

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
УПРАВЛЕНИЕ БИБЛИОТЕКОЙ

Студент гр. 1303 _____ Коренев Д. А.
Преподаватель _____ Беляев С. А.

Санкт-Петербург
2023

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 (<dialog>).

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/>).

Преимуществом будет создание и использование аутентификации на основе passport.js (<http://www.passportjs.org/>), в качестве примера можно использовать <https://medium.com/devschacht/node-hero-chapter-8-27b74c33a5ce>.

Преимуществом будет реализация загрузки и отображения обложек книг.

Выполнение работы.

Логика сервера:

В файле app.js был подключен фреймворк express и настроена работа сервера. В файле index.js настроена работа маршрутизации сервера, создан экземпляр класса Storage. В классе Storage описана работа и доступ к данным: он имеет доступ к json файлу сервера, в котором обновляет данные, а так же временно хранимые данные экземпляров книг (класс Book) в массиве. Для загрузки изображений использовался multer.

Логика отображения:

Вся разметка страниц сделана с помощью pug. Для главной страницы создан файл main.pug, а основные объекты (например, список книг, функциональные кнопки и пр.) были вынесены в прочие pug файлы. Создан файл стилей library.css, подключены иконки fontawesome. Для

кнопок и прочих объектов были добавлены clickListener'ы. По нажатию на большинство из них открывается диалоговое окно (для написания диалогов так же использовался pug). Действия пользователя обрабатываются и при необходимости на сервер отправляется аякс запрос.

Главный экран представлен на рисунке 1.



Рисунок 1 главный экран

Функционал главного экрана:

- 1) Удаление книги с подтверждением по нажатию на кнопку «Удалить» у соответствующей книги, представлено на рисунке 2.

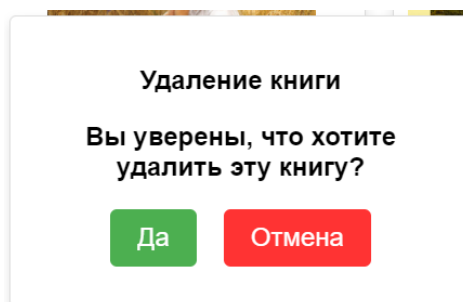


Рисунок 2 удаление книги

- 2) Добавление новой книги с открытием диалога и полями ввода данных книги, представлено на рисунке 3.

Рисунок 3 добавление книги

3) Фильтрация книг по нескольким параметрам, представлено на рисунке 4.

Фильтровать по

Рисунок 4 фильтрация книг

4) Открытие книги (о ней далее)

Экран книги представлен на рисунке 5.

Рисунок 5 страница книги

Функционал данной страницы следующий:

1) Перейти назад – страница всех книг

2) Редактирование книги, аналогичное добавлению книги диалоговое окно, за исключением уже заполненных данных текущей книги.

3) Выдать читателю, диалоговое окно с полями ввода даты возвращения и имени читателя, представлено на рисунке 6.

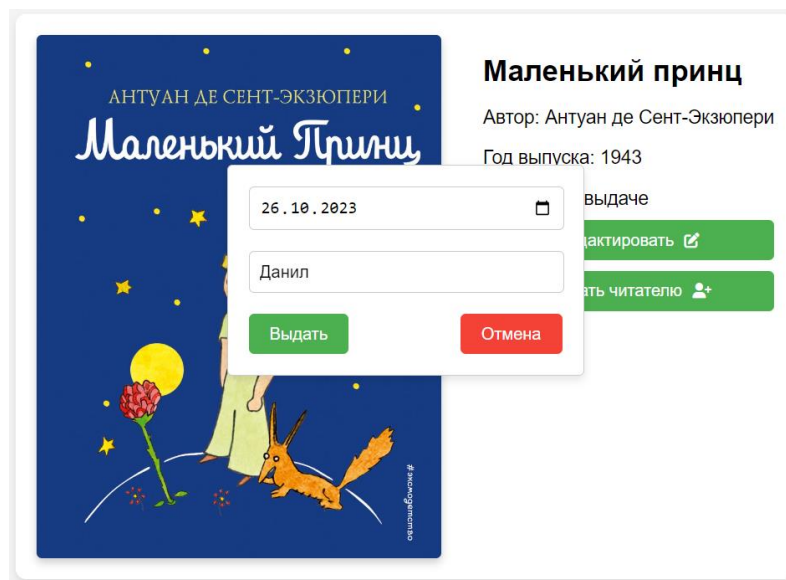


Рисунок 6 выдача книги

4) Если книга была выдана, можно вернуть книгу в библиотеку, представлено на рисунке 7.

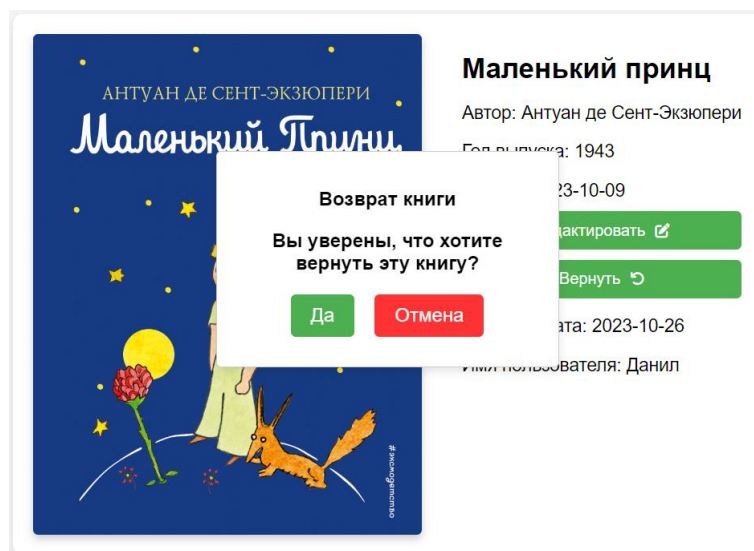


Рисунок 7 возврат книги

Для приведенных страниц были созданы шаблоны в Figma:

На рисунке 8 представлена главная страница.

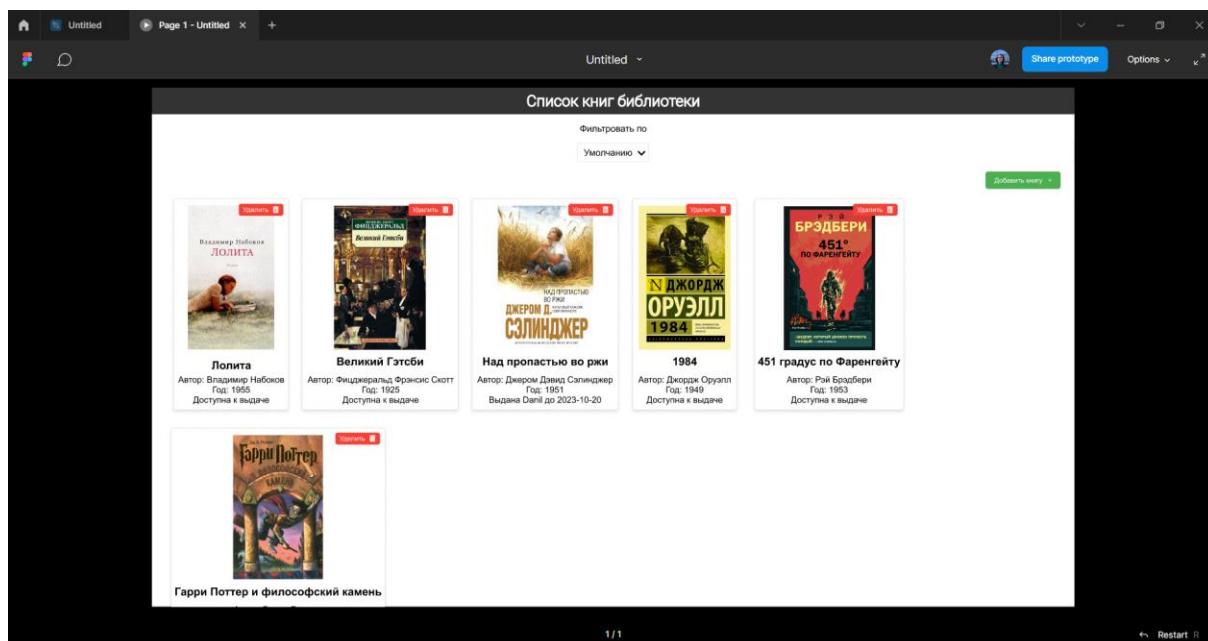


Рисунок 8 шаблон главной страницы в Figma

На рисунке 9 представлена страница книги.

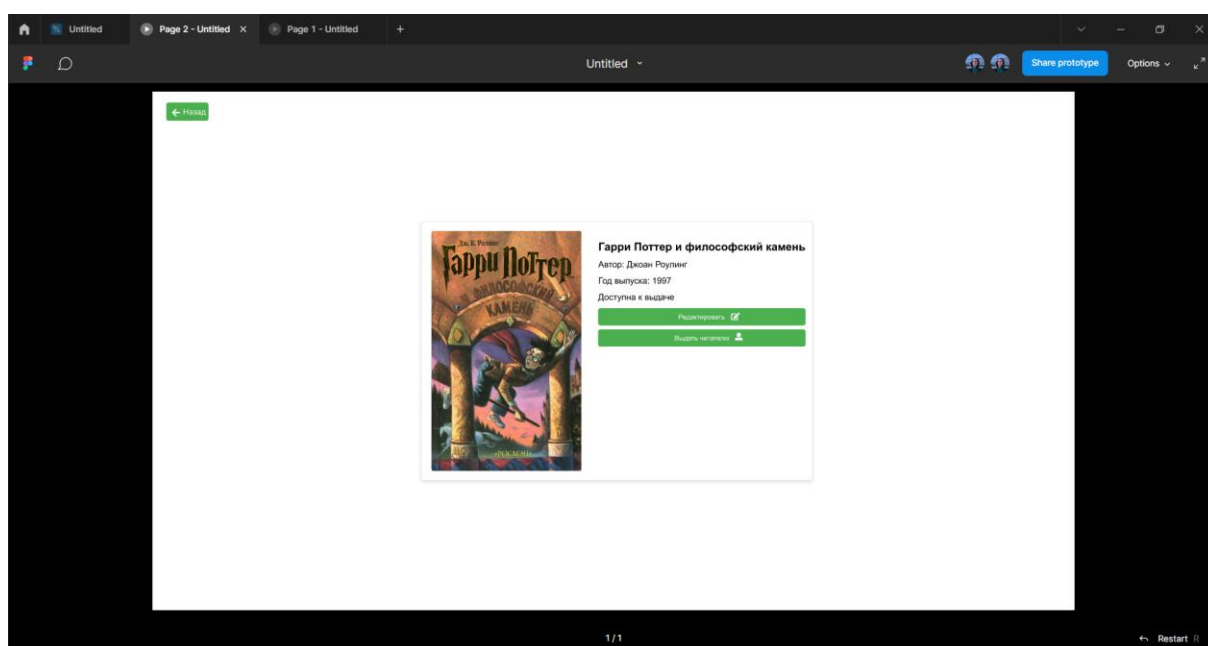


Рисунок 9 шаблон страницы книги в Figma

Вывод.

Изучено взаимодействия клиентского приложения с серверной частью, освоена работа с шаблонами web-страниц и принципы построения приложений с насыщенным интерфейсом пользователя. Написано web-приложение домашней библиотеки с возможностями добавления, удаления, фильтрации, взятия, редактирования и возвращения книг.