**Enhancement Three: Databases**

Description

For this milestone, I worked on adding a database to my React.js War Card Game. This lets me store player stats like wins and losses so players can track their progress. The database keeps records of each player's performance and updates after every game.

Why I Chose This Artifact

I picked this project because it shows my ability to work with databases and connect them to a web app. Before, the game had no way to save results, so every time you refreshed the page, all progress was lost. Adding a database makes the game more useful and realistic. It also shows I can handle backend development, not just frontend stuff.

Enhancements Completed

Added a PostgreSQL database to store player names, wins, and losses.

Created an Express.js server to handle requests and connect the frontend to the database.

Set up API routes so the game can save and retrieve player stats.

Fixed authentication errors by properly setting up PostgreSQL users and permissions.

Updated the React app to fetch and display player stats from the database.

Did I Meet My Goals?

Kind of, I need to fix some issues with connecting to my database but its mostly working.

I learned a lot about setting up a backend, handling API requests, and debugging database errors.

What I still need to do:

Fix some bugs with the database connection.

Improve error handling so bad database connections don’t break the game.

Make sure the game works properly with multiple players saving stats.

Challenges I Faced

One of the biggest issues was fixing the PostgreSQL authentication error (password failure). It took a while to figure out that I needed to set up the right user credentials and update my .env file. I also had to change how my server listens on a port because at one point, I had another process already using port 5000, so I had to switch to 5001. Debugging and testing database connections was tricky, but I almost have it working.

What I Learned

How to set up a PostgreSQL database and connect it to a web app.

How to send and receive data using an API in Express.js.

How to fix database connection issues and handle authentication errors.

How to update the frontend so it properly fetches and displays database data.

Next Steps

Now that the database is working, I want to:

Improve how errors are handled so the game doesn’t crash if the database is down.

Make a leaderboard system so players can compare scores.

Style the UI better so the stats are displayed in a nice way.