

# Front End Web Development

Table of contents:

## **Programming in HTML5 with JavaScript and CSS3**

### **Module 1: Overview of HTML and CSS**

Lessons

- Overview of HTML
- Overview of CSS

### **Module 2: Creating and Styling HTML**

Lessons

- Creating an HTML5 Page
- Styling an HTML5 Page

### **Module 3: Introduction to JavaScript**

Lessons

- Overview of JavaScript
- Introduction to the Document Object Model

### **Module 4: Creating Forms to Collect and Validate User Input**

Lessons

- Creating HTML5 Forms
- Validating User Input by Using HTML5 Attributes
- Validating User Input by Using JavaScript

### **Module 5: Communicating with a Remote Server**

Lessons

- Async programming in JavaScript
- Sending and Receiving Data by Using the XMLHttpRequest Object
- Sending and Receiving Data by Using the Fetch API

### **Module 6: Styling HTML5 by Using CSS3**

Lessons

- Styling Text by Using CSS3
- Styling Block Elements
- Pseudo-Classes and Pseudo-Elements
- Enhancing Graphical Effects by Using CSS3

## **Module 7: Creating Objects and Methods by Using JavaScript**

### Lessons

- Writing Well-Structured JavaScript Code
- Creating Custom Objects
- Extending Objects

## **Module 8: Creating Interactive Pages by Using HTML5 APIs**

### Lessons

- Interacting with Files
- Incorporating Multimedia

## **Module 9: Adding Offline Support to Web**

### Lessons

- Reading and Writing Data Locally
- Adding Offline Support by Using the Application Cache

## **Module 10: Implementing an Adaptive User**

### Lessons

- Supporting Multiple Form Factors
- Creating an Adaptive User Interface

## **Module 11: Creating Advanced Graphics**

### Lessons

- Creating Interactive Graphics by Using SVG
- Drawing Graphics by Using the Canvas API

## **Module 12: Animating the User Interface**

### Lessons

- Applying CSS Transitions
- Transforming Elements
- Applying CSS Keyframe Animations

## **Module 13: Implementing Real-time Communication by Using Web Sockets**

### Lessons

- Introduction to Web Sockets
- Using the WebSocket API

## **Module 14: Performing Background Processing by Using Web Workers**

### Lessons

- Understanding Web Workers
- Performing Asynchronous Processing by Using Web Workers

# **Bootstrap Framework**

## **Introduction to Bootstrap**

- Introduction
- Bootstrap File Structure
- Default Grid System
- Fluid Grid System
- Responsive Design

## **Bootstrap CSS**

- Typography
- Tables
- Forms
- Button
- Images
- Icons

## **Bootstrap Layout Components**

- Dropdown Menus
- Button Groups
- Button with Dropdowns
- Navigation Elements
- Navbar
- Breadcrumbs

- Pagination
- Labels
- Badges
- Typographic Elements
- Thumbnails
- Alerts
- Progress Bars

### **Bootstrap JavaScript Plugins**

- Model
- Dropdown
- Scrollspy
- Toggleable Tabs
- Tooltips
- Popover
- Alerts
- Buttons
- Collapse

## **Angular Framework**

### **Angular Fundamentals**

- Introduction to Single Page Application
- Setting up Development Environment
- Angular Initial Application
- Angular Directory Structure
- Angular Building Blocks
- Working with Modules

### **Angular Component**

- Working with Component
- Angular Components and the @Component Decorator
- Creating an Angular Component using the Angular CLI
- Component Decorator Metadata
- Data Binding
  - Interpolation
  - Property Binding
  - Event Binding
  - Two-way Data Binding [(ngModel)]

## Angular Directive

- Structural Directive
  - \*ngIf
  - \*ngSwitch
  - \*ngFor
  - \*ngFor with trackBy
- Attribute Directive
  - NgStyle
  - NgClass

## Angular Pipes

- Introduction to Pipes in Angular
- Using Built-in Pipes
- Creating Custom Pipes in Angular

## Angular Component Communication

- @Input ()
- @Output ()

## Angular Services

- Dependency Injection
- Working with Services
- Calling Backend
  - Working with JSON Data
  - GET data from a Rest API using HttpClient
- Observables
  - Interface
  - Implement Observables
  - HTTP Methods
  - CRUD Operations using HttpClient, HttpHeaders
  - Adding HTTP Error Handling with RxJS catchError () & HttpClient
  - RXJS Operator – retry, repeat, map, filter, take, tap

## Angular Forms

- Template Driven Form
- Validating Template Driven Form
- Reactive Form
- Validating Reactive Form
- Accessing Form Value
- Resetting Form Value

## **Angular Routing**

- Introduction to Routing
- Implementing Routing in an Angular App
- Implementing Child Routes
- Introduction to Route Guards in Angular
- CanActivate and CanDeactivate Guards in Angular

## **Angular Best Practice**

- Feature Module and Shared Module in Angular
- Lazy Loading Modules
- Integrate Chart library in Angular project
- Working with Angular Material

## **React JS**

### **Welcome to React**

- Introduction to React
- React Developer Tools
- Installation Node JS

### **Emerging JavaScript**

- Declaring Variable in ES6
- Arrow function
- ES6 Objects and Arrays
- Promises

### **Pure React**

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

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

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

## **Enhancing Components**

- React Hooks
- JavaScript Library Integration
- Higher-Order Components

## **Redux**

- State
- Actions
- Reducers
- The Store
- Action creators
- Middleware

## **React Redux**

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

## **React Router**

- Incorporating the Router
- Nesting Routes
- Route Parameters

## **React on the Server**