Project Milestone 3

Group 114-4

What features were completed?

- Creation of Pokemon Database and deployment on Heroku
- Creation of Users Database and deployment on Heroku
- Html Pages created: Main page, Description Pages
- Node js server framework established
- Type Advantage/Disadvantage Functions

What worked during the demo?

- User interface, mainpage including different features-the guide, search area, custom Pokemon, building team. Two guides--what is Pokemon, what is Shiny Pokemon.
- Showing Database through pgAdmin and connections occurring on Heroku
- Showing type advantages/disadvantages with C++ functions that calculate specific values.

What issues were faced either during the development or during the demo?

- For backend, we were struggling to create the database that the rest of the team could access and implement into their own code, but we were able to solve the issue by turning to Heroku and making it accessible. Now, every team member can access the database, which is pivotal for storing values and transferring data from client to server.

What were the suggestions offered by the TA?

- ER Model
    - Architecture of database -> visual representation
    - Draw.io or lucidchart
- JIRA board
    - Go back and update work/progress
    - Make proper use of JIRA board
    - Individual contributions -> check user stories, who contributed, etc.
    - Make sure everything is in an epic, has story points, and an assigned user
    - Sprints must be started -> up to date

Individual contributions by each team member

- **Ben:** Made 2 small static html pages(what are shiny and how to play) for the front end. Made a couple of diagrams showing the backend and the overall project. a wire frame as well. Started learning node js and express package for routing the website. Made a small local server script that allows users to send and receive information from the server and change between our html pages.
- **Christian:** Worked with Trustin to continue updating the homepage, user page and the team builder page.
- **Jamie:** Working with Taryn, researched how to implement the Pokemon database and link it with the code that was created by Kaleb. Using PostgresQL, we created the table and connected it to Heroku in order for it to become accessible to the rest of the team. Created Users database and began to look into how to implement a registration system using postgresQL and node.js. Also began to draw out how users will be able to connect/interact with other users with a friending system.
- **Kaleb:** Connected C++ code to database using Heroku, ultimately establishing a connection and creating a table. Also created functions to calculate type advantage/disadvantage values.
- **Taryn:** During the planning period, I made one of three wireframes for the UI of the web application (user profile). My main contribution, alongside Jamie, entailed starting database creation and successfully connecting a PostGreSQL database to the Heroku platform. In addition, we created a Pokemon table (with all 800 Pokemon stored), a Users table, and we are now working on implementing a registration system for the users using node.js. Moreover, we're planning out the implementation for a friend system and how users will be able to interact with one another.
- **Trustin:** Worked with Christian to finish some of the HTML homepage

Interview Time: 6:25 pm - 6:35 pm

Questions:

- Why are these features required? (effective types)
- In UI, how is that compared between 2 Pokemon?
    - Search Pokemon to add to dummy team
    - Add Pokemon to team -> site will tell you the effectiveness (strengths vs weaknesses)
- Why choose C++?
    - Efficiency won't be a huge issue

- We were all familiar with C++