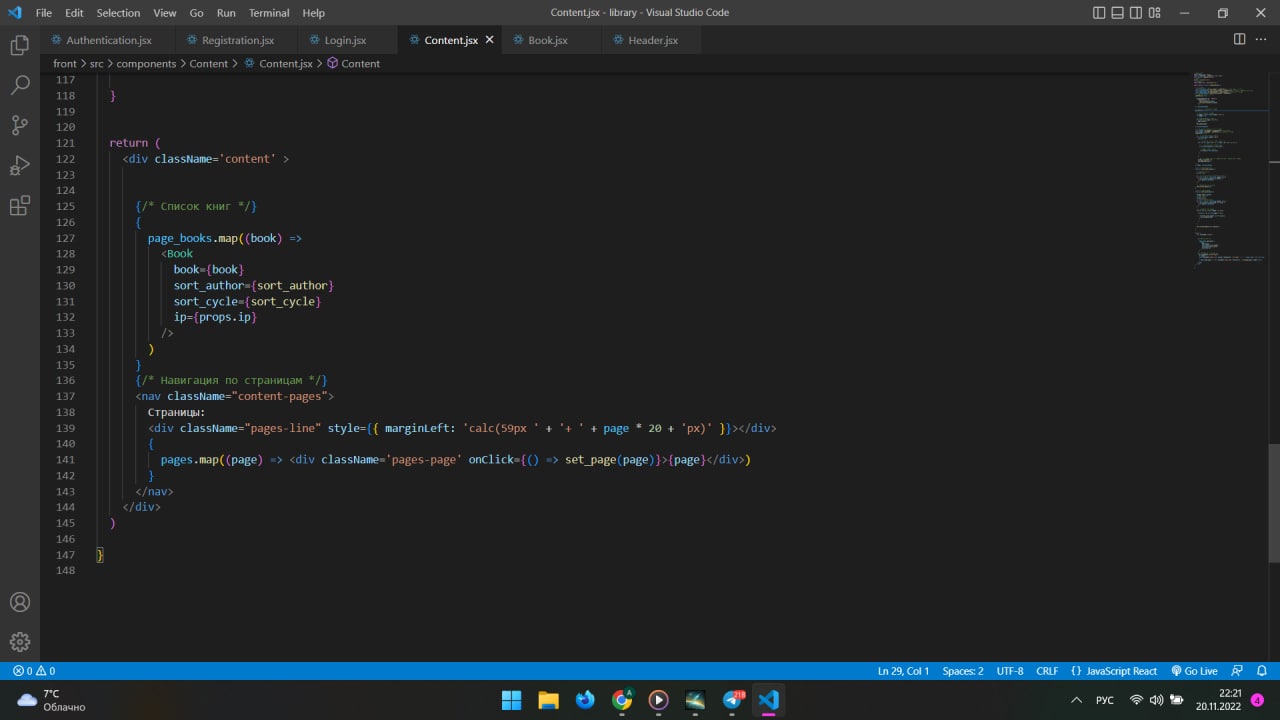
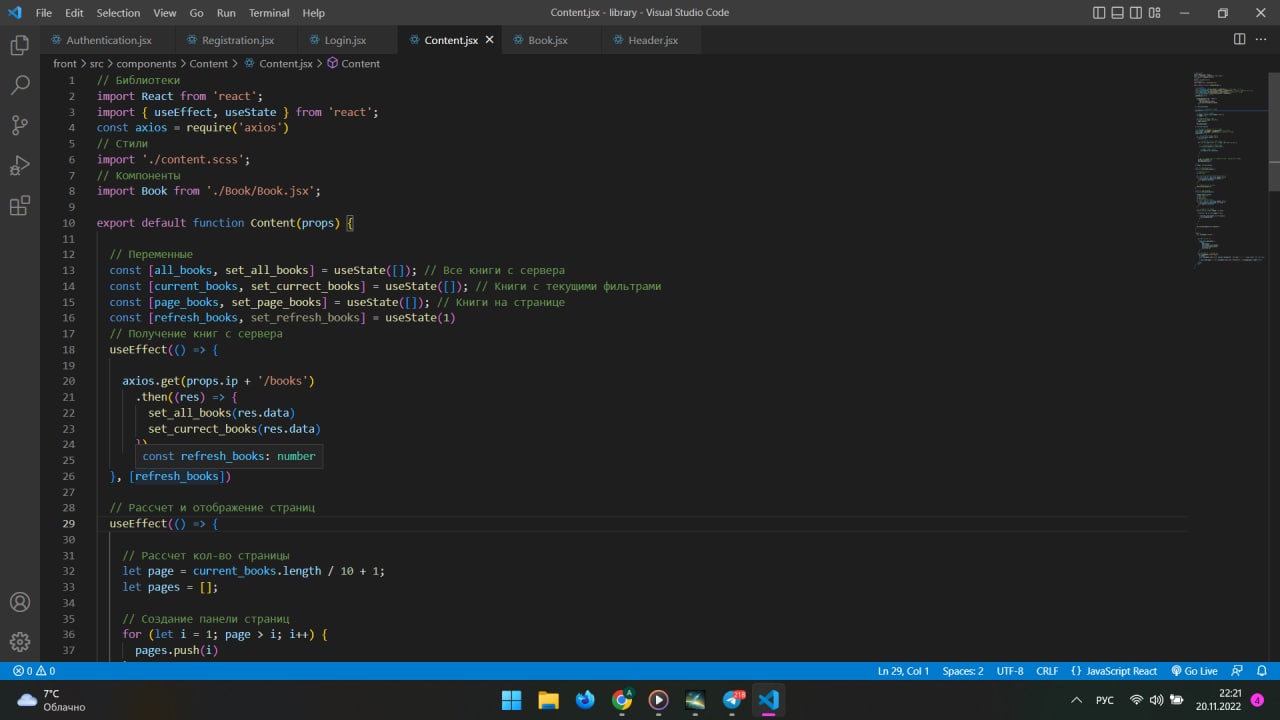
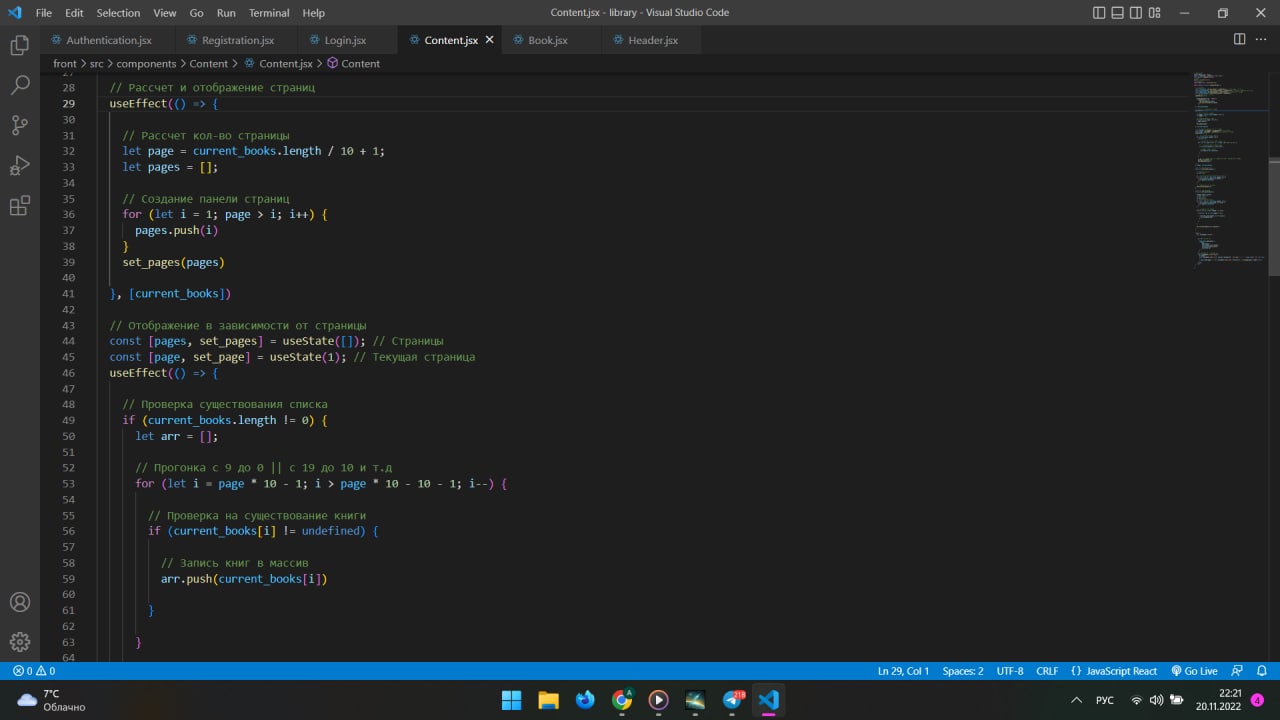
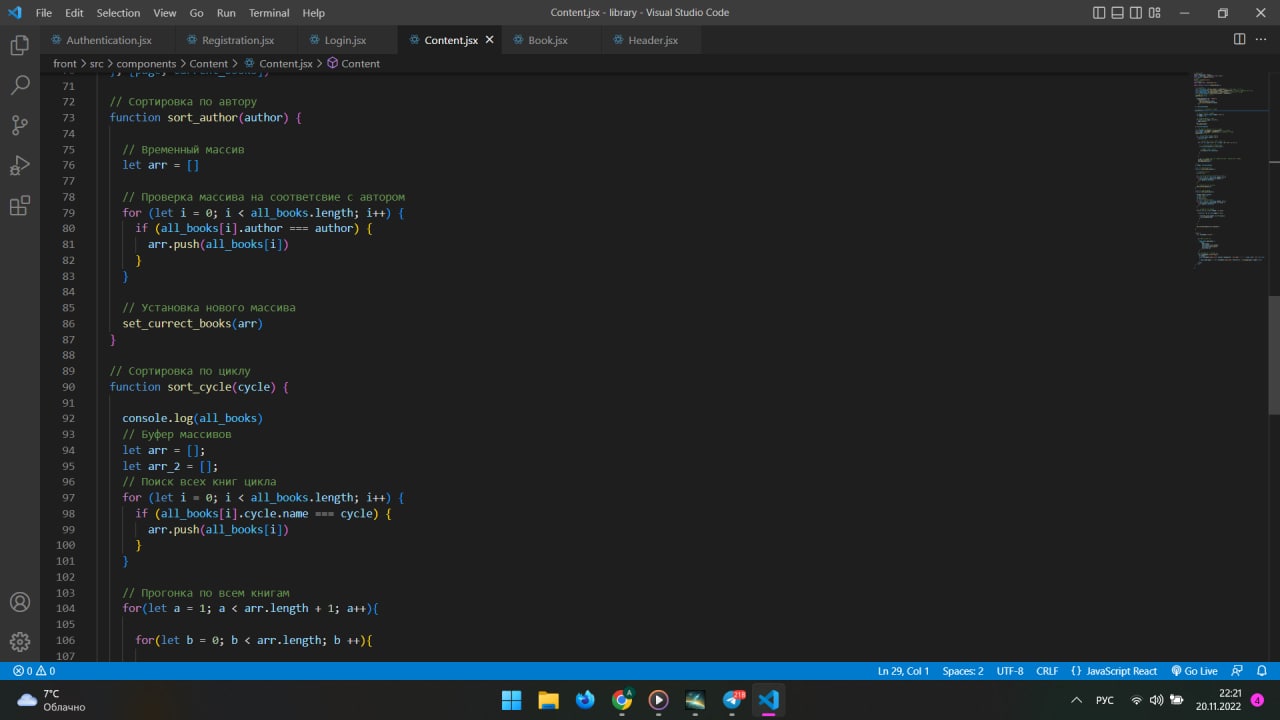


Рисунок А.8 – Контент страници и саписок книг

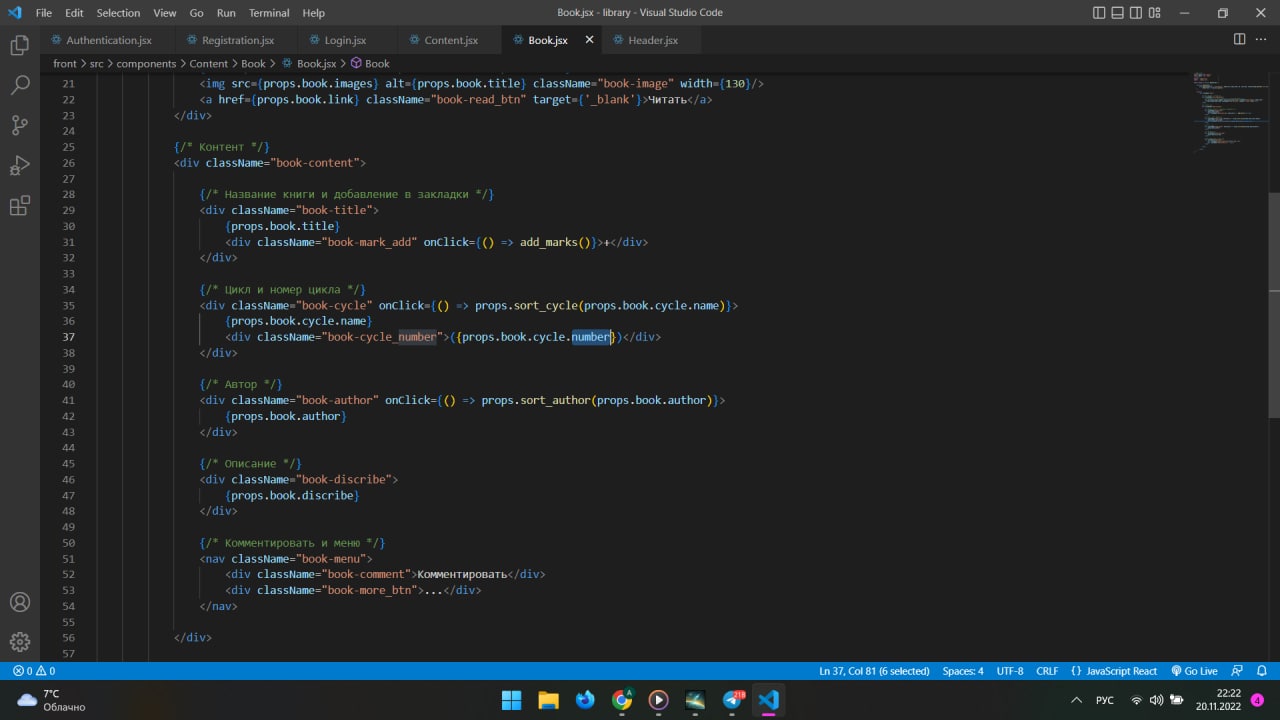
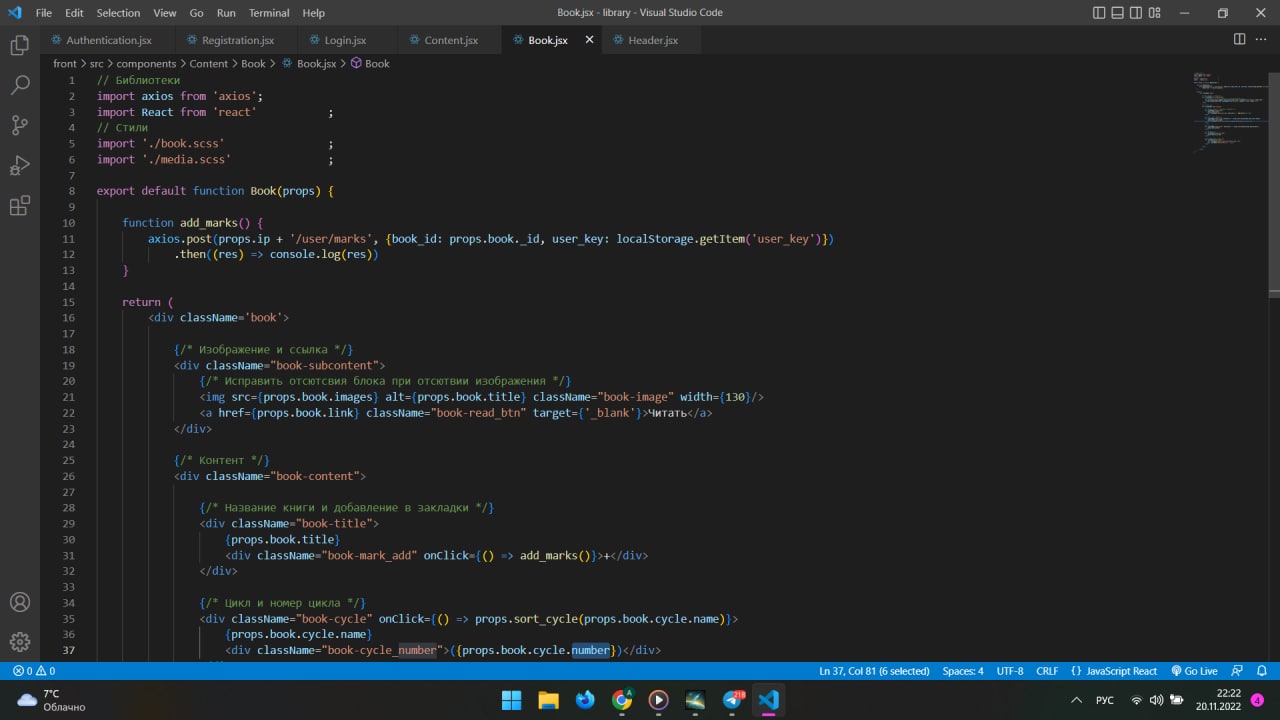


Рисунок А.9 – Книга на сайте

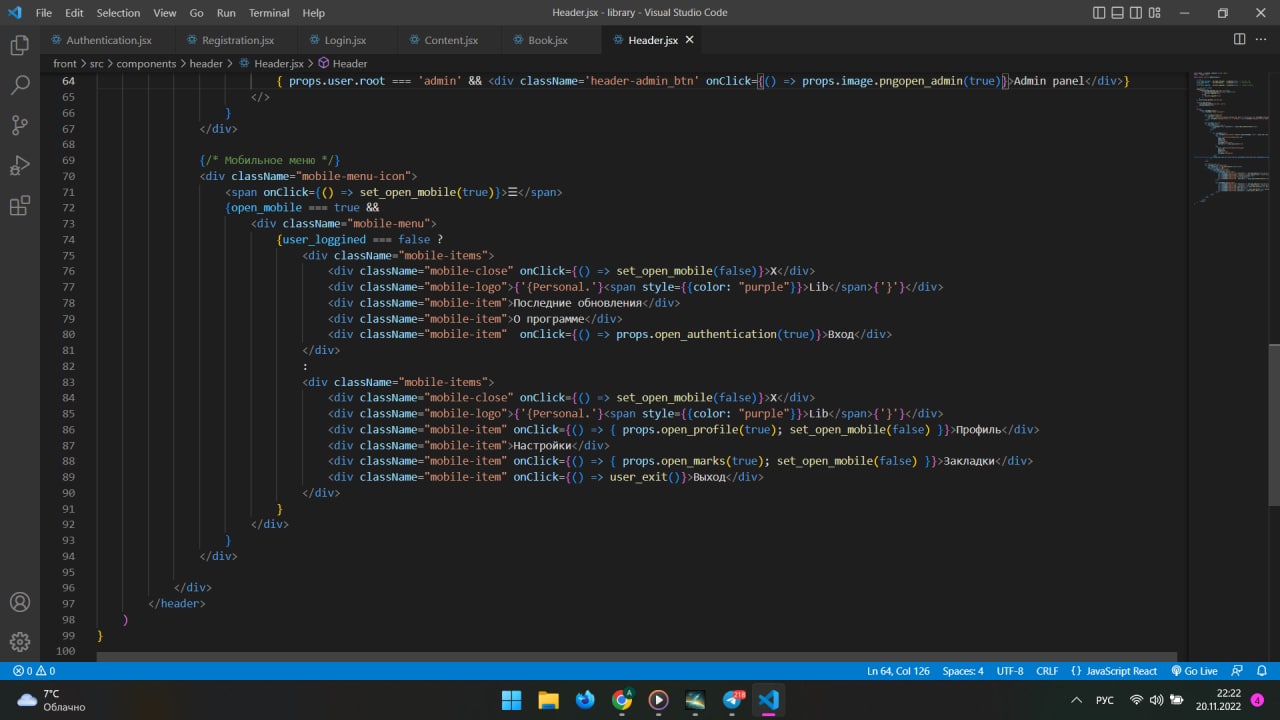
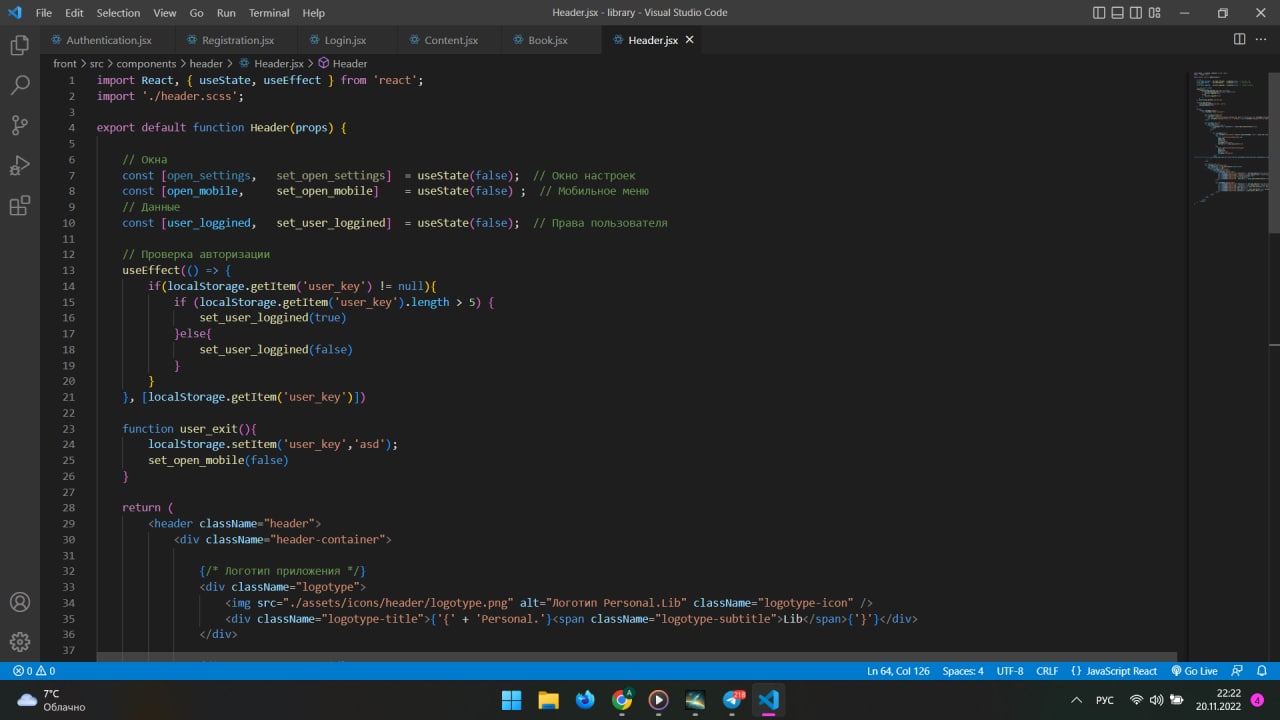
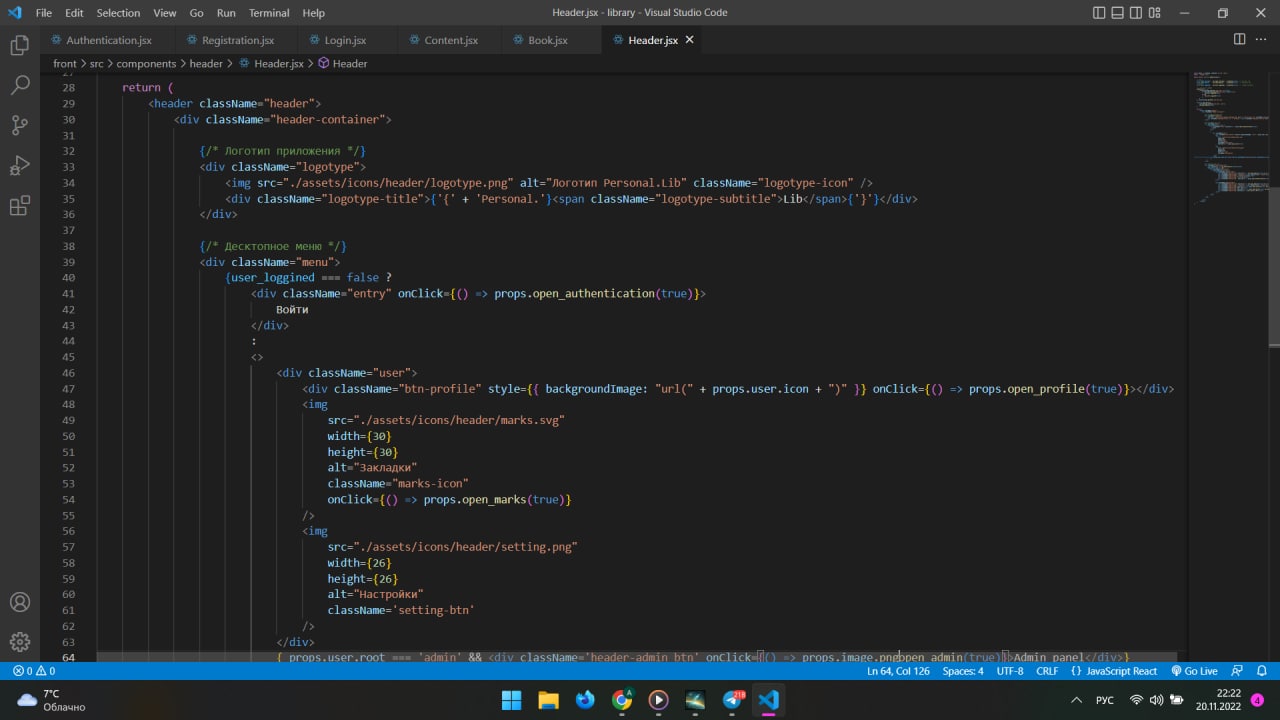
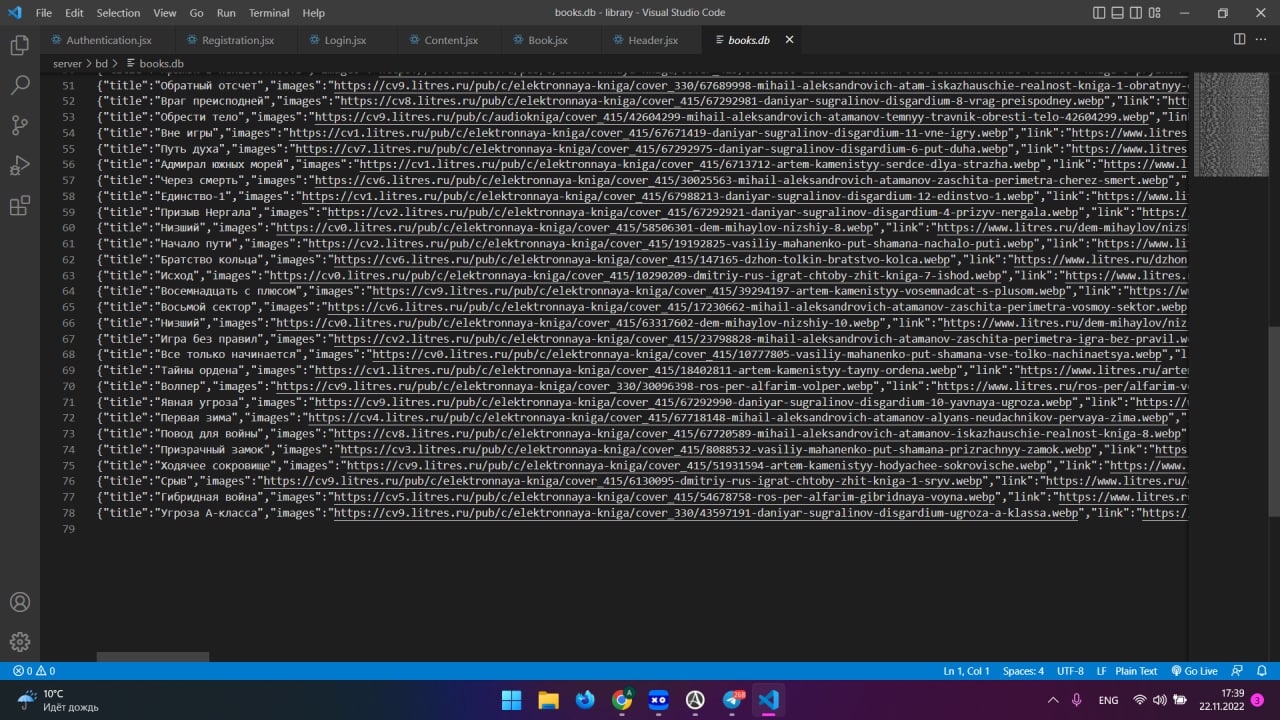


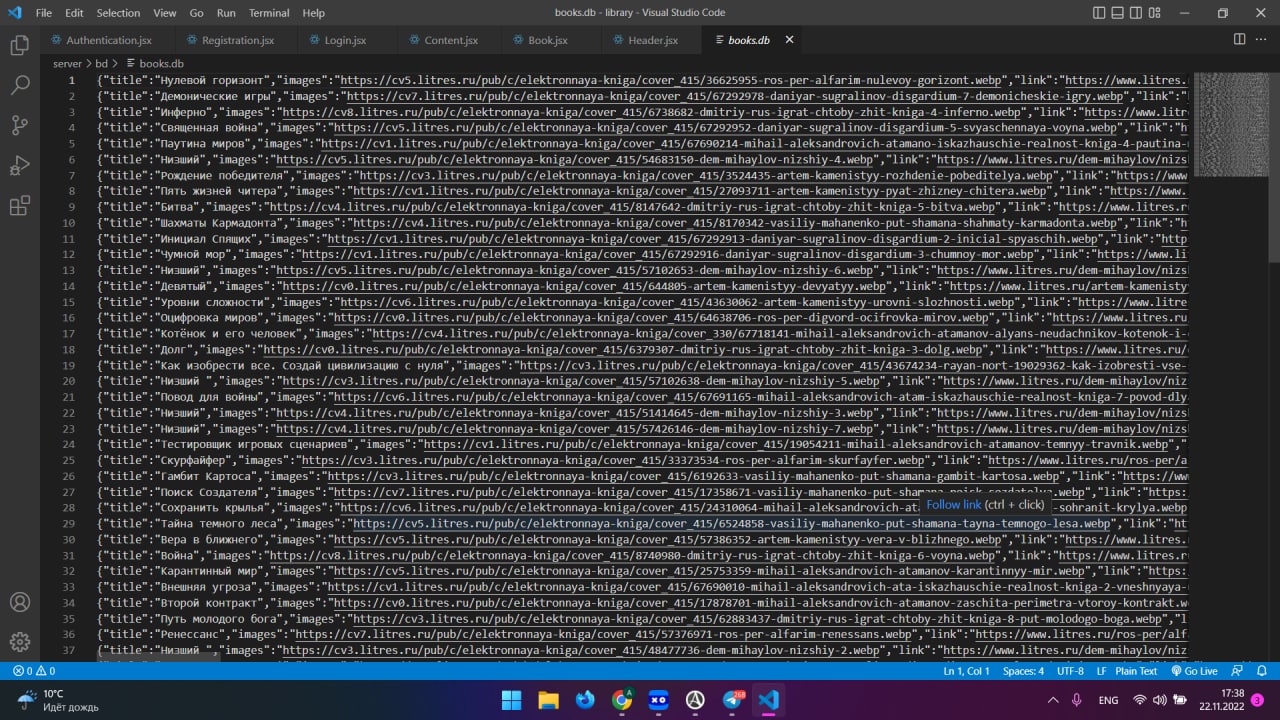
Рисунок А.10 – Шапка сайта

# Приложение Б

Структура (проекта) таблицы базы данных «Library Student» в среде JSON с указанием типов данных, а также первичных и внешних ключей (связей) представлена на рисунках А.1-А.2.



A.1 – Компоненты и состовлящие изданий.



A.2 – Основаная информация (содержание издания).

# Приложение В

Концептуальная модель (conceptual model) — это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта.

Диаграмма развертывания (deployment diagram) – диаграмма, на которой представлены узлы выполнения программных компонентов реального ПО.

Блок-схема — распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

Бэклог проекта (backlog) — это список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации.

Пользовательские истории (User Story) — способ описания требований к разрабатываемой системе, сформулированных как одно или более предложений на ежедневном или деловом языке пользователя.

Burndown chart - всем известный и широко используемый способ, с помощью которого команда отслеживает темп выполнения спринта.

Impact Mapping – это подход к стратегическому планированию, позволяющий построить логическую цепочку от бизнес-целей в голове у продакт-менеджера до изменений в продукте, необходимых для достижения поставленных целей.