Node JS project

Project Title: Employee Management System

Project Description:

Develop REST APIs for Employee Management Systems using MySQL and Node.js. This system aims to streamline the process of managing employee details within an organization. It provides functionalities for adding, updating, and deleting employee records, along with robust user authentication using JWT tokens.

Key Features:

User Authentication APIs:

- Implement registration and login APIs using JWT tokens for secure authentication and while registration it should accept image of user.
- Utilize bcrypt for password hashing to enhance security.
- Ensure that access to employee routes is restricted to authenticated users, returning a 401 unauthorized response if the JWT token is missing or invalid.

Employee APIs:

- Add new employees with details such as name, email, contact information, department, and designation.
- View existing employee records.
- Update employee details, including personal information, department, and designation.
- Delete employee records when necessary.
- Implement filtering of employee records based on designation.

Technologies:

- Node.js, Express.js, MySQL, MVC
- bcrypt.js (for password hashing), jsonwebtoken (for JWT token generation)

Middleware:

- Authentication Middleware: Develop middleware to authenticate incoming requests by verifying JWT tokens. This middleware should be applied to routes requiring authentication to ensure only authorized users can access them.
- Error Handling Middleware: Implement error handling middleware to catch and handle errors that occur during request processing. This middleware should provide appropriate error responses to clients, improving the application's robustness and user experience.

Project Title: Todo Application

Project Description:
Develop REST APIs for Todo Application using MySQL and Node.js, designed to help users organize their tasks effectively. This application offers features such as user registration, authentication, and task management. Users can register for an account, log in securely, view their existing todos, create new todos, update or delete existing ones, and search through their todo list.

Key Features:

User Authentication:

- Users can register for an account by providing necessary details such as username, email, image and password.
- Passwords are securely hashed before storage using bcrypt to ensure data security.
- Registered users can log in securely using their credentials.
- Ensure that access to todo routes is restricted to authenticated users, returning a 401 unauthorized response if the JWT token is missing or invalid.

Todo Management:

- Upon logging in, users are presented with their todo list, showing all existing tasks.
- Users can create new todos by providing a title, description, and optional due date.
- Existing todos can be updated with new information or marked as completed.
- Users have the ability to delete todos that are no longer needed.

Search Functionality:

- Users can search through their todo list using keywords to quickly find specific tasks.
- Search results are displayed in real-time, providing a seamless user experience.

Technologies:

- Node.js, Express.js, MySQL, MVC
- bcrypt.js (for password hashing), jsonwebtoken (for JWT token generation)

Middleware:

- Authentication Middleware: Develop middleware to authenticate incoming requests by verifying JWT tokens. This middleware should be applied to routes requiring authentication to ensure only authorized users can access them.
- Error Handling Middleware: Implement error handling middleware to catch and handle errors that occur during request processing. This middleware should provide appropriate error responses to clients, improving the application's robustness and user experience.