REACT Foundations Training

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Duration: **2 Day**

React Foundations Training provides participants with the essential knowledge and practical skills required to develop modern and dynamic user interfaces using React.

Throughout the course, participants will explore the core concepts of React, beginning with development environment setup and progressing through JSX, component architecture, props, and state management. The course also covers React Hooks, data flow between components, form handling, styling approaches, side-effect management, and application debugging techniques.

Supported by hands-on exercises and practical labs, the training enables participants to build reusable components, manage application data and state effectively, and develop maintainable React-based web applications.

Prerequisites

This course is designed for participants who want to develop practical React skills and already have a foundational understanding of web development.

Participants are expected to have basic to intermediate knowledge of:

- JavaScript
- HTML
- CSS

Participants without sufficient JavaScript experience are advised to complete a JavaScript Fundamentals-level course before attending this training.

Who Should Attend

- Front-End Developers
- Web Developers
- Software Engineers
- Technical professionals working with web application development

REACT Foundations Training

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Duration: **2 Day**

What You Will Learn

Upon completing this course, participants will be able to:

- Configure a React development environment and establish a basic application structure.
- Create dynamic user interfaces using JSX.
- Develop reusable React components and apply component composition techniques.
- Transfer data and event handlers between components using props.
- Manage component state and user interactions with the useState Hook.
- Apply React's one-way data flow principles.
- Organize applications using Container and Presentational Component patterns.
- Implement component-level styling using CSS and CSS Modules.
- Manage side effects and component lifecycle requirements with the useEffect Hook.
- Develop React forms using both Controlled and Uncontrolled approaches.
- Use React Developer Tools to inspect components, props, and state and troubleshoot application issues.

Training Outline

1. Introduction to React

React Fundamentals

- What is React?
- Evolution and common use cases of React
- Key characteristics that distinguish React from other web development approaches
- Benefits of using React
- Component-based user interface architecture

Virtual DOM

- Understanding the Virtual DOM
- How React uses the Virtual DOM
- Basic differences between Virtual DOM and traditional DOM operations

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Importance of the Virtual DOM in modern web applications

Setting Up a React Development Environment

- Preparing the development environment
- Creating a starter project with Create React App
- Overview of React development tools
- Introduction to React Developer Tools

Project Structure

- Creating the first React application
- Understanding the basic folder and file structure
- Organizing application components
- File and component naming approaches

2. JSX

Getting Started with JSX

- HTML entry point and root element
- Working with createRoot and render
- Building a basic React component with JSX
- Understanding relationships between application files

JSX Fundamentals

- What is JSX and why is it used?
- Core JSX syntax rules
- Parent element requirements
- Element and tag closing rules

Writing JSX

- Basic JSX syntax
- Working with attributes
- Embedding JavaScript expressions with {}

JSX Usage Patterns

- Using variables within JSX
- Creating dynamic content with props
- Fundamentals of conditional rendering

3. React Components

Component Fundamentals

- Understanding the component concept
- Creating the first React component
- Writing single-line and multi-line JSX
- Importing and exporting components
- Rendering components

Logic Within Components

- Using JavaScript logic inside components
- Implementing conditional logic
- Handling user interactions

Component Composition

- Nesting components
- Building reusable component structures
- Organizing application components
- Applying component composition principles

4. Props

Props Fundamentals

- Understanding props
- Passing data into components
- Receiving and using prop values
- Prop type and default value approaches
- Working with the children prop
- Rendering prop values

- Passing props between components
- Dynamic rendering with props

Advanced Prop Patterns

- Passing event handlers as props
- Receiving and using event handler props
- Event handler naming conventions

5. State and Event Management

Understanding State

- What is state?
- Understanding stateful components
- When state should be used
- When state should be avoided
- Fundamentals of component re-rendering

useState Hook

- Introduction to React Hooks
- Initializing state
- Updating state values
- Managing multiple state values
- Managing user interactions through state

Practical Review of Core Concepts

- Creating components
- Working with props
- Managing state
- Combining components and state in practical scenarios

6. Component Design Patterns

Component Architecture

- Container Component approach

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Presentational Component approach
- Determining when components should be separated
- Rendering presentational components within container components

React Data Flow

- Understanding one-way data flow
- Parent–Child component communication
- Child–Parent communication scenarios
- Sharing data between sibling components
- Understanding prop drilling
- Applying the Lifting State Up pattern

7. Styling React Applications

- Using standard CSS with className
- Introduction to CSS Modules
- Organizing application styles
- Component-specific styling
- Building maintainable styling structures for React projects

8. Effects and Component Lifecycle

Working with Effects

- Understanding side effects in React
- Why side effects need to be managed
- Basic useEffect Hook syntax and usage
- Understanding when effects execute
- Managing effect dependencies
- Understanding effect cleanup
- Creating cleanup functions

Component Lifecycle

- Component initialization

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Component updates
- Component cleanup
- Managing lifecycle requirements with useEffect

9. React Forms

Form Fundamentals

- Differences between standard HTML forms and React forms
- Controlled and Uncontrolled Form approaches
- Selecting the appropriate form management approach

Uncontrolled Forms

- Using the useRef Hook with forms
- Accessing input values
- Creating an Uncontrolled Form
- Handling submit events
- Retrieving form values
- Common use cases for Uncontrolled Forms

Controlled Forms

- Working with standard form elements
- Managing input values with useState
- Managing multiple form fields
- Working with different input types
- Handling form submissions
- Controlling default browser behavior
- Collecting form data
- Basic form validation
- Managing success and error states

10. Debugging React Applications

React Developer Tools

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Working with the Components tab
- Inspecting props and state
- Navigating the component tree
- Analyzing component behavior

Application Debugging

- Identifying common React errors
- Troubleshooting component-related issues
- Analyzing props and state problems
- Applying fundamental debugging techniques in React applications