Final Project CSC337 - Trivia Game

Garret Wilson, Hengsocheat Pok, Dhruvi Ketan Rathod, Yashi Gupta

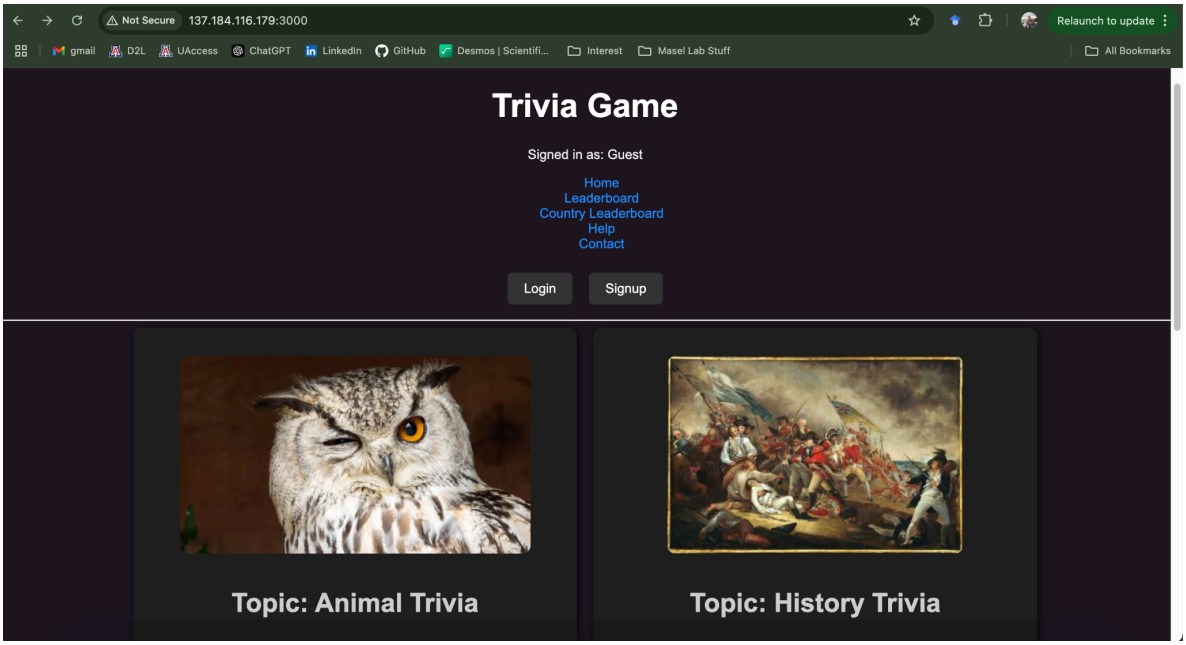
---

Overview

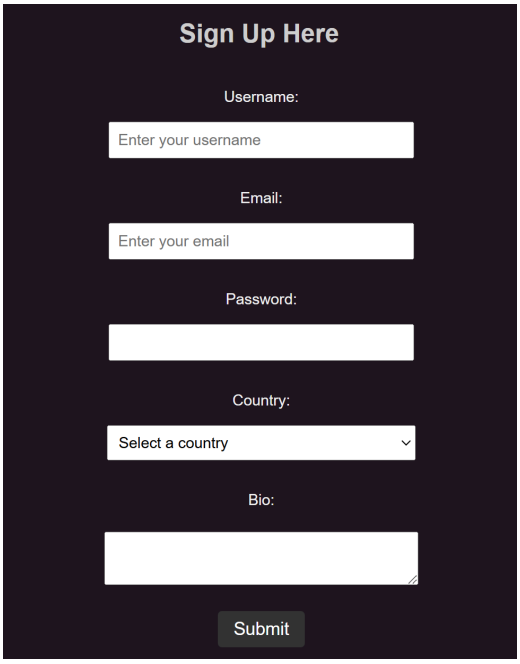
We have created a trivia game where players can choose questions from four different topics (Animal, Plant, World History and Cultural Studies) with different subcategories of differing difficulty levels. Points are awarded based on the difficulty of the selected question, with higher difficulty yielding more points. A leaderboard tracks player scores, fostering friendly competition. There is also a country leaderboard that tracks and ranks the scores of different countries by accumulating all the scores of the users from that country.

Frontend

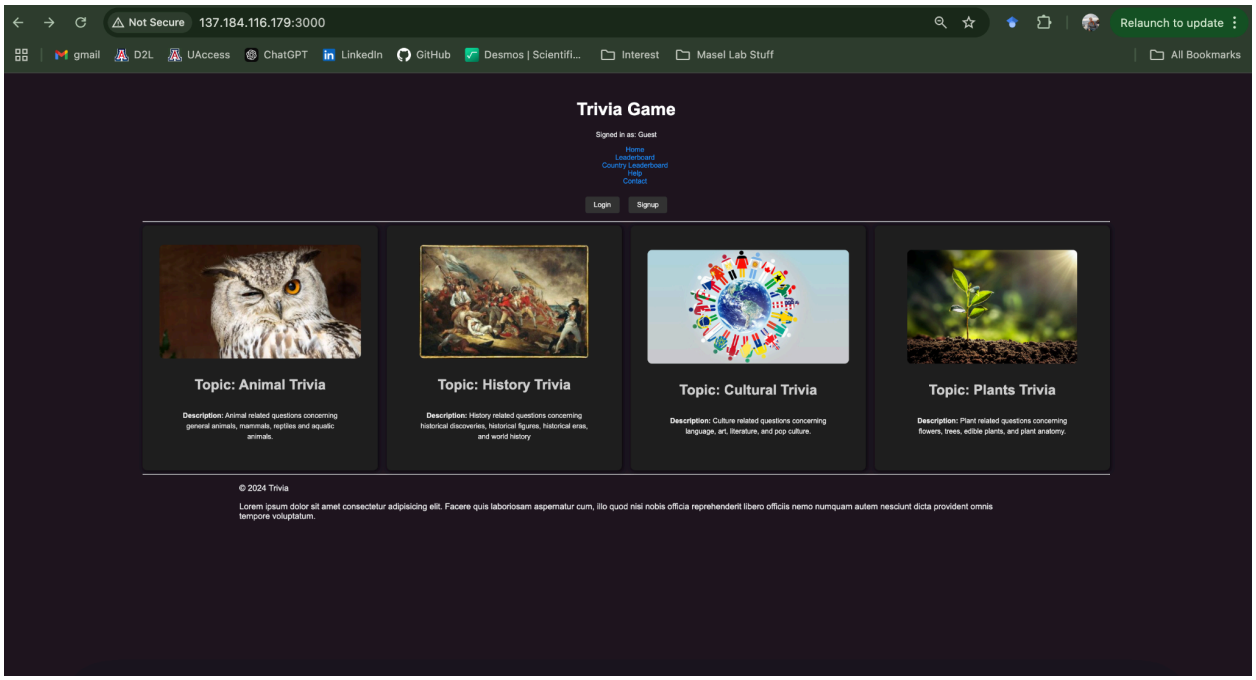
When visiting our server in a web browser, the user will be greeted by this screen with an option to log in or sign up.



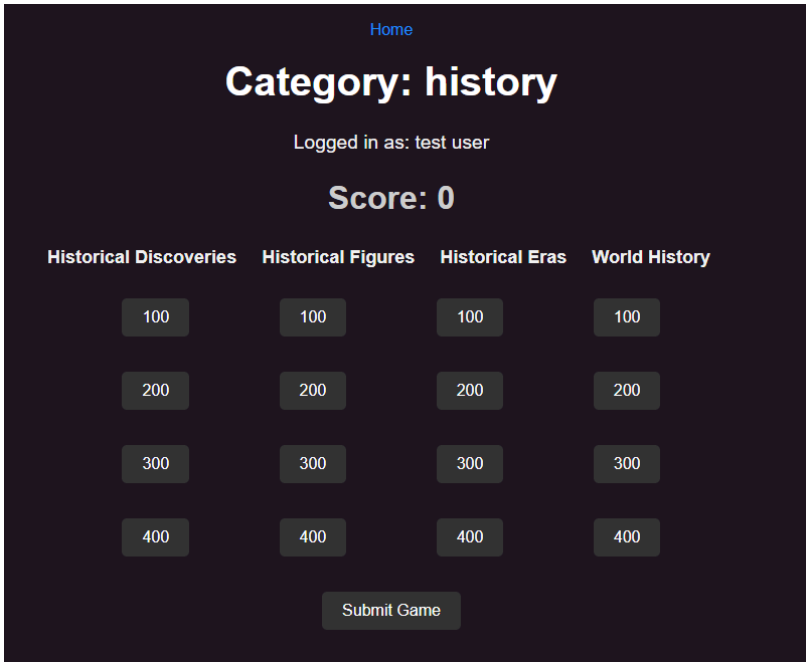
All users will be signed in as ‘Guest’ at the beginning, meaning that their scores will not be saved and they cannot appear on the leaderboard if they choose to play without signing in. To get their scores recorded, a first time user will have to sign up for an account as shown below.



Once the user has logged in (if they are not a first time user) or created an account, they will enter the website’s main screen. At the top are the links which can be clicked on to go view the player leaderboard, country leaderboard, help page and more. Below, the four main categories are displayed. The user can click on the category they desire and answer the questions.



Upon clicking on a category, the user will be redirected to the game page which will appear as a grid as shown below. Each column represents questions from a particular subtopic defined by its heading (eg. historical discoveries). The numbers represent the difficulty and points awarded for answering the question correctly. The lower the number is, the easier it is to answer. The user can click on any difficulty level for any subtopic and answer the question. For every question they get correct, their score will be incremented. After the user is done answering the questions (as many or as little as they would like), they can click on the ‘Submit Game’ button. Clicking this button will add and save the users’ total score from this game to their overall score. Clicking ‘Home’ will lead them back to the main page.



Clicking on ‘Country Leaderboard’ on the main page will lead users to the leaderboard for countries. For this leaderboard, the scores of all users belonging to that country will be accumulated and ranked as shown below. Clicking on ‘Leaderboard’ will give a similar result, but this ranks players' individual scores.

If players need help, they can click on the ‘Help’ link on the main page, which will lead them to the help page which has commonly asked questions and some useful links as shown below.

Home

Help & Support

FAQs

What do I do when I enter the website?

When you enter the website for the first time, do sign up and create an account. Subsequently you can click on any of the main topics on the home screen and select with topic and which difficulty you would like to play.

How do I play the game?

Select a topic among the 4 options available. After clicking on the topic card, you will be led to a screen with a grid full of numbers. The numbers represent the difficulty and points of the question and the subtopic is defined by the header of that column. Click on the difficulty level you desire for any of the four subtopics and answer the question. You will be led back to the grid after submitting the question and you will not be able to answer a question from that subtopic and difficulty any longer. Once you are satisfied that you have answered all the questions in the topic that selected, click on 'Submit Game', which will trigger an update of your score.

Do I need to log in?

You can browse some parts of the site without logging in, but to save your scores and compete on the leaderboard, you need to create an account and log in.

How can I see my score in relation to others?

After logging in, visit the leaderboard section to compare your scores with other players and see your rank on the global or country leaderboard.

Useful Links

Sign Up

Leaderboard

Country Leaderboard

Backend

Tools:

- **React:** A JavaScript library that may be used to make client-side web development more organized and user-friendly.

- **JavaScript:** Essential for client-side functionality, regardless of React integration. It will handle user interactions such as:
    - Displaying questions based on selected topics and difficulty levels.
    - Managing multiple-choice answers
    - Sending and receiving asynchronous requests (e.g., accessing files or databases).

- **HTML:** Responsible for creating web page structures, including:
    - Login/signup pages.
    - Main game interface.
    - Results and leaderboard displays.

- **CSS:** Will define the visual style of the website, emphasizing:
    - Flexbox layouts and semi-transparent containers for a modern aesthetic.
    - Smooth transitions to enhance user experience during key gameplay moments.

- **MongoDB:** A database for:
    - Storing leaderboard data to rank users by scores.
    - Storing country leaderboard data to rank countries by scores.
    - Storing user data like username, email etc.

- We have three schemas. Our user schema is as shown below:

```
const userSchema = new mongoose.Schema({
    username: { type: String, required: true, unique: true },
    email: { type: String, required: true, unique: true },
    password: { type: String, required: true },
    country: String,
    bio: String,
});
```

We have two other collections - one for our leaderboard, which stores a reference to the userId as well as that user's score and another for the country leaderboard, which stores the name of the country and the total score for that country.

- **Express:** A framework for Node.js to simplify and streamline HTTP request handling, making the backend easier to develop and maintain.

- **Node.js:**
    - Facilitates file reading to load question banks dynamically.
    - Acts as a server-side interface for database interactions, such as retrieving leaderboard data.

- **Mongoose:** A Node.js library that provides a schema-based approach to interacting with MongoDB. It simplifies database operations with features like **Schema Definition**: Enforcing the structure of data stored in MongoDB, **Validation**: Ensuring that only valid data is saved to the database, **Query Simplification**: Making database queries more efficient and readable, **Middleware**: Allowing pre- and post-processing during database operations.

- Our questions, their answers and its points are stored in a json file, which was read by our server javascript file and then fetched when a user is on a game screen.

Note

- Inside of our VM we have a **server.js** file that is responsible for handling all requests from the client side to the mongoDB and another file within the VM called triviaData.json. Since this file is not available without access to our VM, we have included it in our Trivia-App/src project folder. The server.js inside the project folder is not used as it is only for the grader's review purposes. It should be the exact duplicate of the one inside the VM.

Reflection:

<u>What went well</u>

- We successfully implemented all the planned functionalities, along with an additional feature: a **country leaderboard**, which adds extra value to the project.
- The website's layout and design are intuitive, ensuring a clear and logical user experience.

<u>What could have been done differently</u>

- **Time Management**: While the project was completed, better time management could have allowed us to address the critical aspects earlier in the development process.
- **Prioritization**: Starting with the most important components sooner would have reduced pressure toward the end and provided more time for refinement and testing