Fantasy MVP App

Shaun Vaidyan
@shaunvaidyan on GitHub
https://shaun.vaidyan.me

Description

Fantasy Football is one of the most active and fastest growing subculture in the intersection of sports fandom, analytics, and gaming. Sleeper.app is on the fastest growing startups in this space and a community favorite but their API is missing a lot of relevant statistics about their players. FantasyMVP bridges the missing statistics with the information available to Sleeper.app users consolidated into streamlined views

Features

- Registration & User Login
- Fetches metadata from Sleeper.app API, stores it in arrays and objects, and displays the results in organized tables
- The data is paginated and searchable. You can select players to view in-depth statistics about them and compare players side by side

Planning - User Stories

I fetched the rosters and metadata about owners from the Sleeper.app API

I implemented a feature to sort the players by Name, Position, and Points

I implemented a feature to hone in on specific players and view more detailed stats about them

I programmatically fetched the players' headshots

Planning - Database

The current iteration of my app has an accounts table for login functionality. The columns are ‘id’, ‘username’, ‘password’, and ‘email’. The password field is hashed and salted using the NodeJS library bcrypt so that users creating accounts don’t reveal their passwords in plaintext.

The future iteration of my app will have a tables for Sleeper User metadata, Sleeper League metadata, and player statistics to store.

Technology Stack

- Javascript & JQuery
- Express(JS)
- NodeJS
  - Bcrypt
- MySQL

Demo

I deployed a demo of this app on my personal infrastructure @ https://fantasymvp.vaidyan.me

- DNS through Cloudflare
- Docker Compose Deployment
  - NGINX as Reverse Proxy
  - MySQL
  - NodeJS + Express + VanillaJS

What I Learned

- I learned how to implement a login frontend that can register and login users by using Express to write API routes and using a MySQL database for storage. I hashed the passwords for security using a NodeJS library: Bcrypt
- I fetched data from a company's API (Sleeper.app) about users on its platform and their metadata
  - What leagues are they in?
  - What players do they have and how many points have they scored?
- I stored that information in arrays & objects and then displayed it in visual tables for the user to see
- I successfully implemented pagination and search features to help users view large rosters
- I learned how to dockerize my application to make it platform agnostic and deploy it to infrastructure viewable to the public

What's Next

- Making the UI more responsive and mobile friendly
- Refactoring DOM manipulation code to utilize Express Templating Engine for more scalable development
- Taking the league information fetched via API and storing the JSON object into the MySQL database for persistence and more sophisticated calculations for further insights
- Implementing historical data for longitudinal insights about NFL players
- Orchestrating application containers in a high availability cluster