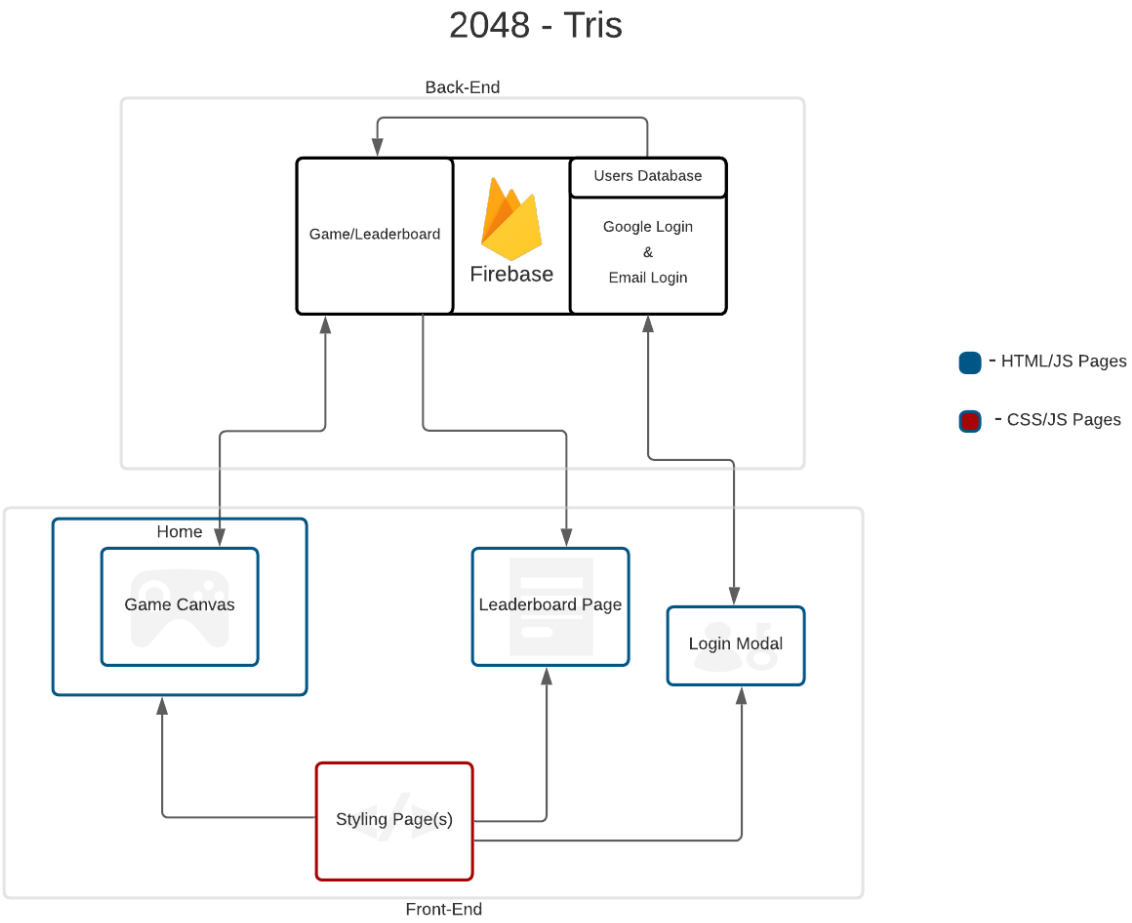
Revised List of Features

1. Game functionality
    a. 2048 and Tetris will be played simultaneously. This will involve two windows taking in user arrow key inputs that move the individual game pieces. It's great.
2. Login system
    a. Storage of password / username so you can login and instantly do entry into the leaderboard system.
3. Leaderboard system
    a. Keep the top 3 scores of every account and display them in a leaderboard tab.
4. Tutorial
    a. Simple thing that teaches how to play the game. Highlights key binds and how they affect each side of the game.
5. Settings
    a. Change themes (incl dark mode).
    b. Account management

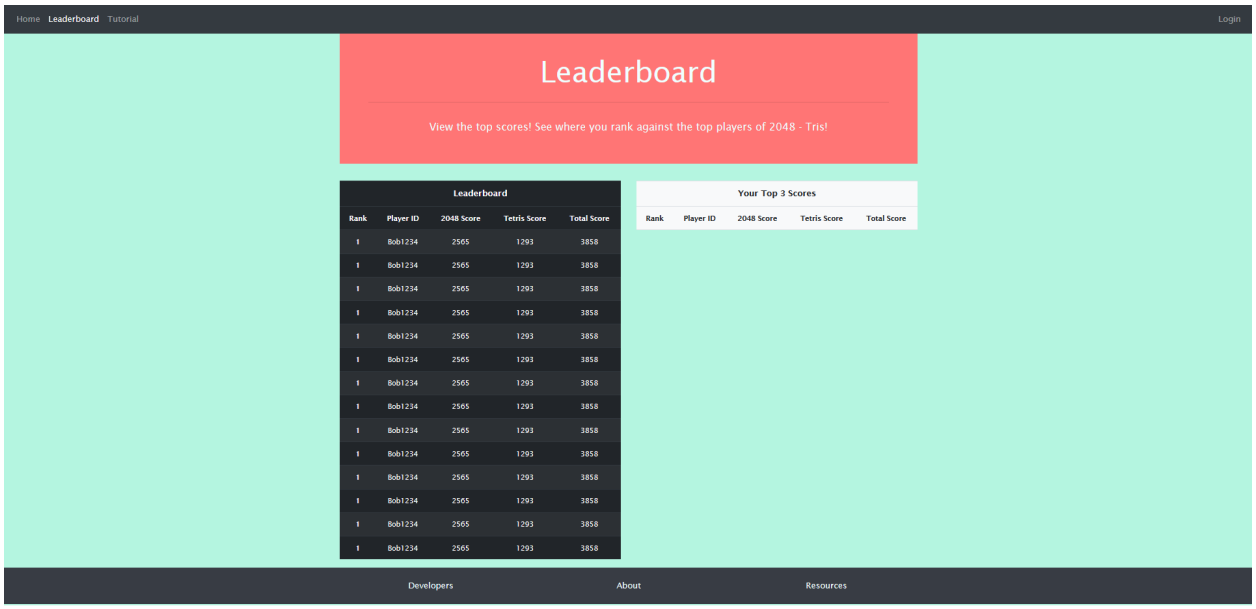
Architecture Diagram



Front End design

Home:

Leaderboard:



Tutorial:

Home Leaderboard **Tutorial** Signup Login

How to play: Tetris

Objective: The goal of the game is rotate and move differnt block shapes to aline them in rows at the bottom of the game board. Be careful! Make sure you fill up some rows before the blacks stack to the top!

Tetris Official Rules

How to play: Tetris

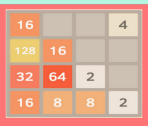
Controls: This game is run by five simple controls: ... ands,abdbwoadbow nd nwndoidnwai noinodiwadon aondoan on noan

How to play: Tetris

Scoring Points: Points are scored by ...

Developers About Resources

Home Leaderboard **Tutorial** Signup Login



How to Play: 2048

Objective: It's quite simply really! Your goal is to sum up blocks until you produce a 2048 block! Watch out though, only blocks with the same number can combined!

How to play: 2048

Controls: The controls for this game are ...

How to play: Tetris

Scoring Points: Points are scored by ...

Developers About Resources

Home Leaderboard **Tutorial** Signup Login

Playing them Together

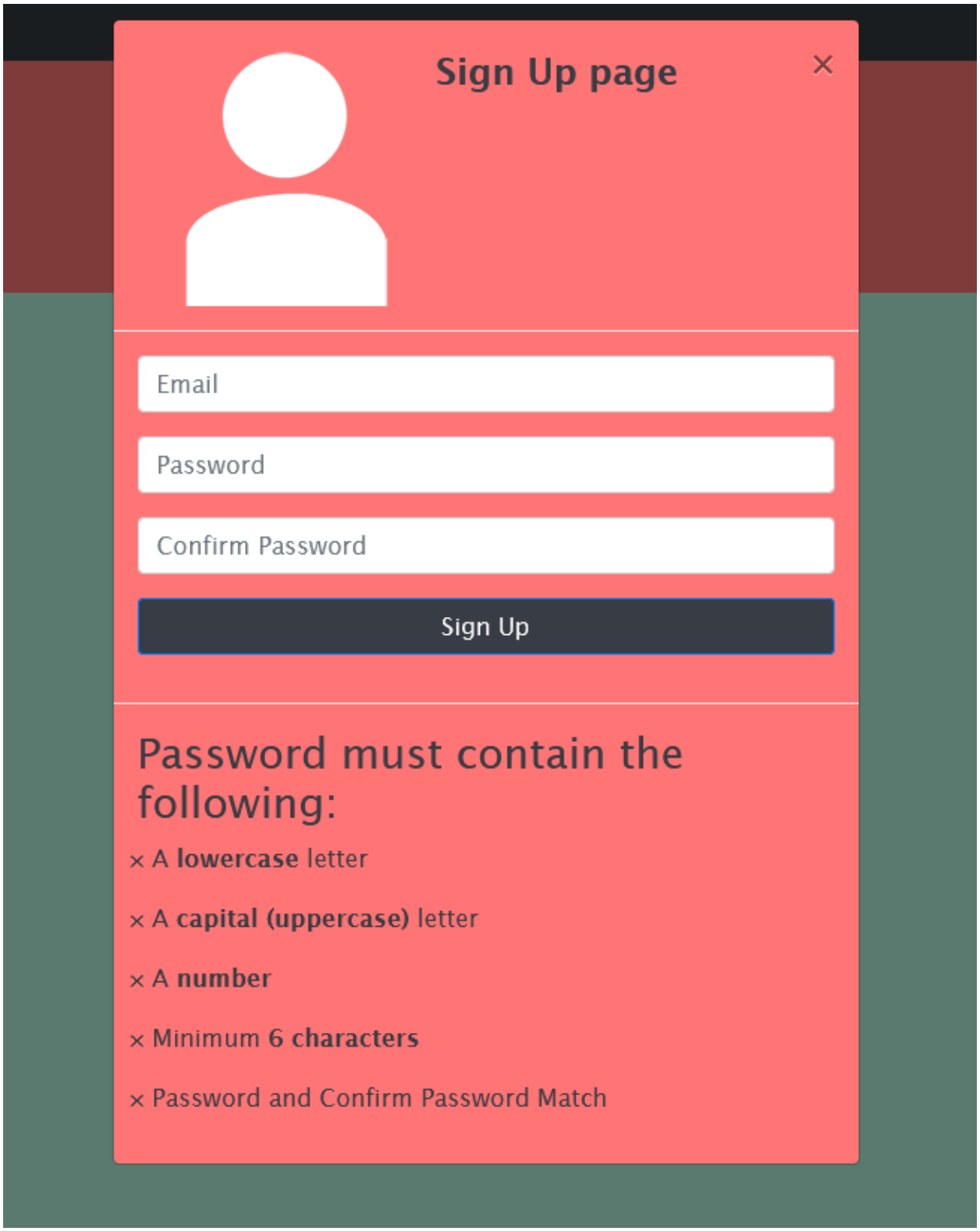
The real challenge here is playing the games at the same time!! Tetris and 2048 are simple but challenging puzzles on their own, however, it takes a higher level of brain power to successfully operate both games simultaneaously! Do you have what it takes?

Playing them Together

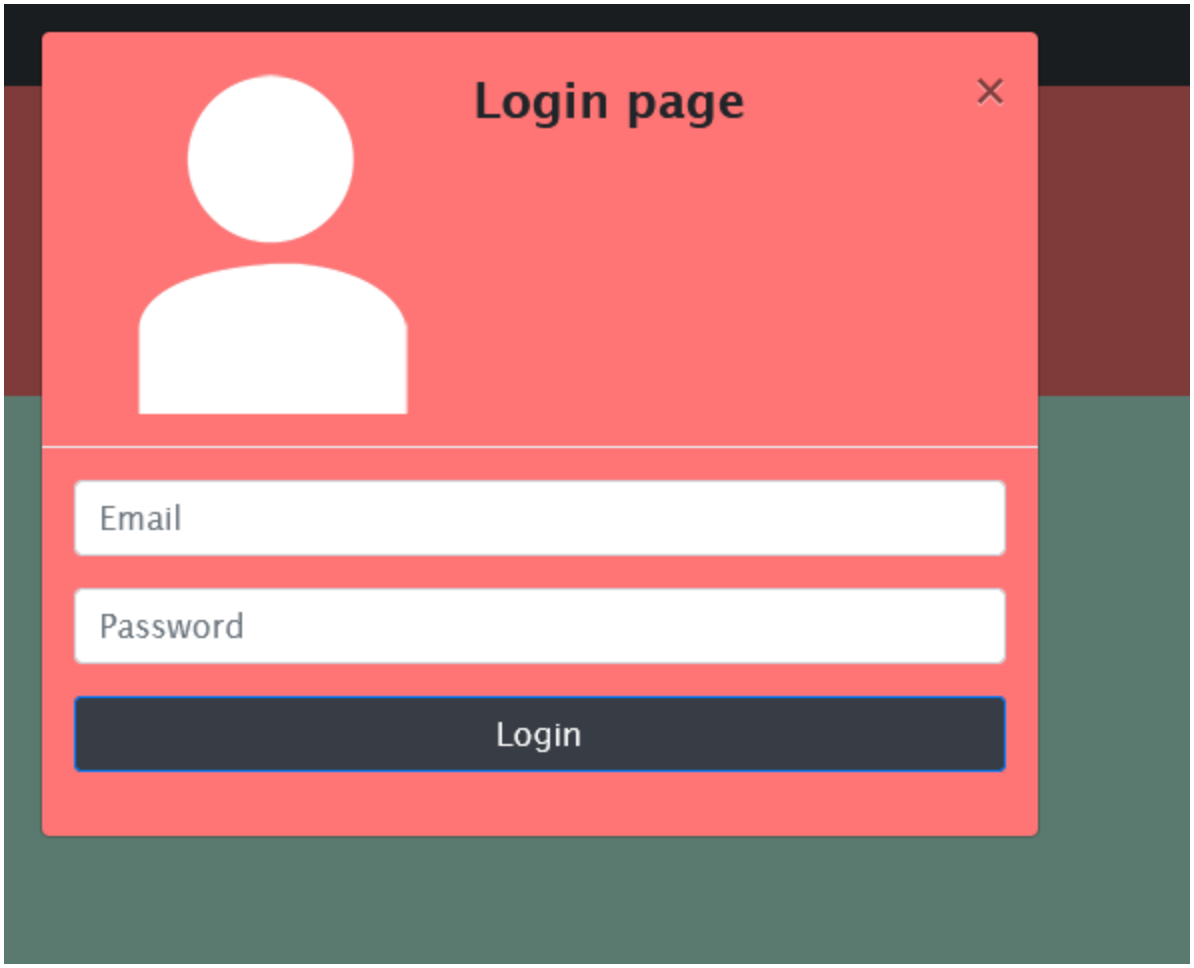
The controls are the same as previously stated for each game, but just keep in mind that each key presss will affect both games! Keep track of where your blocks are falling in Tetris, but be aware, you may move the 2048 tiles in the wrong direction!

Developers About Resources

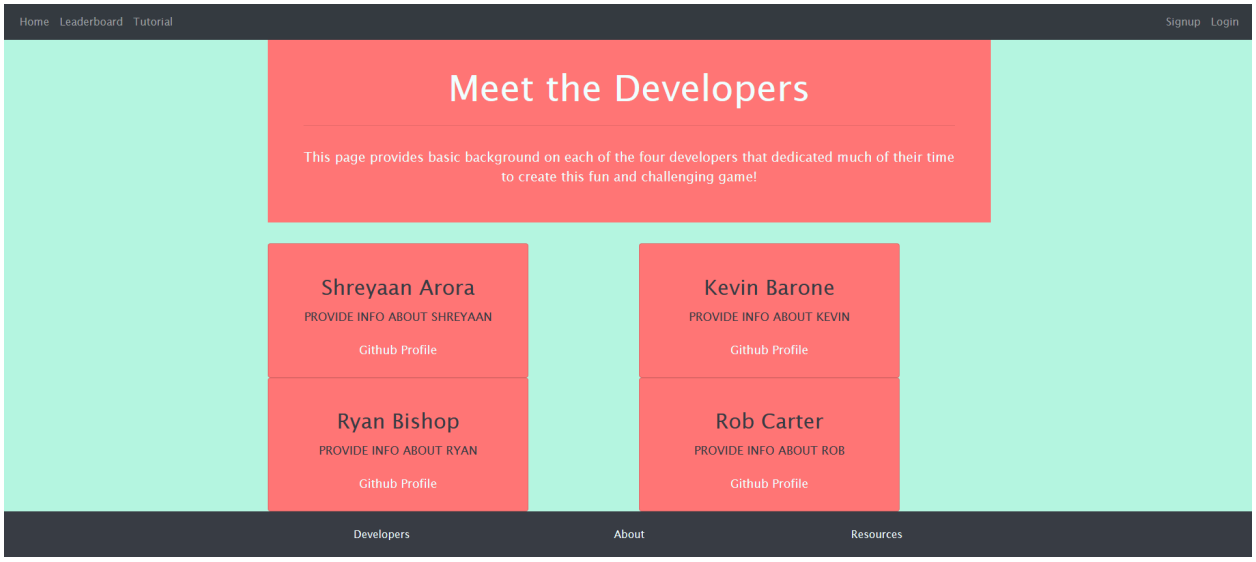
Signup:



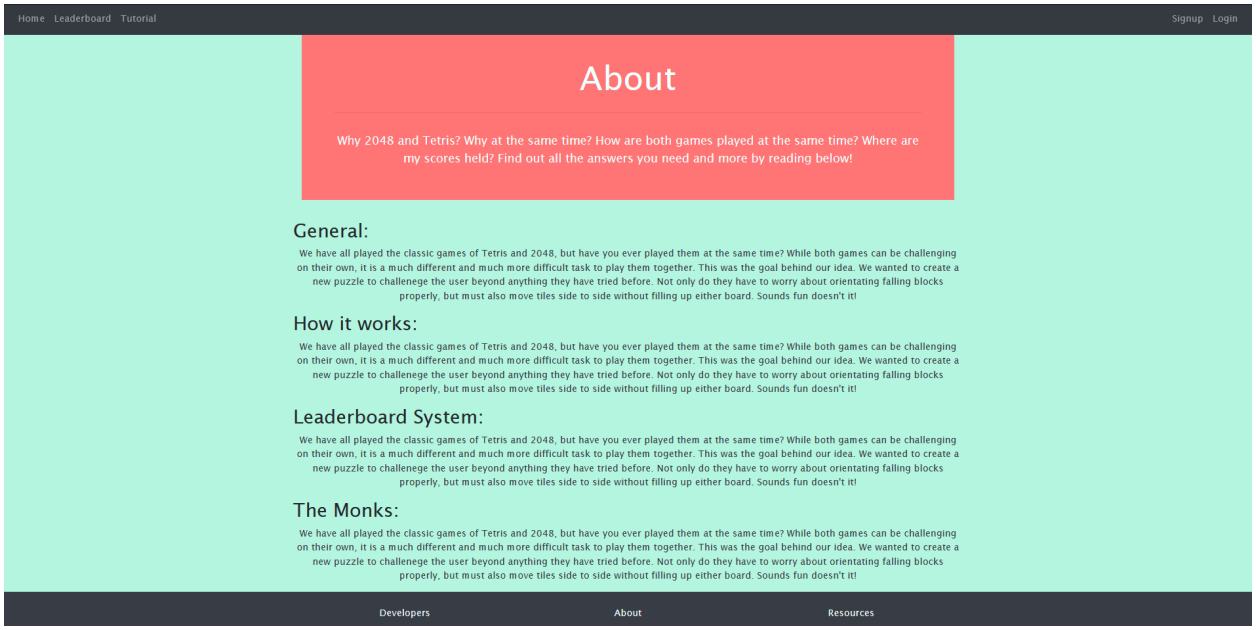
Login:



Developers:



About:



Resources:
To be developed. Will be similar to Developers and About.

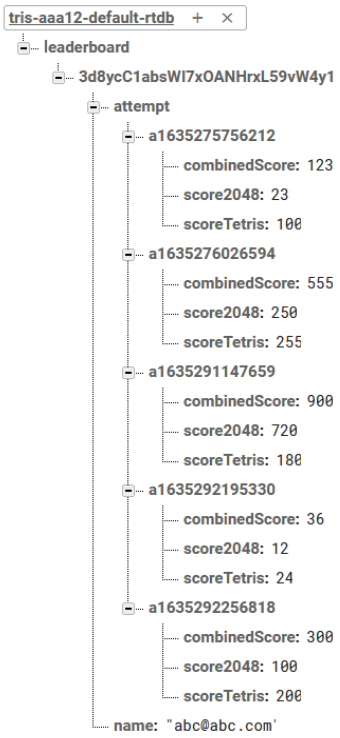
Web Service Design

We are using the firebase auth to create an account with emails, login and check if a user is logged in or not.
We're not using too many APIs in our project yet.

Database design

For our database we're using the realtime database provided by Firebase. Our database stores the unique id provided to each user by the firebase auth as the key for the data. Within the user id there is a name entry that the user makes at sign up to display on the leaderboards and personal top scores. The score achieved in 2048, tetris, and their combined scores are stored within each play the user does. Our datamodel.yaml below is an example of 1 user that has played 3 times and to the right of it is a screenshot of our current database with test data.

```
leaderboard:
 user_id:
   name: string
   attempt:
     a1:
       combinedScore: number
       score2048: number
       scoreTetris: number
     a2:
       combinedScore: number
       score2048: number
       scoreTetris: number
     a3:
       combinedScore: number
       score2048: number
       scoreTetris: number
     ...
```



The user_id entry is a string that is unique to each signed up user (like 3d8ycC1absWI7xOANHrxL59vW4y1). This is used to avoid multiple users somehow storing scores in another user's scores. Each attempt (a1, a2, … , an) is a string in the database created by ‘a’ + the current time in milliseconds (like a1635275756212). This was done to make sure that scores are unique and will not overwrite other scores. The structure of the database is like this in part because more than 1 database on firebase requires a paid membership and this reduces the number of reads and writes that must occur compared to our previous implementation where personal scores and leaderboard scores were separated causing redundancy in data.

Challenges

1. We had issues with loading the website. Solved by using the Live Server extension in Visual Studio Code
2. We had issues with merge conflicts. Solved by messaging the Discord server whenever we were pushing code.
3. Had issues figuring out how to sort/order the data retrieved from the firebase realtime database as I had no idea what an object literal was before this and how to access it.

Individual Contributions

Kevin - Worked on Tutorial, Developers, and About pages. Made everything look nice and still working on this https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-017-07/commit/c2cf5fb9006134fffe1057b646d2cd80fd4fde59

Rob - Further worked on Firebase and our database implementation. Made a yaml document. https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-017-07/commit/654cfa21ec54158243d54ffc7e7abd3c888e6fce

Ryan - Worked on the game implementation. https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-017-07/commit/625c6fa3eba7f32aa3f656f1400d576cbcc190b4

Shreyaan - Worked on signup modal javascript and architectural diagrams.