Project Charter

Not So Trivial Pursuit

Gaspare, Younes, Jazz

Aug 30th, 2023

Description:

The project aims to create an engaging and customizable trivia game web application that allows users to generate and play trivia games for parties. The application will utilize the Open Trivia Database API for question retrieval and Firebase for data storage and user interaction. Users will be able to select question categories, set the number of questions, and save their game progress for future play.

Objectives:

- Develop a user-friendly trivia game web application.
- Integrate with the Open Trivia Database API to fetch trivia questions.
- Utilize Firebase for data storage and user authentication.
- Allow users to customize trivia game settings, such as category and number of questions.
- Enable users to save and revisit their game progress.
- Implement error handling for API calls and user interactions.

Team communication:

1. When are the members meeting?

Due to variable work schedules, the team plans to complete most of the work asynchronously and regroup during class hours to pair (or rather, trio) program and resolve any roadblocks or issues that arise. Should more time be needed, all members are available during evenings on slack, and zoom.

2. Main method of communication?

We plan to use a variety of tools for communication such as slack, zoom, and of course Asana for task management i.e. keeping track of weekly/biweekly goals and ensuring progress is made without exceeding deadlines. Tuesdays and Thursdays will be our check-in days for Asana, but we plan to be available to message daily thanks to the aforementioned apps (isn't technology great?).

3. How are roles and duties allotted?

As mentioned earlier, our task management system, Asana, will undergo biweekly updates to outline individual responsibilities for each team member. While we won't adhere to rigidly defined roles, each member will be assigned tasks that encompass various facets of the project, ensuring a comprehensive and collaborative approach.

4. Calendar and deadlines?

Project Charter Due: Tue, Sep 5th, 2023

Final Dev Branch Update: Tue, Sep 26th, 2023
[this is when we can begin working on debugging/testing to ensure quality goals are met by launch]

Final Project Deadline: Tue, Oct 3rd, 2023

Anything further will be managed in Asana!

Scope:

MVP

Single player (ideally as a component, to be scaled later for multiple players as a stretch goal)

1. Initialize Firebase
   - Import and initialize the Firebase SDK with your project's configuration.

2. Create State Variables
   - Create state variables*, such as:
   - questions (array to hold fetched questions)
   - selectedCategory (string to hold selected category ID)
   - numOfQuestions (number to hold selected number of questions)
   - loading (boolean for API loading state)
   - error (string for error handling)

//* example variables, may change in development process

3. Fetch Categories from Open Trivia Database API
  - Use an effect to fetch categories on component mount
  - Store categories in a state variable

4. Define a function ton fetch questions from the API

- Inside the function:
- Set error to an empty string
- Make an API request to the Open Trivia Database API using state variables
- On success:
- Set questions state with fetched questions
- On error:
- Set error state with an error message

5. Define saveGame Function (MVP = leaderboard only)
   - Define a function named saveGame to save the current game to Firebase
   - Inside the function:
   - Create a reference to the Firebase database collection for saved games
   - Add the current questions and relevant data to the collection

6. Create JSX for the App Component
   - Render a div element containing:
   - An h1 heading with the title of the app
   - Inside another div:
   - A select element with options for categories (map over categories state)
   - An input element for selecting the number of questions (controlled by numOfQuestions state)
   - A button to trigger the fetchQuestions function
   - If questions array has elements:
   - Render the fetched questions with options for a trivia game
   - Render a button to trigger the saveGame function

7. Implement User Interactions
   - Attach event handlers to the select element (onChange)
   - Update selectedCategory state when a category is selected
   - Attach event handlers to the input element (onChange)
   - Update numOfQuestions state when the input changes
   - Attach an event handler to the "Generate Questions" button (onClick)
   - Call the fetchQuestions function
   - Attach an event handler to the "Save Game" button (onClick)
   - Call the saveGame function

8. Export the App Component
   - Export the App component as the default export

9. Styling and UI work

Stretch goals:

10. Use Firebase Authentication (Optional)
    - Implement user authentication with Firebase Authentication if needed

- Allow users to sign in to save and view their games

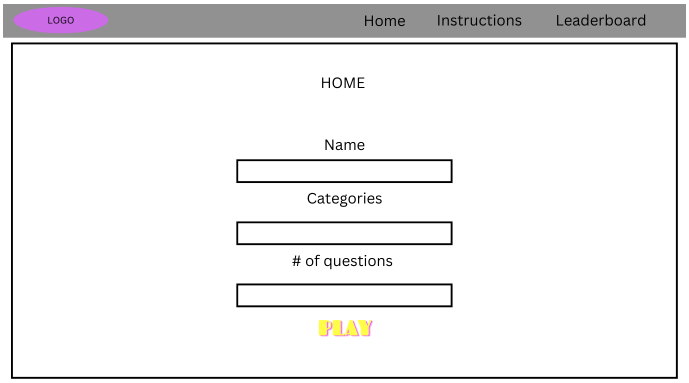
11. Add loading animations

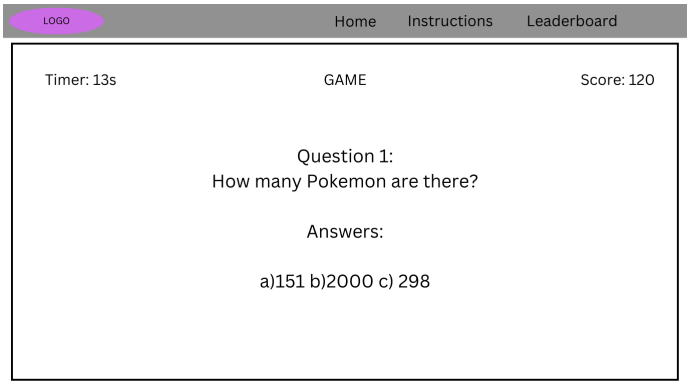
12. Enhanced UI animations

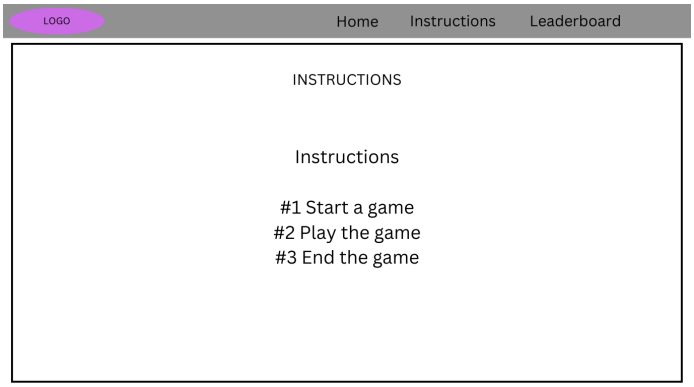
13. Users can set the difficulty

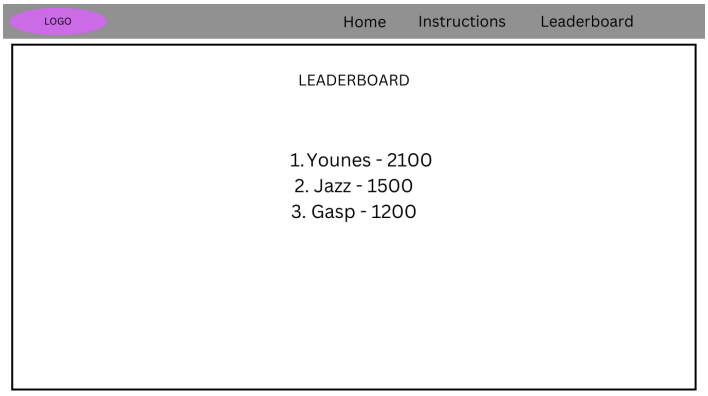
14. Users can mix question types (MC and True/False)

Wireframe:









Styling guide:

Color Palette

#577F90	#86C2AE	#F0917A

Typography

Headings: Caveat Brush (bold)

Heading Text

Body font: Montserrat (normal)

Enet acepudia delit faciure stiaturem doles proris earist fugitatem fuga. Nem quae labor sim quuntota veriorem quis di od magnimodit, site voloreium ullantiam nem dolupti or.