

KAUR'S CODING ACADEMY

Node.js, Express.js and MongoDB Backend Development

Module-wise Teacher Syllabus

A backend development course covering Node.js, asynchronous JavaScript, Express.js, REST APIs, MongoDB, Mongoose, validation, middleware, authentication, authorization, SQL basics, project work and interview preparation.

Course	Audience	Suggested Duration	Final Output
Node.js, Express.js and MongoDB Backend Development	Grades 10+, college students and professionals	50 classes / 12 to 16 weeks	REST API backend with MongoDB, authentication, authorization and project deployment readiness

Teacher Note

Teach backend visually with request-response diagrams. Use Postman frequently so students understand API testing before frontend integration.

Learning Outcomes

- Understand Node.js runtime, NPM, modules and server-side JavaScript.
- Create REST APIs using Express.js with clean route/controller structure.
- Connect MongoDB using Mongoose and perform CRUD, validation and aggregation.
- Implement authentication with bcrypt and JWT and role-based authorization.
- Build a full-stack-ready backend project and prepare for interviews.

Document Use

This PDF is made for teachers to plan classes module by module. Each module includes topic names, teaching points, class practice and revision work.

Complete Module Index

The table below gives a quick view of what will be taught in this course.

Module	Topic Name	Main Things Teacher Will Cover
1	Introduction to Node.js	Overview of Node.js, Benefits of Node.js, Event-driven model, Non-blocking I/O, and more
2	JavaScript Fundamentals for Backend	Variables and data types, Operators, Control structures, Functions, and more
3	Asynchronous JavaScript	Callbacks, Promises, Async/await, Error handling with async code, and more
4	Node.js Setup and NPM	Installing Node.js, NPM package manager, package.json file, Installing packages, and more
5	Core Node.js Modules	Modules in Node.js, File system module, Path module, Event module, and more
6	Understanding Servers and HTTP	What is a server?, Client-server architecture, HTTP request and response, Creating basic HTTP server, and more
7	Express.js Framework	Express introduction, Creating Express server, Routes, Serving static files, and more
8	APIs and Endpoints	Introduction to APIs, Endpoint structure, RESTful API rules, GET, POST, PUT, DELETE, and more
9	Introduction to MongoDB	NoSQL database concept, MongoDB overview, Database, collection and document, Installing/using MongoDB Atlas, and more
10	Mongoose Basics	What is Mongoose?, Connecting Node.js and MongoDB, Schemas and models, Data types, and more
11	Handling HTTP Requests with Express	GET, POST, PUT, PATCH and DELETE, req.params, req.query, req.body, and more
12	Postman API Testing	Postman overview, Creating requests, Headers and body, Testing GET/POST/PUT/DELETE, and more
13	Schema Validation with Mongoose	Required fields, Built-in validators, Custom validators, Async validators, and more
14	MongoDB Queries and Aggregation	find and findOne, Update and delete operations, Logical operators, Array operators, and more
15	Async/Await with Mongoose	Using async/await in controllers, try/catch error handling, Handling rejected promises, Reusable async handler concept, and more
16	Node.js Application Structure	MVC pattern, Routes folder, Controllers folder, Models folder, and more
17	Middleware in Express.js	What is middleware?, Application-level middleware, Router-level middleware, Custom middleware, and more
18	Authentication in Node.js	Authentication concept, User registration, Login flow, Password hashing with bcryptjs, and more
19	Authorization in Node.js	Authentication vs authorization, Role-based access, Admin/user permissions, Authorization middleware, and more
20	Full-Stack Backend Project and Interview Preparation	Project planning, Backend setup, User authentication and authorization, Frontend integration overview, and more

Detailed Module-wise Teaching Plan

Teachers should first explain the concept in simple language, then demonstrate code, then allow students to practice and customize the output.

Module 1: Introduction to Node.js	
Topics to be Covered	What Teacher Will Explain / Demonstrate
- Overview of Node.js - Benefits of Node.js - Event-driven model - Non-blocking I/O - Brief history and use cases - Where Node.js is used in real projects	- Explain JavaScript outside the browser - Use restaurant/order example for non-blocking flow - Show real use cases like APIs and real-time apps
Class Practice / Student Output	Homework / Revision
- Run first Node command - Create hello Node file - Discuss where backend is needed	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 2: JavaScript Fundamentals for Backend	
Topics to be Covered	What Teacher Will Explain / Demonstrate
- Variables and data types - Operators - Control structures - Functions - Objects and arrays - Modules basics	- Review only what is needed for backend - Demonstrate function-based business logic - Show object data used in APIs
Class Practice / Student Output	Homework / Revision
- Write utility functions - Create user/product objects - Practice conditions and loops	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 3: Asynchronous JavaScript	
Topics to be Covered	What Teacher Will Explain / Demonstrate
- Callbacks - Promises - Async/await - Error handling with async code - Why async matters in backend	- Explain waiting for database/API response - Demonstrate callback vs promise vs async/await - Show try/catch with await
Class Practice / Student Output	Homework / Revision
- Create delayed response examples - Convert promise code to async/await - Handle errors gracefully	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 4: Node.js Setup and NPM

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Installing Node.js - NPM package manager - package.json file - Installing packages - Updating and removing packages - Running scripts	- Show version check and npm init - Explain dependencies and devDependencies - Demonstrate npm scripts
Class Practice / Student Output	Homework / Revision
- Create basic Node app - Install nodemon - Run server script	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 5: Core Node.js Modules

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Modules in Node.js - File system module - Path module - Event module - Creating and exporting custom modules	- Explain built-in vs external modules - Demonstrate reading/writing a file - Show module export/import
Class Practice / Student Output	Homework / Revision
- Create custom utility module - Read a text file - Use path for file location	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 6: Understanding Servers and HTTP

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What is a server? - Client-server architecture - HTTP request and response - Creating basic HTTP server - Status codes overview	- Draw request-response cycle - Demonstrate basic HTTP server in Node.js - Explain status codes with examples
Class Practice / Student Output	Homework / Revision
- Create basic server - Return text/JSON response - Test in browser	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 7: Express.js Framework

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Express introduction - Creating Express server - Routes - Serving static files - Route parameters and query parameters - Basic error handling	- Explain why Express simplifies backend work - Demonstrate GET route - Show params and query usage
Class Practice / Student Output	Homework / Revision
- Build Express server - Create multiple routes - Serve static folder	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 8: APIs and Endpoints

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Introduction to APIs - Endpoint structure - RESTful API rules - GET, POST, PUT, DELETE - Request body and response JSON	- Explain REST with user/product examples - Demonstrate all HTTP methods - Show clean endpoint naming
Class Practice / Student Output	Homework / Revision
- Create simple CRUD API with in-memory data - Test endpoints - Return proper status codes	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 9: Introduction to MongoDB

Topics to be Covered	What Teacher Will Explain / Demonstrate
- NoSQL database concept - MongoDB overview - Database, collection and document - Installing/using MongoDB Atlas - Basic MongoDB operations	- Compare SQL tables and MongoDB documents - Show sample document structure - Demonstrate database connection options
Class Practice / Student Output	Homework / Revision
- Create database and collection - Insert sample documents - Read data using basic commands	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 10: Mongoose Basics

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What is Mongoose? - Connecting Node.js and MongoDB - Schemas and models - Data types - CRUD operations with Mongoose	- Explain schema as data blueprint - Demonstrate model creation - Show create, find, update and delete operations
Class Practice / Student Output	Homework / Revision
- Create user/product model - Connect database - Perform CRUD with Mongoose	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 11: Handling HTTP Requests with Express

Topics to be Covered	What Teacher Will Explain / Demonstrate
- GET, POST, PUT, PATCH and DELETE - req.params - req.query - req.body - Response methods - Status codes and messages	- Demonstrate request object parts - Show proper response structure - Explain status codes for success and errors
Class Practice / Student Output	Homework / Revision
- Create CRUD routes - Test request body handling - Return JSON with success messages	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 12: Postman API Testing

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Postman overview - Creating requests - Headers and body - Testing GET/POST/PUT/DELETE - Saving collections	- Explain why backend is tested separately - Demonstrate request setup - Show how to create a Postman collection
Class Practice / Student Output	Homework / Revision
- Test all CRUD APIs - Save Postman collection - Take screenshots of API responses	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 13: Schema Validation with Mongoose

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Required fields - Built-in validators - Custom validators - Async validators - Error message formatting	- Explain validation before saving data - Demonstrate wrong data rejection - Show clean error response to frontend
Class Practice / Student Output	Homework / Revision
- Add validation to schema - Test invalid request - Return readable validation errors	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 14: MongoDB Queries and Aggregation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- find and findOne - Update and delete operations - Logical operators - Array operators - Projection - Aggregation stages: match, group, sort	- Demonstrate common filters - Explain projection to return selected fields - Show simple aggregation for reports
Class Practice / Student Output	Homework / Revision
- Create filter APIs - Add search by name/email/date - Build simple report endpoint	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 15: Async/Await with Mongoose

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Using async/await in controllers - try/catch error handling - Handling rejected promises - Reusable async handler concept - Database error patterns	- Explain async database operations - Demonstrate controller with try/catch - Show common mistakes with missing await
Class Practice / Student Output	Homework / Revision
- Refactor routes to async controllers - Add error handling - Test failed database cases	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 16: Node.js Application Structure

Topics to be Covered	What Teacher Will Explain / Demonstrate
- MVC pattern - Routes folder - Controllers folder - Models folder - Config files - Environment variables	- Explain MVC through restaurant/service counter analogy - Demonstrate moving code from one file to structured folders - Show clean import/export
Class Practice / Student Output	Homework / Revision
- Organize backend project - Create route/controller/model folders - Move database config separately	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 17: Middleware in Express.js

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What is middleware? - Application-level middleware - Router-level middleware - Custom middleware - Error-handling middleware - Logging and security basics	- Explain middleware as a checkpoint - Demonstrate custom logger - Show error middleware flow
Class Practice / Student Output	Homework / Revision
- Create logger middleware - Add error handler - Protect one route with middleware	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 18: Authentication in Node.js

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Authentication concept - User registration - Login flow - Password hashing with bcryptjs - JWT generation - JWT verification	- Explain signup/login securely - Demonstrate hashing password - Show token creation and protected route access
Class Practice / Student Output	Homework / Revision
- Build register API - Build login API - Create JWT-protected route	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 19: Authorization in Node.js

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Authentication vs authorization - Role-based access - Admin/user permissions - Authorization middleware - Protected resources	- Explain role access with school/admin example - Demonstrate checking user role from token - Show forbidden response
Class Practice / Student Output	Homework / Revision
- Add admin-only route - Test user vs admin access - Return 401/403 responses correctly	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 20: Full-Stack Backend Project and Interview Preparation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Project planning - Backend setup - User authentication and authorization - Frontend integration overview - DBMS overview - SQL vs NoSQL - Project assessment - Common interview questions	- Guide final project milestones - Review API design and code structure - Discuss Node, JavaScript and MongoDB interview questions
Class Practice / Student Output	Homework / Revision
- Build task manager/e-commerce backend - Submit Postman collection and GitHub repo - Practice project explanation	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Final Review and Certification Checklist

- Student can explain the main concepts in simple words.
- Student has completed the required practice tasks and mini projects.
- Student can run the project, fix common errors and present the work confidently.
- For advanced courses, student should submit GitHub repository, screenshots and deployment link where possible.