Math Whiz-ard

Test Plans

- Describe how at least 3 features within your finished product will be tested.
- The test plans should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.

We should be testing the three features:

- Interactive Math Page
 - Test each page with a preset list of answers that will be a mix of valid and invalid entries. We'll use different ways to indicate negative symbols, as well as entering a solution as the problem itself (ex: 10 + 5 = '10+5' vs 10 + 5 = 15).
- Login/register
 - We will test the login/register pages by making a list of made up emails, names, and passwords. Some of these will be valid and some will be invalid. We will try to login/register with invalid and valid info. Valid Email Checker (potentially).
- Score tabulation
 - We will check to see if the scores are matched or added correctly in the right column values.

User Acceptance Tests: Copy of UAT plans for at least 3 features.

- Interactive math page
 - We will test answers for each page by inputting correct solutions, incorrect solutions, correct solutions, but incorrect signs, different ways to indicate negative symbols, as well as entering a solution as the problem itself (ex: 10 + 5 = '10+5' vs 10 + 5 = 15)
 - We will test that valid answers are inputted as well, such as inputting and making sure there are no letters or symbols
- Login/register
 - Register
 - We will test this by having a user input strings into the email field that don't contain the '@' symbol or a '.', strings that have one of the two, and strings that have both to ensure that valid emails are being inputted
 - We will also test this by having a user input strings in the full name field containing numbers and symbols to make sure that it is a name being inputted and not an email or string with numbers.

- To make sure that the user email is unique we will try inputting the same email twice in a row and calling the database to compare the inputted email with the emails already store in the database

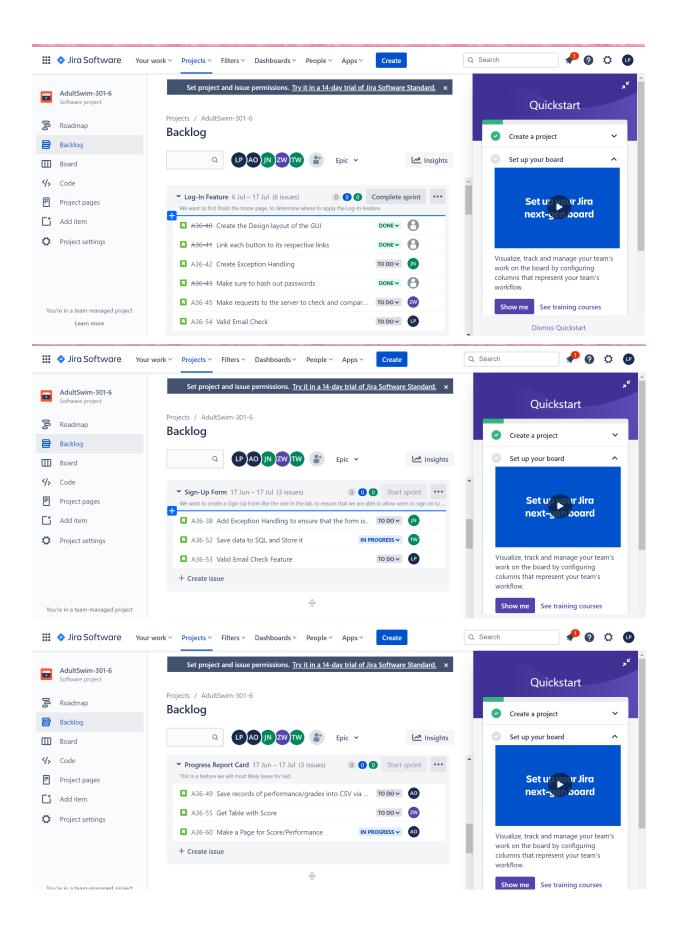
- Login

- We will test this by having a user input strings into the email field that don't contain the '@' symbol or a '.', strings that have one of the two, and strings that have both to ensure that valid emails are being inputted just like in the registration page
- We will check that the page is correctly checking emails and passwords from the database by entering many different combinations of email addresses and passwords with neither correct, both correct, just correct email, just correct password, and more cases like correct email but password has an incorrect password and more like that

Score tabulation

 We will verify that scores are recorded correctly by manually recording scores for a user and verifying that the number correct and incorrect is being properly recorded

Individual Contributions: Screenshot of project management board and contributions from GitHub. (Project Board)



Adult Swim (Group 6) Git Submissions:

- Alex Ojemann
 https://github.com/CSCI-3308-CU-Boulder/3308Summer21_300_6/commit/e4c21_51d8682b653eedaac55f630ab6872e7ff65
- Jon Noranbrock
 No Git Submissions
- Logan Park
 https://github.com/CSCI-3308-CU-Boulder/3308Summer21_300_6/commit/e2443
 1edbd9b1135807d26dfad7fe7c4c5fabd5d
- Tim Wilson https://github.com/CSCI-3308-CU-Boulder/3308Summer21_300_6/commit/1791b 1891eae9429dcf5651e544d42f8ef6885a4

Zach Wrubel

https://github.com/CSCI-3308-CU-Boulder/3308Summer21 300 6/commit/f7647 1afd6421b467718682bdc82cd515e2dc5f2