Employee Task Management System (ETMS) Project Report

**Overview**

The ETMS (Employee Task Management System) project aims to provide a comprehensive solution for managing tasks, users, and roles within an organization. The system is built using Node.js and Express.js for the backend, utilizing MySQL as the database to ensure efficient data storage and retrieval.

**Project Setup and Environment**

The initial phase involved setting up the project in VS Code, establishing the file structure, and configuring the Node.js environment. Key dependencies like Express.js and MySQL were installed to facilitate the development of the API.

**User Model and API**

A significant portion of the work was dedicated to the user model, which includes the implementation of registration, login, and retrieval of user data by ID. These APIs were thoroughly tested using Postman to ensure proper functionality.

**Role-Based Authorization**

Authorization roles were implemented to restrict access to certain functionalities. Specifically, the getall API was secured to be accessible only by users with the "admin" role, ensuring data privacy and security.

**Task Model and API**

The task model was developed with APIs for assigning, deleting, updating, and retrieving tasks. Permissions were assigned to "admin" and "subadmin" roles, allowing them to manage tasks effectively. The system supports CRUD (Create, Read, Update, Delete) operations.

**Pagination**

A pagination feature was added to the task management system, allowing users to view tasks in manageable chunks. This enhancement improves the user experience by loading only selected rows at a time.

**File Uploads**

Implemented file uploading functionality using Multer, which allows users to upload files (e.g., project descriptions or documents) to the system.

**Technologies Used**

* **Node.js**: Node.js is a powerful JavaScript runtime built on Chrome's V8 engine, designed for building scalable network applications. It allows developers to use JavaScript for server-side scripting, enabling the creation of dynamic web applications. Its non-blocking I/O model makes it efficient and suitable for handling multiple connections simultaneously, which is essential for real-time applications.
* **Express.js**: Express.js is a minimalist web application framework for Node.js that simplifies the process of building web applications and APIs. It provides robust features such as routing, middleware support, and error handling, allowing developers to create complex applications quickly and efficiently. With its flexible architecture, Express.js is widely used for developing RESTful APIs and single-page applications.
* **MySQL**: MySQL is a widely used open-source relational database management system (RDBMS) that stores data in structured tables. It supports SQL (Structured Query Language) for querying and managing data, making it ideal for applications requiring complex queries and transactions. MySQL's reliability, scalability, and strong community support make it a popular choice for web applications.
* **VS Code**: Visual Studio Code (VS Code) is a lightweight yet powerful source code editor developed by Microsoft. It supports various programming languages and offers features like debugging, syntax highlighting, and integrated terminal support. With its extensive marketplace of extensions, VS Code enhances productivity by providing tools tailored to specific development needs.
* **Postman**: Postman is a collaboration platform for API development that simplifies testing and documenting APIs. It allows developers to send requests to their APIs and view responses in real-time. Postman's user-friendly interface makes it easy to create test cases, automate workflows, and share API documentation with team members, ensuring effective communication throughout the development process.
* **Multer**: Multer is a middleware for handling multipart/form-data, which is primarily used for uploading files in Node.js applications. It simplifies the process of managing file uploads by providing easy-to-use methods for handling incoming files and storing them on the server. Multer's integration with Express.js allows developers to build robust file upload functionalities seamlessly.