**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №4**

**«Модуль приложения «Участие в аукционе картин»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Ишанина Л.Н. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2020

## Цель работы.

Целью работы является изучение возможностей применения jQuery UI для создания интерфейса пользователя, обработки ошибок, ведение журналов ошибок, реализация взаимодействия приложений с использованием webсокетов, применение статического анализатора Flow, освоение инструмента сборки WebPack и организации модульного тестирования web-приложений с использованием Mocha.

## Задание.

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 jsфайлов.

4) Разработка клиентских js-файлов с использованием библиотек jQuery и jQuery UI.

5) Обработка ошибок на сервере с использованием try/catch, функции catch() у промисов и механизма «domain».

6) Регистрация и подключение в web-приложение журнала ошибок.

7) Создание web-сокета для отправки сообщений всем клиентам.

8) Использование статического анализатора Flow для анализа кода.

9) Создание и запуск Mocha тестов для приложения. 10) Конфигурирование сборщика WebPack.

## Выполнение работы.

## Настройка сервера Express, router, стилизация.

Для настройки сервера был создан проект в WebStorm для веб-приложения на Node JS Express. Были продуманы следующие GET запросы для маршрутизации на сайте, обрабатываемые со стороны сервера с помощью express-router:

* / - страница авторизации
* /admin – панель управления администратора со списком всех пользователей (и информацией о них), списком картин и их состояния, панелью сообщений
* /main – панель управления пользователя со списком доступных для покупки картин, списком приобретенных картин (если аукцион закончился), информацией о пользователе и его счете, панелью сообщений

В качестве набора стилей был использован Bootstrap.

## Обработка и вывод данных.

Данные о пользователях, картинах и настройках изначально хранятся в JSON файлах (paint.json, person.json, setting.json). Сервер загружает данные в память и изменяет уже только в JSON-объекте. Настройки сохраняются в переменные, отслеживается наступление времени начала аукциона.

Соединение клиента с сервером реализовано с помощью socket-io.

## Обработка ошибок на сервере

На сервере были созданы конструкции try/catch для обработки ошибок.

## Использование Flow, Mocha, WebPack

Был установлен и сконфигурирован Flow. Кроме того, анализатор кода Flow был включен в среде разработки WebStorm. Имеются некоторые ошибки, указанные Flow, однако разрешенные в рамках ES5. Также с помощью Mocha были проведены тесты. Для сборки проекта использовался WebPack.

## Тестирование

Страница входа в приложения представлена на рис. 1.

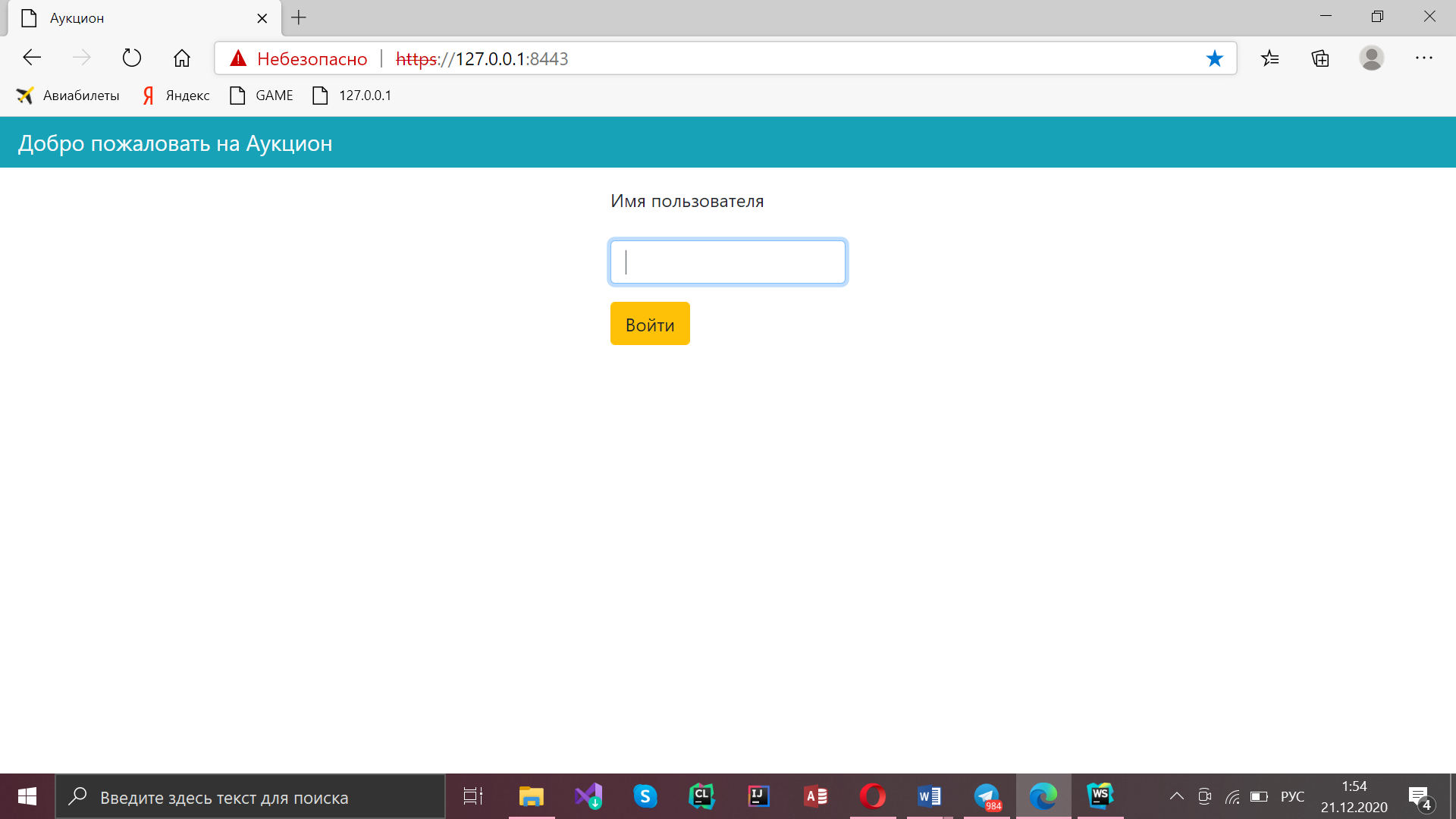


Рисунок 1 – Страница входа в приложение

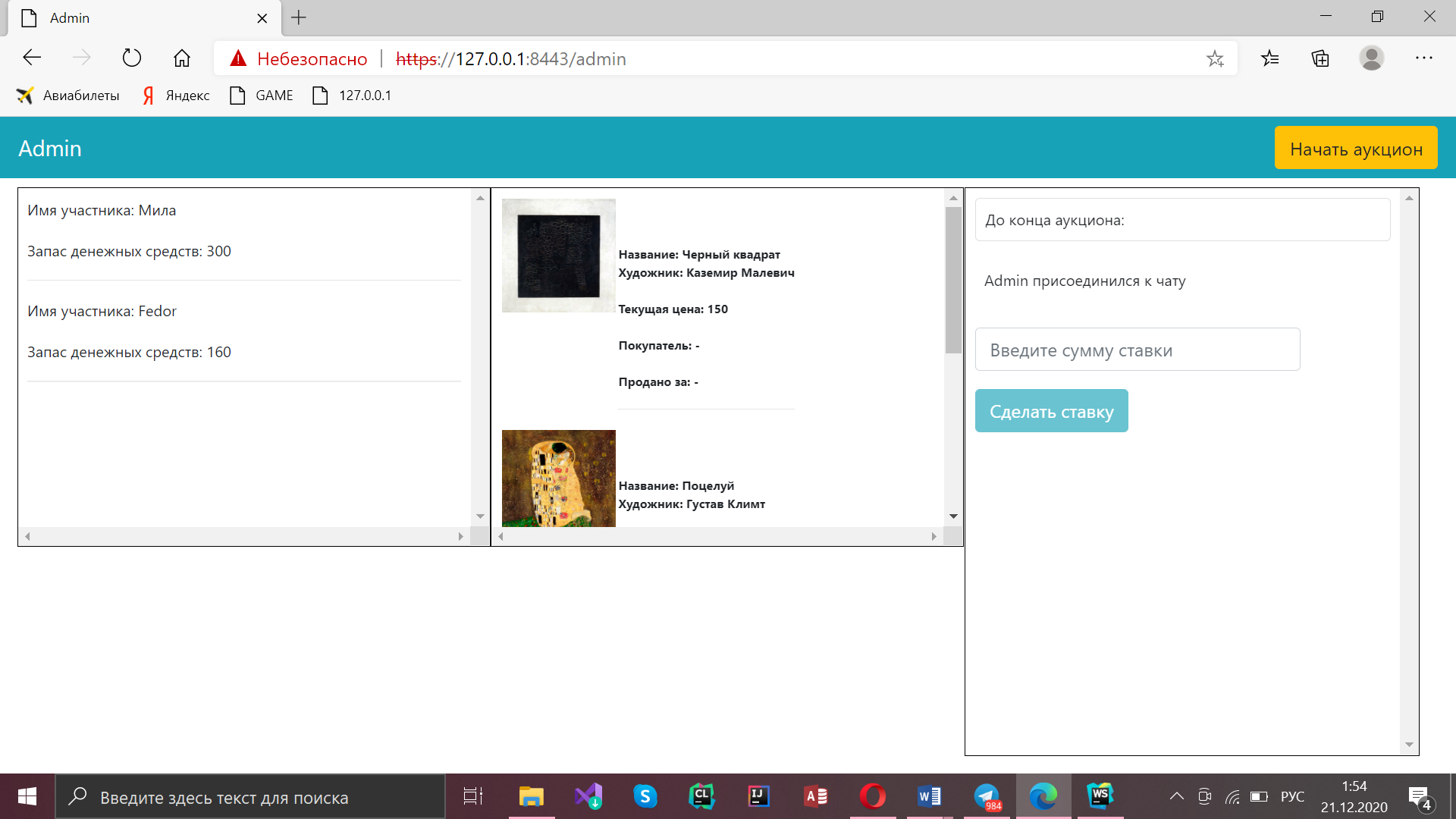


Рисунок 2 – Страница с панелью администратора

Страница пользователя представлена на рис. 4.

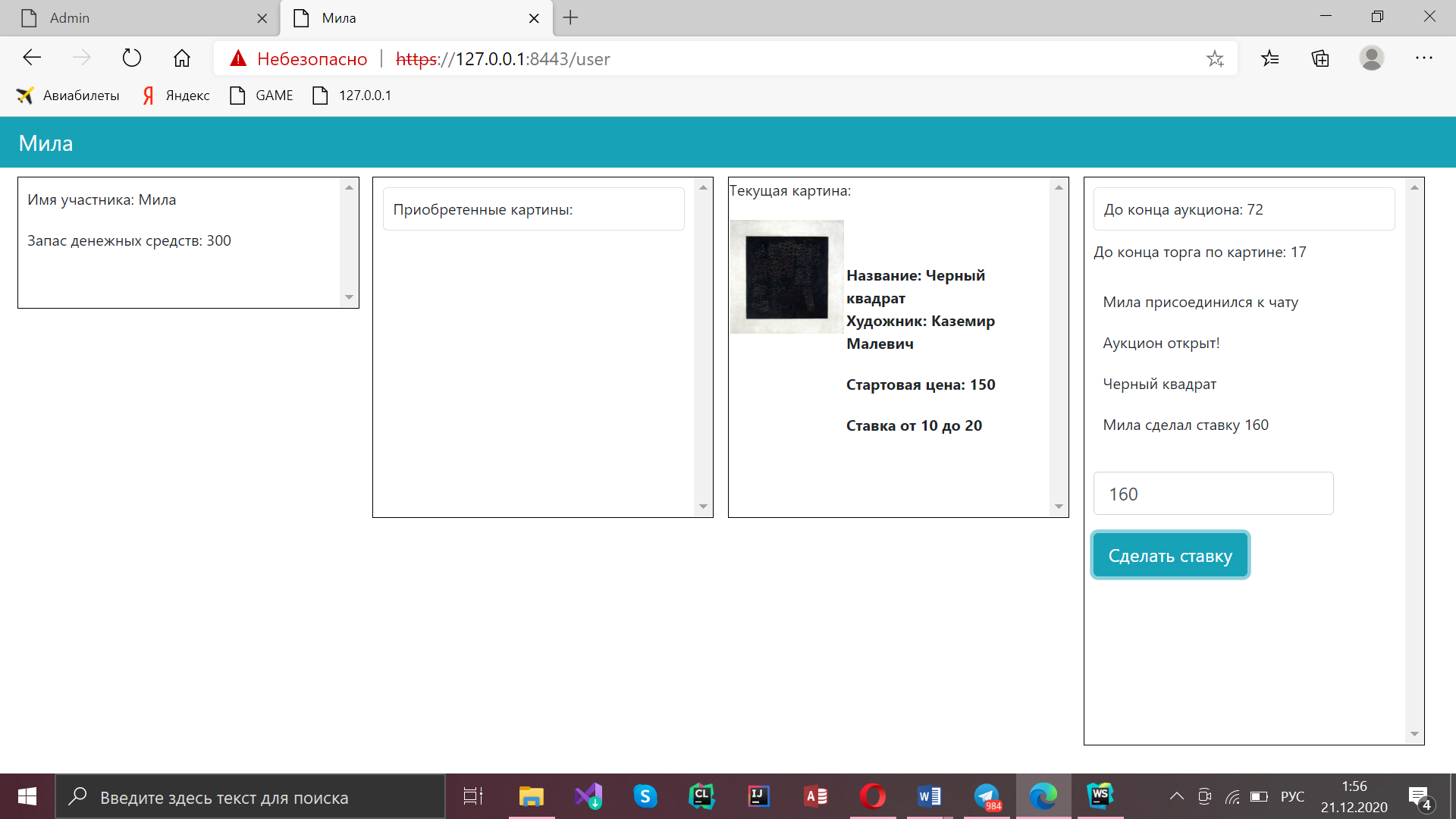


Рисунок 3 – Страница пользователя, который во время аукциона сделал ставку

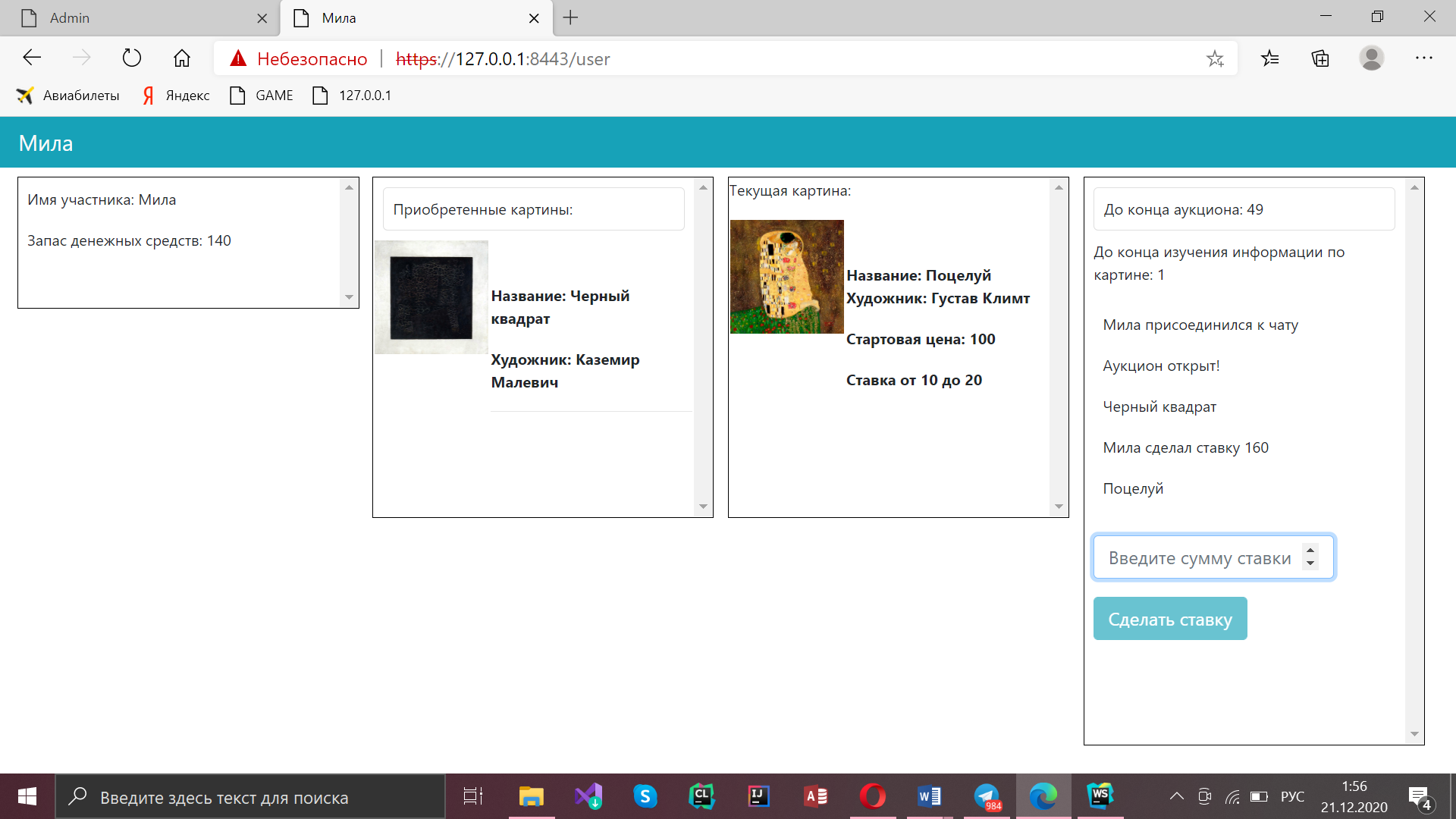


Рисунок 4 – Страница пользователя, который приобрел картину

## Выводы.

В результате выполнения лабораторной работы были изучены возможности применения jQuery UI для создания интерфейса пользователя, обработки ошибок, ведение журналов ошибок, реализованы взаимодействия приложений с использованием webсокетов. А также был применен статический анализатор Flow, освоен инструмент сборки WebPack и было организовано модульное тестирование web-приложения с использованием Mocha.