**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Веб-технологии»**

Тема: Модуль администрирования приложения “Аукцион картин”

|  |  |  |
| --- | --- | --- |
| Студент гр. 8304 |  | Сергеев А. Д. |
| Преподаватель |  | Беляев С. А. |

Санкт-Петербург

2020

**Цель работы**

Изучение возможностей применения компилятора Babel, библиотеки jQuery, предпроцессора LESS, инструмента выполнения повторяющихся задач GULP, регистрация разработанных модулей, формирование навыков построения структурированных web-приложений, освоение особенностей стандартных библиотек.

**Основные теоретические сведения**

LESS — это динамический язык стилей, обеспечивает следующие расширения CSS: переменные, вложенные блоки, миксины, операторы и функции. LESS может работать на стороне клиента или на стороне сервера под управлением Node.JS.

jQuery — библиотека JavaScript, предназначенная для упрощения взаимодействии JavaScript и НТМL. Библиотека jQuery помогает получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими, предоставляет простой API для работы с AJAX.

Babel — компилятор JavaScript, который позволяет разработчику использовать в своих проектах самые последние стандарты ЕСMAScript с поддержкой во всех браузерах.

Gulp — это менеджер задач для автоматического выполнения часто используемых задач, написанный на JavaScript, Программное обеспечение поддерживает командную строку для запуска задач, определенных в конфигурационном файле записи.

**Общая формулировка задачи**

Необходимо создать web-приложение, обеспечивающее администрирование аукциона картин: можно выбрать картины для участия в аукционе, определить перечень участников и параметры аукциона. 0сновные требования следующие:

1. Перечень доступных картин с описаниями и ссылками на рисунки хранится в JSON-файле.

2. В качестве сервера используется Node.JS с модулем express.

3. Разработка ведется с использованием стандарта не ниже ECMAScript2015.

4. Стили описываются c использованием LESS, при этом используются

ключевые методы LESS(переменные, вложенные блоки, миксины, операторы и т.п.).

5. Клиентская часть разрабатывается с использованием jQuery(работа с DOM, AJAX-запросы).

6. Предусмотрена HTML-страница для перечня картин и карточка отдельной картины (название, автор, описание, изображение, начальная цена, минимальный и максимальный шаги аукциона). Предусмотрена возможность редактировать текстовые и числовые параметры, а также включить или исключить картину из участия в предстоящих торгах, загрузить рисунок картины.

7. Предусмотрена HTML-страница для списка потенциальных участников аукциона. Есть возможность добавлять или удалять участников, изменять

запас денежных средств.

8. Предусмотрена HTML-страница для настроек аукциона (настройка даты и времени начала аукциона, настройка таймаута продажи картины, настройка интервала времени отсчета до окончания торга по картине, паузы на изучение информации по картине для начала торга по ней).

9. Взаимодействие браузера с сервером осуществляется по протоколу

НТТРS.

10. Сборка клиентской части (преобразования less, pug, babel, минификация)

осуществляется с использованием GULP.

11. Регистрация и удаление разработанных модулей в m.

12. Сохранение сформированных настроек в JSON-файл.

**Ход работы**

1. Используя среду разработки JetBrains WebStorm, были установлены все необходимые расширения.
2. Используя модуль express, был создан и настроен сервер.
3. Были созданы и настроены pug и less файлы.
4. Для упрощения создания пользовательского интерфейса была использована библиотека bootstrap.
5. Разработка интерфейса пользователя:

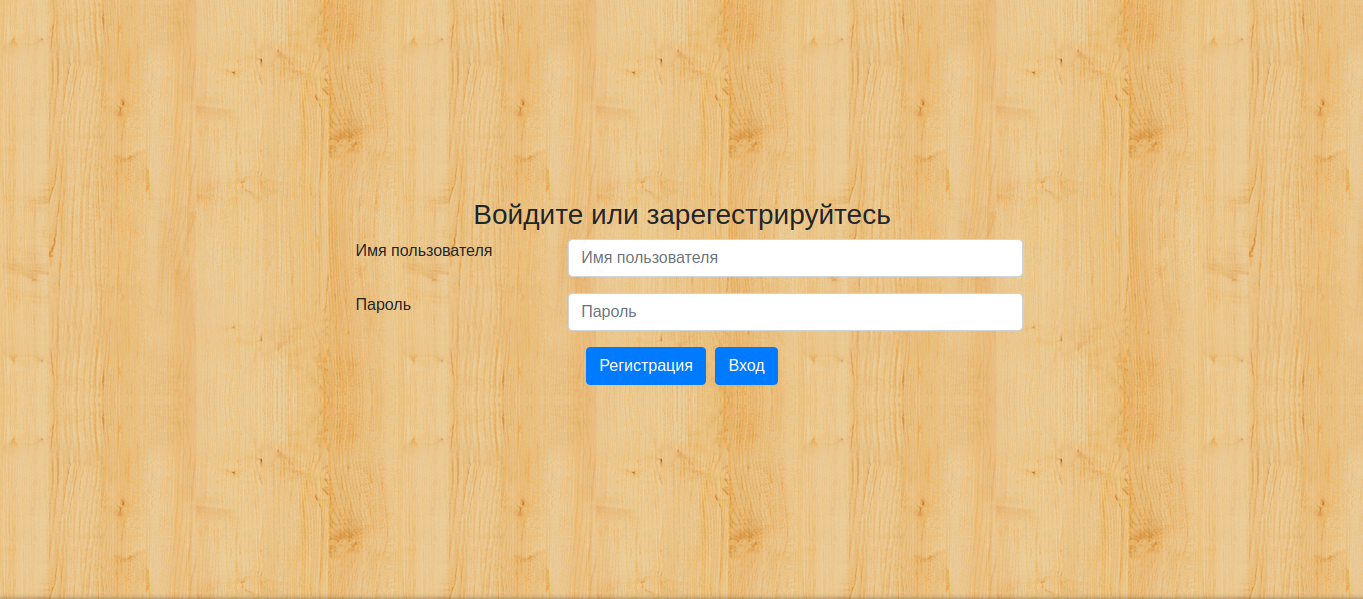
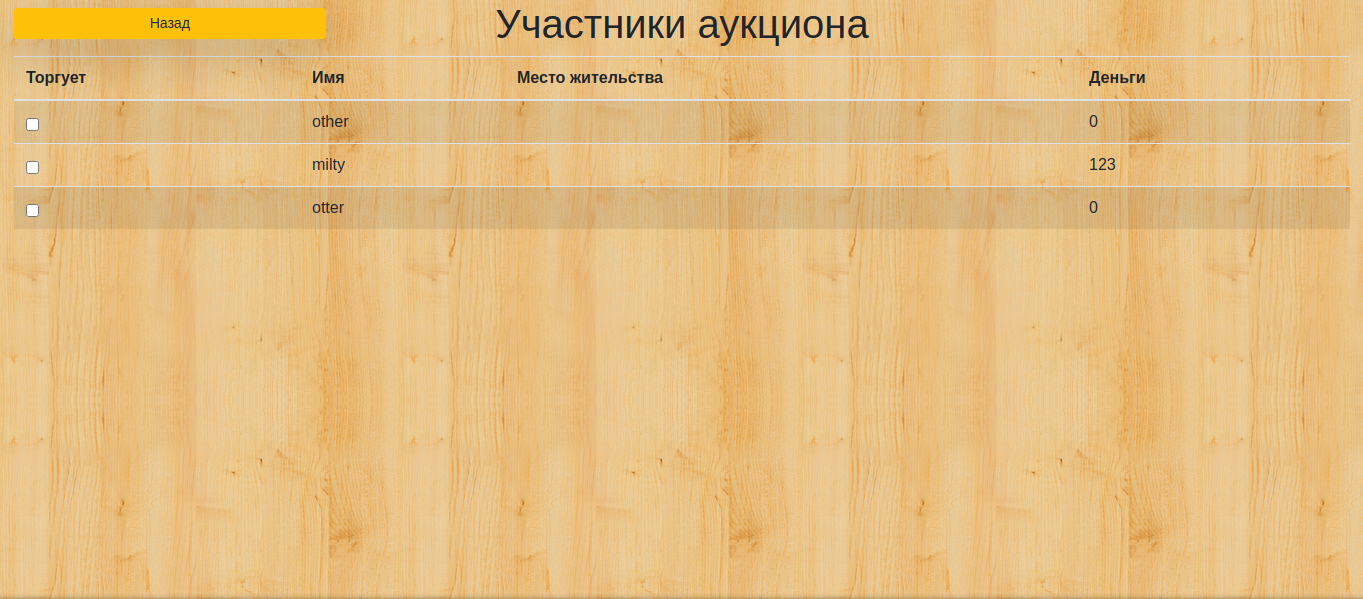
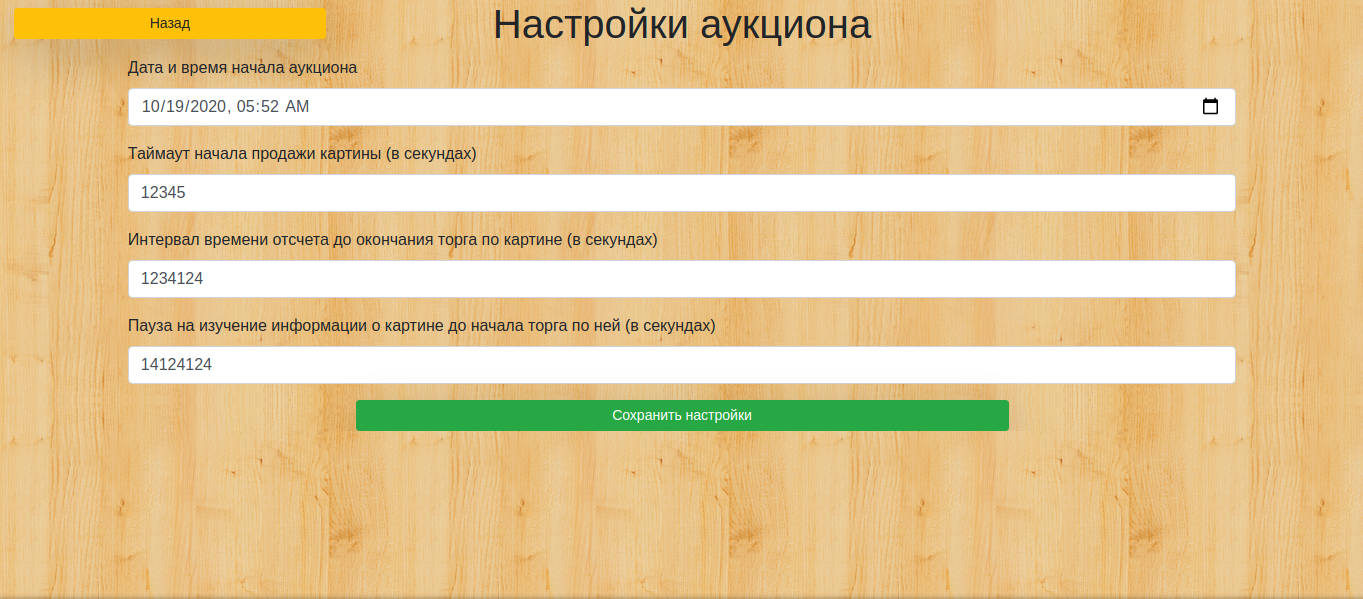
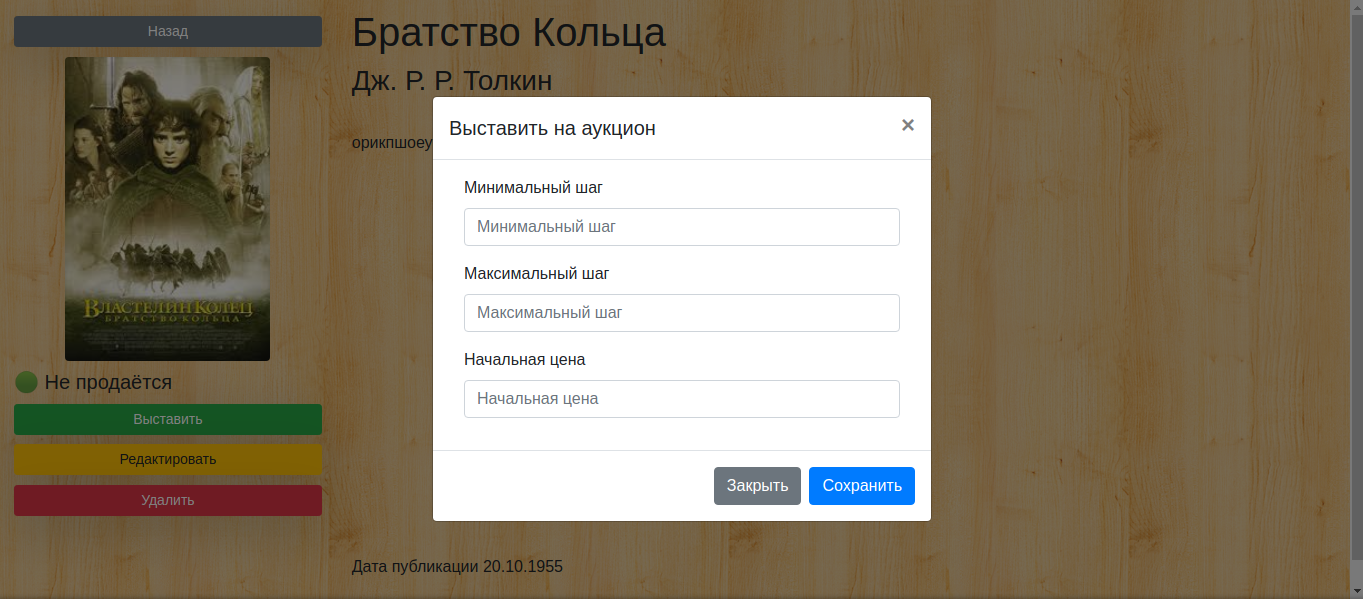
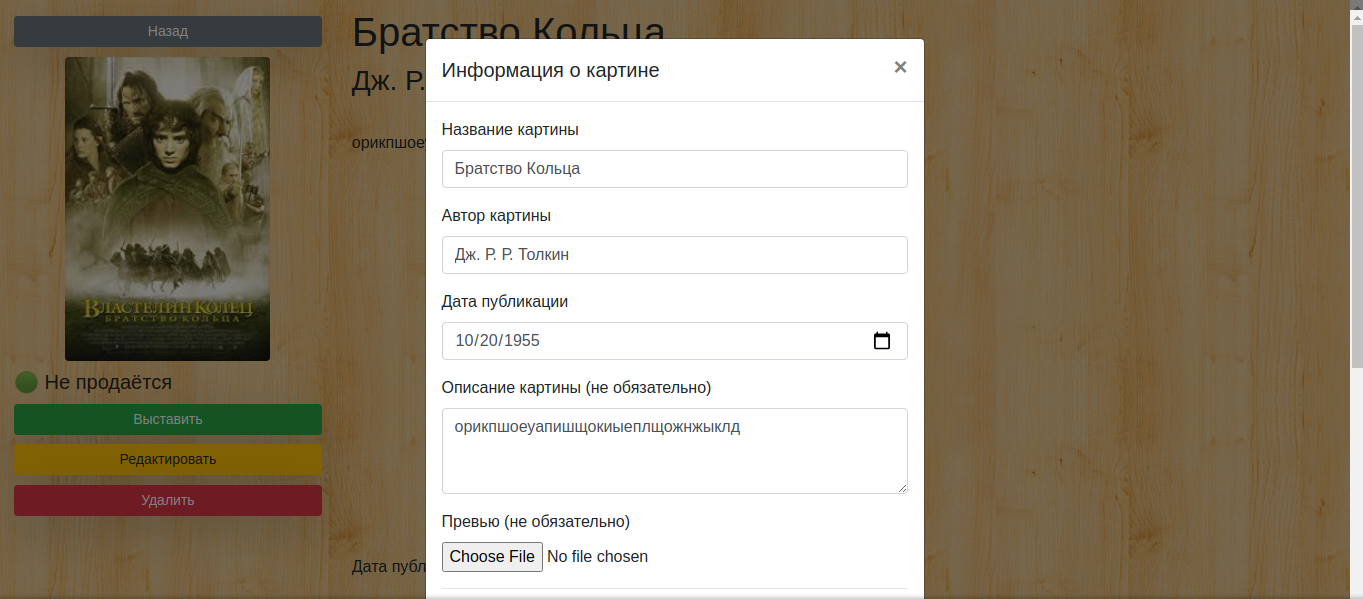
Рисунок 1 – Страница входа на сайт.

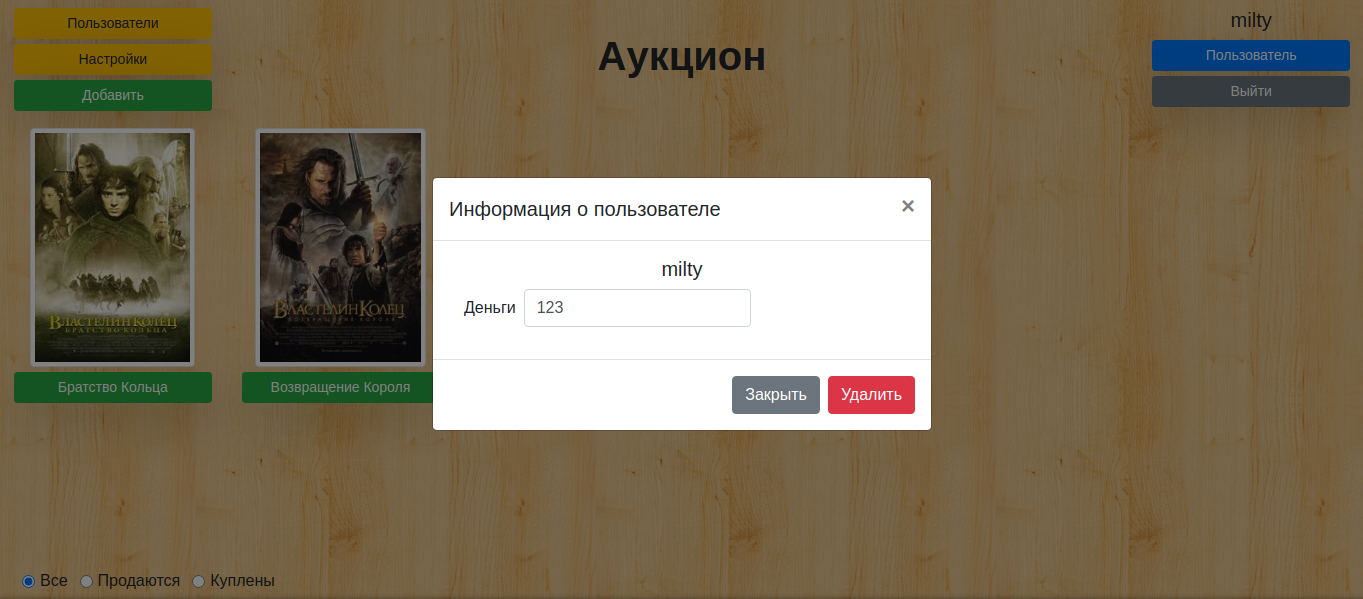
Рисунок 2 – Основная страница сайта.

Рисунок 3 – Список участников.

Рисунок 4 — Настройки аукциона.

Рисунок 5 — Модальное окно выставления картины на аукцион.

Рисунок 6 — Модальное окно добавления новой картины.

Рисунок 7 — Модальное окно информации о пользователе.

1. Для проекта были созданы следующие файлы.
2. custom.less – набор стилей для оформления страниц.
3. base.pug, 404.pug, book.pug, book\_modal.pug, lib.pug, login.pug, settings.pug, take\_modal.pug, user\_modal.pug, users.pug – базовая страница и web-страницы для ненайденной страницы, конкретной картины, модального окна редактирования картины, аукциона, формы входа и регистрации, настроек аукциона, модального окна выставления картины на аукцион, модального окна информации о пользователе, а также страницы с информацией о пользователях соответственно.
4. lib.json, users.json, passwords.json, settings.json – файлы, хранящие в себе начальные настройки книг, информацию о пользователях и их паролях, а также настройках аукциона соответственно.
5. api.js, auth.js, coverage.js, library.js, main.js, population.js, routs.js, settings.js – содержат функции, обеспечивающие работу сервера, отвечающие за rest api, аутентификацию, работу с изображениями, аукцион, основные функции, работу с пользователями, роуты сайта, а также настройки аукциона соответственно.
6. book.js, book\_modal.js, lib.js, login.js, settings.js, take\_modal.js user\_modal.js, user.js, accounter.js, book\_manager.js, general.js, recovery.js, settings.js – первые 8 содержат скрипты, отвечающие за работу конкретных страниц, а последние 6 — скрипты, генерирующие AJAX-запросы для разных частей сайта, а также общие скрипты, использующиеся на нескольких страницах сразу.

**Вывод**

В ходе лабораторной работы был получен опыт работы с pug, less файлами, Gulp и jQuery сервера на основе создания Модуля администрирования приложения “Аукцион картин”.