**RoadMap**

Basic JavaScript: Before starting with Node.js, it's important to have a good understanding of JavaScript. Learn the basics of variables, data types, functions, and control structures.

1. Node.js: Next, you should learn the basics of Node.js. Understand the event-driven, non-blocking I/O model and the modules that come with Node.js. Learn how to create a simple web server using the http module.
2. Express.js: Express.js is a popular web framework for Node.js. Learn how to use it to create web applications. Understand routing, middleware, and templates.
3. MongoDB: MongoDB is a popular NoSQL database that is often used with Node.js. Learn how to install MongoDB and interact with it using the MongoDB driver for Node.js.
4. Mongoose: Mongoose is an Object Data Modelling (ODM) library for MongoDB and Node.js. Learn how to use Mongoose to define schemas and models, and how to perform CRUD operations.
5. Authentication and Authorization: Learn how to implement authentication and authorization in your Node.js application using libraries like Passport.js and jsonwebtoken.
6. Error Handling: Learn how to handle errors in your Node.js application. Understand the different types of errors and how to handle them using try-catch blocks, error middleware, and error logging.
7. Testing: Learn how to test your Node.js application using libraries like Mocha, Chai, and Sinon. Understand different types of tests, including unit tests, integration tests, and end-to-end tests.
8. Deployment: Learn how to deploy your Node.js application to a production server using tools like PM2 and Docker. Understand how to configure and optimize your application for production.
9. Best Practices: Learn best practices for writing efficient, scalable, and maintainable Node.js applications. Understand concepts like code organization, performance optimization, and security.

**Basics**

1. Introduction to Node.js: Understand what Node.js is and why it's used. Learn about the features and benefits of Node.js.
2. Installation: Learn how to install Node.js on your machine and verify that it's working.
3. Node.js modules: Understand the concept of modules in Node.js and how to import and export modules. Learn about the built-in modules in Node.js like **fs**, **http**, and **path**.
4. Asynchronous programming: Node.js is designed to handle asynchronous programming. Learn how to work with call-backs, promises, and async/await to handle asynchronous tasks.
5. File system operations: Learn how to read and write files using the built-in **fs** module.
6. HTTP server: Learn how to create an HTTP server using the built-in **http** module. Understand how to handle incoming requests and send responses.
7. Routing: Learn how to handle different routes in your HTTP server using the **URL** module and the **req.url** property.
8. NPM: NPM is the package manager for Node.js. Learn how to install packages and manage dependencies in your Node.js project.
9. Debugging: Learn how to debug your Node.js application using the built-in **debugger** statement or using tools like **node-inspector**.
10. Error handling: Learn how to handle errors in your Node.js application using try-catch blocks and error middleware.
11. Streams: Learn how to work with streams in Node.js for efficient handling of large files and data.
12. Child processes: Learn how to create and communicate with child processes in Node.js.
13. OS and Process: Learn how to interact with the operating system and process environment variables using the built-in **os** and **process** modules.

**Express Js**

1. Introduction to Express.js: Understand what Express.js is and why it's used. Learn about the features and benefits of Express.js.
2. Installation: Learn how to install Express.js using NPM.
3. Routing: Learn how to handle different routes in your Express.js application using the **Router** middleware.
4. Middleware: Learn how to use middleware to perform common tasks like parsing request bodies, handling errors, and logging requests.
5. Views and templates: Learn how to use views and templates to generate HTML responses in your Express.js application using libraries like **Pug** or **EJS**.
6. Static files: Learn how to serve static files like images, CSS, and JavaScript in your Express.js application.
7. Database integration: Learn how to integrate your Express.js application with databases like MongoDB using the **mongoose** module.
8. Authentication and authorization: Learn how to implement user authentication and authorization in your Express.js application using libraries like **passport** and **jsonwebtoken**.
9. RESTful APIs: Learn how to build RESTful APIs in your Express.js application using the **express. Router** middleware and the **mongoose** module.
10. Error handling: Learn how to handle errors in your Express.js application using middleware like **errorhandler**.
11. Security: Learn how to implement security measures like CSRF protection and XSS prevention in your Express.js application.
12. Testing: Learn how to write automated tests for your Express.js application using frameworks like **Mocha** and **Chai**.

**MongoDb**

1. Introduction to MongoDB: Understand what MongoDB is and why it's used. Learn about the features and benefits of MongoDB.
2. Installation: Learn how to install MongoDB on your machine and verify that it's working.
3. Data modelling: Learn how to create a data model for your MongoDB database. Understand how to use collections and documents to organize your data.
4. CRUD operations: Learn how to perform CRUD (Create, Read, Update, Delete) operations on your MongoDB database using the **mongo** shell and the **mongoose** module in Node.js.
5. Indexing: Learn how to create indexes in your MongoDB database to improve query performance.
6. Aggregation: Learn how to use the aggregation pipeline in MongoDB to perform complex queries and data transformations.
7. Replication and sharding: Learn how to set up replication and sharding in your MongoDB cluster for high availability and scalability.
8. Geospatial data: Learn how to work with geospatial data in MongoDB using the **geoNear** and **geoWithin** operators.
9. GridFS: Learn how to use GridFS to store and retrieve large files like images and videos in MongoDB.
10. Security: Learn how to secure your MongoDB database by enabling authentication, configuring access control, and implementing SSL/TLS encryption.
11. Backup and restore: Learn how to perform backups and restores of your MongoDB database using tools like **mongodump** and **mongorestore**.