**Portfolio (Week-10)**

In this week lab we will create a backend application, we will first create a folder for week 10 and open the vs code to initialize the npm init command:

A screenshot of a computer

Description automatically generatedThis created a package Jason file in our week 10 folder:

A screenshot of a computer

Description automatically generated

Now we will install the other necessary package in the file with the following command:

A screenshot of a computer

Description automatically generated

Now, we will create three files for our backend application:

A screenshot of a computer

Description automatically generated

Now we will add the code into the files with the help of the lab pdf:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

For the server.js file we will first create entry point of the backend server and define REST API endpoints:

A screenshot of a computer

Description automatically generated

Then we will initialize the server by:

A screenshot of a computer

Description automatically generated

Then we will define REST API endpoints to:

Get all books:

A screenshot of a computer

Description automatically generated

Get a single book by ID:

A screenshot of a computer program

Description automatically generated

Add a new book:

A screenshot of a computer

Description automatically generated

Update an existing book by ID:

A screenshot of a computer

Description automatically generated

Delete a book by ID: A screenshot of a computer program

Description automatically generated

And at the end start the server:

A screenshot of a computer program

Description automatically generated

Now we will run the server file with npm command:

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

For the backend of my online library project, I am using Node.js with Express for the server and MongoDB for the database, managed with Mongoose. The backend will expose several REST API endpoints to handle the core functionalities of the library system.

The first endpoint, GET /allbooks, will allow users to retrieve a list of all books available in the library. This will be handled using the MongoDB query Books.find({}), which fetches all documents from the Books collection. To retrieve details of a specific book, I will create a GET /getbook/:id endpoint that uses Books.findById(id) to find a book by its unique ID. For adding a new book to the collection, I will implement a POST /addbooks endpoint that uses new Books(req.body).save() to create and save a new book entry based on the data provided in the request body. To update the details of an existing book, the POST /updatebook/:id endpoint will be used, where Books.findByIdAndUpdate(id, updatedBook, { new: true }) will find the book by its ID and update it, returning the modified document. Finally, to delete a book from the library, I will have a POST /deleteBook/:id endpoint that uses Books.findByIdAndDelete(id) to locate and remove the book by its ID.

These API endpoints will cover the main operations needed for managing the book data in the online library, ensuring smooth interaction between the backend and the frontend application.

**REST API Overview**

1. **GET /api/allbooks**
   * Retrieves a list of all books in the collection.
2. **GET /api/getbook/:id**
   * Fetches details of a specific book by its ID.
3. **POST /api/addbook**
   * Adds a new book to the database using data from the request body.
4. **POST /api/updatebook/:id**
   * Updates details of an existing book by its ID.
5. **POST /api/deletebook/:id**
   * Deletes a book by its ID from the database.

**Reflections and Learning Outcomes**

This lab provided practical experience in backend development, including:

1. **REST API Development:** Understanding the implementation of API endpoints for CRUD operations.
2. **Node.js with Express:** Learning how to set up an Express server and integrate it with MongoDB using Mongoose.
3. **Mongoose Operations:** Gaining proficiency with common Mongoose methods such as find, findById, save, findByIdAndUpdate, and findByIdAndDelete.

These skills are critical for building robust backend systems and integrating them with frontend applications. This project serves as a foundational step toward developing full-stack applications.