

# RIVALYTICS

## Progress Report 1

By

**Camilo Arias**

300356827

For

**CSIS 4495 - Applied Research Project**

Padmapriya Arasanipalai Kandhadai

Winter 2025

This week, I started by setting up the backend of the project. I installed Node.js which will serve as the runtime environment for the backend, as well as installing the necessary dependencies (Express, dotenv).

Inside the backend folder I also organized the project structure by creating any necessary folders. I also created the server.js file with some basic code just to confirm that everything works properly.

As part of the setup process of the entire project, I also downloaded MongoDB Community Edition in order to work locally instead of on the cloud. As part of this process I also installed the Mongoose library for interacting with MongoDB.

In terms of the actual coding written and committed so far, this is all I have done so far. However, I also brainstormed and made a list of some the API endpoints I will need for the project. I will break it down below:

For all related data about the game's characters (Heroes):

- GET /api/heroes → Fetch all heroes and their stats.
- GET /api/heroes/:id → Fetch details for a specific hero.
- PUT /api/heroes/:id → Update hero stats (win rates, pick rates, etc.).

For matchup data between heroes:

- GET /api/matchups → Get all hero matchups.
- GET /api/matchups/:herold → Get matchup data for a specific hero.
- POST /api/matchups → Add new matchup data.
- PUT /api/matchups/:herold → Update matchup data.

For overall data about the game:

- GET /api/trends – Get meta trends (most picked heroes, etc.).
- POST /api/trends – Update trends based on new data.

For user login/authentication, etc:

- POST /api/users/register → Register a new user.

- POST /api/users/login → Authenticate user login.
- GET /api/users/:id → Get user profile.
- PUT /api/users/:id → Update user settings

**Repo Check in of Implementation completed:** The following files/folders have been committed in the implementation folder:

- node\_modules/
- package-lock.json
- package.json
- server.js

### Next steps:

Since I've already defined most of the endpoints I will need, the next steps involve actually setting up the routes in the code. I will also need to create schemas to define how the data will be stored in MongoDB.

## WORK LOGS

Date	Number of Hours	Description of Work Done
January 16, 2025	1	Coming up with ideas for the project
January 17, 2025	1	In-class consultation about 2 different project ideas
January 21, 2025	0.5	Deciding the idea for the project
January 23, 2025	1	Starting the project proposal
January 24, 2025	1	In-class project consultation and initial check-in
January 24, 2025	2	Defining the project's scope and features after consultation
January 25, 2025	3	Continuing project proposal writing
January 26, 2025	4	Finishing project proposal and submitting
<b>February 7, 2025</b>	<b>2</b>	<b>Coming up with the possible endpoints</b>
<b>February 8, 2025</b>	<b>5</b>	<b>Setting up the backend and writing the project report</b>
<b>February 8, 2025</b>	<b>1</b>	<b>Committing to github repo</b>

