MERN Hands-on Workshop

OVERVIEW

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

A Node.js along with the Express framework help quickly create backend web applications. A popular choice of database is MongoDB.

React is a JavaScript library for building user interfaces. It is an open source library and is developed by Facebook. It can be used to create single page applications (SPAs). Redux is a JavaScript library used to manage an application's front-end state. As single-page apps become more complex, the need for proper state management has become more important than ever. The main benefits of using Redux come in when your app involves shared state.

React together with MongoDB, Express and Node.js, this choice for full-stack is popularly referred to as the MERN stack.

PREREQUISITES

- Working knowledge of HTML & CSS
- Good knowledge of JavaScript
- Bootstrap knowledge is a plus, but not necessary
- Very basic algorithms and data structures knowledge (traversing and using arrays, using objects in JavaScript)
- Good knowledge of web application concepts

© Naveen Pete naveen.shenoy.blr@gmail.com
+91-98864-95913

TOPICS COVERED

1. Introduction to Node.js
    a. What is Node.js?
    b. Node.js architecture
    c. Event loop
    d. Blocking vs Non-blocking
    e. NPM
    f. Installing Node.js
    g. Node REPL
2. Node Module System
    a. Global object
    b. Custom Modules
        i. Creating a custom module
        ii. Loading a custom module
        iii. Using a custom module
    c. Core Modules
        i. Path module
        ii. OS Module
        iii. File System Module
        iv. Events Module
        v. Http Module
    d. Third-Party Modules
3. Node Package Manager (NPM)
    a. Introduction
    b. Package.json
    c. Installing a package
    d. Using a package
    e. Viewing Registry info for a package
    f. Installing a specific version of a package
    g. Updating local packages
    h. Dev dependencies
    i. Uninstalling a package
    j. Working with global packages
4. Introduction to Express
    a. What is Express?
    b. Installing Express
    c. Express middleware
        i. Custom middleware
        ii. Built-in middleware
            1. JSON parser
            2. URL Encode parser
            3. Serving static files
        iii. Third-party middleware
            1. Morgan

    d. Template engines
    e. Express App Generator
5. Building RESTful APIs using Express
    a. Introduction to RESTful services
    b. Routing
    c. Route parameters
    d. Query parameters
    e. Handling HTTP GET requests
    f. Handling HTTP POST requests
    g. Handling HTTP PUT requests
    h. Handling HTTP DELETE requests
    i. Testing API endpoints using Postman
    j. Refactoring routes into separate modules
        i. Express Router
6. Asynchronous JavaScript
    a. Synchronous vs Asynchronous code
    b. Patterns for Asynchronous code
    c. Callbacks
    d. Callback hell
    e. Promises
    f. Async and Await – Overview
7. Introduction to MongoDB
    a. NoSQL – What & Why?
    b. Overview of MongoDB
    c. Setting up MongoDB
    d. JSON
    e. BSON
    f. CRUD using Mongo Shell
    g. MongoDB Node.js driver
8. Mongoose
    a. Introduction
    b. Connecting to MongoDB
    c. Schema and schema types
    d. Model
    e. Document
    f. Querying documents
    g. Validation
9. ES6
    a. Overview
    b. Array helper methods
    c. Const & let
    d. Template literals
    e. Arrow functions
    f. Default function arguments
    g. Rest & Spread

    h. Destructuring – Array & Object
    i. Classes & Inheritance
    j. Promises
    k. Fetch
10. Fundamentals of React
    a. What is React?
    b. Why React?
    c. Components
    d. Overview of virtual DOM & diffing algorithm
    e. Introduction to JSX
11. More React
    a. Stateless (Functional) components
    b. Stateful (Class) components
    c. Composing components
    d. Props
    e. Prop Types
    f. State
    g. Props vs State
    h. Lifecycle methods
    i. Fragments
    j. Forms
        i. Handling events
        ii. Handling lists
        iii. Controlled components
12. React Router
    a. Why React Router?
    b. Installing React Router
    c. BrowserRouter component
    d. Link component
    e. NavLink component
    f. Route component
    g. Switch component
13. Introduction to Redux
    a. Why Redux?
    b. Core concepts of Redux
        i. Actions
        ii. Action creators
        iii. Reducers
        iv. The Store
14. React & Redux
    a. The react-redux Node.js package
    b. Provider component
    c. Connecting React components with Redux store
        i. connect() method
        ii. mapStateToProps() method

iii. mapDispatchToProps() method
d. Reducer composition – combineReducers() method
e. Redux Middleware
i. Redux Thunk
15. React & Node.js Integration
a. Implementing user authentication
b. Handling authorization
16. Unit Testing React Applications
a. Overview of unit testing

DAYWISE BREAKUP OF TOPICS

Sl. No.	Topic	Day
1	Introduction to Node.js	1
2	Node Module System	1
3	Node Package Manager (NPM)	2
4	Introduction to Express	2
5	Building RESTful APIs using Express	3
6	Asynchronous JavaScript	3
7	Introduction to MongoDB	4
8	Mongoose	4
9	ES6	5
10	Fundamentals of React	5
11	More React	6
12	React Router	6
13	Introduction to Redux	7
14	React & Redux	7
15	React & Node.js integration	8
16	Unit Testing React Applications	9