**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «WEB-Технологии»**

Тема: **Тетрис на javascript**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6382 |  | Вайгачёв А.О. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2018

**Цель и задачи**

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.

**Для достижения поставленной цели требуется решить следующие задачи:**

1) Разработка интерфейса web-приложения.

2) Создание web-сервера на основе express, настройка маршрутов, подготовка и обработка REST-запросов (серверная часть).

3) Создание шаблонов web-страниц, указание путей подключения js-файлов.

4) Разработка стилей web-приложения с использованием LESS.

5) Разработка клиентских js-файлов с использованием библиотеки jQuery и с использованием новейших возможностей в соответствии последним стандартом ECMAScript.

5) Конфигурирование GULP для решения задач преобразования pug-файлов в формат HTML, less-файлов в css-файлы, обработка js-файлов с использованием Babel.

**Общая формулировка задачи**

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:

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 ведётся с использованием стандарта не ниже ECMAScript2015.

4. Стили описываются с использованием LESS, при этом используются ключевые методы LESS (переменные, вложенные блоки, миксины, операто-ры 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.

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 отсчёта 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, минифи-кация) осуществляется с использованием GULP.

11. Регистрация разработанных модулей в npm.

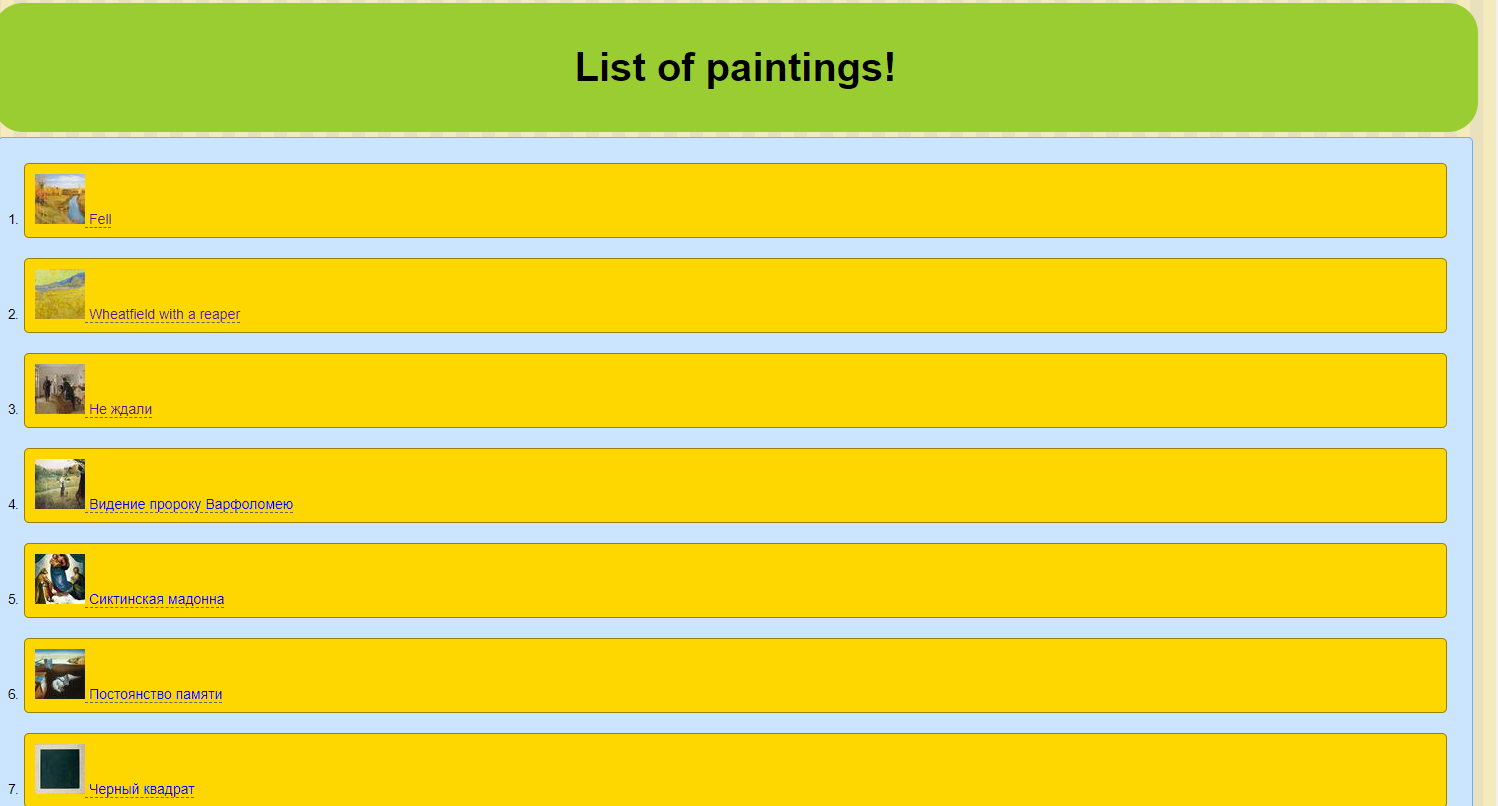
12. Сохранение сформированных настроек в JSON-файл.

**Выполнение работы:**

**Главная страница:**



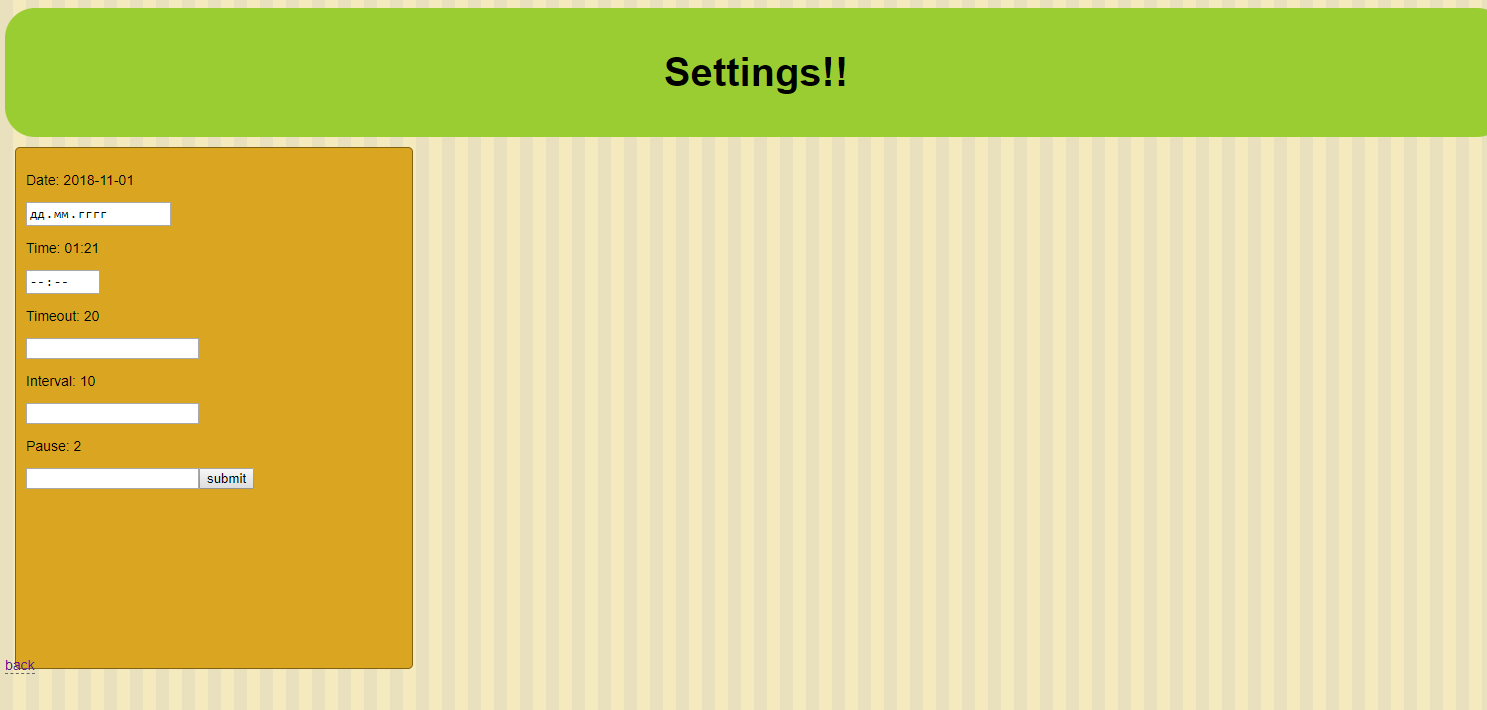
**Страница с картинами:**



**Страниц с участниками:**



**Страница с настройками аукциона:**



**Вывод:**

Мы изучили возможности 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.