**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. 0303 |  | Калмак Д.А. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2022

## **Цель работы.**

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.

## **Задание.**

Необходимо создать web-приложение управления домашней библиотекой, которое предоставляет список книг, их можно отфильтровать по признакам «в наличии», «возврат просрочен», есть возможность выдать книгу для чтения и вернуть книгу. Основные требования следующие:

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

2. В качестве сервера используется Node.JS с модулем express.

3. В качестве модуля управления шаблонами HTML-страниц используется pug либо ejs, все web-страницы должны быть сделаны с использованием pug либо ejs.

4. 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.

5. Предусмотрена страница для карточки книги, в которой ее можно отредактировать (минимум: автор, название, дата выпуска) и выдать читателю или вернуть в библиотеку. В карточке книги должно быть очевидно: находится ли книга в библиотеке, кто ее взял (имя) и когда должен вернуть (дата).

6. Информация о читателе вводится с использованием всплывающего модального диалогового окна ().

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

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

9. Фильтрация списка книг осуществляется с использованием AJAX-запросов.

10. Логика приложения реализована на языке JavaScript.

11. Для всех страниц web-приложения разработан макет интерфейса с использованием Figma (<https://www.figma.com/>).

12. При оформлении элементов управления используются иконки (например, Font Awesome, https://fontawesome.ru/all-icons/).

## **Выполнение работы.**

Настроен сервер Node.JS с модулем express. Так же используется cookieParser для работы с cookie и express-session для работы с сессией. В качестве модуля управления шаблонами HTML-страниц используется pug, а также все web-страницы были сделаны с использованием pug. Сервер работает на 3000 порту.

Начальное состояние библиотеки хранится в JSON-файле. При входе в библиотеку отображается страница entrance с погружением в библиотеку. На странице есть кнопка, предлагающая посетить библиотеку. Страница представлена на рис. 1.



Рисунок 1 – Вход в библиотеку

Библиотека находится на странице library. На странице можно добавить новую книгу или удалить уже существующую. Добавить страницу с тем же идентификатором не получится, пользователю выводится предупреждение, что такой идентификатор уже задействован. Удаление случайно произойти не может, пользователю необходимо подтвердить удаление книги. Книги можно отсортировать: все, в наличии, просрочены по возврату. Фильтрация происходит с помощью AJAX, что позволяет изменить содержимое страницы не обновляя ее. Страница представлена на рис. 2.

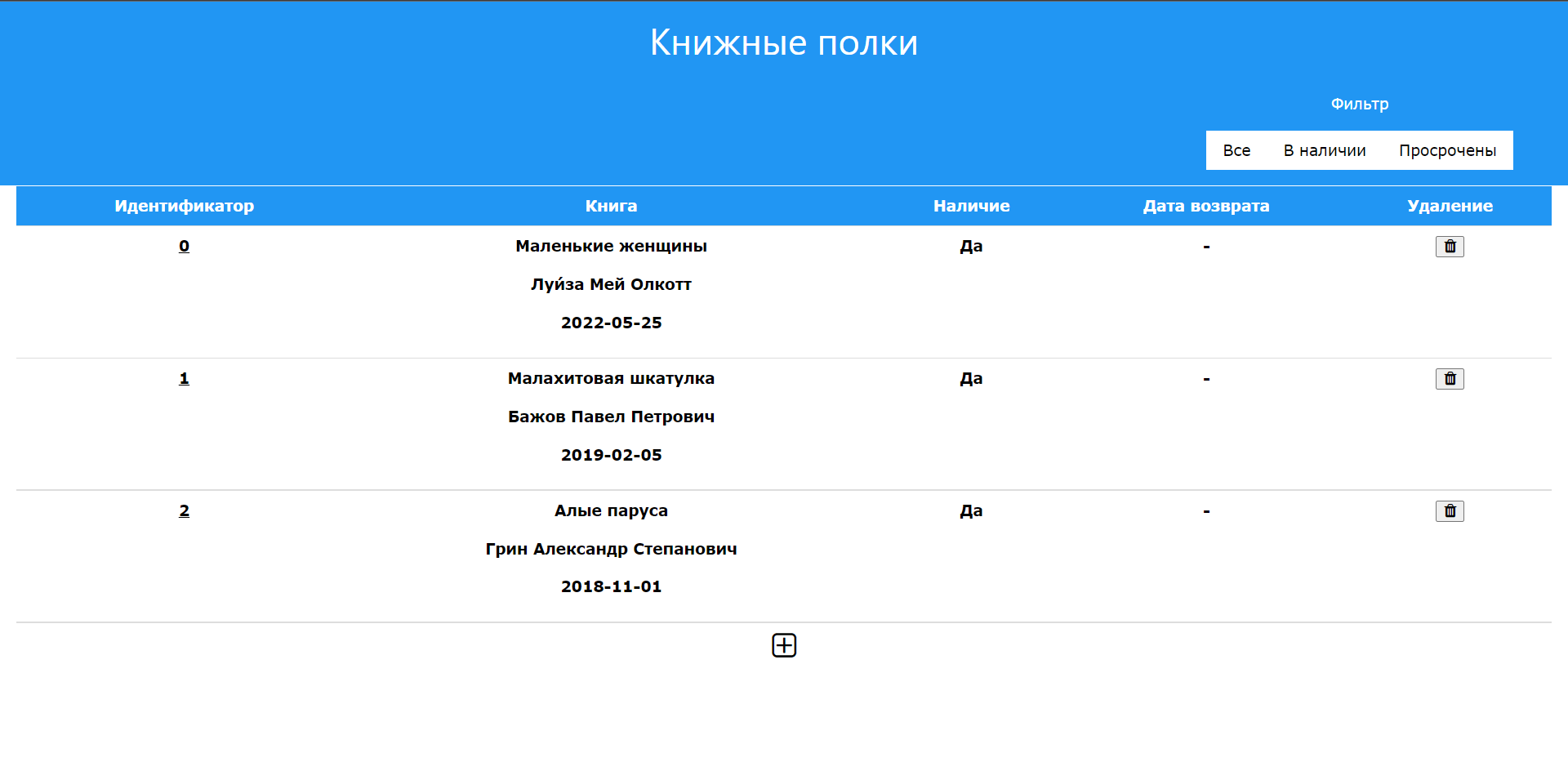


Рисунок 2 – Библиотека

По идентификатору можно перейти на страницу книги. На этой странице полная информация о книге и о ее читателе, если эту книгу взяли. Информацию о книге можно отредактировать. Читатель может взять книгу, заполнив данные о себе и о дате возврата. До взятия кнопка вернуть книгу неактивна. После взятия книги, кнопка взять становится неактивной, а информация о читателе обновляется, а кнопка вернуть становится активной. Нажав на «Читатель» можно получить полную информацию о нем с помощью диалогового окна. Книгу можно вернуть, нажав на кнопку вернуть книгу. После возвращения книги кнопка вернуть книгу становится неактивной, а кнопка взять книгу активной, данные о читателе убираются. Страница представлена на рис. 3. Обрабатывается ошибка отсутствия страницы.

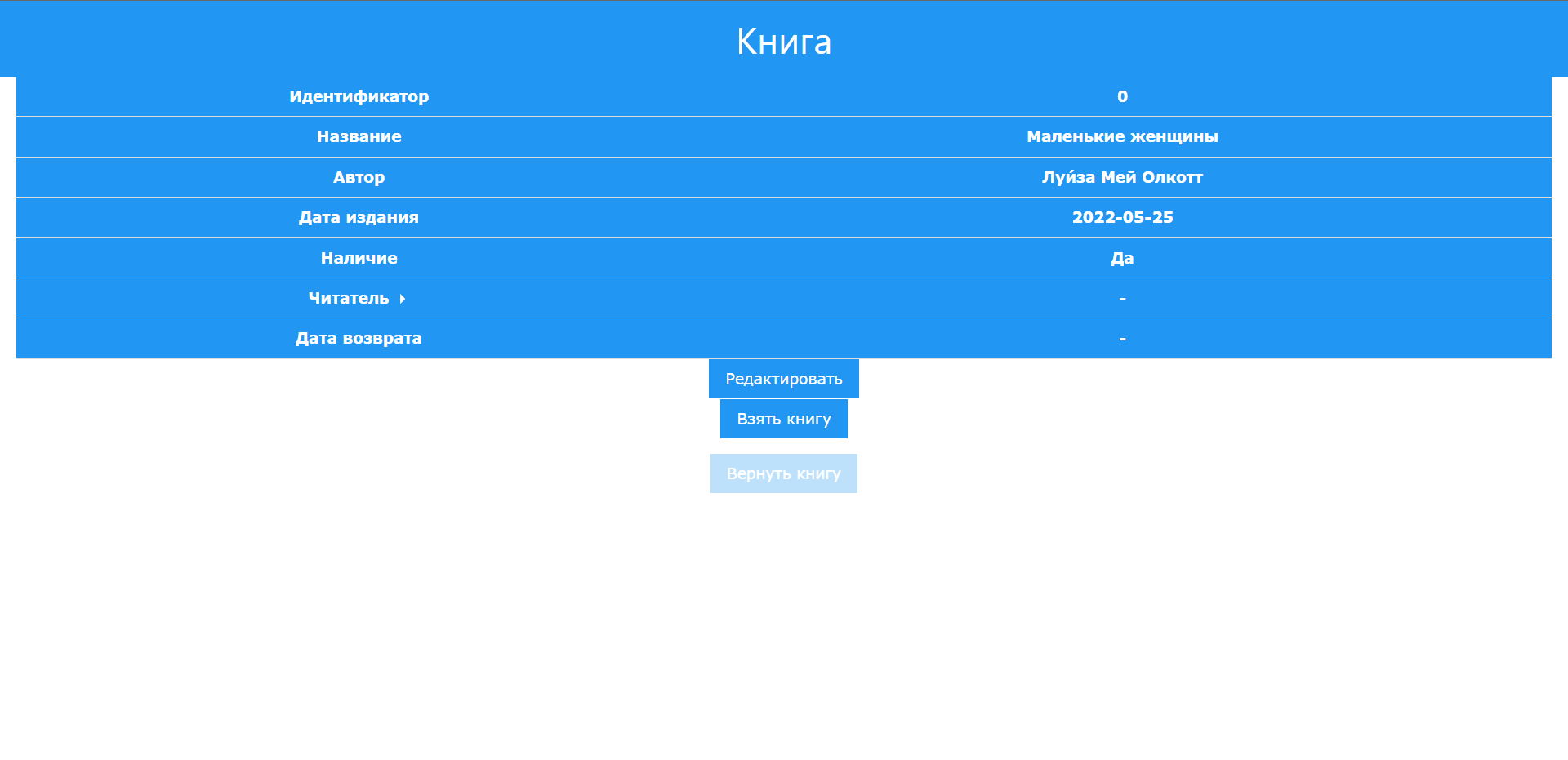


Рисунок 3 – Книга

Оформление страниц выполнено с помощью w3.css, для элементов управления использованы иконки Font Awesome, для всех страниц выполнен макет с использованием Figma. (см. рис. 4-6).

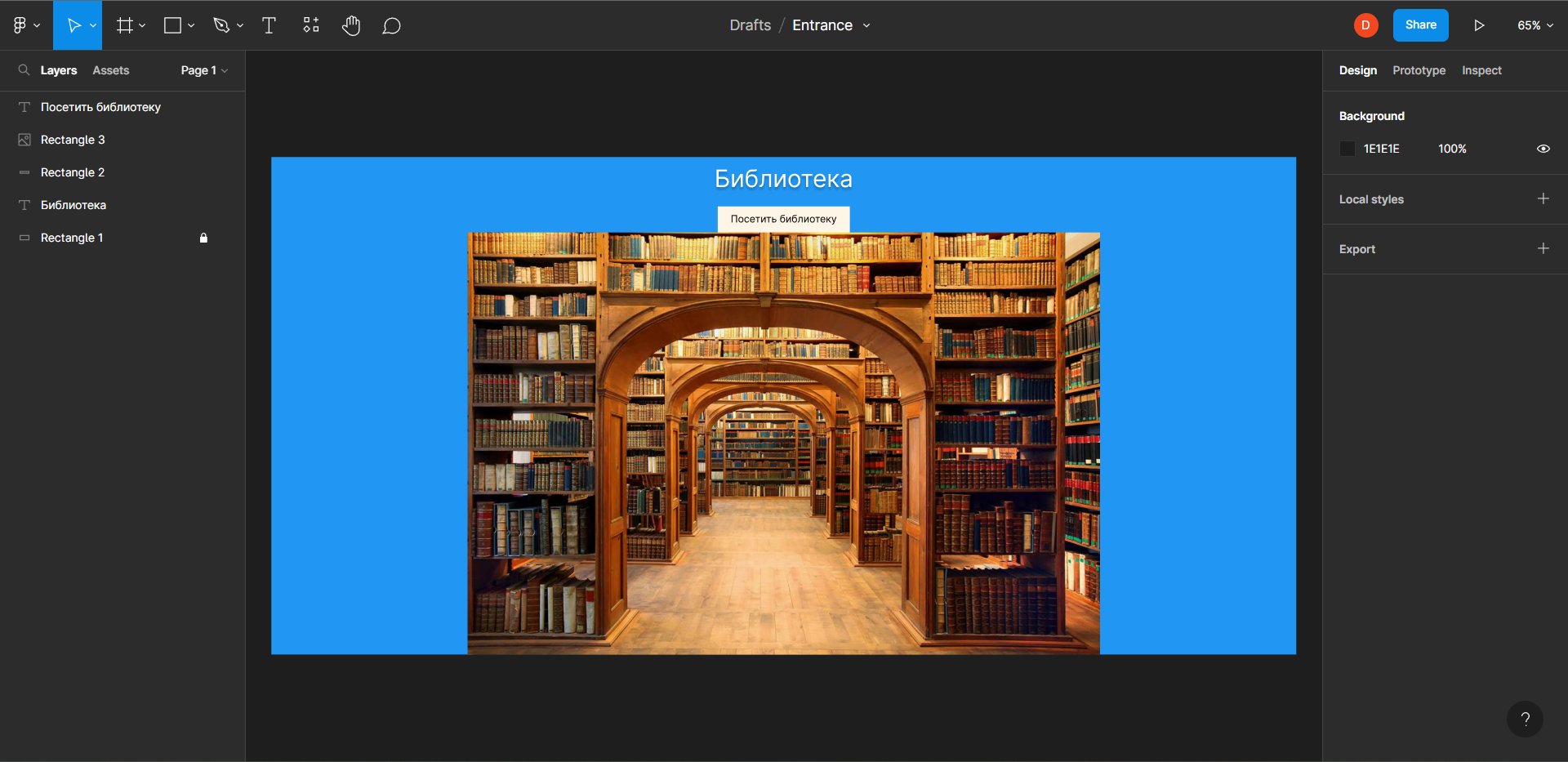


Рисунок 4 – Макет входа в библиотеку

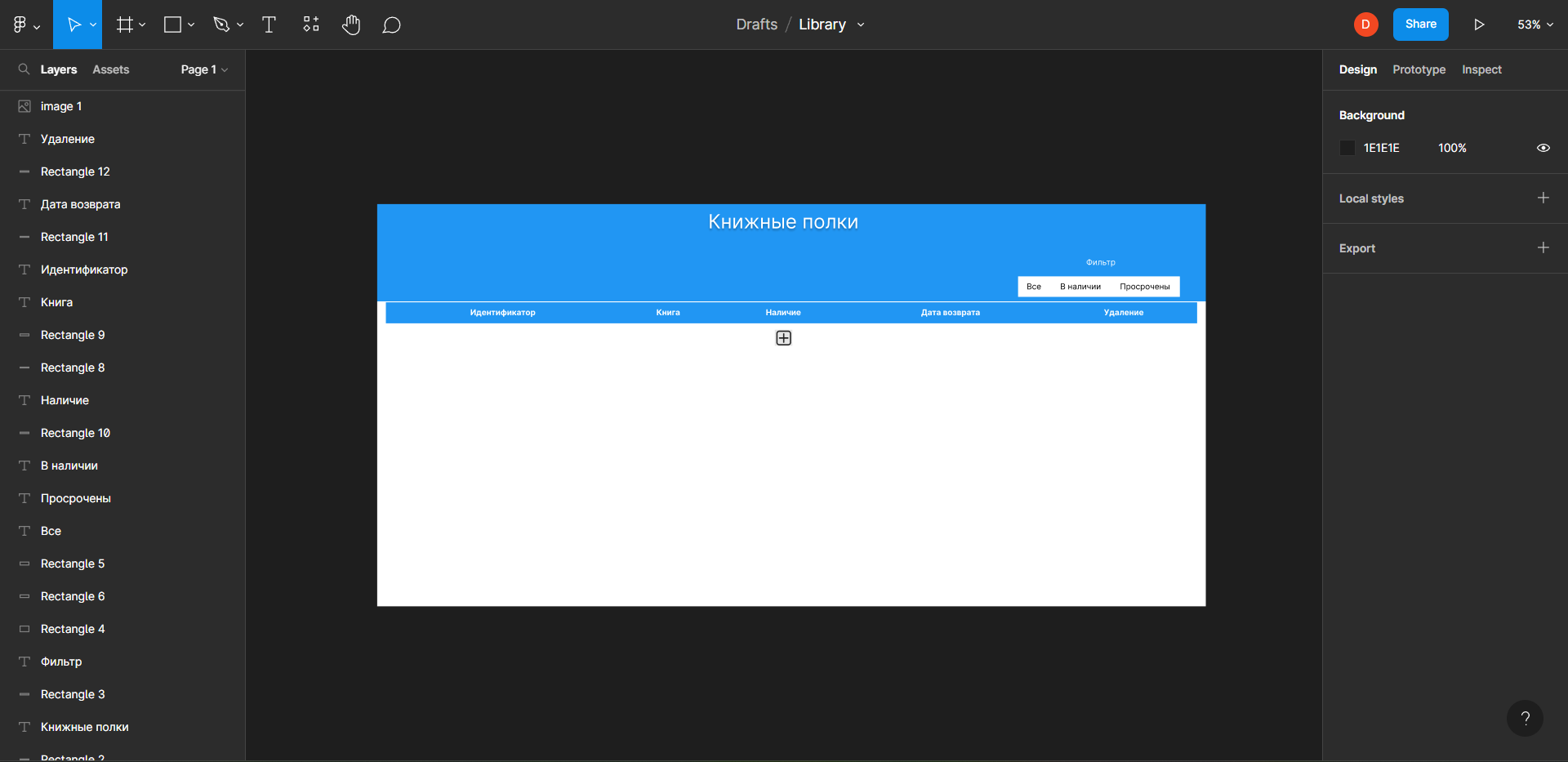


Рисунок 5 – Макет библиотеки

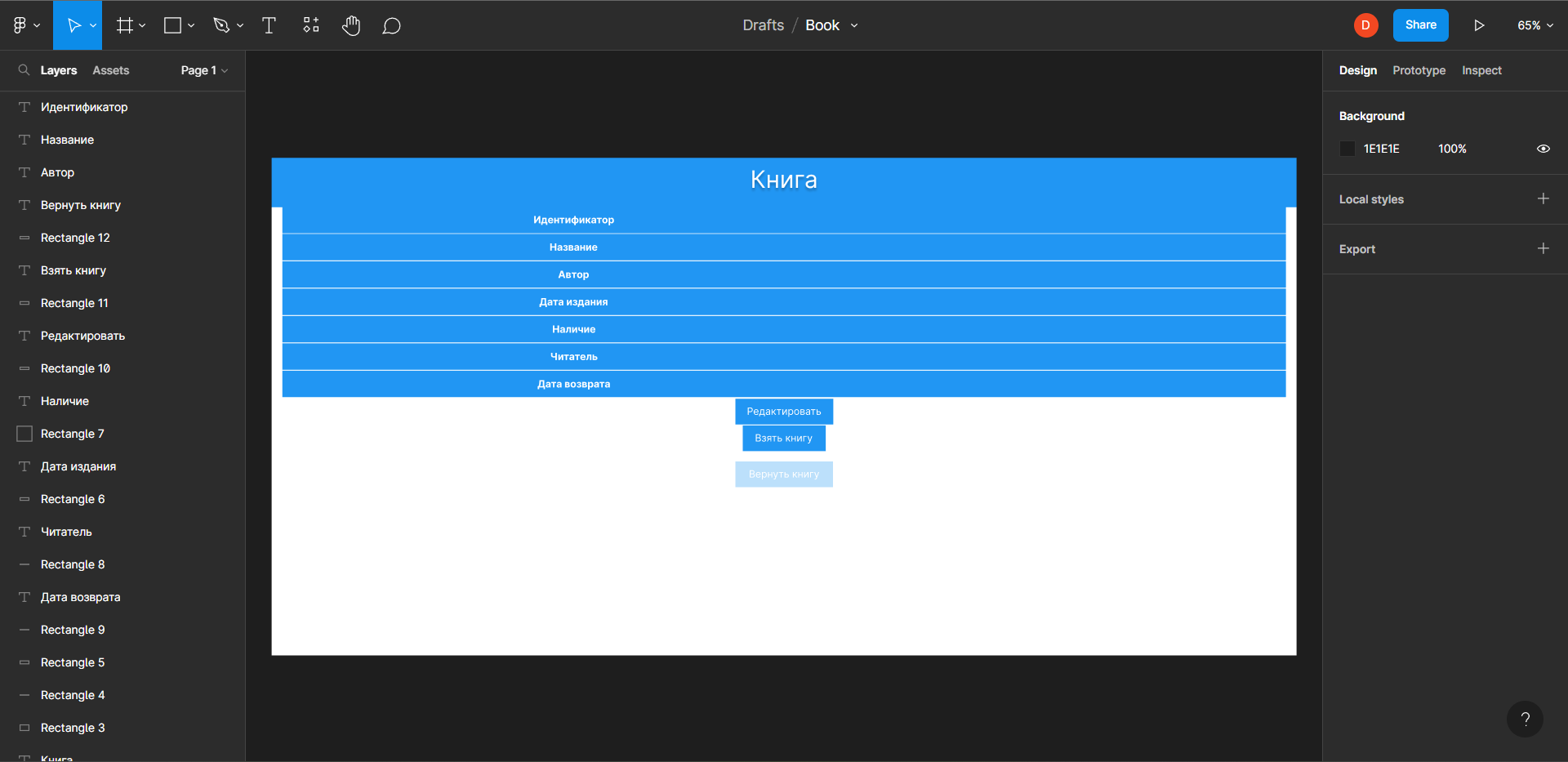


Рисунок 6 – Макет книги

## **Выводы.**

Таким образом, было изучено взаимодействие клиентского приложения с серверной частью, освоены шаблонов web-страниц, сформированы навыки разработки динамических HTML-страниц, освоены принципов построения приложений с насыщенным интерфейсом пользователя. Настроен сервер Node.JS с модулем express. Фильтрация происходит с помощью AJAX. Страницы оформлены с помощью w3.css, элементы управления с помощью Font Awesome, созданы макеты страниц в Figma.