

React App to Progressive Web Apps (PWA)

Prerequisites: Working Knowledge of React Application

Day 1: Introduction to React and PWA Fundamentals

Morning Session: React Basics Refresher

- **Lecture:**
 - React components, state, and props
 - React Router for single-page applications
 - Basic setup of a React application using Create React App (CRA)
- **Lab:**
 - Create a basic React app: A simple To-Do list
 - Implement React Router for navigation

Afternoon Session: Introduction to PWAs

- **Lecture:**
 - What are PWAs? Characteristics and benefits
 - Core principles: Responsiveness, Connectivity Independence, and App-like Experience
 - Key technologies: Service Workers, Web App Manifest, and HTTPS
 - **Lab:**
 - Examine an existing PWA (e.g., Starbucks PWA or Twitter Lite)
 - Identify features and analyze Lighthouse scores using Chrome DevTools
-

Day 2: PWA Architecture and React Integration

Morning Session: Web App Manifest

- **Lecture:**
 - Role of Web App Manifest in PWAs
 - Key properties: name, short_name, icons, start_url, display, theme_color, and background_color
- **Lab:**
 - Add a Web App Manifest to the React app
 - Customize the app's icons, theme color, and display mode

Afternoon Session: Service Workers

- **Lecture:**

- Introduction to Service Workers: Lifecycle, caching, and offline capabilities
 - Tools like Workbox for easier Service Worker implementation
 - **Lab:**
 - Register a Service Worker in the React app
 - Use Workbox to precache assets and enable offline support
 - Test offline functionality using Chrome DevTools
-

Day 3: Advanced Features and Optimization

Morning Session: Advanced PWA Features

- **Lecture:**
 - Push Notifications: Setting up notifications with Firebase
 - Background Sync and Periodic Sync APIs
 - Performance optimization: Lazy loading, image optimization, and code splitting
- **Lab:**
 - Implement Push Notifications
 - Use React.lazy and Suspense for code splitting in the app

Afternoon Session: Testing, Deployment, and Debugging

- **Lecture:**
 - Testing PWAs with Lighthouse for performance, accessibility, and best practices
 - Deploying a PWA
 - Common debugging techniques and troubleshooting issues
 - **Lab:**
 - Run Lighthouse audits and fix identified issues
 - Deploy the PWA to a hosting platform
 - Test the PWA on different devices (mobile and desktop)
-