

# Backend Developer Internship Assessment.

## Task:

Candidates are required to build a backend system with specified routes for User and Admin functionalities using ExpressJS, Hono, or NextJS (file-based routing). The assessment focuses on basic backend development skills, including routing and data handling.

## TODO

### 1. User Routes

- /signin: Implement a sign-in route for users. This should authenticate users based on their credentials (username/email and password).
- /signup: Implement a sign-up route for users. This should register new users with necessary details (e.g., username, email, password).
- /update: Implement a route to update the user's profile data (e.g., name, email, password). Ensure basic authentication before allowing updates.

### 2. Admin Routes

- /signin: Implement a sign-in route for admins. This should authenticate admins based on their credentials.
- /users: Implement a route that fetches all user data. This should be accessible only to admins.
- /user/:id: Implement a route to fetch data of a particular user by their ID. This route should be accessible only to admins.
- /user/delete/:id: Implement a route to delete a specific user by their ID. This route should be accessible only to admins.

**Deadline :** Submit Your work before **11 Sep 11:59PM**

## Submission :

[https://docs.google.com/forms/d/e/1FAIpQLScLoNinquL\\_BMFkFMLIPt7C2Ne5GgskDe04XAGAIhIxyPN8dQ/viewform](https://docs.google.com/forms/d/e/1FAIpQLScLoNinquL_BMFkFMLIPt7C2Ne5GgskDe04XAGAIhIxyPN8dQ/viewform)

## **Help and tips:**

### **1. Set Up the Project**

- Initialize your project with the chosen framework (ExpressJS, Hono, or NextJS).
- Set up a simple database (e.g., SQLite or file-based) and create tables/collections for users and admins.
- Use Node.js
- Use any database (e.g., SQLite, a simple file-based database, or a lightweight database, you can also use free online database providers eg, neon, aiven etc).

### **2. Implement Authentication**

- Create basic authentication for the /signin routes (both User and Admin). Use simple password matching logic.

### **3. Implement User Routes**

- Signup Route: Implement a route to register new users with basic input validation.
- Signin Route: Implement user authentication with basic password matching.
- Update Route: Allow authenticated users to update their profile data with basic validation.

### **4. Implement Admin Routes**

- Signin Route: Implement basic authentication for admins.
- Users Route: Create a route to list all users. Ensure this route is restricted to admins.
- User by ID Route: Implement a route to fetch a specific user's data by their ID.
- Delete User Route: Create a route to delete a user by their ID. Ensure this route is restricted to admins.