



Tweeter

Instructions

- Read the entire question carefully for at least 15 mins, understand it and then code it.
- Don't jump directly into the code.
- Commit your code every hour with a proper commit message to your repository (we will monitor every commit)
- Use **React** to solve this question.
- Use **Redux** for state management.
- Use deployed **JSON-SERVER** to manage your API and data, using localStorage will not be considered.
- Use **Chakra-UI** for styled-components.

Problem Statement

- The task is to build a minimal social networking app like Twitter where users can sign up, create, and manage their posts by editing or deleting them.
- The app should have a Navbar with the following pages.
 - Sign up page
 - Sign in page
 - User Profile (This should be a private route, only users who signed in can access this page)
 - Timeline (This should be a private route, only users who signed in can access this page)

Signup Page

- The user should be prompted by a signup form where the user can register with the following data
- Username (Should be unique)
- Avatar URL
 - You can use this API to generate dummy avatars

<https://pravatar.cc/>

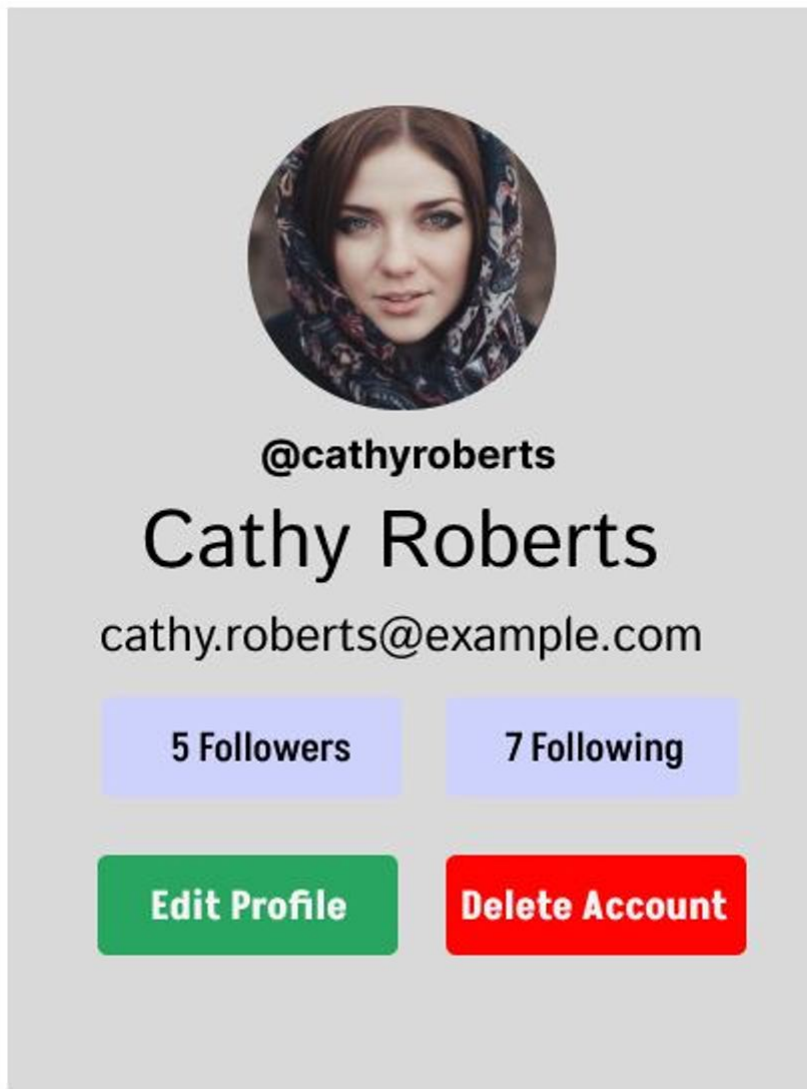
- Full Name
 - Email
 - Password
 - Signup Button
 - Once the user submits the input data, store that data in the JSON server on route `"/user"`.
 - Once they are successfully signed up, the user should be redirected to the sign-in page.
-

Sign-in Page

- This user should be prompted by a login form, where the user will enter the following credentials
 - Username
 - Email
 - Password
 - When a user fills in credentials, match the entered data with the JSON server data you stored while signing up. Display an alert - if login is a success, show a message in the alert box as `login successful`, else show a message as `login failed`
 - After signing in successfully, the user should be redirected to the User Profile page
-

User Profile

- The User profile should show the following details of the user who is logged in.
 - Avatar
 - Username
 - Fullname
 - Email
 - Followers Count (Can be randomly generated)
 - Following Count (Can be randomly generated)
- There should be an **"Edit Profile"** button clicking on it, open up a modal where the user should be able to change their Fullname, Email, and Password.
- There should be a **"Delete Account"** button clicking on it, the user should be able to delete their account and should be redirected to the signup page.
- Note: Edit and Delete Account should be reflected in the JSON-server as well.



User Profile Reference Image

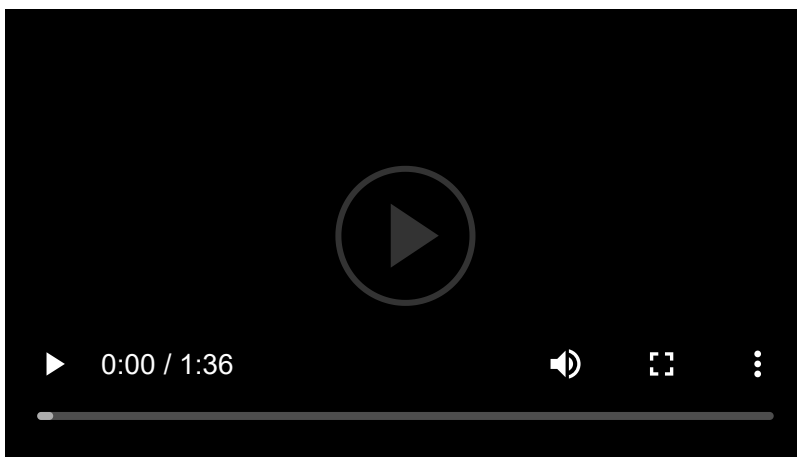
Timeline Page

- The timeline page is divided into two sections
 - The top section will be the "Tweet Section" where the user can post tweets.
 - The bottom section will be the "View Tweets" where the user can view the tweets posted by him/her as well as from other users.

Tweet Section

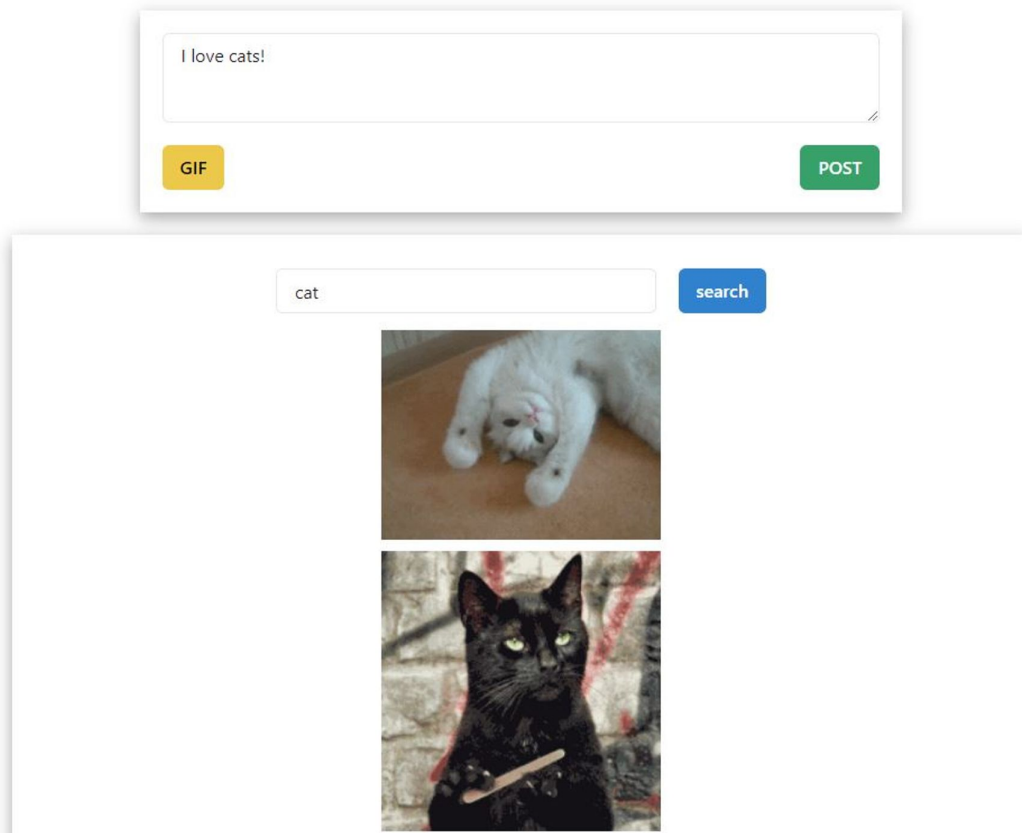
- This is the section where the user can post a tweet.
- The user should be prompted by a text field, where the user can type the tweet.
- Along with it, there will be a Gif button clicking on it, the user should be able to search and select a Gif and post it along with the text.
- You can use this API to implement the search Gif functionality

<https://developers.giphy.com/> **Note:** You have to signup and click on "Create an App" to generate an API key. You will be using "GIPHY Public API" as a resource and "Search" as endpoint



Demo video on how to create Giphy developer API key

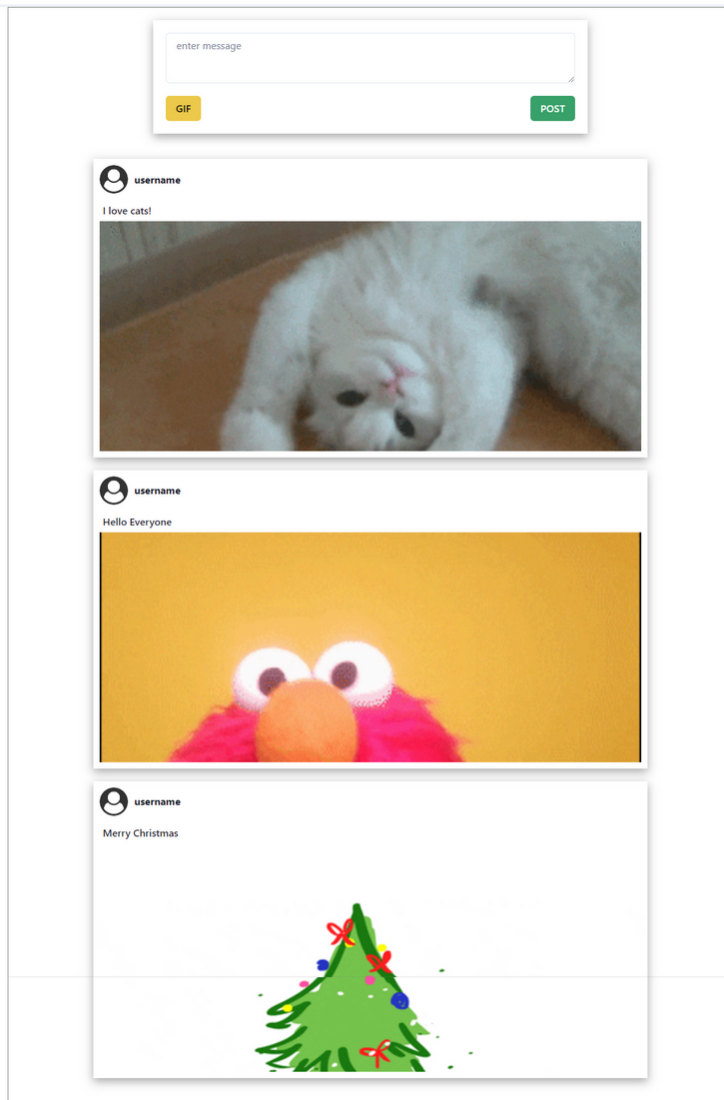
- All the posts by all users should be stored in your json-server on the route `"/posts"`.



Tweet Section Reference Image

View Tweets Section

- This is the section where the user can view all the tweets posted by him/her as well as from other users.
- Pagination should be implemented here with only 4 tweets per page



Timeline Reference Image

- For all the tweets posted by the user, who is currently logged in, there should be "Edit" and "Delete" buttons along with the post.
- Clicking on "Edit" the user should be able to edit the text content of that tweet/post.
- Clicking on "Delete" that tweet/post should be deleted from the frontend as well as from json-server.

Note: The user can only edit and delete tweets that are posted by himself.

Submission

- Please submit deployed link and Github link of code.
- ***Deploy JSON server too.

- Please double check if deployed version works or not (run the deployed version on your laptop and then submit it)
- Any issues in the deployed link, will be considered null and void.