Project Update

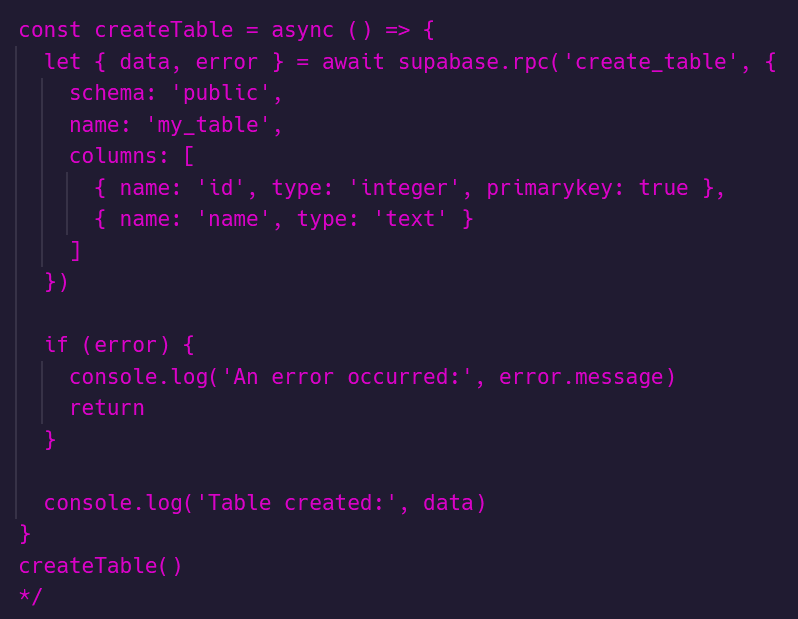
Sellswords and Spellcrafts

Luke Schnetlage, Mason Evans, Nicolas Towery

# Changes/Deviations:

The changes we made to our project and project structure have changed quite drastically. From when we started to this status update, we have almost an entirely different tech stack. When we initially started this project, we were creating our project using Visual Studio code. We were doing so using PHP, Node.JS, and React. The hosting service we were using was iFastNet. We are no longer using iFastNet, as this hosting service lacked a few essential features that we deemed necessary. It lacked the ability to properly implement Node, and once we realized this, we decided to turn to the service’s documentation to try and find some solution. After searching far and wide, we found no such documentation. This was when we realized we needed to change our hosting service. We set our sights on Replit, and in doing so, we had come to the conclusion that the database, which was also structured through iFastNet using PHPMyAdmin, would need to be altered heavily. Before we decided it was time to abandon iFastNet altogether, we tried to find a way to keep the database there, while shifting to Replit for all other purposes, but failed to find an easy fix. After this, we resolved to using Supabase, a database service that is known for working well with Replit. Databases created through Supabase are done so in PostgreSQL, so we had to modify the structure of the database in some ways since the types in PostgreSQL are different from MySQL. Below, you will see the ERD for our Supabase database.

In regards to the languages that we discussed earlier, we came to the conclusion that rather than trying to find some way to make two different languages work together, we would just phase out PHP and simply work only with JavaScript based code to allow for easier implementation, comprehension, and to make it easier to modify.

Since switching to Replit, we now have a way to work in tandem on what is called a “Team Repl”, being able to “spectate” each other while we code thanks to Replit’s “Teams” feature. We are also using Replit’s built-in authentication, which allows us to easily implement a user authentication system for our project, which will allow us to filter out users who have not signed in, and they will not be able to access certain parts of the website until they do log in through Replit. Any database accesses are done using the Supabase library in JavaScript. See the image below for an example of how we did this. 

It is worth noting that while we originally intended to create our game artwork with pixel art created by our own team, we have since decided that due to time constraints, we would switch to using the Midjourney art tool. We ensured that the licensure and/or copyright for all artwork created through this tool was fair use, so as to avoid any form of infringement. For additional details regarding copyright for Midjourney, please read the documentation.

# Accomplishments:

Taking into consideration the level of modifications that were made to the project as a whole, we can now look at what we have been able to complete. Firstly, we have been able to successfully structure our database and implement successful calls in our Repl to pull, insert, or modify information in our database. We have also successfully implemented the ability to connect users for a game. We have done so using a socket connection in our Game.jsx file, which is where all game-based implementations go. When entering the “Play” section of our website, users can challenge other connected users via a connected users list. Upon challenging another player, users can now select a deck from their choice of void, wind, fire, or water. In regards to the website itself, we have adapted it to be responsive, allowing for a smooth transition from a computer to mobile device.

# To-Do List:

Now we can discuss what we have left to accomplish for the original version of this game to be complete. Firstly, we need to finish fleshing out the game logic so as to allow the successful creation, participation, and completion of a game. We need to finish designing all card decks, as we only have a select few decks implemented. We have completed the design of all minion cards in our game, but we still need to complete the invocation cards for the game. We will do so via prompts through Midjourney, as discussed previously. We will also implement a custom login page, as Replit allows for any developers to design and implement custom login pages through their authentication service.

# Issues to Resolve:

Regarding issues we have yet to resolve, we need to find a way to account for large amounts of users connected to play games via the socket connection. We have had no issues as we are a small team testing connections, but if many people, for example, 50 or more, decide to connect, the server may run into issues. We need to resolve the issue of deck selection, as we only have implemented the water deck. If a user were to select a different deck to try and get into a game, there would be a number of errors, since we have no implementation of the other decks. These are the only major issues we have for this version of the project.