

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики
Кафедра математических методов исследования операций

Разработка аукциона на основе технологии блокчейн

Дипломная работа

Направление 01.03.02 – Прикладная математика и информатика

Профиль – Информационная аналитика и обработка больших данных

Зав. кафедрой _____

д. т. н., проф. Азарнова Т. В.

Обучающийся _____

4 курс, 7 группа Лисовский А. В.

Руководитель _____

к. ф.-м. н., ст. преп. Куркин Е.В.

Воронеж 2023

Аннотация

Данная дипломная работа посвящена анализу, проектированию и разработке аукциона на основе технологии блокчейн.

В работе определены основные этапы разработки аукциона: проведен анализ предметной области, осуществлен анализ существующих решений на основе технологии блокчейн, сформированы требования к дизайну и функциональности разрабатываемого приложения, описаны основные функции и логика работы приложения. На основе проведенного анализа разработано и протестировано приложение для проведения аукциона на основе технологии блокчейн. В ходе работы было уделено внимание проблемам безопасности и целостности информации, которые обеспечиваются технологией блокчейн. Результаты данной работы могут быть использованы для разработки подобных приложений и проведения аукционов на платформе блокчейн.

Содержание

Аннотация	2
Список сокращений	5
Введение	6
Глава 1. Постановка задачи и анализ существующих решений.....	8
1.1 Постановка задачи	8
1.2 Теория аукционов.....	8
1.3 Анализ существующих решений интернет аукционов	9
Глава 2. Изучение структуры приложения для проведения блокчейн-аукционов и анализ средств его реализации	14
2.1 Анализ структуры приложения	14
2.2 Структура пользовательского интерфейса	16
2.3 Средства реализации.....	17
2.4 Требования к аппаратному и программному обеспечению	17
Глава 3. Анализ функциональности и логики работы приложения для проведения блокчейн-аукционов.....	19
3.1 Разбор функциональности приложения	19
3.2 Архитектура приложения.....	21
3.3 Описание экосистемы NEAR Protocol	23
Глава 4. Реализация приложения для проведения блокчейн-аукционов	26
4.1 Интерфейс пользователя	26
4.2 Описание серверной части приложения	35
4.3 Описание клиентской части.....	40
4.4 План тестирования приложения	44

Заключение	46
Список литературы	48
Приложение 1. Листинг серверной части приложения.....	50
Приложение 2. Листинг клиентской части приложения.....	51

Список сокращений

Смарт-контракт – компьютерный алгоритм, предназначенный для формирования, управления и предоставления информации о владении чем-либо. В более узком смысле смарт-контракт – это набор функций и данных, находящихся по определённому адресу в блокчейне;

PoS – это механизм консенсуса в блокчейн-системах, который используется для достижения согласия и подтверждения транзакций. Он опирается на концепцию "доказательства доли" или "доли владения";

NEAR – это открытая блокчейн-экосистема, которая используется для создания децентрализованных приложений и сервисов. Она была специально разработана с учетом масштабируемости, безопасности и удобства использования, чтобы сделать технологию блокчейн доступной для массового применения;

Введение

Технология блокчейн – это усовершенствованный механизм базы данных, основанный на распределенной системе данных, который позволяет организовать свободный и открытый обмен информацией в рамках одной бизнес-сети [6]. База данных блокчейна содержит информацию обо всех транзакциях, проведенных участниками системы. Информация хранится в виде цепочки блоков [7].

Централизованный характер аукционов привел к ряду проблем, включая отсутствие прозрачности, уязвимости в системе безопасности и споры по поводу владения. Технология блокчейн, с ее децентрализованным и прозрачным характером, может решить эти проблемы, создав надежную и безопасную среду для онлайн-аукционов. Используя смарт-контракты, аукционы могут быть автоматизированы, а процесс торгов можно сделать прозрачным и защищенным от несанкционированного доступа.

Целью данной дипломной работы является проектирование и разработка онлайн-аукциона на основе технологии блокчейн, а также анализ различных блокчейн-технологий для реализации внутренних процессов аукциона.

Для выполнения данной цели были поставлены следующие задачи:

- выбор средств реализации;
- разработка структуры веб-приложения для проведения аукционов;
- проведение анализа функциональности и составление логики работы веб-приложения;
- написание смарт-контракта веб-приложения для проведения аукционов;
- разработка модели машинного обучения для прогнозирования цен на поддержанные автомобили;
- разработка и реализация веб-приложения для проведения аукционов на основе технологии блокчейн;

- составление плана тестирования разработанного веб-приложения для проведения аукционов.

Глава 1. Постановка задачи и анализ существующих решений

1.1 Постановка задачи

Необходимо разработать веб-приложение – аукцион, основанный на технологии блокчейн, с целью обеспечения безопасности, прозрачности и надежности процесса проведения аукционов. Приложение предоставляет следующие возможности:

Возможности доступные не авторизованным пользователям:

- аутентификация пользователей через систему NEAR Wallet;
- просмотр списка созданных лотов;
- просмотр истории ставок на выбранном лоте;
- просмотр текущей (наивысшей) ставки для выбранного лота;

Возможности доступные авторизованным пользователям:

- просмотр баланса кошелька NEAR Wallet;
- возможность создания лотов на аукционе;
- закрытие или отмена лота на аукционе, созданного текущим пользователем;
- просмотр списка созданных лотов;
- размещение ставок на выбранном лоте;
- просмотр истории ставок на выбранном лоте;
- просмотр прогнозируемой цены поддержанного автомобиля по введенным пользователем данным;
- просмотр текущей (наивысшей) ставки для выбранного лота.

1.2 Теория аукционов

Теории аукционов посвящен отдельный раздел теории игр и экономики. К теории аукционов можно подходить с разных точек зрения – с точки зрения теории игр (аукционы – это байесовские игры с неполной информацией), теории проектирования контрактов и механизмов (аукционы – это механизмы распределения), микроструктуры рынка (аукционы – это модели

ценообразования), а также в контексте различных приложений (закупки, лицензирование патентов, государственные финансы и т.д.) [8].

В данной работе будет рассмотрена практическая реализация аукциона английского типа. В данном виде аукционов все участники торгов начинают участие в аукционе с нулевой цены. Цена увеличивается с небольшими приращениями пока не останется только один заинтересованный участник торгов. Аукцион продолжается до тех пор, пока не останется ни одного участника, готового повысить ставку. Тот участник, который предложил наивысшую цену, становится победителем аукциона и получает право купить товар по предложенной им цене [9].

Аукцион английского типа широко применяется в различных отраслях и сферах, включая торговлю недвижимостью, автомобилями, искусством, антиквариатом и другими ценностями. Он позволяет определить рыночную цену товара или услуги на основе конкуренции между покупателями и создает возможность для получения максимальной выручки для продавца.

1.3 Анализ существующих решений интернет аукционов

В процессе реализации веб-приложения рассматривались существующие решения интернет-аукционов, позволяющих создавать лоты. Так, были отобраны 3 самых подходящих веб-приложения для создания лотов на аукционе. Далее, рассмотрим каждый из отобранных примеров и выявим их достоинства и недостатки.

Проведем анализ платформы auction.ru [3]. Данная интернет-площадка для проведения аукционов обладает широким спектром возможностей по созданию различного рода лотов.

К плюсам данного сервиса относятся:

- возможность создания лотов различных типов;
- для создания лота не нужно регистрировать отдельный аккаунт продавца.

К минусам данного сервиса относятся:

- отсутствуют смарт-контракты для подтверждения сделок;
- долгий срок регистрации и верификации аккаунта;
- присутствует комиссия с продажи лота.

Главная страница платформы изображена на рис. 1.1.

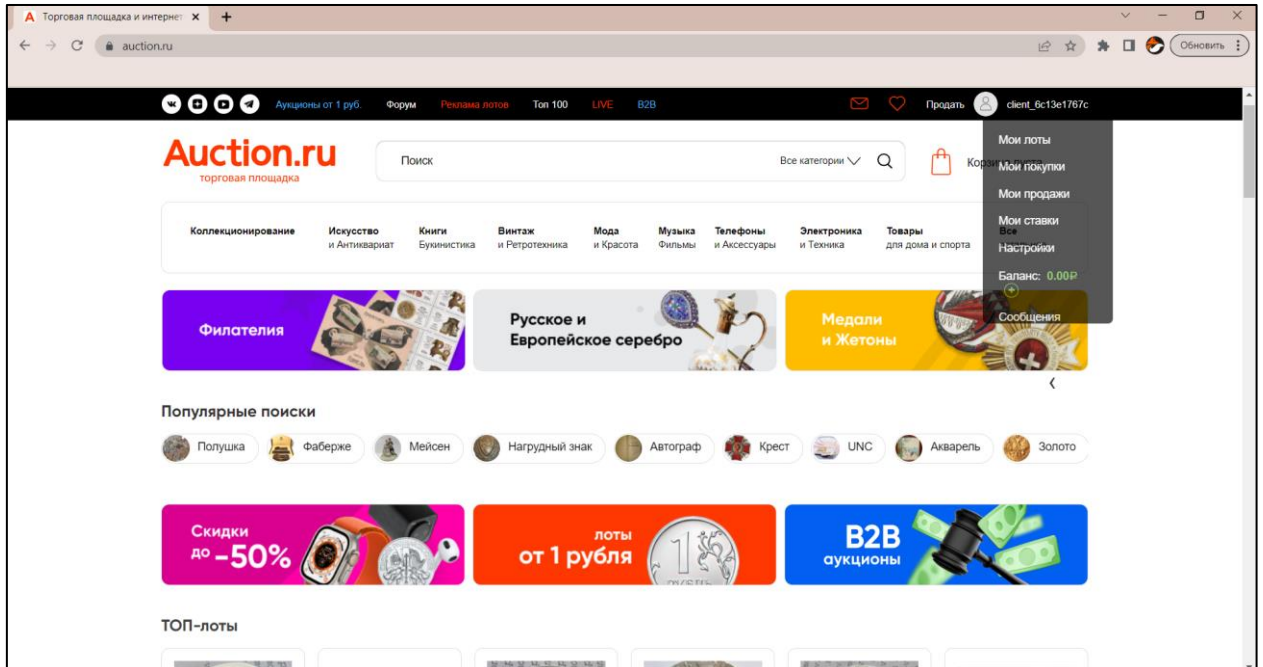


Рис. 1.1. Главная страница платформы Auction.ru

Далее, рассмотрим платформу ebay.com [4]. Данный сервис является самым крупным и самым популярным онлайн-аукционом среди представленных платформ.

Плюсами данного сервиса являются:

- возможность создания лотов различных типов;

Минусами данного сервиса являются:

- отсутствуют смарт-контракты для подтверждения сделок;
- для создания лота нужно регистрировать отдельный аккаунт продавца;
- долгий срок регистрации и верификации аккаунта;
- присутствует комиссия с продажи лота.

Главная страница сервиса изображена на рис. 1.2.

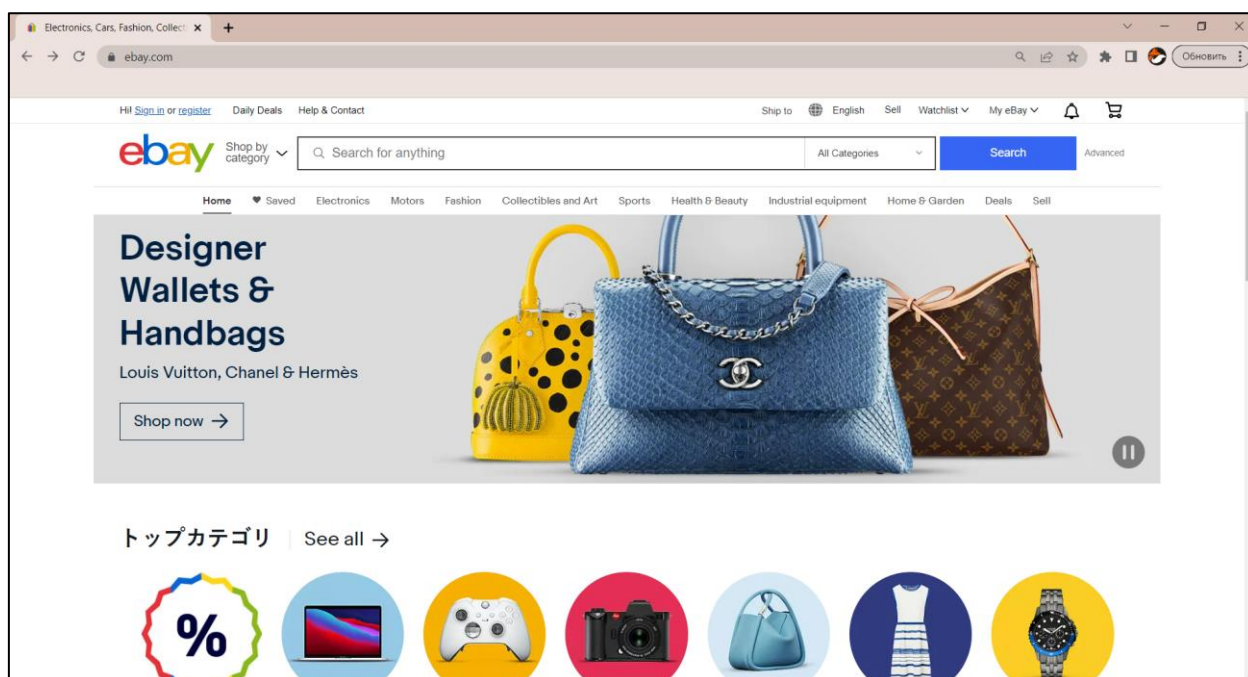


Рис. 1.2. Главная страница платформы Ebay.com

Рассмотрим следующий сервис. Opensea.io [5] является ведущей и наиболее популярной платформой для обмена и торговли цифровыми активами, такими как нон-фангибл-токены (NFT), а также создания аукционов этих токенов.

Это единственная платформа, основанная на технологии блокчейн, из трех представленных сервисов. Ее ключевая особенность – пользователи могут продавать только цифровые активы, такие как виртуальное искусство, виртуальные коллекционные предметы, виртуальные земли, виртуальные персонажи и многое другое.

Плюсами данного сервиса являются:

- используются смарт-контракты для подтверждения сделок;
- для создания лота не нужна верификация, для этого достаточно наличие криптовалютного кошелька;
- отсутствует комиссия с продажи лота.

Минусами данного сервиса являются:

- отсутствует возможность создания лотов различных типов, поддерживается продажа только цифровых активов;

- для создания лота нужно регистрировать отдельный аккаунт продавца.

Главная страница сервиса изображена на рис. 1.3.

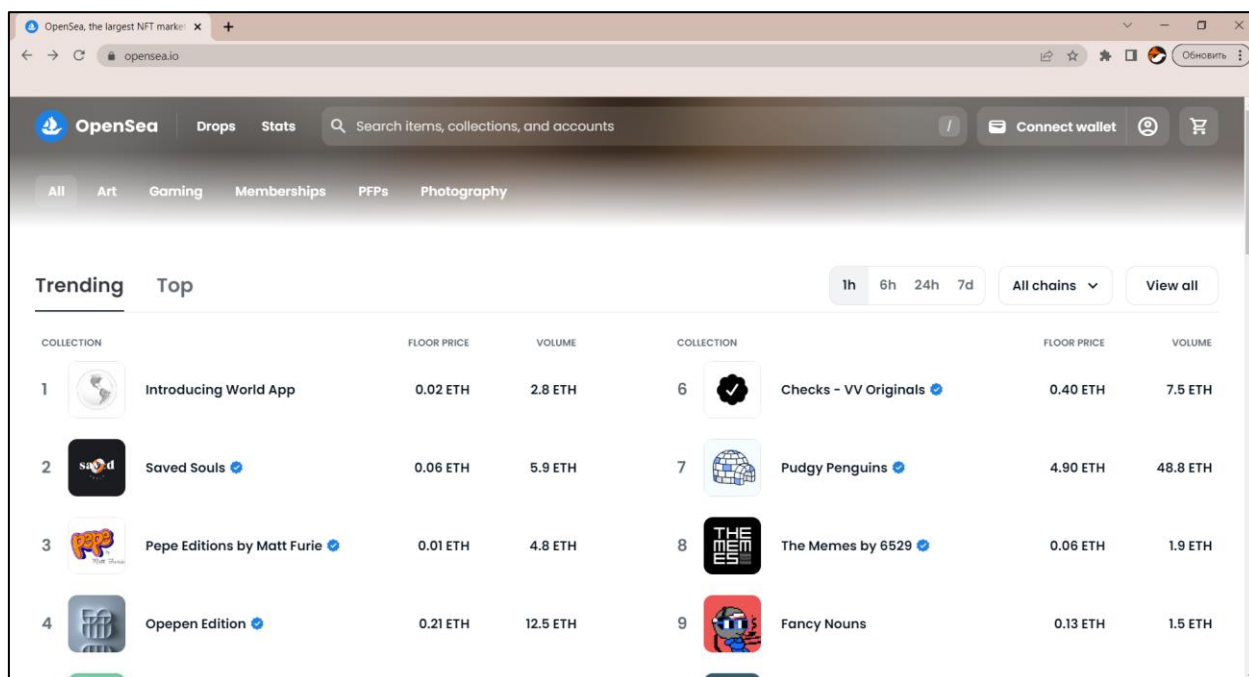


Рис 1.3. Главная страница платформы Opensea.io

Для оценки приложений была составлена таблица (табл. 1), в которой перечислены преимущества и недостатки каждого из рассмотренных сервисов:

Таблица 1. Результаты анализа существующих решений

	Создание различного рода лотов	Используются смарт-контракты для подтверждения сделок	Для создания лотов необходимо регистрировать отдельный аккаунт продавца	Долгий срок регистрации и верификации	Присутствует комиссия с продажи лота
Auction.ru	+	—	—	+	+
Ebay.com	+	—	+	+	+
Opensea.io	—	+	+	—	—

Все рассмотренные варианты обладают достаточным набором хороших средств для проведения онлайн-аукционов. Однако, каждая из платформ имеет свои преимущества и недостатки. Например, Auction.ru [3] и Ebay.com [4] позволяют создавать различные типы лотов, но не используют смарт-

контракты и требуют регистрацию отдельного аккаунта продавца. С другой стороны, Opensea.io [5] использует смарт-контракты, но может иметь длительный процесс регистрации и верификации, а также взимает комиссию с продажи лота.

Исходя из этого, появляется потребность в создании бесплатной платформы, которая будет предоставлять базовую функциональность для создания и продажи лотов, удобную для пользователей. Для реализации такой платформы необходимо учесть все необходимые требования онлайн-аукционирования.

Глава 2. Изучение структуры приложения для проведения блокчейн-аукционов и анализ средств его реализации

2.1 Анализ структуры приложения

Рассмотрим главную страницу приложения. После запуска приложения, пользователю отображается главная страница. На главной странице выводится список всех открытых лотов на аукционе, а также их краткое описание, изображение, статус и наивысшая текущая ставка.

Если пользователь не авторизован, ему потребуется пройти аутентификацию при помощи кошелька «NEAR Wallet», для дальнейшей работы в системе. Для этого ему доступна кнопка «Войти в NEAR Wallet» в правом верхнем углу экрана. При нажатии на кнопку пользователь будет направлен на страницу с аутентификацией через кошелек «NEAR Wallet».

Не авторизованному пользователю доступен частичный функционал системы: он может просматривать открытые лоты на аукционе со всей детальной информацией, но не может делать ставки, размещать свои лоты, отменять или удалять их, а также просматривать историю всех закрытых или отмененных лотов на аукционе.

После успешной авторизации, пользователь автоматически будет перенаправлен на главную страницу приложения.

Если пользователь уже авторизован в системе, то он обладает расширенным функционалом: он может спрогнозировать цену продажи поддержанного автомобиля, может просматривать историю всех закрытых или отмененных лотов, делать ставки, используя свой кошелек «NEAR Wallet», а также создавать свои лоты, отменять или удалять их при необходимости. Кроме того, в «шапке» сайта для авторизованного пользователя доступно отображение текущего состояния кошелька.

Реализация главной страницы приложения должна включать методы:

- метод для получения текущего пользователя системы;

- метод для получения информации о состоянии кошелька пользователя;
- метод для отображения списка открытых лотов аукциона с их детализированной информацией и текущим состоянием;

Перейдем к странице с аутентификацией пользователя в кошельке NEAR. Реализация данной страницы должна включать следующие методы:

- метод для получения текущего пользователя системы;
- метод для аутентификации текущего пользователя приложения через кошелек «NEAR Wallet»;

Далее, рассмотрим страницу создания нового лота аукциона. Она включает следующие методы:

- метод для добавления нового лота аукциона;

Страница со списком лотов текущего пользователя должна включать следующие методы:

- метод для отображения списка всех лотов, созданных текущим пользователем;
- метод для отмены лота, созданного текущим пользователем приложения;
- метод для завершения лота, созданного текущим пользователем приложения;

Страница с подробной детализацией выставленного лота включает следующие методы:

- метод для отображения информации о лоте аукциона;
- метод для отображения изображения лота аукциона;
- метод для отображения информации о текущей самой высокой ставке лота;
- метод для отображения истории ставок лота аукциона;
- метод для добавления новой ставки;

Перейдем к странице с подтверждением ставки в кошельке NEAR. Она включает:

- метод для получения текущего пользователя;
- метод для добавления новой ставки в лот;
- метод для замены текущей наивысшей ставки лота на новую ставку;
- метод для добавления новой ставки в историю ставок лота;

Страница с историей всех лотов аукциона содержит следующие методы:

- метод для получения текущего пользователя;
- метод для получения списка всех закрытых или отмененных лотов аукциона;

Страница с анализом цены автомобиля включает следующий метод:

- метод для предсказания цены автомобиля.

2.2 Структура пользовательского интерфейса

Интерфейс пользователя должен включать следующие страницы:

- главную страницу приложения со списком всех открытых лотов аукциона и их кратким описанием, а также текущим балансом кошелька пользователя, если он авторизован;
- страницу входа в приложение при помощи кошелька «NEAR Wallet»;
- страницу с созданием нового лота на аукционе, на данной странице доступна следующая форма:
 - форма создания нового лота аукциона;
- страницу со списком лотов, созданных текущим пользователем, а также с возможностью просмотра, отмены или завершения таких лотов;
- страницу с детализированной информацией о лоте, изображением лота, на данной странице доступны следующие формы:
 - форма с отображением истории ставок для конкретного лота;

- форма с отображением информации о текущей наивысшей ставке;
- форма с добавлением новой ставки;
- страницу с подтверждением ставки в кошельке «NEAR Wallet»;
- страницу с возможностью просмотра истории всех закрытых или отменных лотов аукциона;
- страницу с возможностью просмотра прогнозируемой цены поддержанного автомобиля.

2.3 Средства реализации

Для разработки приложения выбраны следующие программные средства:

- среда разработки WebStorm;
- язык программирования TypeScript;
- язык программирования Python;
- экосистема NEAR Protocol;
- React – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов;
- Pickle – Python-библиотека для сериализации и десериализации объектов Python;
- Scikit-Learn – Python-библиотека, предназначенная для машинного обучения, написанная на языке программирования Python и распространяемая в виде свободного программного обеспечения;
- система контроля версий Git.

2.4 Требования к аппаратному и программному обеспечению

Требования к программному обеспечению клиента включают:

- пользователь должен иметь установленный браузер, основанный на движке Chromium или WebKit, такой как Google Chrome версии 49 и выше или Safari версии 9.2 и выше;
- для использования Google Chrome, требуются операционные системы Windows XP или новее, Mac OS X начиная с версии 10.6, а также Ubuntu с версии 10.4. Для обеспечения совместимости с приложением, рекомендуется наличие процессоров Intel Pentium 4 или Athlon 64 и более поздних версий;
- для корректной работы веб-приложения необходим доступ к сети Internet;

Требования к серверному программному обеспечению включают:

- наличие операционной системы, поддерживающей работу с NEAR SDK и TypeScript, а также с языком программирования Python;
- доступ к сети Internet;
- на стороне сервера должны присутствовать необходимые конфигурационные файлы и настройки сервера для правильной работы приложения.

Глава 3. Анализ функциональности и логики работы приложения для проведения блокчейн-аукционов

3.1 Разбор функциональности приложения

В результате анализа была определена основная функциональность, которую необходимо реализовать в приложении:

- Аутентификация:
 - Предоставляется возможность входа в систему с использованием кошелька NEAR;
- Управление пользовательским аккаунтом:
 - Авторизованные пользователи могут просматривать информацию о своем аккаунте, такую как уникальный идентификатор и текущий баланс кошелька;
- Просмотр списка открытых аукционов:
 - Все пользователи могут просматривать список доступных аукционов;
 - Предоставляется краткая и подробная информация о каждом аукционе, такая как название, изображение лота создатель лота и т.д.;
- Просмотр истории аукционов:
 - Авторизованные пользователи могут просматривать список завершенных или отмененных аукционов;
 - О них также предоставляется краткая и подробная информация;
- Создание аукциона:
 - Авторизованные пользователи имеют возможность создавать новые аукционы;
 - Для создания аукциона необходимо указать название лота и URL его изображения;
- Размещение ставки на аукционе:

- Авторизованные пользователи могут участвовать в аукционах, размещая ставки на лоты;
- Для подачи ставки необходимо указать сумму ставки в валюте NEAR;
- Отмена аукциона:
 - Авторизованные пользователи могут отменять свои созданные аукционы;
 - Отмена аукциона прекращает прием ставок и удаляет аукцион из списка открытых аукционов;
- Завершение аукциона:
 - После завершения аукциона система прекращает прием ставок и определяет победителя, основываясь на наивысшей ставке;
- Анализ цены поддержанного автомобиля по введенным параметрам:
 - Авторизованные пользователи могут перейти на соответствующую вкладку приложения, заполнить форму и узнать прогнозируемую цену поддержанного автомобиля;
- Валидация данных:
 - При вводе пользователем некорректных данных в поля данных веб-приложение предоставляет обратную связь и выводит сообщения об ошибках.

Далее, рассмотрим непосредственно сам процесс работы приложения на основе описанной выше функциональности. Описание процесса работы веб-приложения представляет собой описание последовательности действий, которые выполняются при взаимодействии пользователя с приложением. Ниже, описан обобщенный процесс работы веб-приложения. В данном обобщенном описании уделяется внимание основным этапам процесса, без углубления в детали реализации (рис. 3.1):

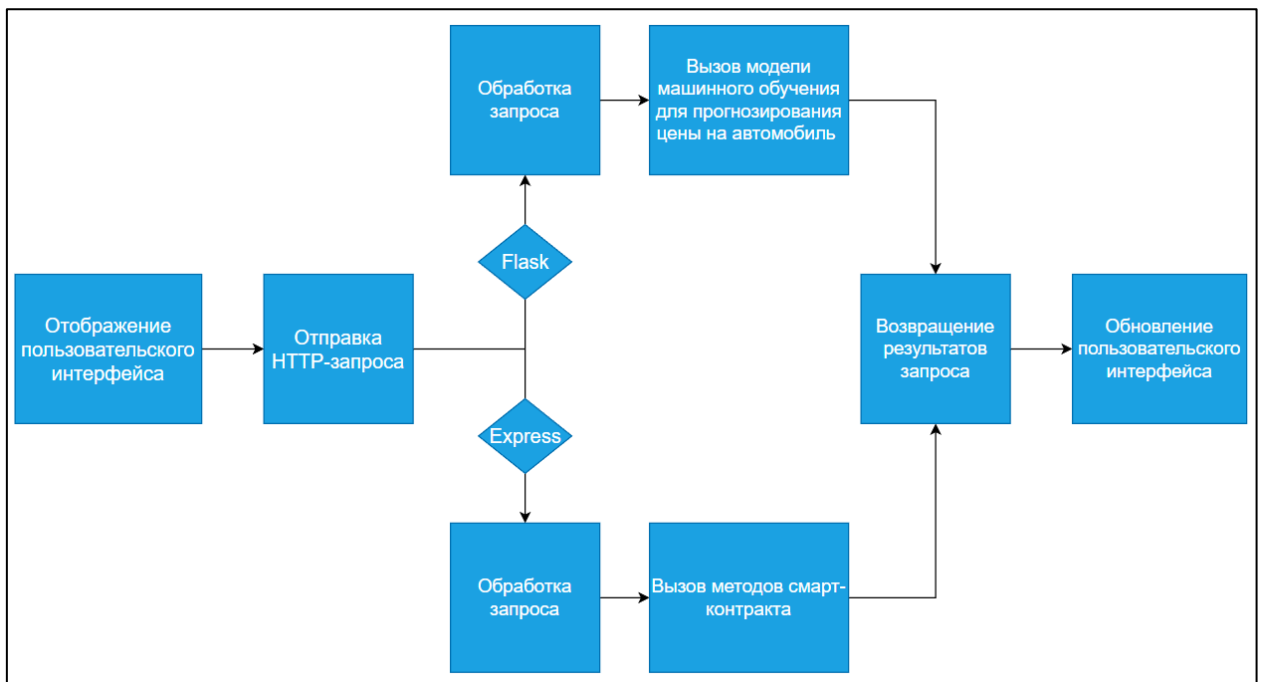


Рис. 3.1 Обобщенный процесс работы веб-приложения

3.2 Архитектура приложения

Логика работы веб-приложения основана на взаимодействии между Frontend-клиентом, написанным с использованием библиотеки React, Backend-сервером, написанным с использованием библиотеки Express и смарт-контрактом. React-клиент обеспечивает пользовательский интерфейс и отправляет запросы на Express-сервер. Express-сервер выполняет проверку и обработку запросов, а также взаимодействует со смарт-контрактом для выполнения операций в блокчейне NEAR. Смарт-контракт управляет логикой аукциона, хранит данные и обрабатывает транзакции в блокчейне NEAR.

Кроме того, развернут отдельный Backend-сервер, написанный на языке Python с использованием библиотеки Flask, использующийся для работы с моделью машинного обучения и обработки запросов со стороны React. Запросы к данному серверу отправляются только с одной страницы веб-приложения, связанной с прогнозированием цен на поддерживаемые автомобили.

Такая структура позволяет веб-приложению обеспечивать безопасное и прозрачное проведение аукционов, основанных на NEAR Protocol, и

обеспечивает эффективное взаимодействие между пользовательским интерфейсом, серверными компонентами и смарт-контрактом.

Веб-приложение состоит из следующих основных частей:

- React-клиент:
 - Представляет пользовательский интерфейс (UI), через который пользователи могут взаимодействовать с аукционами;
 - Отправляет запросы на Express-сервер для просмотра/открытия/закрытия/отмены аукционов, отображения ставок и пр.;
 - Отвечает за визуализацию данных и обновление пользовательского интерфейса в соответствии с ответами от Express-сервера;
- Express-сервер:
 - Представляет собой набор серверных компонентов, которые обрабатывают запросы от React-клиента и взаимодействуют с смарт-контрактом;
 - Проверяет запросы от React-клиента, выполняет необходимую валидацию данных и инициирует вызовы к смарт-контракту в блокчейне NEAR;
 - Получает ответы от смарт-контракта и передает их обратно в React-клиент для отображения пользователю;
- Flask-сервер:
 - Представляет собой набор серверных компонентов, которые обрабатывают запрос от React-клиента на прогнозирование цены поддержанного автомобиля и вызывают модель машинного обучения;
- Смарт-контракт:
 - Представляет собой программный код, который выполняется в блокчейне NEAR и управляет логикой аукциона;

- Создает, отменяет или завершает аукционы, а также сохраняет информацию о состоянии аукциона и подтверждает выполнение операций;
- Взаимодействует с блокчейном NEAR для записи и чтения данных, а также обрабатывает транзакции, связанные с аукционом.

Далее, рассмотрим пример UML-диаграммы последовательности процесса создания аукциона (рис. 3.2).

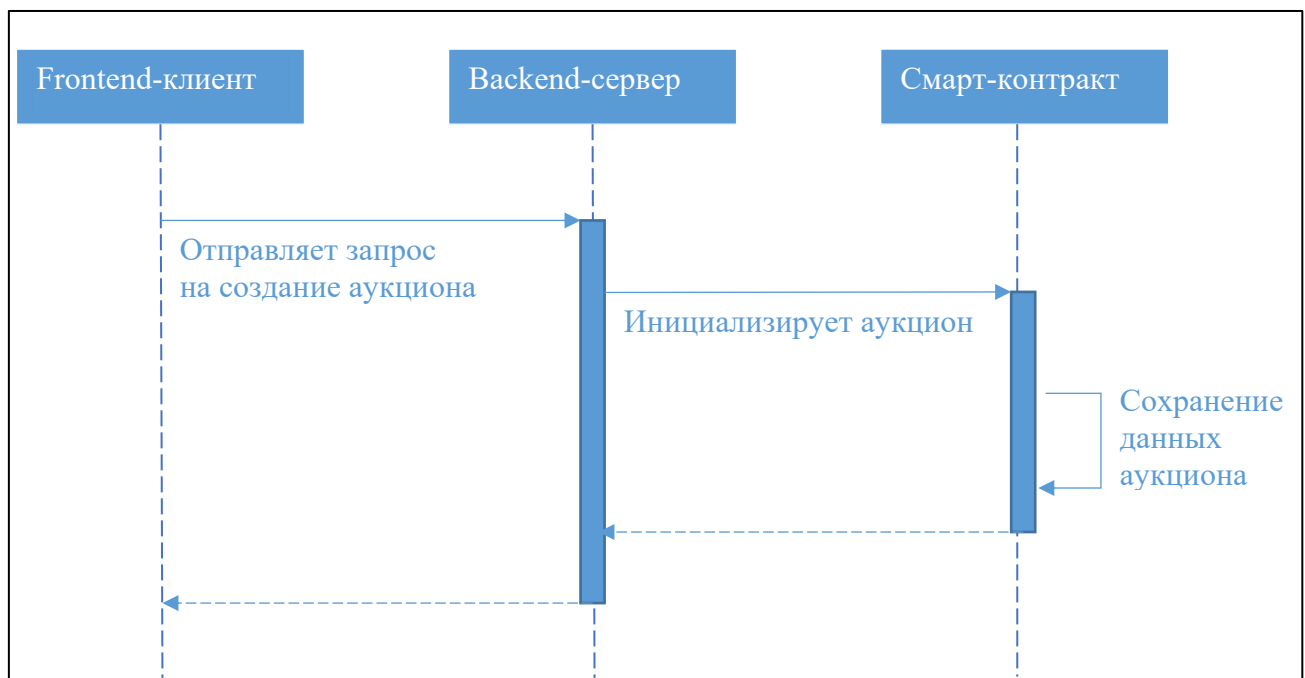


Рис. 3.2 UML-диаграмма последовательности процесса создания аукциона

3.3 Описание экосистемы NEAR Protocol

NEAR Protocol – это блокчейн-платформа, предназначенная для создания и развертывания децентрализованных приложений (DApps) и смарт-контрактов. Она обеспечивает масштабируемость, безопасность и простоту использования, делая разработку и взаимодействие с технологией блокчейн доступным для широкого круга пользователей [10]. Рассмотрим подробнее основные компоненты экосистемы протокола NEAR [1]:

- NEAR Blockchain. Протокол NEAR работает на собственном блокчейне и использует механизм консенсуса Proof-of-Stake (PoS)

под названием "Nightshade". Этот механизм обеспечивает высокую производительность и масштабируемость блокчейна NEAR, он также поддерживает выполнение смарт-контрактов и обеспечивает безопасное и надежное выполнение кода;

- NEAR Wallet. Кошелек NEAR - это криптовалютный кошелек, предназначенный для управления активами на платформе NEAR. Он предоставляет пользовательский интерфейс и инструменты для безопасного хранения, отправки, получения и управления токенами NEAR, а также для взаимодействия с децентрализованными приложениями на платформе NEAR;
- NEAR SDKs. Протокол NEAR предоставляет набор инструментов разработчика (SDKs) для создания приложений на платформе NEAR. Эти SDK включают языковые библиотеки, инструменты для разработки и тестирования смарт-контрактов, инструменты для разработки внешних приложений и т.д. NEAR SDK упрощают процесс разработки DApps и их интеграции с блокчейном NEAR;
- NEAR Explorer. NEAR Explorer – это веб-интерфейс и инструмент для просмотра данных в блокчейне NEAR. Он позволяет пользователям изучать блоки, транзакции, адреса, смарт-контракты и другую информацию, связанную с блокчейном NEAR. NEAR Explorer также предоставляет инструменты для мониторинга состояния блокчейна и отладки смарт-контрактов;
- NEAR Token. Токен NEAR – это основная криптовалюта в экосистеме протокола NEAR, используемая для оплаты транзакций, стимулирования участия в сети, вознаграждения валидаторов и других операций в блокчейне NEAR;
- NEAR Validators. NEAR Validators – это участники сети NEAR, которые подтверждают транзакции и защищают блокчейн. Валидаторы создают новые блоки, проверяют транзакции и поддерживают работоспособность сети. Валидаторы получают

вознаграждение в виде токенов NEAR, которые служат стимулирующим механизмом для участия в сети;

- NEAR Governance. Протокол NEAR имеет систему управления, в рамках которой сообщество NEAR принимает решения по развитию протокола, изменению правил консенсуса и другим аспектам экосистемы. Участники могут голосовать и вносить предложения через систему управления NEAR, чтобы внести свой вклад в будущее развитие платформы.

Глава 4. Реализация приложения для проведения блокчейн-аукционов

4.1 Интерфейс пользователя

Главная страница приложения содержит список всех открытых лотов на аукционе, с детальным описанием каждого из них, а также, на главной странице расположена «шапка», в которой находится кнопка для авторизации в приложении на основе кошелька «NEAR Wallet».

Если пользователь авторизован в приложении, то в «шапке» также отображается меню с кнопками «*Просмотр истории лотов*», «*Создать новый лот*», «*Список моих лотов*», «*Анализ цены поддержанного автомобиля*», «*Выйти из NEAR Wallet*». Кроме того, авторизованному пользователю отображается текущее состояние его кошелька «NEAR Wallet».

Страница неавторизованного пользователя (рис. 4.1).

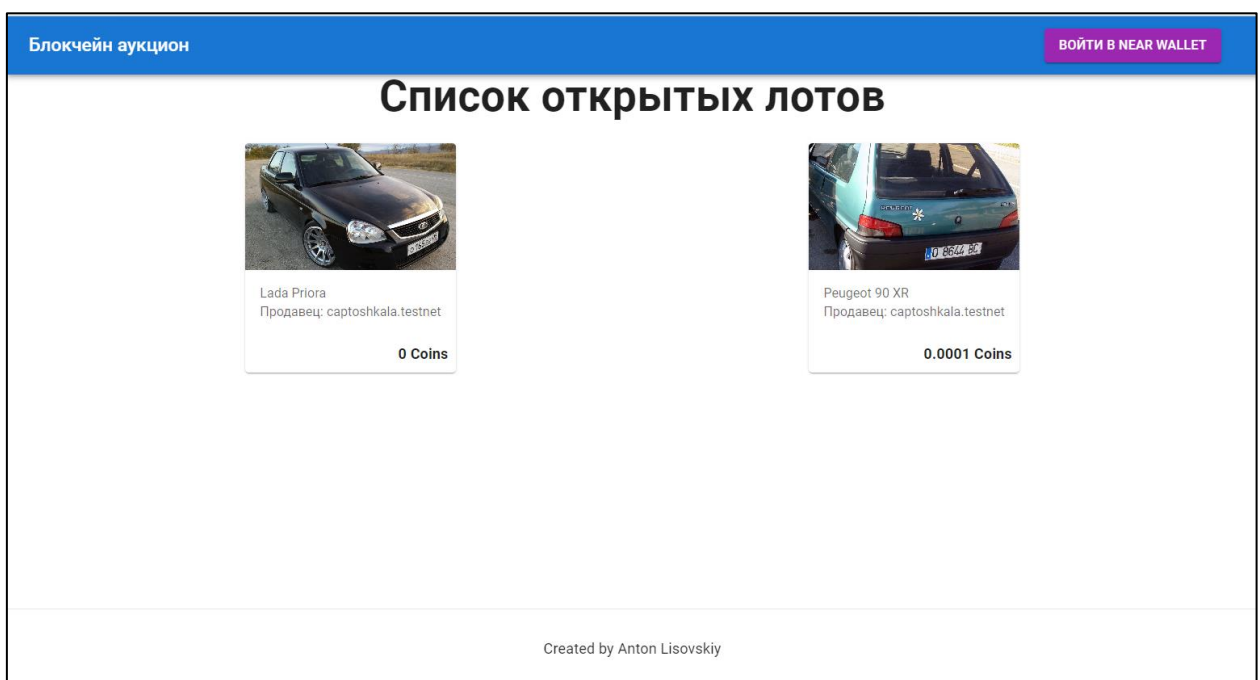


Рис. 4.1. Страница неавторизованного пользователя

Страница авторизованного пользователя (рис. 4.2).

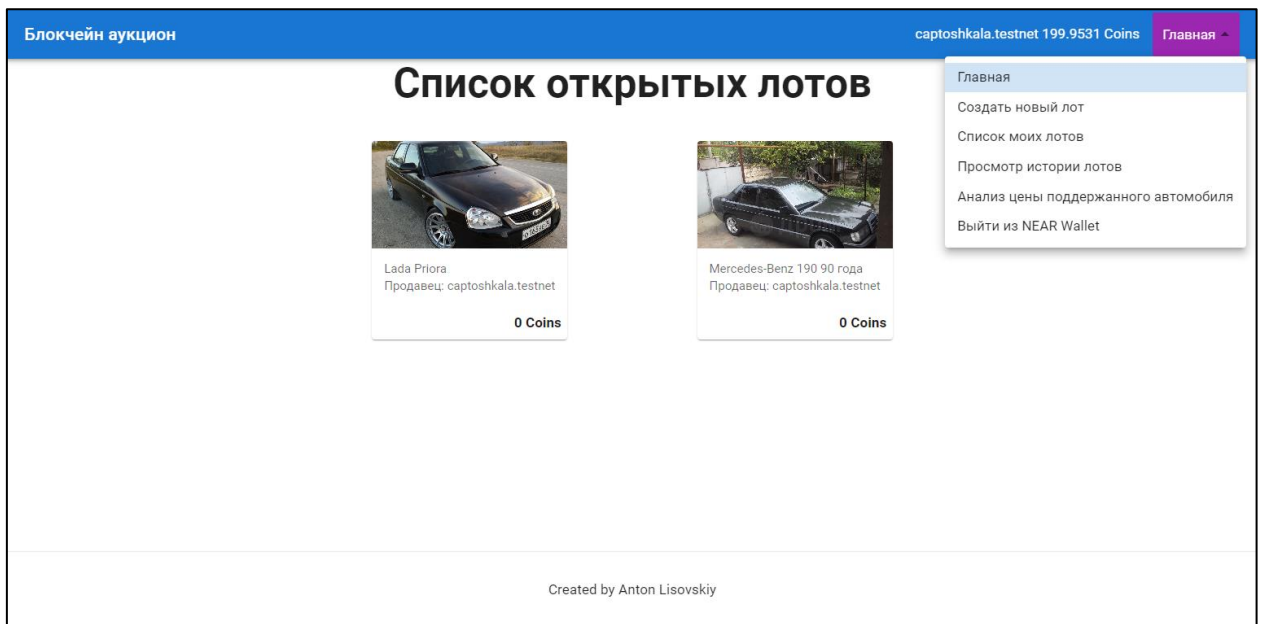


Рис. 4.2 Страница авторизованного пользователя

При нажатии на кнопку «*Войти в NEAR Wallet*» происходит переход на сайт «*testnet.mynearwallet.com*» для авторизации пользователя (рис. 4.3).

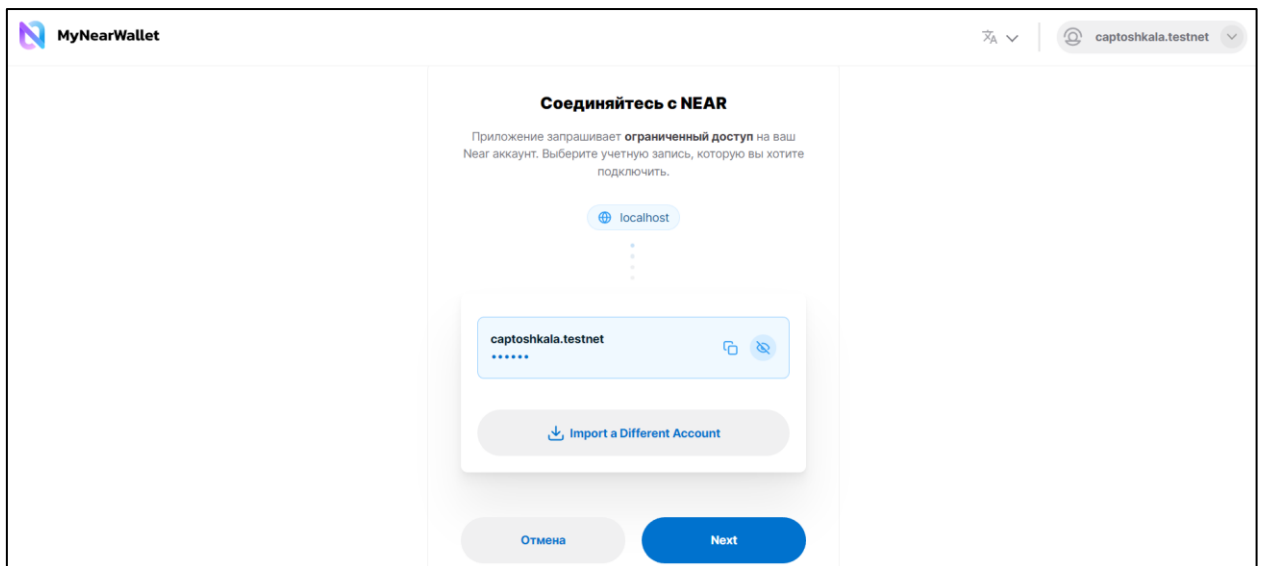
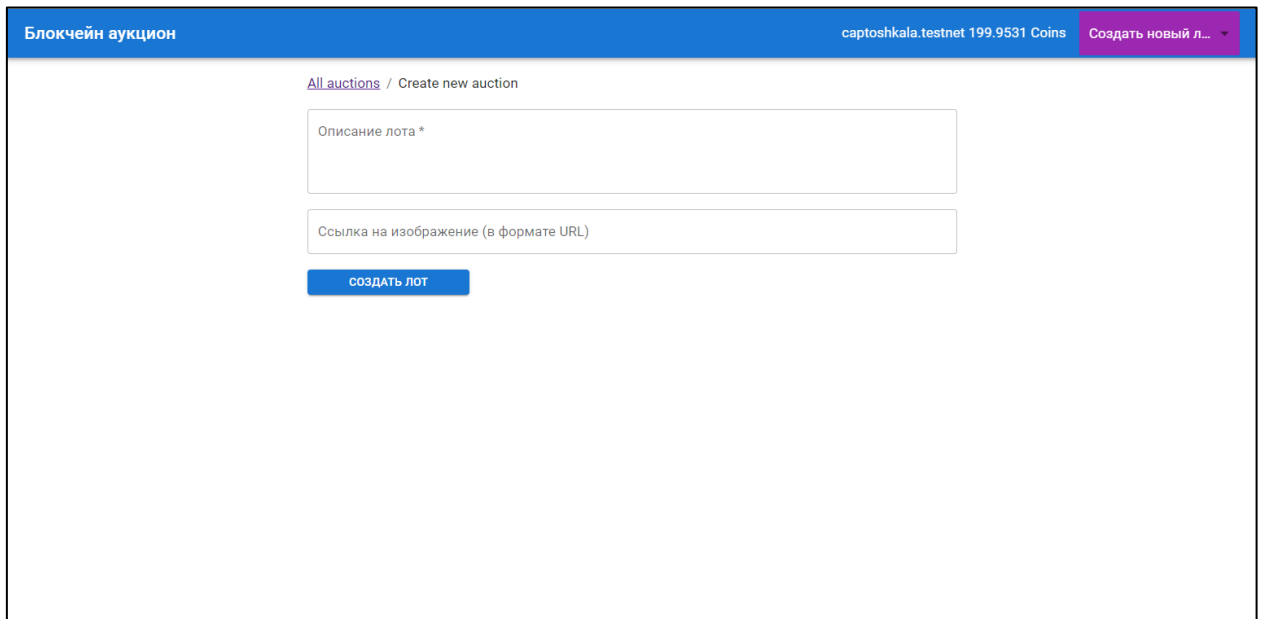


Рис. 4.3 Страница авторизации пользователя

Далее, после успешной авторизации, пользователю доступна возможность создания лотов на аукционе. Пользователь может перейти на страницу создания нового лота, нажав на кнопку «*Создать новый лот*», которая находится в «шапке» сайта.

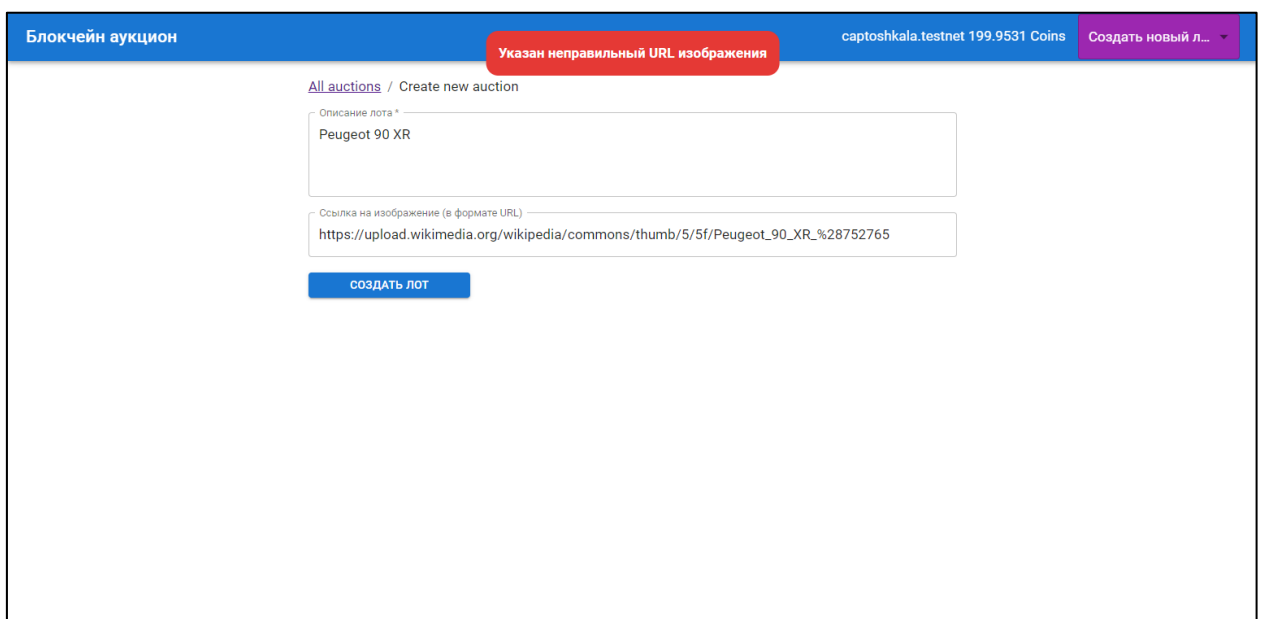
После нажатия происходит переход на страницу создания нового лота. В форме требуется ввести описание продаваемого предмета, а также прикрепить ссылку на изображение предмета в формате URL (рис. 4.4).



The screenshot shows the 'Blockchain Auction' (Блокчейн аукцион) interface. The header bar is blue and contains the site name, the user's balance 'captoshkala.testnet 199.9531 Coins', and a button to 'Create new lot' (Создать новый лот). The main content area has a breadcrumb trail 'All auctions / Create new auction'. Below this is a form with two input fields: 'Description of lot *' (Описание лота *) and 'Image URL (in URL format)' (Ссылка на изображение (в формате URL)). A blue button labeled 'CREATE LOT' (СОЗДАТЬ ЛОТ) is positioned below the second field.

Рис. 4.4 Форма для создания нового лота

Система также проверяет корректность введенного URL изображения. Если не получается загрузить изображение в систему, то на экран пользователя выводится соответствующая ошибка (рис 4.5).



This screenshot shows the same 'Create new auction' form as in Figure 4.4, but with an error state. A red banner at the top of the form area displays the message 'Invalid image URL specified' (Указан неправильный URL изображения). The 'Description of lot' field now contains the text 'Peugeot 90 XR'. The 'Image URL' field contains a long, complex URL: 'https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Peugeot_90_XR_%28752765'. The 'CREATE LOT' button remains visible at the bottom of the form.

Рис. 4.5 Пример ошибки с некорректным URL

В случае успешного создания лота, пользователю выводится соответствующее уведомление (рис 4.6).

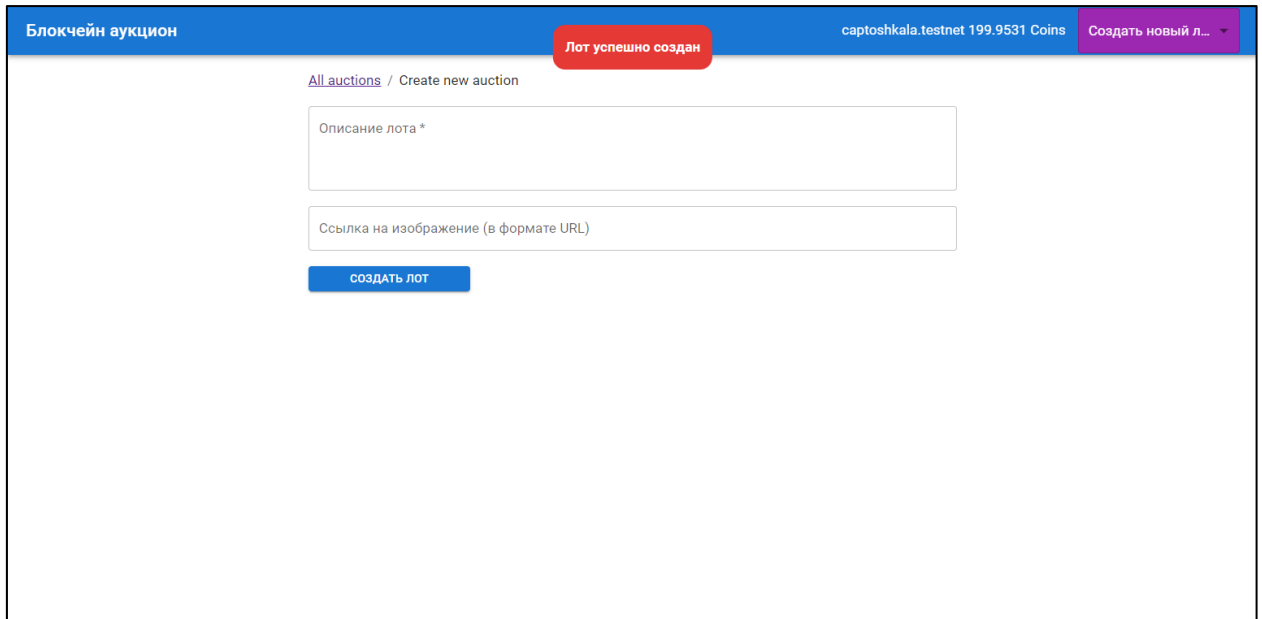


Рис. 4.6 Пример уведомления об успешном создании лота

После успешного создания, лот отображается на главной странице в списке открытых лотов аукциона (рис. 4.7).

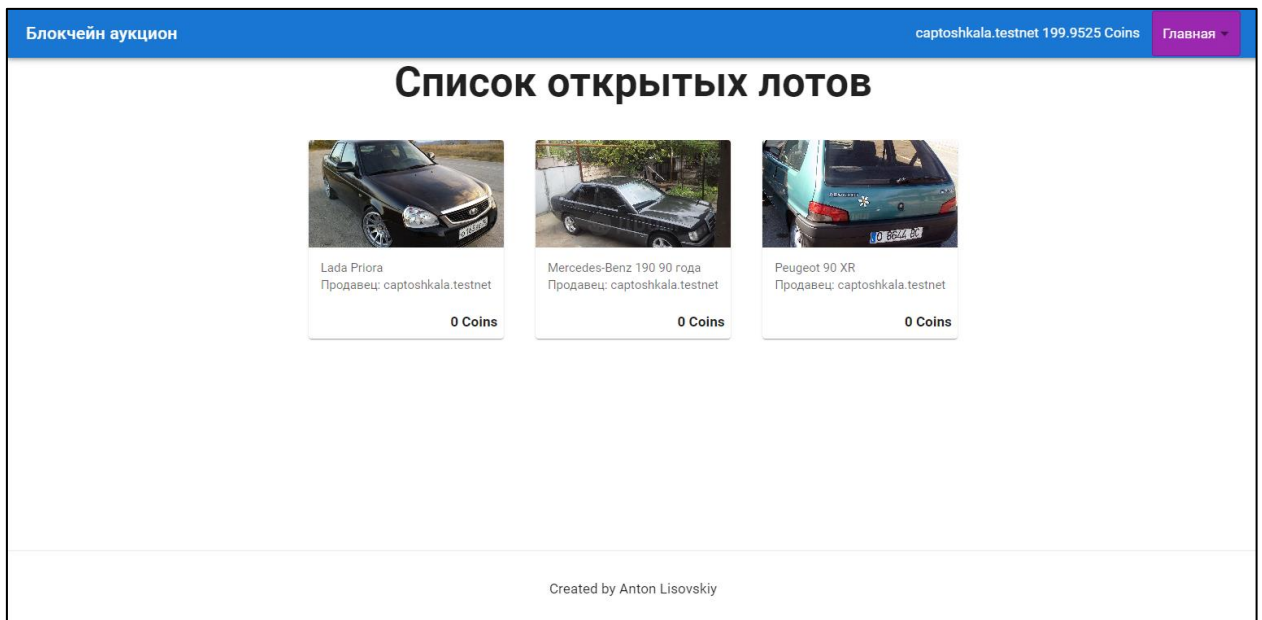


Рис. 4.7 Страница с успешно созданным лотом

Для более детального ознакомления с лотом или при желании сделать ставку, пользователь может перейти на страницу с подробной детализацией лота аукциона. Для этого нужно нажать на «карточку» лота на странице

«Список открытых лотов», либо во вкладке «Список моих лотов». На этой странице он может увидеть описание лота, продавца, текущую наивысшую ставку, а также имя пользователя, сделавшего ее, историю ставок для конкретного лота и форму размещения новой ставки (рис. 4.8).

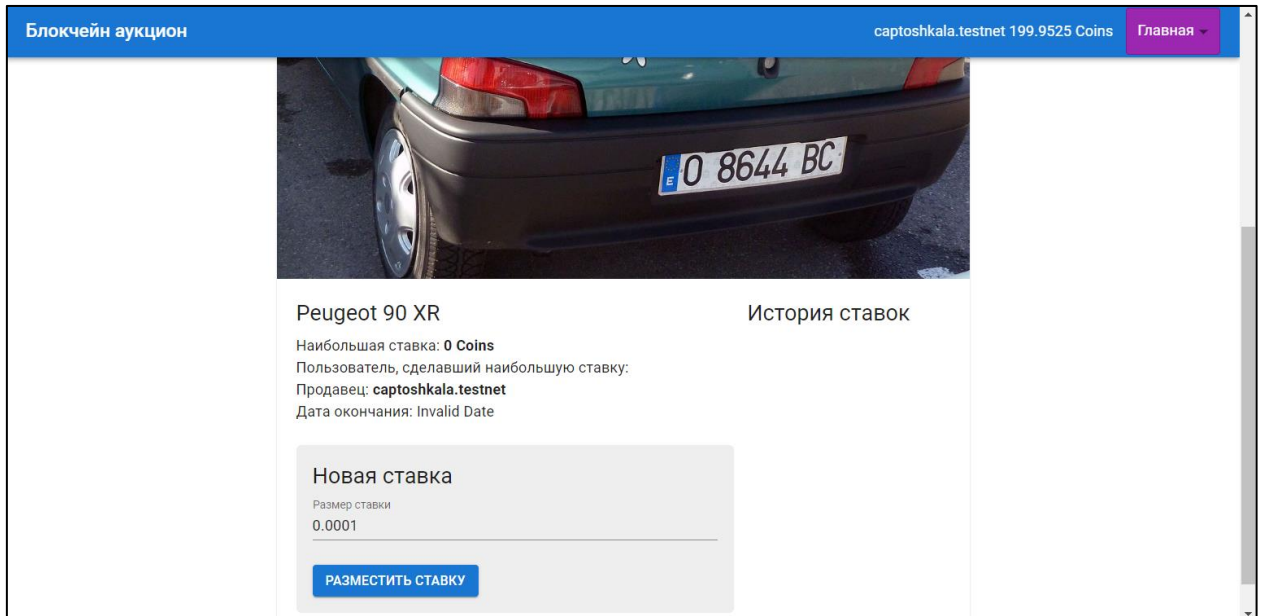


Рис. 4.8 Страница с подробной детализацией лота

Если пользователь хочет сделать ставку, он может заполнить форму для добавления новой ставки. Для этого ему необходимо ввести размер ставки, который обязательно должен превышать размер текущей ставки, и нажать на кнопку «Разместить ставку». После нажатия на кнопку, пользователь будет перенаправлен на сайт «<https://testnet.mynearwallet.com>» для подтверждения ставки (рис 4.9).

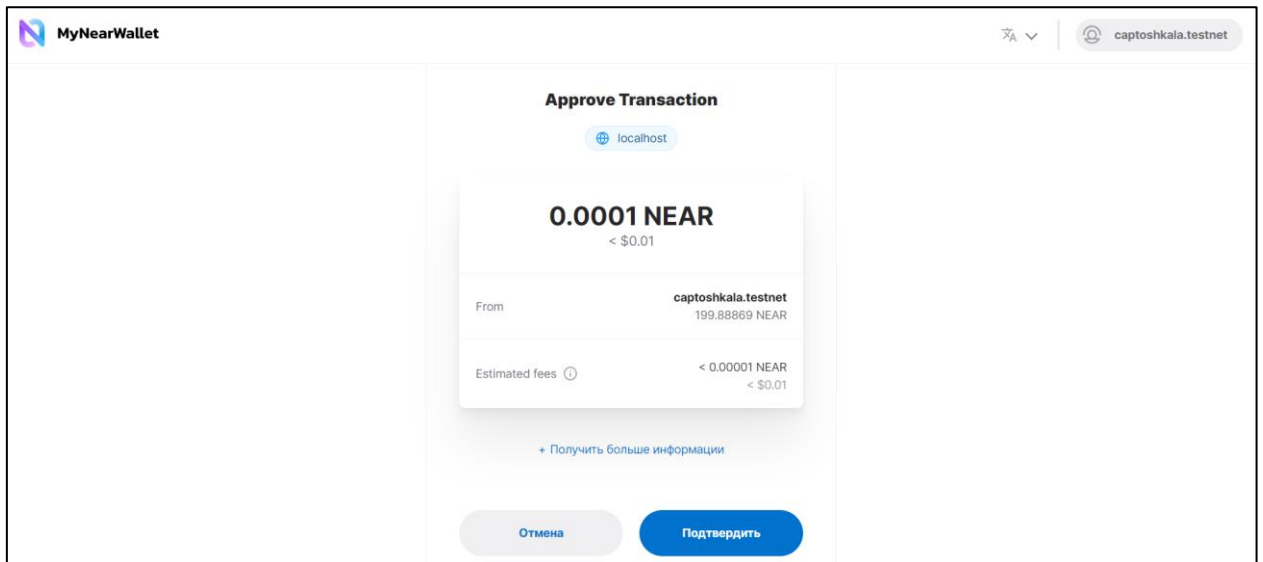


Рис. 4.9 Страница с подтверждением ставки в кошельке «NEAR Wallet»

После подтверждения ставки в кошельке «NEAR Wallet», обновляются поля «*Наибольшая ставка*», «*Пользователь, сделавший наибольшую ставку*», а также обновляется форма «*История ставок*», в которую заносится запись о размере ставки, пользователе, сделавшим ее, а также дата и время транзакции (рис. 4.10).

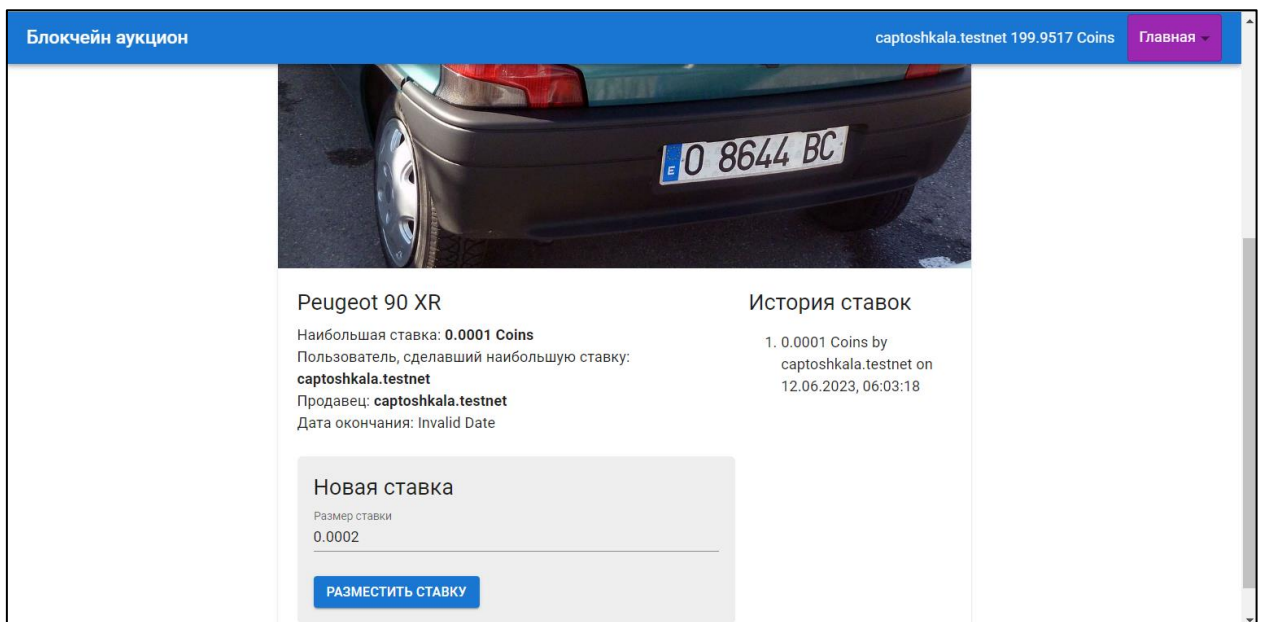


Рис. 4.10 Страница с подтвержденной ставкой

Если у пользователя есть свои лоты, то он может просмотреть их состояние, нажав на кнопку «*Список моих лотов*» в «шапке» приложения. После нажатия на кнопку, откроется страница со списком всех лотов,

созданных текущим пользователем, а также с информацией о текущем состоянии лота и двумя кнопками: «Отменить лот» и «Закончить лот». При нажатии на кнопку «Отменить лот», статус лота поменяется на «Отменен» и последняя ставка будет возвращена владельцам. Если будет нажата кнопка «Закончить лот», то последняя, наивысшая, ставка не будет возвращена владельцу. В таком случае, аукцион считается успешно завершённым, а пользователь, сделавший последнюю ставку, считается победителем (рис. 4.11).

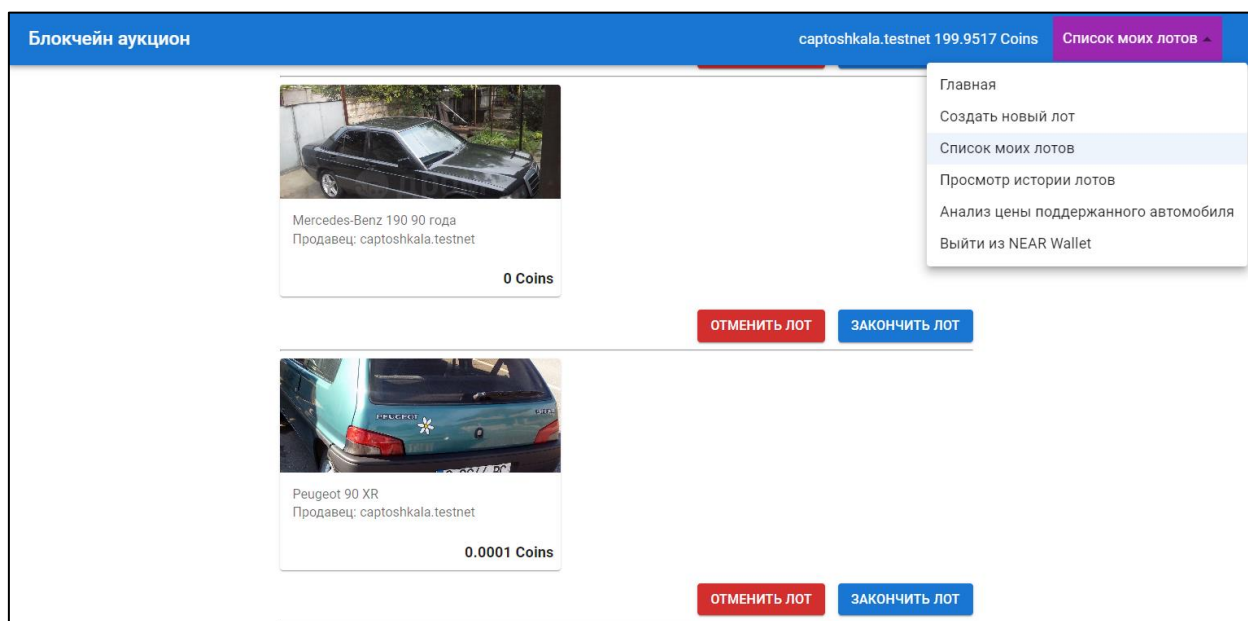


Рис. 4.11 Страница со списком лотов пользователя

Ниже приведен пример «Отмены» лота. В случае успешной «Отмены» лота, появляется соответствующее уведомление о том, что лот отменен (рис. 4.12).

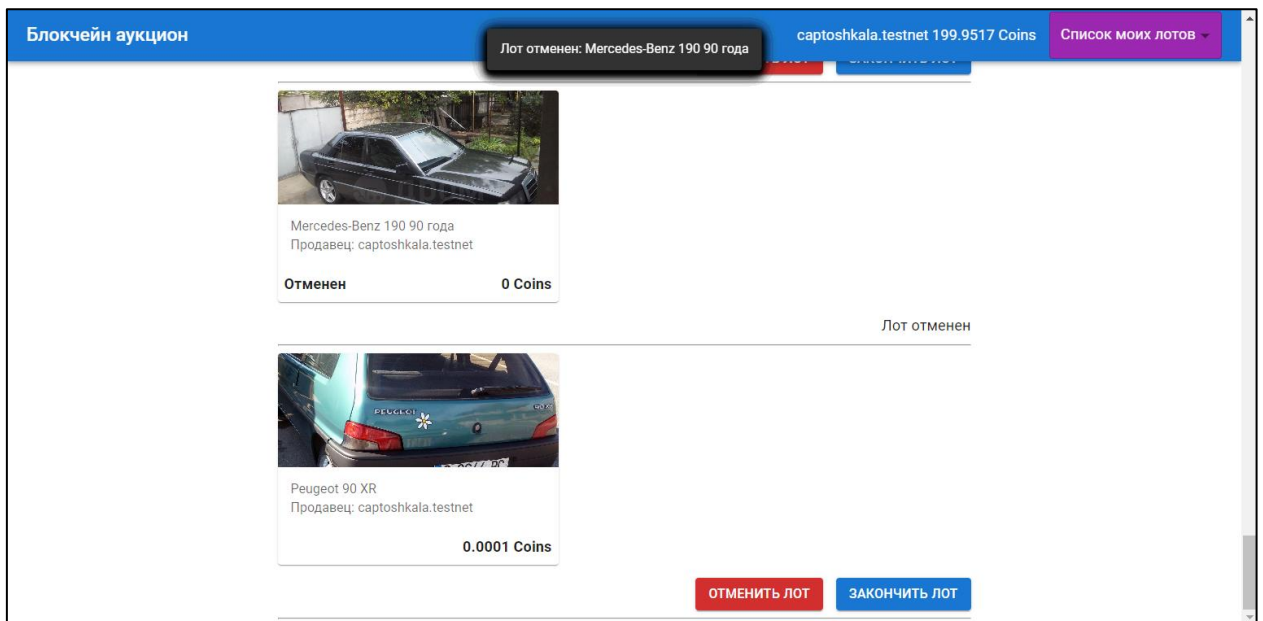


Рис. 4.12 Страница с примером «Отмены» лота аукциона

В результате «Отмены» лота закрывается возможность внесения изменений в него: не получится сделать ставку, закрыть или снова его отменить (рис. 4.13).

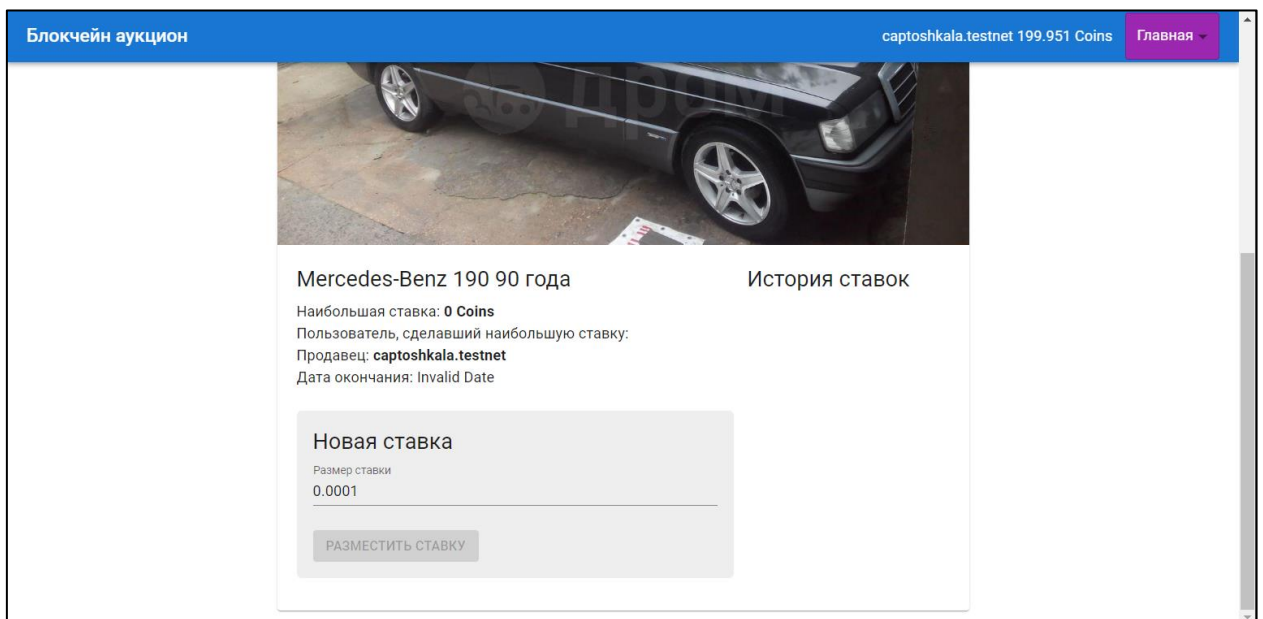


Рис. 4.13 Детализованная страница отмененного аукциона

Авторизованный пользователь, кроме просмотра своих лотов и списка открытых лотов, также может просматривать историю всех «Закрытых» или «Отмененных» лотов аукциона. Для перехода на страницу со списком таких лотов, в «шапке» приложения есть кнопка «Просмотр истории лотов». После

нажатия на кнопку, пользователь будет перенаправлен на страницу с историей всех «Закрытых» или «Отмененных» лотов аукциона (рис 4.14).

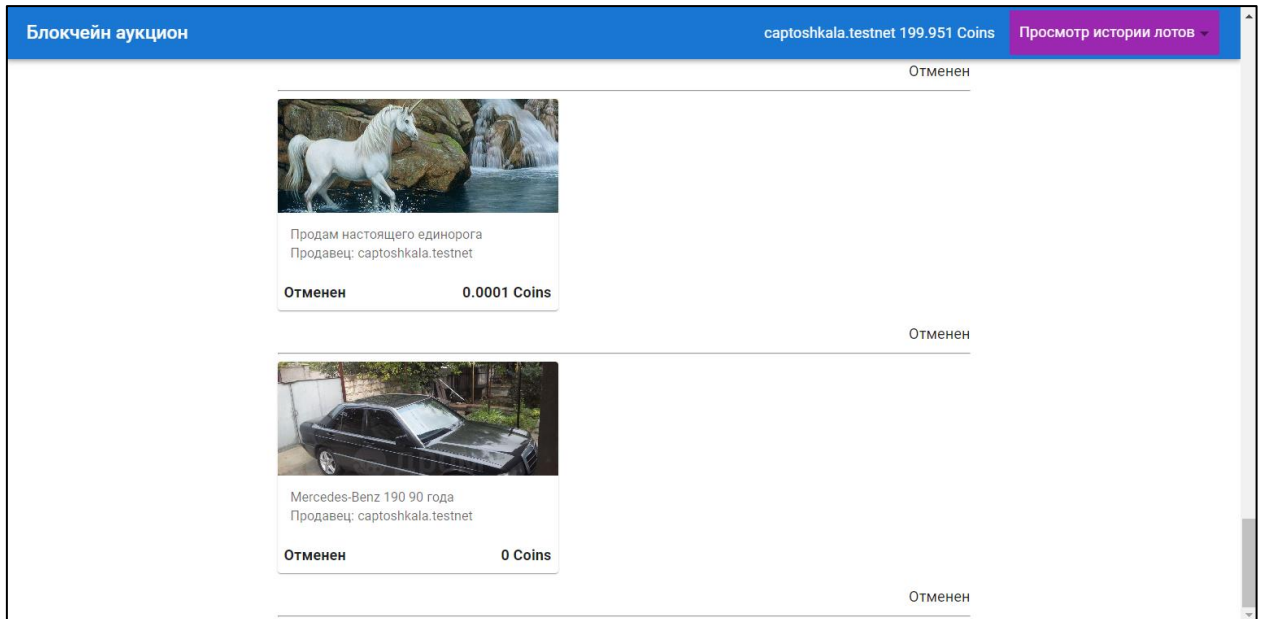


Рис. 4.14 Страница со списком всех «Закрытых» или «Отмененных» лотов

Также, авторизованному пользователю доступна возможность анализа цены поддержанного автомобиля. Для этого необходимо перейти на страницу «Анализ цены поддержанного автомобиля», заполнить форму и нажать на кнопку «Узнать цену» (рис. 15).

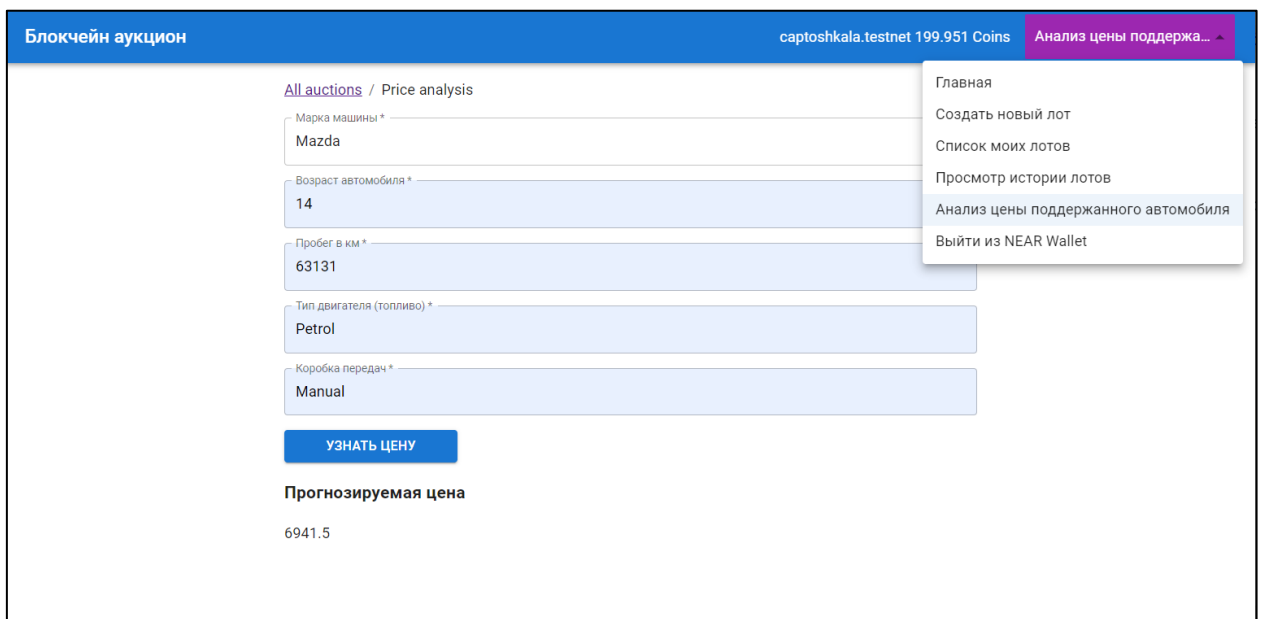


Рис. 4.15 Страница с прогнозированием цены поддержанного автомобиля

4.2 Описание серверной части приложения

Express.js является одним из самых популярных фреймворков для разработки серверной части веб-приложений. Он предоставляет большой набор инструментов и мощную функциональность для создания и развертывания серверов, обработки HTTP-запросов и управления маршрутизацией. Рассмотрим подробнее его описание:

Express.js: Это минималистичный и гибкий веб-фреймворк для Node.js. Он предоставляет простой и интуитивно понятный способ создания веб-приложений и API. Express.js обеспечивает удобную маршрутизацию, обработку HTTP-запросов, управление сессиями, обработку шаблонов и другие функции, которые делают процесс разработки веб-приложений более эффективным [11].

Особенности и преимущества Express.js [12]:

- Простота использования. Express.js предлагает простую и интуитивно понятную архитектуру и API, что делает процесс разработки веб-приложений более эффективным и быстрым;
- Гибкость и расширяемость. Express.js предоставляет гибкую архитектуру, которая позволяет разработчикам выбирать и интегрировать только необходимые компоненты и модули для своих проектов;
- Маршрутизация. Express.js обеспечивает простой и удобный способ определения маршрутов и обработчиков для различных HTTP-запросов, что упрощает организацию и обслуживание маршрутов в приложении;
- Middleware. Express.js использует концепцию Middleware, которая позволяет обрабатывать запросы и ответы до того, как они достигнут обработчиков маршрутов. Это обеспечивает гибкость в обработке запросов и позволяет добавлять функциональность, такую как аутентификация, логирование и многое другое;

- Большое сообщество и поддержка. Express.js имеет широкое сообщество разработчиков, множество документации, руководств и плагинов, что облегчает разработку и поддержку приложений.

NEAR API: Это библиотека, предоставляемая NEAR Protocol, для взаимодействия с блокчейном NEAR из JavaScript-приложений [1]. Она позволяет разработчикам создавать и отправлять транзакции, вызывать смарт-контракты, получать данные с блокчейна и многое другое. NEAR API обеспечивает простой и удобный способ интеграции с блокчейном NEAR и взаимодействия с его функциональностью.

Вместе Express.js и NEAR API образуют мощную комбинацию для разработки серверной части веб-приложений, включая взаимодействие с блокчейном NEAR. С их помощью мы можем создавать высокопроизводительные веб-серверы, обрабатывать HTTP-запросы, управлять маршрутизацией и взаимодействовать с смарт-контрактами на блокчейне NEAR. Эти фреймворки и инструменты упрощают процесс разработки и позволяют вам создавать масштабируемые и надежные веб-приложения.

В основе моего приложения лежит смарт-контракт, отвечающий за логику работы аукционов. Смарт-контракты на платформе NEAR Protocol пишутся на языке AssemblyScript, который является подмножеством языка TypeScript.

Смарт-контракт состоит из трех основных классов:

- Класс Auction представляет аукцион. Он содержит следующие поля:
 - owner: AccountId – Идентификатор аккаунта владельца аукциона;
 - canceled: bool – Флаг, указывающий, отменен ли аукцион;
 - ended: bool – Флаг, указывающий, завершен ли аукцион;
 - highestBidder: AccountId – Идентификатор аккаунта победителя аукциона;
 - highestBid: Money – Сумма самой высокой ставки на аукционе;
 - lot: Lot – Модель лота, связанного с аукционом;

- bids: Array<Bid> – Массив ставок на аукционе;
- endTime: Timestamp – Дата и время окончания аукциона;
- Класс Lot представляет лот. Он содержит следующие поля:
 - name: string – Описание лота;
 - imageUrl: string – URL изображения лота;
- Класс Bid представляет ставку на аукционе. Он содержит следующие поля:
 - sender: AccountId – Идентификатор аккаунта отправителя ставки;
 - bid: Money – Сумма ставки;
 - date: Timestamp – Дата и время размещения ставки.

Описанные выше модели данных смарт-контракта используются в классе Auctions, который реализует список контроллеров для операций с аукционами и ставками. Контроллеры взаимодействуют с методами смарт-контракта для создания, получения и обработки аукционов и ставок.

Для управления аукционами и их лотами мною был написан контроллер Auctions. Контроллер Auctions предоставляет всю необходимую функциональность для создания аукционов, размещения ставок, завершения и отмены аукционов, а также обработки переводов средств. Он использует модели смарт-контракта, таких как Auction, Bid и Lot, для хранения и обработки данных.

Рассмотрим более детально описание методов контроллера Auctions:

- create_auction(lotName:string, lotImageUrl:string)
 - Данный метод создает новый аукцион с указанным названием лота и URL-адресом изображения лота. Возвращает индекс созданного аукциона;
- get_auction(auctionId: u32): Auction | null
 - Данный метод возвращает аукцион по указанному идентификатору аукциона. Если аукцион с указанным идентификатором не существует, возвращает null;

- `list_auctions() : Array<Auction>`
 - Данный метод возвращает массив всех существующих аукционов;
- `end_auction(auctionId: u32) : void`
 - Данный метод завершает аукцион с указанным идентификатором. При завершении аукциона происходит перевод денежных средств от победителя аукциона владельцу аукциона;
- `cancel_auction(auctionId: u32) : void`
 - Данный метод отменяет аукцион с указанным идентификатором;
- `place_bid(auctionId: u32) : void`
 - Данный метод размещает ставку на аукционе с указанным идентификатором, а также проверяет, является ли размещенная ставка самой высокой ставкой, и обновляет информацию о высшей ставке и победителе аукциона;
- `withdraw() : void`
 - Контроллер, который отменяет ставку и возвращает средства ставки отправителю;
- `on_transfer_complete() : void`
 - Обработчик, вызываемый при завершении перевода средств. В данной функции подтверждается успешное выполнение перевода.

В ходе реализации приложения появилась необходимость в написании модели машинного обучения для прогнозирования цен на поддержанные автомобили. Пользователю предоставляется возможность просмотра актуальной цены на выставяемый лот автомобиля при введении дополнительных характеристик автомобиля.

В процессе написания оптимальной модели машинного обучения был проведен анализ 6 разных алгоритмов для сравнения и выбора наилучшего

алгоритма на основе различных метрик оценки качества предсказаний, таких как коэффициент детерминации (R^2), средняя абсолютная ошибка (MAE) и среднеквадратичная ошибка (RMSE). Цель состояла в том, чтобы определить алгоритм, который демонстрирует наилучшую производительность на данном наборе данных.

Ниже, представлена таблица (табл. 2) с итоговыми результатами для каждого алгоритма.

Таблица 2. Результаты анализа алгоритмов машинного обучения

	Train_Score	R^2	MAE	RMSE
LightGBM	0.868353	0.797178	0.186938	0.251137
XGBoost	0.949535	0.769511	0.197808	0.267718
Random forest	0.961061	0.746753	0.214018	0.280624
ExtraTrees	0.999987	0.717887	0.227945	0.296186
Linear regression	0.552253	0.597345	0.278478	0.353850
Ridge regression	0.549262	0.582249	0.285156	0.360422

В результате анализа были получены значения метрик для каждого алгоритма, и алгоритм LGBMRegressor показал наилучшие результаты по метрикам R^2 , MAE и RMSE. Поэтому, LGBMRegressor был выбран в качестве наилучшего алгоритма для обучения модели.

Для подбора оптимальных параметров модели LGBMRegressor использовался метод GridSearchCV, который перебирает различные комбинации параметров и выбирает оптимальные значения, опираясь на заданную метрику оценки качества (в моем случае, это RMSE).

Для интеграции описанной выше модели машинного обучения в веб-приложение, использовалась библиотека Pickle в Python, которая необходима для сериализации (преобразования объектов в байтовый поток) и десериализации (восстановления объектов из байтового потока) данных, а именно для сохранения модели и ее нормализации в файл формата pickle для последующего использования [13].

Для обработки и вызова методов модели, а также для взаимодействия с клиентской частью веб-приложения, был написан отдельный сервер на языке Python.

4.3 Описание клиентской части

Для разработки клиентской части веб-приложений существует множество мощных инструментов, которые помогают создать отзывчивый, динамичный и масштабируемый пользовательский интерфейс, а также обеспечить быстрое взаимодействие с серверной частью таких веб-приложений.

Рассмотрим ключевые инструменты, которые были использованы при разработке клиентской части своего веб-приложения: React.js, Next.js и NEAR API:

- React.js – это наиболее популярная JavaScript-библиотека, используемый для разработки пользовательского интерфейса. Она предоставляет эффективные инструменты и возможность создания компонент, которые позволяют создавать динамичные и интерактивные веб-приложения [2]. React.js использует компонентную модель, где каждый элемент интерфейса представлен отдельным компонентом, что облегчает управление и переиспользование кода. Кроме того, благодаря виртуальной модели DOM и эффективному рендерингу, React.js обеспечивает быструю загрузку и отзывчивость пользовательского интерфейса;
- Next.js – это открытый фреймворк, созданный поверх React.js, предназначенный для разработки мощных и масштабируемых веб-приложений. Он добавляет дополнительные функции поверх React.js, такие как серверный рендеринг, статическая генерация и маршрутизация. Next.js обеспечивает оптимальную производительность, улучшенную индексацию поисковыми системами и удобное управление состоянием приложения. Он также

обладает простым и интуитивно понятным API, что делает процесс разработки более эффективным и быстрым;

- Near API – это набор инструментов, предоставляемых протоколом NEAR, для взаимодействия с блокчейном NEAR из JavaScript-приложений. NEAR – это децентрализованная экосистема, которая обеспечивает безопасное и эффективное взаимодействие с смарт-контрактами. С помощью NEAR API можно создавать и отправлять транзакции, вызывать функции смарт-контрактов, получать данные с блокчейна NEAR и многое другое. Это позволяет интегрировать функциональность блокчейна NEAR в веб-приложение [14].

Для реализации клиентской части веб-приложения были разработаны следующий компоненты:

- `component Auction`
 - Описание: Данная компонента отображает информацию о конкретном лоте аукциона, включая название, продавца, текущую ставку и статус аукциона;
 - Использование: Компонента используется в компонентах `AuctionHistory` и `DeleteOrCancelAuction` для отображения списка аукционов;
- `component PlaceBid`
 - Описание: Данная компонента отображает форму для размещения новой ставки на аукционе;
 - Используется: Компонента используется в компоненте `AuctionDetails` для размещения ставок на аукцион;
- `component AuctionDetails`
 - Описание: Данная компонента отображает подробную информацию о конкретном аукционе, включая изображение лота, название лота, текущую ставку, продавца и историю ставок;

- Используется: Компонента используется в компонентах AuctionHistory и DeleteOrCancelAuction для отображения подробностей о каждом аукционе;
- `component AuctionHistory`
 - Описание: Данная компонента отображает список аукционов и фильтрует их, отображая только отмененные или завершенные аукционы, а также отображает информацию о каждом аукционе с помощью компоненты Auction;
 - Используется: Компонента используются для отображения истории аукционов на отдельной странице веб-приложения;
- `component DeleteOrCancelAuction`
 - Описание: Данная компонента отображает список аукционов, а также предоставляет возможность отменить или завершить открытый аукцион;
 - Используется: Компонента используются для отображения на отдельной странице веб-приложения списка аукционов с возможностью отмены или завершения открытых аукционов;
- `component NewAuctionForm`
 - Описание: Данная компонента отображает форму для создания нового аукциона. Она содержит поля для ввода описания лота и ссылки на изображение лота;
 - Используется: Компонента используются для создания новых аукционов и отображается на отдельной странице веб-приложения;
- `component PriceAnalysis`
 - Описание: Данная компонента отображает форму, после заполнения которой, пользователю предоставляется возможность увидеть спрогнозированную цену на поддержанный автомобиль. Она содержит поля для ввода марки

машины, возраста автомобиля, пробега, типа двигателя и коробки передач;

- Используется: Компонента используются для отображения спрогнозированной цены на поддержанный автомобиль.

Далее, рассмотрим список сервисов, которые требовалось реализовать для взаимодействия с серверной частью приложения, путём отправки HTTP-запросов:

- `getWallet`
 - Это асинхронная функция, которая создает `keyStore` с использованием приватного ключа в локальном хранилище браузера, а также устанавливает соединение с блокчейном NEAR с использованием файла конфигурации;
- `getUser`
 - Это асинхронная функция, которая возвращает информацию о текущем пользователе. Она проверяет, есть ли у текущего пользователя идентификатор аккаунта, и если есть, то возвращает информацию о балансе аккаунта;
- `init`
 - Это асинхронная функция, которая инициализирует подключение к блокчейну NEAR, а также создает кошелек и получает информацию о текущем пользователе;
- `listAuctions`
 - Это асинхронная функция, которая обращается к серверной части приложения и возвращает список аукционов;
- `getAuction`
 - Это асинхронная функция, которая обращается к серверной части приложения и возвращает информацию о конкретном аукционе;
- `placeBid`

- Это асинхронная функция, которая обращается к серверу и размещает ставку на указанном аукционе;
- `createAuction`
 - Это асинхронная функция, которая обращается к серверной части приложения для создания нового аукциона;
- `cancelAuction`
 - Это асинхронная функция, которая обращается к серверной части приложения для отмены указанного аукциона;
- `endAuction`
 - Это асинхронная функция, которая обращается к серверной части приложения для завершения указанного аукциона.

4.4 План тестирования приложения

Чтобы убедиться, что данное веб-приложение работает корректно, необходимо выполнить следующие тесты:

- проверка наличия подключения к блокчейну NEAR;
- проверка корректной работы функциональности кошелька NEAR Wallet;
- проверка корректной работы функциональности смарт-контрактов;
- проверка корректной работы форматирования и парсинга NEAR сумм;
- проверка корректности валидации каждого поля в клиентской части веб-приложения;
- проверка корректности отображения данных в клиентской части веб-приложения;
- проверка корректности работы всех кнопок в клиентской части веб-приложения.

Приложение успешно прошло все указанные тесты.

Тест 1. Проверка работоспособности функциональности смарт-контракта.

Цель. Проверка реакции приложения на размещение ставки на аукционе.

Порядок выполнения. Переход на страницу с детализацией аукциона. Ввод числового значения в форму *Новая ставка*. Нажатие на кнопку *Разместить ставку*.

Результат. Отображается сообщение о размещении новой ставки в форме *История ставок*. Тест пройден.

Тест 2. Проверка корректности валидации полей данных веб-приложения.

Цель. Проведение тестирования реакции приложения на неправильно заполненные поля данных.

Порядок выполнения. Переход на страницу создания нового лота аукциона. Ввод некорректного URL изображения в поле *Ссылка на изображение*. Нажатие на кнопку *Создать лот*.

Результат. Пользователю выводится соответствующее сообщение рядом с полем, содержащим некорректный URL изображения. Тест успешно пройден.

Тест 3. Проверка работоспособности функциональности кошелька NEAR.

Цель. Проверка получения корректной информации о текущем пользователе.

Порядок выполнения. Нажатие на кнопку *Войти в NEAR Wallet*. Переход на страницу авторизации. Ввод данных кошелька. Нажатие на кнопку *Подключить*.

Результат. Пользователь переадресуется на главную страницу веб-приложения. У пользователя появляется форма отображения текущего состояния кошелька и 3 кнопки: *Просмотр истории лотов*, *Создать новый лот*, *Список моих лотов*. Тест пройден.

Заключение

В рамках моей дипломной работы было разработано приложение, которое позволяет пользователям создавать и участвовать в аукционах. Ниже, представлен основной функционал веб-приложения:

- **Создание аукционов:** Авторизованные пользователи могут создавать новые аукционы, указывая название лота и URL изображения лота. Созданные аукционы отображаются в списке открытых аукционов;
- **Участие в аукционах:** Пользователи могут размещать ставки на открытых аукционах, указывая сумму ставки в валюте NEAR. Размещенные ставки отображаются на странице с детализацией аукциона;
- **Отмена аукциона:** Создатель аукциона может отменить его, если это необходимо. Отмененные аукционы исчезают из списка открытых аукционов;
- **Завершение аукциона:** Создатель аукциона может завершить его, если это необходимо. Завершенные аукционы исчезают из списка открытых аукционов;
- **Аутентификация и управление кошельком NEAR:** Приложение поддерживает аутентификацию пользователей через кошелек NEAR. Кошелек пользователя используется для выполнения транзакций, например, для размещения ставок;
- **Валидация данных:** Приложение осуществляет корректную валидацию данных на клиентской стороне веб-приложения. Выводится соответствующая информация об ошибках в введенных данных рядом с полями, содержащими некорректные значения;
- **Просмотр истории аукционов:** Любой авторизованный пользователь может просматривать историю любого закончившегося или отмененного аукциона на странице со списком таких аукционов;

- Прогнозирование цены поддержанного автомобиля. Любой авторизованный пользователь можно узнать актуальную цену на поддержанный автомобиль, после заполнения соответствующей формы.

Таким образом, мое веб-приложение предоставляет авторизованным пользователям возможность создавать, участвовать, отменять или заканчивать свои аукционы, а также использовать кошелек NEAR для осуществления транзакций. Описанный мною функционал позволяет пользователям проводить аукционы в режиме реального времени, делать ставки и управлять своими аукционными лотами.

Список литературы

1. Документация NEAR. – URL: <https://docs.near.org/> (дата обращения: 20.02.2023).
2. Документация React JS. – URL: <https://reactjs.org/docs/getting-started.html> (дата обращения: 10.03.2023).
3. Торговая площадка и интернет аукцион Auction. – URL: <https://auction.ru/> (дата обращения: 15.04.2023).
4. eBay: Electronics, Cars, Fashion, Collectibles & More Ebay. – URL: <https://www.ebay.com/> (дата обращения: 19.04.2023).
5. OpenSea, the largest NFT marketplace Opensea. – URL: <https://www.testograf.ru/> (дата обращения: 19.04.2023).
6. Что такое технология блокчейн. – URL: <https://aws.amazon.com/ru/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc> (дата обращения: 10.03.2023).
7. Что такое блокчейн простыми словами. – URL: <https://trends.rbc.com/trends/industry/5f05c0a79a7947aac5c7577a> (дата обращения: 10.03.2023).
8. Auction Theory. – URL: <https://web.stanford.edu/~jdlevin/Econ%20286/Auctions.pdf> (дата обращения: 15.03.2023).
9. Auction Theory. – URL: <https://uu.diva-portal.org/smash/get/diva2:938852/FULLTEXT01.pdf> (дата обращения: 15.03.2023).
10. Что такое NEAR Protocol? Полное введение. – URL: <https://vc.ru/crypto/244724-что-такое-near-protocol-polnoe-vvedenie> (дата обращения: 09.04.2023).
11. Express JS. – URL: <https://ru.hexlet.io/qna/javascript/questions/express-js-что-это> (дата обращения: 13.04.2023).

12. What is Express JS in Node JS? – URL: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js> (дата обращения: 13.04.2023).
13. Модуль pickle? – URL: <https://pythonworld.ru/moduli/modul-pickle.html> (дата обращения: 03.05.2023).

Приложение 1. Листинг серверной части приложения

TODO: листинг серверной части приложения

Приложение 2. Листинг клиентской части приложения

TODO: листинг клиентской части приложения