React Training Course

Beginner Level

Module 1: Introduction to Programming

- **Topics:**
    - What is Programming?
    - Basic Terminology: Code, Algorithms, Compiler, Interpreter, and IDE
    - Types of Programming Languages: Low-level vs. High-level
    - Understanding Computer Architecture

Module 2: Getting Started with React

- **Topics:**
    - What is React?
    - Why Use React?
    - Installing Node.js and npm
    - Introduction to Package Managers: npm vs. Yarn
    - Setting up a React Development Environment
    - Introduction to IDEs (Visual Studio Code)

Module 3: HTML & CSS Basics for React

- **Topics:**
    - What is HTML?

  - Basic HTML Tags and Structure
  - Introduction to CSS
  - Styling Techniques: Inline vs. External Styles
  - Responsive Design and Media Queries

Module 4: JavaScript Essentials for React

- **Topics:**
  - Variables, Constants, and Data Types
  - Functions and Parameters
  - Conditional Statements and Loops
  - Arrays and Objects
  - Introduction to ES6+ (let, const, arrow functions, template literals, destructuring, etc.)

Module 5: Introduction to React Components

- **Topics:**
  - What is a Component?
  - Functional vs. Class Components
  - JSX (JavaScript XML) Syntax
  - Props and State
  - Rendering Elements in React
  - Event Handling in React

Module 6: React State and Props in Depth

- **Topics:**
  - State in React: What is State and Why is it Important?
  - Setting and Updating State
  - Passing Data with Props
  - Handling Forms in React
  - Handling Events and User Input

Module 7: React Components and Component Lifecycle

- **Topics:**
  - Understanding Component Lifecycle
  - useState vs. useEffect
  - Functional Components Lifecycle Methods
  - Introduction to useRef and useReducer

Module 8: React Dev Tools and Debugging

- **Topics:**
  - Introduction to React DevTools
  - Inspecting React Components with DevTools
  - Debugging React with Console and Breakpoints
  - Performance Monitoring with React DevTools
  - Troubleshooting Common React Issues

Module 9: Introduction to Event Handling

- **Topics:**

- Handling Events in React (click, change, submit)
- Event Pooling in React
- Binding Event Handlers
- Event Propagation (bubbling and capturing)
- Synthetic Events in React

Mini-Project: Building a Simple To-Do App

- **Description:**
  Create a basic to-do list app where users can add, edit, and remove tasks. This project will cover all the basics: components, state, props, and event handling.

Intermediate Level

Module 1: Routing in React with React Router

- **Topics:**
  - What is React Router?
  - Setting up React Router
  - Creating Routes and Navigation Links
  - Dynamic Routing and Parameters
  - Nested Routes and Route Protection

Module 2: Managing State with Context API

- **Topics:**
  - Introduction to Context API
  - Creating a Context
  - Using useContext Hook
  - Sharing State Globally
  - Passing Functions via Context

Module 3: React Hooks: Beyond useState

- **Topics:**
  - useEffect Hook and Side Effects
  - useRef for Referencing DOM Elements
  - useReducer for Complex State Management
  - Custom Hooks
  - Performance Optimization with React.memo

Module 4: Styling React Components

- **Topics:**
  - CSS-in-JS: Styled-components
  - Inline Styling vs. External Styles
  - CSS Modules
  - Working with Preprocessors (Sass, Less)
  - Responsive Design and Flexbox/Grid

Module 5: Handling Data Fetching in React

- **Topics:**
  - Introduction to API Calls in React
  - Fetching Data with Fetch API and Axios
  - Handling Async/Await
  - Error Handling and Loading States
  - Working with External APIs (JSONPlaceholder, etc.)

Module 6: React Performance Optimization

- **Topics:**
  - Reconciliation and Virtual DOM
  - Memoization with React.memo and useMemo
  - Code Splitting and Lazy Loading
  - Performance Profiling in React

Module 7: Component Communication Patterns

- **Topics:**
  - Parent-Child Communication (Props)
  - Sibling Communication via Callbacks
  - Prop Drilling and How to Avoid It
  - Lifting State Up in React
  - Using Context API for Prop Drilling Alternative

Module 8: Error Boundaries in React

- **Topics:**

- What Are Error Boundaries?
- Catching JavaScript Errors in Component Tree
- Implementing Error Boundaries
- Handling Errors Gracefully in Production

Mini-Project: Building a Weather App

- **Description:**
  Create a weather app that fetches data from a weather API and displays it to the user. This project will cover React Router, API calls, state management, and handling loading/error states.

Advanced Level

Module 1: Advanced State Management with Redux

- **Topics:**
  - What is Redux?
  - Actions, Reducers, and Store
  - Using Redux in React
  - useSelector and useDispatch Hooks
  - Middleware in Redux (Thunk, Saga)
  - Managing Complex State with Redux Toolkit

Module 2: Server-Side Rendering with Next.js

- **Topics:**
  - What is Next.js?
  - SSR vs. Client-Side Rendering
  - Static Site Generation (SSG) and Incremental Static Regeneration (ISR)
  - Data Fetching in Next.js (getServerSideProps, getStaticProps)
  - Routing and Dynamic Routes in Next.js

Module 3: TypeScript with React

- **Topics:**
  - Introduction to TypeScript
  - Types, Interfaces, and Type Aliases
  - Using TypeScript with React Components
  - Typing Props, State, and Functions
  - Advanced TypeScript Concepts for React

Module 4: Testing React Applications

- **Topics:**
  - Introduction to Testing in React
  - Jest Testing Framework
  - React Testing Library
  - Writing Unit Tests and Integration Tests
  - Mocking API Calls and Components
  - End-to-End Testing with Cypress

Module 5: Deploying React Apps

- **Topics:**
    - Building a React App for Production
    - Deployment Options (Netlify, Vercel, GitHub Pages, AWS, etc.)
    - Continuous Integration and Continuous Deployment (CI/CD)
    - Using Docker for React App Deployment

Module 6: TypeScript and React Advanced

- **Topics:**
    - Understanding React with TypeScript
    - Typing Functional Components with Props and State
    - Generics in React with TypeScript
    - TypeScript with React Hooks
    - TypeScript in JSX and Advanced Patterns

Module 7: Advanced Routing with React Router

- **Topics:**
    - Route Guarding and Authorization
    - Programmatic Navigation and useHistory
    - Dynamic Routing with Nested Routes
    - Lazy Loading Routes
    - Creating Private Routes for Authentication

Module 8: GraphQL Integration with React

- **Topics:**
  - What is GraphQL?
  - Using Apollo Client for GraphQL Queries in React
  - Writing GraphQL Mutations and Queries
  - Optimizing Data Fetching with GraphQL
  - Pagination and Error Handling with GraphQL

Module 9: Performance Optimization in React

- **Topics:**
  - Analyzing React Performance Bottlenecks
  - Optimizing Renders with shouldComponentUpdate
  - Memoization and useMemo in React
  - Virtualization Techniques for Large Lists (React Virtualized)
  - Avoiding Reconciliation Pitfalls

Module 10: Working with WebSockets in React

- **Topics:**
  - What is WebSocket?
  - Creating Real-Time Applications with WebSockets
  - Integrating WebSockets into React
  - Building Real-Time Chat and Notifications
  - Handling WebSocket Events and Messages

Module 11: Advanced Form Handling

- **Topics:**
  - Complex Form Handling in React
  - Using Form Libraries (Formik, React Hook Form)
  - Validation with Yup
  - Dynamic Forms and Field Arrays
  - Handling Nested Objects and Arrays in Forms

Module 12: Progressive Web Apps (PWA) with React

- **Topics:**
  - Understanding PWAs and Service Workers
  - Implementing Offline-First React Apps
  - Creating a Manifest File
  - Caching Strategies for PWAs
  - Using Workbox for PWA Features

Mini-Project: Building an E-Commerce Website

- **Description:**
  Create a full-fledged e-commerce website with product listings, a shopping cart, and user authentication. This project will integrate advanced React concepts like Redux, Next.js, and TypeScript.

Final Projects (Comprehensive)

1. **Final Project 1: Portfolio Website with Blog (Beginner + Intermediate Concepts)**
   - **Description:**
     Create a portfolio website with sections for projects, about, and contact. Implement routing, state management with Context API, and a blog feature with

posts fetched from an API.

2. **Final Project 2: Task Management App with Real-time Updates (Intermediate + Advanced Concepts)**

   ○ **Description:**
   Build a task management application where users can create, edit, delete, and assign tasks. Use Redux for state management, implement real-time updates with WebSockets, and deploy the app to production.

3. **Final Project 3: Full-Stack E-Commerce App (All Concepts)**

   ○ **Description:**
   Build a complete e-commerce platform with product listings, shopping cart, user authentication, and payment processing. Integrate React with a Node.js/Express backend and MongoDB.