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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Web-технологии»

Тема: Модуль приложения «участие в аукционе картин»

Студент гр. 8383 _____ Шишкин И.В.

Преподаватель _____ Беляев С.А.

Санкт-Петербург

2020

Цель работы

Изучить возможности применения jQuery UI для создания интерфейса пользователя, обработки ошибок, ведения журналов ошибок, реализовать взаимодействие приложения с использование web-сокетов, применить статический анализатор Flow, освоить инструмент сборки WebPack, и организовать модульное тестирования web-приложений с использованием Mocha.

Задача

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.

Основные теоретические положения

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 HTML. 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 ECMAScript 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.

Ход работы

Модульное тестирование с использованием Mocha представлено на рис. 1.1. Происходит проверка на аутентификацию.

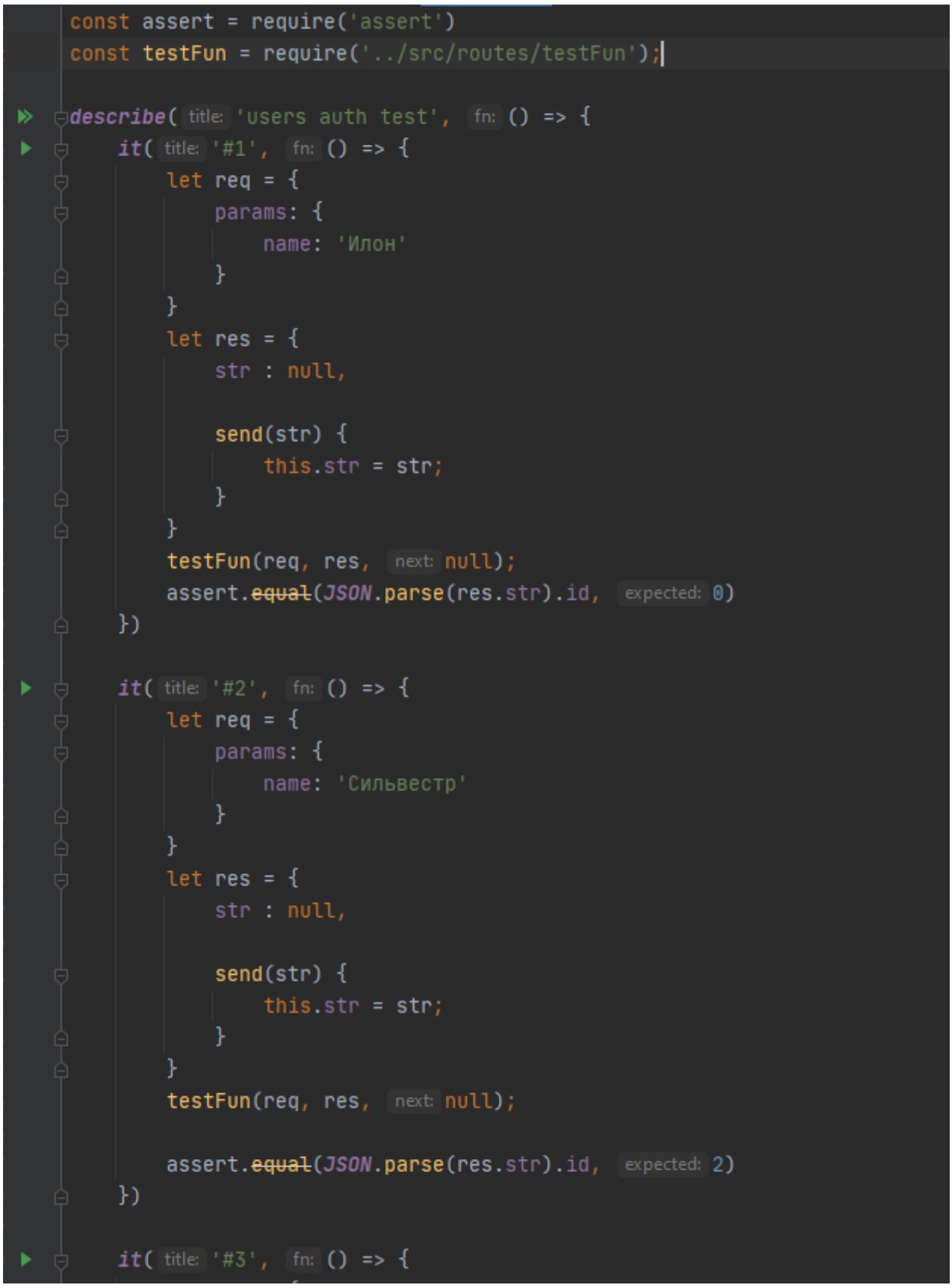

Рисунок 1.1.

Начальная страница представлена на рис. 1.2.

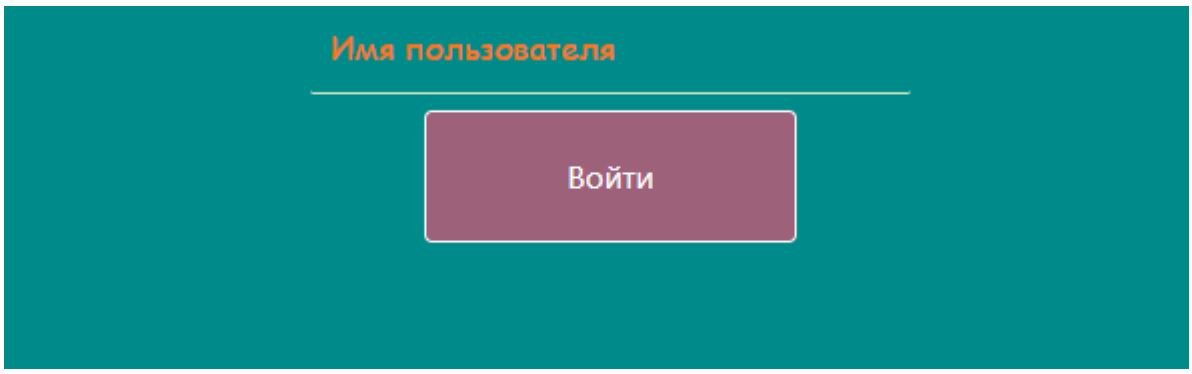

Рисунок 1.2.

Если ввести неверное имя (имя, которого нет в data.json, которая берется из 3 ЛР в node_modules, так как она была опубликована через npm publish), то произойдет ошибка (рис. 1.3.).

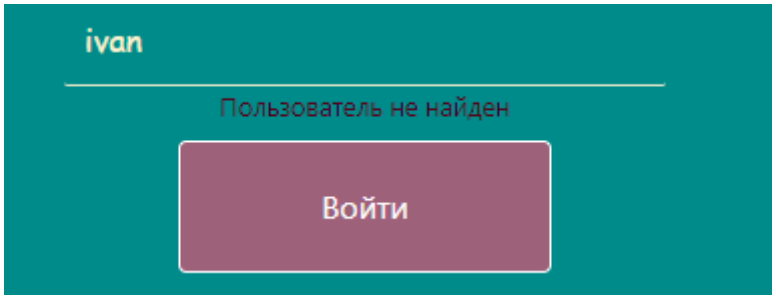

Рисунок 1.3.

После нажатия на кнопку «Войти» за пользователя до начала аукциона, будет то, что на рис. 2.

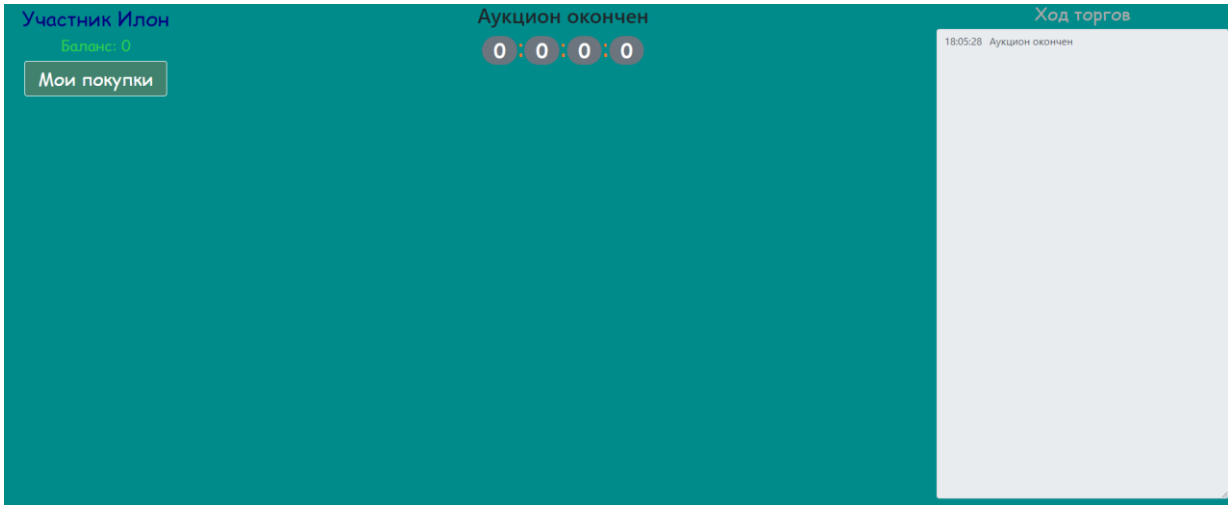

Рисунок 2

Также есть возможность зайти за админа, для этого в поле имени пользователя нужно ввести admin (рис. 3). Здесь реализовано окно «Ход торгов» с использование jQuery UI (resizable). Есть список участников,

перечень картин, а также кнопка «Обнуление», после нажатия на которую обновляется информация о покупателе (см. рис. 4).

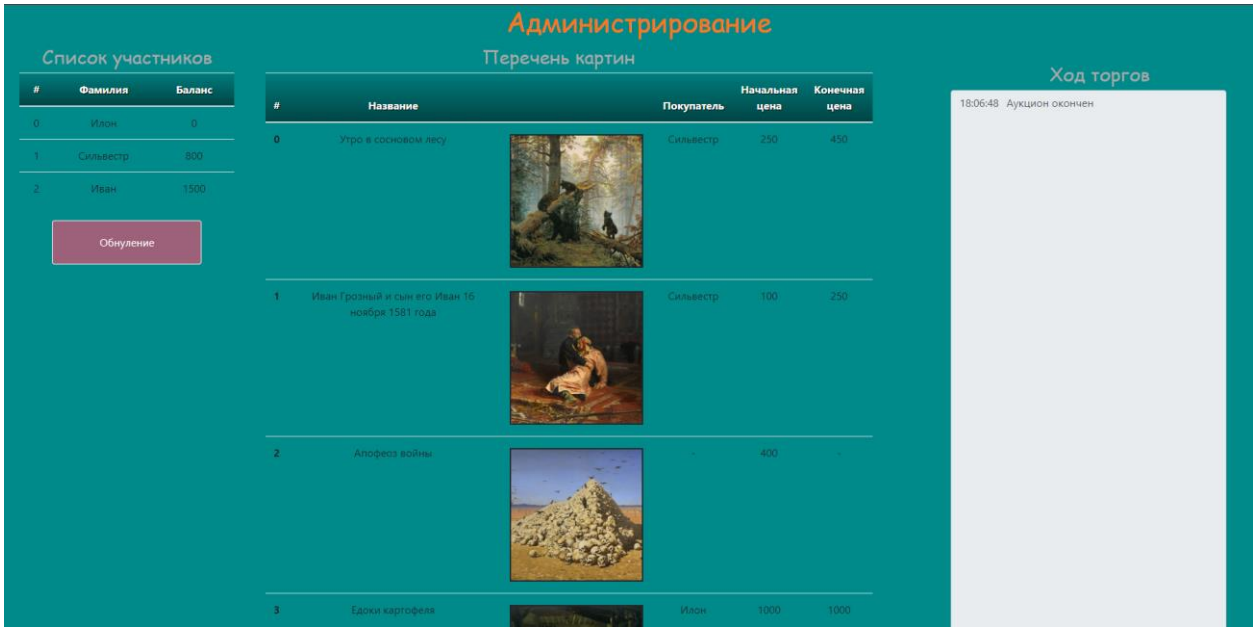

Рисунок 3

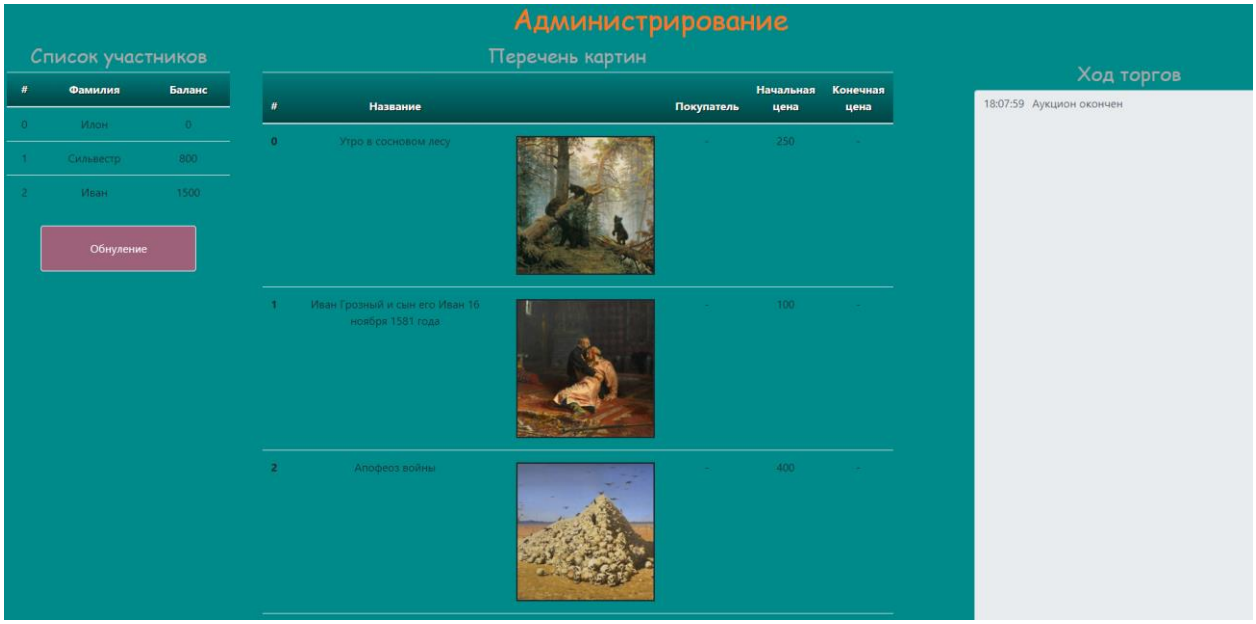

Рисунок 4

Теперь, если изменить дату начала аукциона на время, которое еще не настало, то будет писаться следующее (рис. 5).

Рисунок 5

Когда время до начала аукциона закончится, появится сам аукцион (рис. 6).

Рисунок 6 (сверху обрезано, то, что сверху, можно посмотреть на других скриншотах)

Так как у пользователя Илон баланс равен 0, то он не может сделать ставку на картину (рис. 7).

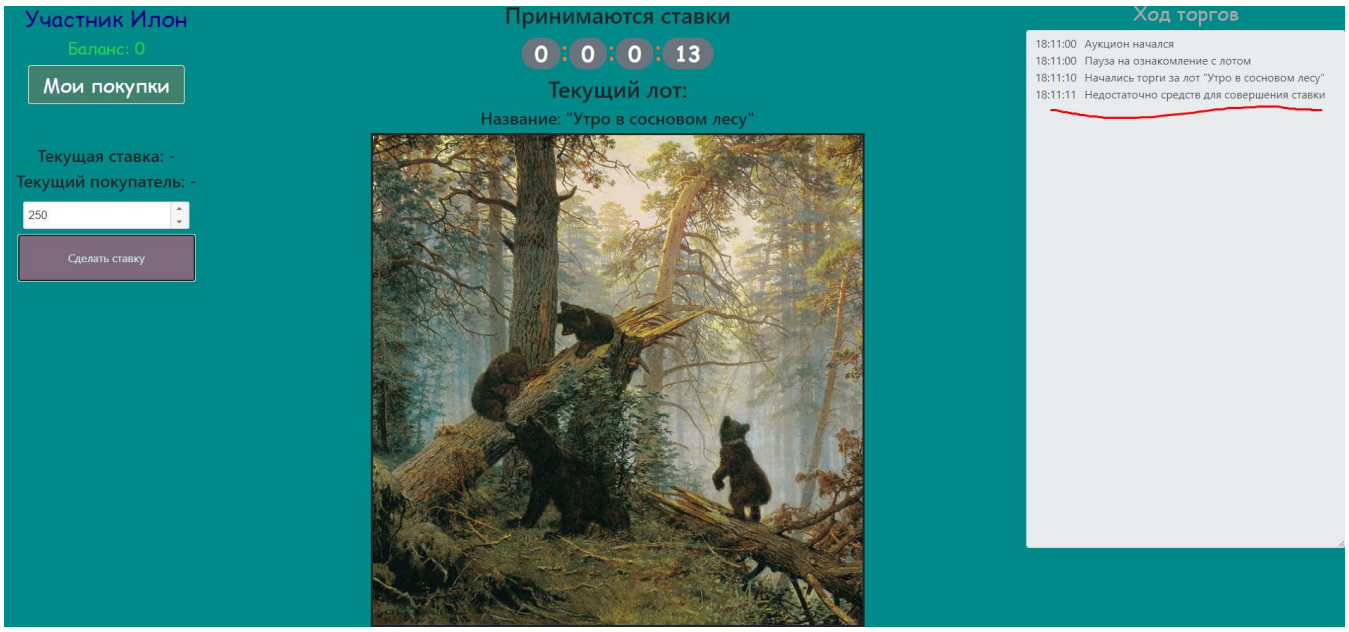

Рисунок 7

Но может сделать ставку Сильвестр с балансом 800 (рис. 8).

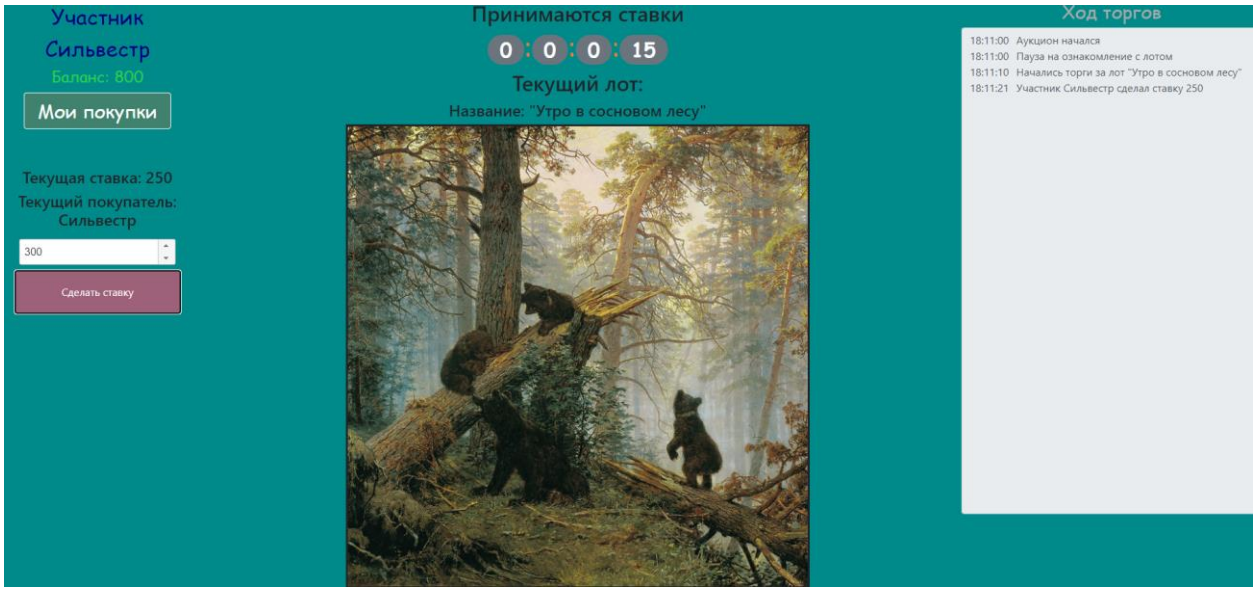

Рисунок 8

Если сделать ставку за Сильвестра, то его ставку может перебить Иван (рис. 9 и 10).

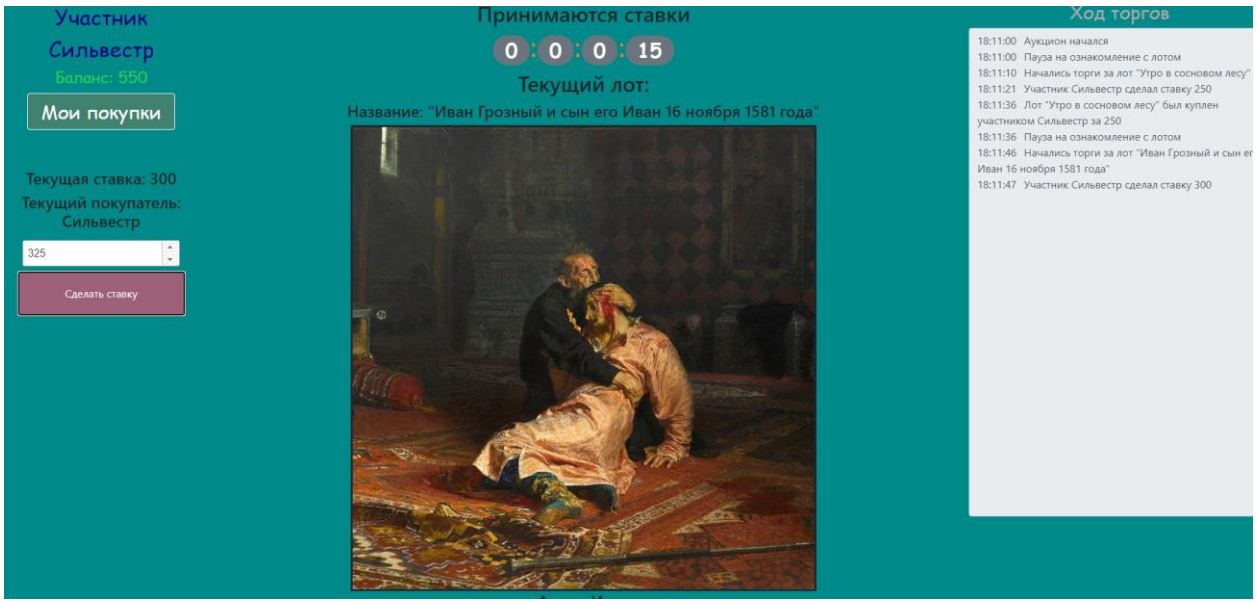

Рисунок 9

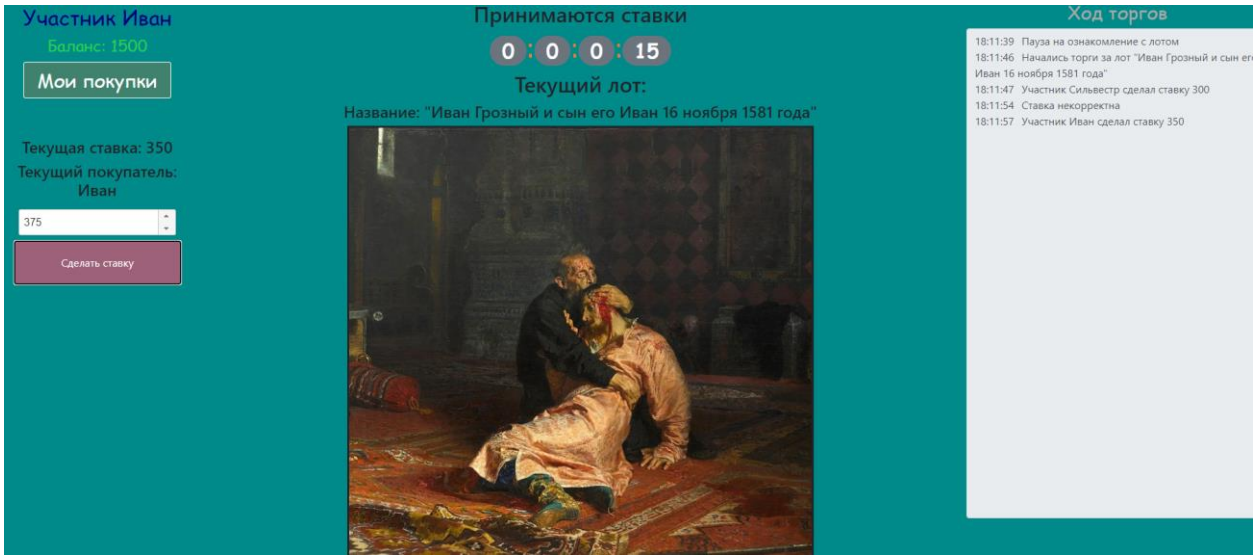

Рисунок 10

Минимальный и максимальные шаги также учтены. Затем, можно посмотреть информацию за администратора (рис. 11).

Рисунок 11

Если зайти в «мои покупки» у Ивана.

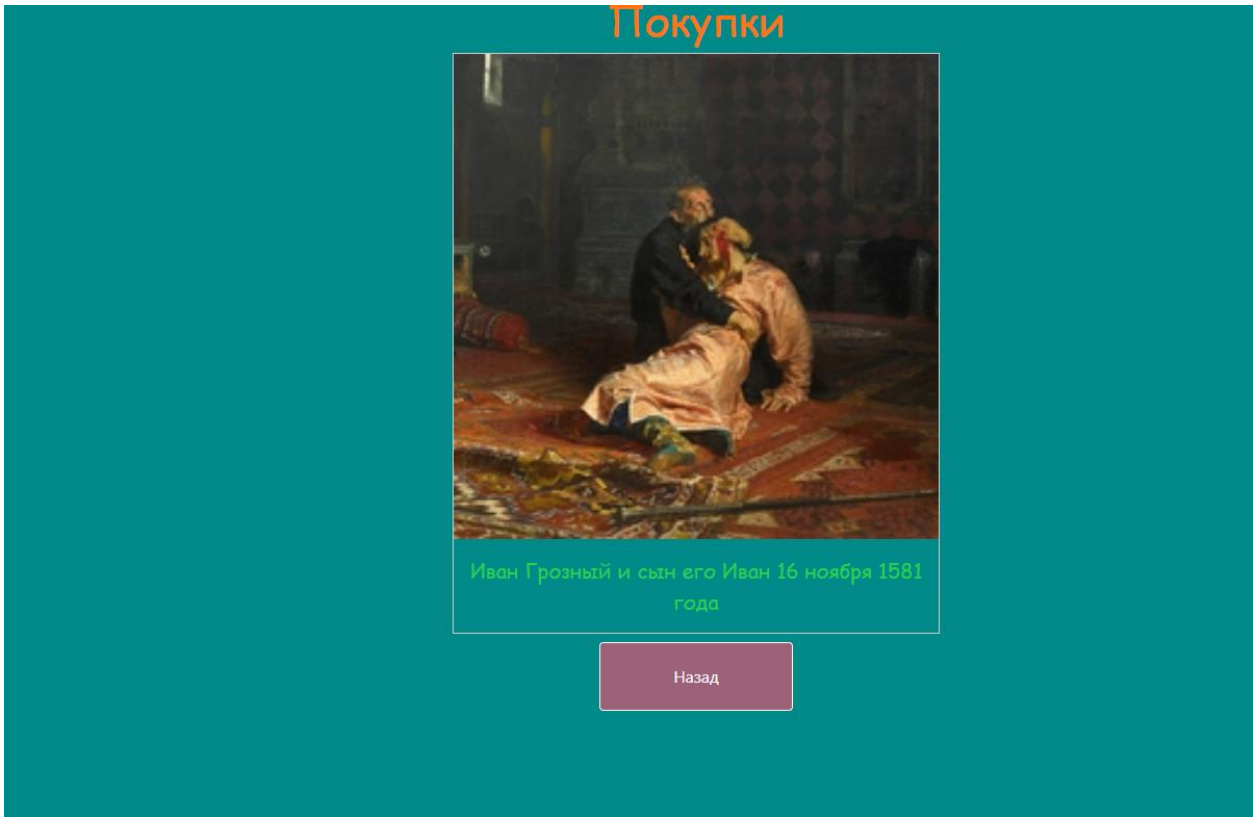

Рисунок 12

Выводы.

В ходе выполнения лабораторной работы были изучены возможности применения jQuery UI для создания интерфейса пользователя, обработки ошибок, ведения журналов ошибок, реализовано взаимодействие

приложения с использование web-сокетов, применен статический анализатор Flow, освоен инструмент сборки WebPack, и организовано модульное тестирования web-приложений с использованием Mocha.