CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

React Apps with Next.js

Duration

5 Days

Description

This comprehensive course offers an in-depth exploration of Next.js, a powerful React framework. Beginning with the basics, you'll understand why Next.js is a preferred choice for React developers. The course guides you through setting up a development environment, creating a Next.js project, and delving into React component fundamentals, including the use of React Hooks and custom hooks. You'll gain practical experience in implementing routes with App Router, fetching data, and understanding isomorphic rendering. The course also covers advanced topics like Next.js caching, styling techniques, and unit testing for React components. Finally, you'll learn how to deploy your Next.js apps, ensuring a well-rounded understanding of both development and deployment processes in the Next.js ecosystem.

Objectives

- Learn why Next.js is a React Framework
- Configure a Development Environment for Next.js
- Create a Next.js Project
- Learn and Apply React Component Fundamentals
- Utilize React Hooks and Custom Hooks
- Implement Routes with App Router
- Fetch Data with Next.js
- Isomorphic Rendering
- Explore Next.js Caching
- Apply Styling to Next.js Apps
- Employ Unit Testing to Test React Components
- Learn How to Deploy Next.js Apps

Prerequisites

All attendees must have experience with modern JavaScript or TypeScript, including the new language features like classes, modules, arrow functions, and destructuring.

CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Node.js and Visual Studio Code on their computers. Also, students will need permission to install NPM Packages and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Outline

- Introduction
- React and Next.js Overview
  - Why React and Next.js?
  - What problem does React solve?
  - What problem does Next.js solve?
- Development Environment
  - Install Node.js
  - Configure Visual Studio code
  - Install React Developer Tools
  - Next.js CLI
- Next.js Project Setup
  - Create a new project
  - Folder Structure
  - Browser Support
  - Styles and Assets
  - Dependencies
- React Components
  - Creating an Element
  - Create a Function Component
  - Rendering a Component
  - Composing & Reuse
- React Component Rendering and JSX
  - What problem does JSX solve?

  - Embedding Expressions
  - Specifying Attributes
  - Using Fragments
  - Virtual DOM and Fiber Nodes
  - Ternary Operator (?)
  - Logical (&& and ||) Operators
  - Rendering a list of data
  - Optimizing rendering with keys
- React Component Props
  - Immutable Props
  - String Literals vs. Expressions
  - Prop Types
  - Default Prop Values
  - Naming Patterns for Props
- React Component Events
  - What are Events?
  - Common Events in React: Click and Change
  - Event Handlers and Functional Component
  - Passing Event Handlers via Props
- React Component Hooks
  - What is Component State?
  - State Hook
  - Effect Hook
  - Callback Hook
  - Custom Hooks
- Capture Data with Forms
  - Controlled and Uncontrolled Components
  - Enable Change Logic across Multiple Form Controls
  - Wiring up Input, Textarea, and Select
  - Handling different types of Input
- React Component Architecture
  - Reusable Components
  - Component Communication
  - Design Patterns
  - Container and Presentation Components

  - Defining Prop Drilling
- App Router
  - Define Routes
  - Pages and Layouts
  - Linking and Navigating
  - Dynamic Routes
  - Error Handling
- Fetching Data
  - Server-Side Data Fetching
  - Client-Side Data Fetching
  - Data Fetching Patterns
  - Connecting Forms to Data Fetching
  - Server-Only Forms
  - Form Validation
  - Updating Cookies
- Isomorphic Rendering
  - Server Rendering
  - Client Rendering
  - Server and Client Composition Patterns
- Caching
- Styling
  - CSS Modules
  - Tailwind CSS
  - CSS-in-JS (Styled Components)
  - Sass
  - Slice and Dice a Graphic Design File
  - Responsive Design
- Unit Testing Overview
  - Jest and Testing Library
  - What are React components tested for?
  - Tests, Test Suites, Assertions, and Mocking
  - Test DOM rendering
  - Test Event Handlers with Spies
  - Test Custom Hooks
  - Mocking Components

  - Mocking Hooks
- Deployment
- Conclusion