**#5 Gipher - Gif For All**

**## Milestone 1**

- Create REST API to fetch data from https://developers.giphy.com/docs/

- Create a static view. This view should have an angular route path as

/my-recommendations.

- It should contain two sections:

- Search section

- Recommendation section

- Create a search bar with a search button, where user should be able to

search based on gif or sticker.

- Search section will have a text field with ID #search-text-field

- A button to submit search text field content with ID #search-button

- On submission of search text should query the results from giphy.com

and display - search results containing GIF images which should be

shown as card like components. If no results are displays display a “No result found” message.

- Get the results displayed in the search results section.

- Give an id #search-results to the search result section.

- Search results should show a series of card like components and a

card should have the following attributes.

- Assign .gipher-card class to each GIF image card and all the GIF

image cards displayed in all sections should have the below details

with attributes.

- assign .gipher-image class to show image.

- toggle .recommend & .unrecommend classes to recommend and unrecommend

buttons.

- Create a recommend button attached to every GIF card. Give recommend

button a class .recommend

- Click on the Recommend button and the button should change to

Unrecommend.

- The Unrecommend button should have a class .unrecommend and Recommend

button should have class .recommend.

- View recommended GIFs under my-recommendations section

- Display all recommended GIFs in this section. Recommended GIFs should

be displayed under my-recommendations section.

- Unrecommend button, the button should change to Recommend again, the

GIF should disappear from my-recommendations section.

**## Milestone 2**

- Create a Dashboard view (Angular Route /dashboard) with three

sections Display Favorite GIFs, recommendations for a GIF from

giphy.com one under the other.

- This Dashboard is the default view to be shown.

- The 3 sections are:

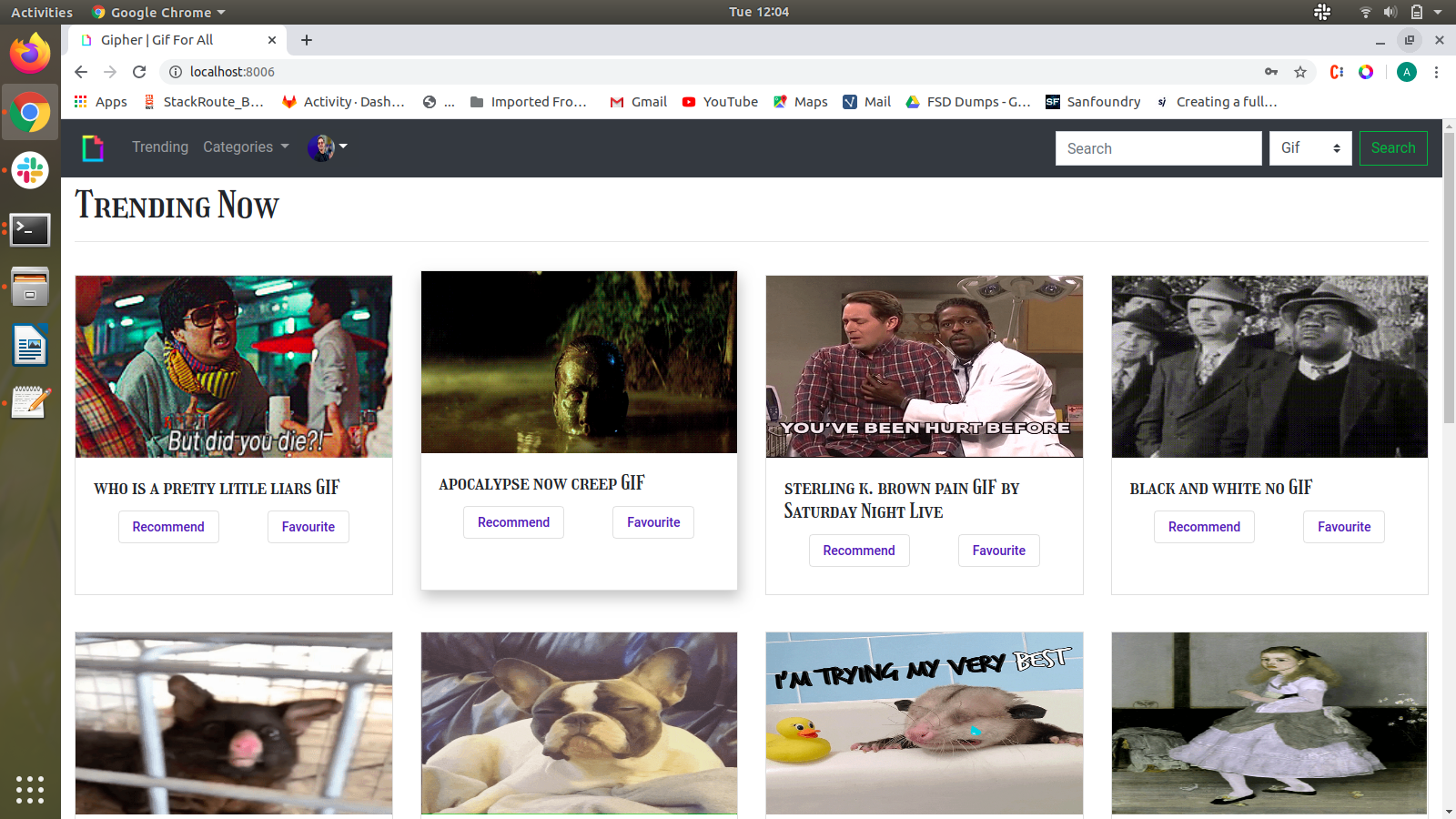
- Favorite with Id #Favorite.

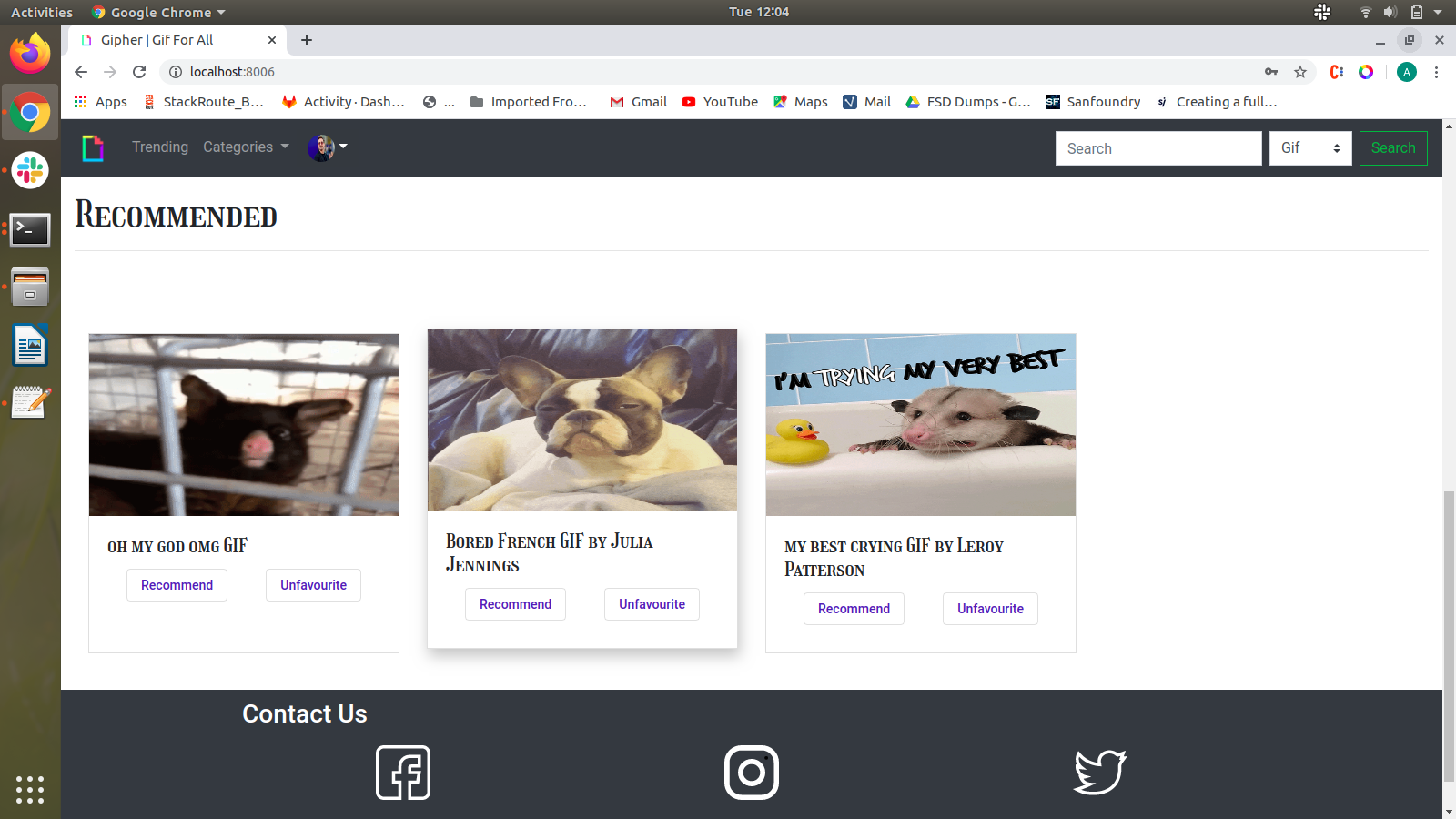
- Recommended GIFs with ID #recommended

- View all Favorite GIFs under Favorite section

- View all GIF recommendations from 3rd party service provider

(giphy.com, in this case) under the recommendations section. Also show the count of the number of users who have recommended a particular Gif.

Illustration 1: Default Page

Illustration 2: Default page includes Recommended Section

**## Milestone 3**

- Create a Card Details Component (Angular Route /card-details) which

should display the details of the selected card and also the comments section

of the selected card.

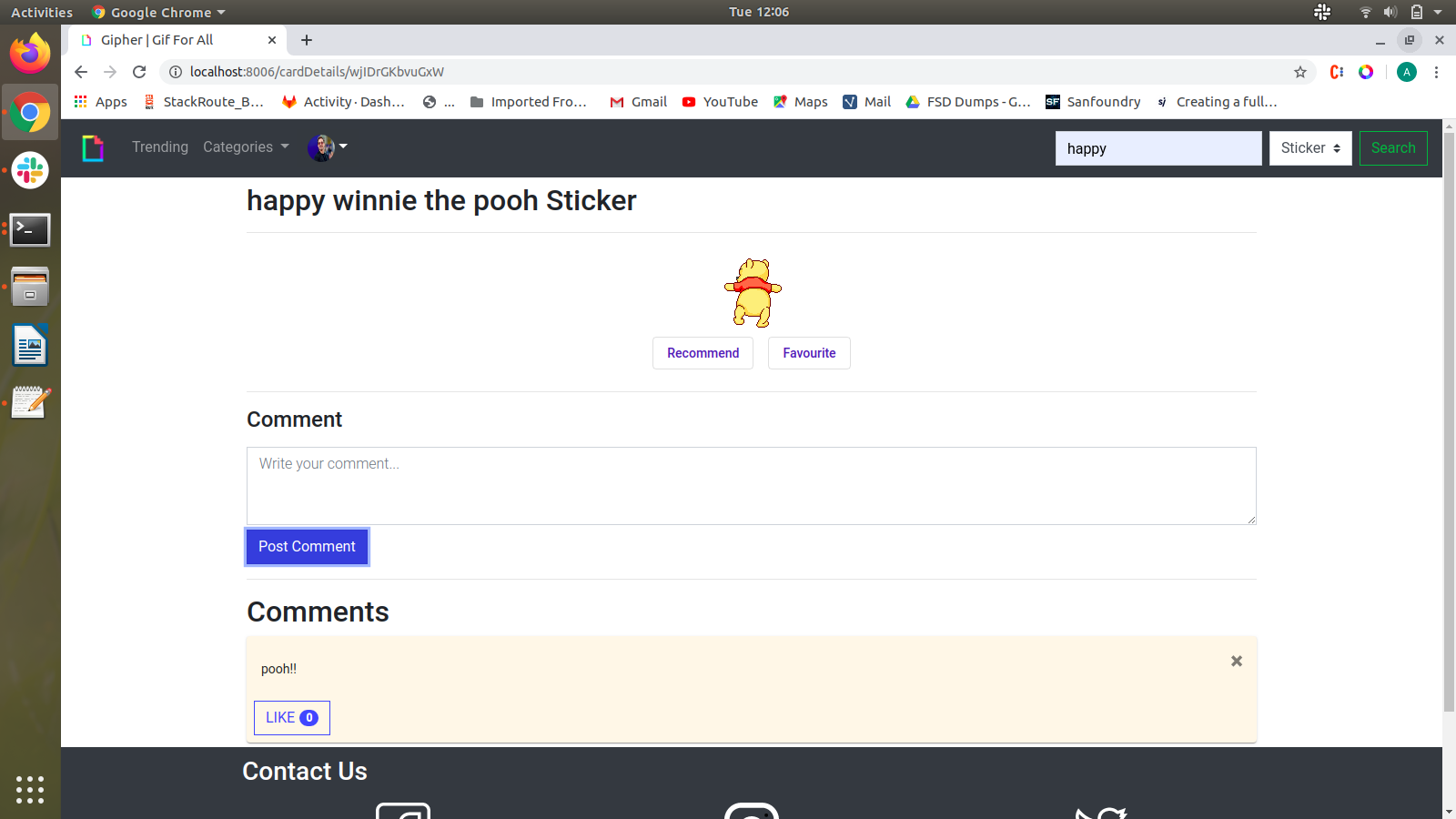
- If a user clicks on a card, the card-details component should open.

- User should be able to comment on the card only if he/she is logged in.

Else, should have a link to login page.

- Comments should appear according to the number of likes for that particular comment.

- User should also be able to delete his/her own comment only.

  
Illustration 3: Gif Details with comments

**## Milestone 4**

- Create a Profile Component (Angular Route /pofile) where the user can

view his/her profile and also should be able to edit his/her profile.

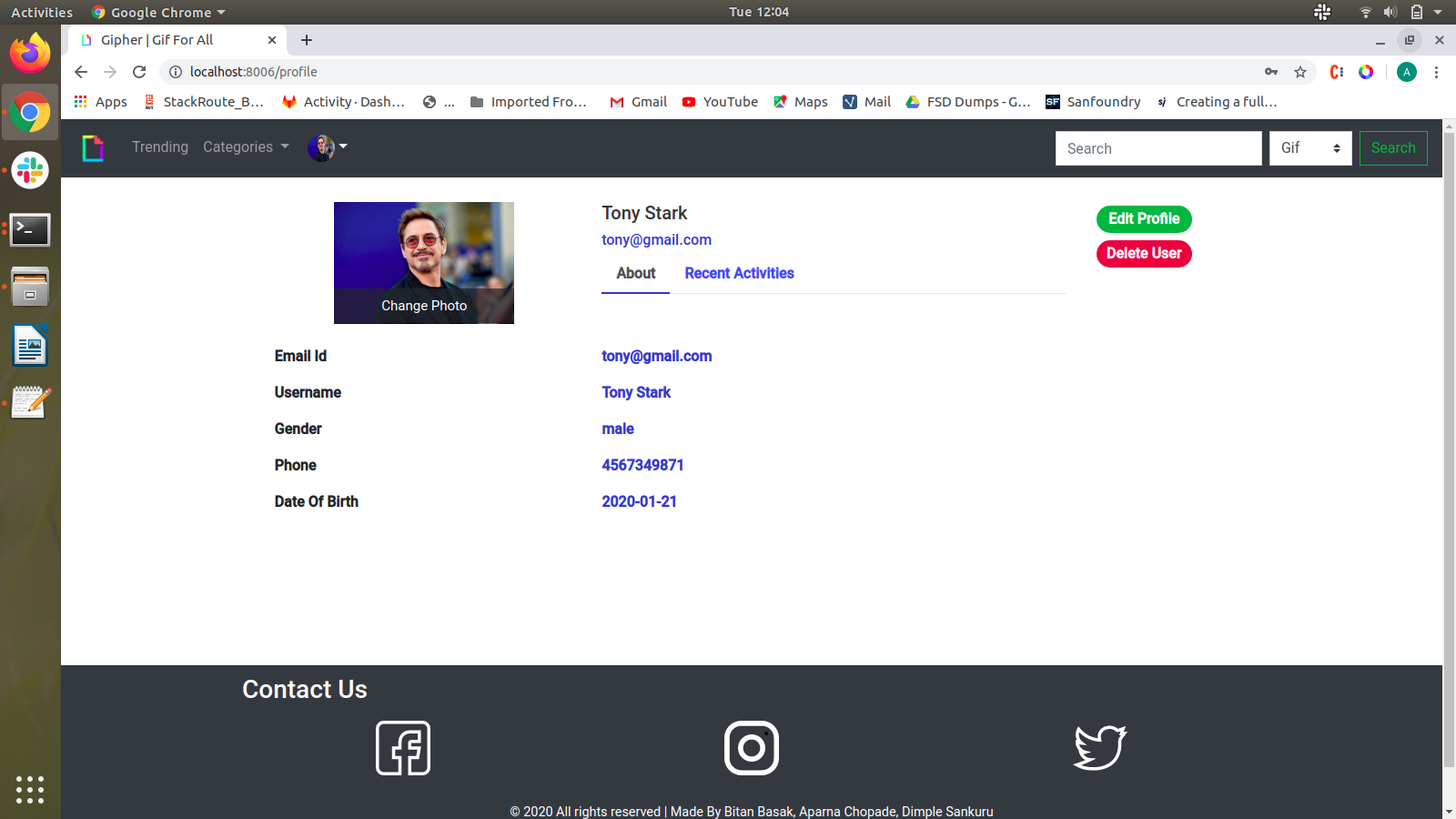
- The logged in user's header component should have a drop-down having favourites,

profile and logout.

- If user clicks on profile, he/she should be able to view his/her profile.

- And the pofile page should also contain edit profile button, which on clicked, user

should be able to edit his/her profile.

  
Illustration 4: Profile Page

**## Milestone 5 - Functional Requirements**

**### Pagination**

- In case of multiple records, the page should have pagination option

to display data in the card layout

**### 404 Not found Page**

- Display a 404 Not Found page is the user visits any random route that does not exist for your website.

**### Implement Authentication**

- Users should not be able to add anything in the favorites &

recommendation list until logged in.

- Create Login Page

- Create Register Page (User’s email address should be a userid)

- Create Edit Profile / Change Password page (Email address cannot be

changed)

- Upload profile image while register & displaying the same in the

toolbar after login

- Encrypt password using bcrypt while storing in the database during

registration

- Registered users should be stored in the MySQL database.

**## Milestone 6**

- Implement Test Automation – write Unit Tests for Backend and

FrontEnd.

- Add E2E Test Cases, Dockerize and Implement CI using Gitlab Runner.

- Create the README.md file with Steps to Execute.

## Acceptance Criteria

The final evaluation of submissions if the following is achieved:

- Meeting the Core functional requirements as stated

- Write Test Cases with each functionality with all the features

- Follow the MVC model

- Evaluation Rubrics (On a Rating from 1 to 10 – wherein 1 is NonExistent to 10 being Exceeded Expectations)

1. Functional Requirements

2. Non-Functional Requirements

3. Code Quality

4. Standards, Styles, and Guidelines

5. Aesthetics and Accessibility