React Full Stack Development with ASP.NET Core 6

Prerequisites: Knowledge of JavaScript and Dot Net Programming

Day 1: React.js Fundamentals

Topics:

- **Introduction to React:**
  - What is React? Key Features and Advantages.
  - Setting up the development environment (Node.js, npm/yarn, React).
- **Core Concepts:**
  - JSX and Virtual DOM.
  - Functional and Class Components.
  - Props and State.
- **Basic Events and Rendering:**
  - Handling Events in React.
  - Conditional Rendering.

Lab:

- Create a basic React app with two components: **Header** and **ItemList**.
- Render a list of items dynamically using map().
- Add event handling to toggle the visibility of the item list.

Day 2: Intermediate React Development

Topics:

- **Component Communication:**
  - Lifting State Up.
  - Parent-Child Communication via Props.
- **React Router Basics:**
  - Setting up React Router.
  - Creating Multi-Page Applications.
  - Navigation using Link and useNavigate (React Router v6+).
- **State Management:**
  - Introduction to Context API.

- Creating and Using Context for Global State.

Lab:

- Create a simple blog application:
  - **Pages:** Home, Blog List, Blog Details.
  - Use React Router for navigation.
  - Manage global state for the blog data using Context API.

Day 3: Advanced React and API Integration

Topics:

- **React Hooks:**
  - useState, useEffect, useRef, and useReducer.
  - Custom Hooks for Reusable Logic.
- **Data Fetching:**
  - Fetching Data from APIs using fetch and Axios.
  - Managing Loading, Success, and Error States.
- **Advanced State Management:**
  - Introduction to Redux Toolkit.
  - Configuring Redux Store and Slices.
- **Performance Optimization:**
  - Memoization (React.memo, useMemo, and useCallback).

Lab:

- Extend the blog app:
  - Fetch blog data from a public API (e.g., JSONPlaceholder).
  - Display a loader while fetching data.
  - Implement filtering and pagination for the blog list.

Day 4: React Authentication and Deployment

Topics:

- **Authentication with JWT:**
  - Implementing Login and Logout functionality.
  - Storing JWT securely (in memory or HttpOnly cookies).

- **Connecting React with ASP.NET Core API:**
    - Setting up CORS to enable cross-origin requests.
    - Consuming secured API endpoints.
- **React App Deployment:**
    - Building and Optimizing the React App for Production.
    - Deployment

Lab:

- Add a login page to the blog app:
    - Use JWT authentication to restrict access to certain features.
    - Save user session and display user-specific data.
- Deploy the app to a cloud platform

Day 5: ASP.NET Core 6 API Development and Integration

Topics:

- **Building a Web API:**
    - Creating an ASP.NET Core Web API Project.
    - Implementing CRUD Operations using Entity Framework Core.
    - Securing APIs with JWT Authentication.
- **Integration with React:**
    - Serving the React App from ASP.NET Core.
    - API Integration for Full-Stack Functionality.
- **Deployment:**
    - Deploying the Full-Stack Application (API + React)

Lab:

- Build a secured API for the blog app:
    - **Endpoints:** Blog CRUD, User Login, and Commenting.
    - Use Entity Framework Core with a SQL Server database.
    - Secure the API endpoints with JWT authentication.
- Integrate the API with the React app.
- Deploy the complete full-stack application.

Final Project

Title: Blog Management System

Features:

1. **Frontend (React):**
    - Multi-page blog app with routing (Home, Blog List, Blog Details, Admin Panel).
    - User authentication and role-based access (Admin vs. Regular User).
    - API Integration for blogs and comments.
2. **Backend (ASP.NET Core 6 API):**
    - User Authentication with JWT.
    - CRUD operations for blogs and comments.
    - Secure endpoints and database integration using EF Core.
3. **Deployment:**
    - Fully deployed application (React + ASP.NET Core API)