

To Do (Sprint Backlog)	In Progress	Testing	Done
			Task 1: Create card components for listings (Eric) [1]
			Task 2: Given a certain set of search criteria, return the listings that meet the criteria with the SQLAlchemy query (Francisco) [2]
			Task 3: Create form for making new posts [3]
			Task 4: When a card is clicked, you can send a friend request to the poster(Francisco) [4]
			Task 5: Sort listings received from backend (Aditya) [5]
			Task 13: Search by content or tag name (Eric) [6]
			Task 6 Create friends component skeleton and add redirect for side navbar (John, Eric) [7]
			Task 7 Create endpoint fetch for frontend and query for backend to retrieve all friends for a user (John) [8]
			Task 8 Create button to add another user as a friend, there should be a fetch from frontend and backend should add the desired user to the friends list (John, Aditya) [9]
			Task 9 Create button to remove a user as a friend, there should be a fetch from frontend and backend should remove the desired user from friends list (John, Aditya) [10]
			Task 10 Create a search bar that allows a user to search through all their friends (John) [11]
			Task 11: Create search endpoint, decide how to structure endpoint (Aditya) [12]
			Task 13: Construct the SQLAlchemy queries for each of the possible search parameters (Francisco) [13]
			Task 12: Filter search results in the frontend (Francisco) [14]
			Task 14: Sort search results in the frontend (Eric) [15]
			Task 15: Construct the SQLAlchemy queries for each of the possible search parameters (Francisco) [16]
			Task 16: Display profile picture in my profile (John, Aditya) [17]
			Task 17: Be able to see other people profile pictures (John)
			Task 18: Use firebase to , store image, generate streamable link ,store link in db (Aditya) [18]
			Task 19: Allow flask to accept files in post request (Aditya) [19]
			Task 20: Upload profile image in the frontend (Eric) [20]

[1] Acceptance Criteria:

Tester: John

1. The user should be able to see all the card components in the Feed section
2. The user should be able to see the contents inside the card components

[2] Tester:

Acceptance Criteria:

- 1.- Queries should return the correct data

[3] Tester: John

Acceptance Criteria:

1. The user is able to click on a listings and route them to a different page

[4] Acceptance Criteria:

Tester: Eric, Aditya

When a user views a listing in depth, there should be other similar listings suggested in a carousel format. If there are no similar listing to the one posted, then there should be a meaningful message that shows "no listings available"

[5] Tester: John

Acceptance Criteria:

1. The user should have the option to sort the listings once they are received
2. The user can select how he/she can sort the listings

[6] Tester: John, Aditya

Acceptance Criteria:

1. The user should see a select input field to select how to search by
2. The user should be able to select an attribute to search by via the select input field

[7] Acceptance Criteria:

Tester: Eric

When a user clicks on the side navbar for friends, they should be redirected to the friends component

[8] Acceptance Criteria:

Tester: Eric, Aditya

Be able to render friends component properly and display all the user friends in a list or display an empty message if the user has no friends

[9] Acceptance Criteria:

Tester: Eric, Aditya

If an invalid user is attempted to be added, then a toast should be displayed on the frontend. When a valid username is entered into the friend submit, then the frontend should display a success toast. When the friends component is loaded again, the new user added should be in the list now.

[10] Acceptance Criteria:

Tester: Eric, Aditya

Acceptance Criteria:

1. The user should have the option to remove a friend
2. The user should not see the person on his/her friend's list once he/she has been removed

≈

[11] Acceptance Criteria:

Tester: Eric

When a user searches a user, it should return all the users that match the value entered. If there are no users that match the value passed in the search, there should be a message that says "no match"

[12] Acceptance Criteria

1. The endpoint structure should be easy to use and elegant. The same endpoint should be usable for multiple different types of queries

[13] Acceptance Criteria

1. For each possible combinations of queries, construct the queries
2. The queries should return the listings that correspond with the search criteria

[14] Tester: John

Acceptance Criteria:

1. The user can click on a button that will open a filter menu
2. The user can select different options to filter by

[15] Tester: John, Francisco

Acceptance Criteria:

1. The user should be able to see an option to sort search results
2. The user should be able to see the listings sorted, depending on what he/she selected

[16] Acceptance Criteria

1. For each possible combinations of queries, construct the queries
2. The queries should return the listings that correspond with the search criteria

[17] Tester: John, Aditya, Francisco

Acceptance Criteria:

1. The user should be able to view other people's profile pictures
2. The user should be able to view other people's profile pictures in a larger modal when clicked

[18] Acceptance Criteria

1. The firebase api generates the streamable link and is stored in the database
2. When the link is clicked, the image should be accessed.

[19] Acceptance Criteria

1. The files are passed in the post request properly
2. the flask app can access the files and manipulate them

[20] Acceptance Criteria:

Tester: Eric

When a user clicks upload photo, a file upload screen should pop up to select a file to upload. When the user logs in, the profile picture should be properly displayed in the profile. When the profile picture is edited, the edited picture should display on re log in.