

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе № 2
по дисциплине «Web-технологии»
Тема: REST-приложение управления библиотекой.

Студент гр. 0382

Корсунов А.А.

Преподаватель

Беляев С.А.

Санкт-Петербург

2022

1. Цель работы.

Целью работы является изучение взаимодействия клиентского приложения с серверной частью, освоение шаблонов web-страниц, формирование навыков разработки динамических HTML-страниц, освоение принципов построения приложений с насыщенным интерфейсом пользователя.

2. Основные теоретические сведения.

CSS (Cascading Style Sheets – каскадные таблицы стилей) – язык описания внешнего вида документа, написанного с использованием языка разметки, используется как средство оформления внешнего вида HTML-страниц. Express – это минималистичный и гибкий web-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и web-приложений. Pug и EJS – модули, позволяющие использовать шаблоны для HTML-страниц. REST (Representational State Transfer – передача состояния представления) – стиль взаимодействия компонентов распределенного приложения. В рамках лабораторной работы – браузера и сервера web-приложения. Для взаимодействия используются стандартные методы:

- GET – получение записи (записей);
- POST – добавление записи;
- PUT – обновление или добавление записи;
- DELETE – удаление записи

3. Постановка задачи.

Задачи:

- разработка интерфейса web-приложения;
- задание стилей для отображения web-приложения с учётом размера экрана (использование на компьютере, на мобильном телефоне);

- создание web-сервера на основе express;
- создание шаблонов web-страниц;
- настройка маршрутов;
- создание json-хранилища;
- обработка REST-запросов;

4. Ход работы.

1) Для всех страниц web-приложения был разработан макет интерфейса с использованием Figma.



Рисунок 1 — макет стартовой страницы

Макет стартовой страницы (рис. 1) состоит из экрана приветствия пользователя и кнопки перехода на страницу библиотеки.

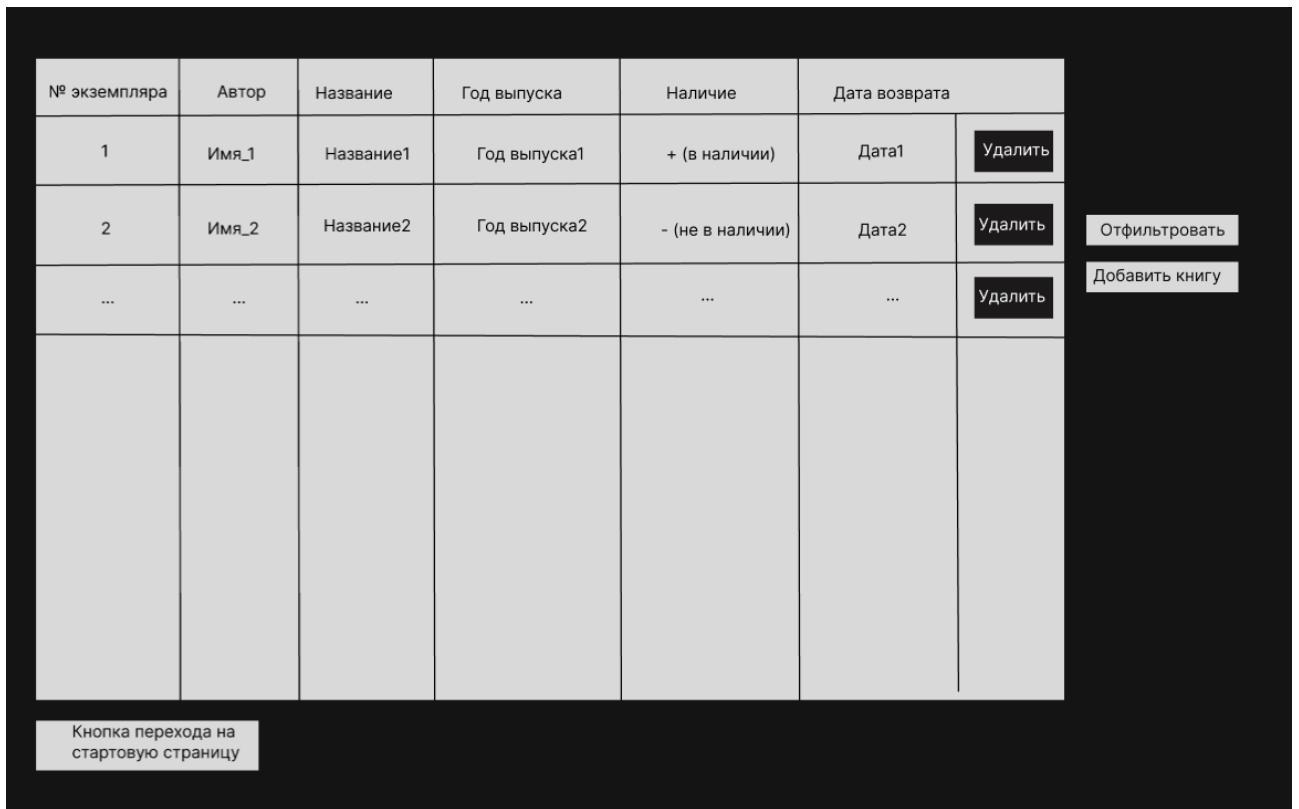


Рисунок 2 — макет страницы библиотеки

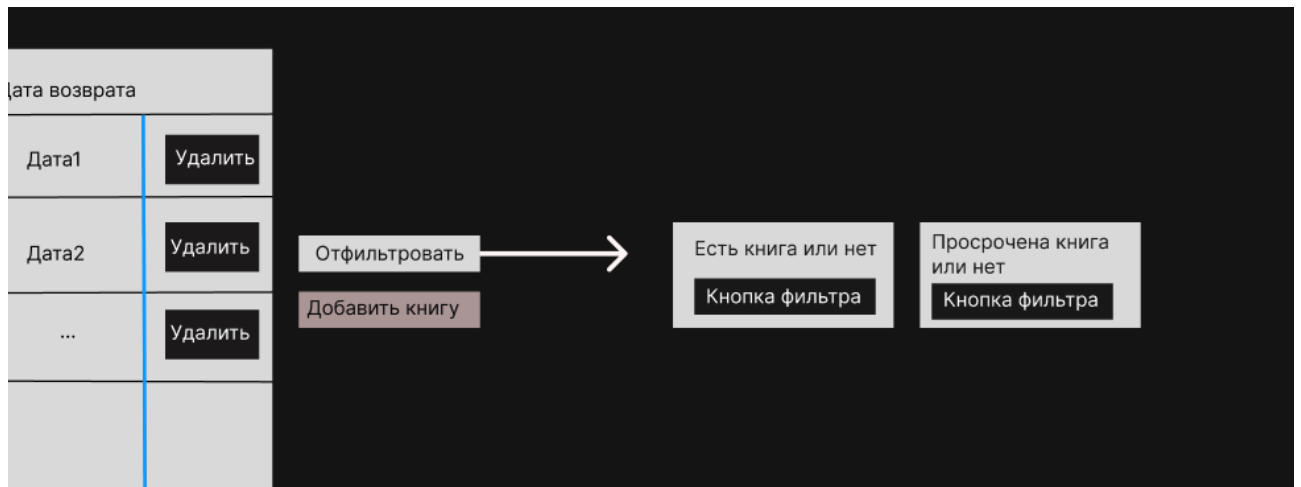


Рисунок 3 — макет интерфейса фильтрации

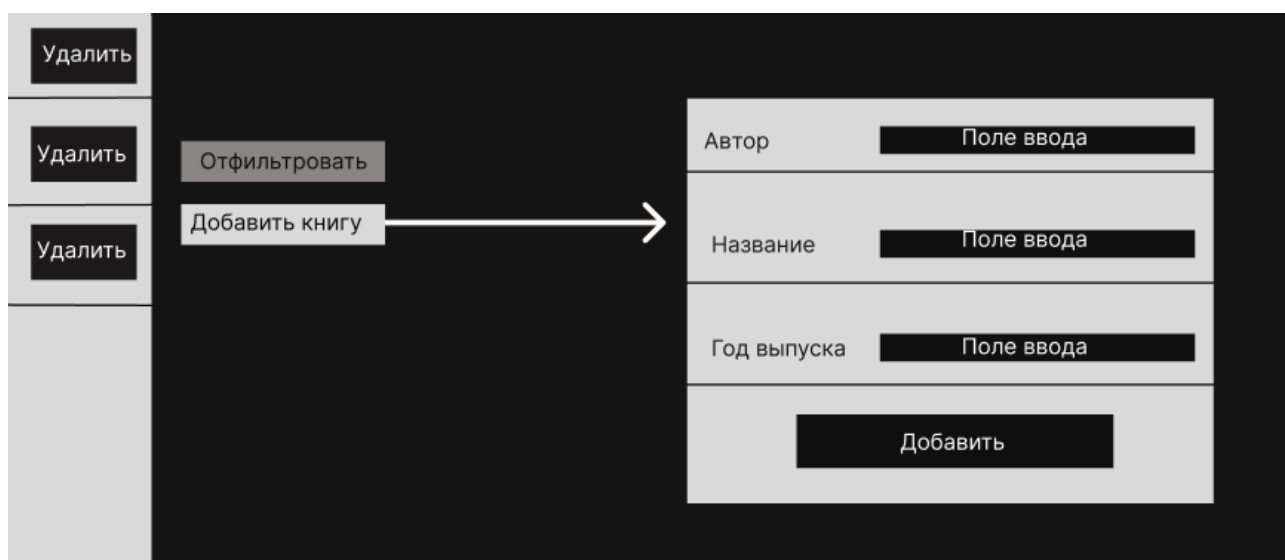


Рисунок 4 — макет интерфейса возможности добавления книги

Макет страницы библиотеки (рис. 2) состоит главным образом из таблицы, в которой перечислены атрибуты каждого экземпляра книги (кто является автором, название книги и подобное). Каждый экземпляр можно удалить из таблицы, нажав на соответствующую иконку (удаление с подтверждением).

Список книг можно отфильтровать по четырем критериям (рис. 3): первый и второй — по наличию экземпляра книги в библиотеки, третий и четвертый — просрочена ли книга или нет. При нажатии на кнопку фильтрации, кнопка добавления новой книги перестанет быть активной и будет затемнена до тех пор, пока пользователь не закроет окно фильтрации (повторным нажатием на кнопку фильтрации), либо же пока не отфильтрует книги или так или иначе перезагрузит страницу библиотеки.

Новую книгу можно добавить при нажатии на соответствующую кнопку (рис. 4). Пользователю будет представлена таблица, в которой он может указать три атрибута нового экземпляра книги (заполнив соответствующие поля ввода), после чего книга будет добавлена в таблицу при нажатии на кнопку добавления.

Автор	Имя автора (можно изменить)	Кнопка выдачи книги клиенту
Название	Название книги (можно изменить)	
Год выпуска	Год выпуска (можно изменить)	
В наличии	"+"	
		Кто берет <input type="text" value="Поле ввода"/> Когда вернуть <input type="text" value="Поле ввода"/>
Кнопка возврата в библиотеку		Кнопка сохранения

Рисунок 5 — макет страницы экземпляра книги (в случае, если она есть в библиотеке)

Автор	Имя автора (можно изменить)	Кнопка возврата книги
Название	Название книги (можно изменить)	
Год выпуска	Год выпуска (можно изменить)	
В наличии	"_"	
Должна быть возвращена	дата	
Кем была взята	Фамилия И.О.	
Кнопка возврата в библиотеку		Кнопка сохранения

Рисунок 6 — макет страницы экземпляра книги (в случае, если книги была арендована)

При нажатии на один из экземпляров книг таблицы (рис. 2) пользователь будет переадресован на страницу этой книги. Информация о книге представляется таблицей, в которой указаны автор, название, год выпуска,

наличие книги в случае, если книга не была арендована и находится в библиотеке (рис. 5), а также датой возврата книги и инициалами пользователя, который ее взял (рис. 6). Если книга в наличии, пользователь может ее взять, заполнив соответствующие поля и нажав на соответствующую кнопку, после чего пользователь может ее вернуть, нажав на кнопку возврата книги в библиотеку.

2) Начальное состояние библиотеки хранится в JSON-файле на сервере. Текущее состояние – в переменной в памяти сервера.

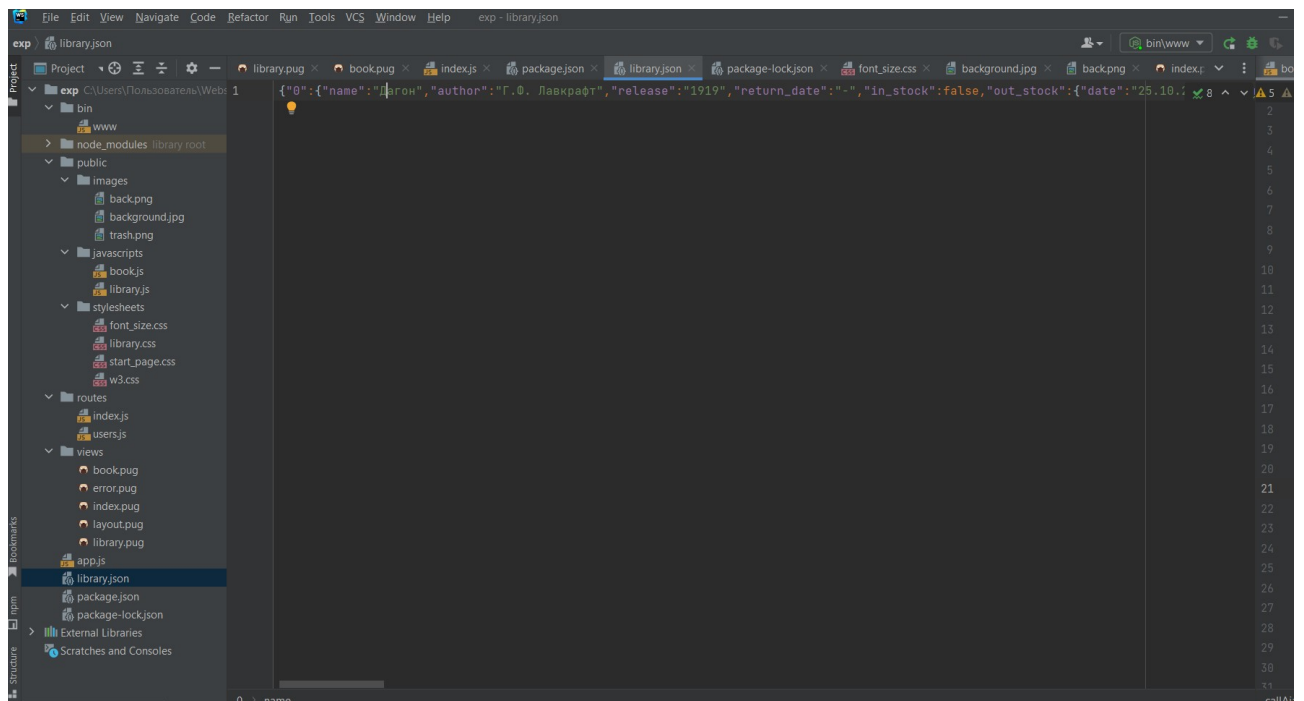


Рисунок 7 — json-файл, в котором хранится начальное состояние библиотеки

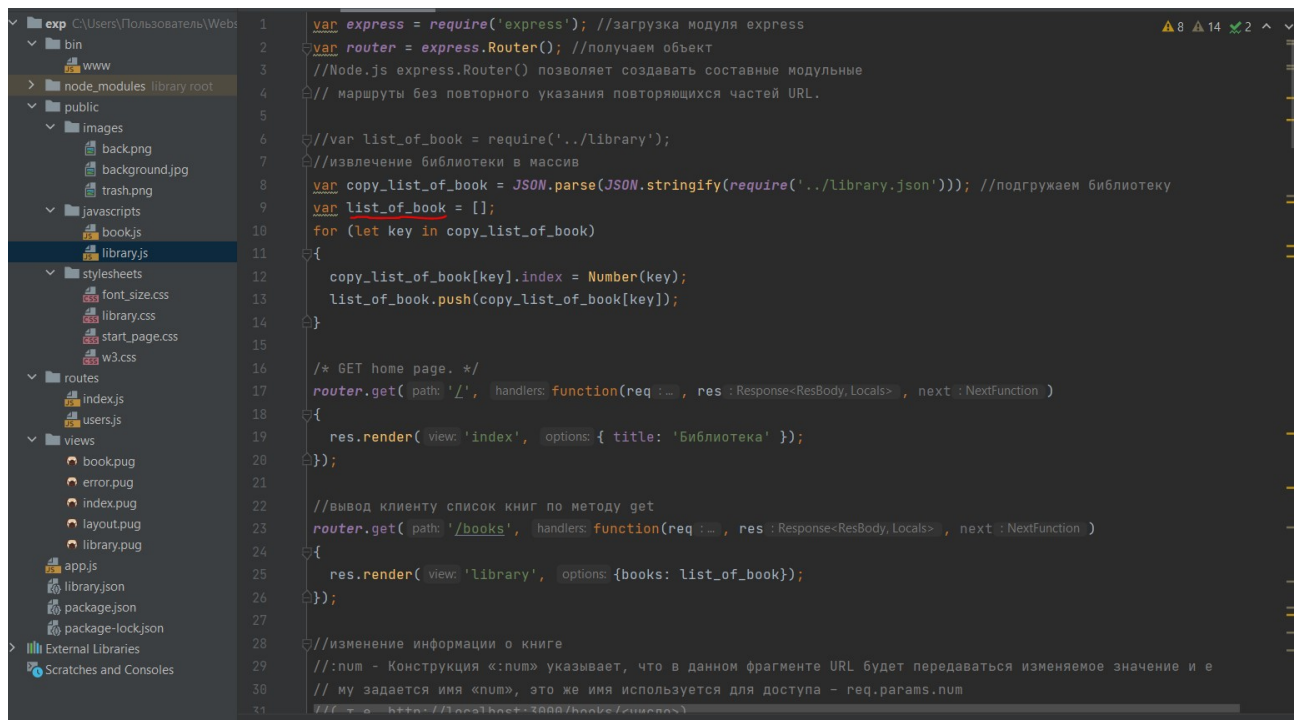


Рисунок 8 — массив `list_of_book`, в которой хранится текущее состояние библиотеки

Начальное состояние библиотеки хранится на сервере в файле «`library.json`», при запуске сервера на нем создается массив `list_of_book`, в который копируется библиотека и к каждому экземпляру добавляется поле индекса, которое отвечает за порядковый номер каждого экземпляра книги. После окончания работы сервера из массива удаляются все индексы, а преобразованная библиотека сохраняется в `json`-файл.

3) В качестве сервера используется Node.JS с модулем `express` (рис.9).

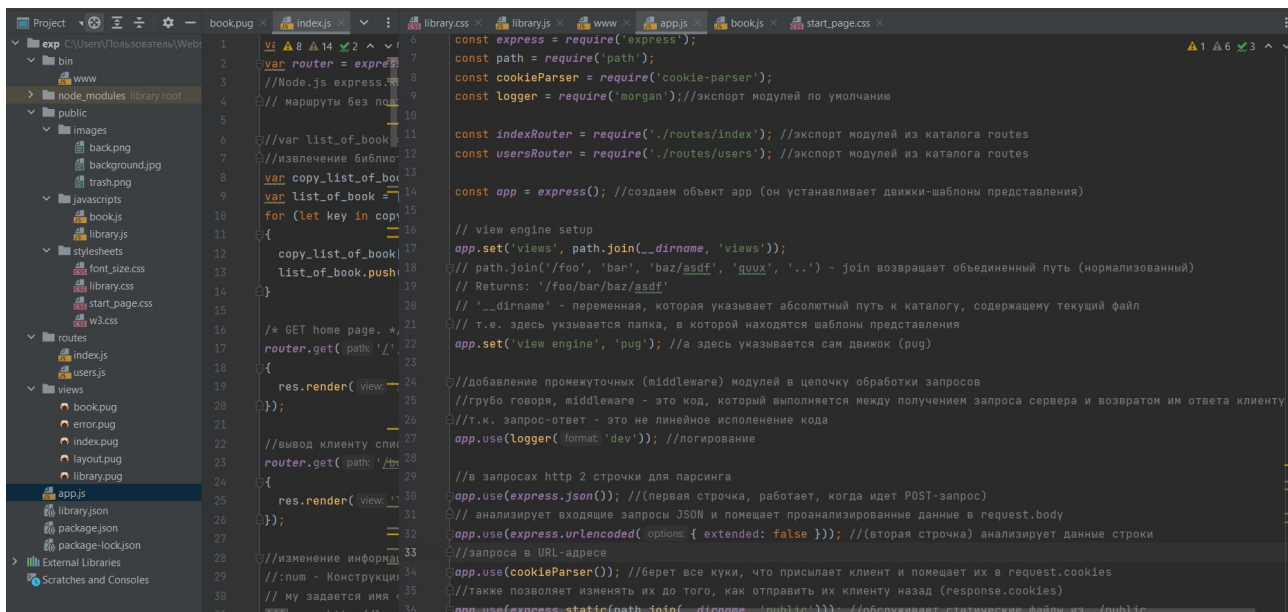


Рисунок 9 — Node.js с модулем express

4) В качестве модуля управления шаблонами HTML-страниц используется pug, все web-страницы сделаны с использованием pug (рис. 10).

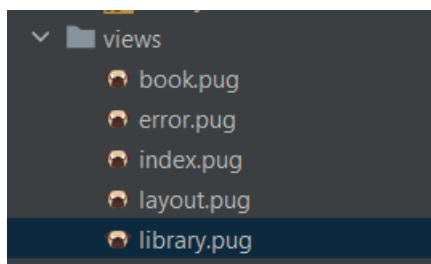


Рисунок 10 — pug-файлы проекта

5) Предусмотрена страница для списка книг, в списке предусмотрена фильтрация по дате возврата и признаку «в наличии», предусмотрена возможность добавления и удаления книг. Удаление книг – с подтверждением.

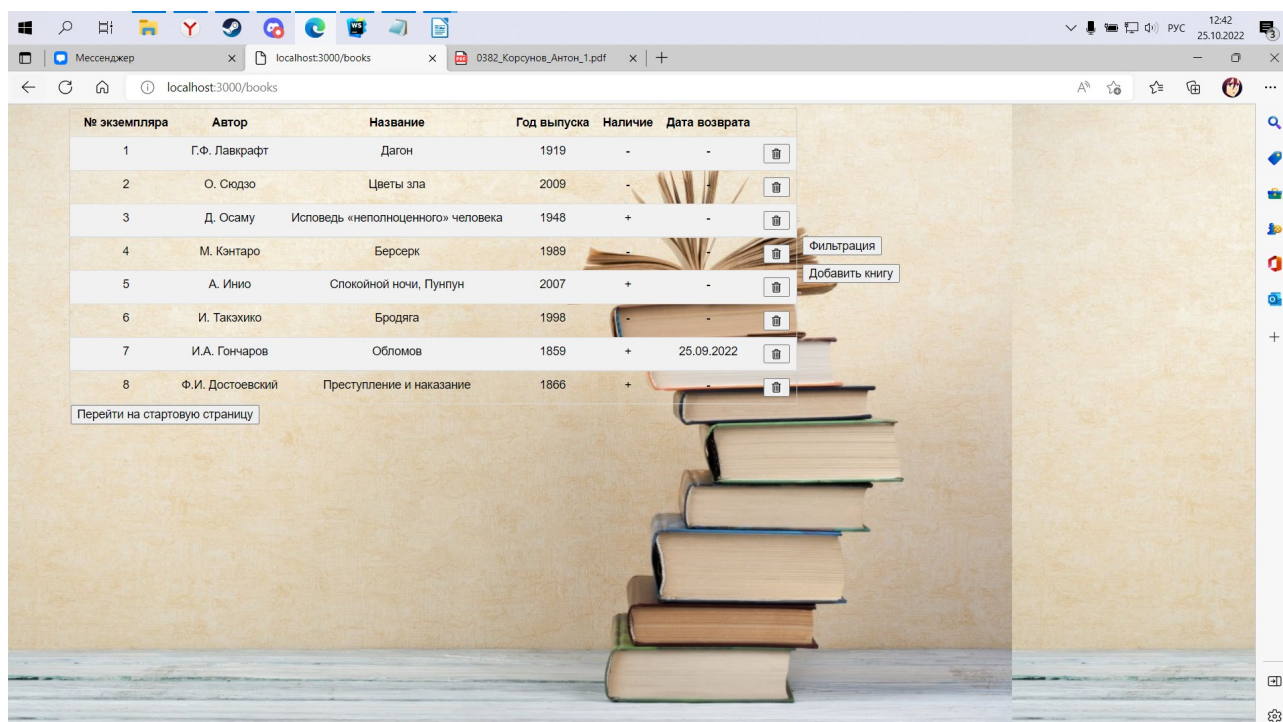


Рисунок 11 — страница библиотеки (список книг)

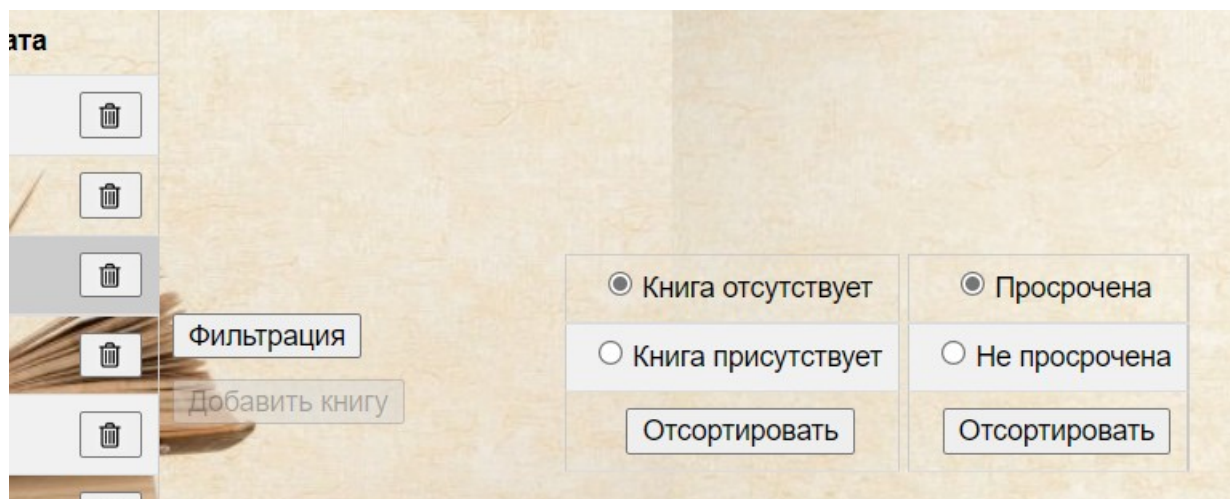


Рисунок 12 — интерфейс фильтрации

№ экземпляра	Автор	Название	Год выпуска	Наличие	Дата возврата	
3	Д. Осаму	Исповедь «неполноценного» человека	1948	+	-	
5	А. Инио	Спокойной ночи, Пунпун	2007	+	-	
7	И.А. Гончаров	Обломов	1859	+	25.09.2022	
8	Ф.И. Достоевский	Преступление и наказание	1866	+	-	

Перейти на стартовую страницу

Фильтрация
Добавить книгу

Рисунок 13 — фильтрация по признаку «Книга присутствует»

№ экземпляра	Автор	Название	Год выпуска	Наличие	Дата возврата	
1	Г.Ф. Лавкрафт	Дагон	1919	-	-	
2	О. Сюдзо	Цветы зла	2009	-	-	
4	М. Кэнтаро	Берсерк	1989	-	-	
6	И. Такэхико	Бродяга	1998	-	-	

Перейти на стартовую страницу

Фильтрация
Добавить книгу

Рисунок 14 — фильтрация по признаку «Книга отсутствует»

№ экземпляра	Автор	Название	Год выпуска	Наличие	Дата возврата	
4	М. Кэнтаро	Берсерк	1989	-	-	
6	И. Такэхико	Бродяга	1998	-	-	

Перейти на стартовую страницу

Фильтрация
Добавить книгу

Рисунок 15 — фильтрация по признаку «Не просрочена»

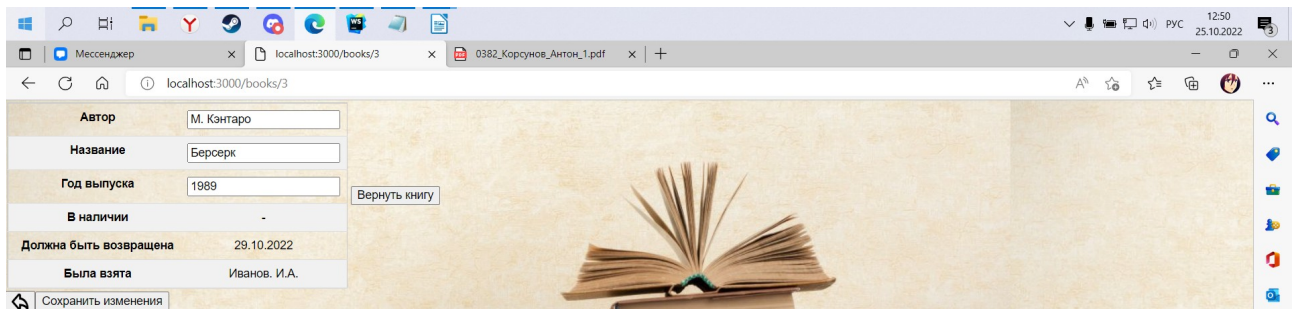


Рисунок 16 — книга не просрочена (дата предполагаемого возврата меньше сегодняшней даты)



Рисунок 17 — фильтрация по признаку «Просрочена»

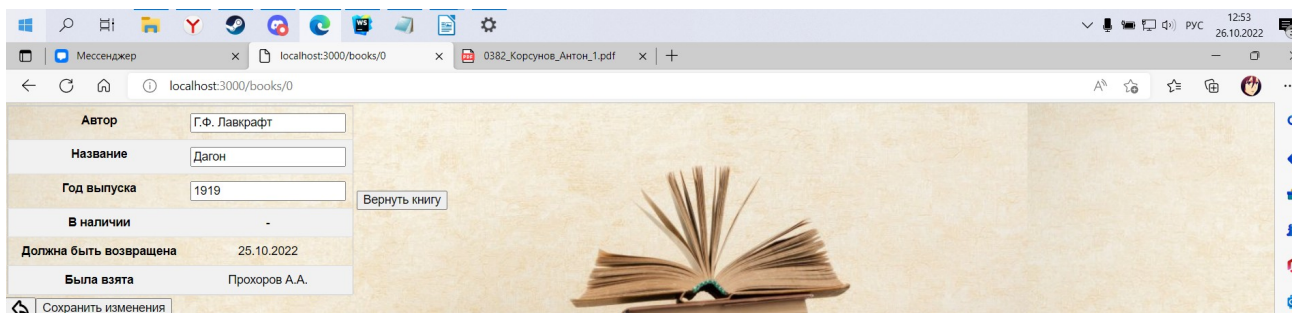


Рисунок 18 — книга просрочена (дата предполагаемого возврата больше сегодняшней даты)

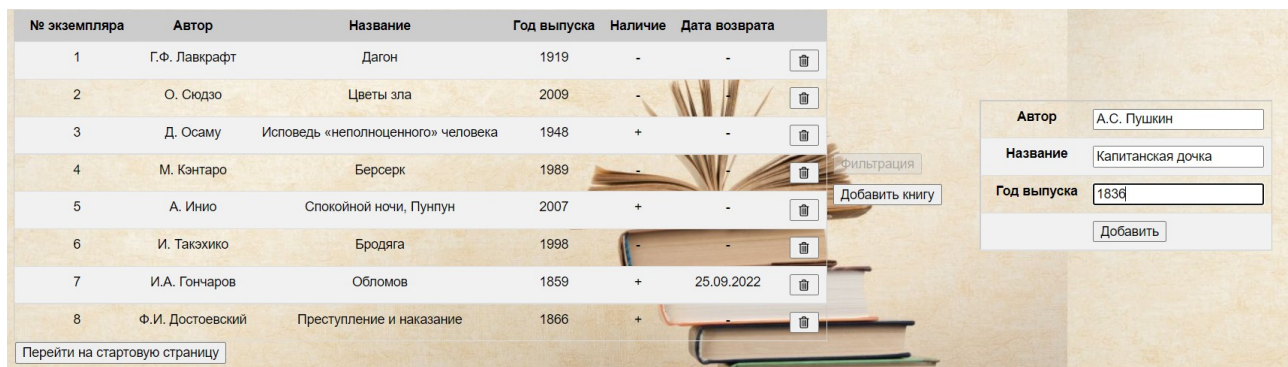


Рисунок 19 — добавление книги

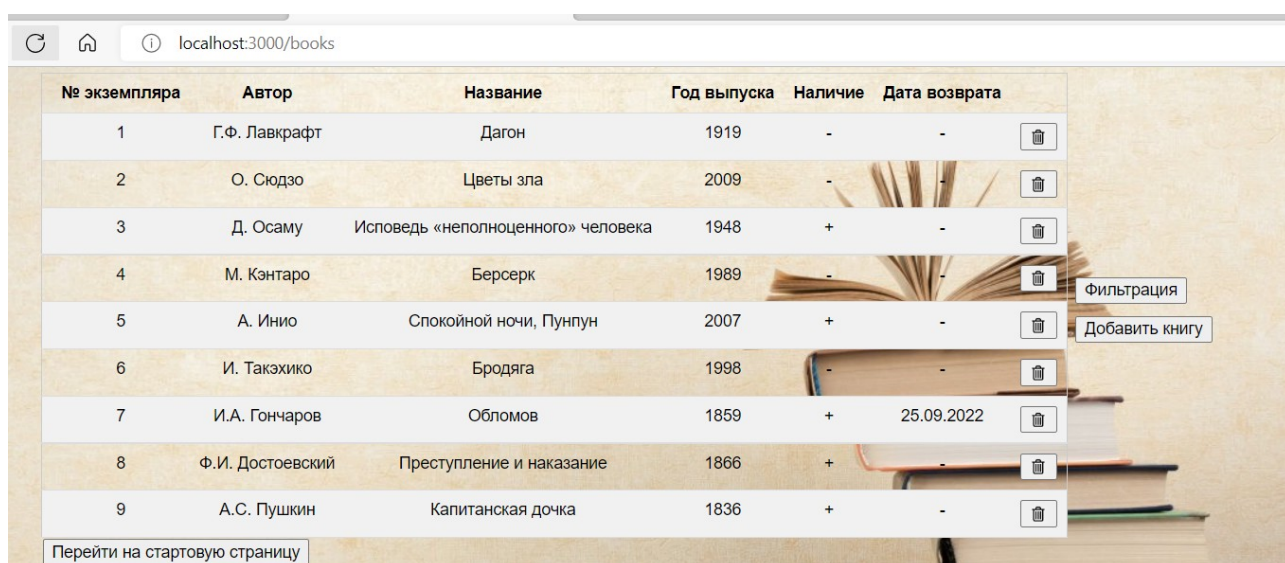


Рисунок 20 — книга добавлена

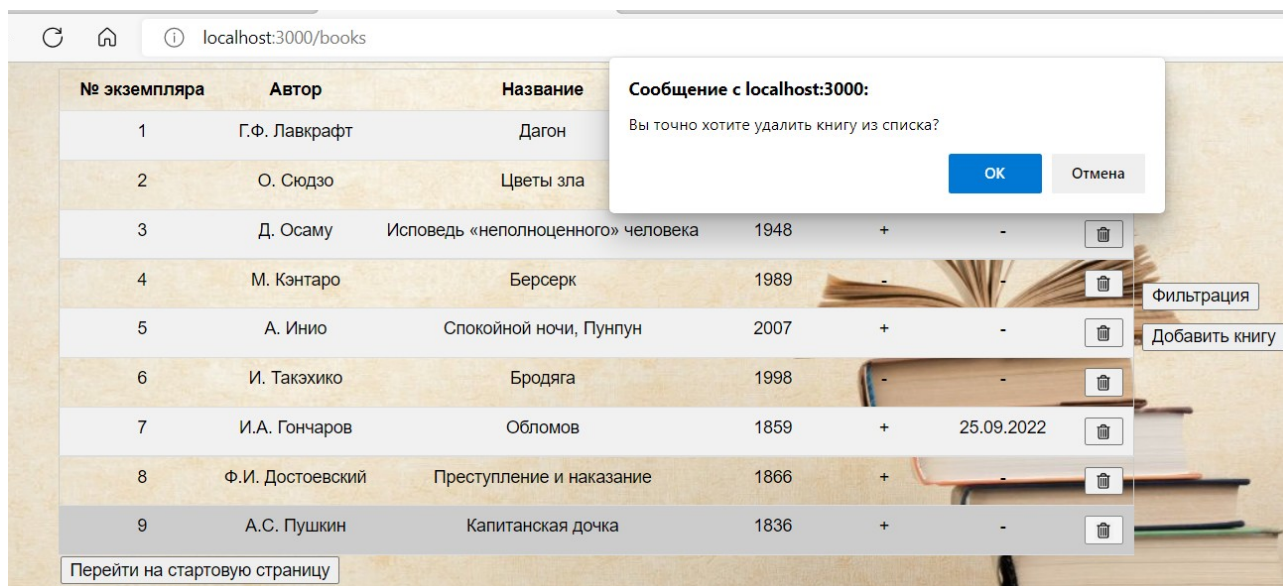


Рисунок 21 — удаление книги с подтверждением



Рисунок 22 — книга удалена

Взаимодействие с интерфейсом программы было описано в пункте 1, возможности продемонстрированы на рисунках 11-22.

6) Предусмотрена страница для карточки книги, в которой ее можно отредактировать (автор, название, дата выпуска) и выдать читателю или

вернуть в библиотеку. В карточке книги есть: находится ли книга в библиотеке, кто ее взял (имя) и когда должен вернуть (дата).

localhost:3000/books/1

Автор	О. Судзо	Взять книгу	
Название	Цветы зла		
Год выпуска	2009		
В наличии	+		
← Сохранить изменения		Кто берет	Иванов И.А.
		Когда вернуть	28.10.2022

Рисунок 23 — страница карточки книги (книга в наличии)

Автор	О. Судзо	Вернуть книгу
Название	Цветы зла	
Год выпуска	2009	
В наличии	-	
Должна быть возвращена	26.10.2022	← Сохранить изменения
Была взята	Корсунов А.А.	

Рисунок 24 — страница карточки книги (книга взята)

Автор
Название
Год выпуска
В наличии

Взять книгу
Кто берет
Когда вернуть

Сохранить изменения

Рисунок 25 — редактирование информации о книге

№ экземпляра	Автор	Название	Год выпуска	Наличие	Дата возврата	
1	Г.Ф. Лавкрафт	Дагон	1919	-	-	
2	О. Сюдзо	Цветы зла	2009	+	25.09.2022	
3	Д. Осаму	Исповедь «неполноценного» человека	1948	+	-	
4	М. Кэнтаро	Берсерк	1989	-	-	
5	А. Инио	Спокойной ночи, Пунпун	2007	+	-	
6	И. Такэхико	Бродяга	1998	-	-	
7	И.А. Гончаров	Обломов	1859	+	25.09.2022	
8	Ф.И. Достоевский	Идиот	1867	+	-	

Фильтрация
Добавить книгу

Перейти на стартовую страницу

Рисунок 26 — обновленная библиотека

Интерфейс карточки книги также описан в пункте 1, возможности продемонстрированы на рисунках 23-26.

7) Информация о читателе вводится с использованием диалогового окна.

Взять книгу
Кто берет
Когда вернуть

Рисунок 27 — окно ввода информации о читателе

8) Оформление страниц выполнено с использованием CSS (допустимо использование w3.css).

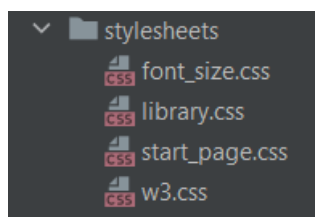


Рисунок 28 — CSS -файлы

9) Взаимодействие между браузером и web-сервером осуществляется с использованием REST.

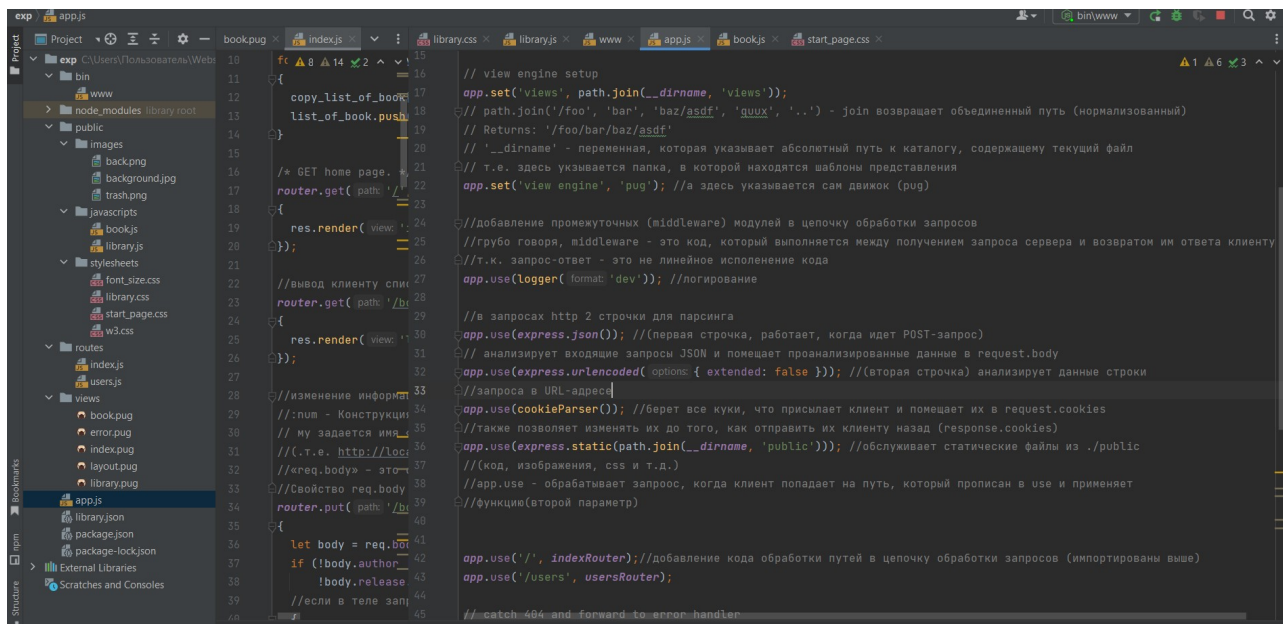


Рисунок 29 — обработка REST-запросов

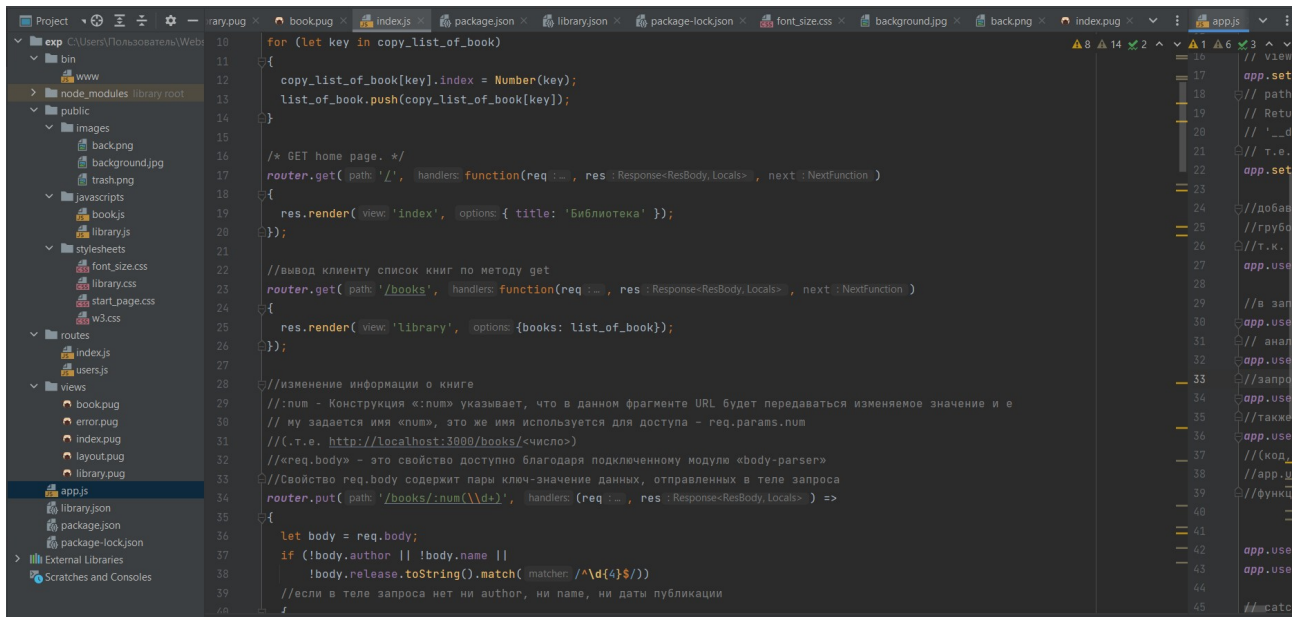


Рисунок 30 — запросы get, post, delete, put

Взаимодействие пользователя с сервером осуществляется с помощью REST-запросов (рисунки 29-30).

10) Фильтрация списка книг осуществляется с использованием AJAX-запросов

```

function book_filter_by_instock()
{
  // в in_stock идет 0, если книга отсутствует, и 1, если присутствует (приведение типов потом превращает в false
или
  //true соответственно
  let in_stock = document.getElementById("element4").checked ?
    document.getElementById("element4").value : document.getElementById("element5").value;

  callAjaxGet('in_stock', in_stock);
}

function book_filter_by_overdue()
{
  // в in_stock идет 0, если книга просрочена, и 1, если не просрочена (приведение типов потом превращает в
false или
  //true соответственно

```

```

let overdue = document.getElementById("element6").checked ?
    document.getElementById("element6").value : document.getElementById("element7").value;

callAjaxGet('overdue', overdue);
}

function callAjaxGet(par, val)
{
    const xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200)
        { //window.location.href - свойство, которое сообщает текущее местоположение URL-адреса
            //браузера. Изменение значения свойства приведет к перенаправлению страницы
            window.location.href=`http://localhost:3000/books/filter/${par}/${val}`;
        }
    };

    //open - инициализирует новый запрос или повторно инициализирует уже созданный
    xhttp.open("GET", `/books/filter/${par}/${val}`, true); //указываем метод get и url, асинхр
    xhttp.send(); //отправляем запрос на получение данных
}

```

Рисунок 31 — фильтрация с использованием AJAX

Фильтрация таблицы происходит с XMLHttpRequest, чтобы запросить данные с сервера (рис. 31). В начале создается объект XMLHttpRequest, в котором инициализируется новый запрос с методом get и отправляется на сервер.

11) Логика приложения была написана на JS

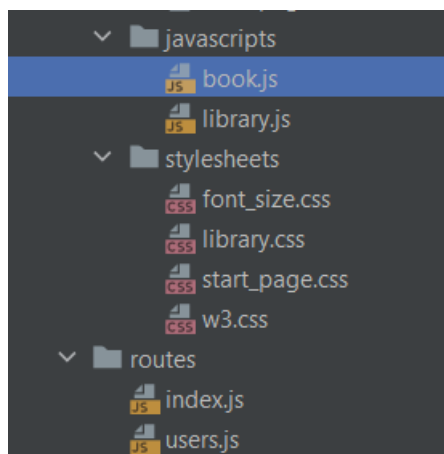


Рисунок 32 — логика приложения

Файл `library.js` (рис.32) содержит скрипты для обработки действий пользователя на страницу библиотеки (соответствует `library.pug`)

Файл `book.js` содержит скрипты для обработки действий пользователя на страницу карточки экземпляра книги (соответствует `book.pug`)

Выводы.

Были изучены взаимодействия клиентского приложения с серверной частью, освоение шаблонов web-страниц, формирование навыков разработки динамических HTML-страниц, освоение принципов построения приложений с насыщенным интерфейсом пользователя.