ONE MONTH TRAINING AT HB INFOTECH WEEK4

Day 16 — MongoDB Introduction & Database Basics

• Learned what MongoDB is and how it stores data in **documents**, **collections**, and **databases**.

• Installed MongoDB Compass and practiced creating a new database and collection.

• Inserted sample documents manually using JSON format and understood how _id is auto-generated.

• Explored basic CRUD operations in MongoDB Compass:

– Insert Document

– Find, Filter, and Search

– Update Fields

– Delete Records

• Understood the difference between SQL tables and MongoDB's flexible schema structure.

Day 17 — Mongoose Setup & Schema Creation

• Installed Mongoose and connected Node.js backend to MongoDB using mongoose.connect().

• Created schemas for users/events with fields like name, email, date, description, status.

• Understood Mongoose Schema types (String, Number, Boolean, Date, Array, ObjectId).

• Built Mongoose Models and used them to perform CRUD operations programmatically.

• Practiced using methods like .find(), .findById(), .updateOne(), .deleteOne().

DAY 18 — Setting Up REST API Structure (Express + Mongoose)

• Set up the backend folder structure for building REST APIs using Express and Mongoose.

• Created separate folders for routes, controllers, models, and configuration.

• Installed required packages like express, mongoose, and dotenv.

• Connected Express server with MongoDB using mongoose.connect().

• Created the first basic route and tested it in the browser to ensure the server was working.

• Learned how Express Router helps divide the backend into clean, manageable modules.

DAY 19 — Creating CRUD Routes (Create & Read Operations)

• Created REST API endpoints for CREATE and READ operations.

• Built endpoints such as:

– POST /api/users (create user)

– GET /api/users (get all users)

– GET /api/users/:id (get a single user by ID)

• Connected each route with Mongoose Models to perform database operations.

• Added basic request body validation to ensure correct data is sent from the client.

• Tested all GET and POST routes manually in the browser and terminal.

• Verified that new data was successfully saved and retrieved in MongoDB Compass.

DAY 20 — Updating & Deleting (PUT and DELETE Routes)

• Implemented UPDATE and DELETE operations in the REST API.

• Created the following endpoints:

– PUT /api/users/:id (update a user's details)

– DELETE /api/users/:id (delete user record)

- Returned proper HTTP status codes (200, 201, 400, 404) based on success or failure.
- Tested update and delete behavior manually before using Postman.
- Confirmed all database updates and deletions through MongoDB Compass.