Day 1: Introduction to Node.js

- **Topics**:
  - What is Node.js? Why use it?
  - Node.js installation and setup.
  - Basic Node.js syntax and first program.
  - Node.js global objects (`__dirname`, `__filename`, `process`).
- **Exercises**:
  - Install Node.js and run a “Hello, World!” script.
  - Explore the global objects.

Day 2: Understanding Modules in Node.js

- **Topics**:
  - What are modules in Node.js?
  - Built-in modules: `fs`, `path`, `http`, etc.
  - Creating and exporting custom modules.
  - Importing modules using `require`.
- **Exercises**:
  - Create a basic custom module.
  - Use built-in modules like `fs` and `path` to read/write files.

Day 3: Working with File System (fs)

- **Topics**:
  - File reading and writing: Sync vs Async.
  - Reading and writing JSON files.
  - Working with directories.
  - Handling errors in file operations.
- **Exercises**:
  - Write a script that reads from and writes to a file.
  - Create a file system utility that checks if a directory exists, and creates it if not.

Day 4: Asynchronous Programming in Node.js

- **Topics**:
  - Callback functions.
  - Introduction to Promises.
  - Using `async` and `await` for asynchronous operations.
  - Error handling in asynchronous code.
- **Exercises**:
  - Convert a callback-based file read operation into a Promise-based one.
  - Practice error handling in asynchronous functions.

Day 5: Understanding HTTP and Server Creation

- **Topics**:
  - HTTP basics and creating a simple server.

  - Handling different routes and methods (GET, POST).
  - Sending responses in different formats (HTML, JSON).
  - HTTP status codes.
- **Exercises**:
  - Build a basic HTTP server that serves a simple HTML page.
  - Handle different routes and return appropriate content.

Day 6: Introduction to Express.js

- **Topics**:
  - What is Express.js? Why use it?
  - Setting up an Express server.
  - Defining routes and handling requests.
  - Using middleware in Express.
- **Exercises**:
  - Install Express and create a basic server.
  - Define a few routes and return responses.

Day 7: Handling Data with Express (GET & POST Requests)

- **Topics**:
  - Handling query parameters and route parameters.
  - Parsing request body with `body-parser`.
  - Handling form submissions.
- **Exercises**:
  - Create a form that submits data to your Express server via POST.
  - Practice handling query parameters in GET requests.

Day 8: Working with Databases (MongoDB)

- **Topics**:
  - Introduction to MongoDB and Mongoose.
  - Connecting Node.js to MongoDB.
  - CRUD operations: Create, Read, Update, Delete.
- **Exercises**:
  - Set up a MongoDB database and create a simple model with Mongoose.
  - Perform CRUD operations using MongoDB from your Node.js app.

Day 9: REST APIs with Express and MongoDB

- **Topics**:
  - What is a REST API?
  - Creating routes for API endpoints.
  - Performing CRUD operations using Express and MongoDB.
  - Testing your API using tools like Postman.
- **Exercises**:
  - Build a basic REST API for a resource (e.g., tasks, users).
  - Test your API endpoints using Postman.

Day 10: Authentication and Security

- **Topics**:
  - Introduction to authentication and JWT (JSON Web Token).
  - Securing routes with JWT.
  - Basic security practices in Node.js (e.g., environment variables, bcrypt for password hashing).
- **Exercises**:
  - Implement user authentication using JWT.
  - Secure routes to require valid tokens for access