

## Full Stack Web Development with JavaScript

### 1. Introduction to JavaScript

- Introduction to web development and JavaScript
- How to create a JavaScript application
- How to test and debug a JavaScript application
- XHTML
- CSS

### 2. JavaScript Essentials

- How to get input and displays output
- How to work with numbers, strings, dates
- How to code control statements
- How to create and use array
- How to create and use function
- How to create and use objects
- How to use regular expressions, handle exceptions, and validate data

### 3. DOM Scripting

- Basic DOM scripting
- event handling
- DOM manipulation
- How to script CSS
- How to script tables and forms
- Animation with DOM scripting

### 4. Advanced JavaScript: Objects and Functions

- Object: Inheritance and the Prototype Chain
- Creating Objects: Function Constructors
- The Prototype Chain in the Console
- Creating Objects: Object.create
- Primitives vs. Objects
- First Class Functions: Passing Functions as Arguments
- First Class Functions: Functions Returning Functions
- Immediately Invoked Function Expressions (IIFE)

### 5. Asynchronous JavaScript: Promises, Async/Await and AJAX

- An Example of Asynchronous JavaScript
- Asynchronous JavaScript: The Event Loop
- Asynchronous JavaScript with Callbacks
- From Callback Hell to Promises
- From Promises to Async/Await
- AJAX and APIs
- Making AJAX Calls with Fetch and Promises
- Making AJAX Calls with Fetch and Async/Await

### 6. JavaScript: Intro to ES6

- What's new in ES6

- Variable Declarations with LET and CONST
- Strings in ES6
- Arrow Functions: Basics
- Arrow Functions: Lexical 'this' Keyword
- DE structuring
- Arrays in ES6
- The Spread Operator
- Rest Parameters
- Default Parameters
- Classes

## 7. TypeScript

- What is typescript?
- Typescript installation
- TypeScript vs. JavaScript o OOPs in TypeScript
- Creating Classes and class properties
- Methods
- Access Modifiers
- Constructor
- Getters and Setters
- Static properties and methods
- Inheritance
- Interfaces
- Modules and Namespaces

## 8. Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Appropriate Use Cases
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- Anatomy of an Angular Application
- Running the Application

## 9. Components

- What is a Component?
- An Example Component
- Creating a Component Using Angular CLI
- The Component Class
- The @Component Decorator
- Registering a Component to Its Module
- Component Template
- Using a Component
- Run the Application
- Component Hierarchy

- The Application Root Component
- The Bootstrap File

## **10. Component Templates**

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example - Deleting an Item
- Item Tracking with ngFor
- Swapping Elements with ngSwitch
- Template Reference Variable

## **11. Inter Component Communication**

- Communication Basics
- The Data Flow Architecture
- Preparing the Child to Receive Data
- Send Data from Parent
- More About Setting Properties
- Firing Event from a Component
- @Output () Example - Child Component
- @Output () Example - Parent Component
- Full Two-Way Binding
- Setting up Two Way Data Binding in Parent

## **12. Template Driven Forms**

- Template Driven Forms
- Importing Forms Module
- Basic Approach
- Setting Up a Form
- Getting User Input
- Omitting ngForm Attribute

- Initialize the Form
- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons

### **13. Reactive Forms**

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template
- FormControl Constructor
- Getting Form Values
- Setting Form Values
- The Synchronous Nature
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator

### **14. Services and Dependency Injection**

- What is a Service?
- Creating a Basic Service
- The Service Class
- What is Dependency Injection?
- Injecting a Service Instance
- Injectors

### **15. Pipes and Data Formatting**

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- The number Pipe
- Currency Pipe
- Create a Custom Pipe

- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- Filter Pipe

## **16. HTTP Client**

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Simple Example
- Service Using HttpClient
- ES6 Import Statements
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- Error Handling
- Customizing Error Object with . catch ()
- Making a POST Request
- Making a PUT Request
- Making a DELETE Request

## **17. The Angular Component Router**

- The Component Router
- View Navigation
- The Angular Router API
- Creating a Router Enabled Application
- Hosting the Routed Components
- Navigation Using Links and Buttons
- Programmatic Navigation
- Passing Route Parameters
- Navigating with Route Parameters
- Obtaining the Route Parameter Values
- Retrieving a Route Parameter
- Routing Enabled Feature Module
- Using the Feature Module
- Lazy Loading the Feature Module
- Creating Links for the Feature Module
- Components
- More About Lazy Loading
- Preloading Modules
- routerLinkActive binding
- Default Route
- Wildcard Route Path
- redirectTo
- Child Routes

- Defining Child Routes
- for Child Routes
- Links for Child Routes
- Navigation Guards
- Creating Guard Implementations
- Using Guards in a Route

## **18. Advanced HTTP Client**

- Request Options
- Returning an HttpResponse Object
- Setting Request Headers
- Creating New Observables
- Creating a Simple Observable
- The Observable.create() Method
- Observable Operators
- More About map
- Piping Operators
- The flatMap() Operator
- The tap() Operator
- The zip() Operator
- Caching HTTP Response
- Making Sequential HTTP Calls
- Making Parallel Calls
- Customizing Error Object with catchError ()

## **19. Unit Testing Angular Applications**

- Unit Testing Angular Artifacts
- Testing Tools
- Typical Testing Steps
- Test Results
- Jasmine Test Suites
- Jasmine Specs (Unit Tests)
- Expectations (Assertions)
- Matchers
- Examples of Using Matchers
- Using the not Property
- Setup and Teardown in Unit Test Suites
- Example of before Each and after Each Functions
- Angular Test Module
- Example Angular Test Module
- Testing a Service
- Injecting a Service Instance
- Test a Synchronous Method
- Test an Asynchronous Method
- Using Mock HTTP Client

- Supplying Canned Response
- Testing a Component
- Component Test Module
- Creating a Component Instance
- The Component Fixture Class
- Basic Component Tests
- The Debug Element Class

## **20. Welcome to React**

- Obstacles and Roadblocks
- React's Future
- Working with Files
  - React Developer Tools
  - Installation Node JS

## **21. Pure React**

- Page Setup
- The Virtual DOM
- React Elements
- React DOM
- Children
- Constructing Elements with Data
- React Components
- DOM rendering

## **22. React with JSX**

- React Elements as JSX
- Babel
- Intro to Webpack

## **23. Props, State and the Component Tree**

- Property Validation
- Refs
- React State Management
- State within component Tree

## **24. Enhancing Components**

- Component Lifecycle
- JavaScript Library Integration
- Higher-Order Components
- Flux

## **25. Redux**

- State
- Actions

- Reducers
- The Store
- Action creators
- Middleware

## **26. React Redux**

- Explicitly Passing the Store
- Passing Store via Context
- Presentation Versus Container Components
- The React Redux Provider
- React Redux Connect

## **27. React Router**

- Incorporating the Router
- Nesting Routes
- Route Parameters

## **28. Understanding Node environment**

- Extending JavaScript
- V8
- The Process Object

## **29. Understanding Asynchronous Event Driven Programming**

- Broadcasting Events
- Listening for Events
- Timers
- Understanding the Event Loop
- Callbacks and errors

## **30. Streaming Data Across Node And clients**

- Exporting Steams
- Creating and HTTP Server
- The Request objects
- Working with Headers
- Handling Post Data

## **31. Access the File System**

- Directories & iterating overs files and folders
- Reading from File
- Writing to a File

## **32. Introduction to Express**

- What is use of Express Framework?
- Install Node and necessary dependencies



### **33. Getting Started with Express**

- Express: The Basics
- Basic Routing
- Serving Static Files

### **34. Middleware and Rendering**

- Express Middleware
- Responding with JSON
- Express with View Engine
- API/ Server-Side rendering

### **35. Request & Response**

- Request Object – Forms and Cookies
- Query String
- Data Params
- Headers
- Router
- Express Generator
  - Rendering in Express (EJS)
  - HandleBars
  - Pub/Jade

### **36. Project: Building API**

- Project Overview and Setup
- Routes and Middleware
- Movie Routes and Headers

### **37. Databases**

- Relational Database and SQL
- Node & MYSQL
- NoSQL and Documents
- MongoDB and Mongoose
- CRUD operation using MongoDB
- Working with MySQL
- CRUD Operation using MYSQL

### **38. API Integration with JS Frameworks**

- Connecting Angular with API
- Connecting React with API
- Wrapping Up.