

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. 1304

Шаврин А.П.

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.

1. Разработаны макеты страниц web-приложения в Figma

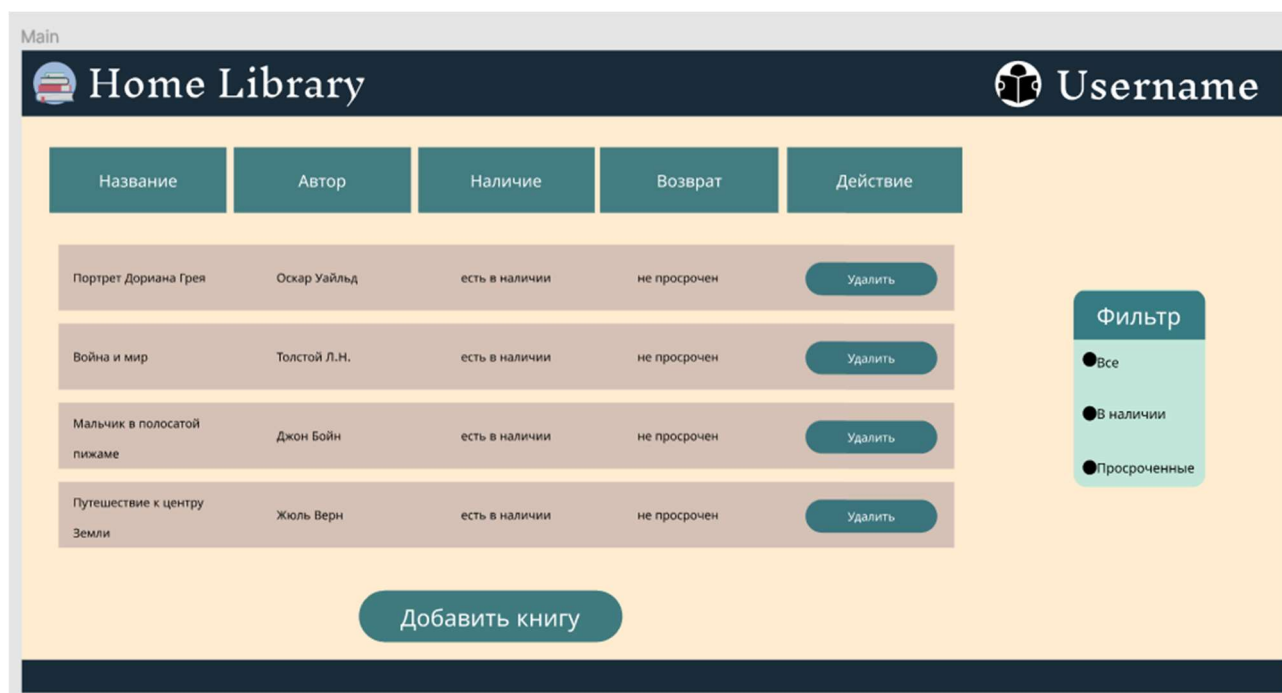


Рис 1. Макет главной страницы.

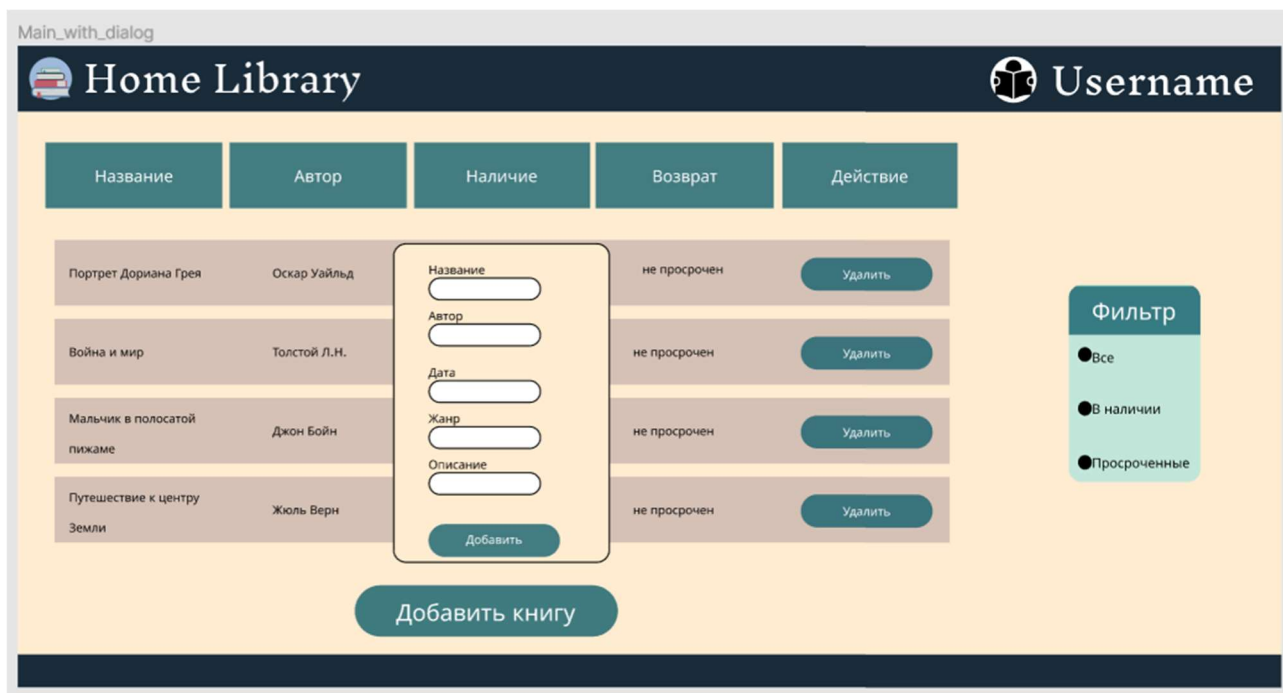


Рис 2. Макет главной страницы с диалоговым окном на добавление книги.



Рис 3. Макет страницы книги в наличии.

Book_card_2

 Home Library

 Username

Название: Портрет Дориана Грея

Автор: Оскар Уайльд

Жанр: Роман

Дата выпуска: 2023-01-18

Читатель: Алексей

Дата возврата: 2023-11-01


Описание: Сюжет романа завязывается, когда талантливый художник Холлуорд пишет портрет прекрасного красавца. Самовлюбленный юноша любит себя на картине ежедневно и вскользь упоминает о своем желании никогда не стареть, мечтая, чтобы вместо него старело его изображение.


Изменить данные

Вернуть книгу

Рис 4. Макет страницы книги не в наличии.

Book_card_change_dialog

 Home Library

 Username

Название: Портрет Дориана Грея

Автор: Оскар Уайльд

Жанр: Роман

Дата выпуска: 2023-01-18

Описание: Сюжет романа завязывается, когда талантливый художник Холлуорд пишет портрет прекрасного красавца. Самовлюбленный юноша любит себя на картине ежедневно и вскользь упоминает о своем желании никогда не стареть, мечтая, чтобы вместо него старело его изображение.

Название

Автор

Дата

Жанр

Описание

Изменить

Изменить данные

Взять книгу

Рис 5. Макет страницы книги с диалоговым окном на изменение книги.

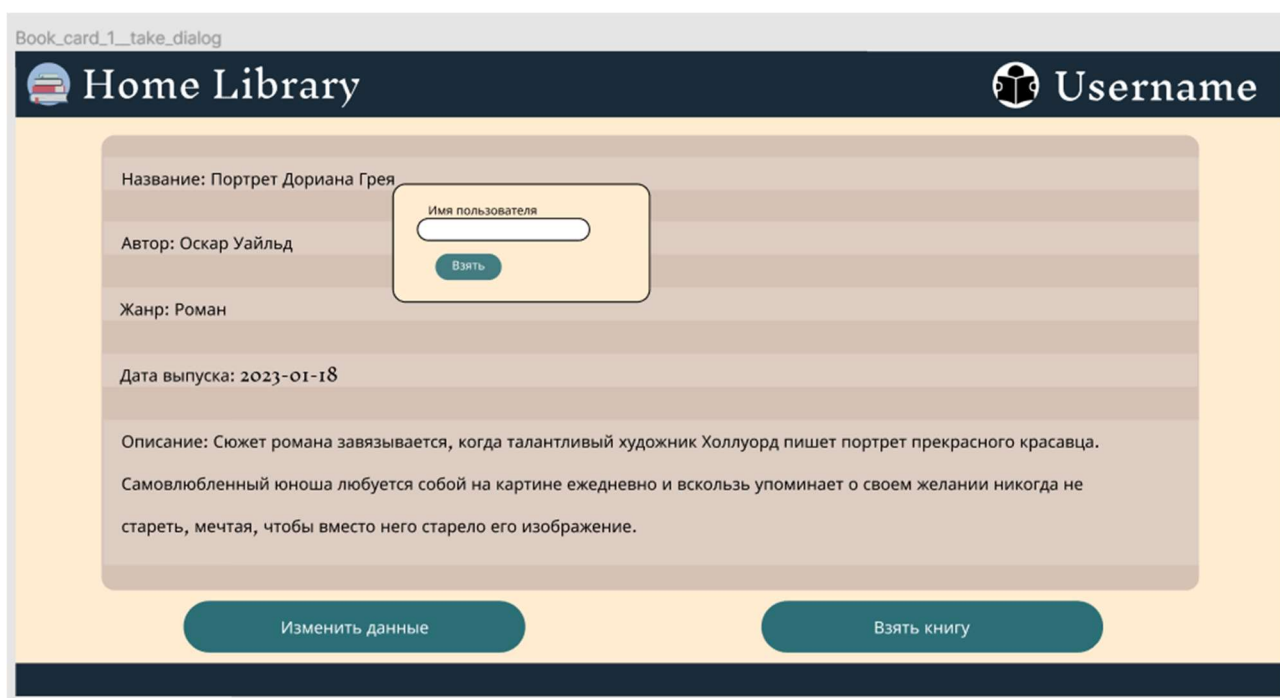


Рис 6. Макет страницы книги с диалоговым окном на взятие книги.

2. Наиболее интересным и важным фрагментом кода у клиентской части приложения является перерисовка таблицы после получения отфильтрованных данных с серверной части приложения после отправки GET-запроса. При получении данных, они парсятся в JSON объект, который средствами JavaScript создает новое наполнение таблицы.

3. Наиболее интересным и важным фрагментом кода у серверной части приложения является выдача книги новому пользователю. В функции `take_book()` происходит проверка корректности имени пользователя и после, в данных книги записывается, что теперь у нее установлена дата возврата и ее нет в наличии у библиотеки, а также пользователь который ее взял.

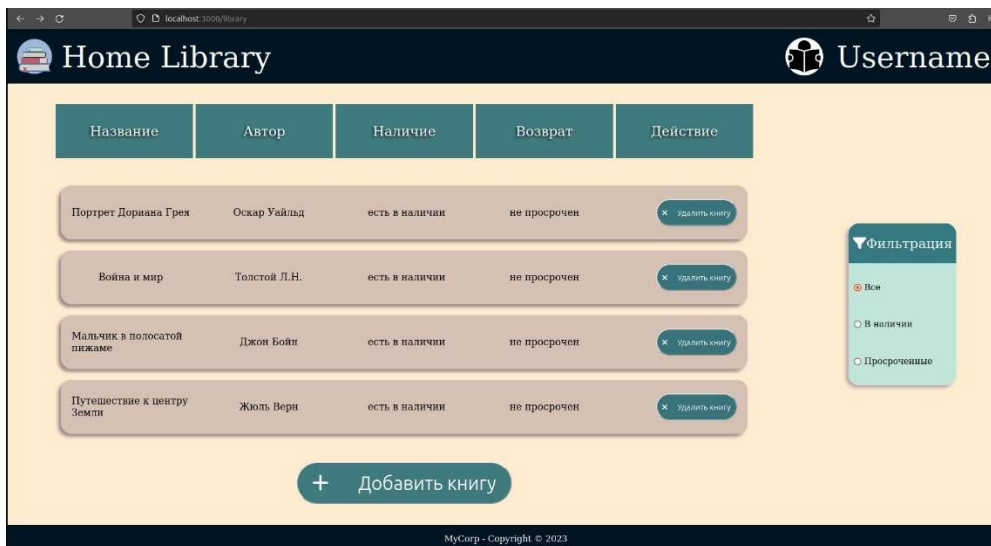


Рис 7. Скрин приложения на главной странице.



Рис 8. Скрин приложения на главной странице при добавлении книги



Рис 9. Скрин приложения на странице книги

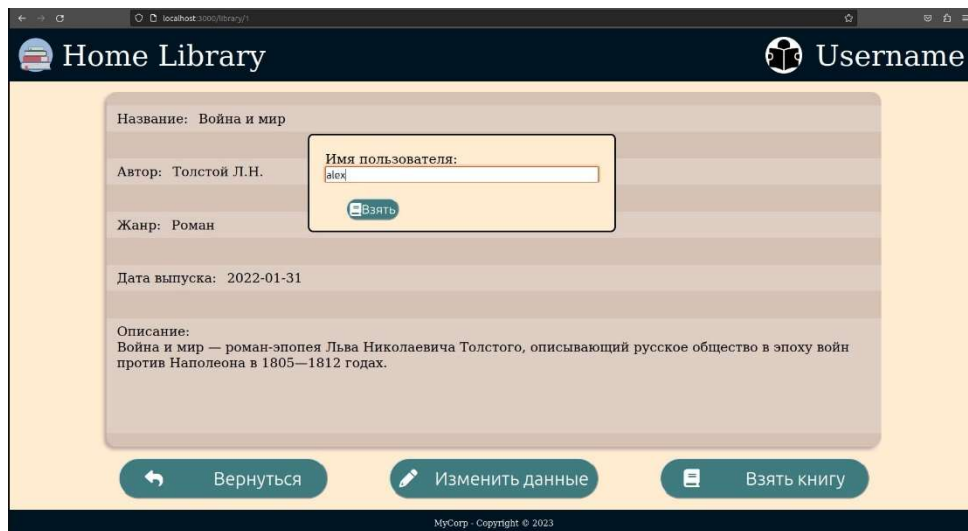


Рис 10. Скрин приложения на странице книги при взятии книги

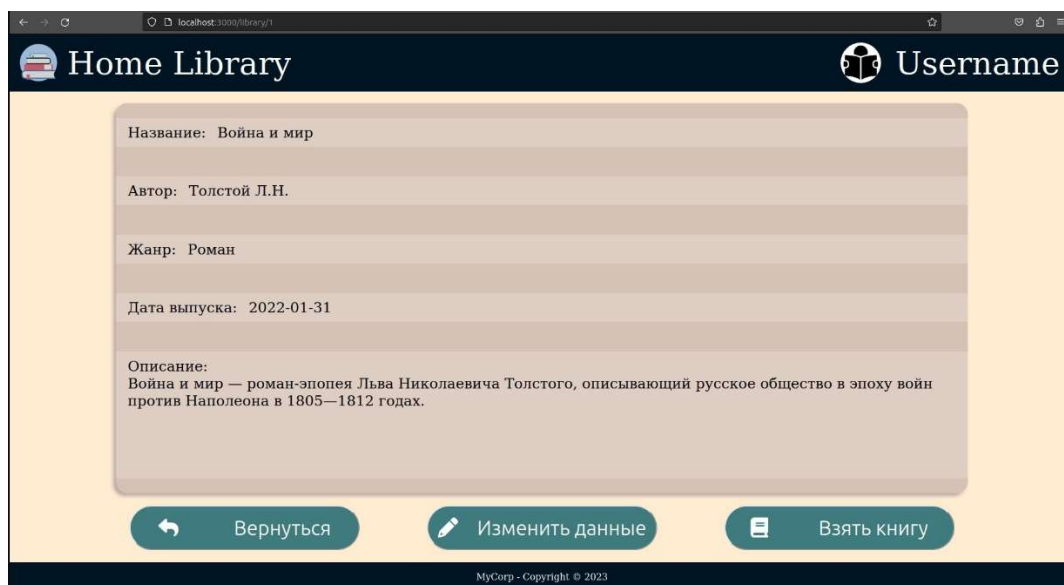


Рис 11. Скрин приложения на странице книги после взятия книги

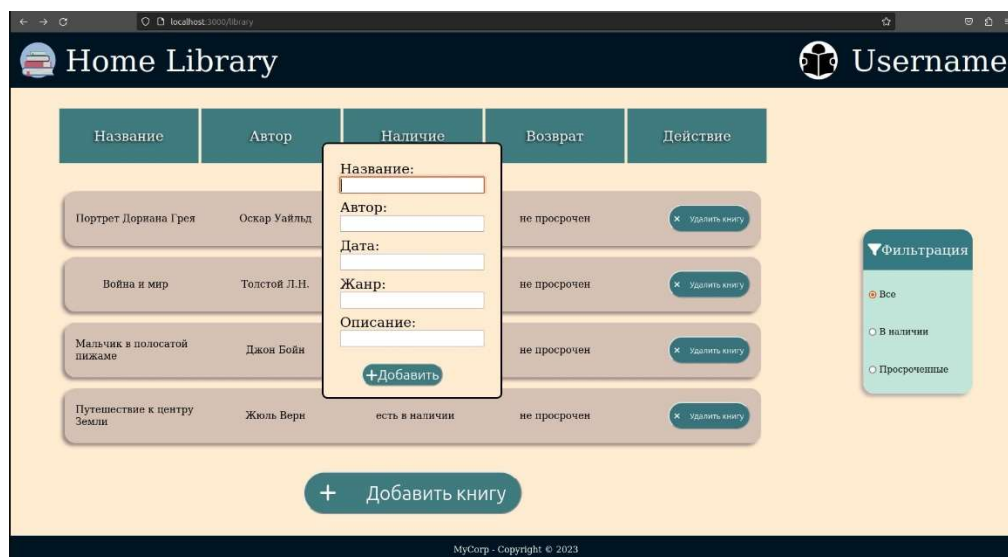


Рис 12. Скрин приложения на странице книги при изменении книги

Выводы.

Исследована, изучена работа взаимодействия клиентского приложения с серверной частью, освоены шаблоны web-страниц, сформированы навыки разработки динамических HTML-страниц, освоены принципы построения приложения с насыщенным интерфейсом пользователя.

Разработана программа, выполняющая поставленные задачи, а именно предоставляющая интерфейс web-приложения, с серверной частью на основе express, создающее шаблоны web-страниц, имеющая настройку маршрутов, json-хранилище и обработку REST-запросов.