

**Capstone Project: Converting Iterative Design Mockup to Functional Application**  
**Group Members: Damian Wasilewicz, Geireann Lindfield Roberts, and Ahmed Amin**  
**Faculty Sponsor: Jeff Huang**

### **Overview**

Our Capstone is a cumulation of the skills we've learned throughout our Computer Science degrees, and involves a good deal of our own research into skills outside of class as well. Our original assignment involved finding an early stage startup and iteratively developing a High Fidelity prototype for the problem that the startup was tackling. Our capstone project involved converting this prototype to a functional mobile application. The application allows users to purchase (with fake currency) and manage tokens that represent athletes (at the moment, Basketball and Soccer athletes) with values that vary depending on the real-life athletes' performance in their respective fields.

Our project's work is divided into two different repositories, trendex, and trendex-server. We mainly made use of react native and typescript for our application, and other tools included a mongoDB cluster for our backend, an express router, and a python api package we developed for token valuation.

### **Running Our Project**

To run our project, you can clone the trendex and trendex-server repos, and run npm install on both to install our dependencies. From there, you can run npm start to launch both of the projects. In the terminal running the trendex app, you will be prompted to enter options (web app mode, etc). You will also be given a QR code. If you have the Expo app downloaded you can scan the code to run the app on expo; otherwise you can press w to run the app as a web app.

### **Trendex-server**

Our trendex server makes use of express to create several endpoints that our frontend trendex app interacts with. The different routes primarily interact with our mongoDB cluster. They allow us to create users, fetch users with particular usernames and passwords, and update different fields for users including the tokens that they own and the quantities in which they own them, which tokens they have in their watchlist, and their remaining balance.

### **Trendex**

#### **Login/Authentication**

Our app requires users to be signed in to a valid account to use the app. Users can toggle between a sign up screen and a log in screen. We made use of the crypto-js library to encrypt user passwords when storing them and checking for valid passwords - this prevents storing the raw passwords in our database, which would pose a liability for our users in case our database ever got hacked. We start new accounts with a balance of 100 "dollars" to give them some purchasing power.

### **Investments**

The Investments page is where a user can see their current tokens and the total value of their investments, as well as the total history of the total investment value. They can also view and manage the tokens that they have purchased and have on their watchlist. The graph has different ranges that can give users context on how their investments have fared over time.

### **Athlete Page**

The Athlete page pops up when the user clicks on a token. From here, they can buy or sell more tokens, or add/remove them from their watchlist. They can also see relevant news articles for that individual.

### **News Article Fetching**

Using the News API (<https://newsapi.org/>) we were able to get the latest news articles, but we were also able to submit specific queries to get results for different countries and relevant to the particular athlete page that the user is currently on.

### **Discover**

Our discover page allows users to find new tokens to purchase, including the use of a search engine. The page shows news articles and trending athletes that the user might be interested in.

### **Profile**

The profile page is the hub from which users can view their current balance/purchasing power, their information, and sign out, ending their active session.

### **Token Value Calculation \*\***

For both basketball and football (or soccer) players, we used a combination of current statistical averages and social media presence to place a real-world value on a given athlete's token. For the basketball players, the statistics that carried weight were points, assists, rebounds, blocks, and steals. Meanwhile, for soccer, depending on the position of the player, clean sheets, goals conceded, goals, and assists were the statistics of value. As for the social media presence, we used each athlete's instagram follower count and scaled it down to have comparable weight to the statistical score. From there, we were able to combine two scores, one representing statistical performance and the other representing social media presence, to produce a final token value.

The scaling of each athlete's social media follower count was done by initially taking the cube root of the total number. From there, we can divide this number by whatever the cube root of the athlete with the most instagram followers in order to get a number between 0 and 1 that we can then weight similarly to the statistics score. This way, instead of simply dividing each follower count by the largest follower count to get a number between 0 and 1, we get social media weights that are closer together than they otherwise would be. For example, someone like LeBron James has over 140 million followers and a lesser known athlete may have less than a million. Now, if we were to simply divide everyone's follower count by 140 million, the lesser known athletes would get very little token value from their social media presence simply because a few athletes have such a vast presence themselves.

These calculations can be found in the TrendexTokenAPI directory.

\*\* Ahmed is an Applied Mathematics-Computer Science concentrator, so in order to incorporate Applied Math into the project, he was given more responsibility in the token calculation process.

Video Demo:

[https://drive.google.com/file/d/16NOru4RwhulHp35\\_ojPb9Z066BW6DMx9/view?usp=share\\_link](https://drive.google.com/file/d/16NOru4RwhulHp35_ojPb9Z066BW6DMx9/view?usp=share_link)

Repository Links:

Trendex: <https://github.com/geireann/trendex.git>

Trendex-Server: <https://github.com/geireann/trendex-server.git>