



# Learning Objectives: Node.js/Express

Last updated Spring 2026.

## Course Objectives

By the end of this course, students will:

- Apply fundamental Node.js programming concepts to build server-side applications.
- Work effectively with asynchronous JavaScript, including Promises, async/await, and the event loop.
- Utilize Node.js core modules for file system operations, HTTP servers, and process management.
- Build RESTful APIs using Express.js with proper middleware, routing, and error handling.
- Implement data validation and input sanitization to create secure web applications.
- Use SQL databases with Node.js, including both raw SQL queries and ORM tools like Prisma.
- Implement user authentication and authorization using JWT tokens and secure session management.
- Write comprehensive automated tests for Node.js applications using Jest and Supertest.
- Deploy Node.js applications to cloud platforms and manage production environments.
- Apply security best practices including password hashing, CORS, rate limiting, and input validation.
- Collaborate effectively on Node.js projects using version control and modern development practices.

## Lesson-by-Lesson Objectives

Week	Lesson Name	Students will be able to...
1	Node.js Fundamentals	Gain foundational knowledge of the Node.js runtime environment, including the event loop, asynchronous programming concepts, and core differences between browser and server-side JavaScript. They will learn to work with Node.js core modules for file system operations, understand the V8 engine, and practice with Promises and async/await patterns. Students will also explore Node.js documentation and package management with npm.
2	Events, HTTP Servers,	Learn about event emitters and listeners in Node.js, create

## CTD Learning Objectives: Node.js/Express

---

	and Express	HTTP servers using the built-in http module, and understand the transition to Express.js for easier web development. They will implement basic REST APIs, handle different HTTP methods and routes, parse request bodies, and implement graceful shutdown procedures. Students will also learn to test APIs using tools like Postman.
3	Express Middleware and Application Structure	Deepen their understanding of Express.js by learning about middleware functions, application structure, and the middleware chain. They will implement logging middleware, error handling middleware, and 404 handlers. Students will learn about request/response objects, content-type validation, and how to organize Express applications with proper separation of concerns.
4	Security, Middleware, Validation, and Password Hashing	Learn to protect routes using middleware and implement comprehensive security measures. They will learn data validation using Joi schemas for input sanitization and security, implement password hashing with bcrypt, and create authentication middleware. Students will build a complete task management system with CRUD operations while implementing proper security practices and input validation.
5	Intro to SQL and Databases	Gain foundational knowledge of SQL databases, learning about relational database concepts, primary and foreign keys, and basic SQL operations (SELECT, INSERT, UPDATE, DELETE). They will understand database schemas, constraints, and relationships. Students will practice writing complex SQL queries including JOINS, subqueries, GROUP BY, and HAVING clauses. They will also learn to integrate PostgreSQL databases with Node.js applications using raw SQL queries, set up database connections, and perform CRUD operations with the <code>pg</code> library.
6	Prisma ORM Integration	Learn to transform their PostgreSQL applications from raw SQL queries to using the Prisma ORM. They will set up and configure Prisma in their projects, create Prisma schemas, implement migrations, and perform CRUD operations using Prisma Client. Students will understand the benefits of ORMs, learn about type-safe database operations, and implement Prisma features like relationships and error handling. They will replace raw SQL queries with Prisma methods while maintaining the same functionality.
7	Advanced Prisma and Data Modeling	Master Prisma ORM features including advanced querying, relationships, and data modeling. They will implement complex database operations, learn about Prisma

## CTD Learning Objectives: Node.js/Express

---

		migrations, and build analytics features. Students will understand database performance considerations and learn to optimize queries. They will also implement bulk operations and advanced data retrieval patterns.
8	Authentication and Security	Implement comprehensive authentication and authorization systems using JWT tokens, password hashing, and secure session management. They will learn about security best practices including CORS, rate limiting, input sanitization, and protection against common vulnerabilities. Students will implement user registration, login, logout, and protected routes with proper access control.
9	Automated Testing	Learn automated testing concepts and write comprehensive test suites using Jest and Supertest. They will write unit tests for validation schemas and controller functions, integration tests for API endpoints using Supertest, and learn testing best practices including mocking, test organization, and achieving good test coverage. Students will also learn about test-driven development (TDD) principles and how to test authentication flows and protected routes.
10	Deployment and Production	Learn to deploy Node.js applications to cloud platforms, specifically using Render.com for hosting and Neon.tech for database services. They will understand production environment considerations, environment variables, database migrations in production, and monitoring. Students will also learn to integrate their backend with a React frontend and handle CORS configuration.
11	Project Enhancement and Best Practices	Explore advanced Node.js concepts and project enhancement ideas. They will learn about project initialization, team collaboration workflows, and advanced features they can add to their applications. Students will understand production considerations, performance optimization, and how to continue learning and growing as Node.js developers.

## Final Project Outcomes

Students will build a comprehensive Task Management application that demonstrates mastery of all course concepts.

## Core Requirements

- **Backend API:** Full REST API with user authentication and task management

## CTD Learning Objectives: Node.js/Express

---

- **Database Integration:** PostgreSQL database with Prisma ORM
- **Security:** JWT authentication, password hashing, input validation, and security middleware
- **Testing:** Comprehensive test suite with Jest and Supertest
- **Deployment:** Live deployment on Render.com with Neon.tech database
- **Frontend Integration:** Works with provided React frontend

### Technical Skills Demonstrated

- Node.js and Express.js proficiency
- Database design and ORM usage
- Authentication and security implementation
- Automated testing practices
- Cloud deployment and production considerations
- API design and documentation
- Error handling and validation
- Code organization and best practices