React
Practice Assignment

Techademy

CONTENTS

1 Skill Description 2
2 Objectives 2
3 Duration 2
4 Problem Statement 2
5 Learning Outcome 3

1 SKILL DESCRIPTION

This assignment is designed to advance your skills in React development, focusing on topics such as composable components, multiple components interaction, data management from in-memory and backend server, form handling using Formik and Yup, higher-order components, render props, Context API, React Router, and routing transition concepts. Participants will engage in hands-on exercises to strengthen their understanding and practical application of these concepts.

2 OBJECTIVES

The objective of this 2-hour assignment is to equip participants with advanced skills in React application development. Participants will learn to create composable components, manage data from in-memory and backend servers, handle forms effectively using Formik and Yup, utilize higher-order components and render props for component reusability, leverage the Context API for state management, implement navigation using React Router, and understand routing transition concepts.

3 DURATION

2 hours

4 PROBLEM STATEMENT

Task 1: Composable Components

- Define the concept of composable components in React.
- Explain the benefits of creating modular and reusable UI elements.

Task 2: Multiple Components Interaction

- Demonstrate various methods for components to interact with each other.
- Provide examples of scenarios where multiple components need to communicate.

Task 4: React Application with Backend Server

- Introduce the json-server and explain its role in simulating a backend server for a React application.
- Demonstrate how to interface a React application with a backend server.

Task 5: Form Handling with Formik and Yup

- Explain the concepts of Formik and Yup for handling forms in React.
- Guide participants through practical examples of form implementation.

Task 6: Higher Order Components

- Define higher-order components and their role in code reuse.
- Demonstrate the implementation of higher-order components in React.

Task 7: Render Props

- Explain the concept of render props in React.
- Provide examples and use cases for implementing render props.

Task 8: Context API

- Introduce the Context API for state management in React.
- Demonstrate how to use the Context API in complex applications.

Task 9: React Router

- Provide an overview of React Router for implementing navigation in React applications.
- Demonstrate the creation of routes and navigation components.

Task 10: Routing Transition Concepts

- Explain routing transition concepts, including transition into and transition from.
- Practical exercises on implementing routing transitions.

5 LEARNING OUTCOME

- Create composable components for building modular and reusable UI elements.
- Implement communication between multiple components for seamless interaction.
- Read and manage data from in-memory storage within a React application.
- Integrate a React application with a backend server using json-server.
- Effectively handle forms in React applications using Formik and Yup for validation.
- Understand and utilize higher-order components for code reuse.
- Implement render props for flexible and dynamic component composition.
- Leverage the Context API for state management in complex applications.
- Implement navigation in React applications using React Router.
- Understand routing transition concepts, including transition into and transition from.