

Backend Question: Enhanced Authentication API

Your task is to enhance the existing(create a new project) backend API for an authentication system to include a new feature allowing users to set their profiles as public or private. Additionally, you should implement functionality to allow admin users to view both public and private user profiles, while normal users can only access public profiles. You are required to use Node.js for the backend.

User Stories:

As a user, I can register a new account.

As a user, I can log in.

As a user, I can log in or register with at least one of the following services: Google, Facebook, Twitter, or GitHub.

As a user, I can sign out.

As a user, I can see my profile details.

As a user, I can edit my details including: photo, name, bio, phone, email, and password.

As a user, I can upload a new photo or provide an image URL.

As a user, I can choose to make my profile public or private.

As an admin user, I can see both public and private user profiles.

As a normal user, I can only see public user profiles.

Requirements:

- Use Node.js for the backend.
- Enhance the existing authentication system to include the option for users to set their profiles as public or private.
- Implement authorization checks to allow admin users to access both public and private profiles.
- Normal users should only be able to access public profiles.
- Ensure that private user details remain private to unauthorized users.
- Include endpoints for listing public profiles and retrieving user profiles based on user roles.
- Update the user profile editing functionality to include the option to set the profile as public or private.
- Implement proper error handling, validation, and security measures.
- Optionally, host the API on a platform like Heroku or similar.
- Optionally, use a library like Swagger to create an API playground for testing the API.

Hosting and API Playground (Optional):

Host the API on a platform like Heroku to make it accessible online. Provide clear instructions on how to access the hosted API.

Use a library like Swagger to create an API playground, allowing users to test the API endpoints interactively. Include instructions on how to access the Swagger documentation and utilize the playground.

Your solution will be evaluated based on its integration with the existing authentication system, adherence to the new user stories and requirements, and overall usability of the public and private profile features. Additionally, you should ensure that the API is secure, well-documented, and easy to use.

Please note:

Share the REPO link (Github)

Sharing Zip file will be Rejected