2048 Game

Ananya Banerjee

2048 Game

Talent Sprint

October 16, 2023

Introduction

- 2048 is a popular tile-matching puzzle game. The game is played on a 4x4 grid, and the objective is to combine tiles with the same number to create a tile with the number 2048.
- With that, a login page was also developed, which will enable you to sign up or Login into the game.

Tools Used

1. HTML: HTML (Hypertext Markup Language) is used to structure the basic layout of the game. In a 2048 game, HTML may be used to define the game board as a grid of cells. Each cell represents a tile in the game.
2. CSS: CSS (Cascading Style Sheets) is used for styling and layout. You can use CSS to make the game look visually appealing by customizing the colors, fonts, and overall design. It's also used for responsive design, ensuring that the game scales properly on different screen sizes.
3. JavaScript: JavaScript is the core technology used to implement the game's logic and functionality. Here are some key aspects of how JavaScript is used in a 2048 game:
    - Game Logic: JavaScript is used to handle the game's logic, including the movement of tiles when the player swipes or presses arrow keys. It manages the merging of tiles with the same values and calculates the player's score.

Login Page - Tools

1. Express.js: Express.js is a popular web application framework for Node.js.
2. Handlebars (hbs): Handlebars is a templating engine for JavaScript. It allows you to create dynamic HTML templates.
3. MongoDB: MongoDB is a NoSQL database that is used for storing and managing the data of the people who logged in.

Learning

- The HTML, and CSS were comparatively easier as I had prior knowledge.
- For JavaScript, there were multiple resources available hence I figured that out in a week.
- MongoDB configuration was also moderate, using the resources online and trainers' help.
- I had no prior experience with Node and Express, hence it was difficult to configure it.
- Using Handlebars was also new to me, hence it took some time to learn, but it is very similar to HTML.
- How to work with Google Authentication.

Setbacks and common mistakes made

- I did not download the mongoose module as I thought downloading mongodb would be enough.
- The frontend was not giving data to the backend, which I figured out using Postman, I used ChatGpt's help to add an extra function to the code and it started working.
- After fixing that, the HBD file was not executing its CSS and JS, after using online resources, I learned that adding an extra function to make the nodejs using static pages fixed the problem.
- I made the entire login page in the Google Oth Services. But I was unable to connect it to the backend due to my lack of knowledge and hence could had no choice but to start working on my nodejs and mongodb.
- A More detailed explanation of all of this is in the ReadMe files.

Time Line of my Project

- Week 1 - I finished the HTML, JS and the CSS of the entire project.
- Week 2 - I started working on the Login Page, started learning NodeJS, and also exploring Google Authentication.
- Week 3 - Started working on my Login Page and finished it with a few problems.
- Week 4 - Finished the Login Page and made the Readme files explaining everything in detail.

Learning and References

- https://www.youtube.com/watch?v=XM2n1gu4530ab
- https://www.freecodecamp.org/news/how-to-make-2048-game-in-react/
- https://projects.raspberrypi.org/en/projects/cd-beginner-javascript-sushi/7
- https://www.youtube.com/watch?v=V8dYGNfHjfkab
- https://github.com/kubowania/2048
- ChatGPT