Assignment: Post a Tweet on 100x Microblogging Platform Using Fetch API

Objective:

Create a JavaScript code snippet that allows users to post a tweet on the 100x Microblogging Platform by sending an HTTP POST request to the backend. The code should also fetch tweet details from the response and log the appropriate error messages for both success and failure cases. Since a React app has not been created yet, this code will be a standalone HTML and JavaScript page.

Instructions:

Part 1: JavaScript Code

Write JavaScript code to perform the following tasks:

1. Create an HTML file (`index.html`) and a JavaScript file (`app.js`) in the same directory.
2. In the `index.html` file, create a simple HTML structure with a form that allows users to input their tweet and a button to submit it.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>100x Microblogging</title>
</head>
<body>
    <h1>Post a Tweet</h1>
    <form id="tweetForm">
        <textarea id="tweetText" placeholder="Enter your tweet" rows="4" cols="50"></texta
rea>
        <br>
        <button type="submit">Post Tweet</button>
```

```
    </form>
    <div id="response"></div>
    <script src="app.js"></script>
</body>
</html>
```

1. In the `app.js` file, write JavaScript code that handles the form submission and makes an HTTP POST request to the backend.

JavaScript

```
document.addEventListener('DOMContentLoaded', function () {
    const tweetForm = document.getElementById('tweetForm');
    const tweetText = document.getElementById('tweetText');
    const responseDiv = document.getElementById('response');

    tweetForm.addEventListener('submit', function (event) {
        event.preventDefault();

        const tweet = tweetText.value;

        // Make an HTTP POST request to the backend
        fetch('backend-url-for-posting-tweet', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ tweet }),
        })
        .then(response => {
            if (response.ok) {
                return response.json(); // Parse response as JSON
            } else {
                throw new Error('Tweet posting failed'); // Handle failure
            }
        })
        .then(data => {
            // Handle successful tweet post
            responseDiv.innerText = `Tweet posted successfully! Tweet ID: ${data.tweetId}
`;
        })
        .catch(error => {
            // Handle error
            responseDiv.innerText = `Error: ${error.message}`;
        });
    });
```

```
});
```

Replace `'backend-url-for-posting-tweet'` with the actual URL of the backend API endpoint for posting tweets.

Part 2: Test the Code

1. Open the `index.html` file in a web browser.
2. Enter a tweet in the text area and click the "Post Tweet" button.
3. Observe the console in the web browser's developer tools to view the responses and error messages logged.

Note:

- This code snippet provides a basic example of how to send an HTTP POST request using the Fetch API to post a tweet on the 100x Microblogging Platform.
- You should have access to the backend API endpoint for posting tweets in order to test this code effectively.
- Ensure that you replace `'backend-url-for-posting-tweet'` with the actual URL of the backend API endpoint in your implementation.

Following these instructions will create a standalone HTML and JavaScript page that allows users to post tweets and handles both success and failure cases by logging appropriate error messages to the console.

Dummy API url - https://one00x-data-analysis.onrender.com/posts