**Portfolio (Week-8)**

This week, we will focus on learning how to perform CRUD (Create, Read, Update, Delete) operations using MongoDB in a Node.js application. The tutorial guid us through integrating MongoDB commands into our application using Visual Studio Code. Below is a detailed account of what I learned and implemented during this week’s lab.

I started by creating a new folder named "MongoWeek" on my computer to store the project files.

A black rectangular object with white text

Description automatically generated

I opened the terminal and navigated to the "MongoWeek" directory. Then, I ran the command:

A screenshot of a computer

Description automatically generated

This command initialized a new Node.js project and created a package.json file.

A screenshot of a computer program

Description automatically generated

I created a new file named index.js in the project folder where all the coding would take place.

A screenshot of a computer

Description automatically generated

To connect to MongoDB, I needed to install Mongoose. I ran the following commands in the terminal:

A screenshot of a computer

Description automatically generated

In index.js, I added code to connect to MongoDB:

A screenshot of a computer

Description automatically generated

This code connects to a MongoDB database named "Week8". It also handles connection errors and logs when the connection is successful.

Next, I created a schema for storing person data:

A screenshot of a computer program

Description automatically generated

Then, I added code to create and save a single document:

The doc1.save() method saves the document to the database, and .then is used to handle successful insertion.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

To add multiple documents at once, I used this code:

A screenshot of a computer program

Description automatically generated

The insertMany method is used here to insert multiple documents into the collection.

To retrieve all documents from the collection without filtering, I added this code:

A screenshot of a computer program

Description automatically generated

For filtering documents based on criteria (e.g., gender), I wrote:

A screenshot of a computer program

Description automatically generated

To delete records where age is greater than 25, I used this code:

A screenshot of a computer program

Description automatically generated

Finally, to update all female records' salaries to 5555, I wrote:

A screenshot of a computer program

Description automatically generated

This week's lab provided me with hands-on experience in using MongoDB with Node.js for CRUD operations. I learned how to connect to a database using Mongoose, create schemas and models, insert single and multiple documents, fetch data with filtering options, delete records based on criteria, and update existing documents.The tutorial emphasized understanding how each function works within Mongoose and how they interact with MongoDB. This knowledge will be valuable for developing applications that require database interactions in real-world scenarios.I will include all relevant code snippets along with explanations in my portfolio for future reference.