

Tanzim Azad
Eric Osicka
Andrew Best
Blaizun Diamond
Ryan Slocum

Tuple Trivia

Project Description:

Tuple Trivia is an online, fast-paced trivia game that was developed by our team in order to give friends and family the chance to test their knowledge of random facts. We used a fairly simple structure for our project which includes front end, logic, and backend. For the front end, we used EJS (templating tool) and CSS (styling) to achieve our iconic aesthetic consisting of yellow and red themes. For the logic, we built a NodeJS web server that consisted of multiple endpoints that rendered the EJS pages and allowed players to play simultaneously. We also built the web server to use API calls to retrieve questions of different difficulties and categories from the-trivia-api.com. For the backend, we used PostgreSQL to store user and game information. This was used to keep a historical record of players' scores and user information. After all of these features were fully developed, we used docker containers to be able to easily deploy the application on Microsoft Azure. Azure proved to be quite the challenge because none of us had experience with the website let alone deploying an application in general. Despite these challenges, we were able to get everything working and running smoothly.

Project Tracker - GitHub project board:

The GitHub project board displays the following items across four columns:

- To Do**: 0 items
- In Progress**: 0 items
- Testing**: 0 items
- Done**: 17 items

Items in the Done column include:

- CSCI3308-FinalProject-011-02 #21 Leaderboard Frontend (Leaderboard)
- CSCI3308-FinalProject-011-02 #7 Trivia Page Skeleton (User Interfaces)
- CSCI3308-FinalProject-011-02 #37 Two Player Trivia gameplay (Multiplayer)
- CSCI3308-FinalProject-011-02 #46 Multi-player backend score recording (Multiplayer)
- CSCI3308-FinalProject-011-02 #10 Trivia Options Backend (Trivia Options)
- CSCI3308-FinalProject-011-02 #12 Main Database (Database & APIs)
- CSCI3308-FinalProject-011-02 #47 Multi-player leaderboard (Leaderboard)
- CSCI3308-FinalProject-011-02 #17 Play Page Frontend (Play Page)
- CSCI3308-FinalProject-011-02 #20 Leaderboard Backend (Leaderboard)

Link to project board:

<https://github.com/users/RyanS161/projects/1/views/1>

Demo Video:

https://cuboulder.zoom.us/rec/share/FQOk7jAsnp0esJ1_ACvgiAuNt6twbKuVtO4yEoKY_o9I7VOlZh1q7EbiciO7GhxmM.uBtUvy1jsi5HBScj?startTime=1683134621000

Link to VCS:

<https://github.com/RyanS161/CSCI3308-FinalProject-011-02>

Contributions:

Eric Osicka - I mainly worked on the front end and touched some of the back end as well. For the front end, I played around with the home page, the register UI/UX, and some of the CSS styling for the overall theme of the website (the navbar was a pain). For the backend, I worked on building the API call and endpoint for the game settings and building that logic into the EJS.

Ryan Slocum - I primarily worked on the gameplay logic on both the frontend and the backend. I started by using the trivia api to pull questions for single player games, and implementing the frontend for the gameplay. When the players finished the game, I added endpoints to record their scores and send them to the database. I then extended this functionality to work with multiple players with the socket.io library, allowing more than one person to play with the same questions in sync with one another.

Andrew Best - I worked mostly on backend and frontend for the leaderboard as well as the how to play page. I also wrote the test cases for the registration page. For leaderboards I made most of the leaderboard page as well as some of the leaderboard routes. As for the how to play page that was completely html, adding just a basic description on how to play on tuple trivia.

Tanzim Azad - I mostly worked on frontend stuff mainly with the pages attached to the headers like the about me page. I also wrote user stories and test cases that we should implement in the website. I also helped out with planning and storyboarding with how the application should flow and the hierarchy of what comes first, in the middle, and in the last.

Blaizun Diamond - Mostly worked on front-end, specifically the overarching css themes. Created the .css file and imported a custom font. Changed the nav bar and login page, added custom animated title and color scheme. Changed the login/registration forms and submit button to match the color scheme. Improved the checkbox and radio buttons for the settings page and changed the play button. Outside of the code I also set up a lot of the stories on the project board.

Apr 2, 2023 – May 3, 2023

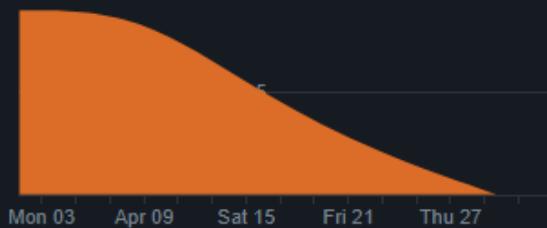
Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts



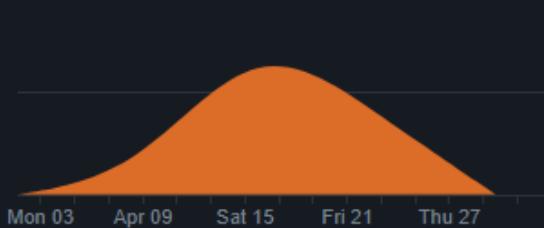
RyanS161

25 commits 6,277 ++ 279 --



blaizun

13 commits 784 ++ 198 --



anbe8868

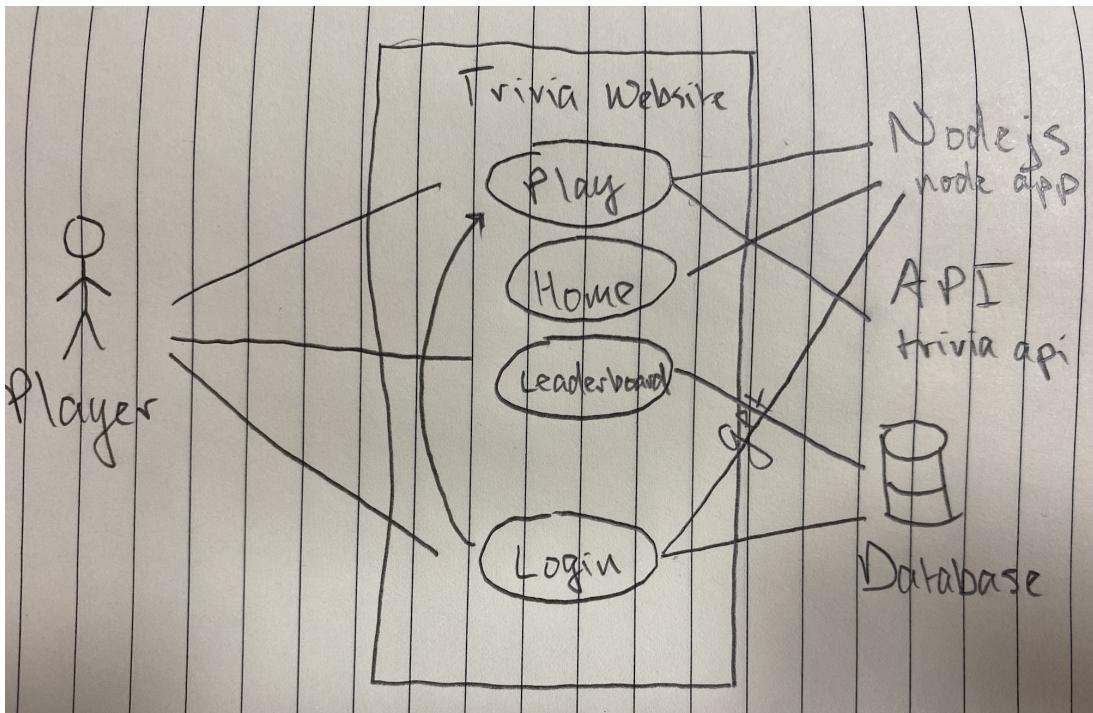
10 commits 92 ++ 66 --



taaz6965

5 commits 29 ++ 5 --





Test Cases:

1. Leaderboard UAT:

- Login with credentials: username: testUser password: password
- Navigate to play page, play short game of trivia
- Navigate to leaderboard
- Navigate to play page, play a longer game of trivia (i.e. get a higher score than the first game).
- Navigate back to leaderboard

Expected Results - User's first score will be reflected on the leaderboard with username and ranking correctly displayed, then the user's second score will correctly replace the first score on the leaderboard and the ranking will be accurately adjusted.

2. Category UAT:

- Login with credentials: username: testUser password: password
- Navigate to play page
- Click category dropdown. Choose 'Science'
- Click the play button

Expected Results - Users will only be given questions whose category is listed as 'Science'.

3. Difficulty UAT:

- Login with credentials: username: testUser password: password
- Navigate to play page.
- Click Question type dropdown. Choose 'Hard'.
- Click play button

Expected results - User will only be given questions who's difficulty is listed as 'Hard'

4. Multiplayer UAT:

- Player 1 login with credentials
 - username: testUser
 - password: password
- Player 2 registers as new user
- Player 1 Navigates to play page
- Player 1 chooses game setting and selects "multiplayer"
- Player 1 creates lobby
- Player 2 joins lobby
- Click play button

Expected results - While the users play the game simultaneously, each subsequent question should be displayed either when 1) both users have answered the question or 2) the time runs out on the question. At the end of the game, the user should receive their score.

Observations:

1. Leaderboard UAT (Failed Case):

- The user logged in, played a short game, and scored 5 points.
- They checked the leaderboard, and their score was displayed correctly with their username and ranking.
- The user then played a longer game, scored 12 points, and checked the leaderboard again.
- The second score did not replace the first score, and their ranking remained unchanged.
- The user's reasoning for their actions was to ensure that the leaderboard accurately updates scores and rankings.
- The behavior was not consistent with the use case.
- The reason for the deviation was a bug in the leaderboard update logic because the endpoint for calculating player scores was pulling the player's first score instead of the most recent. This was an issue with the SQL logic.
- The team fixed the bug, and after retesting, the leaderboard updated scores and rankings correctly.

2. Category UAT:

- The user logged in, navigated to the play page, and selected 'Science' from the category dropdown.
- They played the game and received only 'Science' questions.
- The behavior was consistent with the use case, and no deviations were observed.

3. Difficulty UAT:

- The user logged in, navigated to the play page, and selected 'Hard' from the question type dropdown.
- They played the game and received only 'Hard' questions.
- The behavior was consistent with the use case, and no deviations were observed.

4. Multiplayer UAT (Failed Case):

- Player 1 logged in, and Player 2 registered as a new user.
- Both players navigated to the play page, and Player 1 created a multiplayer lobby.
- Player 2 joined the lobby, and they started the game.
- Both players were able to participate in the game session and answer questions simultaneously.
- However, the subsequent questions were displayed even if both users hadn't answered the question or if the time hadn't run out, causing confusion and inconsistency in the gameplay.
- At the end of the game, the user did not receive their score
- The behavior was not consistent with the use case.
- The reason for the deviation was a bug in the question display logic and score presentation for the multiplayer mode.
- We ended up fixing the bugs, ensuring that questions were displayed only when both users had answered or when the time ran out, and scores were presented at the end of the game. After retesting, the multiplayer mode functioned correctly according to the expected results.

Deployment Link:

<http://recitation-011-team-02-1.eastus.cloudapp.azure.com:3000/>