### Частное учреждение образования

### Колледж бизнеса и права

### СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЕ «АВТОМОБИЛЬНЫЙ САЙТ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### к курсовому проектированию по дисциплине

### «Практика веб-программирования»

Руководитель проекта (Д.А.Бункевич )

Учащийся (Е.И.Войтехович )

Минск, 2024

**СОДЕРЖАНИЕ**

Изм.

Изм.

Лист

Лист

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

3

3

КП Т.19604.401 ПЗ

ПЗ

ДП Т.517074.401 ПЗ

Разраб.

Разраб.

Макатревич Д.В.

Мовчан П.И.

Провер.

Провер.

Бункевич Д.А.

Багласова Е.В.

Т. контр.

Т. контр.

*.*

*Якимович К.О.*

Н. контр.

Н. контр.

Багласова Е.В.

Утверд.

Утверд.

*Багласова Т.Г.*

### *СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ «РЕЦЕНЗИИ НА КНИГИ»*

*Программное средство для автоматизации учета заявок жителей Фрунзенского района г.Минска*

Лит.

Лит.

Листов

Листов

22

54

КБП

КБП

[Введение 3](#_Toc169936802)

[1. Описание задачи 4](#_Toc169936803)

[1.1. Анализ предметной области 4](#_Toc169936804)

[1.2. Постановка задачи 4](#_Toc169936805)

[2. Проектирование веб-приложения 6](#_Toc169936806)

[2.1. Проектирование модели 6](#_Toc169936807)

[2.2. Требования к веб-приложению 8](#_Toc169936808)

[2.3. Структура веб-приложения 9](#_Toc169936809)

[2.4. Структура проектирование макета веб-приложения 9](#_Toc169936810)

[2.5. Программно-технические средства, необходимые для разработки приложения 12](#_Toc169936811)

[3. Реализация веб-приложения 14](#_Toc169936812)

[3.1. Инструменты разработки и применения технологии Описание разделов веб-приложения 14](#_Toc169936813)

[3.2. Описание используемых функций и процедур 14](#_Toc169936814)

[3.3. Функциональное тестирование 14](#_Toc169936815)

[4. Применение 16](#_Toc169936816)

[4.1. Назначение программного средства 16](#_Toc169936817)

[4.2. Условия применения 17](#_Toc169936818)

[Заключение 18](#_Toc169936819)

[Список использованных источников 19](#_Toc169936820)

[Приложение А Текст программных модулей 20](#_Toc169936821)

[Приложение В Результат работы программы 22](#_Toc169936822)

# **Введение**

* Создание веб-приложения для публикации отзывов на автомобили, которое позволит пользователям делиться своими мнениями о различных моделях автомобилей, а также искать и выбирать автомобили.
* В эпоху цифровизации, когда количество моделей автомобилей, доступных на рынке, растет с каждым днем, возникает потребность в надежной и удобной платформе для обмена мнениями об автомобилях. Веб-приложение для отзывов на автомобили отвечает на эту потребность, предоставляя автолюбителям инструмент для информированного выбора транспортных средств и общения с единомышленниками, а производителям и дилерам — ценный обратный отклик от аудитории.
* Это приложение будет способствовать культуре осознанного выбора автомобилей, помогая пользователям находить модели, которые соответствуют их вкусам и требованиям, и облегчая процесс выбора среди огромного количества новых и б/у автомобилей. Кроме того, автоматизация процессов управления отзывами и комментариями позволит упростить работу с контентом и повысить уровень взаимодействия и удовлетворенности пользователей.
* Пояснительная записка к курсовому проекту содержит пять разделов:
* Описание задачи: В этом разделе раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть решены. Описываются существующие аналоги.
* Проектирование системы: В этом разделе перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются требования к приложению, строится концептуальный прототип, описывается логическая и физическая организация данных, проектируется справочная система.
* Описание реализации программного средства: В этом разделе представлены общие сведения о программном средстве и его функциональном назначении, входные и выходные данные.
* Применение: Этот раздел предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы.
* Заключение: В заключении будет проанализировано созданное программное средство, определена степень соответствия поставленной задачи и выполненной работы.
* Приложение А будет содержать текст программы.
* Графическая часть содержит в себе все необходимые диаграммы для проектирования данного программного средства, такие как:
* ­ диаграмма вариантов использования;
* ­ диаграмма классов;
* ­ структура веб-приложения.

# **Описание задачи**

# **Анализ предметной области**

Исследование предметной области: Веб-приложение “Рецензии на книги” входит в сферу социальных медиа и электронной коммерции, где основными бизнес-процессами являются публикация контента пользователями, его ранжирование и агрегация, а также предоставление рекомендаций. Пользователи создают рецензии, оценивают книги, взаимодействуют друг с другом через комментарии и обсуждения. Издатели и авторы могут использовать платформу для продвижения своих книг и получения обратной связи от читателей.

Описание алгоритмов получения промежуточных и итоговых показателей: Алгоритмы включают в себя методы машинного обучения для анализа текста рецензий, вычисления среднего рейтинга книг, и алгоритмы коллаборативной фильтрации для предложения пользователю книг на основе его предпочтений и отзывов других пользователей с похожими вкусами. Итоговые показатели могут включать рейтинги книг, статистику посещаемости и вовлеченности пользователей, а также данные о продажах книг через аффилированные ссылки.

Обоснование необходимости компьютерной обработки информации или процессов: Компьютерная обработка информации позволяет автоматизировать сбор и анализ больших объемов данных, что невозможно выполнить вручную. Это обеспечивает более быстрый и точный поиск книг, адаптацию рекомендаций под индивидуальные предпочтения пользователей и улучшение пользовательского опыта за счет персонализации контента.

Определение потенциальной аудитории: Потенциальная аудитория веб-приложения «Рецензии на книги» включает в себя активных читателей, блогеров, критиков, издателей и авторов. Это люди, которые ищут платформу для обмена мнениями о книгах, получения рекомендаций и взаимодействия с сообществом единомышленников. Также это могут быть новые или случайные читатели, ищущие рецензии и советы для выбора книги для чтения.

Этот анализ помогает определить ключевые функции и направления развития веб-приложения, а также стратегии привлечения и удержания пользователей.

# **Постановка задачи**

* На основе проанализированных бизнес-процессов в веб-приложении «Отзывы на автомобили», следующие функции должны быть автоматизированы;
* Публикация отзывов: пользователи должны иметь возможность легко публиковать свои отзывы на автомобили.
* Поиск по категориям: поиск автомобилей по различным критериям, таким как марка, модель, тип кузова и т.д.
* Интерактивные элементы: комментарии, обсуждения и социальные функции для взаимодействия с сообществом.

Существующие аналоги, такие как CarGurus или Edmunds, предлагают широкий спектр функций для поиска и обсуждения автомобилей. Однако, разрабатываемое веб-приложение будет иметь следующие отличия:

Персонализированный интерфейс: более интуитивно понятный и настраиваемый интерфейс, который адаптируется под предпочтения пользователя.

Улучшенные алгоритмы рекомендаций: использование последних достижений в области машинного обучения для предоставления более точных рекомендаций.

Интеграция с социальными сетями: упрощенный процесс обмена отзывами и мнениями в социальных сетях.

Дополнительные ресурсы: включение видеообзоров, сравнительных тестов, а также интеграция с онлайн-магазинами для покупки автомобилей и аксессуаров.

Сайт «Отзывы на автомобили» уже имеет несколько ключевых элементов, таких как закрепленный хедер, футер, форма отзывов и страница контактов. Для функции поиска по категориям можно рассмотреть следующие категории:

Эти функции и отличия помогут веб-приложению выделиться на фоне конкурентов и предложить пользователям уникальный и богатый опыт.

# **Проектирование веб-приложения**

# **Проектирование модели**

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями.

Наиболее распространенные средства моделирования являются: диаграмма вариантов использования, диаграмма классов, диаграмма последовательности, диаграмма деятельности.

Диаграмма вариантов использования – это тип диаграммы, используемой в области разработки программного обеспечения для визуализации системы с точки зрения её акторов и вариантов использования.

Основные элементы диаграммы вариантов использования:

* актеры: представляют собой людей или системы, которые взаимодействуют с системой. Они находятся вне системы и могут быть как людьми (например, пользователи или администраторы), так и другими системами или устройствами;
* варианты использования: - это функции или действия, которые система может выполнять в ответ на запросы от акторов. Они представляют собой цели, которые акторы хотят достичь с помощью системы;
* отношения: Отношения между акторами и вариантами использования показывают, как акторы и варианты использования взаимодействуют друг с другом. Они могут быть обозначены стрелками или линиями.

Диаграмма вариантов использования помогает разработчикам и стейкхолдерам лучше понять, как система будет работать и какие функции она будет выполнять. Она также помогает определить требования к системе и спланировать её разработку. Это важный инструмент в процессе проектирования системы.

Диаграмма вариантов использования представлена на рисунке 2.1

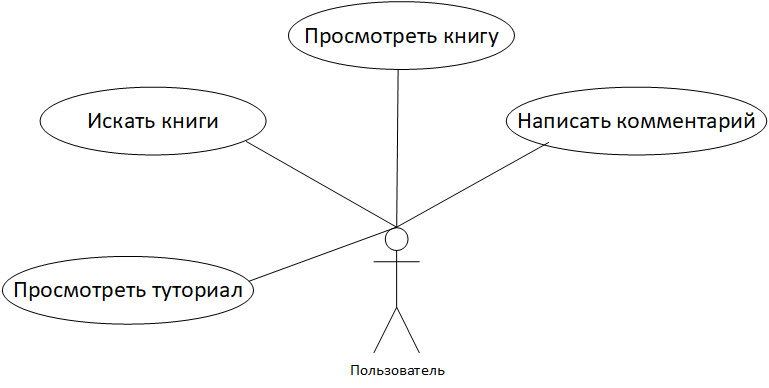


Рисунок 2.1 – Диаграмма вариантов использования

Диаграмма вариантов использованияпредставлена на листе 1 графической части.

На этой диаграмме можно выделить пользователя сайта. Пользователь, который смотрит книги, оставляет отзывы, ищет книги и смотрит видео на сайте.

Структура сайта — это иерархическая модель организации контента и страниц на веб-сайте, которая позволяет пользователям и поисковым системам находить информацию эффективно. Она отражает способ, которым различные части сайта связаны между собой и как они взаимодействуют. Вот основные элементы структуры сайта:

Структура сайта должна быть логичной, интуитивно понятной и масштабируемой, чтобы обеспечить удобство использования и оптимизацию для поисковых систем. Она также должна учитывать цели бизнеса и потребности пользователей. Создание четкой и эффективной структуры сайта является ключевым аспектом веб-разработки и помогает обеспечить успешное взаимодействие с посетителями.

Структура сайта представлена на рисунке 2.2.

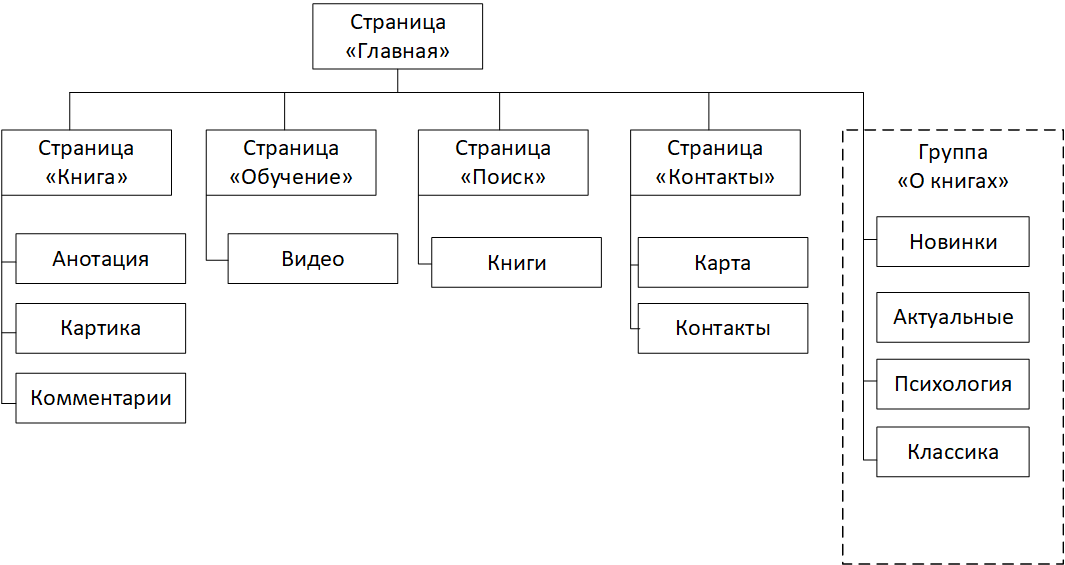


Рисунок 2.3 – Структура сайта

# **Требования к веб-приложению**

Стиль сайта должен соответствовать тематике и целевой аудитории. Это включает в себя выбор цветовой палитры, которая должна быть приятной для глаз и не отвлекать от контента. Элементы дизайна должны быть сдержанными, но в то же время выразительными, чтобы подчеркивать индивидуальность сайта.

Графика должна быть высокого качества и оптимизирована для быстрой загрузки. Изображения обложек книг, авторов и иллюстрации должны быть четкими и актуальными. Важно также обеспечить единообразие стилей для всех графических элементов.

Шрифты должны быть читабельными и доступными. Рекомендуется использовать не более трех различных шрифтов на сайте: один для заголовков, один для основного текста и один для акцентов. Размер шрифта должен обеспечивать комфортное чтение на любом устройстве.

 Веб-приложение должно корректно отображаться во всех современных браузерах и быть адаптированным для мобильных устройств. Важно регулярно тестировать сайт на различных платформах и разрешениях экранов.

Контент должен быть полным, актуальным и профильтрованным. Обзоры книг должны быть информативными и содержать все необходимые детали, такие как название, автор, жанр, краткое содержание и оценка. Важно также предусмотреть возможность пользовательских отзывов и рейтингов.

CMS должна быть интуитивно понятной и обеспечивать легкость управления контентом. Необходимо предусмотреть функции для модерации отзывов, управления пользователями и аналитики посещаемости.

Страницы должны быть структурированы таким образом, чтобы пользователи могли легко находить нужную информацию. Важно обеспечить логичную иерархию информации и использовать адаптивный дизайн для удобства просмотра на разных устройствах.

# **Структура веб-приложения**

Графическое представление логической структуры представлено на рисунке 2.3

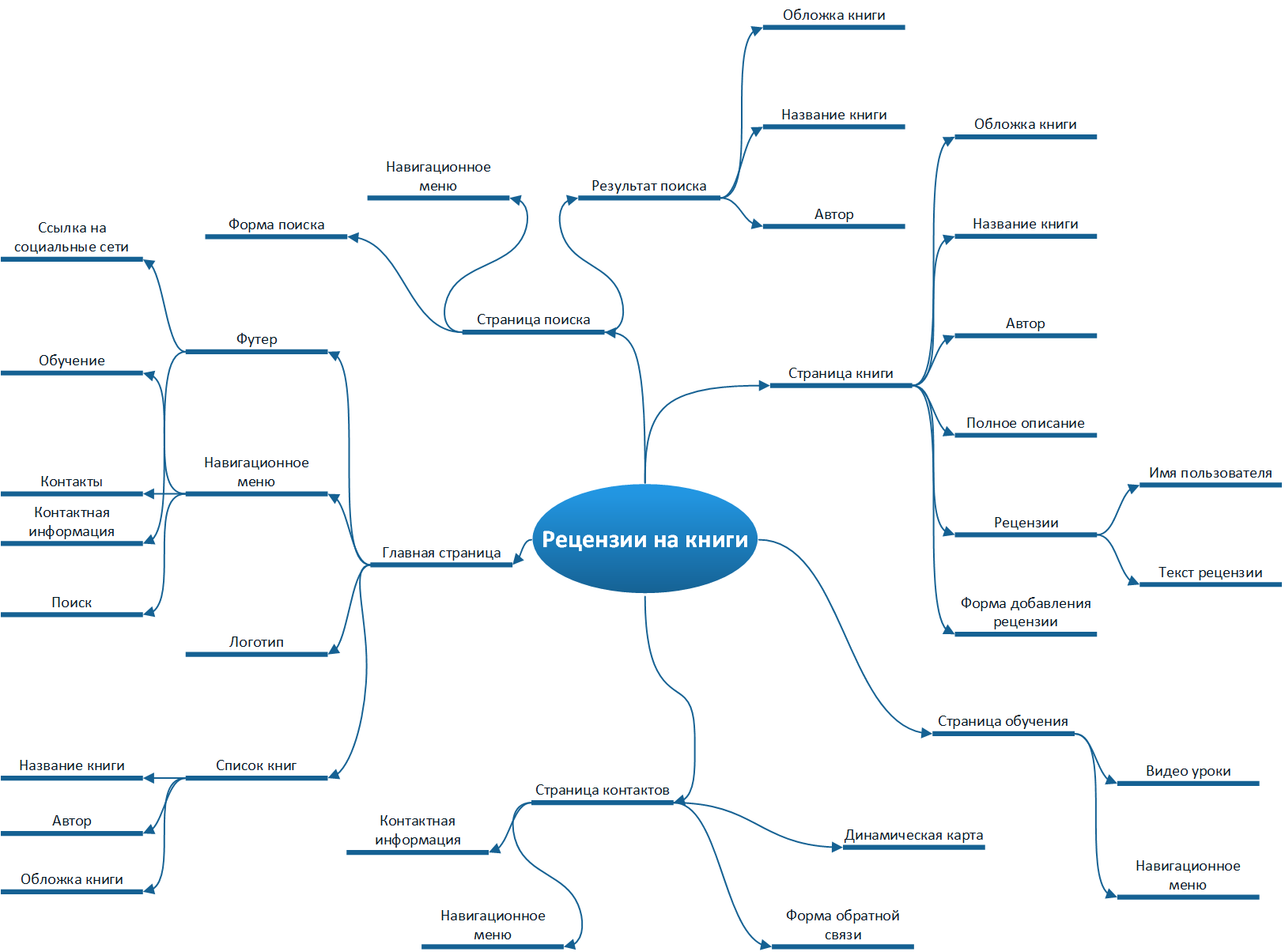


Рисунок 2.3 - Ментальная карта веб-приложения

# **Структура проектирование макета веб-приложения**

Веб-сайт использует блочный дизайн. Это означает, что элементы на странице организованы в блоки, которые могут быть легко масштабированы и перемещены для различных размеров экрана и ориентаций. Блочный дизайн обеспечивает гибкость и адаптивность, что делает его идеальным для современных веб-сайтов. Блочный дизайн в веб-приложении имеет ряд преимуществ, которые делают его популярным выбором среди разработчиков:

Гибкость: Блочный дизайн позволяет легко масштабировать, перемещать и изменять размеры элементов на странице. Это обеспечивает большую гибкость при создании адаптивных веб-сайтов, которые выглядят хорошо на различных устройствах и размерах экрана.

Читаемость: Блоки помогают организовать контент на странице, делая его более читаемым и понятным для пользователей. Они обеспечивают четкую визуальную иерархию и помогают направлять внимание пользователя на важные элементы.

Удобство обслуживания: Блочный дизайн делает код более структурированным и легким для понимания. Это упрощает обслуживание и обновление сайта в будущем.

Совместимость с CSS: Блочный дизайн хорошо работает с CSS, которые предоставляют мощные инструменты для создания сложных блочных макетов.

Адаптивность: Блочный дизайн поддерживает адаптивные веб-дизайны, которые автоматически адаптируются к размеру экрана пользователя. Это обеспечивает оптимальный пользовательский опыт на любом устройстве.

В целом, блочный дизайн – это эффективный подход к веб-дизайну, который обеспечивает гибкость, читаемость и удобство обслуживания. Он подходит для создания современных, адаптивных веб-сайтов и веб-приложений.

Графическое представление главной страницы веб-приложения представлено на рисунке 2.4

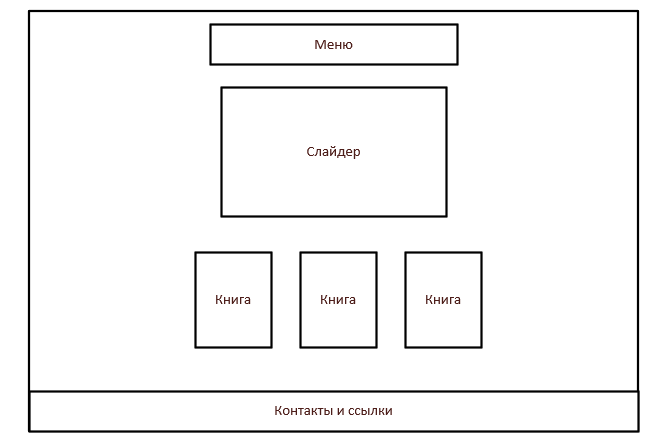


Рисунок 2.4 – Главная страница веб-приложения

На главной форме находятся такие элементы:

* меню: содержит навигационные ссылки на другие разделы сайта;
* слайдер: содержит ссылки на другие разделы сайта;
* книга: содержит в себе картинку книги, автора, название книги и ссылку на личную страницу книги;
* контакты и ссылки: содержит контактную информацию и ссылки на социальные сети.

Графическое представление второстепенной страницы веб-приложения представлено на рисунке 2.5

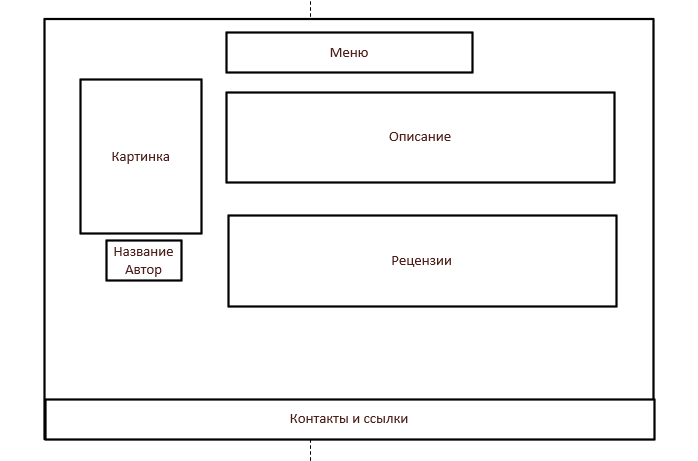


Рисунок 2.5 – Второстепенная страница веб-приложения

На второстепенной форме находятся такие элементы:

* меню: содержит навигационные ссылки на другие разделы сайта;
* картинка: содержит обложку книги;
* описание: содержит в себе аннотацию книги;
* рецензии: содержат в себе комментарии, которые пользователи могут оставлять;
* контакты и ссылки: содержит контактную информацию и ссылки на социальные сети.

# **Программно-технические средства, необходимые для разработки приложения**

Для создания этого веб-приложения понадобились такие языки программирования и инструменты разработки как:

* html;
* css;
* javaScript;
* visual studio code.

Visual Studio Code: Visual Studio Code - это бесплатный редактор кода с открытым исходным кодом от Microsoft. Он поддерживает множество языков программирования и фреймворков, включая HTML, CSS и JavaScript. Visual Studio Code предлагает такие функции, как подсветка синтаксиса, интеллектуальное автозавершение кода, поддержка Git и многие другие.

HTML, CSS и JavaScript - это стандартные технологии для разработки веб-страниц и веб-приложений. Они широко поддерживаются всеми современными веб-браузерами и обеспечивают все необходимые инструменты для создания функциональных и визуально привлекательных веб-сайтов.

Visual Studio Code был выбран в качестве инструмента разработки из-за его гибкости, мощных функций и поддержки большого количества расширений, которые могут упростить и ускорить процесс разработки.

Ваш веб-сайт не поддерживает сетевые функции.

# **Реализация веб-приложения**

# **Инструменты разработки и применения технологии Описание разделов веб-приложения**

Главная страница:

* Шапка сайта с логотипом и названием “DoshikLib”, которое также является ссылкой на главную страницу;
* Навигационное меню с пунктами “Контакты”, “Поиск” и “Обучение”. Каждый пункт ведёт на соответствующую страницу;
* Слайдер с четырьмя слайдами, каждый из которых содержит заголовок и предназначен для презентации контента;
* Список книг, разделённый на категории “Актуальные” и “Новинки”, с изображениями обложек, названиями и авторами книг.

Навигационное меню:

* Контакты: перенаправляет пользователя на страницу с контактной информацией;
* Поиск: позволяет пользователю искать книги по сайту;
* Обучение: содержит материалы или инструкции для пользователей, желающих узнать больше о литературе или использовании сайта.

# **Описание используемых функций и процедур**

В данном проекте реализованы функции добавления комментариев:

Функция добавления комментариев находиться на сайтах книг. После заполнения всех полей и нажатия на кнопку «Добавить» на данной форме происходит обработка события Button\_Click\_AddBox. Все данные, вводимые в поля формы, будут сохранены и видны пользователю.

Листинг функций представлен в приложении А

# **Функциональное тестирование**

Функциональное тестирование – это тестирование функций приложения на соответствия требованиям. Оценка производится в соответствии с ожидаемым и фактическим результатом.

Тестирование программы будет производится последовательно, переходя из одной программы в другую. Во время теста будут проверятся все действия с программой, навигация пунктов меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

Первый тест будет проведён в окне авторизации программы.

Тест-кейс для тестирования функции добавления комментария представлен в таблице 3.1

Таблица 3.1 – Тест-кейс функции добавления

|  |  |  |  |
| --- | --- | --- | --- |
| № Тест-кейса | Функция | Шаги выполнения | Результат |
| 1 | Добавление комментариев | 1.Открыть страницу книги.  2. Ввести комментарий:  - комментарий: «Класс, книга хорошая. Оценю как 1 из 10»  - имя: «Андрюшка» | Ожидаемый результат: добавление комментария |
| Фактический результат: успешное создание комментария |

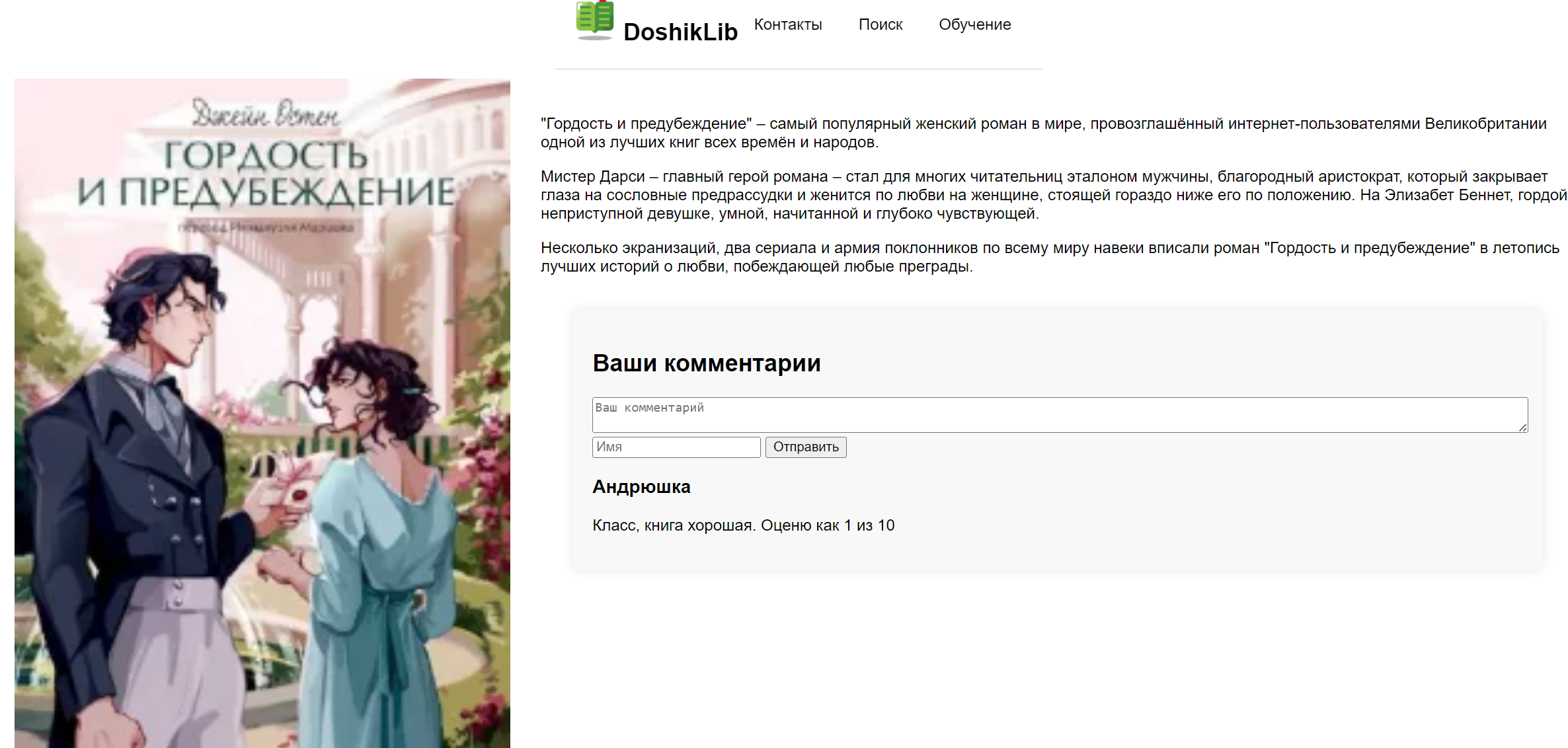


Рисунок 3.1- результат тест кейса

# **Применение**

# **Назначение программного средства**

Веб-приложение для рецензий на книги предназначено для создания платформы, где пользователи могут читать и публиковать рецензии на книги, обсуждать литературные произведения и делиться мнениями. Оно служит централизованным ресурсом для любителей книг и помогает в выборе литературы для чтения.

Цель разработки:

* Создать удобный и доступный ресурс для обмена мнениями о книгах.
* Обеспечить возможность обсуждения и оценки литературных произведений.
* Содействовать продвижению чтения и литературного образования.

Установка веб-приложения:

* Развертывание файлов на веб-сервере или хостинге.

Дерево файлов представлено на рисунке 4.1

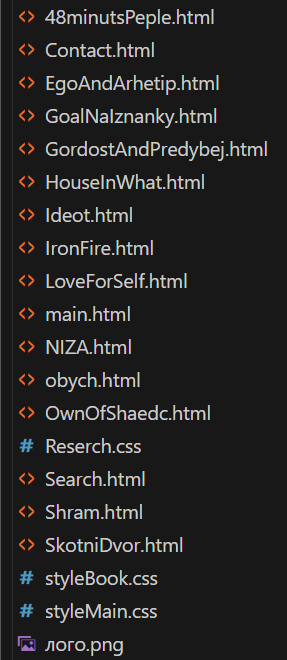


Рисунок 4.1 – Дерево файлов

Запуск веб-приложения:

* Открыть браузер и ввести URL веб-приложения.
* Веб-приложение должно автоматически загрузиться и отобразить главную страницу.

Область применения и класс решаемых задач:

* Предоставление информации о книгах и их рецензиях.
* Формирование сообщества любителей книг.
* Образовательные цели, такие как изучение литературы и критического анализа.

Ограничения:

* Веб-приложение ограничено функциональностью, предоставляемой текущей технологией веб-разработки.
* Требуется постоянное подключение к интернету для доступа к содержимому.
* Зависимость от качества и объективности пользовательских рецензий.

# **Условия применения**

Для применения данного программного средства необходимы следующие технические требования:

* процессор Intel Core i5 или выше;
* минимальный объём оперативной памяти 500 Мбайт;
* оперативная система Windows 11;
* платформа .Net Framework v4.8;
* видеокарта с объёмом памяти не менее 256 Мбайт;
* клавиатура, мышь, принтер.

# **Заключение**

Разработать веб-приложение для рецензий на книги, которое позволяет пользователям читать, публиковать рецензии и обсуждать литературные произведения.

Веб-приложение было разработано с использованием HTML, CSS и JavaScript, что обеспечило его интерактивность и удобство использования. Все ключевые функции были реализованы, включая навигацию по сайту, отображение книг, поиск и систему комментариев.

Разработанные функции:

Навигационное меню для удобного перехода между разделами сайта.

Слайдер для презентации избранных книг и новинок.

Список книг с возможностью перехода на страницу с детальной рецензией.

Форма для публикации комментариев с сохранением в localStorage.

Адаптивный дизайн, обеспечивающий корректное отображение на различных устройствах.

Преимущества:

Интуитивно понятный интерфейс.

Возможность обмена мнениями и обсуждения книг.

Легкость доступа и использования без необходимости регистрации.

Недостатки:

Отсутствие серверной части ограничивает возможности масштабирования и управления данными.

Зависимость от localStorage может привести к потере комментариев при очистке кэша браузера.

Необходимость постоянного подключения к интернету для доступа к контенту.

# **Список использованных источников**

1. Ахромов Я.В. Технологии Web-дизайна и Flash-технологии: учеб. пособие / Я.В. Ахромов. - Ростов н/Д: Феникс, 2009. - 265 с.
2. Алексеев Ю.М. Быстро и легко создаем, программируем, шлифуем и раскручиваем web-сайт: учеб. пособие / Ю.М. Алексеев. - М.: Лучшие книги, 2011. - 189 с.
3. Белунцов В. Новейший самоучитель по созданию Web-страниц / В. Белунцов. - М.: NT Press, 2012. - 185 с.
4. Благодатских В.А. Стандартизация разработки программных средств: учеб. пособие / В.А. Благодатских. - М.: Финансы и статистика, 2011. - 544 с.

# **Приложение А**

**(обязательное)**

**Текст программных модулей**

    <script>

 window.onload = function() {

          var pageCommentsKey = 'comments-' + location.pathname;

          var comments = JSON.parse(localStorage.getItem(pageCommentsKey)) || [];

          comments.forEach(function(comment) {

            var commentElement = document.createElement('div');

            commentElement.innerHTML = '<h3>' + comment.name + '</h3><p>' + comment.comment + '</p>';

            document.getElementById('comments').appendChild(commentElement);

          });

        };

        document.getElementById('commentForm').addEventListener('submit', function(event) {

          event.preventDefault();

          var commentText = document.getElementById('commentText').value;

          var name = document.getElementById('name').value;

          var commentElement = document.createElement('div');

          commentElement.innerHTML = '<h3>' + name + '</h3><p>' + commentText + '</p>';

          document.getElementById('comments').appendChild(commentElement);

          var pageCommentsKey = 'comments-' + location.pathname;

          var comments = JSON.parse(localStorage.getItem(pageCommentsKey)) || [];

          comments.push({name: name, comment: commentText});

          localStorage.setItem(pageCommentsKey, JSON.stringify(comments));

          document.getElementById('commentText').value = '';

          document.getElementById('name').value = '';

        });

 </script>  <script>

        let currentFrame = 1;

        const totalFrames = 4;

        setInterval(() => {

            currentFrame = currentFrame % totalFrames + 1; // Обновляем текущий слайд

            document.getElementById(`frame${currentFrame}`).checked = true; // Переключаем радиокнопку

        }, 3000);

    </script>

# **Приложение В**

**(справочное)**

**Результаты работы приложения**

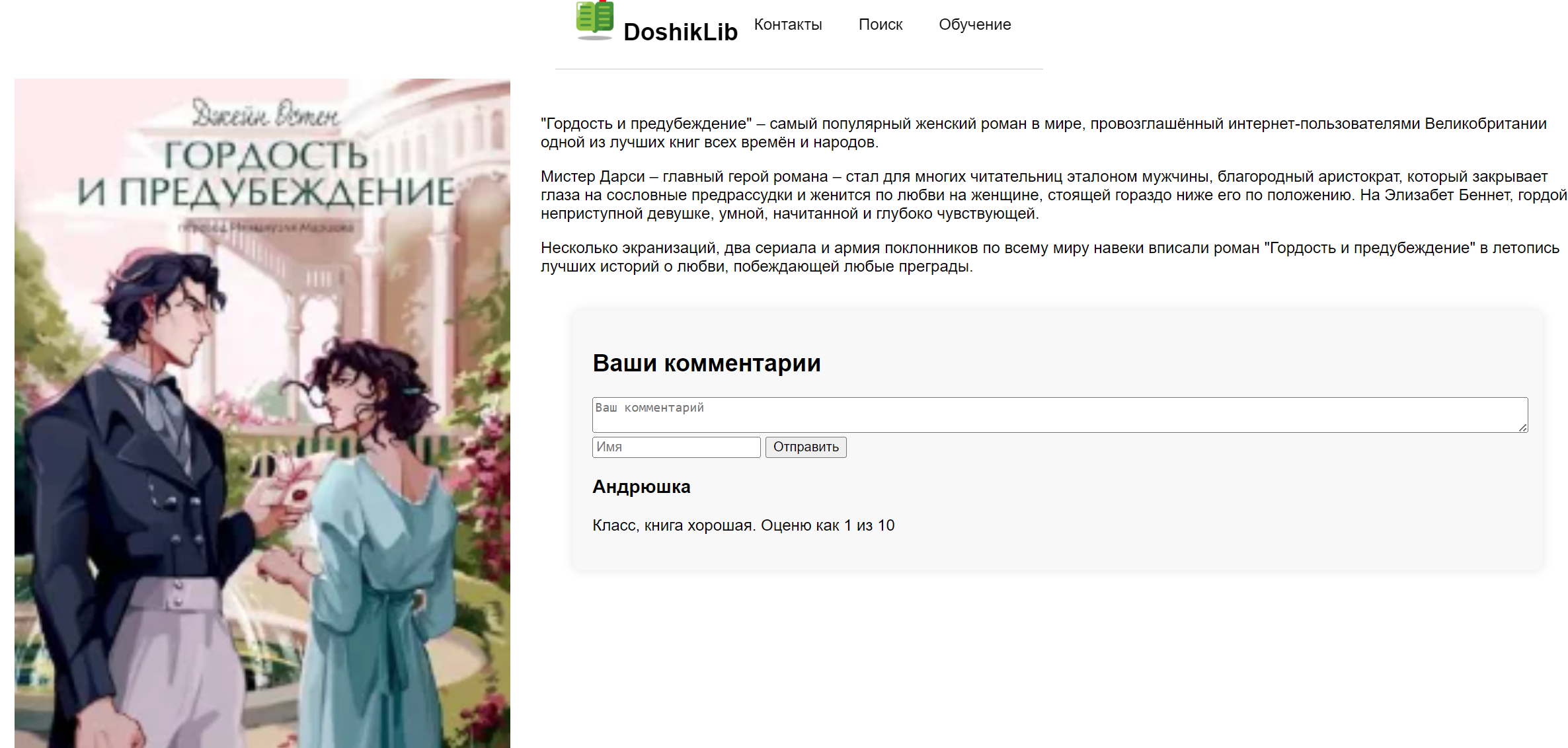


Рисунок Б.1- результат тест кейса