**Session 1: Introduction to Node.js and NPM (Module 1)**

* **Duration:** 1 hour
* **Topics Covered:**
  + What is Node.js?
  + Key features and benefits
  + Installing Node.js and NPM
  + Writing and running a simple Node.js script
* **Practical:**
  + Install Node.js
  + Run a simple "Hello, World!" script
  + Explore Node.js REPL
  + Install and explore basic NPM packages (chalk, etc.)

**Session 2: Node.js Core Concepts (Module 2)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Event-Driven Architecture
  + The Event Loop in Node.js
  + Synchronous vs Asynchronous programming
  + Introduction to Callbacks
* **Practical:**
  + Write asynchronous code using callbacks
  + Explore the setTimeout and setInterval functions

**Session 3: Promises and Async/Await (Module 2)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to Promises
  + Error handling in Promises (catch)
  + Async/Await in Node.js
* **Practical:**
  + Refactor callback-based code into Promises
  + Write asynchronous code using Async/Await

**Session 4: Node.js Modules (Module 2)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Built-in modules (fs, http, path)
  + Creating and using custom modules
  + Exporting and importing modules
  + Introduction to NPM
* **Practical:**
  + Use the fs module to read/write files
  + Build and use a custom module
  + Install and use a third-party module from NPM

**Session 5: File System in Node.js (Module 3)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Reading and writing files asynchronously
  + Managing directories and files
  + File permissions and file system utilities
* **Practical:**
  + Write a file explorer utility using the fs module

**Session 6: Streams and Buffers (Module 3)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Understanding Streams (Readable, Writable, Duplex, Transform)
  + Working with Buffers in Node.js
  + Streaming data between files
* **Practical:**
  + Create a file streaming service to read and write large files

**Session 7: Building an HTTP Server (Module 4)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to HTTP server in Node.js
  + Handling different HTTP methods
  + Routing and serving static content
* **Practical:**
  + Build a basic web server that handles GET and POST requests
  + Serve static files (HTML, CSS, JS)

**Session 8: Introduction to RESTful APIs (Module 4)**

* **Duration:** 1 hour
* **Topics Covered:**
  + What are RESTful APIs?
  + JSON as the data format
  + Designing a simple REST API
* **Practical:**
  + Build a REST API for a task management system
  + Test API using Postman

**Session 9: Introduction to Express.js (Module 5)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to Express.js
  + Setting up an Express.js application
  + Routing in Express
* **Practical:**
  + Set up a basic Express app
  + Implement simple routing with GET and POST methods

**Session 10: Middleware in Express.js (Module 5)**

* **Duration:** 1 hour
* **Topics Covered:**
  + What is Middleware?
  + Using built-in and third-party middleware in Express
  + Error handling middleware
* **Practical:**
  + Use middleware to log request details and handle errors
  + Add morgan and body-parser middleware

**Session 11: Working with MongoDB and Mongoose (Module 6)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to MongoDB
  + Setting up and connecting MongoDB with Node.js
  + Introduction to Mongoose ODM (Object Document Mapping)
  + CRUD operations in MongoDB using Mongoose
* **Practical:**
  + Create a MongoDB database
  + Build a CRUD app using Mongoose to manage a collection

**Session 12: Working with MySQL/PostgreSQL (Module 6)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to relational databases
  + Setting up MySQL or PostgreSQL
  + Connecting Node.js to relational databases
  + Using Sequelize ORM for database operations
* **Practical:**
  + Build a CRUD app using Sequelize with a relational database

**Session 13: User Authentication with JWT (Module 7)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to JWT (JSON Web Token)
  + Password hashing with bcrypt
  + Implementing user authentication
* **Practical:**
  + Implement JWT-based user authentication
  + Build a login and registration system

**Session 14: Role-Based Authorization (Module 7)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Role-based access control (RBAC)
  + Defining user roles and permissions
  + Protecting routes based on roles
* **Practical:**
  + Implement role-based access control for protected routes

**Session 15: WebSockets and Real-Time Communication (Module 8)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to WebSockets
  + Setting up socket.io in Node.js
  + Broadcasting and handling real-time data
* **Practical:**
  + Build a real-time chat application using socket.io

**Session 16: Task Scheduling with Node.js (Module 8)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Task scheduling in Node.js
  + Using node-cron to automate tasks
  + Running scheduled jobs in the background
* **Practical:**
  + Implement a scheduled task (like email notifications) using node-cron

**Session 17: Unit Testing with Mocha and Chai (Module 9)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Introduction to unit testing
  + Using Mocha for writing test cases
  + Asserting results with Chai
* **Practical:**
  + Write unit tests for a simple API using Mocha and Chai

**Session 18: Debugging Node.js Applications (Module 10)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Built-in debugging tools in Node.js
  + Using console.log for debugging
  + Using Node Inspector and VSCode debugger
* **Practical:**
  + Debug a Node.js application using the built-in debugger and VSCode

**Session 19: Deploying Node.js Applications (Module 11)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Deployment basics (Heroku, AWS, etc.)
  + Setting up a production environment
  + Using process managers (PM2) for production applications
* **Practical:**
  + Deploy a Node.js app to Heroku
  + Set up PM2 to manage the app

**Session 20: Security and Performance Optimization (Module 12)**

* **Duration:** 1 hour
* **Topics Covered:**
  + Best practices for securing Node.js applications
  + Preventing common vulnerabilities (SQL injection, XSS, etc.)
  + Performance optimization techniques
* **Practical:**
  + Implement security features (like input validation)
  + Optimize a Node.js app using caching (Redis)