Feature Tests (Same as PM5)

- Login to website
    - We will test to make sure that a user with a previously created account can log-in, and that their session persists (i.e. keeps them logged in) until they close out of the website.
    - Test Cases
        - On entering valid credentials (either username and password or email and password), the user should be redirected to the home page (POSITIVE).
            - Tested by inserting existing accounts during initialization
            - Tested by generating new accounts, logging in with them.
        - If a user enters invalid credentials, reload the log-in page with an error message (NEGATIVE).
            - Tested with pre-existing accounts, generated though initialization, incorrect information was used for username, password, log in through email, incorrectly imputed in 1 or both fields, and tested individually for each instance across multiple accounts.
            - Tested with newly created accounts, same testing methodology as for existing accounts.
        - Upon logging in, a cookie is properly created and read by website to make sure the user stays logged in and the navbar is correct for a logged in user sitewide
            - Tested during user log in, navbar/certain features on the website could only be accessed if the user session id was valid and recognized by the website. Otherwise those features/parts of the site were not accessible.
        - On clicking the “Logout” button while logged in, the cookie for the user session should be deleted and the navbar should display the correct options for a logged-out user.
            - Tested during user log in, navbar would return to default options.
- Register for an account
    - We will test to make sure that it will not continue without all fields being filled out, and that an email address is there, and that an adequate password is present. We will make sure that an account was not created when these fields are not filled out correctly!
    - Test Cases:
        - Make sure email address is present
            - Tested by entering invalid emails, prompt after/during submission (two levels) asking for valid email address
        - Try to submit account with an inadequate password
            - Tested by entering passwords that either did not meet requirements or included banned works/characters
        - Try to submit account without a name

      - Tested by attempting to create an account without a name, could not bypass submission
    - Try to submit an account with all fields filled out; and when fields are filled out correctly, the user is directed to the home page and is logged in with the account that they just created.
      - Tested during log in phase, validated by having console.log read out existing accounts in our database to ensure data was properly received and stored for use.
    - Check to make sure that you cannot register for an account while you are logged in to a separate account, and that you have to log out of the account that you are logged in with before registering a new account.
      - Removed register option in nav bar during log in, checked to make sure that user could not access registration while logged in.
    - Check to make sure that a user cannot create a duplicate account!
      - Attempted to create either two accounts one after another that were the same, or create one account against a pre-inserted existing account.
- Creating a post
  - We will test to make sure that a post can successfully be created and that all fields are filled correctly. An image can be uploaded, text is limited to a specific amount of characters (500), usertag should link to user profile.
  - Test Cases
    - Unit: Make sure post inputs (Post title, post text) are received correctly and are limited to 100 and 500 characters respectively (subject to change) Empty submits are rejected and all fields must be required to post.
    - Unit: Make sure photos can be input and received by the website to be used when posts are loaded and displayed
    - Integration: User Tag on post should be able to allow someone viewing a post to travel to the respective post creator's profile page.
    - User Acceptance: Check to see that a post can be created and viewed in the feed after it is created.
      - ALL ABOVE TESTING WERE DONE SIMULTANEOUSLY TO ENSURE THAT THE CREATE AND VIEW POST PROCEDURE WORKED AS ONE ENTITY
      - Tested by uploading existing info into the database, load the website to see if posts would load correctly and check for formatting errors. Variations of text length, picture size, etc. were used to determine what would be unusable for a post, restrictions applied accordingly.
      - Tested by creating multiple posts on different accounts and verifying that these accounts were correctly connected to each post and could view others and their posts through the links in each post seen as the post writer's username.

- User Profile Page
    - We will need to test the ability to update user profile information such as pictures and additional catches with image uploads and field inputs. User posts should also be seen and updated. Buddies system was also tested as part of user profile page.
    - Test Cases:
        - User should be able to update profile picture, which can be seen by other users as well when they visit a user profile page.
            - Multiple existing and/or created accounts had images uploaded to be their profile picture. Each account was checked to see if the image had been uploaded correctly, and subsequent users could see that same image attached to the particular user.
        - User should be able to create a catches post on their profile, update the information in the fields, and see their catches on their user profile website.
            - Multiple existing and/or created accounts were tested to create and fill out a catches post that was verified to be displaying correct information and could be seen on the user profile page of that user as well as being seen by other users.
                - NOTE: There are no restrictions on what users can put in the required fields besides data type
        - User should be able to see recent posts that they made in the user profile page
            - Posts tied to a user account created during create post/feed testing were noted and user profile page was viewed by user and other users to verify that these posts could be seen.
            - Posts were created by user and were verified to have been updated in the user profile page.
        - User should be able to see new friends in user profile page when a friend is added or user should no longer see a friend when user removes a friend.
            - Done in conjunction with search functionality (refer to testing on search feature)
- Home Page/Search Functionality
    - We need to test home page loading, ability to navigate from home page, search for a user feature, and buddy system.
    - Test Cases:
        - Home page should be the first page to automatically load with no user logged in.
            - Verified by making sure the page was loaded correctly, no user logged in immediately.
        - Home page could successfully access all parts of the website during log in and logged out

  - Became comprehensive testing for flow of website in general. User tracks were determined (ie home page to registration to login to feed to create post, etc.). Validated with specifics from all other tests.
- Searching for a user (Buddy System)
  - Mix of creating users and inserting pre-existing users, usernames were searched (correctly and incorrectly) and users were discoverable or not found.
  - Users who were found were friends, and this would be updated on the user profile page. The same user would then be unfriended, and user profile page was checked to see if status had updated correctly and user was gone.