ReactJS Web Application Development

Duration: 4 Days *(Face-to-Face & Remote-Live)*, or 28 Hours *(On-Demand)*

Price: $2095 *(Face-to-Face & Remote-Live)*, or $1495 *(On-Demand)*

Discounts: We offer multiple discount options. Click here for more information.

Delivery Options: Attend face-to-face in the classroom, remote-live or on-demand training.

Students Will Learn

- Rendering React Components
- Styling Components with CSS and Bootstrap
- Using `Create React App`
- Fetching External Data with Fetch API
- Leveraging JSX for UI Design
- Creating Functional and Class-based Components
- Working with Forms
- Using React Hooks
- Single Page Applications with React Router
- Validating Props with PropTypes
- Using Lifecycle Methods
- Maintaining Component and Global State
- Registering Event Handlers
- Animating React Components

Course Description

React (a.k.a. ReactJS or React.js) is a JavaScript library for developing user interfaces. This hands-on course introduces students to the React JavaScript library and covers essentials such as using `Create React App`, defining components, writing and styling JSX elements, passing props, using state and registering event handlers. Students will also learn how to use React Hooks, the Context API, Lifecycle Methods and how to implement global state using the Redux JavaScript library.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Course Prerequisites

Knowledge of HTML5, CSS, and JavaScript equivalent to attending the Website Development with HTML5, CSS and Bootstrap and JavaScript Programming courses.

Course Overview

<

React Fundamentals

- Overview of React
- Components
- React Tree Reconciliation
- Uni-directional Data Flow
- `React` and `ReactDOM`

React Development Environment

- Environment Setup
  - NodeJS
  - VS Code
  - VS Code Extensions for React
  - Installing React
  - `Create React App`
- Overview of Babel and Webpack
- Using `Create React App`

ES6 Features in React

- ES Modules
- Declaring variables with let and const
- Arrow functions
- Template Literals
- Rest Parameters and Spread Operator
- Array/Object Destructuring
- ES6 Classes
- Callbacks
- Promises
- `Async/Await`

JSX

- Role of JSX
- Rendering JSX Elements
- Writing JSX Expressions
- Comments in JSX
- React Fragments

Responding to User Events

- Defining Event Handlers
- Issues with Events
- Meaning of this
- SyntheticEvent
- Accessing Event Properties
- Event Pooling

Working with Forms

- Using Interactive Properties
  - `value`
  - `defaultValue`
  - `checked`
  - `selected`
- Controlled inputs
- Uncontrolled inputs
- Capturing Component Updates with onChange
- Intercepting Form Submission

Styling JSX

- How to style JSX Elements
- Applying CSS Classes to JSX
- Defining Style Objects
- Styling with Bootstrap and other CSS Frameworks
- Animating React Components

Working with State

- What is State?
- Setting Initial State
- Mutating State with `setState()`
- Problems with State

Fetching External Data

- Using the Fetch API
- `Async/Await`
- Using Promises

Defining Components

- What are Components?
- Presentational vs Container Components

- Communicating with External APIs
- Using axios for fetching data

- Rules for Defining Components
- Creating Functional Components
- Limitations of Functional Components
- Creating Class-based Components
- Defining a `constructor()` Method

Lifecycle Methods

- What are Lifecycle Methods?
- Mounting
  - `constructor`
  - `componentWillMount`
  - `render`
  - `componentDidMount`
- Rendering
  - `componentWillReceiveProps`
  - `shouldComponentUpdate`
  - `componentWillUpdate`
  - `render`
  - `componentDidUpdate`
- Unmounting
  - `componentWillUnmount`
- Dealing with State and Prop Changes
- Error Handling
  - `componentDidCatch`

Using React Router

- Creating Single Page Applications in React
- What is React Router?
- Fundamentals of React Router
- Defining Routers
- Route and Switch components
- React-router objects
  - `Match`
  - `History`
  - `Location`
- Authenticating Routes
- Route Parameters

Using the Context API

- Why use Context?
- Creating a Context Object
- Defining Providers and Consumers
- Using the Render props pattern

Using React Hooks

- What are Hooks?
- Adding state to functional components with `useState`
- `useEffect`
- `useContext`
- `useReducer`

Passing Data with Props

- What are Props?
- Passing Props to Components
- Receiving Props
- Handling Children
- Validating props with PropTypes
  - Data Type Validation
  - Making Properties Required
- "Prop drilling" Explained
- Simplifying Prop drilling with the spread operator
- Communicating with Parent Components

Deploying a React Application

- Using npm run build
- Using npm run deploy
- Configuring Client-side Routing
- Deploying to GitHub Pages

14 Fletcher Street
Chelmsford, MA 01824