Course Title:
React with Redux

Course Overview:
The React with Redux course is designed to provide participants with a comprehensive understanding of building modern web applications using React.js and Redux. The course covers the fundamentals of React.js, state management with Redux, and the integration of Redux with React components. Participants will learn how to design and develop scalable and maintainable applications by leveraging the power of React.js and Redux.

Course Duration:
5 Days

Course Pre-requisites

- Attendees should have knowledge of
  - HTML 5 & CSS 3
  - ECMASCRIPT 2015+ (Preferred knowledge on ECMAScript 2018)
  - Bootstrap 5

Course Objectives:

- Build powerful, fast, user-friendly and reactive web apps
- Provide amazing user experiences by leveraging the power of JavaScript with ease
- Become fluent in the toolchain supporting React, including NPM, Webpack, Babel, and ES6/ES2015 JavaScript syntax
- Build amazing single page applications with React JS and Redux
- Master fundamental concepts behind structuring Redux applications
- Realize the power of building composable components
- Learn React Hooks & Class-based Components

Lab Setup:

- Node.js – Latest LTS Version (https://nodejs.org/en/)
- Visual Studio Code – Latest Version (https://code.visualstudio.com/)
- Internet Connection

Course Outline

- Introduction to React
  - Why React?
  - Advantages of React
  - React Development Style

- Get Started with React
  - Setting up a React App
  - Using React CLI (create-react-app)
  - Running a React App
  - React Initialization Process
  - React Virtual DOM
  - React Ecosystem

- Create React App (CRA) Setup and Development Workflow
  - Installing Create React App (CRA)
  - Creating a new React project with CRA
  - Exploring the project structure and configuration files
  - Running the development server and building for production

- JSX and React Components
  - Introduction to JSX
  - Render Elements
  - Understanding Component
  - Creating a Component
  - Container vs. Presentation Components
  - Props and State
  - Component Lifecycle

- Data Flow
  - Introduction to One Way Data Flow in React
  - Props and State
  - Passing Data to Child Component
  - Handling State of Component

- Event and Actions
  - Events in React
  - Data Binding
  - Events and Actions
  - Handling Events
  - Communication Among Components

- Debugging React Application
  - React Dev Tools

    - Chrome Debugger
    - Chrome Extension for Visual Studio Code

- Ajax Calls and Error Handling
    - Create AJAX Status Actions
    - Create AJAX Status Reducer
    - Show and Hide Component Based on Ajax Status
    - Using ES6 Promise
    - Error Handling

- Context API
    - When to Use Context
    - Before You Use Context
    - API
        - React.createContext
        - Context.Provider
        - Class.contextType
        - Context.Consumer
        - Context.displayName
    - Updating Context from a Nested Component
    - Consuming Multiple Contexts

- React Hooks
    - What are React Hooks?
    - React Hooks
        - useState
        - useEffect
        - useRef
        - useContext
    - Custom Hooks

- React Forms and Controls
    - To be done by participants (Manual will be shown using React Bootstrap)
    - Introduction to React Hook Forms library and its benefits
    - Installing and setting up React Hook Forms in a React project
    - Basic Form Structure and Usage
        - Creating a basic form with React Hook Forms
        - Handling form submission and data management
        - Accessing form values and managing form state
    - Form Input Components and Validation
        - Working with different input components in React Hook Forms
        - Built-in and custom validation rules
        - Displaying form validation errors and error messages
    - Customizing Form Validation Behavior
        - Implementing custom validation logic with React Hook Forms

        - Building custom validation rules and error messages
        - Handling real-time validation and input feedback

- Routing
    - Understanding Routing
    - Defining Routes
    - Routes Handling
    - Router Context
    - URL Parameters
    - Custom Link
    - Preventing Transitions
    - Animated Transitions
    - No Match (404)
    - Ambiguous Matches
    - Securing Routes
        - Understanding JWT
        - Redirects (Auth)
    - Lazy Loading the components
        - lazy()
        - Suspense

- Introduction to Redux
    - Core principles of Redux (store, actions, reducers)
    - Setting up Redux in an application
    - Understanding the Redux data flow

- Redux Toolkit Overview
    - Introduction to Redux Toolkit and its benefits
    - Simplifying Redux setup with configureStore
    - Using createSlice for reducer and actions boilerplate reduction

- Reducers and Actions with Redux Toolkit
    - Creating reducers with createSlice
    - Defining actions and action creators
    - Handling actions and updating state

- Asynchronous Operations with Redux Toolkit
    - Handling asynchronous operations with createAsyncThunk
    - Managing loading and error states
    - Updating state based on asynchronous responses

- Advanced State Management with Redux Toolkit
    - Combining multiple slices with createSlice
    - Utilizing advanced reducer logic and immer integration
    - Working with selectors for efficient state access

- Performance Optimization with Redux Toolkit
  - Memoization techniques with createSelector for efficient state updates
  - Using Redux Toolkit's built-in immutability and performance optimizations

- Debugging and DevTools with Redux Toolkit
  - Leveraging Redux Toolkit's built-in DevTools support
  - Debugging Redux state and actions with Redux DevTools
  - Analyzing Redux state changes and time-travel debugging