Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

по направлению: 09.03.04 Программная инженерия

(код и наименование направления)

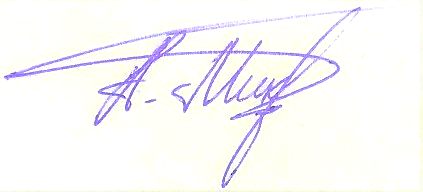
на тему: Разработка web-приложения «Литература». Frontend-разработка.

Комплексный проект.

(наименование темы)

  ИКСИБ.КИСП.09.03.04.002.ПП

(обозначение документа)



Автор  / А.А. Михалков / (подпись, дата, расшифровка подписи)

Руководитель   / д-р техн. наук, проф. В.Н. Марков / (подпись, дата, расшифровка подписи)

Консультанты:

Раздел безопасности и

экологичности проекта  / д-р техн. наук., проф. Т.Г. Короткова / (подпись, дата, расшифровка подписи)

Нормоконтролер  / ст. преп. А.А. Ковтун /

(подпись, дата, расшифровка подписи)

Выпускная квалификационная работа  
допущена к защите

(дата)

Заведующая кафедрой / канд. техн. наук, доц. М.В. Янаева /

(подпись)

Краснодар

2020

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

УТВЕРЖДАЮ

Заведующая кафедрой ИСП

канд. техн. наук, доц.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.В. Янаева

« 14 » мая 2020 г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

по направлению 09.03.04 Программная инженерия

(шифр и наименование)

студенту Михалкову Алексею Алексеевичу

(фамилия, имя, отчество)

Тема выпускной квалификационной работы: Разработка web-приложения

«Литература». Frontend-разработка. Комплексный проект.

Утверждена приказом ректора университета от 13.05.2020 г. № 650-Ст

Руководитель   д-р техн. наук, проф. В.Н. Марков

(должность, фамилия, инициалы)

Консультанты:

Раздел безопасности и

экологичности проекта д-р техн. наук, проф. Т.Г. Короткова

(должность, фамилия, инициалы)

Нормоконтролер ст. преп. А.А. Ковтун

(должность, фамилия, инициалы)

Срок сдачи выпускной квалификационной работы на кафедру 23.06.2020 г.

**Содержание выпускной квалификационной работы**

Введение

1 Нормативные ссылки

2 Термины, определения и сокращения

3 Анализ предметной области

4 Постановка задачи и техническое задание

5 Обзор используемых технологий и сред разработки

6 Проектирование и дизайн веб-страниц

7 Мобильная версия веб-сайта

8 Оценка эффективности программного продукта

9 Безопасность и экологичность проекта

Заключение

Список использованных источников

Общее количество листов ПЗ 69

**Объем иллюстративной части**

1. Основная концепция проекта.

2. Клиентская часть.

3. Мобильная версия

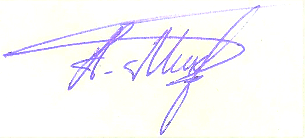
Общее количество слайдов иллюстративной части ХХ

**Список основной и рекомендуемой литературы**

1. Чамберс Дж., Пэкетт Д. Тиммс С. ASP.NET Core. Разработка приложений – Пер. с англ. – СПб : Питер, 2018 – 464 с.

**Календарный план выполнения выпускной квалификационной работы**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Месяц | Числа месяца | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | | 6 | 7 | 8 | 9 | 10 | 11 | 12 | | 13 | 14 | | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Март |  | | | | | | | | | | | | | | | | | | *Описание требований к ПП и постановка задачи ВКР* | | | | | | | *Анализ предметной области ВКР.*  *Техническое задание на ПП.* | | | | | | | | | |
| Апрель | *Анализ предметной области ВКР.*  *Техническое задание на ПП* | | | | | *Обзор существующих технологий и сред разработки* | | | | | | | | *Построение модели ПП.*  *Проектирование и реализация клиентской части ПП.* | | | | | | | | | | | | | | | | | | | | |  |
| Май | *Реализация программного обеспечения* | | | | | | | | | | | | | | | | | | | | | | | | | | | *Безопасность и экологичность проекта* | | | | | | | |
| Июнь | *Оформление пояснительной записки* | | | | | | | | | | | | | | | | *Нормоконтроль, антиплагиат. Сдача ВКР на кафедру.* | | | | | | | | | | | *Защита ВКР* | | | | | | |  |



Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Михалков Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_д.т.н., проф. В.Н. Марков

(подпись, дата) (подпись, дата)

**Реферат**

Выпускная квалификационная работа содержит 71 страницу, 41 рисунок, 2 таблицы, 8 листингов, 9 источников, 1 приложение.

ИНФОРМАЦИОННАЯ СИСТЕМА, ВЕБ-ДИЗАЙН, АДАПТИВНАЯ ВЁРСТКА, HTML, CSS, JAVASCRIPT, JQUERY, DOM, FRONTEND, MICROSOFT VISUAL STUDIO, ASP.NET CORE, ТЕГ, КЛАСС, МОДЕЛЬ, ПРЕДСТАВЛЕНИЕ, КОНТРОЛЛЕР, ВЕБ-БРАУЗЕР.

Целью выпускной квалификационной работы является разработка и реализация клиентской части веб-приложения «Литература»

В результате выполнения выпускной квалификационной работы разработана клиентская часть веб-приложения, являющаяся частью комплексного проекта «Литература».

Реализованная платформа предоставляет пользователям комфортный и легко восприимчивый пользовательский интерфейс вне зависимости от устройства, с которого выполняется вход на веб-ресурс.

Содержание

[Введение 8](#_Toc42720438)

[1 Нормативные ссылки 9](#_Toc42720439)

[2 Термины, определения и сокращения 10](#_Toc42720440)

[3 Анализ предметной области 12](#_Toc42720441)

[4 Постановка задачи и техническое задание 13](#_Toc42720442)

[4.1 Общие требования 13](#_Toc42720443)

[4.2 Платформа 14](#_Toc42720444)

[5 Обзор используемых технологий и сред разработки 15](#_Toc42720445)

[5.1 Язык HTML5 15](#_Toc42720446)

[5.2 Язык CSS3 16](#_Toc42720447)

[5.3 Язык программирования JavaScript 16](#_Toc42720448)

[5.4 Язык программирования C# 17](#_Toc42720449)

[5.5 Twitter Bootstrap 4 18](#_Toc42720450)

[5.6 Среда разработки Microsoft Visual Studio 2019 20](#_Toc42720451)

[5.7 Веб-фреймворк ASP.NET Core 21](#_Toc42720452)

[6 Проектирование и дизайн веб-страниц 25](#_Toc42720453)

[6.1 Мастер-страница 25](#_Toc42720454)

[6.2 Домашняя страница 28](#_Toc42720455)

[6.3 Каталог…………………………………………………………………..29](#_Toc42720456)

[6.4 Форум……………………………………………………………………36](#_Toc42720457)

[6.5. FAQ (Часто задаваемые вопросы) 43](#_Toc42720458)

[6.6 Раздел «Управление сайтом» 44](#_Toc42720459)

[6.7 Профиль, настройки, регистрация и авторизация 50](#_Toc42720460)

[7 Мобильная версия веб-приложения «Литература» 53](#_Toc42720461)

[8 Оценка эффективности программного продукта 56](#_Toc42720462)

[9 Безопасность и экологичность проекта 60](#_Toc42720463)

[9.1 Значение и задачи безопасности жизнедеятельности 60](#_Toc42720464)

[9.2 Анализ условий труда и мероприятия по защите от воздействия вредных производственных факторов 61](#_Toc42720465)

[9.3 Обеспечение электробезопасности 64](#_Toc42720466)

[9.4 Обеспечение пожарной безопасности 65](#_Toc42720467)

[Заключение 68](#_Toc42720468)

[Список использованных источников 69](#_Toc42720469)

[Приложение А 70](#_Toc42720470)

# Введение

В настоящее время все большее количество людей получает информацию при помощи электронных носителей, таких как персональные компьютеры, смартфоны и планшеты.

Одновременно с этим, несмотря на доступность информации, отмечается низкий уровень читательской культуры, в связи с такой причиной, как перегруженность школьных литературных программ. Из этой причины вытекает такое следствие, как снижение интереса школьников к изучению литературы. В свою очередь, из следствия выходит ещё одно – неумение учащихся словесно выражать свои мысли и/или чувства, неумение создать оригинальный текст и прочее [1].

Опираясь на исследование, представленное в источнике, указанном выше, возникла идея создать литературный клуб, объединив его с достижениями человечества в сфере информационных технологий, что стало еще более актуальным с связи с пандемией COVID-19.

Соответствуя возникшей идее, целью выпускной квалификационной работы является разработка веб-приложения на соответствующую тематику. Используя данное веб-приложение, предполагается, что пользователь будет иметь возможность найти необходимую информацию об интересующей книге, оставить о ней отзыв и участвовать в форумных дискуссиях относительно литературной тематики. Также, учитывая то, что в исследовании было упомянуто также и про развитие талантов у пишущей молодежи, разрабатываемое приложение будет только поощрять публикацию творчества собственного сочинения.

В рамках данного комплексного проекта необходимо реализовать клиентскую часть веб-приложения, позволяющая осуществить связь между пользователем и серверной частью веб-приложения.

Выполнение ВКР поспособствует повышению квалификации в области программировании, в частности, во frontend-разработке и веб-дизайне.

# 1 Нормативные ссылки

В настоящей выпускной квалификационной работе использованы ссылки на следующие нормативные документы:

1. ГОСТ Р 1.12-2004. Стандартизация в Российской Федерации;

2. ГОСТ 20886-85 Организация данных в системах обработки данных. Термины и определения.

3. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными.

4. ГОСТ 15971-90 Системы обработки информации. Термины и определения.

5. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.

6. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ (ред. от 10.01.2016).

7. ТОИ Р-45-084-01 Типовая инструкция по охране труда при работе на персональном компьютере.

8. ГОСТ 12.0.003-2015 Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация.

9. ГОСТ Р 2.2.2006-05 Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда.

10. ГОСТ 12.1.019-2017 Электробезопасность.

11. Технический регламент о требованиях пожарной безопасности от 22.07.2008 №123-ФЗ (ред. от 29.07.2017).

# 2 Термины, определения и сокращения

В настоящей выпускной квалификационной работе применяются термины с соответствующими определениями и сокращениями, установленные нормативными документами, указанными ниже:

– ГОСТ Р 1.12-2004. Стандартизация в Российской Федерации;

– ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными;

– ГОСТ 15971-90 Системы обработки информации. Термины и определения;

– ГОСТ 20886-85 Организация данных в системах обработки данных.

Термины, определения и сокращения:

**1 Язык программирования (ЯП)** – формальный язык, предназначенный для записи компьютерных программ.

**2 Электронно-вычислительная машина (ЭВМ**) – комплекс технических, аппаратных и программных средств, предназначенных для автоматической обработки информации, вычислений, автоматического управления. При этом основные функциональные элементы (логические, запоминающие, индикационные и др.) выполнены на электронных элементах.

**3 Персональная электронно-вычислительная машина (ПЭВМ)** – настольная микро-ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности.

**4 Персональный компьютер (ПК)** – распространенное в обиходе название, частный случай ПЭВМ.

**5 Карманный персональный компьютер (КПК)** – портативное вычислительное устройство, обладающее широкими функциональными возможностями.

**6 Смартфон** – мобильный телефон, дополненный функциональностью КПК.

**7 Планшетный компьютер, планшет** – разновидность устройств, имеющих функциональность КПК. Как правило, отличаются от смартфонов размерностью диагонали экрана (от 7 дюймов).

**8 HyperText Markup Language (HTML)** – гипертекстовый язык разметки, в основном использующийся для установки расположения элементов на веб-странице.

**9 Cascade Style Sheet (CSS)** – «каскадный таблицы стилей», язык, описывающий внешний вид веб-страниц.

**10 Uniform Resource Locator (URL)** – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса.

**11 Integrated Development Environment (IDE)** – система программных средств, используемая программистами, для разработки программного обеспечения.

**12 Фреймворк** – платформа, определяющая структуру программной системы, объединяет воедино различные компоненты программного проекта и упрощает процесс разработки.

**13 Тег, дескриптор** ­ основной элемент языков разметки.

**14 Тег, хештег** ­ ключевое слово для тематического группирования информации. Применяется в социальных сетях.

**15 Хедер (англ. header)** – верхняя часть веб-страницы. В большинстве случаев содержит в себе навигационную панель.

**16 Футер (англ. footer)** – нижняя часть веб-страницы. Зачастую содержит себе сведения об авторстве, также в футере нередко дублируется информация из хедера.

**17 Тогглер, переключатель (англ. toggler)** ­ компонент веб-страницы, служащий для сокрытия и раскрытия данных.

**18 Рунет** – российская часть сети Интернет.

# 3 Анализ предметной области

Библиотека, в общем понимании – это информационное, культурное и образовательное учреждение, располагающее организованным фондом документов (книг, газет и пр.) и представляющее их во временное пользование абонентам, а также осуществляющее другие библиотечные услуги.

Литературный клуб – это «добровольное творческое объединение» [1] авторов и читателей, в рамках которого ведется публикация новых и уже имеющихся произведений, а также организовывается дискуссия.

Деятельность библиотек по обслуживанию читателей осуществляется следующим образом: библиотечный абонемент предоставляет читателю право получить издание из библиотеки в своё полное распоряжение на определённый срок, в другом случае читатель имеет возможность знакомиться с книгой только в помещении библиотеки (как правило, в специально отведённом «читательском зале»).

В некоторых библиотеках работает только абонемент или только читательский зал, в других же, эти формы обслуживания сочетаются, хотя не для всех единиц хранения возможны обе, в силу редкости и/или ветхости издания.

Веб-дизайн — это процесс производства веб-сайтов, который включает техническую разработку, структурирование информации, визуальный (графический) дизайн и доставку по сети.[2]

Под frontend-разработкой понимается программирование веб-сайта на стороне клиента. Размечаются и подвергаются дизайну данные, которые используются в приложении, а также может описываться поведение (скрипт) данных на стороне непосредственно клиента. Для разработки клиентской части приложения программист использует набор необходимых ему аппаратно-программных средств.

# 4 Постановка задачи и техническое задание

Для понимания сути проектирования, а также для разработки программного продукта необходимо корректно сформулировать постановку задачи.

Постановка задачи содержит описание входной и выходной информации, а также формулировку технического задания.

Техническое задание содержит необходимые требования к программному продукту, а также выбор технологий и платформ для разработки веб-приложения и подробное описание объектов проектирования.

4.1 Общие требования

Клиентская часть предназначена для рядовых пользователей сайта. Исходя из вышесказанного, а также учитывая то, какие технологии на данный момент являются трендовыми, общие требования имеют следующий вид:

* Сайт должен обладать простым, понятным, и в то же время комфортным и адаптивным пользовательским интерфейсом;
* Сайт должен обладать необязательной для пользователей регистрацией, чтобы у любого посетителя был беспрепятственный доступ ко всей электронной библиотеке. Регистрация пользователей необходима в случае добавления, изменения и/или удаления данных о книге, а также для участия в дискуссиях на форуме;
* Сайт обязан иметь раздел «Управление сайтом», доступ к которому ограничен. Данный раздел содержит скрипты, позволяющие администраторам частично автоматизировать процесс контроля за сайтом;
* Как и многие веб-ресурсы, данный сайт должен иметь раздел «Часто задаваемые вопросы», в котором пользователи могут найти ответы на самые распространенные проблемы, которые могут возникнуть в процессе с ознакомлением функционала веб-ресурса.

4.2 Платформа

Для разработки клиентской части веб-сайта, потребуется следующий набор технологий:

* для написания основной разметки веб-страниц используется сочетание языка разметки HTML и объектно-ориентрованного языка программирования, имеющего специальные библиотеки для работы с веб-приложениями;
* для оформления дизайна веб-страниц используется язык CSS и набор стилей Twitter Bootstrap 4;
* для описания поведения отдельных элементов используется язык программирования: JavaScript.

Исходя из данных требований, веб-сайт следует разработать на платформе, поддерживающей данный набор технологий. Одной из таких платформ является платформа ASP.NET Core, разработанная корпорацией Microsoft. Одним из плюсов данной платформы является то, что при создании проекта, можно выбрать подходящий для целей шаблон. Также, автоматически с проектом, загружается и набор стилей Bootstrap.

# 5 Обзор используемых технологий и сред разработки

Для разработки каждого приложения следует определиться, какой стек (набор) технологий необходимо использовать, чтобы достичь конечного результата.

После того, как был проведен анализ технического задания, было решено использовать следующий набор технологий для разработки клиентской части веб-сервиса:

* Гипертекстовый язык разметки HTML5;
* Язык стилей CSS3;
* Язык программирования JavaScript;
* Язык программирования C#;
* Набор стилей Twitter Bootstrap 4;
* Среда разработки Microsoft Visual Studio 2019;
* Веб-фреймворк ASP.NET Core.

5.1 Язык HTML5

Гипертекстовый язык разметки HTML – это стандартизированный язык разметки документов в рамках Всемирной паутины (World Wide Web). Подавляющее большинство веб-страниц имеет разметку, созданную при помощи HTML как в чистом варианте, так и в паре с другими языками программирования.

Основа языка HTML – теги, которые бывают либо одинарными, такие как <img>, либо парными, такими как <span></span>, <div></div>, <p></p>, между которыми заключено какое-либо содержимое.

В настоящее время во всем мире применяется стандарт языка HTML5.

Каждая веб-страница имеет основной каркас, состоящий из нескольких тегов. В листинге 5.1 описывается каркас страницы согласно стандарту HTML5.

Листинг 5.1 – Каркас страницы согласно HTML5

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>HTML Document</title>

</head>

<body>

<!-- Содержимое страницы -->

</body>

</html>

5.2 Язык CSS3

CSS– это формальный язык описания внешнего вида компонентов документа, описанного при помощи языка разметки. В подавляющем большинстве случаев используется для описания стилей веб-страниц, размеченных при помощи языка HTML.

Также, учитывая тот факт, что для поиска информации в интернете используются планшеты и смартфоны, ширины экранов которых значительно меньше экранов мониторов, язык применяется также и для медиа-верстки.

Текущий на 2020 год стандарт языка – CSS3.

Синтаксис языка выглядит следующим образом: (листинг 5.2)

Листинг 5.2 – Задание стиля тегу в языке CSS3

p {

margin-bottom: 5px;

font-size: 14px;

}

5.3 Язык программирования JavaScript

Язык программирования JavaScript (JS) – мультипарадигменный язык программирования, поддерживающий такие парадигмы программирования как объектно-ориентированная, императивная и функциональная.

Наиболее известен как язык сценариев для придания интерактивности веб-страницам на стороне клиента (браузера).

Язык JS и веб-страница связаны между собой программным интерфейсом DOM (Document Object Model – объектная модель документа). Для того, чтобы обратиться к какому-либо элементу или группе элементов, используется такие методы, как:

* getElementsByTagName(string) – результат метода – массив элементов с тегом, указанным в скобках;
* getElementsByClassName(string) – результат метода – массив элементов с классом, указанным в скобках;
* getElementById(string) – результат метода – элемент с уникальным идентификатором, указанным в скобках. Если таких элементов в документе несколько – результатом поиска будет первый элемент из массива;
* querySelector(string) – результат метода – первый в документе элемент, содержащий селектор (класс, id или тег), указанный в скобках;
* querySelectorAll(string) – то же, что и querySelector, только в результате будет массив всех элементов, содержащих селектор.

5.4 Язык программирования C#

Язык программирования C# - объектно-ориентированный язык программирования, разработанный на рубеже XX-XXI веков программистами корпорации Microsoft в качестве языка разработки приложений для платформы Microsoft .NET Framework.

Язык испытал на себе влияния таких языков, как С++, Модула, Smalltalk и Java. Но, также с этим, язык исключает модели, зарекомендовавшие себя в языках-предшественниках, как проблемные. Так, в отличие от языка С++, язык С# не поддерживает множественное наследование классов, но при этом допускает множественную реализацию интерфейсов.

Синтаксис языка С# во многом схож с такими языками, как Java и C++, отчего его относят к языкам с Си-подобным синтаксисом.

Любая программа, написанная для .NET Framework, переводится компилятором в единый для платформы байт-код Common Intermediate Language (CIL), после чего уже байт-код исполняется такой виртуальной машиной, как Common Language Runtime (CLR).

Основные объектные классы, входящие в .NET Framework – это Windows Forms, Windows Presentation Foundation, ASP.NET, ADO.NET и другие.

Текущая версия языка C# на первую половину 2020 года – C# 8.0, вышедшая в сентябре 2019 года.

Текущая версия платформы .NET Framework по состоянию на первую половину 2020 года – 4.8, вышедшая 18 апреля 2019 года, одновременно со средой разработки Microsoft Visual Studio 2019.

5.5 Twitter Bootstrap 4

В связи с тем, что число людей, использующих смартфоны и планшеты, неуклонно растёт, на рынке являются востребованными инструменты, способствующие сделать веб-сайт удобным для восприятия под любые размеры экрана.

Одним из таких инструментов является библиотека стилей и скриптов Bootstrap, разработанная программистами известной социальной сети Twitter.

Документы данного фреймворка можно скачать с официального сайта, на котором также размещена необходимая документация на английском языке[3], есть также и русскоязычная версия[4]. Таким образом, данный инструментарий, равно как и ASP.NET является open-source проектом, код которого можно скачать прямо с официального сайта.

Для начала работы с компонентами Bootstrap, необходимо проделать операции, указанные в листинге 5.3.

Листинг 5.3 – Подключение компонентов Bootstrap

<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

…

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="scripts/bootstrap.min.js"></script>

Фреймворк, начиная с 2013 года, следует принципу mobile-first (сначала мобильные), то есть, приоритет, в первую очередь отдаётся именно мобильным устройствам.

Платформа разделяет ширины экранов на пять категорий. Кодовое название и значения приведены в таблице 5.1.

Таблица 5.1 – Распределение по ширинам экрана

|  |  |
| --- | --- |
| **Кодировка** | **Значение ширины экрана** |
| XS | До 575 пикселей включительно |
| SM | От 576 пикселей и выше |
| MD | От 768 пикселей и выше |
| LG | От 992 пикселей и выше |
| XL | От 1200 пикселей и выше |

Из всех кодировок не применяется к стилям только XS, поскольку она стоит выше всех в приоритете.

Особенность фреймворка состоит в том, чтобы разработчик писал как можно меньше кода в CSS, заменяя его собственными классами.

Разберём один класс из библиотеки Bootstrap, устанавливающий размеры колонок.

Bootstrap позволяет разделить блок (ряд, колонку и т.п.) на 12 равных частей. Шаблон класса для установления размеров колонки выглядит следующим образом:

<div class=”col-{кодировка}-{число}”></div>

Таким образом, если задать класс col-md-8, начиная с ширины браузера в 768 пикселей, образуется колонка, равная 8/12 частям блока-родителя.

Такой класс представляет собой целый CSS-запрос, представленный на листинге 5.4.

Листинг 5.4 – Медиа-запрос

@media (min-width: 768px) {

.col-md-8 {

float: left;

width: 66.666667%;

}

}

В случае отсутствия кодировки в классе, стиль распространяется на все ширины экранов, если нет иных кодировок, служащими точками перехода.

Для некоторых компонентов, таких как «карусель», необходимо также работать с языком JS. Так, на листинге 5.5 указан скрипт, задающий частоту смены кадров в «карусели».

Листинг 5.5 – Скрипт для «карусели»

$('.my-carousel').carousel({

interval: 4000

});

5.6 Среда разработки Microsoft Visual Studio 2019

Среда разработки Microsoft Visual Studio 2019 – IDE от корпорации Microsoft, включающий в себя инструментарий для разработки приложений под объектные классы платформы .NET Framework.

IDE имеет три версии выпуска – базовая (Community), профессиональная (Professional) и расширенная (Enterprise).

Для разработки клиентской части, равно как и всего веб-приложения, использовалась свободно распространяемая версия для независимых разработчиков, а также для некорпоративных организаций – Community.

Корпоративными называются такие организации, в наличии у которых более 250 персональных компьютеров или чей годовой доход превышает один миллион долларов США.

Данная IDE предоставляет удобную навигацию по коду, а также различные виды рефакторинга.

MS VS, равно как и многие другие среды, выполняет инспекцию и предлагает автоматическое дополнение во время написания кода, которая основывается на информации, получаемой во время исполнения кода.

5.7 Веб-фреймворк ASP.NET Core

ASP.NET Core – платформа для создания различных веб-приложений, разработанная корпорацией Microsoft. Данный фреймворк был выбран в качестве основного средства разработки за счет следующих преимуществ:

1. Быстрота и удобство разработки. ASP.NET Core содержит различные стартовые шаблоны проектирования. При создании нового проекта, фреймворк автоматически определяет конфигурацию приложения, его взаимодействие с сервером и так далее. Благодаря этому, разработчик уже при первой компиляции проекта сможет запустить приложение на сервере и начать внедрение собственных идей и наработок, экономя свои силы и время.

2. Кроссплатформенность. Фреймворк может быть использован на всех известных ныне операционных системах: Windows, Linux или MacOS. Все веб-приложения, созданные в этой среде, могут быть запущены на любой из вышеперечисленных ОС.

3. Расширяемость. Так как ASP.NET Core состоит из различных независимых компонентов и библиотек, мы с легкостью можем расширить их реализацию с помощью внедрения зависимостей и механизма. Также можно использовать собственные компоненты с последующем расширением функциональности.

4. Открытый исходный код. На данный момент, фреймворк является opensource-проектом, а все его коды и файлы размещены на репозитории GitHub [5].

Основным языком программирования серверной части на ASP.NET Core является C#, для программирования же клиентской части используется следующий набор инструментов:

1. Для разметки страниц используется сочетание стандартного HTML с языком программирования C#, реализованного при помощи движка Razor.

2. Для придания страницам стилей используется стандартный язык CSS.

3. Для описания поведения компонентов страниц используется язык программирования JavaScript, в частности, используется набор функций jQuery, позволяющий обращаться к атрибутам и содержимому DOM (Document Object Model, объектная модель документа).

4. Для экономии времени на задание стилей, а также с целью сделать страницы адаптивными под любые ширины экранов, платформа ASP.NET предлагает использование компонентов Bootstrap, разработанных компанией Twitter.

Основной IDE всего веб-приложения является Microsoft Visual Studio.

Данные факты являются основными в пользу выбора данной технологии для разработки клиентской части веб-приложения.

Отдельно пойдёт речь про такой паттерн (шаблон), как MVC.

Данный паттерн предназначен для упрощения процесса создания приложения путём разделения графического интерфейса, бизнес-логики и хранимых данных на отдельные, связанные между собой составляющие. Приложение условно разделяется на три компонента: модель (M - Model), представление (V - View) и контроллер (C - Controller). Рисунок 5.1, представленный источником [6], и описывающий схему разделения трёх частей проекта, выглядит следующим образом:

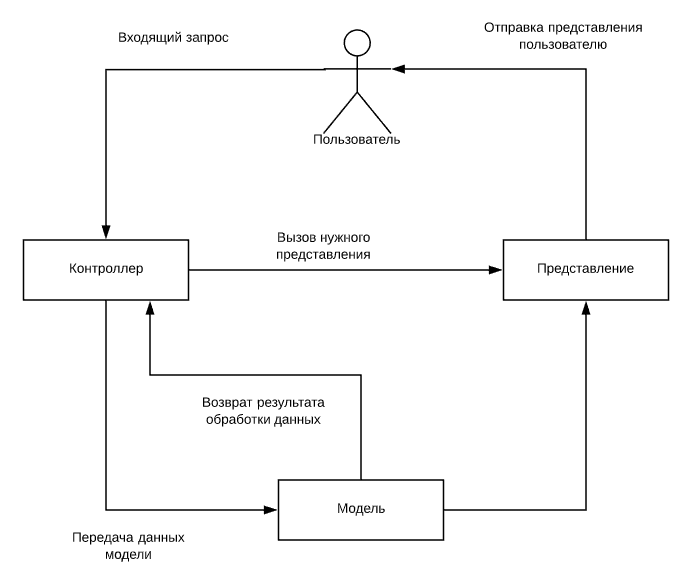


Рисунок 5.1 – Визуальная модель паттерна MVC

Модель (Model) содержит описание данных приложения на сервере, а также логику управления с ними.

В данном проекте также используется модель представления данных (VM - View-Model), которая используется представлениями для отображения данных, переданных из сервера.

Представление (View) описывает графический пользовательский интерфейс приложения.

В случае разработки веб-сайта обычно используются HTML-страницы в связке с языком стилей CSS. Также представления содержат логику отображения данных и их передачи на сервер. Обычно для этого используется язык JavaScript или PHP, однако в ASP.NET Core доступна технология Razor Pages, которая позволяет внедрять C# код прямо на пользовательской стороне. Для этого используется одноименный движок, благодаря которому, при помощи специального символа @ можно внедрять необходимые операции.

Контроллер (Controller) же является связующим звеном между представлением и моделью. Он содержит логику обработки пользовательских запросов, передачи их на модель. Затем контроллер отправляет пользователю представление с определенным выводом, в зависимости от результатов обработки данных моделью.

Благодаря вышеперечисленному разделению достигается эффект динамичности веб-страниц. Вместо того, чтобы под каждую страницу создавать отдельный документ, он создается под целую группу, в виде шаблона.

# 6 Проектирование и дизайн веб-страниц

Проектирование веб-страниц, особенно в том случае, если разрабатывается целый портал – труд не из лёгких. В связи с трендом на динамичные сайты, фреймворк ASP.NET упростил задачу frontend-разработчикам, создав так называемую «мастер-страницу», а также шаблоны представлений для веб-страниц.

6.1 Мастер-страница

«Мастер-страница» - это частичное представление веб-страницы, зачастую содержащее в себе данные тега <head></head>, общие скрипты для всех страниц, а также часть тега <body></body> (навигационная панель – «хедер» и подвал сайта – «футер»). Остальная часть разметки и кода располагается в отдельных шаблонах (представлениях), и связывается двумя компонентами:

* Со стороны мастер-страницы присутствуют методы @RenderBody() и @RenderSection("Scripts", required: false), которые принимают на обработку представление и скрипты, содержащиеся в нем;
* Со стороны представления присутствует компонент @Section Scripts, который будет обработан соответствующим методом.

В разрабатываемом проекте в состав «мастер-страницы» входит набор, указанный выше, с небольшими дополнениями и особенностями.

В теге, указывающем на «тело» документа, поставлен обработчик события onload="getCountReports()", указывающий на то, что при загрузке тела страницы запускается скрипт, который подсчитывает количество непрочитанных администраторами жалоб. Результатом скрипта, описанного в листинге 6.1, являются значки, уведомляющие исключительно администраторов о том, что поступило некоторое количество жалоб на комментарии. Сам же исход представлен на рисунке 6.1.

Листинг 6.1 – Скрипт подсчёта новых жалоб

function getCountReports() {

$.getJSON('/Report/GetCountReports', function(data) {

var count = data.toString();

var list = document.getElementsByClassName('notes');

if (count == 0)

for (var i = 0; i < list.length; i++)

list[i].setAttribute('style', 'display: none;');

for (var i = 0; i < list.length; i++)

list[i].innerHTML = count;

});

}

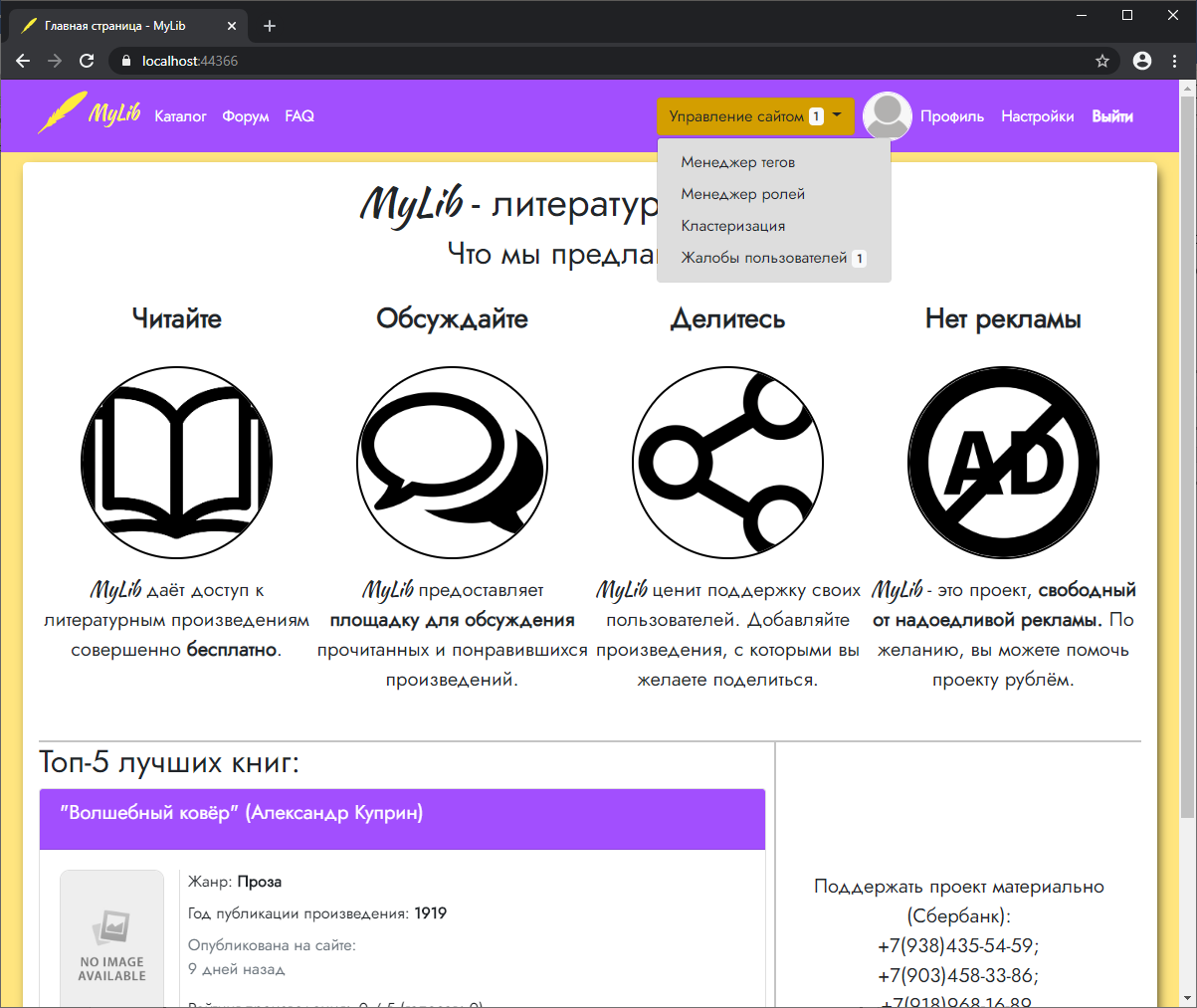


Рисунок 6.1 – Уведомление о непрочитанных жалобах

Как можно видеть, навигационная панель состоит из следующих компонентов:

* логотипа сайта и его названия (иконка пера и надпись «MyLib»);
* ссылки на разделы «Каталог», «Форум» и «FAQ»;
* доступная лишь администраторам панель «Управление сайтом», содержащая в себе менеджер тегов, менеджер ролей, раздел кластеризации и раздел «Жалобы пользователей»;
* аватарка пользователя и ссылка на личную страницу пользователя;
* ссылка на раздел «Настройки профиля»;
* кнопка, осуществляющая выход из профиля.

Для незарегистрированных пользователей навигационная панель имеет несколько другой вид. Изменения касаются правой части, а именно наличие всего двух кнопок: регистрации новой учетной записи и входа в уже существующую.

Дизайн мастер-страницы проектировался следующим образом:

1. Поскольку тематика сайта связана с литературой, и одними из ассоциаций являются книги, перья и пергамент, было принято решение установить на фон страницы цвет, приближенный к пергаментному (в кодировке HEX – #FFE57F);
2. Одним из сочетаемых с пергаментным цветом, является светло-фиолетовый, который присутствует на навигационной панели, а также в оглавлении книжных «карточек». Кодировка цвета в формате HEX – #A24FFE;
3. В качестве используемых для всех страниц шрифтов применяются Kaushan Script (только для логотипа MyLib) и Jost – весь текст. Шрифт Jost был выбран, поскольку ассоциируется с одним из книжных шрифтов, применявшихся в СССР (рисунок 6.2).

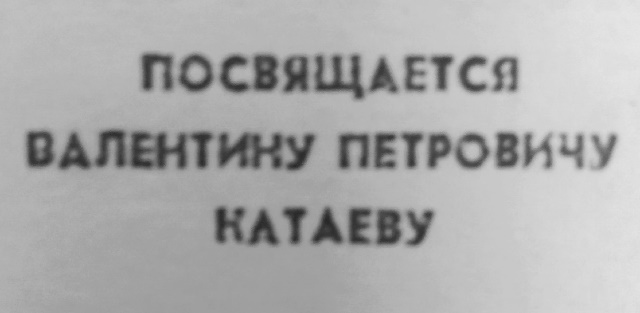


Рисунок 6.2 – Предисловие в произведении «12 стульев»

6.2 Домашняя страница

Домашняя страница, как и все последующие далее, содержит в своей основе контейнер, в котором располагается вся остальная разметка.

Главная страница состоит из трёх сегментов: блока описания основных преимуществ сайта (features), блока рекламы и «карусели», внутри которой находится до пяти «карточек» книг, которые имеют наилучший рейтинг среди пользователь.

Карточка состоит из нескольких частей, а именно:

1. Верхняя часть, содержащая в себе название и автора произведения;
2. Центральная часть, в которой расположены: фотография обложки, жанр произведения, год первой публикации, время публикации на сайте, краткое описание произведения и рейтинг на сайте.

Время публикации отображается в формате «некоторое время назад», активно применяющееся во многих социальных сетях. Подобный формат применяется для даты публикации книг, тем на форуме и комментариев (Приложение А).

Время смены слайдов в «карусели» устанавливается при помощи скрипта (листинг 5.5).

6.3 Каталог

Главная страница раздела «Каталог», помимо указанного ранее контейнера, содержит в себе поисковую строку, путеводители по страницам каталога в начале и конце веб-страницы, а также до трёх книжных «карточек» на каждой странице каталога. В случае, если пользователь авторизован, он имеет право загрузить новую книгу. Общий вид веб-страницы представлен на рисунке 6.3.

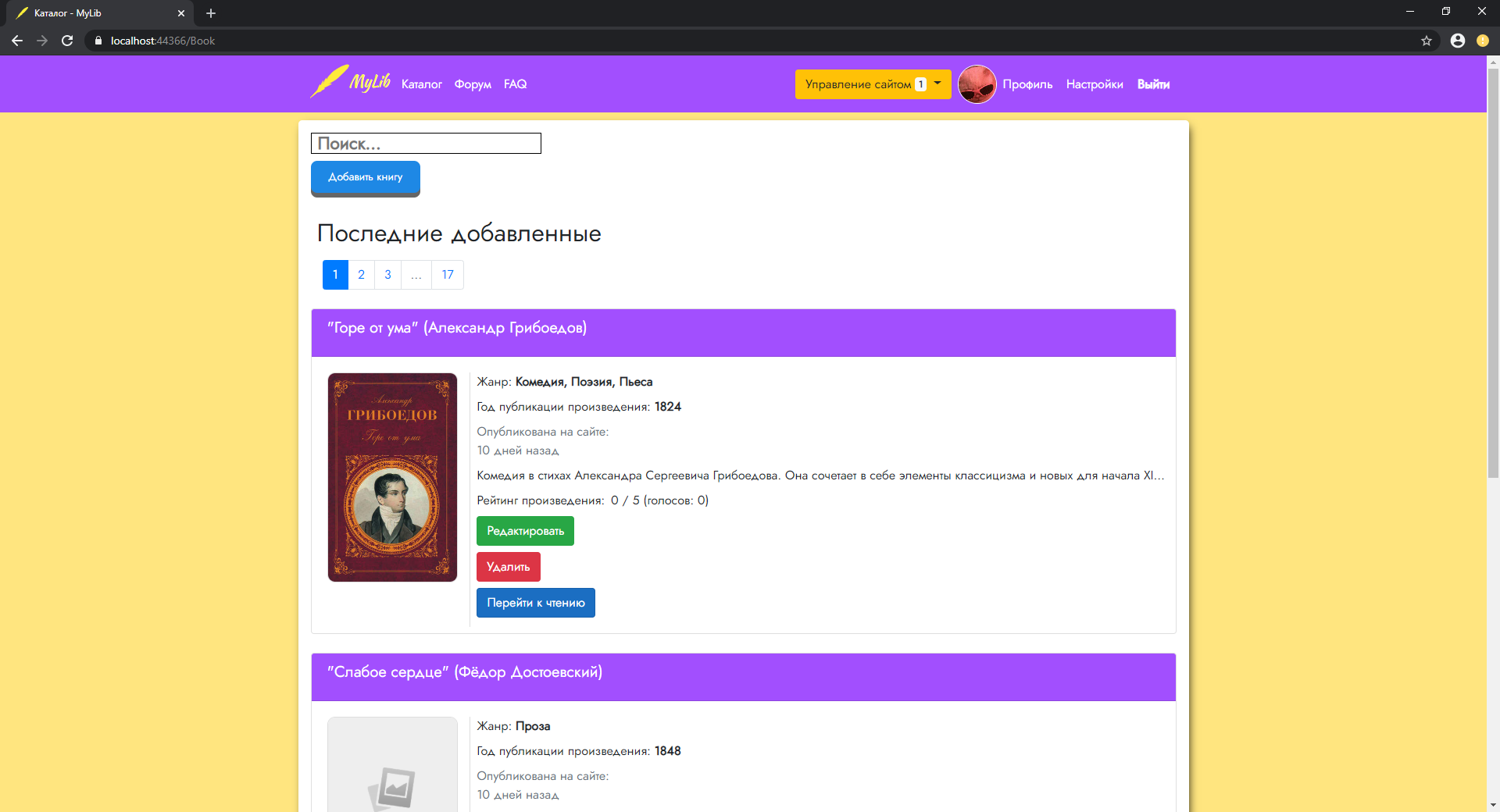


Рисунок 6.3 – Первая страница каталога

Для того, чтобы добавить книгу, пользователь переходит по соответствующей кнопке, в результате чего загружается страница, указанная на рисунке 6.4.

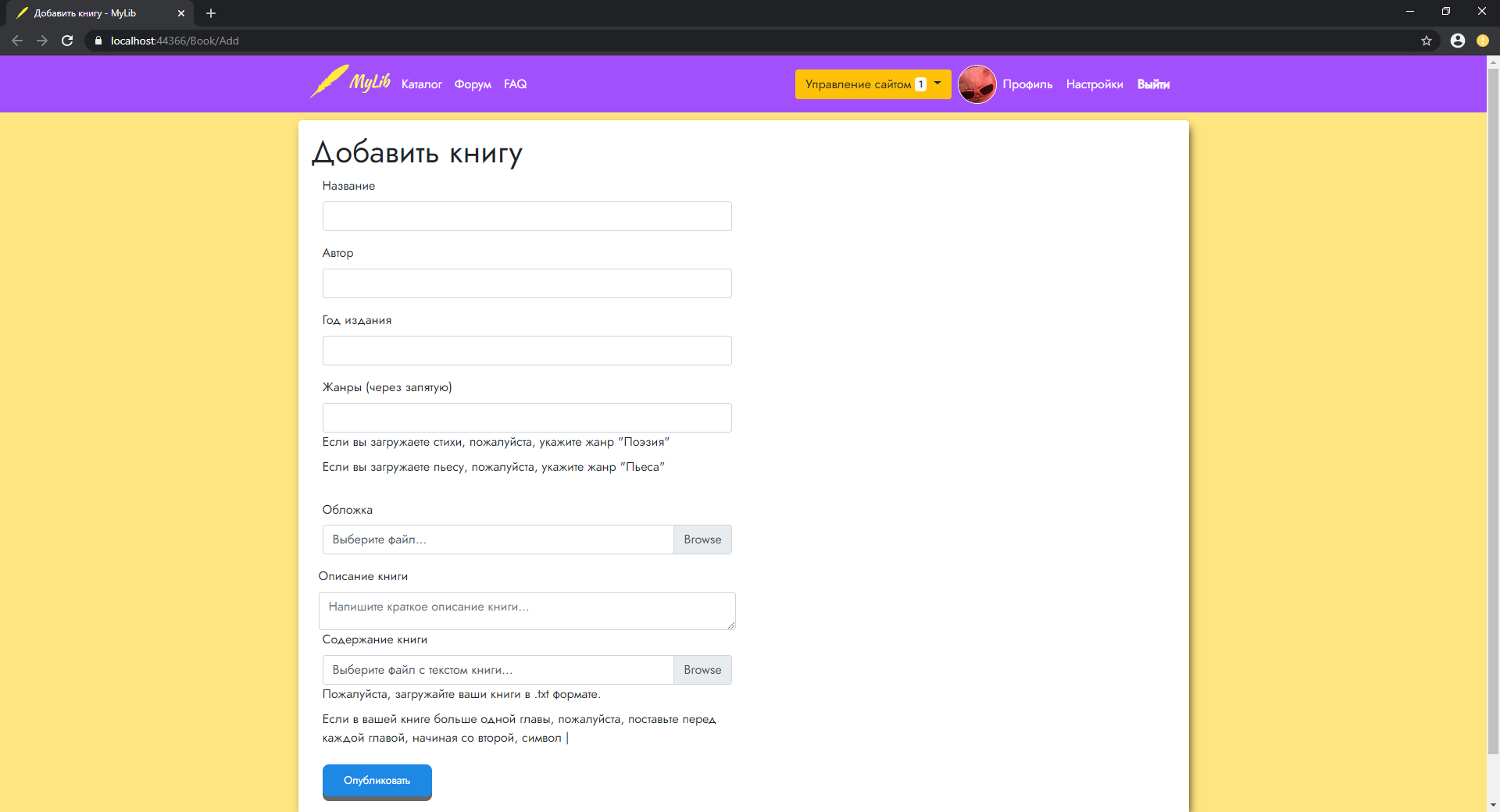


Рисунок 6.4 – Форма добавления книги

Перед тем, как добавить книгу, пользователь обязан заполнить некоторые данные, а именно:

* название книги;
* автор книги (примеры: Александр Пушкин, Лев Толстой);
* год издания или первой публикации книги;
* список жанров, указанных через запятую;
* обложка (необязательно к заполнению);
* краткое описание (необязательно к заполнению);
* содержание книги в формате .txt.

Также, в форме указаны несколько примечаний, остановимся подробно на каждом из них.

Постраничное разбиение текста происходит следующим образом:

1. Текст из файла считывается контроллером и преобразовывается в строку;
2. Переход на новую строку состоит из двух символов – перехода каретки влево (‘\r’) и собственно перехода на новую строку (‘\n’);
3. При помощи метода Split текст из одной строки разбивается на массив строк. В качестве символа-разделителя выступает символ перехода на новую строку;
4. Если «пустых» строк между текстом больше одной, то лишние строки ликвидируются.

Полученный массив строк подвергается постраничному выводу. Количество строк определяется следующей переменной, указанной в листинге 6.2.

Листинг 6.2 – Постраничный вывод книги

var book = \_bookService.GetById(id);

var genres = book.Genre.ToLower().Split(new char[] { ',', ' ' }).ToList();

int pageSize = genres.Contains("поэзия") || genres.Contains("пьеса") ? 50 : 25;

Таким образом, количество абзацев в «книжной» странице имеет вариацию: если в жанрах присутствует поэзия или пьеса – на одну страницу выводится 50, в противном случае – 25.

В случае поэзии, необходимо ещё и расположить текст по центру, для чего уже в представлении ставится следующее условие и соответствующий вывод на экран (листинг 6.3):

Листинг 6.3 – Расположение текста поэтических произведений по центру

@if (Model.Genre.ToLower().Split(new char[] { ',', ' ' }).ToList().Contains("поэзия"))

{

@foreach (var str in Model.Sentences)

{

if (str == "\r")

{

<p class="mb-2 text-center">&nbsp;</p>

}

else

{

<p class="mb-2 text-center">@str</p>

}

}

}

else

{

@foreach (var str in Model.Sentences)

{

if (str == "\r")

{

<p class="mb-2">&nbsp;</p>

}

else

{

<p class="mb-2">@str</p>

}

}

}

Следует отметить несколько ключевых моментов:

1. Символ перехода каретки влево не отображается в HTML-формате, следовательно имеется пустая строка;
2. Абзац, в котором отсутствуют какие-либо читаемые языком HTML символы, становится невидимым;
3. Чтобы осуществить видимость, используется символ &nbsp; (no-brake space, неразрывный пробел);
4. В случае, если среди жанров книги присутствует поэзия, абзацу присваивается класс text-center из библиотеки Bootstrap.

Что же касается следующего примечания, здесь принцип практически тот же – в тексте, который пользователь собрался публиковать в качестве книги, ему необходимо расставить определенное количество символов-распределителей (если книга состоит более чем из одной главы) или же не расставлять вовсе. Принцип разделения в файле представлен на рисунке 6.5.



Рисунок 6.5 – Принцип разделения теста на главы

После того, как книга загрузилась, она разбивается на главы при помощи того же метода Split. Далее, для каждой главы повторяется процесс, описанный выше.

Проекцией текста книги занимаются такие шаблоны страниц, как Detail и Read.

В случае, если книга состоит из одной главы, Detail представляет из себя страницу, на которой расположен текст книги. В противном случае – список ссылок на главы, проецируемые шаблоном Read. Результаты разбиения продемонстрированы на рисунках 6.7-6.9.

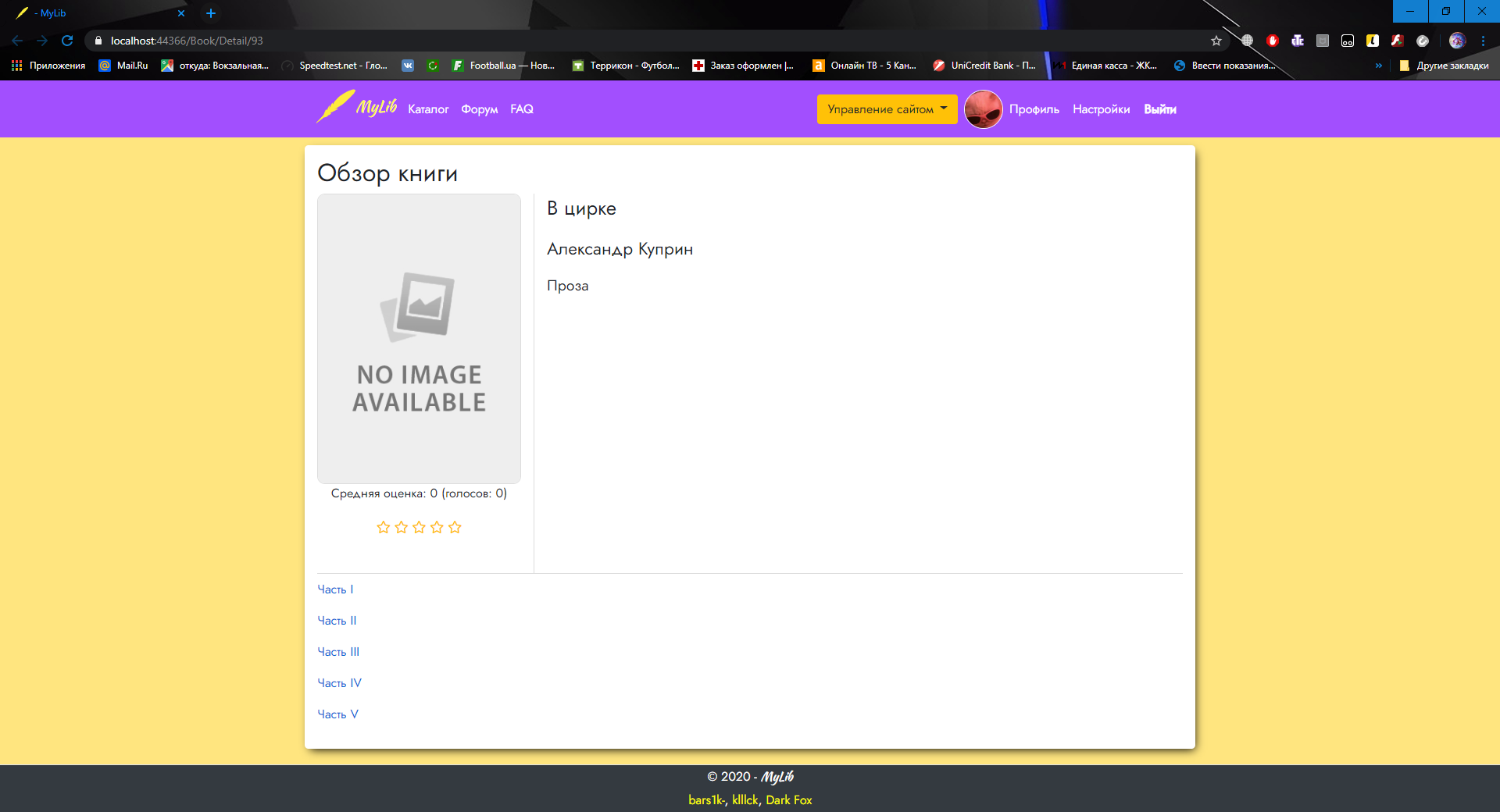


Рисунок 6.7 – Detail при наличии в книге более, чем одной главы

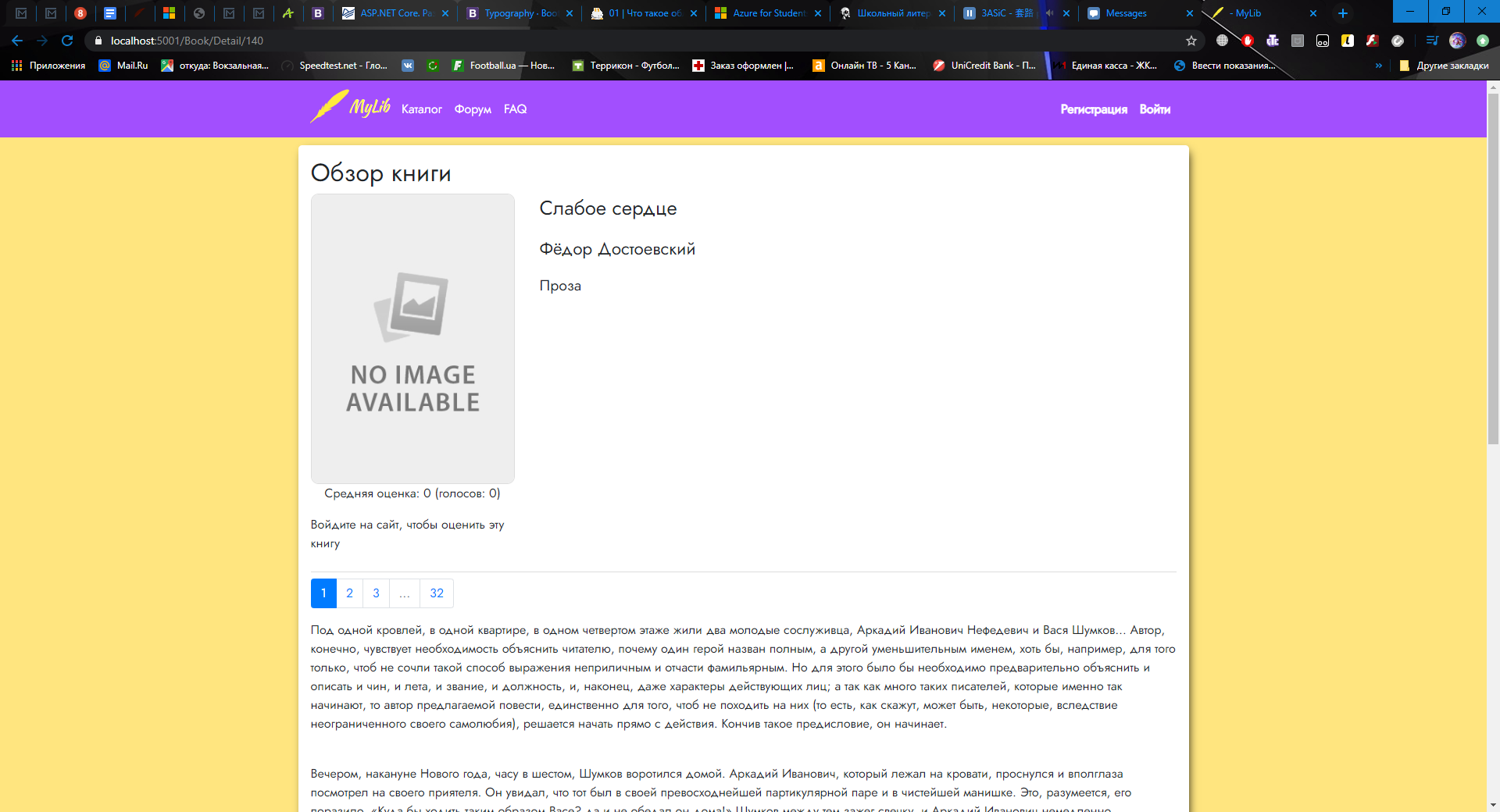


Рисунок 6.8 – Detail при наличии одной главы (прозаический текст)

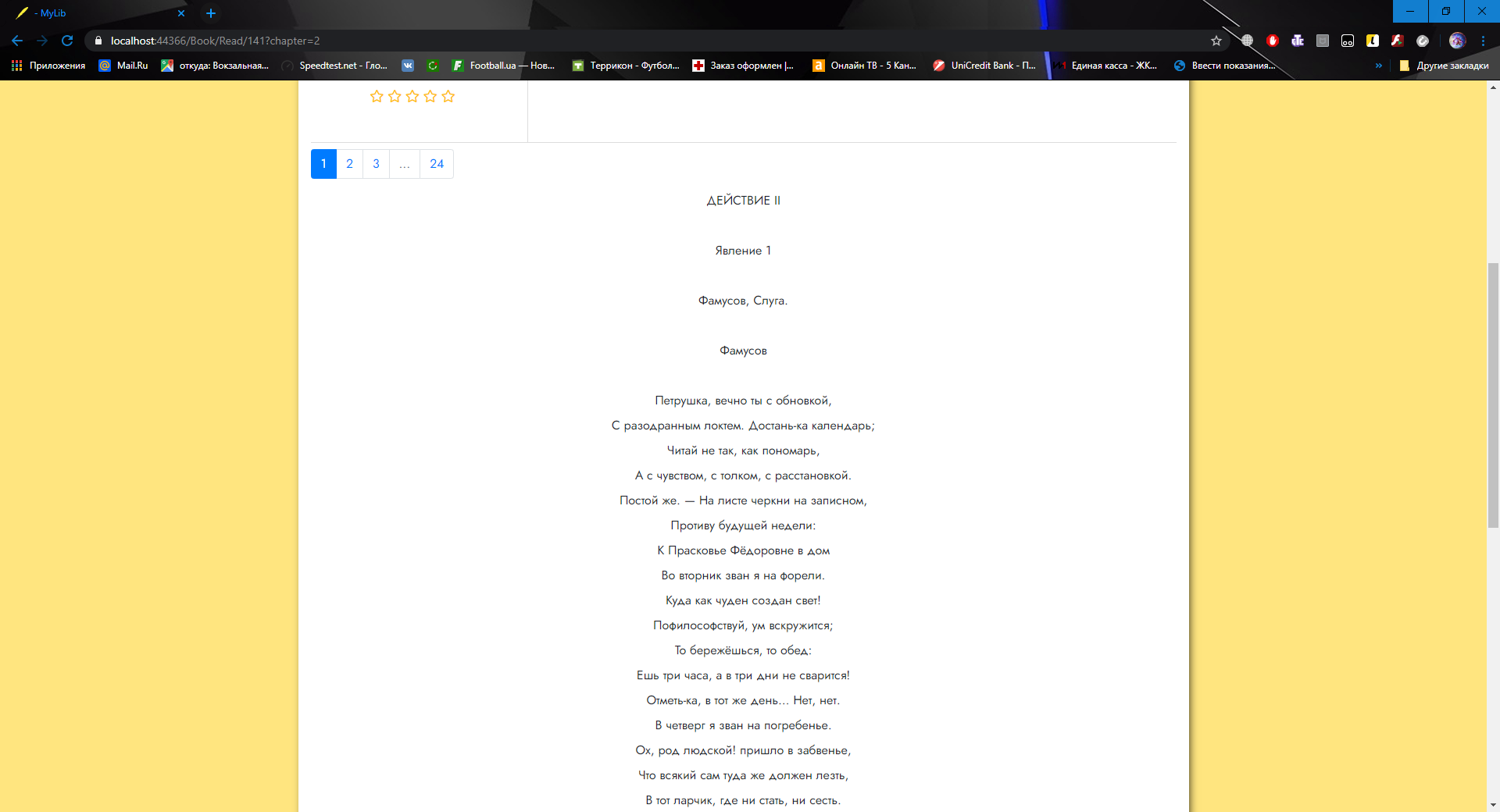


Рисунок 6.9 – Read при поэтическом тексте

Также, следует заметить, что в Detail и Read присутствует краткое описание книги, а также блок для оценки книги. Для того, чтобы оценить книгу, необходимо выбрать необходимое количество звезд, в соответствии с оценкой. После выбора, пользователю становится доступной кнопка «Проголосовать». Если пользователь еще не оценивал книгу, то контроллер обработает данные об оценке, которые отправил пользователь. Если же пользователь уже оценивал данную книгу, тогда кнопка будет из себя представлять всего лишь имитацию, ведь при нажатии на нее срабатывает функция языка JS alert с сообщением о том, что пользователь уже голосовал за данную книгу.

6.4 Форум

При переходе на раздел «Форум», взору пользователя предстает страница вида, указанного на рисунке 6.10.

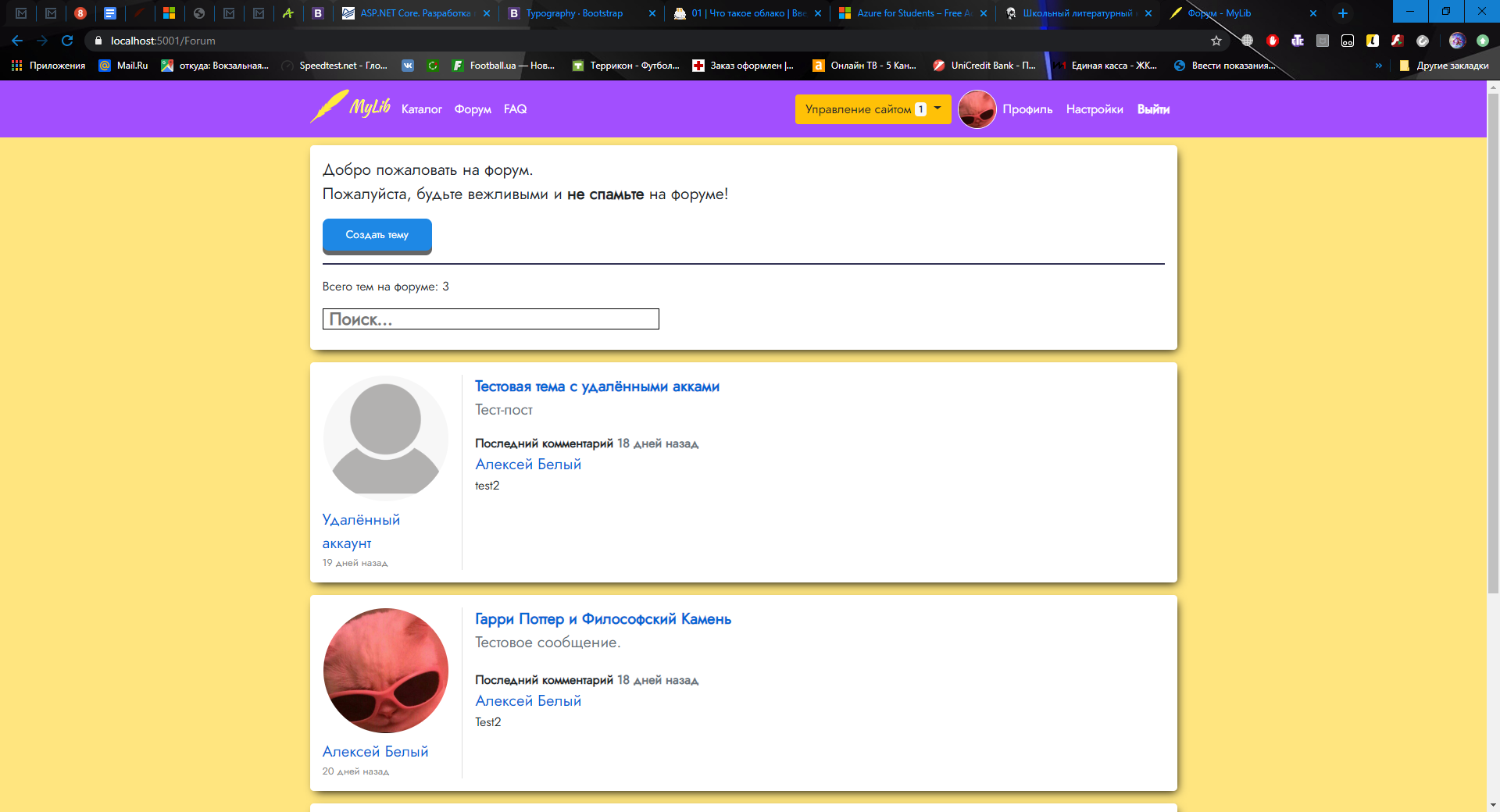


Рисунок 6.10 – Главная страница форума

Как можно видеть из рисунка, помимо уже знакомых кнопки добавления и поисковой строки, на странице присутствуют блоки с темами форума.

В левой части располагается фото профиля пользователя, создавшего тему, ссылка на его личную страницу и время, когда была создана тема.

В правой части присутствуют:

* ссылка на тему форума, представленная в виде названия темы;
* сообщение (пост), оставляемое создателем темы;
* блок, показывающий последний комментарий в теме.

Часть «Последний комментарий» состоит из даты, когда был оставлен данный комментарий, ссылки на профиль комментатора и содержания комментария.

Пользователи вправе редактировать содержимое темы (название, сообщение), а также удалять её. При наведении курсора мыши на блок с темой, в нижней трети правой части появятся соответствующие кнопки (рисунок 6.11).

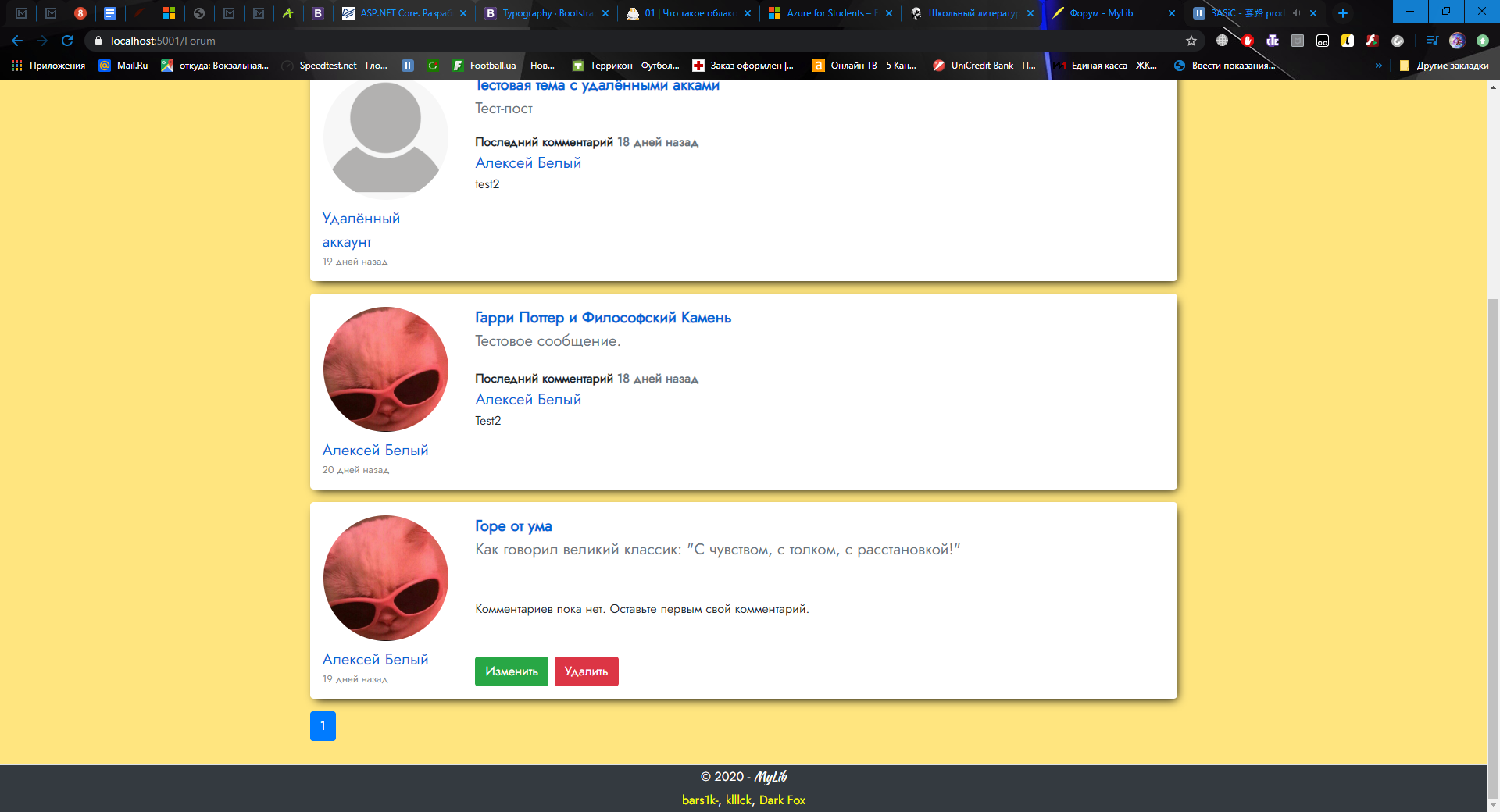


Рисунок 6.11 – Тема форума в двух состояниях

При попытке удалить тему форума, внутри браузера появляется форма, вызываемая командой языка JS confirm, предлагающая либо подтвердить действие, либо его отменить.

Перед тем, как описать страницу для изменения темы, следует рассмотреть форму создания темы.

Внешний вид представлен на рисунке 6.12.

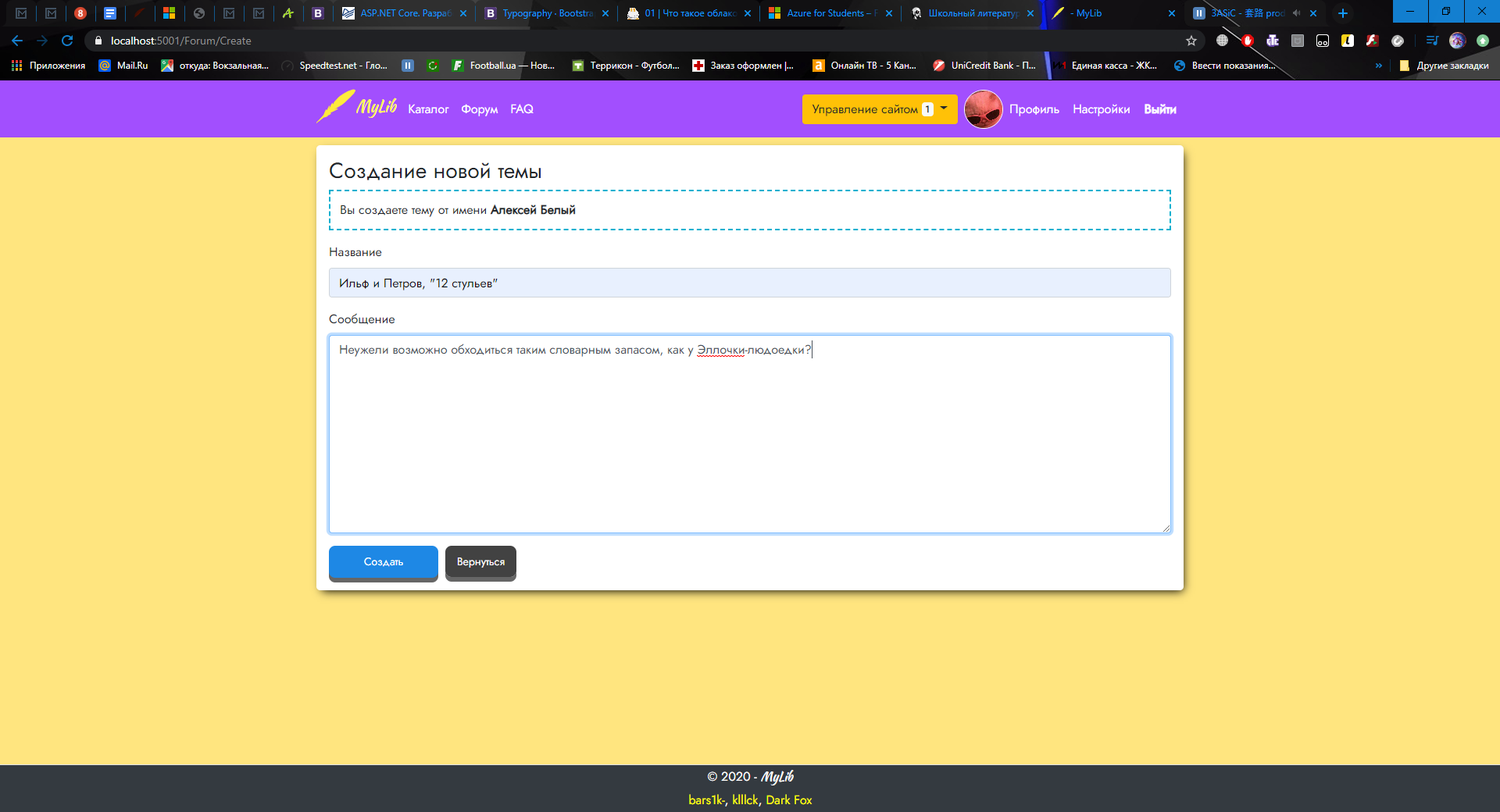


Рисунок 6.12 – Процесс создания темы форума

После создания темы, контроллер перенаправляет пользователя на главную страницу форума. При этом новая тема становится первой в списке, что и демонстрируется на рисунке 6.13.

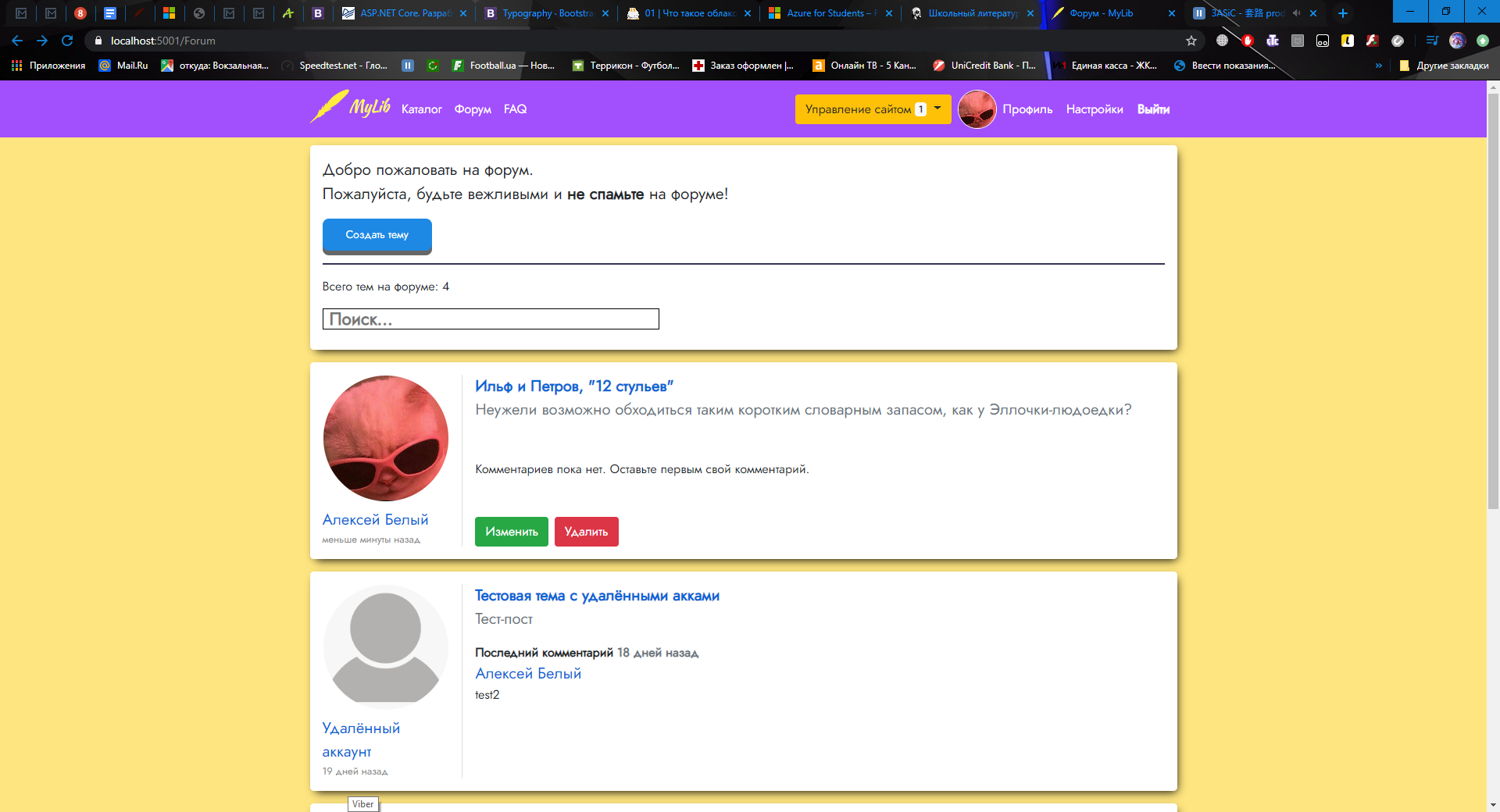


Рисунок 6.13 – Главная страница форума после создания новой темы

Каждая тема имеет в базе данных два значения, связанных со временем – дата создания и дата последнего обновления. В качестве последнего аргумента, при создании темы, принимается то же самое время, что и для даты создания. Однако, в отличие от даты создания, дата последнего обновления постоянно изменяется, с каждым новым комментарием.

Список тем сортируется по дате последнего обновления, причем чем ближе дата к настоящему времени, тем выше тема в списке.

На одной странице располагается до 10 тем, переключаться между страницами можно при помощи того же переключателя внизу страницы, который был рассмотрен в каталоге.

Теперь рассмотрим кнопку «Изменить содержимое». Содержимое страницы практически ничем не отличается от страницы создания темы, разве что отсутствует напоминание, от чьего имени создается тема.

Страница для комментариев создана в похожем стиле, что и форум. Внешний вид представлен на рисунке 6.14.

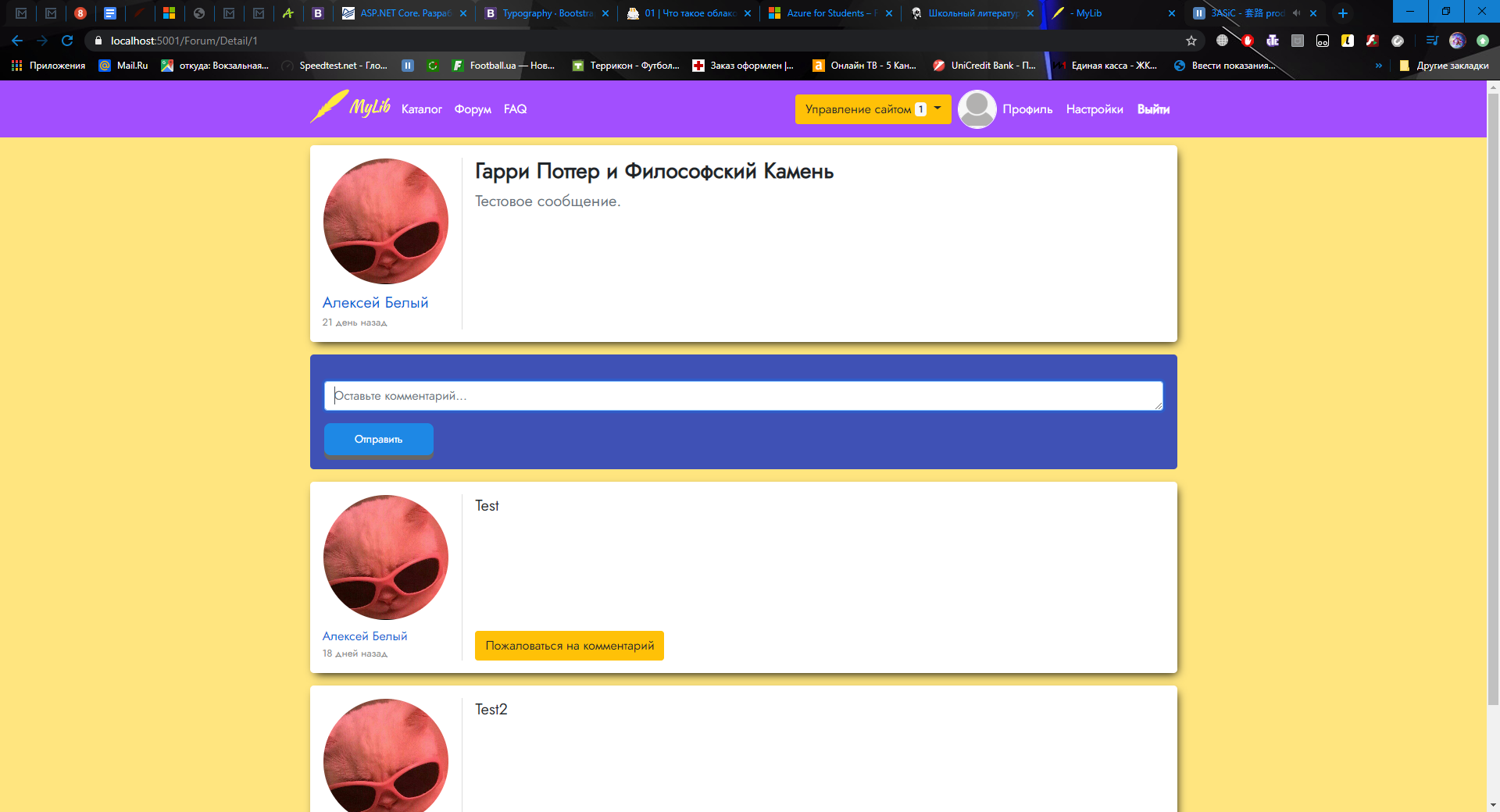


Рисунок 6.14 – Страница темы форума с комментариями

Из рисунка выше, можно разделить страницу примерно на три части: главного сообщения, блока для написания комментария и самих комментариев.

Левые части главного сообщения и комментариев оформлены в стиле, аналогичном тому, что были у тем на главной странице форума. Главное изменение – наличие кнопки «Пожаловаться на комментарий», при нажатии на которую пользователю предстает модальное окно с небольшой анкетой, которую следует заполнить (рисунок 6.15).

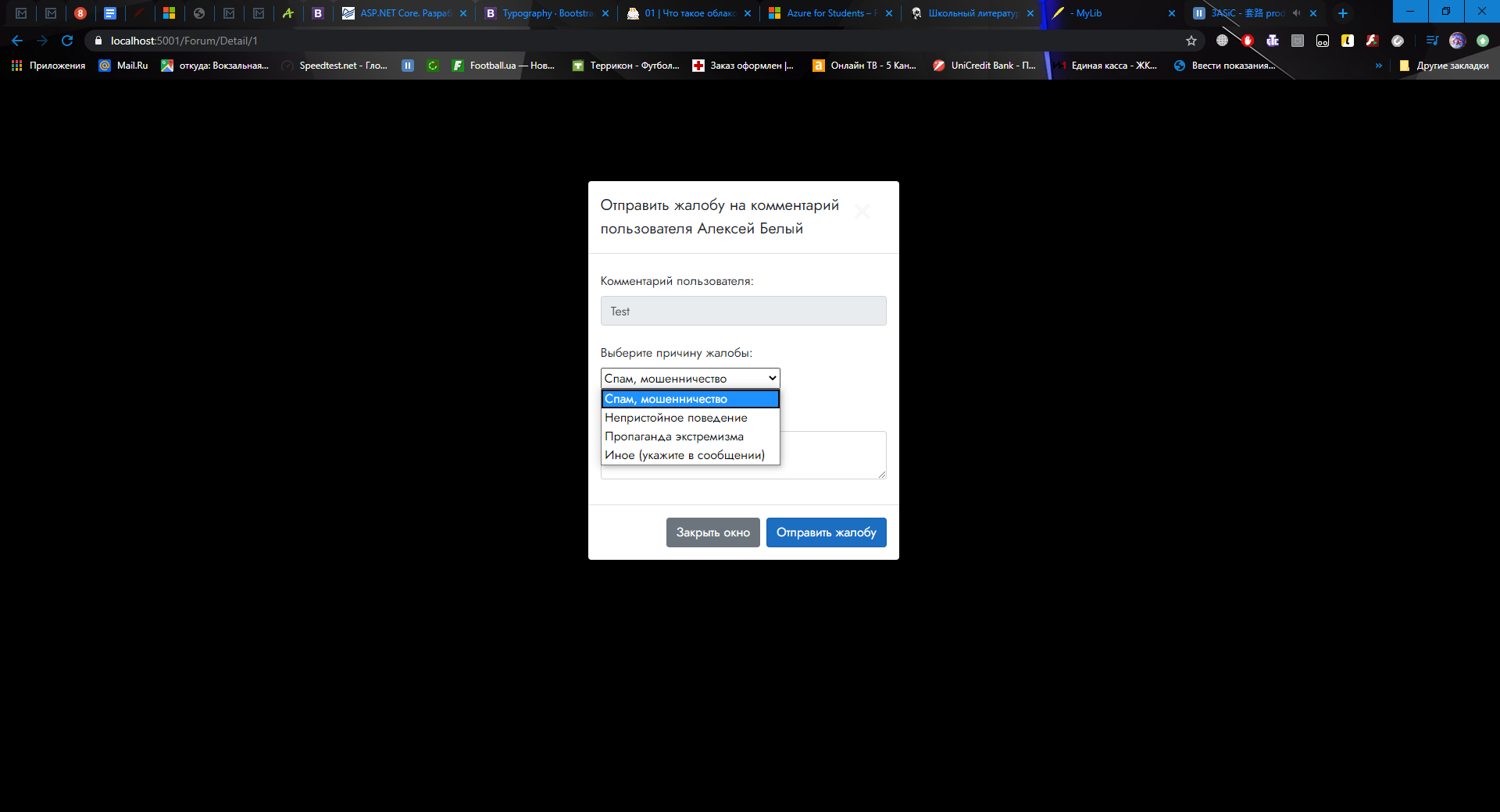


Рисунок 6.15 – Модальное окно с формой отправки жалобы

После того, как пользователь отправил жалобу, жалоба поступает в список непрочитанных, благодаря наличию в базе данных контрольного атрибута, который при создании жалобы равен логическому нулю.

После отправки жалобы, количество непрочитанных жалоб увеличивается на одну, и соответствующее уведомление изменит значение на единицу.

Оставим комментарий под одной из тем. После набора необходимого текста и нажатия на кнопку отправить, можно видеть, что новый комментарий находится внизу страницы (рисунок 6.16).

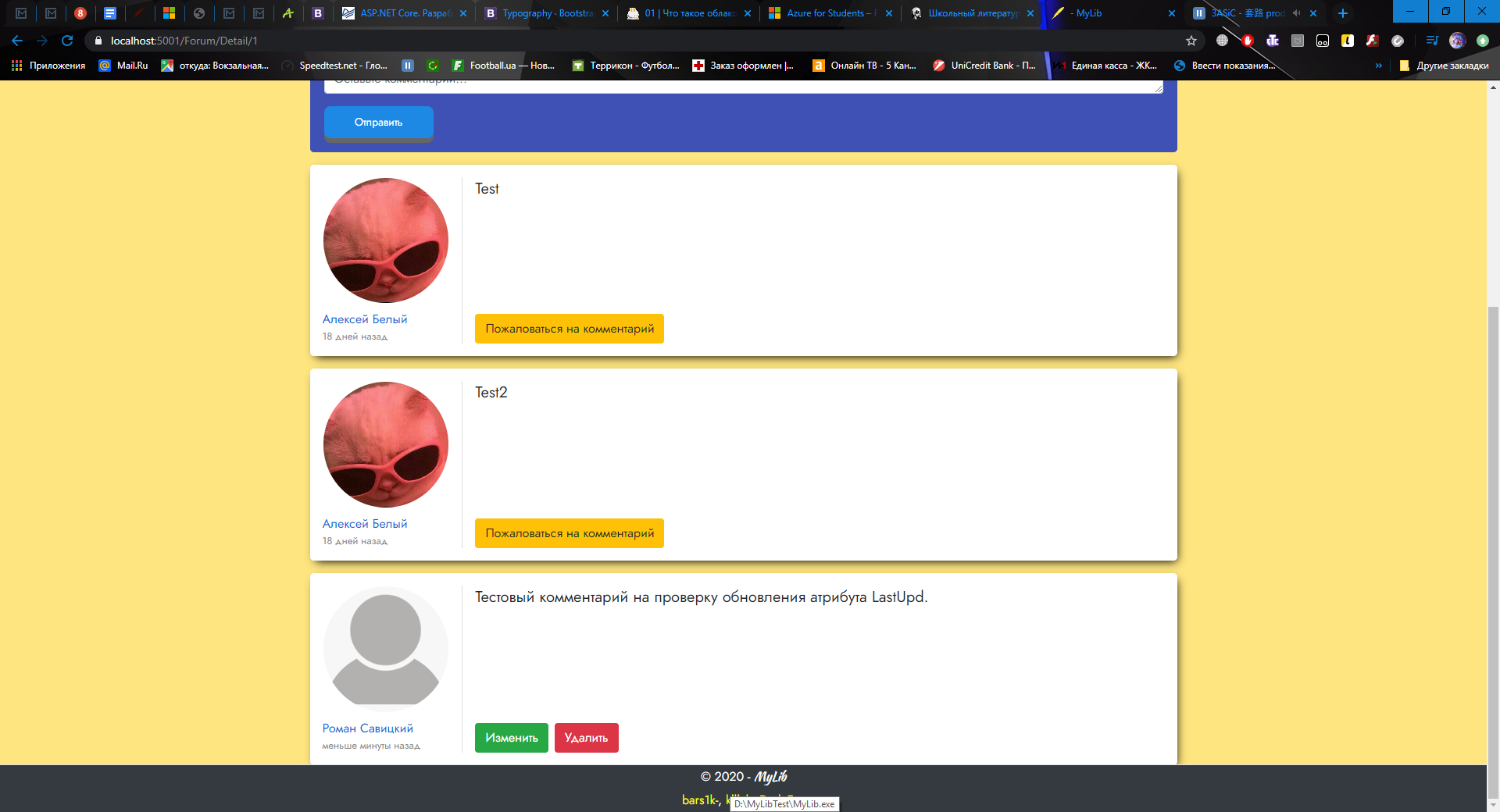


Рисунок 6.16 – Расположение новых комментариев под темой

По рисунку выше можно видеть, что под своими (для пользователя) комментариями, расположены две кнопки – изменить и удалить. Разберем функциональность лишь первой кнопки.

При нажатии на кнопку «Изменить», блок с комментарием будет иметь вид, представленный на рисунке 6.17.

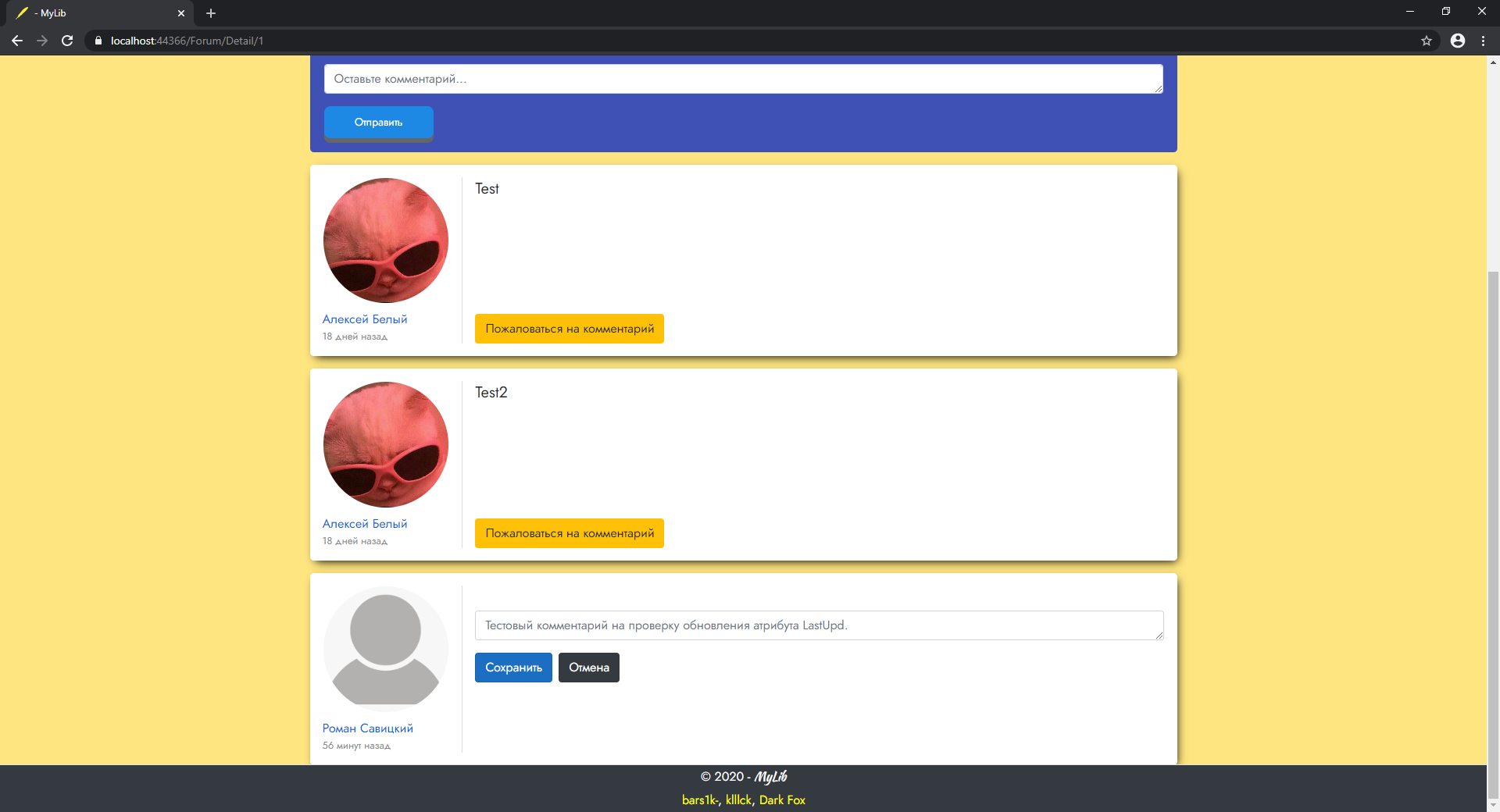


Рисунок 6.17 – Комментарий в состоянии редактирования

Поскольку ранее упоминалось, что атрибут даты последнего обновления изменяется с публикацией каждого комментария, проверим, изменился ли порядок тем на главной странице. Комментарий ради этого специального добавлялся в уже существовавшую тему.

Рисунок 6.18 подтверждает, что изменения действительно произошли.

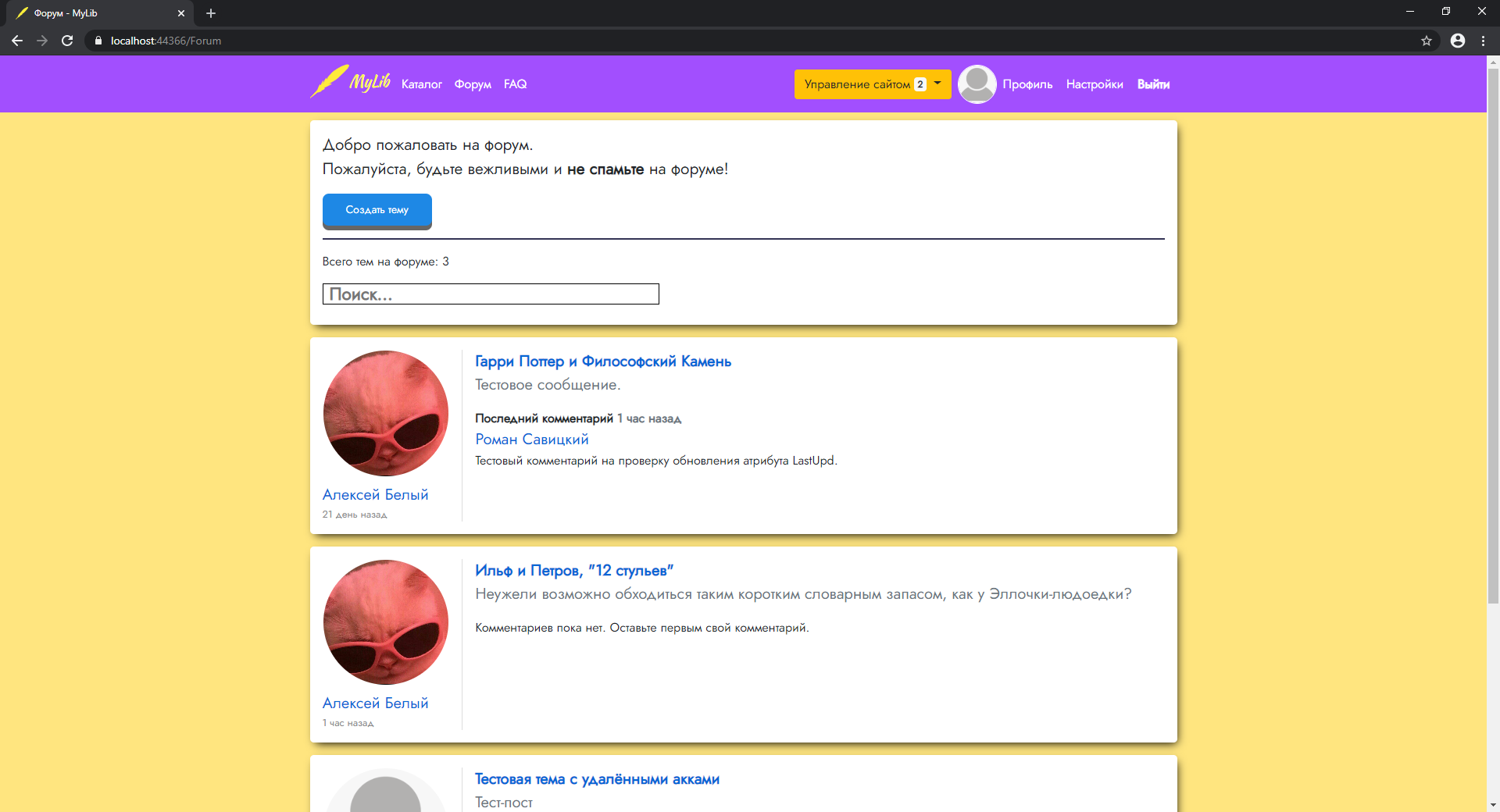


Рисунок 6.18 – Изменения в списке тем на форуме

6.5. FAQ (Часто задаваемые вопросы)

Раздел «Часто задаваемые вопросы» является самым простым в реализации, поскольку представляет из себя лишь набор некоторого количества карточек с вопросами и ответами на них.

Внешний вид страницы и один из развернутых вопросов демонстрируется на рисунке 6.19.

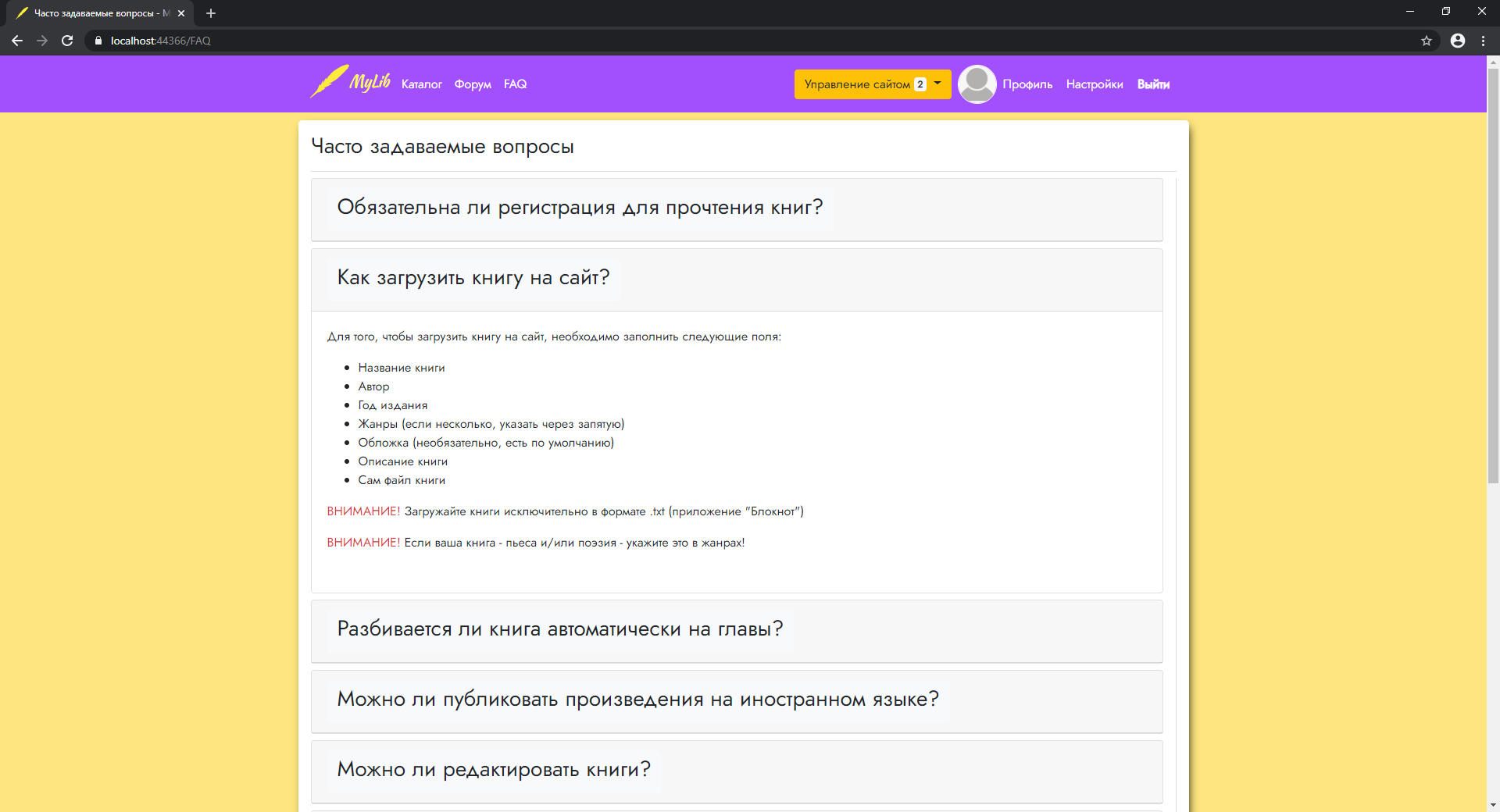


Рисунок 6.19 – Раздел «Часто задаваемые вопросы»

Для того, чтобы оформить такие карточки, применяются такое свойство, как схлопывание. Применяются два элемента: «toggler» или же кнопка, по которой происходит скрытие-раскрытие второго элемента, имеющего класс «collapse», и в котором содержится основной текст. Подобная технология также присутствует в разделе «Кластеризация» и в навигационной панели в мобильной версии.

6.6 Раздел «Управление сайтом»

Раздел «Управление сайтом» состоит из четырех частей – Менеджера тегов, Менеджера ролей, Кластеризация и Жалобы пользователей.

Как следует из названия и требований к проекту, данный раздел служит для того, чтобы администраторы имели возможность управлять сайтом без вмешательства в код программы.

Менеджер тегов позволяет администраторам собственноручно добавлять или удалять теги, по которым пользователи могут искать книги.

Главное меню представлено рисунком 6.20.

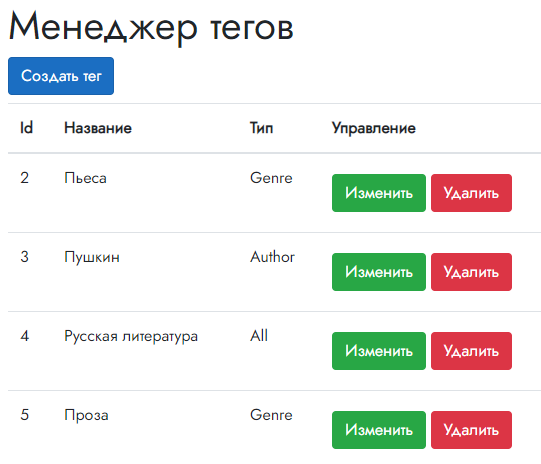


Рисунок 6.20 – Главною меню менеджера тегов

При нажатии на кнопку «Создать тег», администратору необходимо заполнить два поля: Название тега и его тип. Количество типов сведено до трех единиц – тег автора, тег жанра и глобальный тег. После того, как тег был создан, контроллер перенаправляет администратора на главное меню.

Из-за однотипности по функциональности всех кнопок удаления, опишем лишь кнопку «Изменить».

При переходе в раздел редакции тега администратору демонстрируется страница следующего вида (рисунок 6.21).

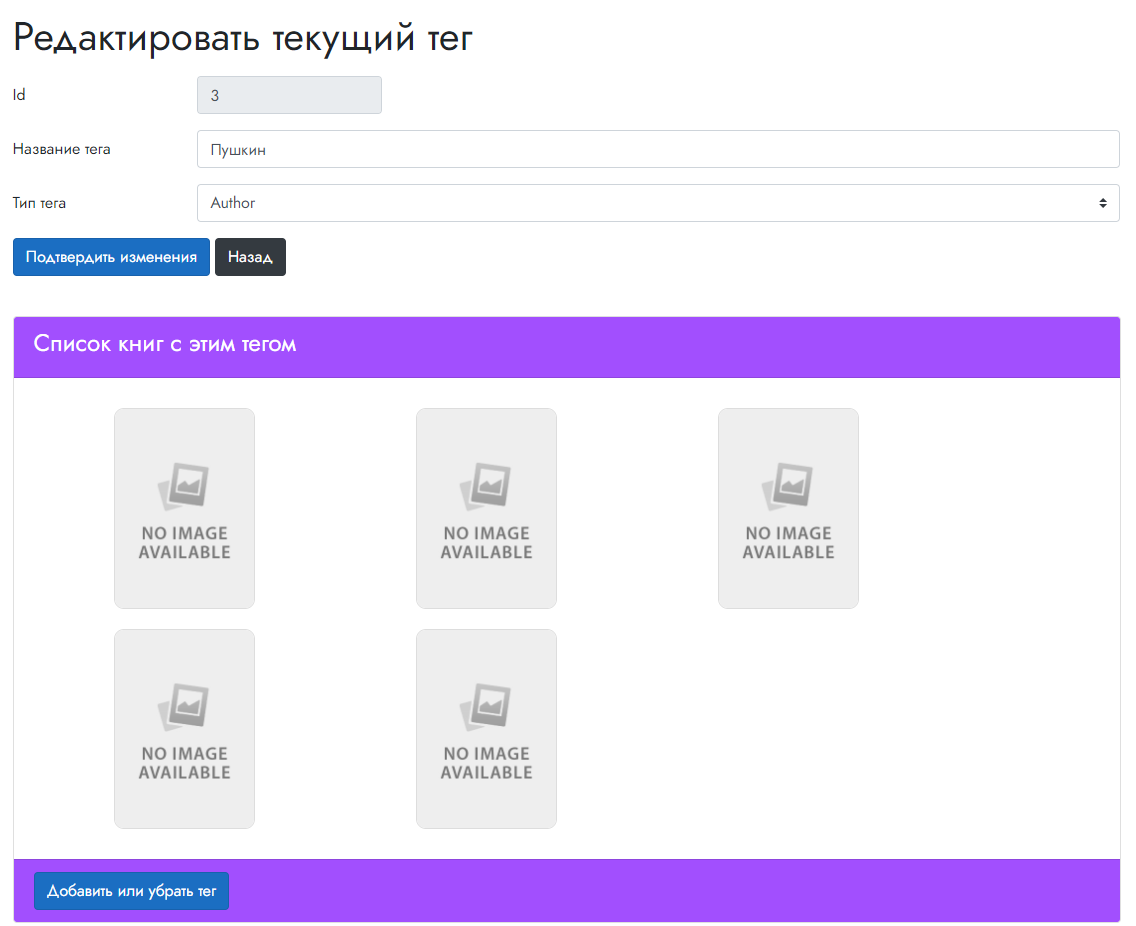


Рисунок 6.21 – Меню редакции тега

Меню состоит из двух частей: редактирование данных о теге, а также список книг, в которых присутствует данный тег. При переходе на кнопку «Добавить или удалить тег» отображается меню, указанное на рисунке 6.22.

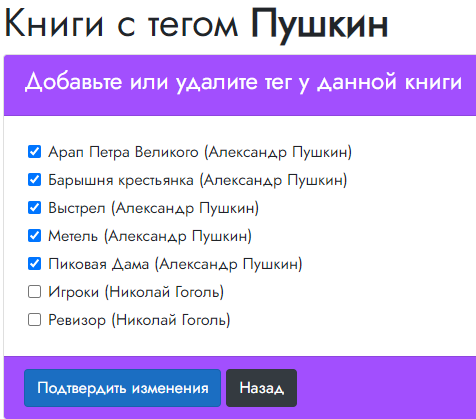


Рисунок 6.22 – Страница редактирования списка книг

В данном меню отображается весь список книг на сайте, и при помощи соответствующих указателей (checkbox) администратор может установить список книг, которые будут иметь данный тег.

При нажатии на любую из двух кнопок внизу, осуществляется переход на менеджер тегов.

Подобный дизайн и функциональность имеет и менеджер ролей.

Раздел «Кластеризация» предназначен для исследования и анализа книг, первичной сортировке данных по критериям (кластерам), и в конечном итоге, для присвоения тегов книгам, минуя соответствующий менеджер.

Главное меню раздела «Кластеризация» отображено на рисунке 6.23.

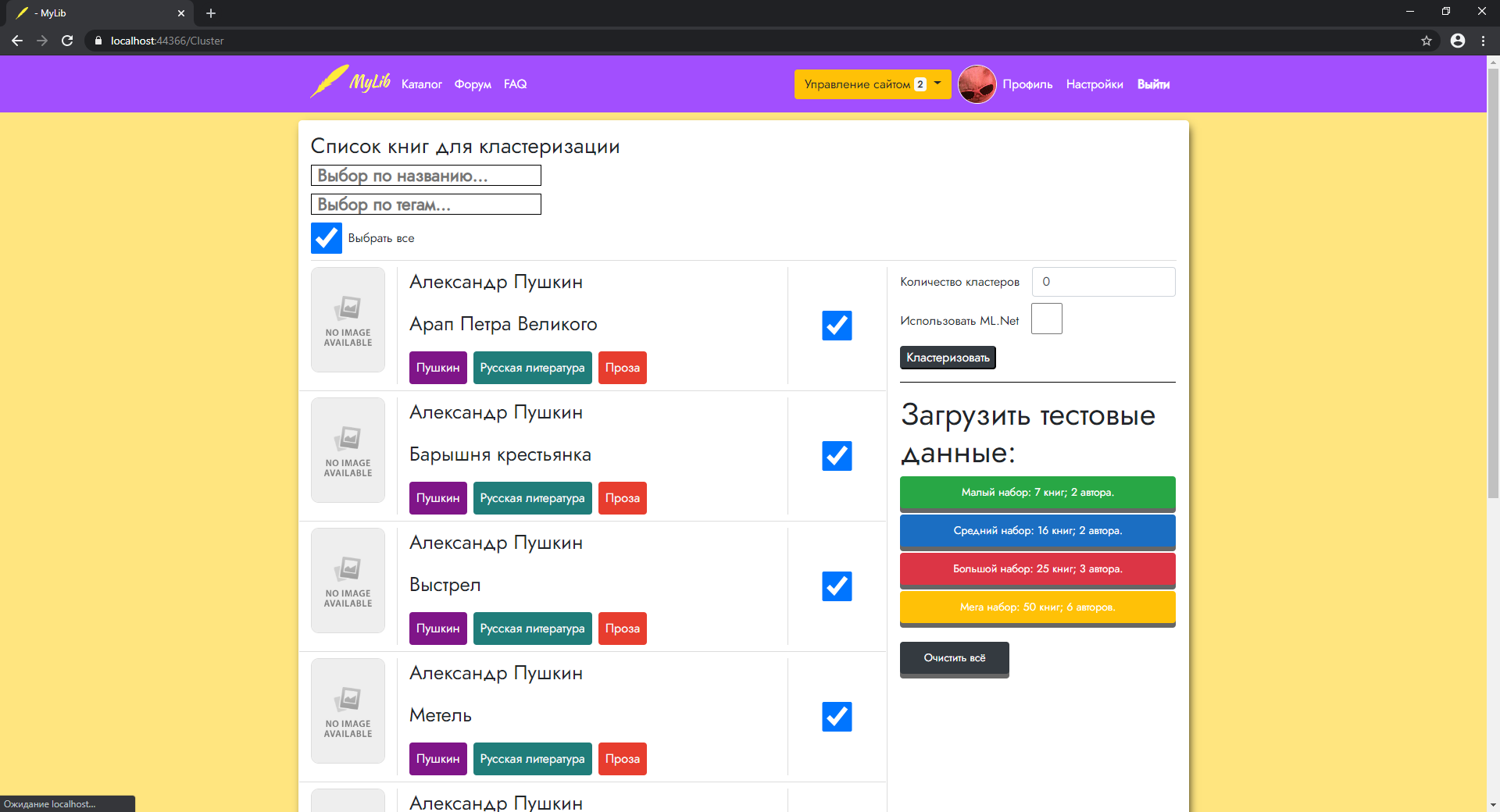


Рисунок 6.23 – Главное меню кластеризации

Как можно видеть из рисунка выше, страница имеет уже описанные ранее поисковые строки, кнопку «Выбрать все», список книг, расположенных на сайте, список входных данных (количество кластеров и выбор метода кластеризации), а также список выбора тестовых данных.

Предварительно загружен малый набор – семь книг от двух авторов. В качестве выходных данных установим три кластера и метод ML.Net. В качестве результата получим следующую страницу (рисунок 6.24).

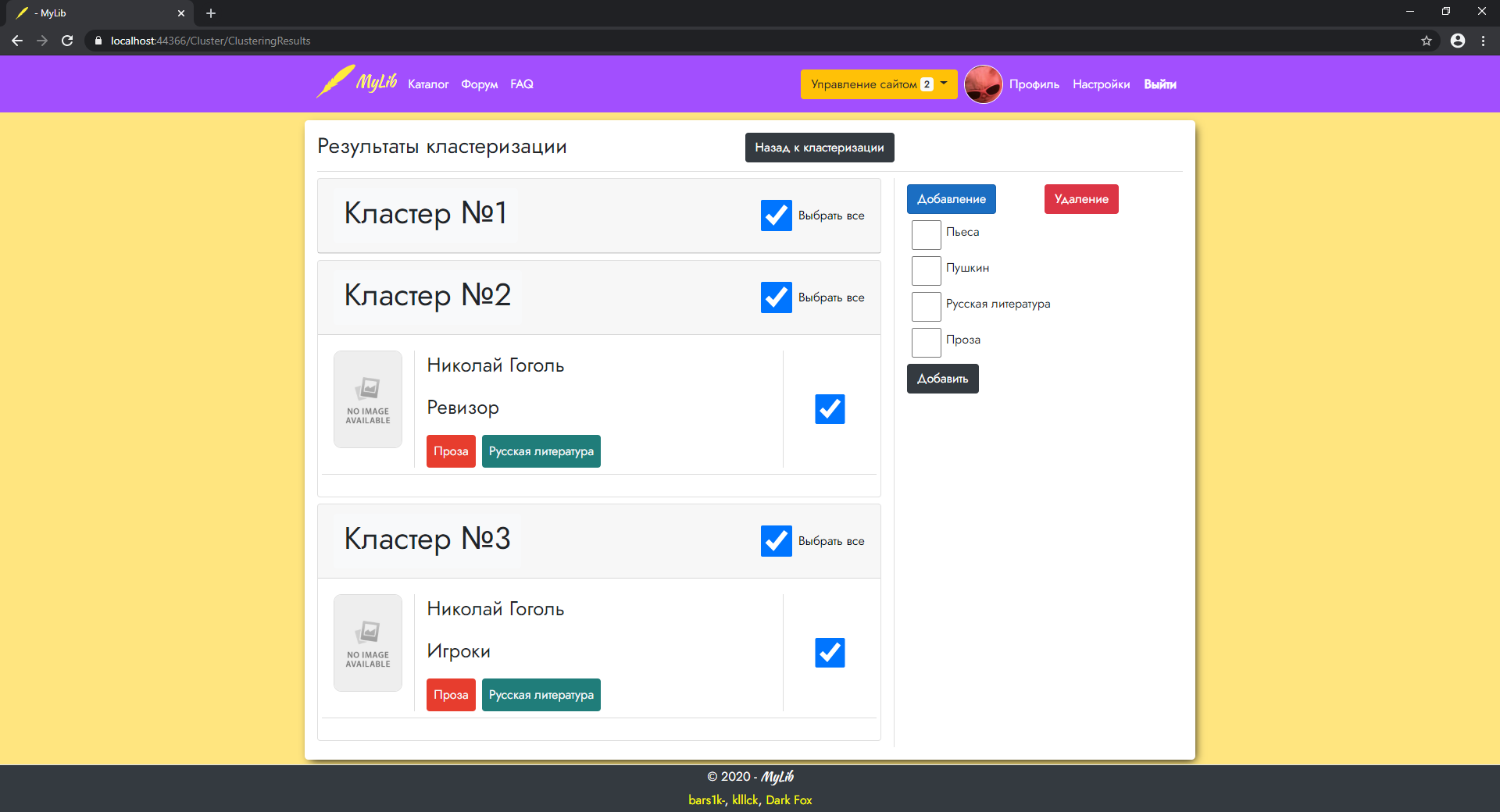


Рисунок 6.24 – Результат кластеризации данных

Набор книг, подверженный кластеризации, состоял из пяти книг Александра Пушкина и двух книг Николая Гоголя. В кластере №1 расположены все книги Пушкина, по двум остальным распределены книги Гоголя.

В правой части данной страницы расположены две цветные кнопки – добавление и удаление тега.

При нажатии на одну из этих кнопок, кнопка с чёрным цветом приобретает соответствующий функционал. При добавлении и/или удалении тега обновляется страница, на которой присутствует уже видоизменный набор тегов у книг.

Раздел «Жалобы пользователей» состоит из списка жалоб, которые ещё не были проверены администраторами. Сама страница отображена на рисунке 6.25.

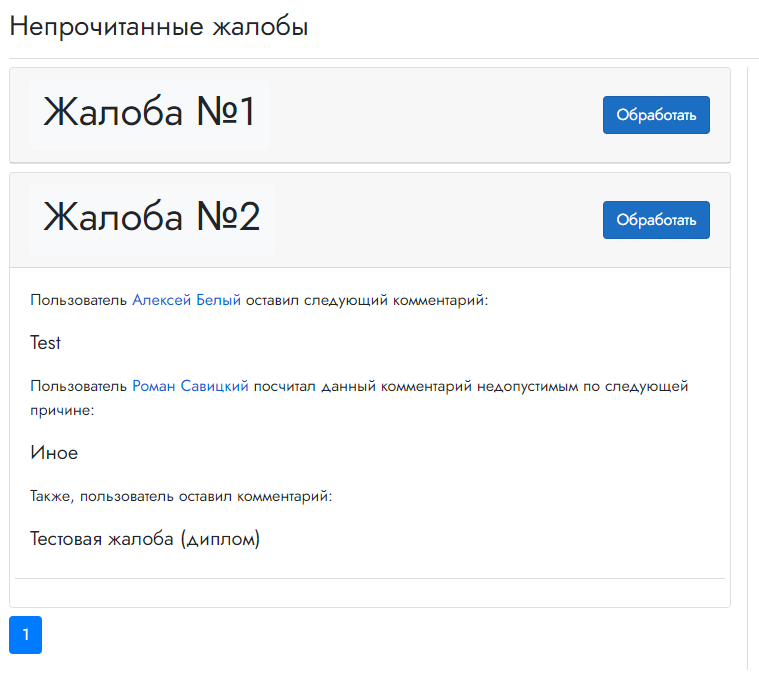


Рисунок 6.25 – Меню жалоб пользователей

На вышеуказанном рисунке уже раскрыта одна из жалоб, которая содержит в себе информацию о:

* пользователе и его комментарии, в отношении которых была подана жалоба;
* пользователе, который отправил жалобу, с указанной причиной и примечанием.

По нажатии на кнопку «Обработать», жалоба исчезает из списка. Рисунок 6.26 это подтверждает.

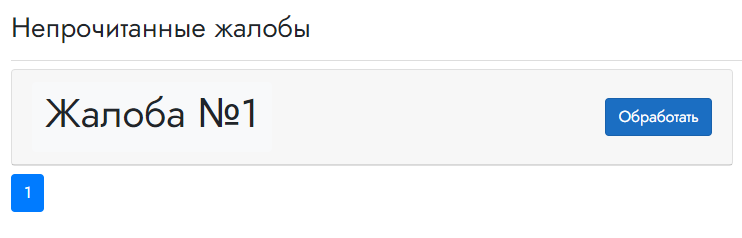


Рисунок 6.26 – Меню после обработки одной из жалоб

6.7 Профиль, настройки, регистрация и авторизация

При переходе по аватарке или по кнопке «Профиль», пользователь перенаправляется на свой профиль. В случае со ссылками на профили в других разделах – на профили соответствующих лиц. Личная страница пользователя выглядит так, как указано на рисунке 6.27.

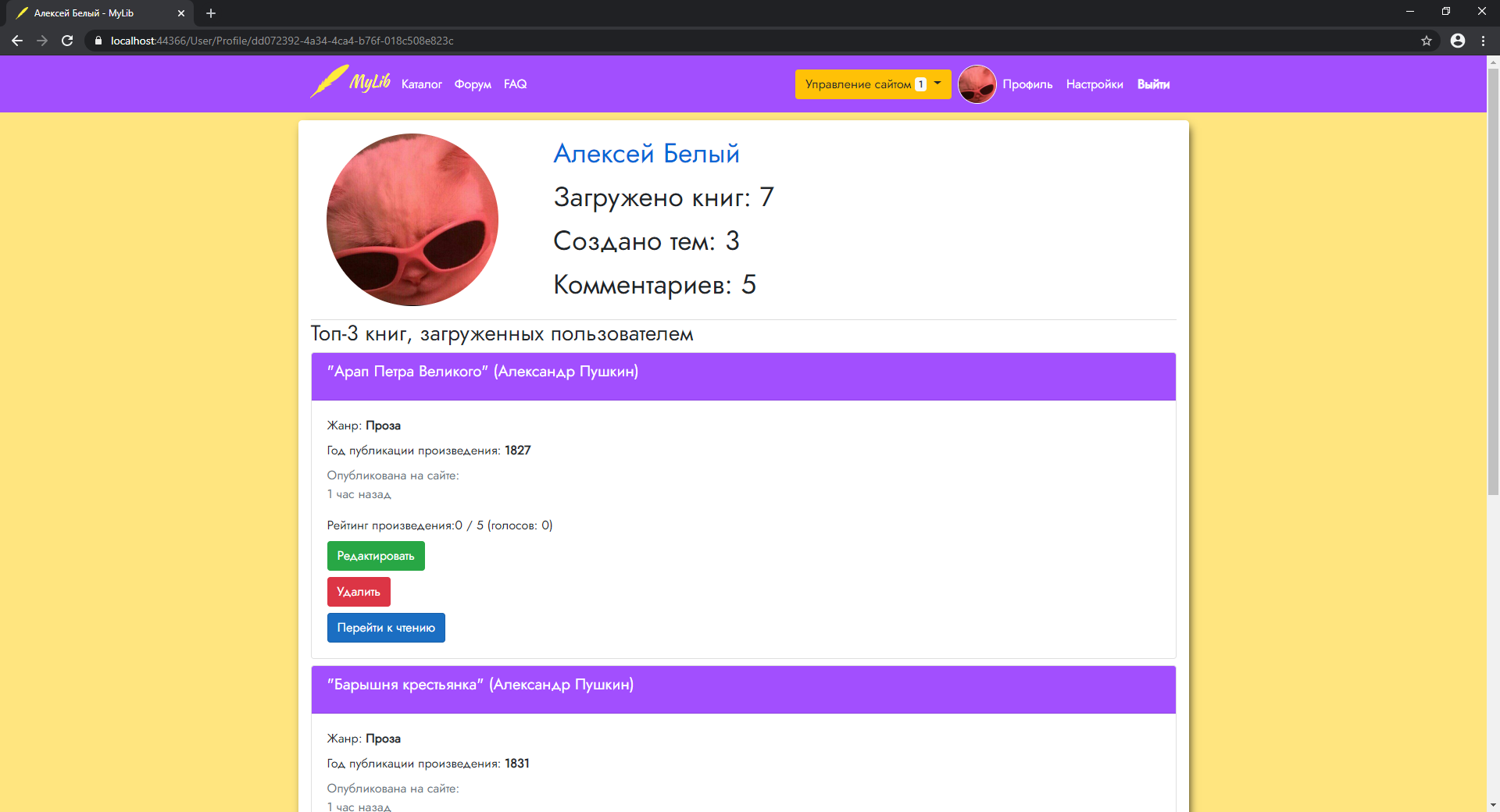


Рисунок 6.27 – Личная страница пользователя

Как можно видеть, страница содержит в себе некоторые статистические данные: количество загруженных книг, созданных тем и оставленных комментариев.

Также, на странице отображается до трех книг, загруженных пользователем, причем книги отсортированы по рейтингу, начиная с наиболее лучших.

При переходе в раздел «Настройки», пользователю предлагается изменить часть данных о своей учетной записи. Подробнее указано на рисунке 6.28.

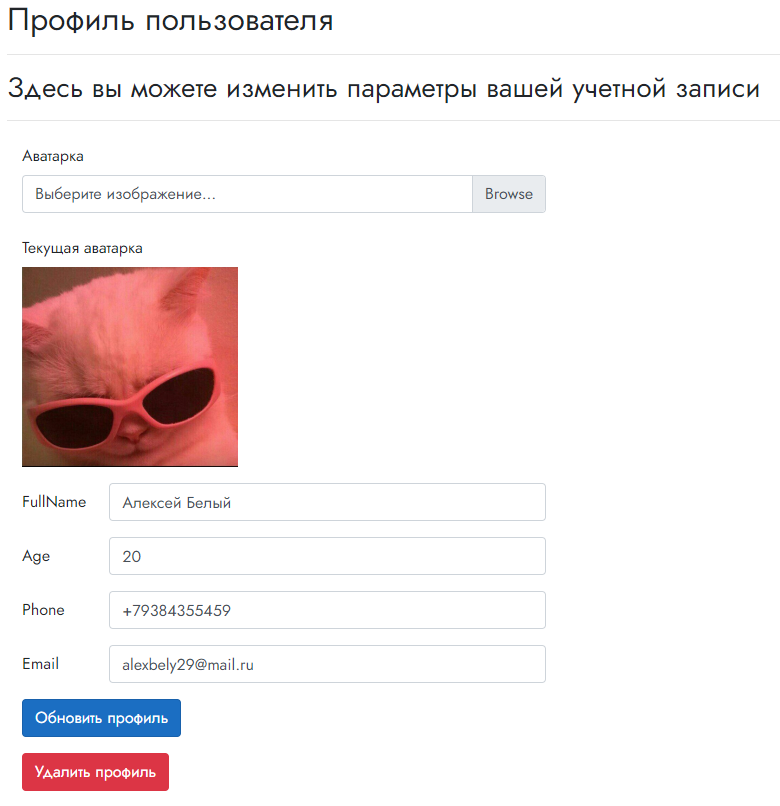


Рисунок 6.28 – Меню редакции профиля пользователя

Как видно из рисунка, пользователь вправе изменить фото профиля, свое на сайте, возраст, номер телефона и электронную почту. Изменения подтверждаются при помощи кнопки «Обновить профиль», удаление аккаунта осуществляется путём соответствующей кнопки, вызывающей при нажатии форму подтверждения.

Формы регистрации и авторизации имеют стандартный вид, представлены рисунками 6.29-6.30.

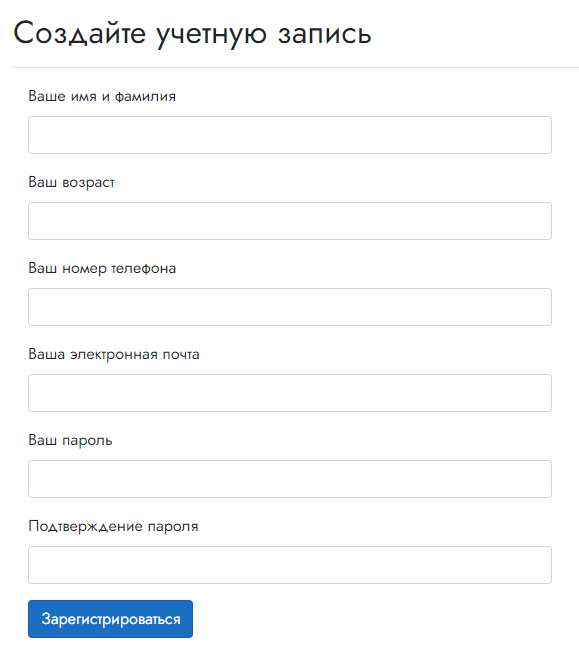


Рисунок 6.29 – Форма регистрации

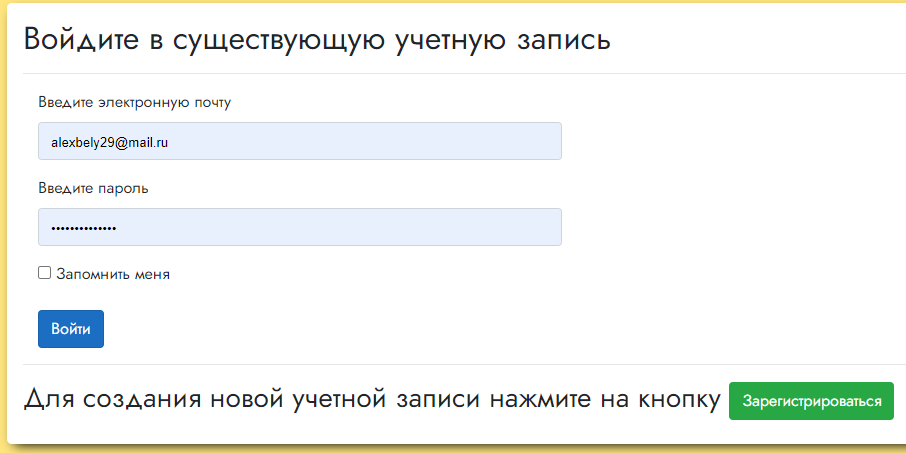


Рисунок 6.30 – Форма авторизации

# 7 Мобильная версия веб-приложения «Литература»

Фреймворк Bootstrap, как было ранее сказано в пункте 5.5, основывается на подходе, заключающемся в том, что при стилизации, дизайн для мобильных устройств является более приоритетной целью, чем дизайн для персональных компьютеров.

В мобильной версии данного веб-приложения, существенному видоизменению подверглась навигационная панель, которая теперь состоит из постоянно видимой части и части, которая скрывается при помощи тогглера.

Внешний вид веб-страницы для смартфона iPhone 5 указан на рисунке 7.1.



Рисунок 7.1 – Главная страница для неавторизованного пользователя

Расположенный в левой части навигационной панели тогллер позволяет показывать и/или скрывать ссылки на разделы «Каталог», «Форум», «FAQ». Также, если пользователь – администратор, тогглер содержит в себе выпадающий список с разделом «Управление сайтом».

Под выпадающим списком, расположенным справа, располагаются ссылки правой части стандартного навигационного меню.

Уведомления о новых жалобах для администраторов дублируются и на тогглере, что представлено рисунком 7.2.

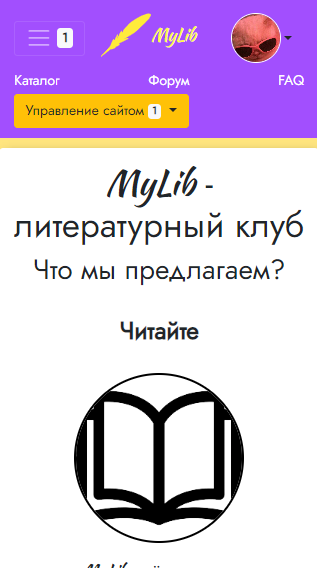


Рисунок 7.2 – Раскрытая навигационная панель

Небольшие изменения коснулись книжных «карточек», тем и комментариев. Внешний вид компонентов представлен рисунками 7.3-7.5.

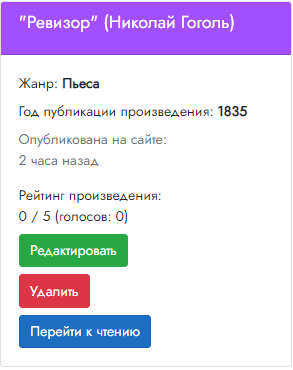


Рисунок 7.3 – Книжная «карточка» в мобильной версии

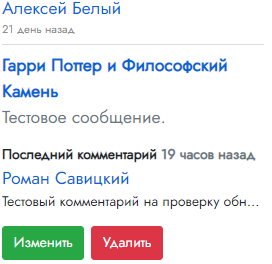


Рисунок 7.4 – Тема форума в мобильной версии

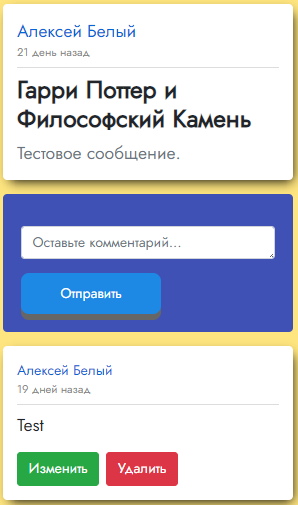


Рисунок 7.5 – Комментарий под темой форума в мобильной версии

# 8 Оценка эффективности программного продукта

Поскольку идея создать веб-приложение на литературную тематику не является новаторским, а также в связи с тем, что Рунете уже существуют довольно популярные сайты на данную тематику, в частности, «Стихи.ру»[7] и «Проза.ру»[8], имеет смысл применить в качестве оценки эффективности метод сравнения.

Так, как заданием на выпускную квалификационную работу являлась разработка клиентской части веб-приложения, сравним данные сайты с разработанным проектом с точки зрения графического оформления.

Оба портала, представленные источниками выше, работают под эгидой Российского союза писателей, и имеют практически идентичный дизайн.

Главное меню портала «Стихи.ру» представлено рисунком 8.1.

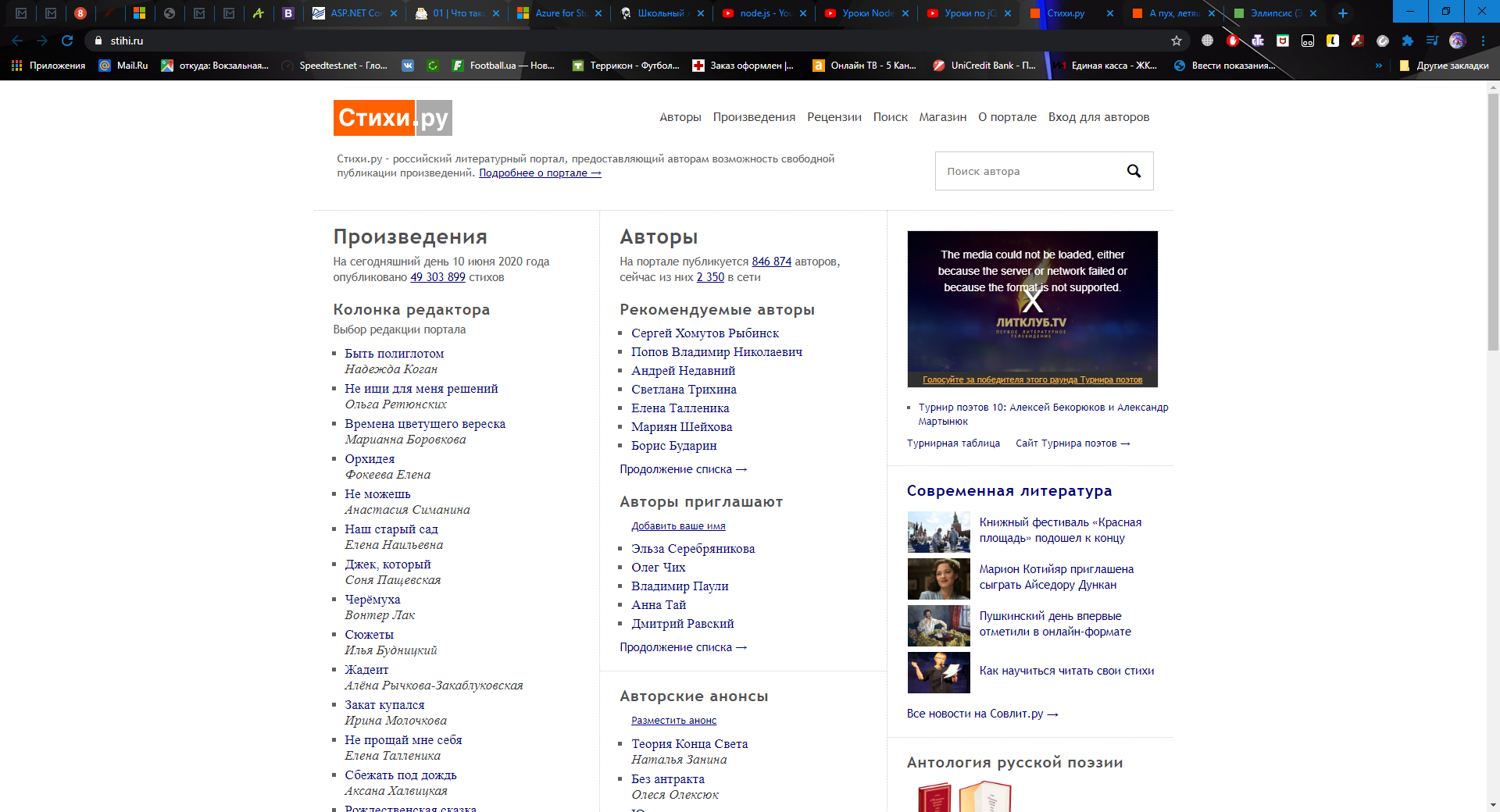


Рисунок 8.1 – Главное меню «Стихи.ру»

Исходя из структуры главного меню, можно видеть, что данный портал отличается предельной простотой (отсутствием сложных элементов), а вместо «карточек» для наилучших произведений, разработчики ограничились лишь тегом <li></li>. Рисунок 8.2, представляющий собой фрагмент из инспектора кода в браузере Google Chrome, подтверждает данное суждение.

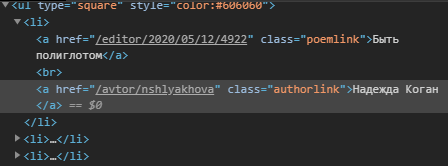


Рисунок 8.2 – Фрагмент кода главной страницы «Стихи.ру»

Что же касается форматов текста на данных порталах, тут ситуация выглядит печальнее. Выражается в том плане, что текст не является удобочитаемым, поскольку фон сайта является белым (прозрачным) (в формате RGBA – 0; 0; 0; 0), а цвет самого текста – черным (в формате RGBA – 0; 0; 0; 1). Данный контраст является нарушением одного из основных правил эргономичности сайта[9].

Также, тексты произведений на портале «Стихи.ру», не являются расположенными по центру, в то время, как разработанный проект придает внимание тому, чтобы произведение имело как можно больше сходств с обычной бумажной книгой.

Поскольку речь пошла о книгах, следует обратить также внимание, что тексты произведений на обоих порталах не имеют абсолютно никакого разбиения по страницам, в отличие от книг, размещаемых на разработанном веб-приложении (рисунок 8.3).



Рисунок 8.3 – Одна из повестей на сайте «Проза.ру»

Сочетание таких факторов делает чтение произведений менее комфортным, а в случае долгого чтения, глаза будут находиться в большем напряжении, что может привести к болезням органов зрения. Схема из источника [9] продемонстрирована на рисунке 8.4.



Рисунок 8.4 – Анализ рисков при чтении черного текста на белом фоне

Исходя из вышеуказанной схемы, следует использовать темно-серый текст на белом фоне. Такое стилистическое оформление уже встроено в библиотеку стилей Bootstrap: для всей страницы применяется темно-серый текст (в кодировке HEX - #212529).

Открыв инспектор кода в любом браузере на любой странице с произведением, можно убедиться, что данное цветовое значение осталось нетронутым (рисунок 8.5, селектор body, свойство color).

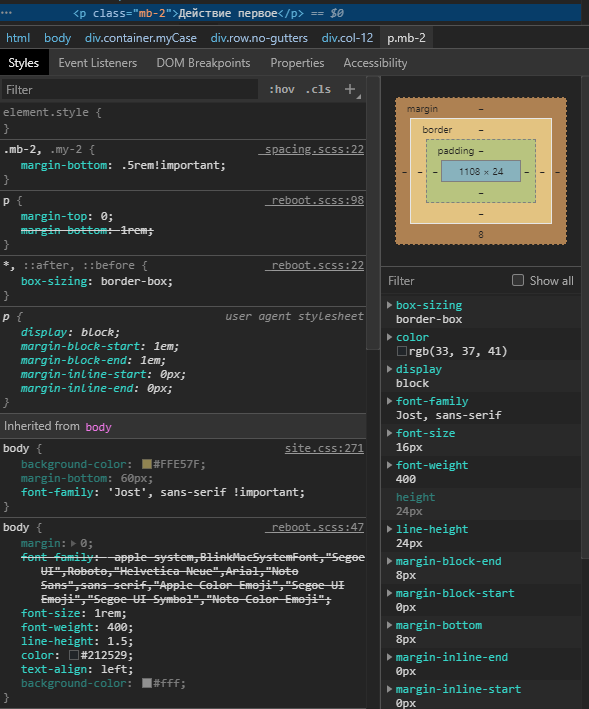


Рисунок 8.5 – Цвет текста для произведения унаследован от тела документа

Таким образом, перед тем, как проектировать веб-страницы, ставилась задача придать произведениям больше аналогии с обычными книгами, при этом не отклоняться от правил эргономичности сайта.

# 9 Безопасность и экологичность проекта

9.1 Значение и задачи безопасности жизнедеятельности

В настоящее время всемирная паутина пользуется большой популярностью практически в любой стране мира. Все успешные компании очень активно используют сеть Интернет для обмена различной внутрикорпоративной информацией, а также в целях рекламирования своей продукции. В совокупности это даёт высокий социально-экономический эффект, в частности, увеличивается скорость обмена информацией, а также повышается рост производительности труда.

Персональные ЭВМ, а также КПК негативно воздействуют на человека незаметным образом. При постоянной и продолжительной работе с ПЭВМ и/или КПК, определённые факторы могут неблагоприятно повлиять на самочувствие человека, его безопасность и здоровье. Длительное же времяпрепровождение за ПЭВМ и/или КПК может способствовать быстрому переутомлению, нарушению функций зрения, а также эмоционально-нервному перенапряжению.

Главными задачами безопасности жизнедеятельности и охраны труда является: ликвидация, минимизация и снижение уровня влияния неблагоприятных факторов на человеческий организм, а также защита от отрицательных последствий их влияния. Для того, чтобы избежать эти негативные последствия, выполнены необходимые меры безопасности при работе с ПЭВМ и/или КПК, которые регламентированы в следующих нормативных документах:

– СанПиН 2.2.2/2.4.1340‑03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»;

– «Трудовой кодекс Российской Федерации» от 30.12.2001 № 197-ФЗ (ред. от 10.01.2016);

– ТОИ Р‑45‑084‑01 «Типовая инструкция по охране труда при работе на персональном компьютере»;

– ГОСТ 12.0.003‑2015 «Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация»;

– ГОСТ 12.1.019-2017 «Электробезопасность»;

– Федеральный закон от 22.07.2008 №123-ФЗ (ред. от 29.07.2017) «Технический регламент о требованиях пожарной безопасности».

9.2 Анализ условий труда и мероприятия по защите от воздействия вредных производственных факторов

«Литература» – это веб-сайт, который сочетает в себе функционал общественной электронной библиотеки, а также дискуссионного клуба. Работа администратора и модераторов сайта требует определённых усилий и концентрации внимания. Из-за этих факторов присутствует риск приобретения следующих нарушений: переутомление, усталость, раздражительность, а также нарушение функций зрения.

Отрицательное воздействие на человека вычислительной техники менее выражено, сглажено многими положительными моментами в связи с постоянным развитием технологии производства мониторов. Однако, у людей, длительно использующих ПЭВМ и/или КПК могут быть отмечены такие реакции как: переутомление, усталость, раздражительность, а также нарушение функций зрения.

Для того, чтобы минимизировать и ликвидировать последствия, описанные выше, были созданы комфортные условия для работы, а также были соблюдены все нормы по устранению тяжести и напряжённости трудового процесса.

Поскольку речь в начале данного пункта шла про администраторов и модераторов сайта, оценка напряжённости проведена в отношении именно данной категории лиц, находящихся в одном офисе.

Таблица 9.1, в которой производится расчёт оценки напряженности трудового процесса, является шаблоном, описанным в нормативном документе Р 2.2.2006-05.

Таблица 9.1 – Оценка напряженности трудового процесса

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Показатели** | **Класс условий труда** | | | | |
| **1** | | **2** | **3.1** | **3.2** |
| 1 Интеллектуальные нагрузки | | | | | |
| 1.1 Содержание работы | |  | + |  |  |
| 1.2 Восприятие информации и их оценка | |  | + |  |  |
| 1.3 Распределение функций по степени сложности задания | |  | + |  |  |
| 1.4 Характер выполняемой работы | |  | + |  |  |
| 2 Сенсорные нагрузки | | | | | |
| 2.1 Длительность сосредоточенного наблюдения (% времени смены) | |  |  | + |  |
| 2.2 Плотность сигналов (световых, звуковых) и сообщений в среднем за 1 час работы | | + |  |  |  |
| 2.3 Число производственных объектов одновременного наблюдения | | + |  |  |  |
| 2.4 Размер объекта различения (при расстоянии от глаз работающего до объекта различения не более 0,5 м) в мм при длительности сосредоточенного наблюдения (% времени смены) | |  | + |  |  |
| 2.5 Работа с оптическими приборами при длительности сосредоточения наблюдения | | + |  |  |  |
| 2.6 Наблюдение за экранами видеотерминалов (часов в смену) | | + |  |  |  |
| 2.7 Нагрузка на слуховой анализатор | | + |  |  |  |
| 2.8 Нагрузка на голосовой аппарат | | + |  |  |  |

*Окончание таблицы 9.1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 Эмоциональные нагрузки | | | | |
| 3.1 Степень ответственности за результат собственной деятельности. Значимость ошибки |  | + |  |  |
| 3.2 Степень риска для собственной жизни | + |  |  |  |
| 3.3 Ответственность за безопасность других лиц |  |  | + |  |
| 3.4 Количество конфликтных ситуаций в течение смены |  | + |  |  |
| 4 Монотонность нагрузок | | | | |
| 4.1 Число элементов, необходимых для реализации простого задания или многократно повторяющихся операций |  | + |  |  |
| 4.2 Продолжительность выполнения простых заданий или повторяющихся операций |  | + |  |  |
| 4.3 Время активных действий | + |  |  |  |
| 4.4 Монотонность производственной обстановки (время пассивного наблюдения за ходом техпроцесса в % от времени смены) |  | + |  |  |
| 5 Режим работы | | | | |
| 5.1 Фактическая продолжительность рабочего дня |  | + |  |  |
| 5.2 Сменность работы |  | + |  |  |
| 5.3 Наличие регламентированных перерывов и их продолжительность | + |  |  |  |
| Количество показателей в каждом классе | 9 | 12 | 2 | 0 |
| Общая оценка напряженности труда | 2 | | | |

После проведённого расчёта и анализа, были получены следующие результаты:

В первый класс («Оптимальный») входят 9 показателей, во второй («Допустимый») – 12 показателей, в класс 3.1 («Вредный первой степени») – 2 показателя, а в класс 3.2 («Вредный второй степени») – ни одного.

Учитывая, что при оценке было выявлено менее шести критериев, относящихся к классам 3.1 – 3.2, то согласно пункту 6.3.2 Приложения 16 нормативного документа «Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификации условий труда» (Р 2.2.2006-05), уровень напряженности труда следует считать *допустимым*.

9.3 Обеспечение электробезопасности

Требования электробезопасности изложены в ряде нормативных документов, основными из которых являются следующие:

– Правила технической эксплуатации электроустановок потребителей (ПТЭЭП), утвержденные приказом Минэнерго России от 13.01.2003 N 6;

– Правила по охране труда при эксплуатации электроустановок, утвержденные приказом Минтруда России от 24.07.2013 N 328н;

– Инструкция по применению и испытанию средств защиты, используемых в электроустановках, утвержденная приказом Минэнерго России от 30 июня 2003 N 261.

Для обеспечения электробезопасности, а также в случаях со сбоями в электропитании помещений, каждый персональный компьютер, а также сервер, оборудованы источниками бесперебойного питания (ИБП), ресурс которых позволяет при небольшом сбое в питании не потерять важные файлы, а в случае длительного отключения – сохранить документы в текущем состоянии и завершить работу персонального компьютера в обычном режиме.

Рабочие места сотрудников имеют уровень умеренной опасности, согласно ГОСТ 12.1.019-2017 «Электробезопасность». Основанием для такого вывода являются следующие факторы:

– относительная влажность в помещениях – 50-55%;

– отсутствие токопроводящей пыли;

– полы сделаны из материалов, не способных проводить ток, следовательно, они изолированы;

– температура в офисных помещениях поддерживается на уровне 20-22°С;

– возможность прикосновения человека к металлоконструкциям, а также к металлическим корпусам электрооборудования сведена к минимуму;

– для защиты от поражения сотрудников электрическим током в случая появления напряжение на корпусах и других частях электрооборудования предусмотрено наличие заземления, а также регулярное проведение профилактических мер по выявлению повреждений в изоляции электропроводки.

9.4 Обеспечение пожарной безопасности

Основные документы, регламентирующие нормы пожарной безопасности в офисе, являются: Федеральный закон № 123-ФЗ от 22 июля 2008 г. «Технический регламент о требованиях пожарной безопасности» и Правила противопожарного режима Российской Федерации, утвержденные постановлением Правительства РФ от 25 июля 2012 г. № 390 «О противопожарном режиме».

Категория помещений по пожарной и взрывопожарной опасности определяется Федеральным законом от 10.07.2012г. № 117-ФЗ «Определение категории зданий, сооружений и помещений по пожарной и взрывопожарной опасности».

Поскольку помещения, в которых располагается всё необходимое для работы оборудование является офисным, то согласно пункту 2 статьи 27 Федерального закона №117-ФЗ, данное здание не подлежит разделению на категории, поскольку не является ни производственным зданием, ни складским.

Так как проект является веб-сайтом, ему требуется расположение на определенном веб-сервере. На первых порах, веб-сайт может располагаться на чужих серверах, а в случае хорошей окупаемости проекта, будет стоять вопрос о приобретении комплектующих для собственного веб-сервера.

Поскольку веб-серверы, особенно в последнее время, подвержены перегрузкам, влекущими за собой отказ в обслуживании, являются источниками потенциального пожара, то во всех помещениях созданы условия повышенной пожарной безопасности.

Для обеспечения пожарной безопасности в офисном помещении, проведены следующие меры:

1. Установлена пожарная сигнализация и система пожаротушения;

2. Серверная, а также офисные комнаты оборудованы системой пожаротушения, основанные на «сухой воде» (хладон ФК-5-1-12). Причиной выбора данной системы является тот факт, что вещество, применяемое при пожаротушении, является диэлектриком (таким образом не портит электроприборы и проводку), а также при распылении не снижает концентрацию кислорода (что очень важно при пожарной эвакуации;

3. Для всего офисного помещения разработан план эвакуации при пожаре, согласно статье 53 Федерального закона №123-ФЗ;

4. В доступных местах вывешены наглядные материалы (наклейки, указатели), показывающие путь к выходу;

5. На путях к выходу из зданий отсутствуют какие-либо заграждения;

6. В каждом помещении вывешены таблички, на которых указан номер пожарной охраны, а также вывешена инструкция по определению порядка обесточивания всего электрооборудования в случае пожарной тревоги, а также в конце рабочего дня;

7. Определён порядок и сроки прохождения противопожарного инструктажа и занятий по пожарно-техническому минимуму, назначены люди, ответственные за их проведение;

8. Назначен сотрудник, отвечающий за пожарную безопасность в здании;

9. Здание оснащено основным и запасным выходом, которые в случае пожарной тревоги, подадут звуковой и световой сигналы.

Данные мероприятия направлены на уменьшение риска возникновения возгорания, а также на проведение оперативных мер по ликвидации возгорания, если все же оно произошло.

# Заключение

В результате выполнения выпускной квалификационной работы был проведен детальный анализ уже существующих веб-приложений по литературной тематике, а также реализована клиентская часть веб-приложения «Литература». В сравнении с веб-дизайном проанализированных сайтов, разработанная клиентская часть предоставляет пользователям более эргономичный пользовательский интерфейс для чтения книг и их дальнейшего обсуждения.

Пользователи могут читать, загружать, оценивать и обсуждать книги, размещенные самими же пользователями, вне зависимости от используемого для веб-серфинга типа устройства.

В ходе разработки данного проекта были созданы представления, а также модели для представления данных, являющимися проекциями данных из БД.

При разработке клиентской части приложения была детально изучена библиотека стилей Bootstrap, в частности подход Mobile-First. Также, были существенно расширены знания в области frontend-разработки, в связи с полученным бесценным опытом работы в платформе ASP.NET Core.

При дальнейшем сопровождении, а также постоянном улучшении данной системы, существует перспектива того, что разработанный веб-сайт будет иметь охват аудитории, больший, чем у порталов «Стихи.ру» и «Проза.ру» в сумме, и в конечном итоге, дорастет до популярного портала, как в свое время таким порталом стала социальная сеть «ВКонтакте».

# Список использованных источников

1. Научная электронная библиотека «КиберЛенинка». [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/shkolnyy-literaturnyy-klub-kak-sredstvo-razvitiya-literaturno-tvorcheskih-sposobnostey-uchaschihsya/viewer (дата обращения 22.05.2020).
2. DesigNonStop – о дизайне без остановки! [Электронный ресурс]. URL: https://www.designonstop.com/webdesign/article/chto-takoe-veb-dizajn-ego-osnovnye-elementy-etapy-i-principy.htm (дата обращения: 22.05.2020)
3. Bootstrap - The most popular HTML, CSS, and JS library in the world. [Электронный ресурс]. URL: https://getbootstrap.com (дата обращения: 23.05.2020)
4. Bootstrap – Документация на русском языке. [Электронный ресурс]. URL: https://bootstrap-4.ru (дата обращения: 23.05.2020)
5. ASP.NET – GitHub. [Электронный ресурс]. URL: https://github.com/aspnet (дата обращения: 22.05.2020)
6. Metanit – сайт о программировании. [Электронный ресурс]. URL: https://metanit.com/sharp/mvc5/1.1.php (дата обращения: 23.05.2020)
7. Проза.ру [Электронный ресурс]. URL: https://proza.ru (дата обращения: 08.06.2020)
8. Стихи.ру [Электронный ресурс]. URL: https://stihi.ru (дата обращения: 08.06.2020)
9. Юзабилити аудит сайта [Электронный ресурс]. URL: https://prof-mk.ru (дата обращения: 08.06.2020)

Приложение А

**Листинг программы-конвертера дат**

string ConvertDate(DateTime date)

{

TimeSpan timeDifference = DateTime.Now - date;

if (timeDifference.TotalDays > 365)

return date.ToString();

else

if (timeDifference.TotalDays > 31)

{

int months = (int)timeDifference.TotalDays / 31;

if (months == 1) return $"{Math.Floor(timeDifference.TotalDays)} месяц назад";

else if (months <= 4) return $"{Math.Floor(timeDifference.TotalDays)} месяца назад";

else return $"{Math.Floor(timeDifference.TotalDays)} месяцев назад";

}

else

if (timeDifference.TotalDays > 1)

{

int decs = (int)timeDifference.TotalDays / 10;

int solo = (int)timeDifference.TotalDays % 10;

if (decs == 1)

return $"{Math.Floor(timeDifference.TotalDays)} дней назад";

else

{

if (solo == 1) return $"{Math.Floor(timeDifference.TotalDays)} день назад";

else if (solo >= 2 && solo <= 4) return $"{Math.Floor(timeDifference.TotalDays)} дня назад";

else return $"{Math.Floor(timeDifference.TotalDays)} дней назад";

}

}

else

if (timeDifference.TotalHours > 1)

{

int decs = (int)timeDifference.TotalHours / 10;

int solo = (int)timeDifference.TotalHours % 10;

if (decs == 1)

return $"{Math.Floor(timeDifference.TotalHours)} часов назад";

else

{

if (solo == 1) return $"{Math.Floor(timeDifference.TotalHours)} час назад";

else if (solo >= 2 && solo <= 4) return $"{Math.Floor(timeDifference.TotalHours)} часа назад";

else return $"{Math.Floor(timeDifference.TotalHours)} часов назад";

}

}

else

if (timeDifference.TotalMinutes > 1)

{

int decs = (int)timeDifference.TotalMinutes / 10;

int solo = (int)timeDifference.TotalMinutes % 10;

if (decs == 1)

return $"{Math.Floor(timeDifference.TotalMinutes)} минут назад";

else

{

if (solo == 1) return $"{Math.Floor(timeDifference.TotalMinutes)} минута назад";

else if (solo >= 2 && solo <= 4) return $"{Math.Floor(timeDifference.TotalMinutes)} минуты назад";

else return $"{Math.Floor(timeDifference.TotalMinutes)} минут назад";

}

}

else

return "меньше минуты назад";

}