**Candidate Name:**

Faishal Ridha

## **1. Implement a REST API endpoint to fetch a list of users.**

Problem:  
You are tasked with creating an API endpoint /users that returns a list of users in JSON format. Assume the user data is stored in the following array:

const users = [

{ id: 1, name: "Alice", email: "alice@example.com" },

{ id: 2, name: "Bob", email: "bob@example.com" },

{ id: 3, name: "Charlie", email: "charlie@example.com" }

];

Requirements:

* Use Express.js to create the endpoint.
* Ensure the response has the correct HTTP status code and headers.
* Write code to handle potential errors (e.g., if the users array is empty).

<<Type your answer below>>

## **2. Handle API errors when fetching data from an external service.**

Problem:

Write a function in Node.js that fetches data from an external API (e.g., https://jsonplaceholder.typicode.com/posts) using the axios library. If the API call fails (e.g., network error or invalid URL), your function should log the error and return a meaningful response to the client.

Requirements:

* Use async/await syntax.
* Handle both network errors and HTTP errors (e.g., 404 or 500).
* Return a JSON response with an appropriate HTTP status code.

<<Type your answer below>>

## 

## **3. Create middleware to validate incoming JSON data in a POST request.**

Problem:  
You need to create middleware for an Express.js application that validates incoming JSON data for a POST request to /create-user. The request body should include name, email, and password. If any of these fields are missing or invalid, return a 400 Bad Request error with a descriptive message.

Requirements:

* Implement the middleware function.
* Use it in an Express route for /create-user.
* Ensure proper error handling and response formatting

<<Type your answer below>>