

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа №3

Выполнил:

Таякин Даниил

Группа:

K33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## **Цель**

На примере собственного готовящегося коммерческого проекта составить отчет по ранее реализованным элементам в web-приложении.

## **Идея проекта**

Dekast – это бесплатная платформа, которая облегчает сделки с коллекционными предметами между пользователями. Она поддерживает прямые, безопасные и анонимные продажи, а также публичные аукционы, в которых каждый может принять участие в течение определенного периода времени. Для прозрачного предоставления этой услуги Dekast использует все преимущества TON, технологии блокчейн, изначально разработанной Telegram.

## **Примечание**

Данный проект все еще находится на стадии разработки. Некоторые функции не реализованы до конца, а конечная версия релиза может отличаться от текущей.

## **Ход работы**

В предыдущих отчетах по лабораторным работам №1 и №2 была описана структура приложения, какие страницы и компоненты оно содержит и как обрабатываются API-запросы.

В данном отчете рассмотрим, работу с router'ом. Фреймворк Next.js автоматически решает проблемы с router. На рисунке 1 показана структура раздела pages, где каждая страница уже имеет автоматически ссылку. Но есть и страница с динамичным сегментом – username, которая обозначается в квадратных скобках.

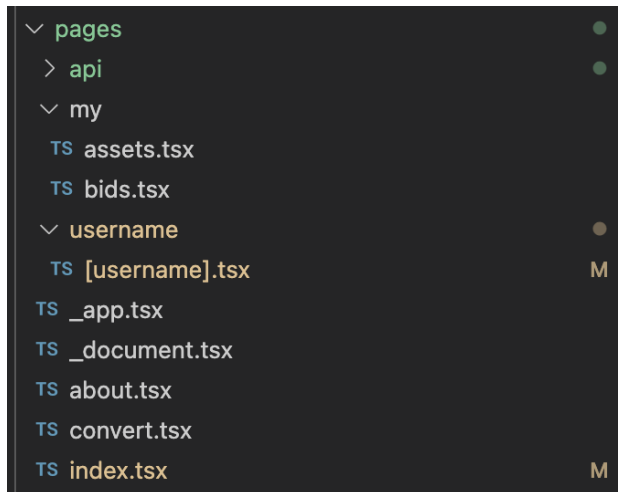


Рис. 1 – Структура pages

На рисунке 2, 3 и 4 приведен код страницы [username].tsx. На рисунке 3 есть строки, которые используют роутер и достают username из параметров url страницы.

```
import Link from 'next/link';
import { useRouter } from 'next/router';
import Layout from '@components/Layout';
import PriceView from '@components/PriceView';
import StatusBadge, { AuctionStatus } from '@components/StatusBadge';
import MyCountdown from '@components/MyCountdown';

import { getToncoinPrice } from '@utils';

import { Username } from '@services/usernames';
```

Рис. 2 – Импортированные модули в [username].tsx

```
function UsernamePage(props: {
  toncoinPrice: number
}) {
  const router = useRouter();
  const { username } = router.query;

  let auctionStatus = AuctionStatus.forSale;

  let bidInfoTitle: string = '';
  let bidInfoElement: any;
  let bidInfoTableBody: any = null;
  let mainButtons: any;

  switch (auctionStatus) {
    case AuctionStatus.taken:
      bidInfoTitle = 'Can I buy this username?';
      bidInfoElement =
        <div>
          Someone already claimed this username on Telegram. You can make an offer, which
          who may be encouraged to sell. <Link href='/about'>How does this work?</Link>
        </div>;
      mainButtons =
        <div>
          <button className='btn mainButton'>Make an offer</button>
        </div>
      break;
  }
}
```

Рис. 3 – [username].tsx

На рисунках 4, 5, 6, 7 и 8 представлены скриншоты страниц web-приложения.

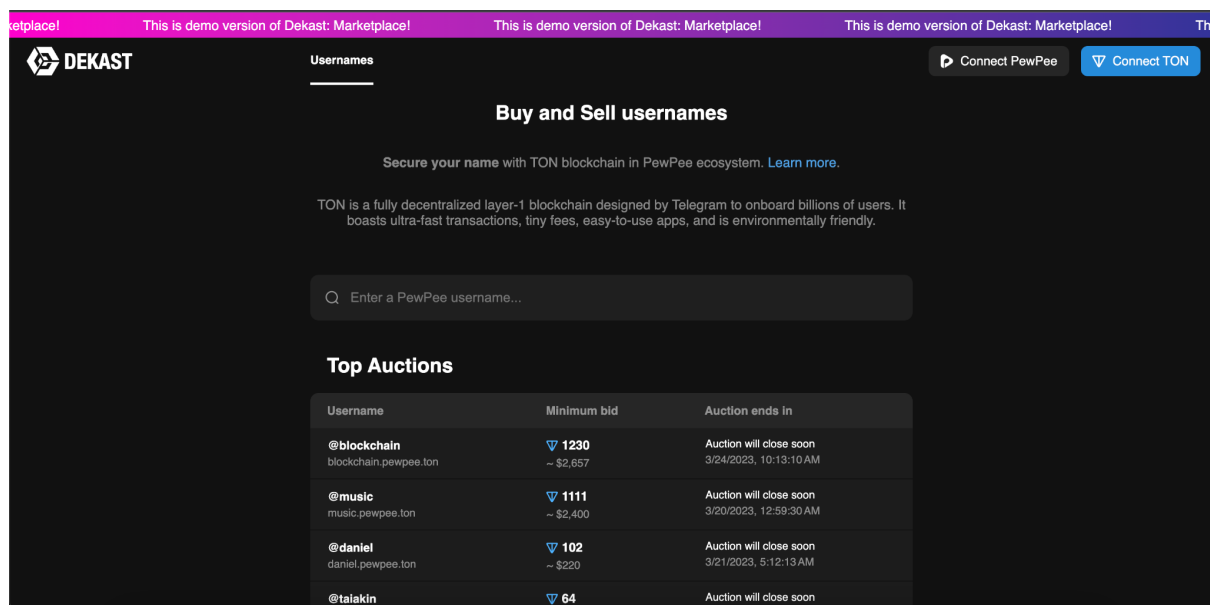


Рис. 4 – Главная страница Dekast

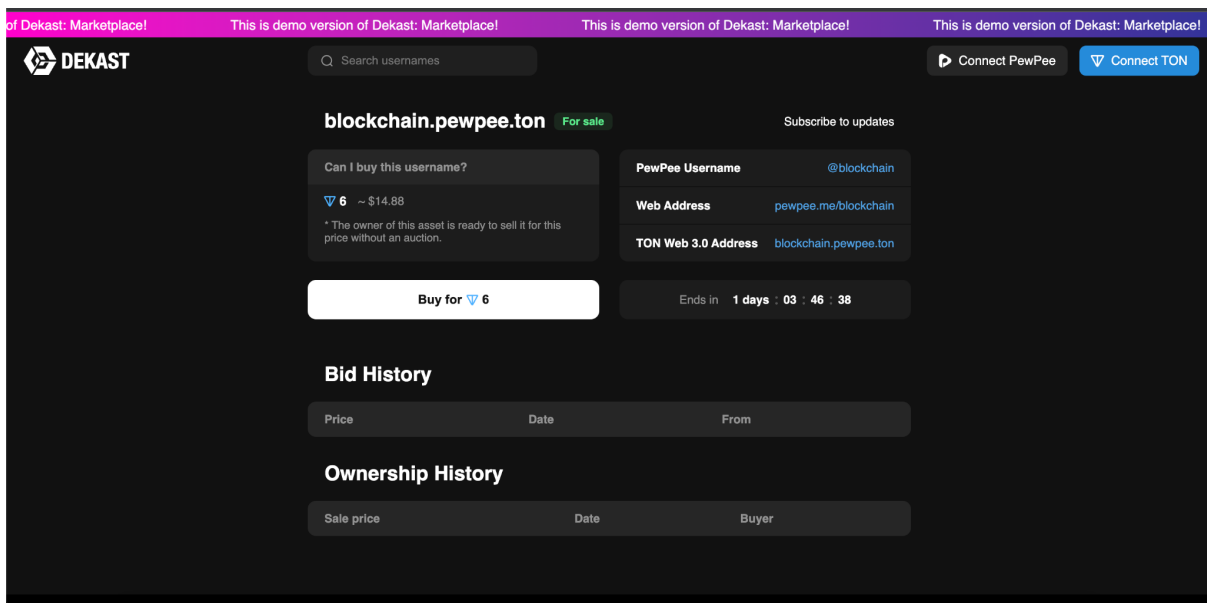


Рис. 5 – Страница аукциона

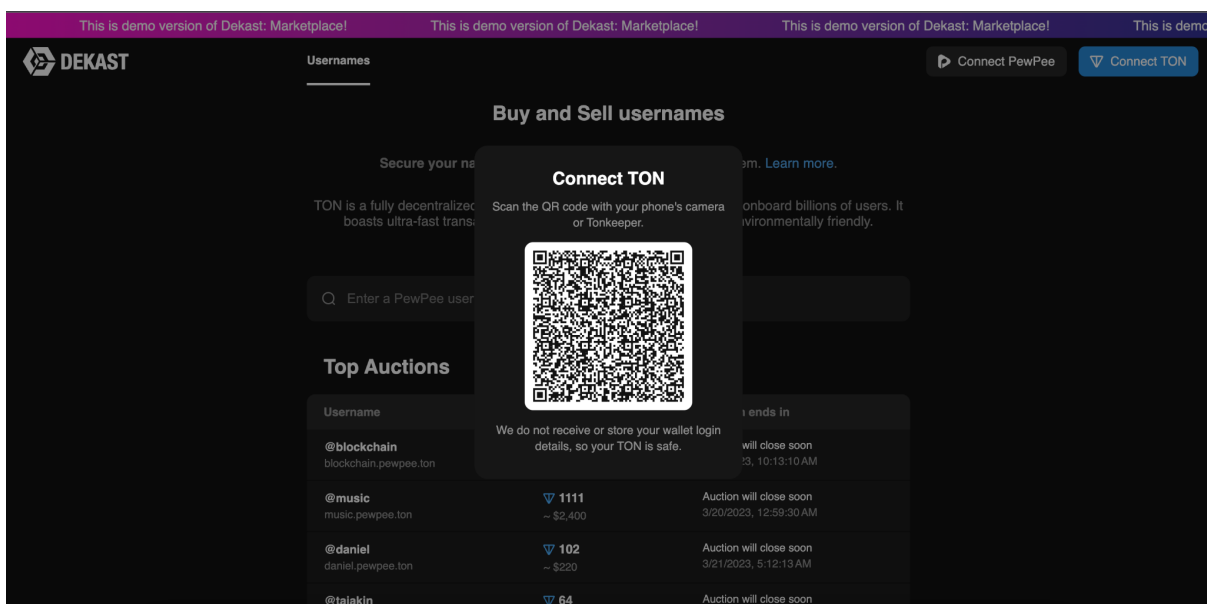


Рис. 6 – Модальное окно на подключение Web3 кошелька – Tonkeeper

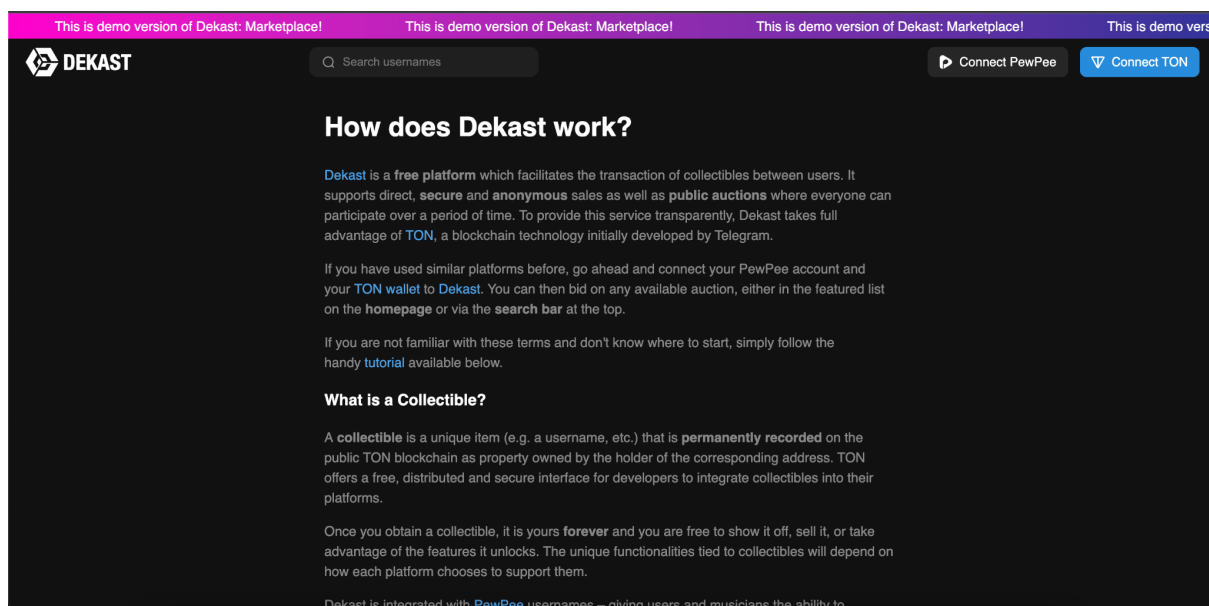


Рис. 7 – Страница с описанием проекта и как все работает на платформе

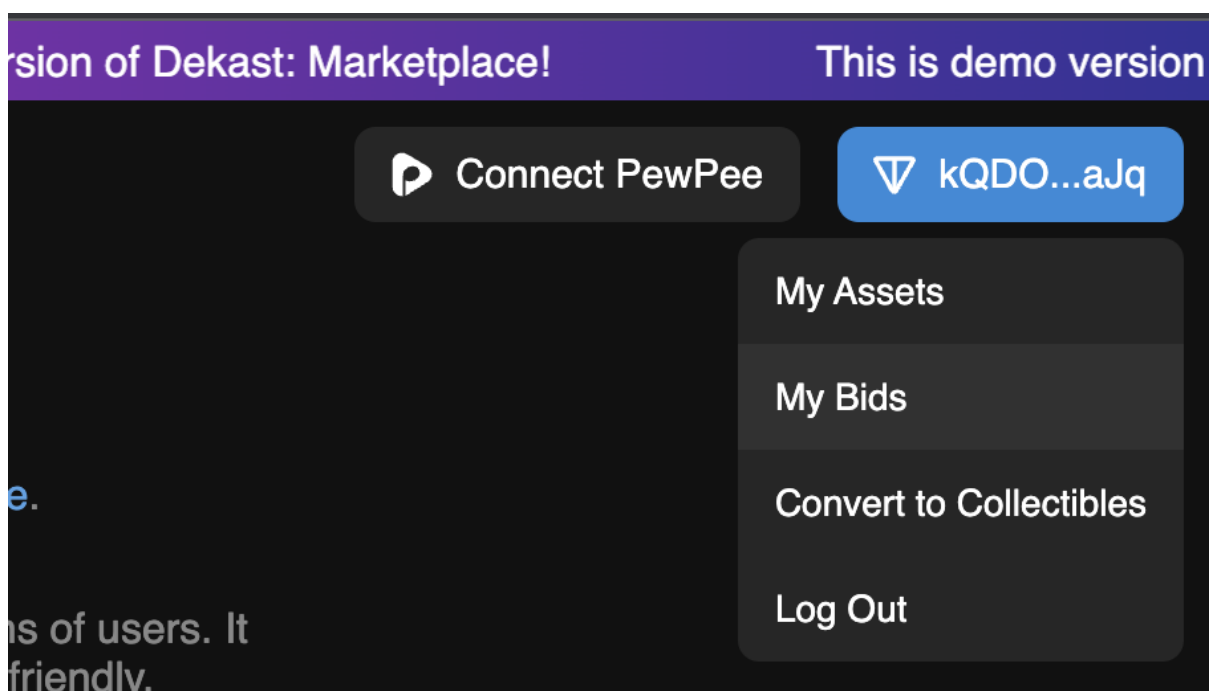


Рис. 8 – Dropdown-меню после подключения кошелька

## Вывод

Демо версию проекта можно посмотреть по адресу: <https://dekast.io>.