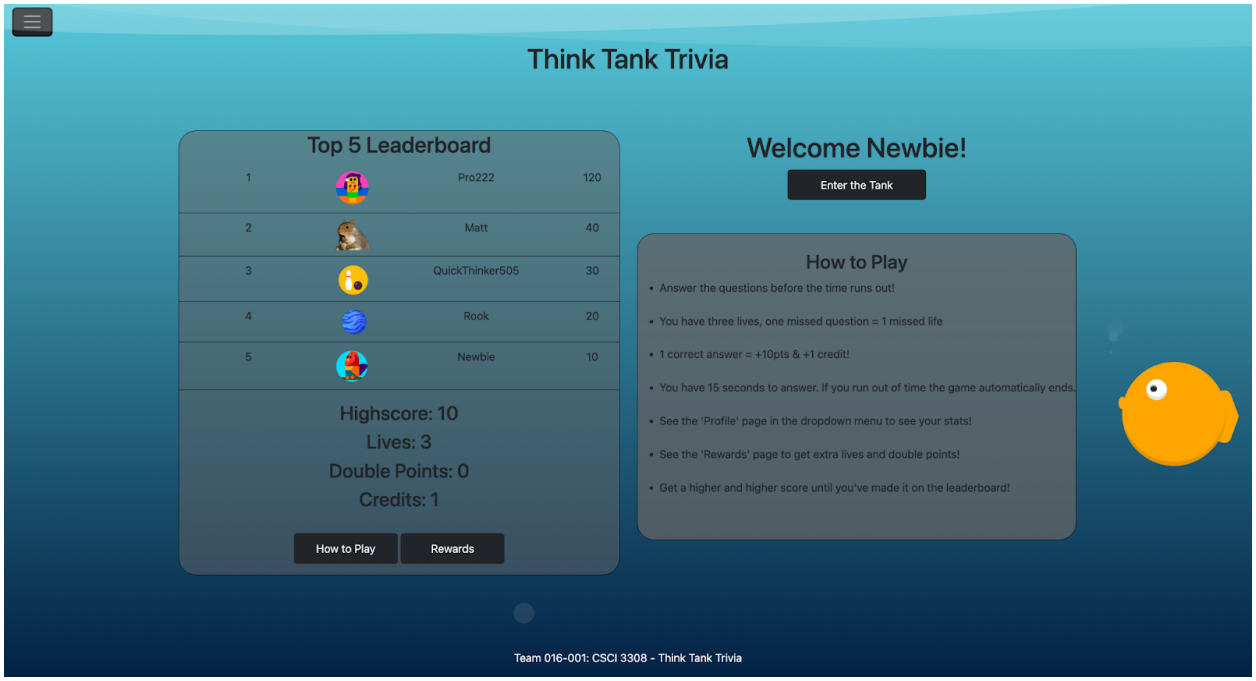
TEAM 001 PROJECT REPORT

Think Tank Trivia



Keaton Glassman, Alex Norby, Quincy Carter, Joao Nicastro Silveira

12/11/23

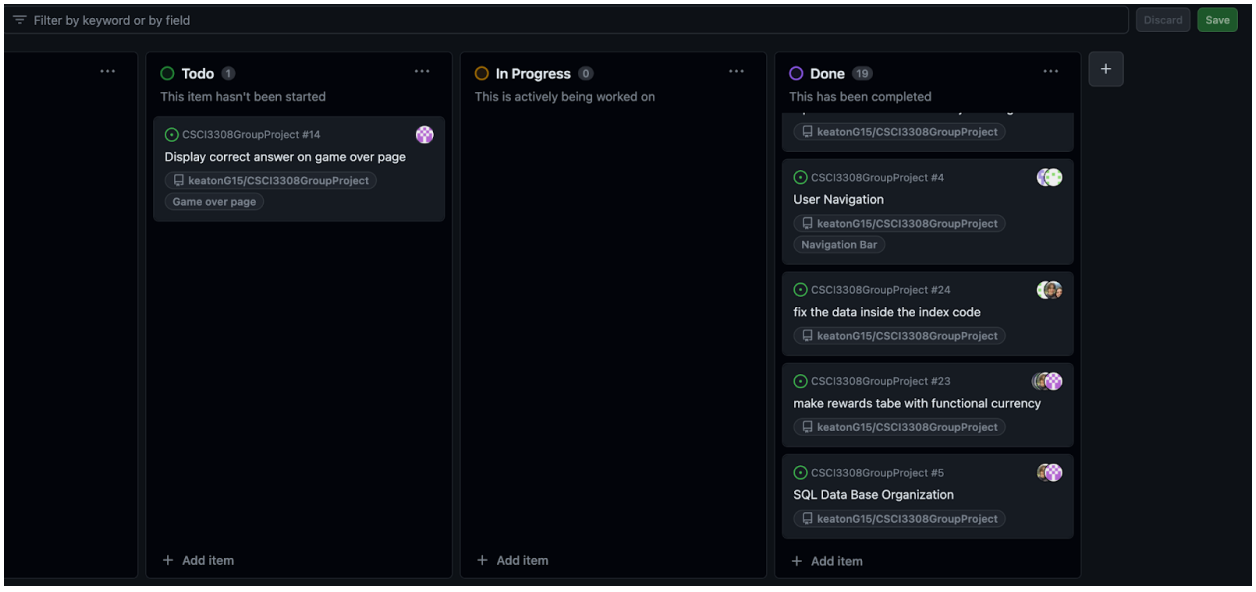
CSCI 3308-016

Our Application Description

Our trivia game, Think Tank Trivia, will provide users with a fun and engaging game that challenges the random knowledge of users! Users will be able to register for their account, upon logging in, they will be greeted with a homepage that displays the option to play a new game, view the history of previously played games, and show the user's unique high score. The user will be prompted with a trivia question and 4 buttons, with one button containing the correct answer. If the user selects the correct answer, their score will increase, if not, the game will end and display the score. If the score exceeds the user's high score, it will update the database and redirect to the homepage. If the user wants to view a history of played games, they can see a list of users who answered x number of questions with a score of x. Based on a high score, we may order this to make a 'leaderboard.' We will build this with basic implementations, and using our agile planning strategy, we will make changes and add functionality as we see fit. What makes this game unique is the plethora of genres and thousands of questions that are stored in the database!

Project Tracker

https://github.com/users/keatonG15/projects/1/views/1



Project Video

https://www.loom.com/share/30596cc42bc6493fa1821af617a87dfc

VCS

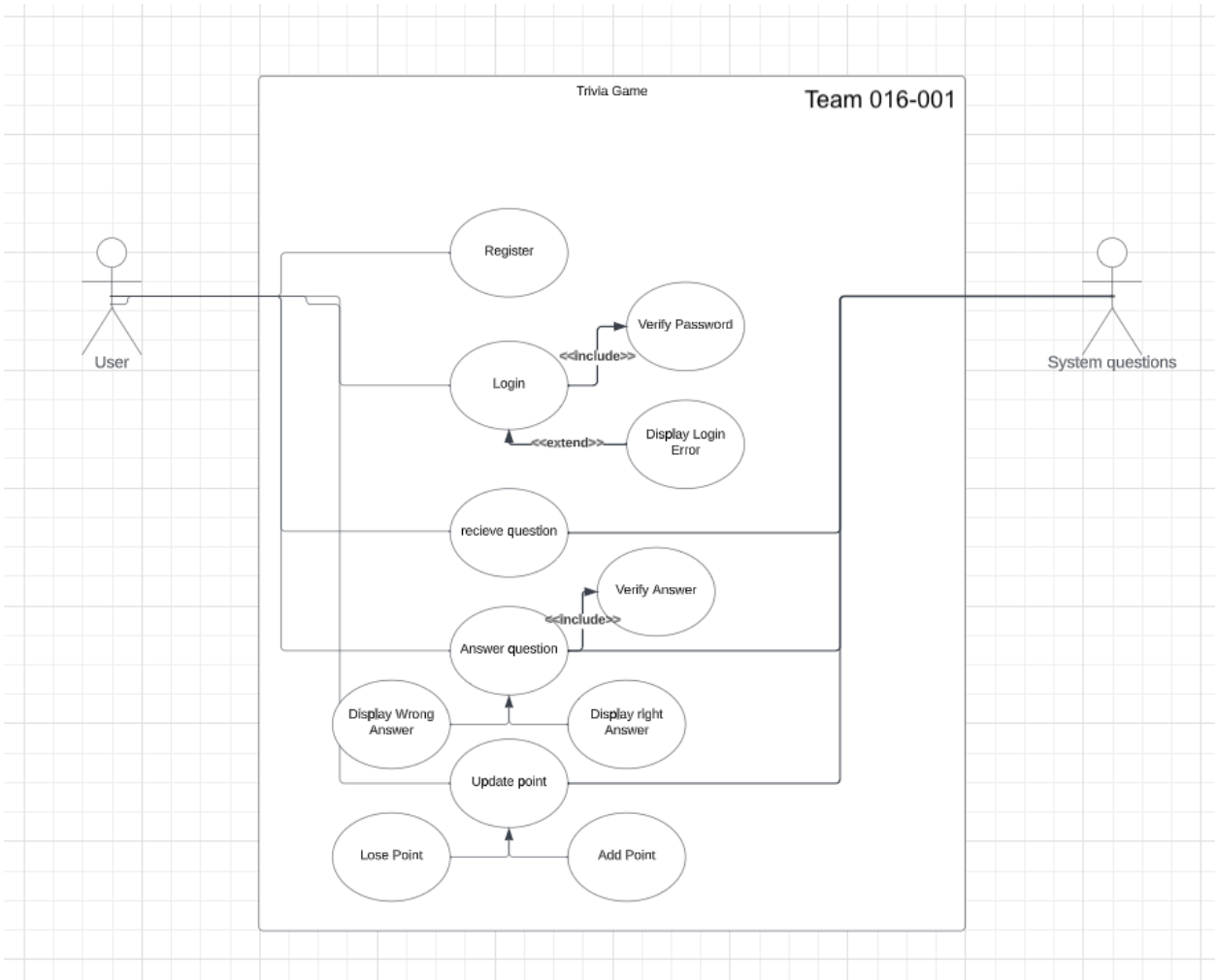
https://github.com/keatonG15/CSCI3308GroupProject

Contributions

Name	Contribution
Keaton	• Built API's such as '/play' which includes an axios call to the Trivia API • Built '/verifyAnswer' that runs the 'behind the scenes' of the game: verifying the answer and rewarding or punishing the user depending on the circumstances. • Built trivia.ejs which connects the user to these API's through a UI. • Worked in every field front end, back end, etc.
Quincy	• Worked on login page • Worked on home page • Worked on register page • Fixed errors when needed • Came up with color theme ideas • Came up with button layout ideas • Tried to come up with currency deletion • Helped with project deployment • Worked on deliverables • Maintained project board updates
Joao	• Created all test cases • Worked on login page • Worked on register page • Worked completely on the rewards tab • Worked on the home page css • Created sql and data base • Creation of currency • Creation of different rewards of game • Creation of timeset in the game • Worked on the index.js

	• Css of the home page • Css of the rewards tab • Helped with multiple api • Css for rewards tab
Alex	• Built entirety of profile page • Made APIs such as /xpSystem and /rankSystem • Created Rank system • Added pictures for background and rank pictures • UAT Test Plan • Updated all release notes when it came time • Helped with CSS and styling • Created the Cloud to host application • Helped Joao with the test cases

Use Case Diagram



Test Results

Test 1:

A USER SHOULD BE ABLE TO LOG IN WITH A VALID USERNAME AND PASSWORD

For testing, we will need to register the following credentials: Username: matt Password: 123 We will be using the mocha environment for testing with a description titled: 'Testing Login APIs'. The ‘it’ is named “Positive Login”. We will send the above credentials to the '/login' API in our test case. When the test case works, we expect the status to have a 302. We also expect to redirect the user to the '/home' page so the user can get started with the game. This test case ensures that registered users will be loaded appropriately and redirected to the home page when they log in with their personal accounts.

Observations

What are the users doing?

The users are attempting to log into our game with the credentials they registered with

What is the user's reasoning for their actions?

To have stats unique to the user, the user must log in with a unique profile that tracks these stats. Users register and want to use this account but cannot do so without a working login page. Without logging in, the user may go directly to the home or play page without a section in the database to track their stats. We handle this later by requiring a valid session in a later test.

Is their behavior consistent with the use case?

A classic log-in test case is often very consistent with user behavior. There isn’t much a user can do to break the login page unless they try to log in with an account that doesn’t exist.

If there is a deviation from the expected actions, what is the reason for that?

Sometimes a user may try to log in with an account that hasn’t been made or try to register an already created account. We have accounted for this case in the next test.

Did you use that to make changes to your application? If so, what changes did you make?

We included safety nets to ensure we don’t have duplicate entries in the database during

registration.

Test 2:

USER SHOULD NOT BE ABLE TO LOGIN WITH INVALID CREDENTIALS

A user cannot log in with an incorrect username and/or password. For testing, we can use any combination of username and password that is not already registered to the database. We will be using the mocha environment for testing with a description titled: 'Testing Login APIs'. The ‘it’ is named: 'Negative Login'. For this example, we sent the following credentials because they were not registered. Username: mattfregahgjnd Password: 123 Since we expect a failed login, we must return a 200 status, and our response is NOT REDIRECT. This is because we throw an error message on the login page to tell users they have the wrong credentials. If the page redirects, this is incorrect because we do not want unauthorized users to enter and start playing the game because this would cause a lot of errors and flaws. Overall, in this test case, we ensure that users do not bypass the login page without a valid session that can accurately track their stats.

Observations

What are the users doing?

The users are attempting to log in with invalid credentials

What is the user's reasoning for their actions?

Some users may think an account has already been created or may have forgotten a password. Other users may be trying to log into another account for malicious purposes.

Is their behavior consistent with the use case?

The user’s behavior is consistent with the use case because users often forget their credentials and try to log in anyway.

If there is a deviation from the expected actions, what is the reason for that?

There isn’t much deviation from the expected action except for accidentally guessing the correct username and password. In this case, we can’t really do much.

Did you use that to make changes to your application? If so, what changes did you make?

The only significant change we made to the application was to notify the user that they inputted an incorrect username or password so they know to try a different credential.

Test 3:

USER SHOULD SUCCESSFULLY RENDER HOME PAGE WHEN LOGGED IN

In this test case, we check that the username and high score match the correct entry in the database when the home page is rendered. Just like we did in test case 1, we are going to need to use the following credentials to send to the API. Username: matt Password: 123 We will be using the mocha environment for testing with a description titled: 'Testing home APIs'. The ‘it’ is named: 'Should successfully render the home page'. Once a user successfully logs in, they will be welcomed by our home page. If the user is successfully redirected, then we expect the server to have a 200 (successful status) This test case's main goal is to ensure that the home page is successfully rendered when logged in. This ensures that the user encounters no errors when trying to play the game after logging in.

Observations

What are the users doing?

Users log into their accounts to reach the home page, which allows them to interact with our project.

What is the user's reasoning for their actions?

Users will want a home page that welcomes them personally and can display the correct high score and other information for them to make the game more enjoyable

Is their behavior consistent with the use case?

The user should consistently be redirected to the home page upon login. It wouldn’t make sense to direct them anywhere else in the application.

If there is a deviation from the expected actions, what is the reason for that?

There shouldn’t be much deviation from the expected actions because users can’t do much more than login and go straight to the home page. One instance could be attempting to jump straight to the actual game upon log in, but can’t do so without going through the homepage first.

Did you use that to make changes to your application? If so, what changes did you make?

The only change we made to our application was to ensure a session is created for the user to access the home page.

Test 4:

USER SHOULD NOT RENDER HOME PAGE WITHOUT AUTHENTICATION

In this test case, we check whether the user can access the home page without establishing proper session/login credentials. In this test case, we call our ‘/logout’ function, which destroys the user’s session, and ensures the user gets redirected to the login page, not the home page.

We will use the mocha environment for testing with a description titled: 'Testing home APIs'. The ‘it’ is named” 'Should not successfully render the home page'. When the test case works, we expect the status to have a 302. This ensures that after logout, the user cannot access the game again without logging in again.

Observations

What are the users doing?

Logging out after completing the game

What is the user's reasoning for their actions?

Users may not want to leave their session open after playing the game, as other users may want to use their accounts on the same computer. Or someone could mess up their account if left unattended.

Is their behavior consistent with the use case?

Users often opt to log out after using our application; sometimes, people may close the tab before logging out, which may affect our project.

If there is a deviation from the expected actions, what is the reason for that?

As mentioned above, users may not always log out of their sessions, maybe this is because closing the browser/tab is easier, or it may be because they forgot to do so. This is likely the test case that expects the most deviation in user actions.

Did you use that to make changes to your application? If so, what changes did you make?

We added a section to index.js that will redirect the user to the login page if a proper session has not been established yet. This ensures that users can't type in ‘localhost:3000/home’ and successfully render the page without a proper authentication

Deployment

Note - for lab 13 we used azure to deploy our site using the link found in lab13host.png under MilestoneSubmissions but then shut it down. Below are the steps for running on localhost.

Use the following steps:

1. Open vs code
2. Open the project file
3. Navigate to the folder called project_code
4. Navigate to the folder called src
5. Open docker
6. If this is your first time, run ‘docker compose down -v’
7. Run ‘docker compose up’ on a vs code terminal
8. If you experience difficulty connecting to the database, comment out src/test/server.spec.js and rerun steps 6 and 7
9. Open Google browser
10. Search in the address bar localhost:3000