1. Create 1 document per team, within the milestones folder in the project directory, that describes how, at least, 3 features within your finished product will be tested.
2. 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.
3. The test plans should include a description of the test data that will be used to test the feature.
4. The test plans should include a description of the test environment that will be used to test the feature.
5. The test plans should include a description of the test results that will be used to test the feature.
6. The test plan should include information about the user acceptance testers.

Logout Test
Ensure that the user is able to logout after successfully being logged in.

Description
Test to ensure that the user is successfully logged out of their account already logged in to.

The test environment will take place in the '/logout' environment.

Test from Developer Perpsective

Positive:

First, check that the session exists with a user logged in from the '/login' page. Then, check that the session expires once on the '/logout' page.

If all are valid, print 'Success'

Negative:

First, check that the session does not exist with a user not logged in from the '/login' page. Then, check that the session does not expire once on the '/logout' page.

If all are valid, print 'Invalid Logout'

User perspective

IF user goes to '/logout' page without previously logging in, show an error stating 'not logged in'.

IF user goes to '/logout' page are the session does not successfully terminate, show and error stating 'invalid logout'.

IF user goes to '/logout' page and the session exists and is successfully terminated, show the success statement 'successfully logged out'.

Budget Tab Test

Description

Ensure that the Budget tab functionality is up to user requirements and expectations, by testing the ability to input and track budget related information.

Test Environment

Test will take place in the ‘/budget’ environment

** Test from Developer Perspective**

Positive- Check that the user entered a reasonable positive budget amount, added an income source with valid positive amount, added expense with valid positive amount. Check that the budget was saved. If so, print ‘correct’

Negative- Check that the user entered a valid budget amount, added an income source with a invalid negative amount, adds a expense with an invalid negative amount. If so, print ‘invalid input’

User perspective

IF user attempts to save a income with negative amount, show an error stating, ‘Invalid budget amount, please enter a positive value’

IF user enters a non-numeric value, show an error stating, ‘Invalid input. Please enter numeric value’.

IF user tries to save a budget with incomplete information, show an error stating, “INvalid Input. Please complete the required field.”

IF user creates a reasonable positive budget, adds positive income source, adds positive expenses, saves the budget, then a message is shown displaying “Budget created successfully”

Register Test

Ensure the user is able to register a new account using a unique string username and unique email with a valid password.

Description

Test user inputs string username, email, and string password, ensuring all three are valid.

The test environment will take place in the ‘/register’ environment.

Test from Developer Perspective

Positive:

Check that the username inserted does not exist in the users database, check that the email inserted does not exist in the users database. Check to ensure that username, email, and password are all strings.

If all are correct, print ‘Success’

Negative:

Check that the username inserted already exists in the users database, check that the email inserted already exists in the users database. Check to ensure that the username, email, or password are not strings.

If ANY of the above match cases, print ‘Invalid Input’

User perspective

IF user enters a username that already exists in the users database, show an error stating ‘username already exists’.

IF user enters an email that already exists in the users database, show an error stating ‘email already exists’.

IF user does not enter a username or email or password, show an error stating ‘invalid input’.

IF user inputs a unique username, unique email, and a valid password, show success stating ‘successfully registered’ and reroute user to ‘/budget’ page.