**Weekly Report**

**Week: 03/08 - 03/14**

**Team Leader:** Aisha

**Team Member #1:** Brandon

**Total Hours Worked:** 9

**Task Completed:**

**• Backend Development**

* Set up new API routes for querying the News API based on keywords and categories.
* Implemented flexible filtering options to allow users to fetch news articles relevant to their selected topics.
* Optimized the news retrieval logic to improve response time and reduce unnecessary API calls.

**• Database Management**

* Implemented Widget\_Interest and Interest Tables following Spring Boot conventions
* Interest Table: Stores different interest categories (e.g., Tech, Finance, Sports, Weather, Stocks).
* Widget\_Interest Table: Establishes a many-to-many relationship between widgets and interests.
* Spring Boot JPA Entities, Repositories, and Services were implemented to manage database interactions.

**• API Integration**

* NewsAPI:
  + Designed API endpoints to fetch news content based on keywords or categories.
  + Integrated news results into the widget system for real-time updates.
  + Designed API endpoints to provide relevant news content to widgets.
* WeatherAPI:
  + Established a connection to the Weather API to fetch current weather data.
  + Implemented endpoints to provide temperature, wind speed, humidity, and weather conditions.
  + Designed a flexible query system where users can get weather updates based on location.
* AlphaVantageApi:
  + Integrated support for Time Series data on stocks, crypto, forex, and commodities.
  + Implemented API endpoints to fetch historical and real-time market data for use in widgets.
* **Security:**
  + Migrated all API keys to be securely loaded from environment variables (.env files).

**Team Member #2:** Vijay

**Total Hours Worked:** 12

**Tasks completed:**

* I worked on producing news articles when given recommendations for certain topics, such through the use of news API.
* Worked on making it output data which would be a summary of the stock provided (short description of stock and company in general)

**Tasks for next week:**

* Work on improving the performance of stock recommendations so more than 50 companies could be added .
* Using this same implementation into other sections of the project (sports, music).
* Start with implementation

**Team Member #3:** Aisha

**Total Hours Worked:** 10

**Task Completed:**

* + Started implementing user authentication with JWT
  + **User Model (User.java)**
    - Created a database entity to store user details.
    - Added encrypted password storage.
    - Ensured unique usernames & emails.
* **User Repository (UserRepository.java)**
  + Handles user lookups by email or username.
  + Checks if a user already exists before registration.
  + Supports database interactions for authentication.
* **Authentication Service (AuthService.java)**
  + Registers users with password encryption (BCrypt).
  + Validates login credentials securely.
  + Returns a placeholder JWT token (to be implemented next).
  + Ensures email uniqueness during registration.
* **Data Transfer Objects (DTOs)**
  + **AuthRequest**: Receives login credentials (email, password).
  + **AuthResponse**: Returns authentication success/failure + token.
* **Tested in thunder client:**
  + - Initially 401 unauthorized error due to spring security restrictions
    - Created config package to explicitly allow public access to authentication endpoints.
    - Now, 403 Forbidden errors, so spring security is still blocking access
  + Finalized all data sources except missing sports
  + Ensured proper documentation of the classes is being maintained along the way

**Tasks for next week:**

* + Fix 403 Forbidden issue on /auth/register
  + Continue debugging using IntelliJ logs.
  + Apply necessary fixes to SecurityConfig.java and retest API requests.
  + Once fixed, verify login flow (POST /auth/login).
  + Then implement real JWT authentication to replace the placeholder token.

**Team Member #4:** Aidan

**Total Hours Worked: 7**

**Task Completed:**

* Learned how to use APIs in react and parse json information from them
* Found an api to display company logo images
* Wrote Javascript code for variables and arrays to prepare for storing api information about each company’s stocks
* Made the main page look nicer (with tabs instead of a dropdown) and added some things to the Settings and Profile tabs
* Learned about useState, useEffect and started using them to prepare for sending requests to apis upon user searches

**Tasks for next week:**

* Learn and add apis that the backend subgroup is using (AI API, Yahoo Finance, etc.)
* Get the stock graph to change data for different button clicks
* Link stocks page to main page with react router dom

**Team Member #5:** Unaisa

**Total Hours Worked:** 8

**Task Completed:**

* + Built the basic structure for the Music Hub React app
  + Created the main screen with dark/light mode option
  + Added the different sections (AI help, search, playlists, history)
  + Made the dropdown menu and navigation buttons work
  + Added sample trending songs for display
  + Fixed the blank screen issue by adding the missing router setup
  + Set up the system to save user choices and playlists

**Tasks for next week:**

* Connect to real music data instead of sample data
* Add ability to edit and delete playlists
* Create song pages with play buttons
* Test that all parts work correctly

**Team Member #6:** Aamir

**Total Hours Worked:** 5

**Task Completed:**

* Continued work on the Sports section of the front end and have been working to convert it to work with React framework and javascript and make it so that all components are functioning
* Did some research into how to implement an API and how that will look when it works, especially for sports

**Task for next week:**

* Work through spring break with the others to make sure everything better functions with working APIs and other functionality now that it is working in React.
* Make sure to reach out to everyone to get on the same page as we head into spring break and making sure we can all get to a good spot in our progress
* Work to make sure all parts of the front end work well together and have proper UI/UX design
* Work with backend to implement more features as well such as profile information along with API implementation.