

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

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

Студентка гр. 8383

Ишанина Л.Н.

Преподаватель

Беляев С.А.

Санкт-Петербург

2020

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.

Задание.

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

Основные требования:

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.
4. Предусмотрена страница для списка книг, в списке предусмотрена фильтрация по дате возврата и признаку «в наличии», предусмотрена возможность добавления и удаления книг.
5. Предусмотрена страница для карточки книги, в которой её можно отредактировать (минимум: автор, название, дата выпуска) и дать читателю или вернуть в библиотеку. В карточке книги должно быть очевидно: находится ли книга в библиотеке или кто её взял (имя) и когда должен вернуть (дата).
6. Информация о читателе вводится с использованием всплывающего модального 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.

Выполнение работы.

1. Настройка сервера Express, router, стилизация.

Для настройки сервера был создан проект в WebStorm для веб-приложения на Node JS Express. Были продуманы следующие GET запросы для маршрутизации на сайте, реализованные с помощью express-router:

- / - приветственная страница
- /library – страница со списком всех книг, фильтрами, добавлением книги
- /book/:id – страница книги с возможностью добавить/удалить читателя, просмотреть и изменить информацию о книге

В качестве набора стилей был использован Bootstrap 4.

2. Обработка и вывод данных.

Данные о книгах изначально хранятся в JSON файле (lib.json). Сервер загружает данные в память и изменяет уже только в JSON-объекте. Была реализована обработка POST запросов с изменением данных об объектах. Также были реализованы AJAX запросы, с помощью которых осуществляется фильтрация книг в приложении. Макеты страниц и их наполнение были реализованы с помощью Pug.

3. Дополнительная функциональность.

В объекте книги может быть сохранена информация о читателе, если книга была взята. В зависимости от того, есть ли информация о читателе, на

страничку книги выводятся кнопки «добавить читателя» или «удалить читателя».

4. Тестирование.

На рис. 1-3 представлены скриншоты страниц веб-приложения.

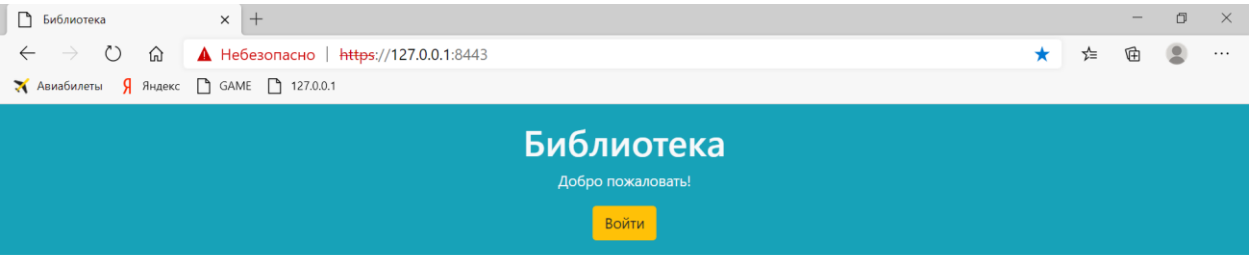


Рисунок 1 – Приветственная страница приложения

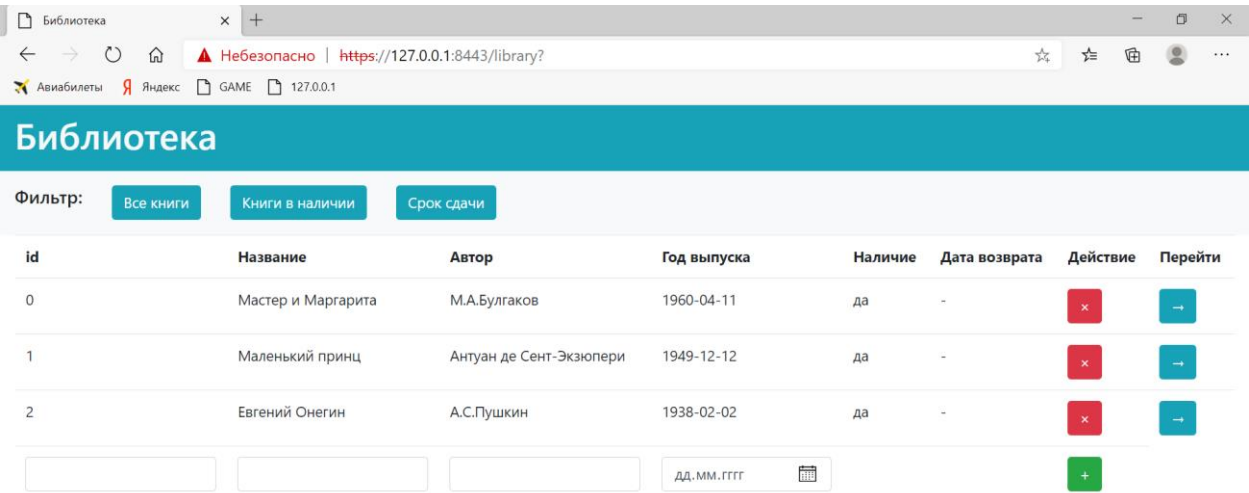


Рисунок 2 – Список всех книг, добавление книги, фильтрация

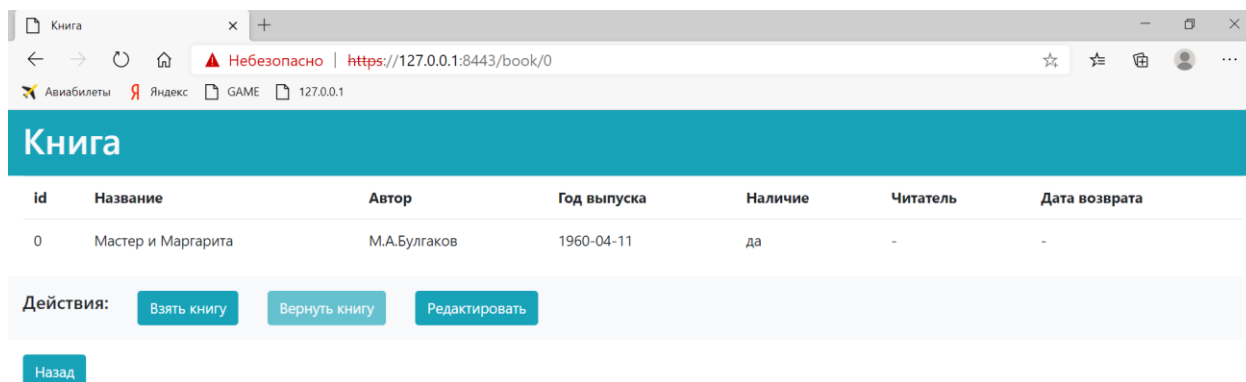


Рисунок 3 – Страница книги

Выводы.

В результате выполнения лабораторной работы 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. Было создано веб-приложение для локальной библиотеки с возможностью осуществлять контроль за читателями книг, фильтрацией, добавлять, удалять и редактировать список книг.