

(Building Futures Through Digital Knowledge and Innovation)

Node.js & Express.js (03 Months)

Syllabus

01: Introduction to Node.js

- ✓ What is Node.js?
- ✓ Features and advantages of Node.js
- ✓ Node.js vs traditional server-side languages
- ✓ Installing Node.js and using the REPL
- ✓ Running your first Node.js script
- ✓ Understanding package.json
- ✓ Using npm (Node Package Manager)
- ✓ Introduction to CommonJS and ES Modules

02: Core Node.js Modules

- ✓ File System module (fs)
- ✓ Path module
- ✓ OS module
- ✓ Events module and EventEmitter
- ✓ Streams and buffers
- ✓ Working with asynchronous code: callbacks, promises, and async/await

03: Creating a Web Server with Node.js

- ✓ HTTP module
- ✓ Creating a basic HTTP server
- ✓ Handling routes and requests manually
- ✓ Sending responses with different headers and status codes
- ✓ Building a minimal REST API with core modules

04: Introduction to Express.js

- ✓ What is Express.js?
- ✓ Installing and setting up an Express project
- ✓ Creating routes (GET, POST, PUT, DELETE)
- ✓ Route parameters and query strings
- ✓ Using middleware in Express
- ✓ Static file serving

05: Express Middleware and Routing

- ✓ Built-in vs custom middleware
- ✓ Using `express.json()`, `express.urlencoded()`
- ✓ Creating reusable router modules
- ✓ Error handling middleware
- ✓ Logging with morgan
- ✓ CORS and security headers

06: Templating and Views (Optional)

- ✓ Introduction to template engines
- ✓ Using EJS or Pug with Express
- ✓ Rendering dynamic content
- ✓ Layouts and partials
- ✓ Passing data from controllers to views

07: Working with Databases (MongoDB)

- ✓ Introduction to NoSQL and MongoDB
- ✓ MongoDB vs relational databases
- ✓ Installing MongoDB locally or using MongoDB Atlas
- ✓ Connecting MongoDB to Node.js using Mongoose
- ✓ Mongoose schemas and models
- ✓ CRUD operations with MongoDB and Mongoose
- ✓ Validation and schema design

08: Authentication and Authorization

- ✓ Introduction to authentication
- ✓ Using bcrypt for password hashing
- ✓ Creating login and registration APIs
- ✓ Generating and verifying JWT tokens
- ✓ Protecting routes with middleware
- ✓ Role-based access control

09: RESTful APIs and Postman

- Understanding REST principles
- Building a complete RESTful API in Express
- Status codes and error messages
- Testing APIs using Postman
- Handling file uploads (Multer)
- Pagination, filtering, and sorting data

10: Express with Frontend and CORS

- ✓ Connecting frontend with backend (React, Angular, or HTML forms)
- ✓ Handling CORS issues
- ✓ Cross-Origin Resource Sharing and security

11: Sending Emails and Notifications

- ✓ Sending emails with NodeMailer
- ✓ Configuring SMTP (e.g., Gmail, Mailtrap)
- ✓ Sending transactional emails (password reset, welcome, etc.)
- ✓ Optional: Using external services like SendGrid

12: Unit Testing and Debugging

- ✓ Introduction to testing with Jest or Mocha/Chai
- ✓ Writing unit and integration tests for APIs
- ✓ Using Postman collections and scripts for testing
- ✓ Debugging Node.js apps with built-in debugger or VSCode

13: Job Scheduling and Background Tasks

- ✓ Using node-cron or agenda for scheduled jobs
- ✓ Performing background tasks
- ✓ Real-world use cases (sending reminders, clearing expired tokens, etc.)

14: Advanced Topics (Optional / Bonus)

- ✓ Socket.io for real-time communication (chat apps, notifications)
- ✓ Implementing WebSockets in Express
- ✓ Redis for caching or session storage
- ✓ Rate limiting and request throttling
- ✓ Using environment variables and .env
- ✓ Optimizing performance and memory usage

15: Deployment and Project Structure

- ✓ Environment configuration using .env
- ✓ Using PM2 for process management
- ✓ Deploying Node.js/Express apps to:
 - Heroku
 - Vercel
 - DigitalOcean or AWS (basic intro)
- ✓ Structuring a scalable Node.js project

16: Final Project

- ✓ Complete real-world project (Blog API, E-Commerce backend, Task Manager, or Chat App)
 - Full CRUD operations
 - Authentication and roles
 - API testing via Postman
 - Database integration
 - Hosting live version (optional)
- ✓ GitHub code submission