

Team Members:

- Lakshya Jaishankar
- Sarah Zendle
- Matthew Teta
- Samuel Mast
- Rebecca Coryell
- Charlie Koepke
- Brian Mayers

Project Title: envi

User Acceptance Tests:

1) Signup Page (* I'm not sure what functions are doing the checks, so I just put [function] instead)

Test Cases: when a user uses an invalid username, when a user uses a valid username, when the user uses an invalid password/doesn't meet the criteria, when the username a user chooses is already taken, when the username is not chosen already, if one of the required fields is not filled, if they have not checked off the terms and conditions agreement, if the passwords do not match

- a) When a user uses an invalid username (has an invalid character) - the [function] will check to make sure that all of the characters used in the name are valid characters
- b) When a user uses a valid username - when they click the "Submit" button, the username and information will populate in the database, so no one else can have the same username
- c) When the user uses an invalid password - the [function] will check to see if all the characters in the password are valid or not. If they are not then the user will not be able to use that password for their account. Additionally, if the password does not meet all the criteria, then the user won't be able to use that password for their account
- d) When the username a user chooses is already taken - when the user tries to complete the sign-up process, the [function] will check if the username is already in the database, and if it is, the user will have to choose a different username
- e) If one of the required fields is not filled - if they try to click the submit button the "required" tag in the form will not allow the user to submit the form without filling out all of the required fields

- f) If they have not checked off the Terms and Conditions agreement - if the user tries to finish the sign-up process but has not checked of the checkbox that indicates they read the Terms and Conditions (no one does though), then they will be notified that they didn't check it off
- g) If the passwords do not match - the [function] will check and see if the passwords match, and if they don't then the user cannot complete the sign-up process

2) Login Page

Test Cases: when a user's login information is correct, if the user's username is incorrect, if the user's password is incorrect, if the user has not signed up yet and tries to use the login page

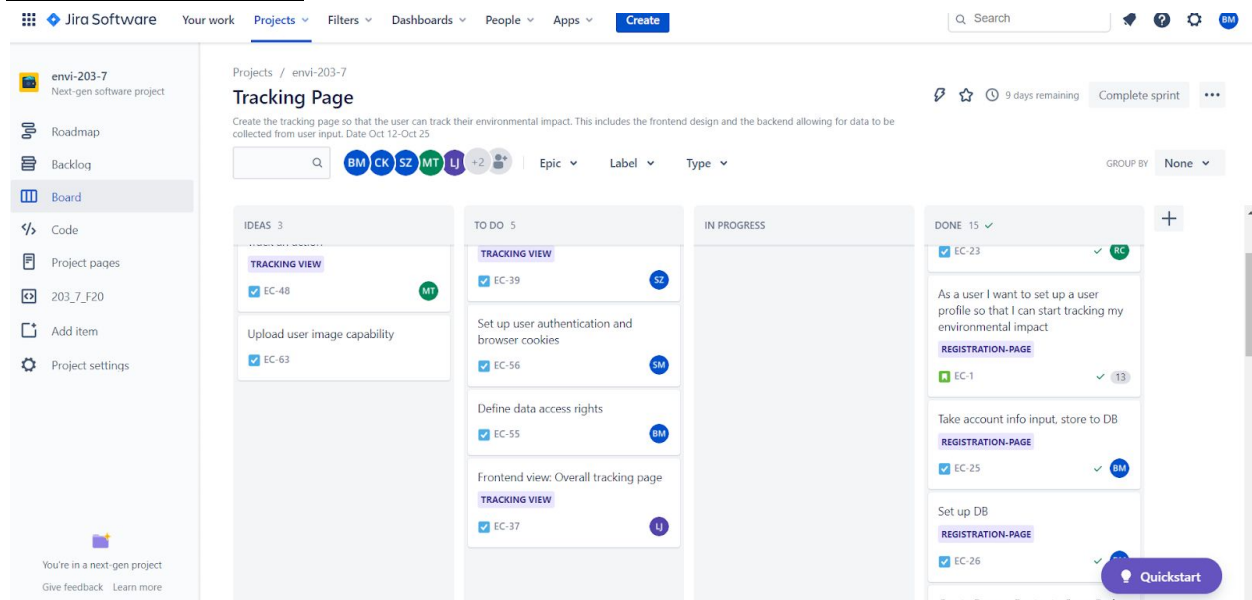
- a) If the user has an invalid username, the username inputted will be checked against the username stored in the database. The user will have to try different usernames until every character matches the characters in the database.
- b) If the user has an invalid password, the password inputted will be checked against the password stored in the database. The user will have to try different passwords until every character matches the characters in the database.
- c) If the user has not signed up yet, it will check in the database to see if the username and password exists. If the credentials do not exist, then the user will be prompted to create an account on the signup page.

3) User Profile Page

Test Cases: when the user wants to change their name, when the user wants to change their username, if the user wants to change their password, if the username they chose is taken, if they want to update their profile picture

- a) If the user wants to change their username to a username that is already taken, they should be unable to submit the form and an error should be displayed.
- b) If the user wants to change their username to a username that is not already taken, the username should be updated in the database so that login verification works with the new username.
- c) If the user wants to change their password, the new password will not submit to the database until the old password is entered correctly and the new password matches criteria (length, contains a number, etc.)
- d) If the user changes any other information, the data should be sent to the database so that the new information is loaded when the user logs in again.

Jira Board Screenshot:



Individual Contributions:

Brian Mayers- Worked on back end routing for the donate page. Updated database to get rid of unnecessary components.

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/0a77dd32f0af50b3f867f186ee5c2689d51f3b44

Charlie Koepke - finished up the tracking page. Making good progress on the search bar function in the friends page.

Github:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/7b73e8ae6be8d94619a2c87865f311530a9e08b6

Samuel Mast: Updated the database to bet fit current needs. Worked on reconfiguring node files to account for changes to repository when deploying the application.

Github:https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/f23125e1cd694aa4db93fe67942f9bdc25a94c8b

Lakshya Jaishankar - Got display function and delete function to work on tracking page so that specific tasks can be created and deleted. Connected frontend to back for updating profile page information

Github Commit:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/7a8f8c796119cb1fe1f1b9916167d0e5b380c711#diff-3b6302c4f3031efb433ecc543401c4c491e504510b41c31caad8983787dc4dcbb

Sarah Zendle - Added new stylized sidenav and navbar to all pages. Updated friends page to match standardized style. Added functionality to load friends list and activity feed from sample user data rather than static html. Started work on navbar login/logout popup modal that loads dynamically based on whether the user is logged in.

Github Commit:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/8051fdca3d512bb4387a486e88897ba7c0cdd2c8

Rebecca Coryell - Worked on routing for the user profile and friends page, including routes to get the list of friends for the user, to add a friend in the user's friend list and to update user info in the database.

Github Commit:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/273e9e0e42f2c7faea7dc4ef5b80dc7eb50f6796

Matthew Teta - Separated the frontend pages into folders, and reorganized assets. Moved frontend pages into "public" folder and configured express to serve them on the same server as the (renamed) /api/ routes statically. Solved the bug with the authentication not remembering user sessions and integrated auth with frontend register/login pages.

Github Commit:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/d0b53dec7c15d5012f6041ef239492c263834837