MarkeTree

Final Presentation

Table of Contents

MarkeTree
Team

Anthony Bonadies: CSCI/ITWS 2023

Listing/PayPal/Browse Listing

Bolai Lin: ITWS 2021/22

Dashboard / FAQ

Michael Peters: CSCI/ITWS 2021

Integration / Back end / CSS

Yanshen Lin: ITWS/CSCI 2023
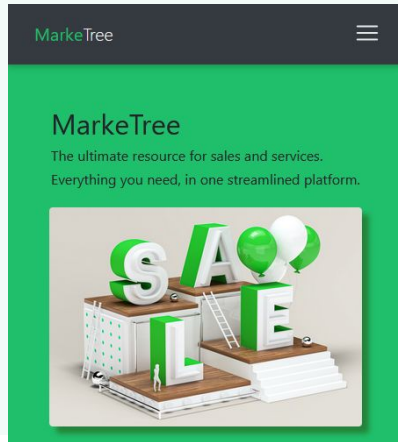
UI / Homepage / Event / Create Listing

Problem

- Moving is Tedious

    ○ Moving Companies are expensive

    ○ Storage Unit filled up quickly

- Wasted Resources

- Lack of Marketplace resource around campus

- Scam and Misinformation

MarkeTree
Does it Better

MarkeTree
The ultimate resource for sales and services.
Everything you need, in one streamlined platform.

- Locality

- Accessibility

- Legitimacy

Competitors

Offerup

OfferUp is an online mobile-first C2C marketplace with an emphasis on in-person transactions.

Craigslist

Craigslist is an American classified advertisements website with sections devoted to jobs, housing, for sale, items wanted, services, community service, gigs, résumés, and discussion forums

Facebook Marketplace

Facebook Marketplace allows Facebook users, 18 years old and up, to post pictures of their garage sale items and collectibles and browse items offered by nearby sellers as well.
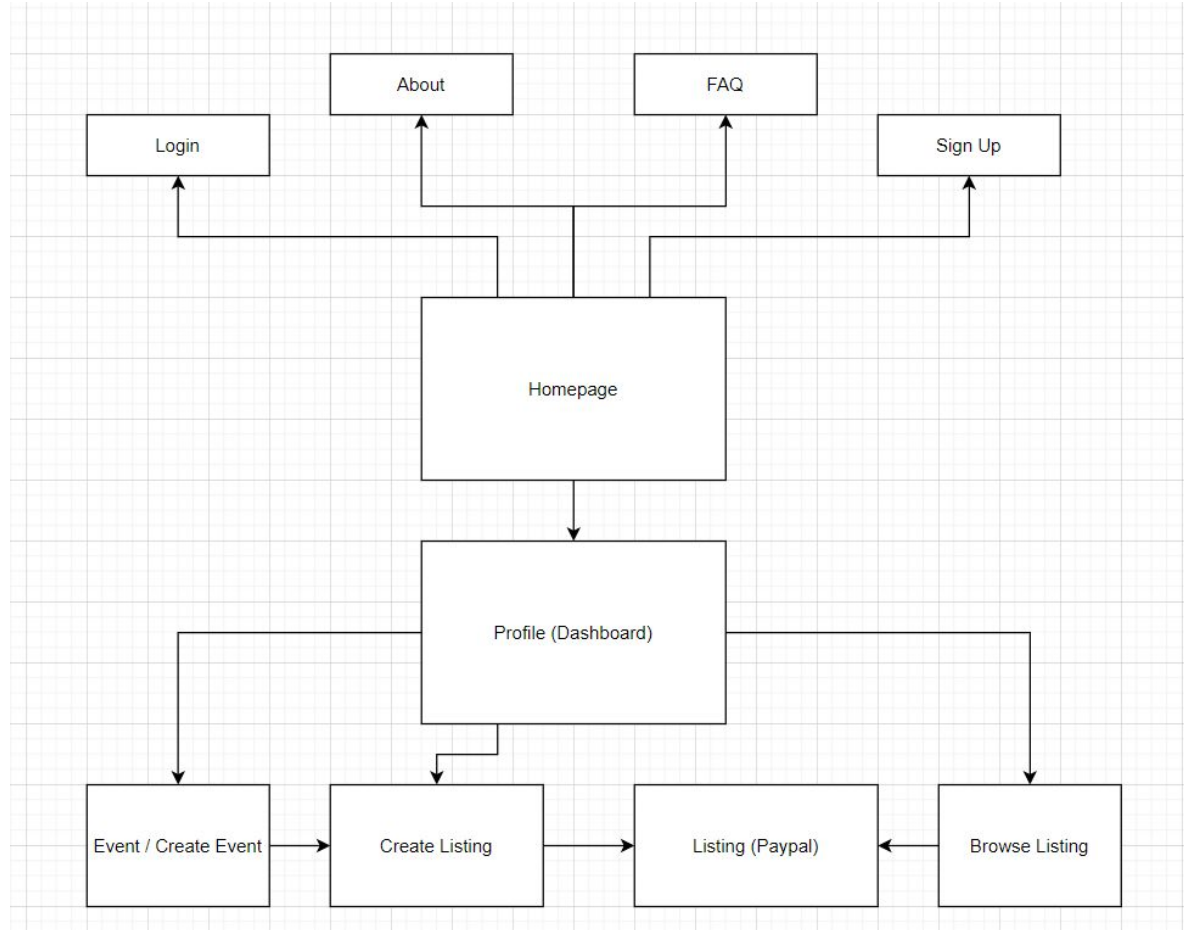
MarkeTree

Key Features/Functional requirements

MARKETPLACE

- User Rating system

- Listings

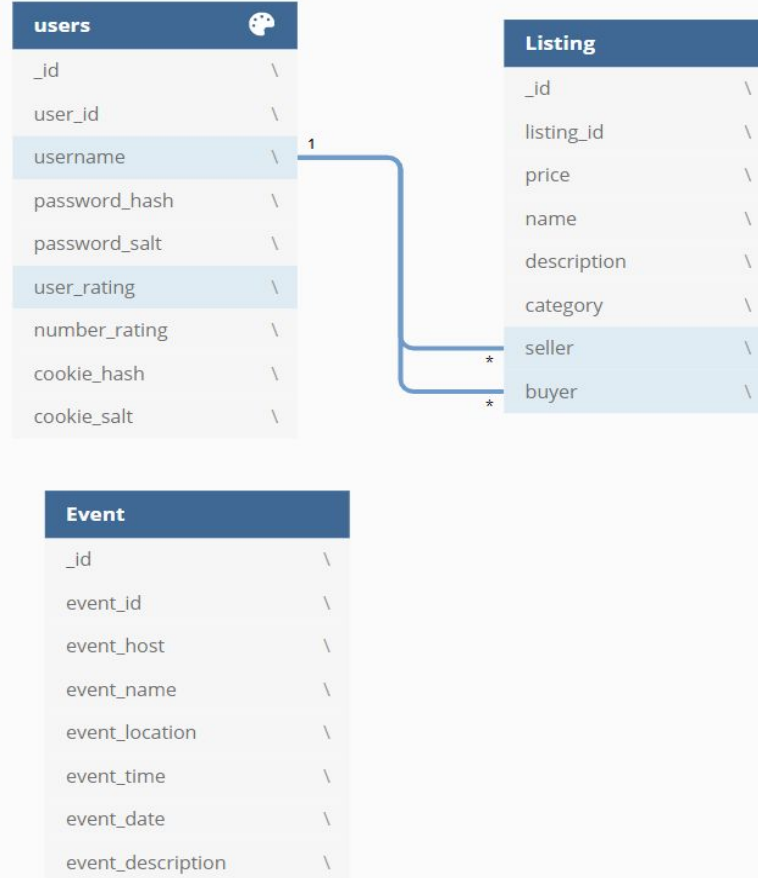- Categories

- Events

- Public Facing API for other developers

Site Map

Non-Functional Requirements

Database Structure

users 🎨
_id	\
user_id	\
username	\
password_hash	\
password_salt	\
user_rating	\
number_rating	\
cookie_hash	\
cookie_salt	\

Listing
_id	\
listing_id	\
price	\
name	\
description	\
category	\
seller	\
buyer	\

Event
_id	\
event_id	\
event_host	\
event_name	\
event_location	\
event_time	\
event_date	\
event_description	\

API Endpoints

```
app.get('/api/listings', async (req, res) => {
  let listings = null;
  await ensureConnection();
  try {
    let db = client.db('MarkeTree');
```

Made easy with express.js

Logins
POST /api/create-user
POST /api/authenticate-user
POST /api/verify-user
POST /api/rate-user

Listings
GET /api/listings
GET /api/filter-listing/:cat
GET /api/get-listing/:id

Events
POST /api/post-event
GET /api/get-event

Login Structure

user_id: "12998905-2943-42cd-bcb8-798295f923a0"
username: "MichaelP"
password_hash: "cdb3c88cd87e491273f0060c0f31a879fe91d96e161e738f7ad22ac3c6a7224f"
password_salt: "c399ea2f4903630e917a3fe5c10afe3c356d666354eca4ad8c882e48645a4419adb9e7..."
user_rating: null
number_ratings: 0
cookie_hash: "ea424451f7fdd892fdd058e06d4af554faca21fcabf7c387f259f5a05cbc0d36"
cookie_salt: "8c343c1d7f677a8436cab54ef47552eb9d12dd480373455e62b3d249634ad79dcb832f..."

Hashed and salted

Create User
POST /api/create-user
Adds a user to the database

Authenticate User
POST /api/authenticate-user
With a username/password
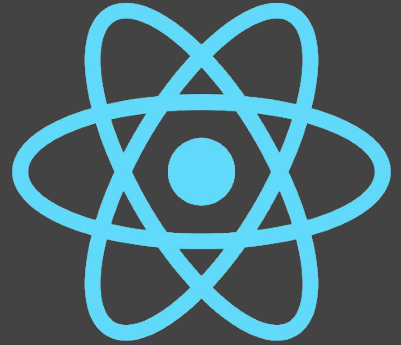
Verify (Re-authenticate) User
POST /api/verify-user
With an authentication cookie

Note: Authentication cookie is also salted in the database

Demonstration

Challenges

- React
- PayPal API
- Login system

For the Future

- User communication system (email api and client)
- Add location services
- Fleshed out profile system
- RPI Student verification

Thanks!

Does anyone have any questions?