User Management Dashboard

### **Note: Please record your intro using** [**this link**](https://interviews.tacnique.com/async-interview/0c2a488c-d009-40a1-a90c-ca0d7fff5289/invite) **if you have not already.**

## Objective

Develop a simple web application where users can view, add, edit, and delete user details from a mock backend API.

## Requirements

* User Interface:
  + Display a list of users with details such as ID, First Name, Last Name, Email, and Department.
  + Provide buttons or links to "Add", "Edit", and "Delete" users.
  + A form to input details of a new user or edit details of an existing user.
* Backend Interaction:
  + Use [JSONPlaceholder](https://jsonplaceholder.typicode.com/), a free online REST API that you can use for demonstration and test purposes.
  + Specifically, use the '/users' endpoint to fetch and manipulate user data.
* Functionality:
  + View: Display all users by fetching data from the '/users' endpoint.
  + Add: Allow adding a new user by posting to the '/users' endpoint. (Note: JSONPlaceholder won't actually add the user, but will simulate a successful response.)
  + Edit: Allow editing an existing user. This should involve fetching the current data for a user, allowing for edits, and then putting the updated data back via the API.
  + Delete: Allow users to be deleted, by sending a delete request to the API.
* Error Handling:
  + Handle scenarios where the API request might fail - show an error message to the user in such cases.
* Bonus (Optional):
  + Implement pagination or infinite scrolling for the user list.
  + Add client-side validation for the user input form.
  + Make the interface responsive.

## Guidelines

* Your primary focus should be on functionality. However, a clean UI will be appreciated.
* You may use vanilla JavaScript or any JavaScript framework/library of your choice like React, Angular, Vue, etc.
* Consider using tools like Axios or Fetch API for HTTP requests.
* Ensure the solution is modular and scalable.
* Document any assumptions you make.

## Submission

* Share the above implementation using a public git repo
* Include a README detailing setup and run instructions.
* Reflect on the challenges faced during the development process and any improvements you would make if given more time.

## Timeline

* Your submission needs to be within **2 days** of receiving this document.
* The faster turn-around will be awarded bonus points.

**Please check the** [**Code Evaluation Criteria**](https://docs.google.com/document/d/1yA7S1w62iJFJpXQRsHHMfUEbs1ibaz8jn3P9S4n5rJQ/edit?usp=sharing) **carefully before proceeding.**