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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Веб-технологии»

Тема: REST-приложение управления библиотекой

Студент гр. 8383 _____ Переверзев Д.Е.

Преподаватель _____ Беляев С. А.

Санкт-Петербург

2020

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.

Основные теоретические сведения

CSS – язык описания внешнего вида документа, написанного с использованием языка разметки, используетсся 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 – модуль, позволяющий использовать шаблоны для HTML-страниц.

REST – 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

Общая формулировка задачи

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 модулем express.

3. В качестве модуля управления шаблонами HTML-страниц используется pug, все web-страницы должны быть сделаны с использованием 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 ее можно от. редактировать (минимум: автор, звание, дата выпуску) и дать читателю вернуть в библиотеку. В карточке книги должно быть очевидно: находится ли книга в библиотеке, кто с взял (имя) и когда должен вернуть (дата).

6. Информация о читателе вводится с использованием всплывающего модульного окна

7. Оформление страниц выполнено с использованием 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,

Ход работы

1. Используя среду разработки vs code, были установлены все необходимые расширения.
2. Используя модуль express, был создан и настроен сервер.
3. Были созданы и настроены pug и css файлы.
4. Разработка интерфейса пользователя.
   1) Страница регистрации через GitHub OAuth представлена на рисунке 1

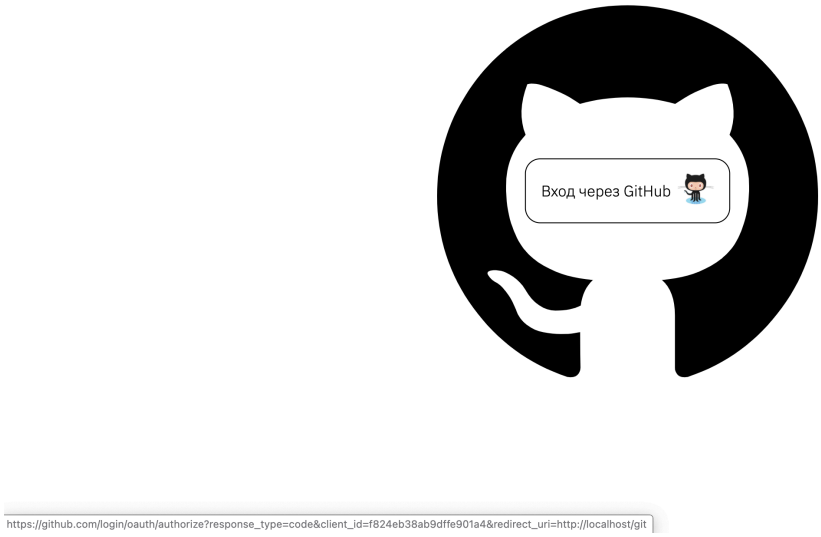

Рисунок 1 – Страница регистрации сайта

   2) Окно библиотеки представлено на рисунке 2.

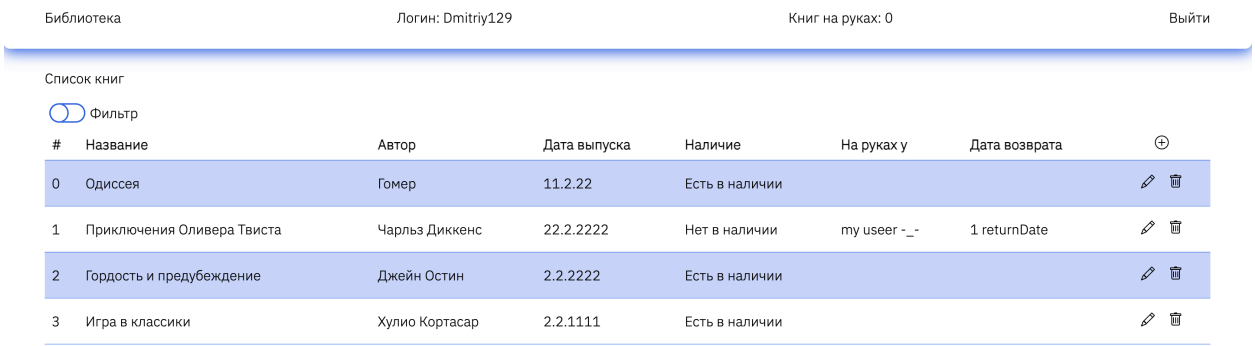

Рисунок 2 – Основная страница

3) Краткая информация о пользователе рисунок 3.

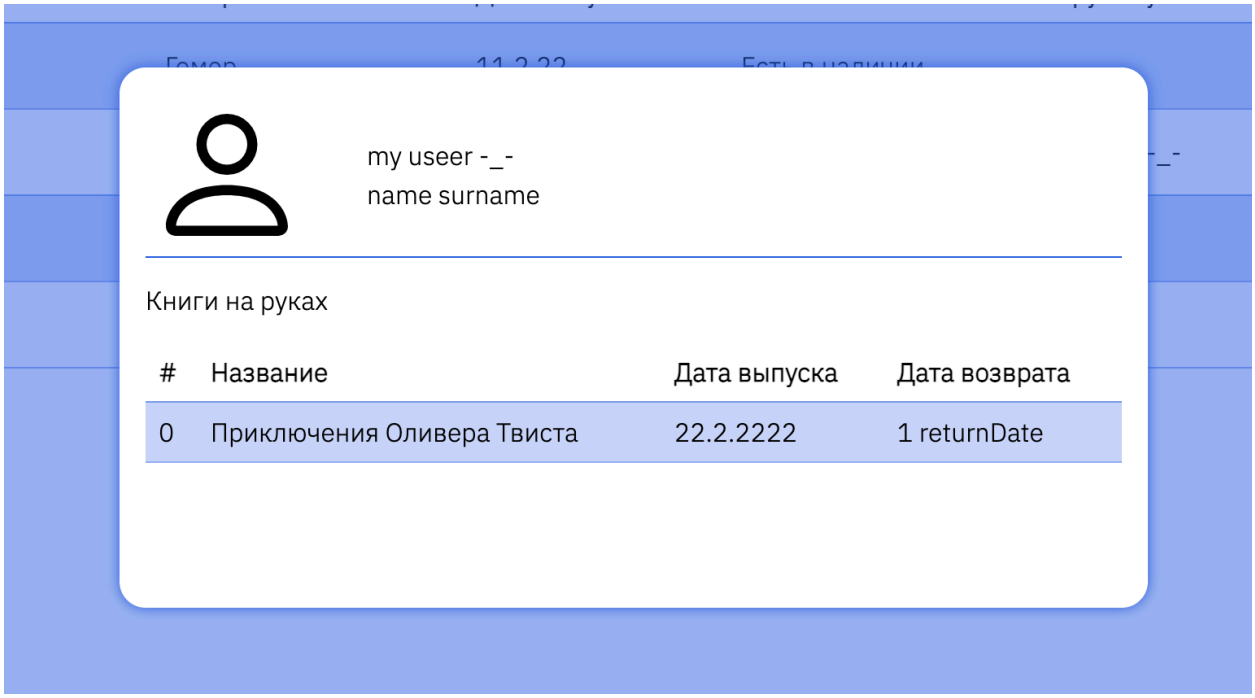

Рисунок 3 – Модальное окно юзера.

4) Информация о книге о рисунок 4.

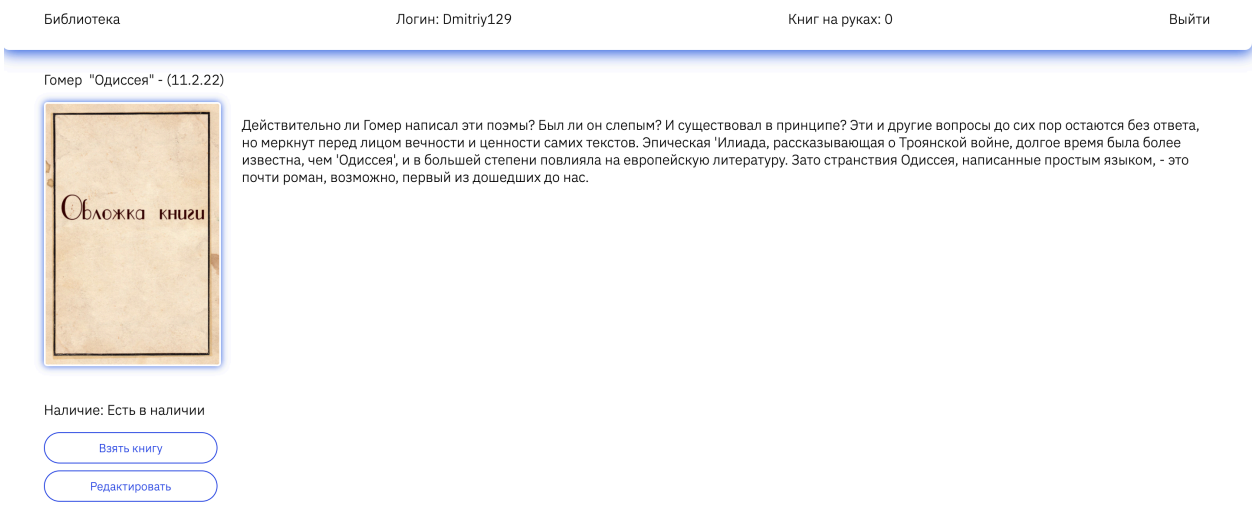

Рисунок 4 – Модальное окно.

5. Для проекта были созданы следующие файлы.
    1) index.js, s/rc/app.js,/src/routes – файлы сервера
    2) /front – файлы html/js/css для клиента
    3) /api - вспомогательные функции

Вывод

В ходе лабораторной работы был получен опыт работы с pug, css файлами, создании сервера на основе модуля express и генерации AJAX-запросов, на основе создания REST-приложения управления библиотекой