Practical No. 9

Study and implementation of node.js

Perform following problem statements using Node.js

Problem Statement 1: Database Connectivity using SQL or Oracle

- Write a Node.js program that connects to an Oracle/SQL database, retrieves data from a table, and displays the results.

Problem Statement 2: Middleware (Express.js)

- What is middleware in Node.js, particularly in the context of Express.js?
- How do you create custom middleware in Express.js?
- Explain how middleware is executed in order in an Express.js application.

Problem Statement 3: File System (fs) Module

- How do you read and write files using the fs module in Node.js?
- What is the difference between fs.readFile() and fs.readFileSync()?
- How can you check if a file or directory exists in Node.js?
- How do you handle file operations in an asynchronous manner?

Problem Statement 4: File Upload and Download API

Develop a file upload and download API using Node.js and Express. The API should allow users to upload files (e.g., images, documents) and download them later.

- Create an API to upload files to the server.
- Implement routes to retrieve and download files.
- Ensure proper error handling (e.g., file size limits, invalid file formats).
- Implement file versioning to allow multiple uploads of the same file name without overwriting.

Problem Statement 5: Real-time Chat Application with Socket.io

Create a real-time chat application using Node.js, Express, and Socket.io that allows multiple users to join and communicate in a chat room.

- Set up a Node.js server with Socket.io for real-time bi-directional communication.
- Implement event listeners to handle user connections, disconnections, and message broadcasting to all connected users.

1. Create a **document** of the above questions.
2. Scan the document and **create a pdf file** with **"ExamSeatNum_P#PS#" as its name**.
3. Upload the file on the **WCE /ERP** before the given deadline.