INT252:WEB APP DEVELOPMENT WITH REACTJS

L:2 T:0 P:2 Credits:3

Course Outcomes: Through this course students should be able to

CO1 :: Explain ES6+ JavaScript features and immutability concepts relevant to React applications.

CO2 :: Build reusable React functional components using JSX and props.

CO3 :: Apply React hooks to manage component state and user interactions.

CO4 :: Demonstrate form handling in React using client-side validation.

CO5 :: Use client-side routing and API-based data fetching in React applications.

CO6 :: Use shared state management techniques in React applications.

Unit I

Refreshing JavaScript : Arrow Functions, block-scoped variables using let and const, array iteration and transformation methods, destructuring assignments, spread and rest operators, ES modules, immutability principles, functional programming concepts relevant to React, shallow and deep copying mechanisms, referential equality and state comparison

React Fundamentals and Tooling : Single Page Applications and Multi Page Applications, React philosophy and declarative UI paradigm, component-based architecture, setting up React applications using Vite, modern React development environment and tooling, feature-based project folder structure and best practices, JSX syntax and expressions, rendering components into the DOM, functional components and render lifecycle, introduction to Virtual DOM and reconciliation process

Unit II

Component Design : Functional components, props-driven component reusability, component composition patterns, reusable component abstractions, conditional rendering strategies, lifting state up in component hierarchies, defining component responsibility boundaries

Styling in React : Styling approaches in React applications, Tailwind CSS fundamentals, utility-first styling methodology, responsive design using Tailwind breakpoints, reusable UI styling patterns, accessibility fundamentals, semantic HTML practices in React

Unit III

States and Events : Event handling mechanisms in React, local state management using useState, state update behavior and batching, re-render triggers and optimization awareness, derived state concepts, state management best practices

Hooks and Side Effects : Built-in hooks including useState, useEffect, and useContext, managing side effects using useEffect, dependency array semantics and effect cleanup, performance optimization using useCallback and useMemo, introduction to custom hooks for logic reuse, rules of hooks, common pitfalls and anti-patterns

Unit IV

Form Handling : Controlled and uncontrolled components, form state management patterns, handling form submission workflows, reusable form component design, conditional rendering in form interfaces, accessibility considerations in form handling

Validation : Client-side form validation concepts, commonly used validation patterns, schema-based validation principles, form validation using Zod, integrating schema validation with form logic and error handling

Unit V

Client Server Communication : HTTP fundamentals and REST principles, Fetch API and Axios usage patterns, handling loading, error, and empty states, side effects for data fetching, distinction between client state and server state, limitations of useEffect for data synchronization, introduction to server state management using React Query, caching strategies and background synchronization concepts

Client-Side Routing and Navigation : Client-side routing fundamentals, routing configuration setup, navigation between application views, dynamic routes and route parameters, query parameter handling, nested routes and layout routes, protected routes, route-based code splitting

Unit VI

Unit VI

Global State Management : State sharing challenges in React applications, Context API for global state management, Redux Toolkit fundamentals, Redux data flow and slice architecture, dispatching actions and using selectors, overview of async logic handling, lightweight global state management using Zustand, comparison of local state, global state, and server state, guidelines for selecting appropriate state management solutions

Application Debugging and Production Deployment : Debugging React applications, React Developer Tools and Profiler usage, identifying rendering and performance issues, build optimization and production readiness techniques, environment configuration using environment variables, deployment workflow overview, introduction to modern deployment platforms, best practices for deploying production-ready React applications

List of Practicals / Experiments:

List of Practicals

- Implement JavaScript utility functions using arrow functions, array methods, destructuring, and immutability principles.
- Set up a React application using Vite and explore project structure, JSX syntax, and functional component rendering.
- Develop reusable functional components demonstrating props usage, component composition, and conditional rendering.
- Design a responsive user interface using Tailwind CSS utility classes and reusable styling patterns.
- Implement event handling and local state management using useState and analyze component re-render behavior.
- Manage side effects using useEffect and demonstrate effect cleanup and dependency handling.
- Optimize component performance using useCallback, useMemo, and React.memo.
- Create and use a custom hook to encapsulate and reuse component logic
- Build controlled and uncontrolled form components with proper form submission handling.
- Implement schema-based form validation using Zod and display validation errors dynamically.
- Fetch and display server data using Fetch API or Axios with proper loading and error state handling.
- Manage server state using React Query, demonstrating caching and background data synchronization.
- Implement client-side routing with dynamic routes, nested routes, and protected routes.
- Develop a React application using global state management with Context or Zustand and deploy the production build.

Text Books:

1. REACT.JS DESIGN PATTERNS by ANTHONY ONYEKACHUKWU OKONTA, BPB PUBLICATIONS

References:

1. REACT: UP & RUNNING by STOYAN STEFANOV, O'REILLY