

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Web технологии»
Тема: МОДУЛЬ АДМИНИСТРИРОВАНИЯ ПРИЛОЖЕНИЯ
«АУКЦИОН КАРТИН»

Студент гр. 9382

Иерусалимов Н.

Преподаватель

Беляев С.

Санкт-Петербург

2021

Цель работы.

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

Задание.

Необходимо создать web-приложение, обеспечивающее администрирование аукциона картин: можно выбрать картины для участия в аукционе, определить начальные ставки, перечень участников и параметры аукциона. Основные 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 с 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 HTTPS. 10. Сборка клиентской части (преобразования less, pug, babel, минификация) осуществляется с использованием GULP. 11. Регистрация и удаление разработанных модулей в npm. 12. Сохранение сформированных настроек в JSON-файл. Преимуществом будет предоставление возможности с использованием карты (maps.yandex.ru, maps.google.com, openstreetmap.org и т.п.) указать для каждого участника место его проживания.

Описание структуры проекта

Приложение было написано при помощи node.js серверная часть была написана на express. frontend был написан на pug и java script.

Index.js файл откуда запускается сервер, где прописывается статические папки и указывается что в качестве разметки будет использован pug.

В папке pages прописанна разметка для всех страниц

В папке public находится css стили, главный js файл где идет работа фронта с сервером и файл json где хранятся данные по библиотеке.

В папке route находится файл route.js который обрабатывает запросы от фронта и либо обновляет данные на сервере либо отправляет данные по запросу на фронт.

- ▼ database
 - 📄 paintings.json
 - 📄 participants.json
 - 📄 settings.json
- Image
- ▼ src
 - > 📄 main.js
 - 📄 paintingPage.js
 - 📄 participants.js
 - 📄 settingsPage.js
- ▼ style
 - ▼ 📄 mainAuc.less
 - 📄 mainAuc.css
 - 📄 mainAuc.css.map
 - > 📄 mainPage.less
 - > 📄 paddingForPage.less
 - ▼ 📄 pages.less
 - 📄 pages.css
 - 📄 pages.css.map
 - > 📄 settingsPage.less
- ▼ route
 - 📄 controlerdb.js
 - 📄 pages.js
 - 📄 painting.js
 - 📄 participants.js
 - 📄 routes.js
- > ssl
- ▼ views
 - ▼ pages
 - 📄 mainPage.pug
 - 📄 paintings.pug
 - 📄 participants.pug
 - 📄 settings.pug
 - 📄 temp.pug
 - 📄 head.pug
 - 📄 jquery.pug
 - 📄 navBar.pug
- 📄 .babelrc
- 📄 gulpfile.js
- 📄 index.js
- 📄 package.json
- 📄 package-lock.json

Тестирование.

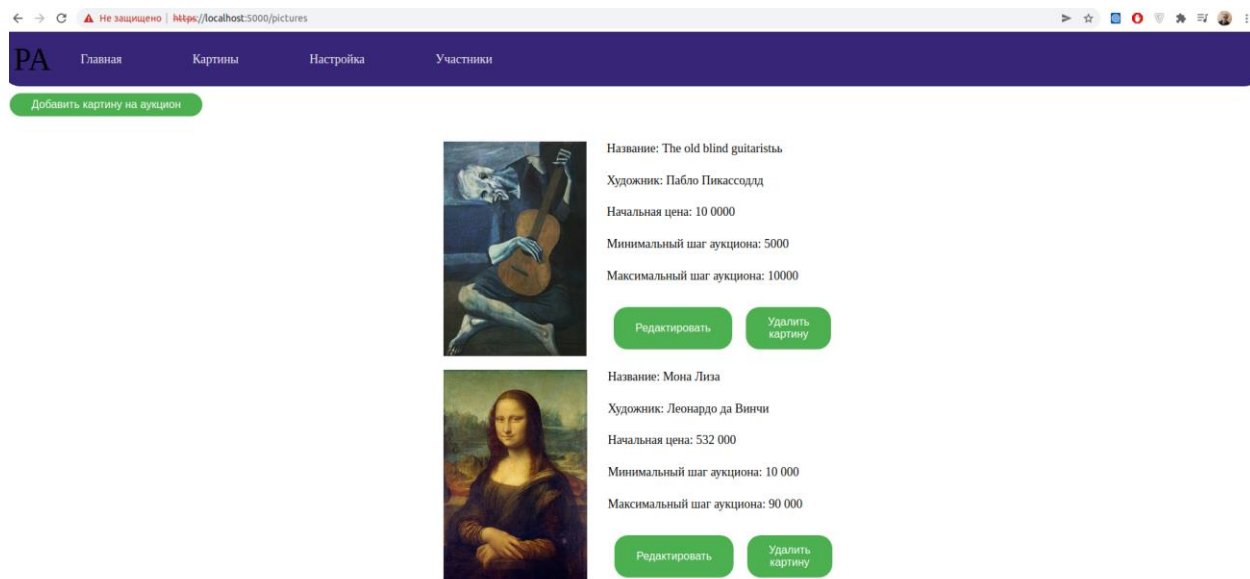


Рис 1<Страница с картинами>

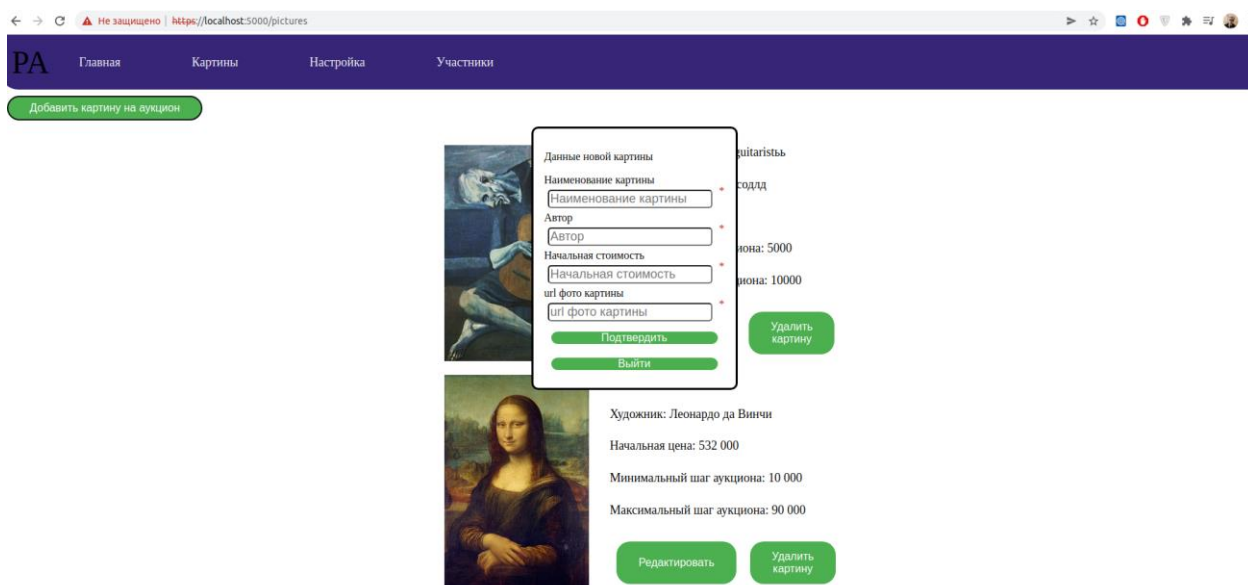


Рис 2<Добавление картины>

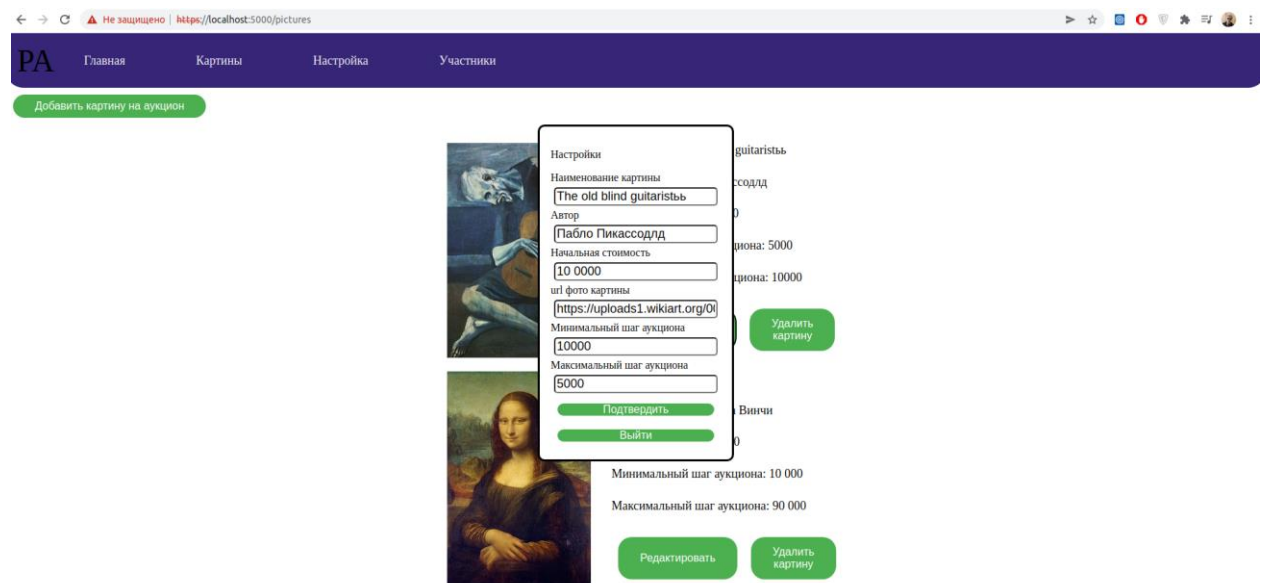


Рис 3<Редактирование картины>

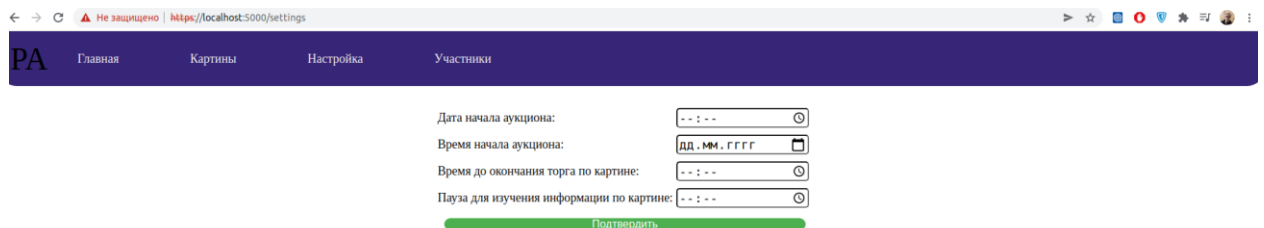


Рис 4<Страница с настройкой аукционна>

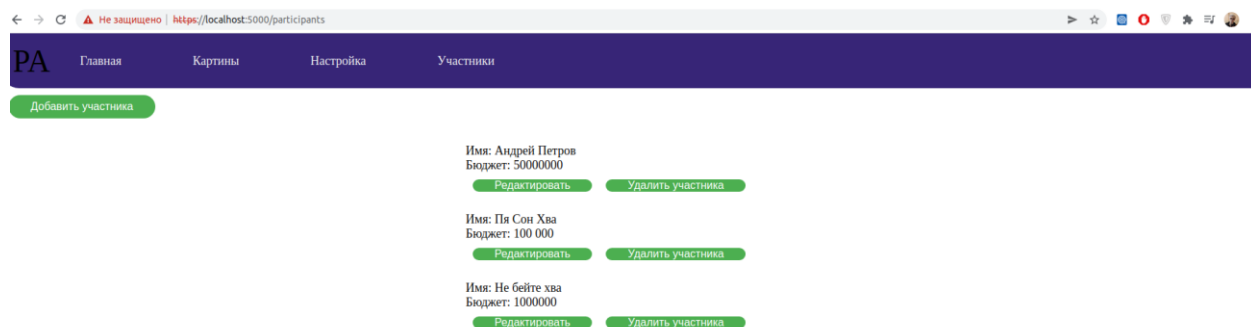


Рис 5<Страница с Участниками>

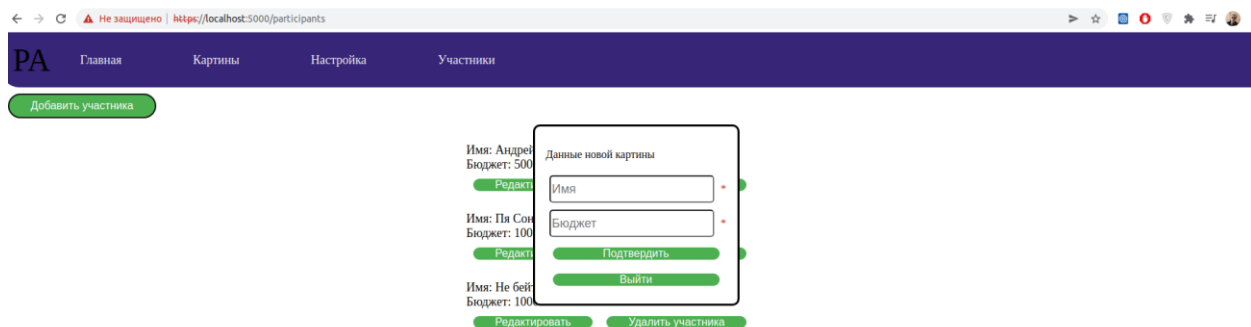


Рис 6<Добавление участника>

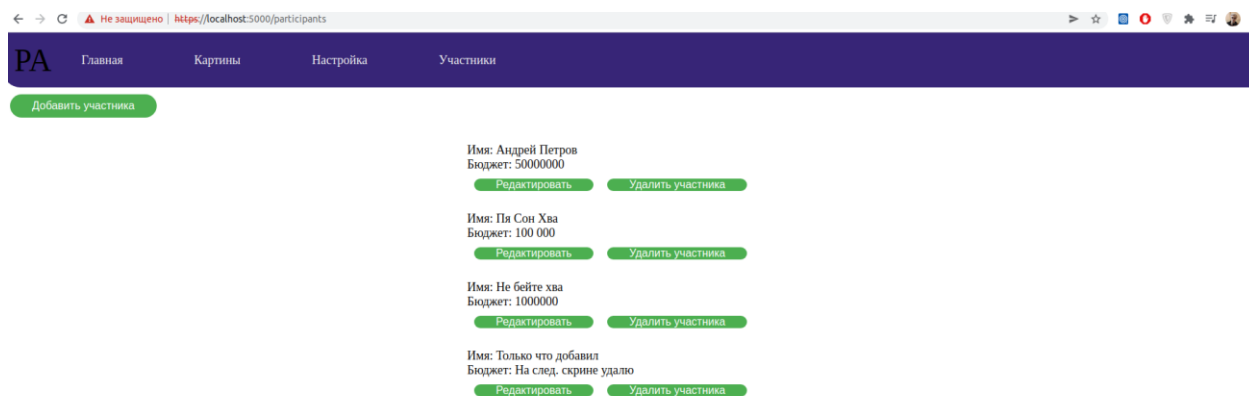


Рис 7<Добавление нового участника>

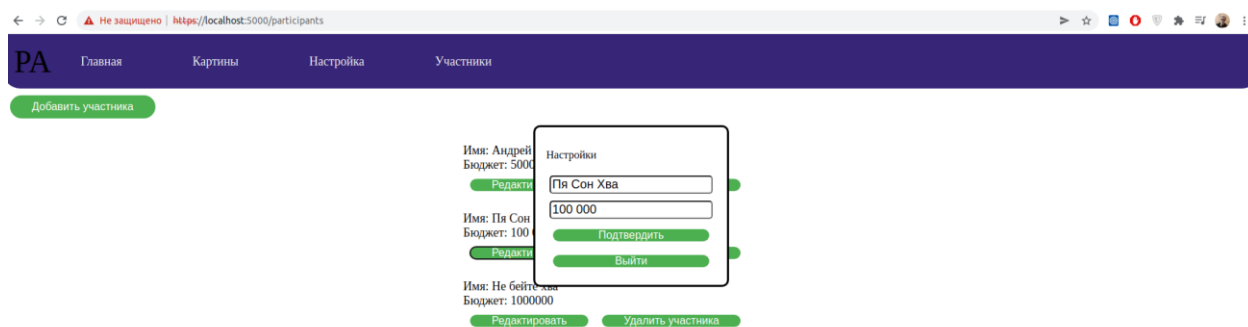


Рис 8<Редактирование участника и удаление>

Выводы.

В процессе выполнения работы были изучены инструменты взаимодействия для оптимизации работы такие как glup. Были получены навыки разработки с помощью less, jquery. Так же использовался babel для конвертации ES6 кода в ES5 и дальнейшей минификацией с помощью glup.

По итогу был создан скелет для аукциона в котором можно добавлять картины, участников и настраивать параметры для аукционна.