Assignment 11

Roll No 71 T14

Aim : Create a web application that performs CRUD operations (database connectivity)

Thoery :

Creating a web application that performs CRUD (Create, Read, Update, Delete) operations with MongoDB and Mongoose involves several components and steps.

Database: MongoDB: MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It's well-suited for applications where data structures may evolve over time

Mongoose: Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a structured way to interact with MongoDB, defining schemas, models, and validation for your data.

CRUD Operations:

Create: In the context of MongoDB and Mongoose, creating a new record involves creating an instance of a Mongoose model and then saving it to the database using the save method. This process typically involves validating the data against the defined schema.

Read: Reading data from MongoDB can be done in various ways. You can perform simple queries, use filters, and projection to retrieve specific data. Mongoose provides methods like find, findOne, and findById to retrieve data.

Update: Updating data in MongoDB involves finding the document to update, making changes, and then saving the updated document. Mongoose provides methods like updateOne, updateMany, and findByIdAndUpdate.

Delete: Deleting a document can be done using the deleteOne or deleteMany methods in Mongoose, or you can use the findOneAndDelete method to find and delete a specific document by its ID.

Middleware: Middleware functions in Express or your chosen web framework can be used to handle common tasks such as input validation, authentication, and error handling. Middleware can be applied globally or to specific routes.

Routing: Define routes for your CRUD operations. For example, you might have routes like /create, /read/:id, /update/:id, and /delete/:id. These routes map to the corresponding CRUD operations.

Validation: Implement data validation using libraries like Joi or Express Validator. Ensure that the data sent by the client adheres to the schema defined for your MongoDB documents.

Schema Definition: Define schemas to specify the structure of your data. Schemas define the fields, their types, and any validation rules. This helps maintain consistency in your database.

Conclusion :

Created a web application which performs CRUD operations on Database.

Used MongoDb for Database and mongoose which is a JavaScript object-oriented programming library that creates a connection between MongoDB and the Node.js JavaScript runtime environment.