**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.

**SOLUTION**

**1. Introduction**

* The present article describes the features, implementation, and usage guidelines of the Enhanced Authentication API for User Profile Management System. Users of the system can create accounts, log in, edit their profiles, and choose whether to make them public or private. Additionally, admin users can access both private and public profiles with this functionality.
* The backend of the User Profile Management System is constructed with Node.js. By adding the ability for users to designate their profiles as private or public, it improves the current authentication system. Normal users can only access public profiles, but admin users can access both private and public profiles thanks to authorization checks implemented by the system.

**2. Technologies Used**

* Node.js
* Express.js
* MongoDB
* bcrypt for password hashing
* JSON Web Tokens (JWT) for authentication
* API Testing – Postman

**3. Tools Used**

* Visual Studio Code
* Postman

**4. API Endpoints**

* User Registration
* User Login
* User Login Token Verify
* User logout
* Admin Registration
* Admin Login
* Admin Login Token Verify
* Admin View Both the Profile (Public and Private)
* Get User Profile
* Edit User profile
* Image Upload
* Normal User View only Public User

**5. Authentication and Authorization**

* Because of their email address and password, users can register and log in.
* After successful authentication, JWT tokens are created and used for incoming requests.
* Normal users can only access public profiles; authorization checks are used to grant admin users access to both public and private profiles.

**6. Public and Private Profiles**

* Users have the option of keeping their profiles private or public.
* All users have access to public profiles.
* Only authorized users (admin users and the profile owner) can access private profiles.

**7. Handling Errors and Validation**

* The system as a whole implements proper error handling to deliver informative error messages.
* Validating input is done to make sure the data is secure and preserved.

**8. Conclusion**

Users can manage their profiles on a safe and convenient platform with the help of the Enhanced Authentication API for User Profile Management System. The system guarantees user data privacy and restricts access to authorized users only by following the given user stories and requirements. The documentation for the API and hosting options improve accessibility and usability.