**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. 1303 |  | Чубан Д.В. |
| --- | --- | --- |
| Преподаватель |  | Беляев С. А. |

Санкт-Петербург

2023

**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.

Для достижения поставленной цели требуется решить следующие задачи:

– разработка интерфейса web-приложения;

– задание стилей для отображения web-приложения с учётом размера экрана (использование на компьютере, на мобильном телефоне);

– создание web-сервера на основе express;

– создание шаблонов web-страниц;

– настройка маршрутов;

– создание json-хранилища;

– обработка REST-запросов;

**Выполнение работы.**

С помощью Figma был разработан макет приложения.



Рисунок 1 – макет страницы входа.

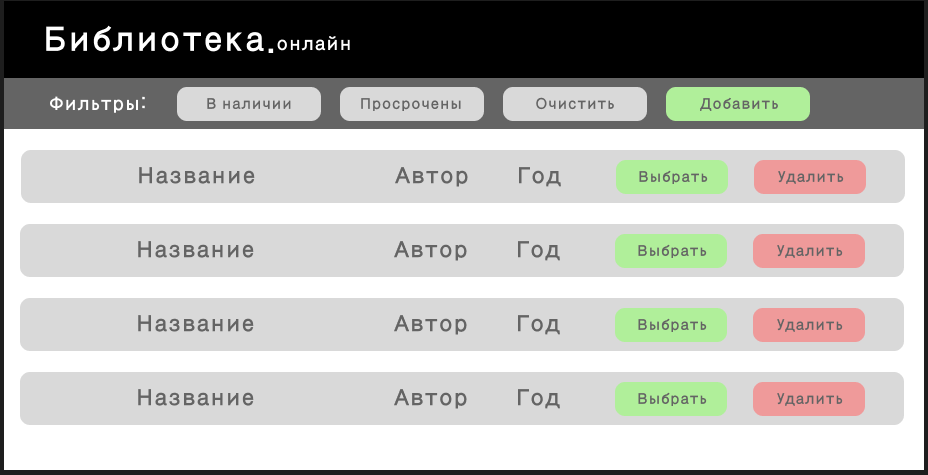


Рисунок 2 – макет главной страницы библиотеки.

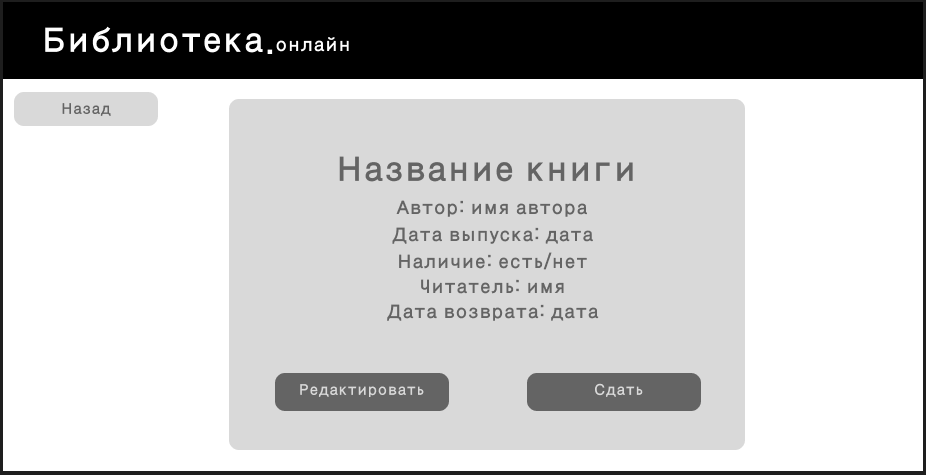


Рисунок 3 – макет страницы книги.

В файле *app.js* описан сервер, на котором работает приложение.

Для навигации между страницами используется файл *router.js.* В нем описаны методы переходов между страницами, добавление, изменение и удаление записей. Также чтение первоначального состояния библиотеки происходит из json-файла.

Внешний вид страниц описан с использованием pug-файлов: *book.pug, library.pug* и *index.pug*. Также для упрощения работы с разметкой страниц используется *w3.css*.

AJAX-запросы, используемые для фильтрации книг в библиотеке описаны в файле *ajax.js* (В наличии, просрочены, очистить фильтры).

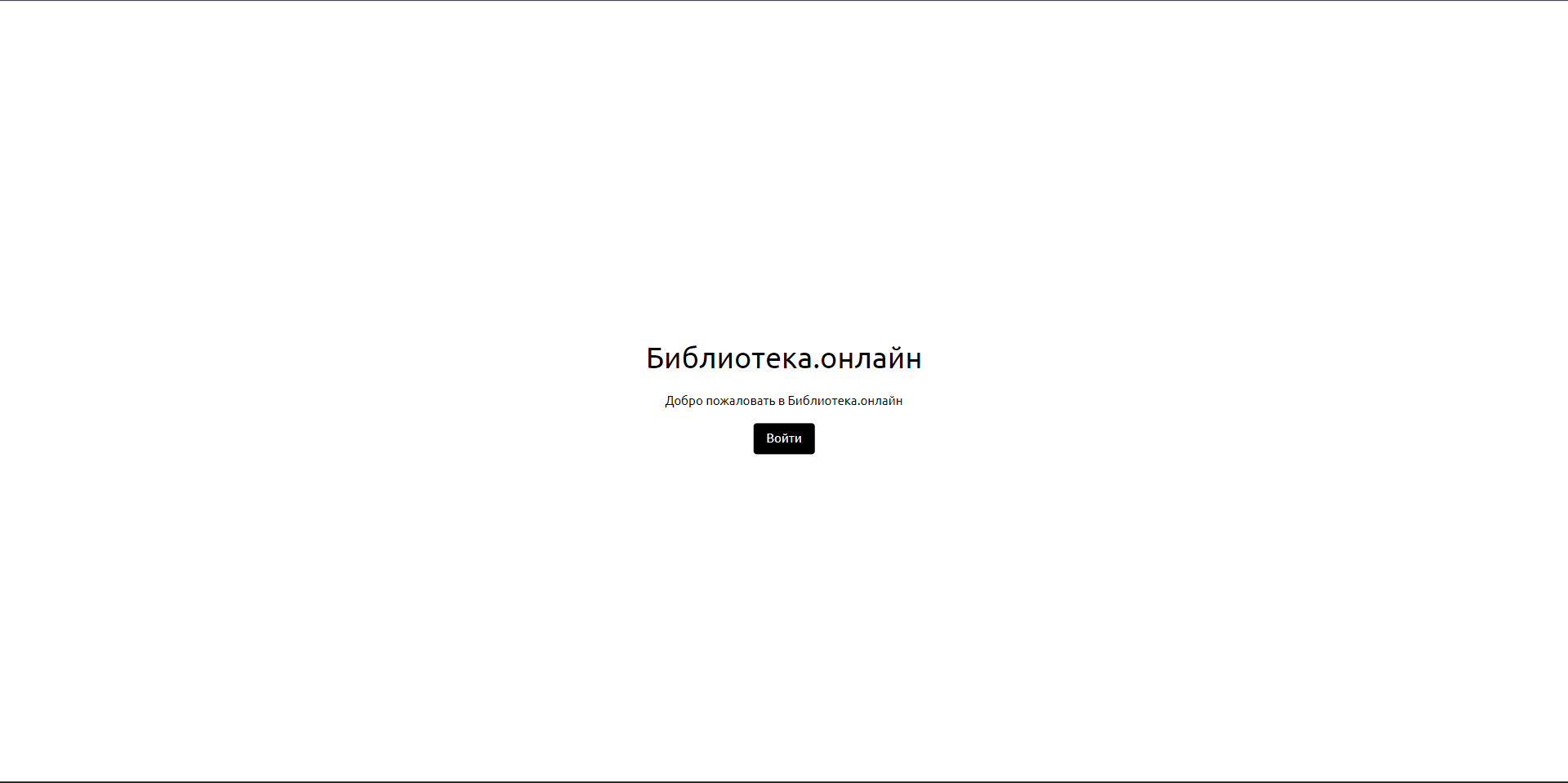


Рисунок 4 – итоговый вариант страницы входа.

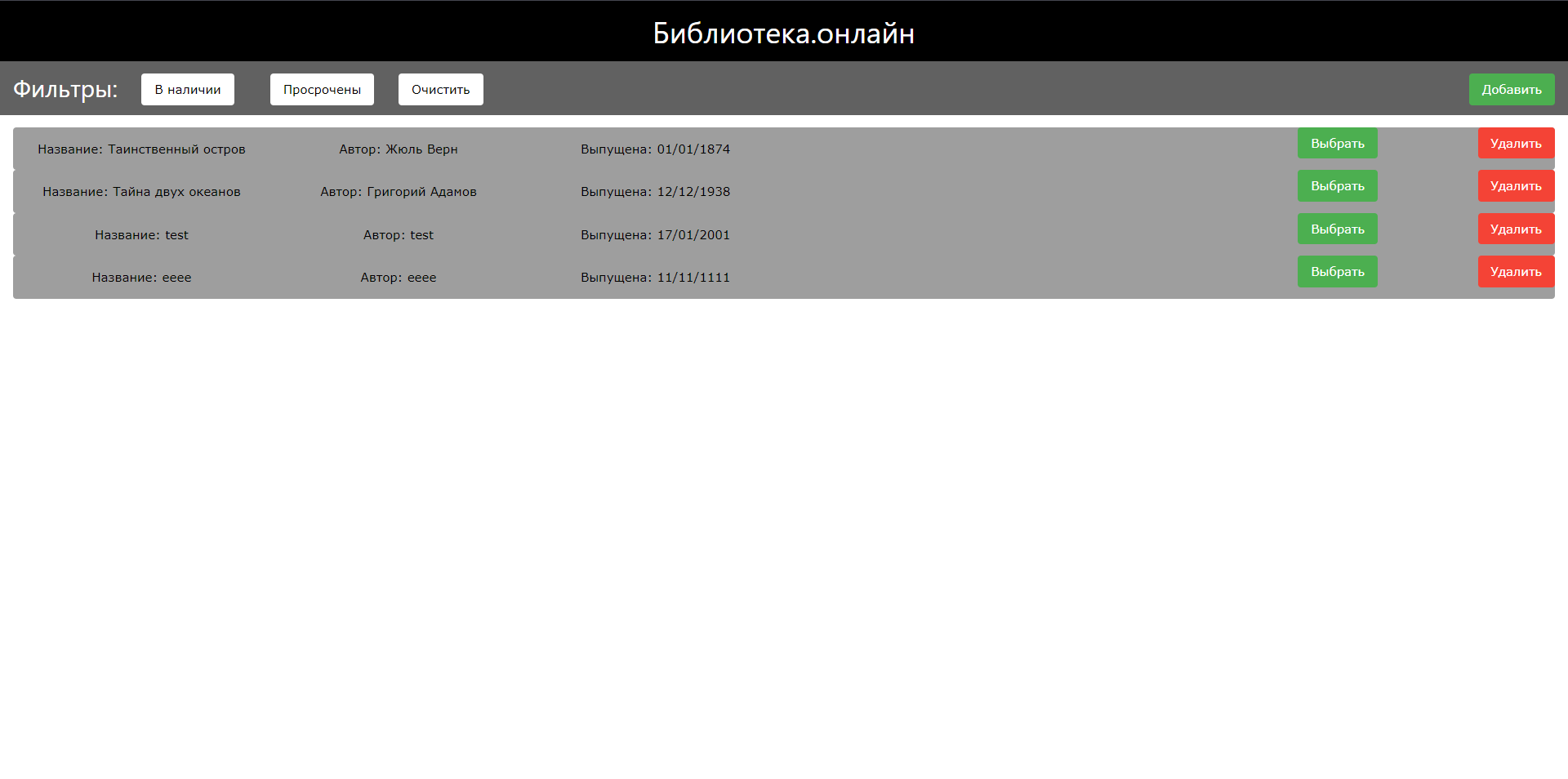


Рисунок 5 – итоговый вариант страницы библиотеки.

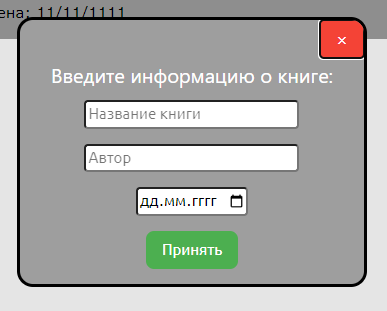


Рисунок 6 – диалоговое окно добавления новой книги.

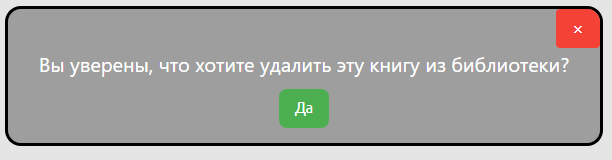


Рисунок 6 – диалоговое окно подтверждения удаления книги.

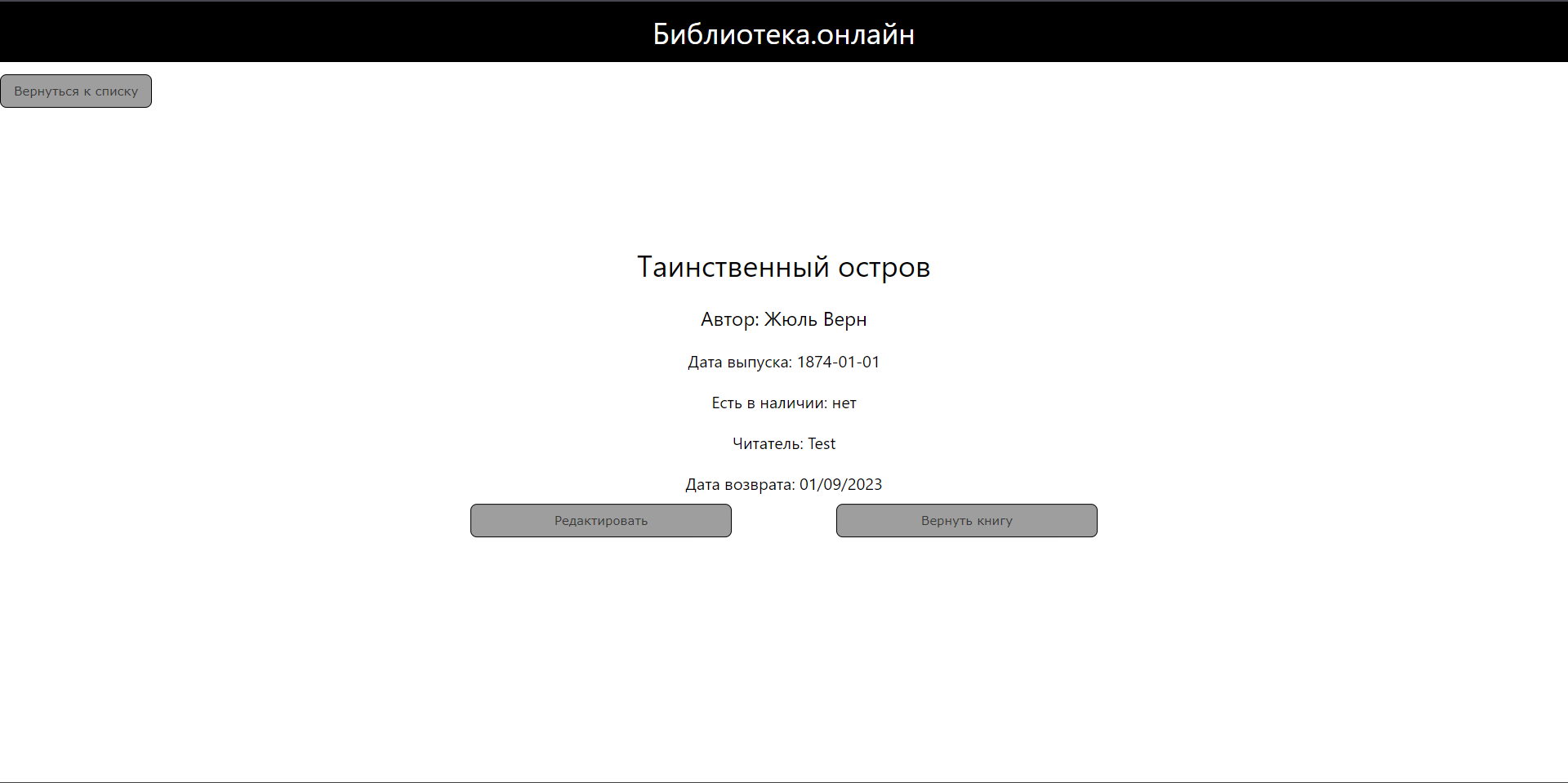


Рисунок 7 – итоговый вариант страницы книги.

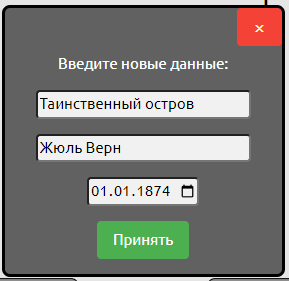


Рисунок 8 – диалоговое окно редактирования данных о книге.

При открытии окна редактирования книги уже существующие данные будут подставлены в поля ввода.

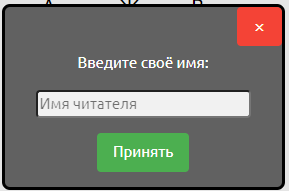


Рисунок 9 – диалоговое окно взятия книги читателем.

**Приложение А  
Исходный код**

**app.js:**

import express from 'express'

import {router} from "./router.js";

let app = express();

app.set('views', './views');

app.set('view engine', 'pug');

app.use(express.static('./public'));

app.use('/', router);

app.set('port', '3000');

app.listen('3000')

**router.js:**

import express from 'express';

import body\_parser from "body-parser";

import fs from "fs";

let router = express.Router();

router.use(body\_parser.json({limit: '50mb'}));

router.use(body\_parser.urlencoded({limit: '50mb', extended: true}));

router.use(express.json());

let data = fs.readFileSync('./public/json/lib.json', 'utf-8');

let book\_json = JSON.parse(data);

router.get('/', (req, res) => {

res.render('index', {title: 'Библиотека.онлайн'});

});

router.get('/library', (req, res) => {

res.render('library', {title: 'Библиотека.онлайн', books: book\_json});

});

router.get('/library/:id', (req, res) => {

const id = req.params.id;

if (id === 'no\_filter') {

let array = [];

for(let obj of book\_json){

array.push(obj.id)

}

res.end(JSON.stringify(array));

return;

}

if (id === 'in\_stock') {

let array = [];

for(let obj of book\_json){

if (obj.in\_stock === "нет")

array.push(obj.id)

}

res.end(JSON.stringify(array));

return;

}

if (id === 'overdue') {

let array = [];

let cur\_date = new Date();

for(let obj of book\_json){

let obj\_date = new Date(obj.date\_return)

if (obj\_date > cur\_date || obj.in\_stock == "да") {

array.push(obj.id);

}

}

res.end(JSON.stringify(array));

return;

}

});

router.post('/book/edit/:id', (req, res) => {

let id = req.params.id;

for (let value of book\_json){

if (value.id == id){

value.title = req.body.title;

value.author = req.body.author;

value.date = req.body.date;

res.redirect('/book/'+id);

return;

}

}

});

router.post('/book/delete/:id', (req, res) => {

let id = req.params.id;

for(let i = 0; book\_json.length; i++){

if(book\_json[i].id == id) {

book\_json.splice(i, 1);

res.redirect('/library');

return;

}

}

});

router.post('/book/new', (req, res) => {

book\_json.push({

"id": book\_json.length,

"title": req.body.title,

"author": req.body.author,

"date": req.body.date,

"in\_stock": "да",

"reader": "-",

"date\_return": "-"

});

res.redirect('/library');

});

router.post('/book/take/:id', (req, res) => {

const id = req.params.id;

let cur\_date = new Date();

cur\_date.setDate(cur\_date.getDate() + 30);

for (let value of book\_json){

if (value.id == id){

value.reader = req.body.reader;

value.date\_return = cur\_date;

value.in\_stock = "нет";

res.redirect('/book/'+id);

return;

}

}

});

router.post('/book/return/:id', (req, res) => {

const id = req.params.id;

for (let value of book\_json){

if (value.id == id){

value.reader = '-';

value.date\_return = '-';

value.in\_stock = "да";

res.redirect('/book/'+id);

return;

}

}

});

router.get('/book/:id', (req, res) => {

const id = req.params.id;

for (let value of book\_json) {

if (value.id == id) {

res.render('book', {

page\_title: 'Библиотека.онлайн', ID: `${value.id}`, title: `${value.title}`,

author: `${value.author}`, date: `${value.date}`, in\_stock: `${value.in\_stock}`,

reader: `${value.reader}`, return\_date: `${value.date\_return}`

});

return;

}

}

});

router.get("\*", (req, res) => {

res.status(404);

res.end("Page not found")

});

export {router};

**ajax.js:**

function no\_filter(button) {

let id = button.id;

callAjax(id, (response)=>{

let arr = JSON.parse(response)

for(let el of arr){

let cur\_tile =document.getElementById(el)

cur\_tile.style.display = 'block'

}

});

}

function in\_stock(button) {

let id = button.id;

callAjax(id, (response) => {

let arr = JSON.parse(response);

for(let el of arr) {

let cur\_tile = document.getElementById(el);

cur\_tile.style.display = 'none';

}

});

}

function overdue(button) {

let id = button.id;

callAjax(id, (response)=>{

let arr = JSON.parse(response);

for(let el of arr) {

let cur\_tile = document.getElementById(el);

cur\_tile.style.display = 'none';

}

})

}

function callAjax(id, callback) {

let xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function () {

if(this.readyState == 4 && this.status == 200)

callback(this.responseText);

};

xhttp.open("GET",`/library/${id}`,true);

xhttp.send();

}

**book.pug:**

doctype html

html

head

title= title

link(rel='stylesheet', href='https://cdnjs.cloudflare.com/ajax/libs/w3-css/4.1.0/w3.min.css')

body

block content

block var

block content

div.w3-container.w3-black

h1.w3-center= 'Библиотека.онлайн'

div

p

form(action="/library" method="GET")

div.w3-col.s1.w3-center

button.w3-button.w3-gray.w3-border.w3-border-black.w3-round-large.w3-text-dark-gray(type='Submit') Вернуться к списку

div.w3-container.w3-center.w3-col.s5.w3-display-middle

p

div.w3-col.s2

div(style="display: flex; flex-direction: column; text-align: left").w3-center

h1 #{title}

h3 Автор: #{author}

h4 Дата выпуска: #{date}

h4 Есть в наличии: #{in\_stock}

h4 Читатель: #{reader}

if return\_date === '-'

h4 Дата возврата: #{return\_date}

else

h4 Дата возврата: #{new Date(return\_date).toLocaleDateString('en-GB')}

div.w3-center

button(onclick="window.editDialog.showModal()").w3-left.w3-button.w3-gray.w3-border.w3-border-black.w3-round-large.w3-text-dark-gray.w3-col.s5 Редактировать

dialog(id="editDialog").w3-round-large.w3-dark-gray.w3-border-black

button(onclick="window.editDialog.close()").w3-button.w3-red.w3-display-topright.w3-round-medium &times

p

form(action="/book/edit/" + ID method="POST").w3-col

div.w3-container.w3-serif

p Введите новые данные:

input(required name="title" id="title" value=title).w3-round-medium.w3-light-gray

p

input(required name="author" id="author" value=author).w3-round-medium.w3-light-gray

p

input(required type="date" id="date" name="date" max = "2023-10-30" value = date).w3-round-medium.w3-light-gray

p

button.w3-button.w3-green.w3-round-medium(type='Submit') Принять

if in\_stock === "да"

button(onclick="window.takeDialog.showModal()").w3-gray.w3-border.w3-border-black.w3-round-large.w3-text-dark-gray.w3-right.w3-button.w3-col.s5 Взять книгу

dialog(id="takeDialog").w3-round-large.w3-dark-gray.w3-border-black

button(onclick="window.takeDialog.close()").w3-button.w3-red.w3-display-topright.w3-round-medium &times

p

form(action="/book/take/"+ID method="POST").w3-col

div.w3-container.w3-serif

p Введите своё имя:

input(required name="reader" id="reader" placeholder="Имя читателя").w3-round-medium.w3-light-gray

p

button.w3-button.w3-green.w3-round-medium(type='Submit') Принять

else

form(action="/book/return/"+ID method="POST")

button.w3-gray.w3-border.w3-border-black.w3-round-large.w3-text-dark-gray.w3-right.w3-button.w3-col.s5(type='Submit') Вернуть книгу

p.w3-col.s2

**index.pug:**

doctype html

html

head

title= title

link(rel='stylesheet', href='https://cdnjs.cloudflare.com/ajax/libs/w3-css/4.1.0/w3.min.css')

body.w3-white

div.w3-container.w3-white.w3-center.w3-serif.w3-display-middle

h1.w3-center.w3-serif= title

p Добро пожаловать в #{title}

form(action="/library" method="GET")

button.w3-button.w3-black.w3-round-medium Войти

**library.pug:**

doctype html

html

head

title= title

link(rel='stylesheet', href='https://cdnjs.cloudflare.com/ajax/libs/w3-css/4.1.0/w3.min.css')

body

block content

block var

block content

script(src = '/javascripts/ajax.js')

div.w3-container.w3-black.w3-center

h1= title

div.w3-dark-gray.w3-container.w3-center

div.w3-col.l1.w3-left

h2.w3-left Фильтры:

div.w3-col.l1.w3-left

p

button(onclick='in\_stock(this)' id="in\_stock").w3-round-medium.w3-left.w3-button.w3-white В наличии

div.w3-col.l1.w3-left

p

button(onclick='overdue(this)' id="overdue").w3-round-medium.w3-left.w3-button.w3-white Просрочены

div.w3-col.l4.w3-left

p

button(onclick='no\_filter(this)' id="no\_filter").w3-round-medium.w3-left.w3-button.w3-white Очистить

div.w3-col.s3.w3-right

p

button(onclick="window.addDialog.showModal()").w3-round-medium.w3-right.w3-button.w3-green Добавить

dialog(id="addDialog").w3-round-xlarge.w3-gray.w3-text-white.w3-border-black

button(onclick="window.addDialog.close()").w3-button.w3-red.w3-display-topright.w3-round-medium &times

p

form(action="/book/new" method="POST").w3-col

div.w3-container.w3-serif

h4 Введите информацию о книге:

input(required name="title" id="title" placeholder="Название книги").w3-round-medium

p

input(required name="author" id="author" placeholder="Автор").w3-round-medium

p

input(required type="date" id="date" name="date" max = "2023-10-30").w3-round-medium

p

button.w3-button.w3-green.w3-round-large.w3-border-black(type='Submit') Принять

div.w3-container

each item, ind in books

p

div(id = item.id).w3-col.l12.w3-border-black.w3-center.w3-gray.w3-round-medium

p.w3-col.l2 Название: #{item.title}

p.w3-col.l2 Автор: #{item.author}

p.w3-col.l2 Выпущена: #{new Date(item.date).toLocaleDateString('en-GB')}

div.w3-right.w3-col.m2

form(action="/book/"+item.id method="GET")

div

button.w3-button.w3-green.w3-round-medium.w3-left(type='Submit') Выбрать

button(onclick=`window.deleteDialog[${ind}].showModal()`).w3-button.w3-red.w3-round-medium.w3-right Удалить

dialog(id="deleteDialog").w3-round-xlarge.w3-gray.w3-text-white.w3-border-black

button(onclick=`window.deleteDialog[${ind}].close()`).w3-button.w3-red.w3-display-topright.w3-round-medium &times

p

form(action="/book/delete/"+item.id method="POST").w3-col

div.w3-container.w3-serif

h4 Вы уверены, что хотите удалить эту книгу из библиотеки?

button.w3-button.w3-green.w3-round-large.w3-border-black(type='Submit') Да

//div.w3-white

// p.w3-text-white .