Numbernauts!

=========================================================================================================

- This game uses react, react-hooks and react redux to generate a gaming expeience for kids to practice their multiples and factors in a fun way.

- This game also uses mongo db to track high scores and keep usernames for the kids to login and keep track of their scores.

- The game also uses socket io to allow 2p head to head.

How it works:

There is a base react tool that will create starts with a user login page. This logon page will either store or read user name and password so kids can login. There is no contact information nor is there personal data, the user account merely to keep track of the user score.

After login, the user is taken to a lobby built by socket io. From there they can decide either to start a small one player game or join a room created by another player.

* Single player game
  + There are various components at play here but the main idea of the game is that the levels are spontaneously generated from criteria in an array of levels. Starting at level 1 (position 0 in the array), the answer array will then be built at random from a number generation of possible numbers 0-400. Half will be answers that are correct and the rest will be false. The user needs a total number of 15 correct answers to proceed to the next level and criteria, however 3 wrong answers will end the game and the user will have to start back over at level 1.
* Multiplayer game
  + Both users are loaded into the room with both of them on the board at the same time. The board is built the same way as in the single player game but the level is built at random. The users must now compete for the most right answers but the game is not over until the board calculates 15 correct answers.

The game itself consists of several levels of components reading state and storing it redux and using hooks for shorthand here and there. The start is the Game component that pulls the information on the criteria and then stores that in an array to be used by the children. From there it loads the board.

The board itself mostly exists to load the grid and modals for the end of the game. If the user has won the level, the next level modal will show and if the user loses, a Game Over modal will show. The losing modal will then check the users score against their top score in the database and store it in the database if it is higher. The board is what loads the grid with is the more visual component in the game.