

Walkthrough with React

Part - 4

Assignment



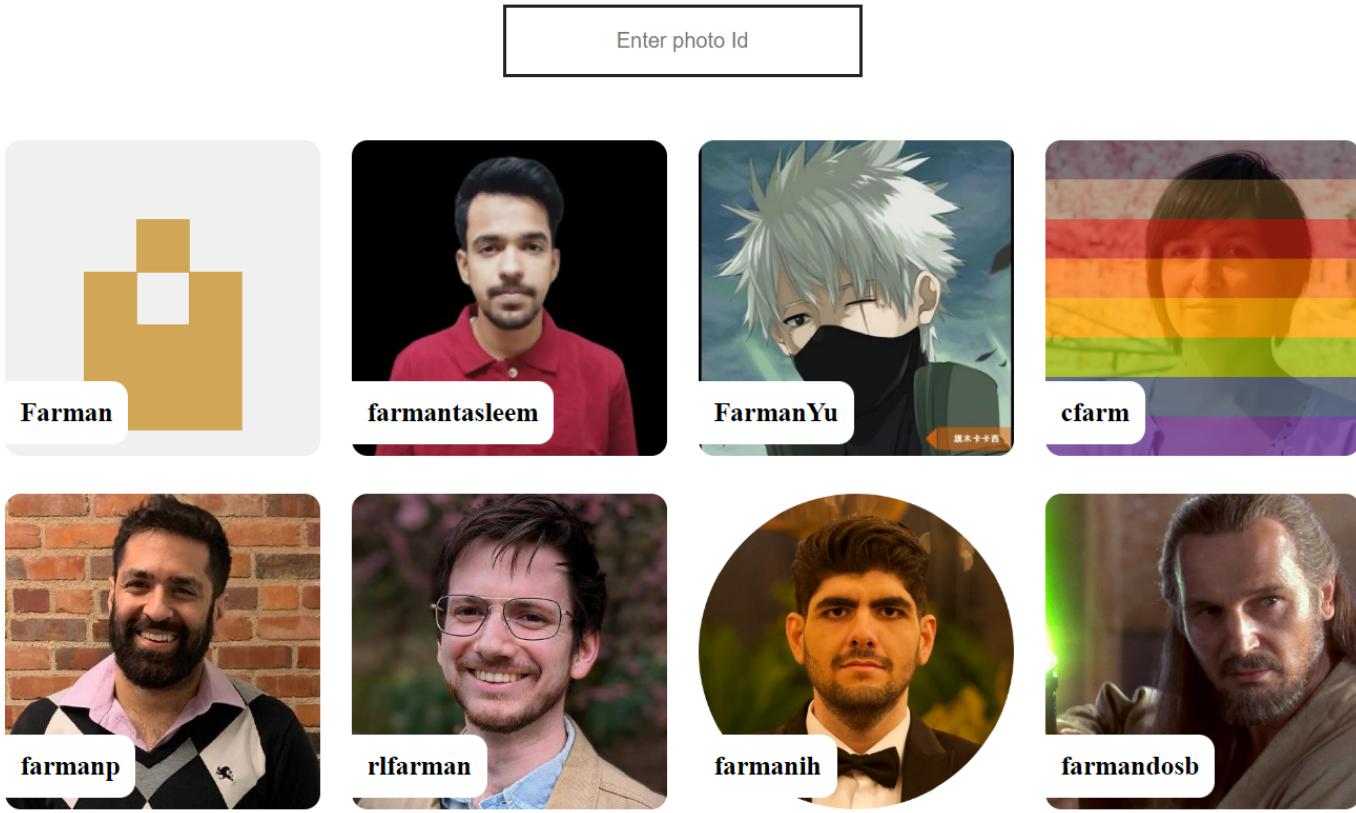
Assignment

Task: GitHub User Avatar Finder Web App

Objective:

Develop a web application using React that allows users to input a GitHub username. Upon user input, the app should display the avatar of the user fetched from the GitHub API. The app should incorporate debouncing to enhance user experience and prevent excessive API calls. To implement debouncing, create a custom hook called `useDebounce`.

Requirements:



User Interface (UI): Create a simple and user-friendly UI with a text input box and an image container.

The user should be able to type in a GitHub username in the input box.

GitHub API Integration:

Utilize the GitHub API to fetch user information, specifically the user's avatar image.

Handle API requests and responses effectively.

Avatar Display: Upon successful API response, display the user's avatar image in the designated container.

Debouncing: Implement debouncing to improve performance and prevent excessive API calls while the user is typing in the input box.

Create a custom hook named `useDebounce` to manage the debouncing functionality.

Custom Debounce Hook (`useDebounce`):

Assignment



The `useDebounce` hook should take an input value and a delay time as arguments.

Implement the debounce logic within the hook using `setTimeout`.

Return the debounced value after the specified delay has passed.

Submission Guidelines:

- Create a GitHub repository for the project and regularly commit your changes.
- Include a README file with instructions on how to run the app locally.
- Ensure your code is well-organized, follows best practices, and is free from any major bugs.
- Submit the GitHub repository link along with any additional instructions or notes.
- Do record video/images of you showing your code and output attach video link in github readme (highly Recommended not mandatory)
- Evaluation Criteria:

Your project will be evaluated based on the following criteria:

1. Implementation of the required UI components and functionality.
2. Proper integration of the GitHub API and handling of API responses.
3. Successful implementation of debouncing using the custom `useDebounce` hook.
4. Code quality, including modularity, readability, and appropriate use of comments.
5. Effective error handling and user-friendly error messages.
6. Overall user experience and visual appeal of the app.
7. Adherence to best practices and project structure.

Note:

Remember that this assignment is focused on assessing your skills in React development, API integration, and debouncing implementation. Focus on fulfilling the functional requirements outlined above. Good luck!