

Assignment: Create an Interactive Landing Page with React Hooks, Context, and Routing

Context:

You are tasked with building a modern, interactive landing page for a website. This landing page should include common sections such as Home, About, and Contact Us. Your goal is to create an attractive and user-friendly interface that incorporates state management, context for theme toggling, form handling with validations, and React routing to navigate between different sections.

Objective:

Develop a landing page with a styled navigation bar, incorporating interactive features such as state management with `useState`, effects with `useEffect`, context with `useContext`, form handling with validations, and routing with React Router.

Requirements:

Navigation Bar:

1. **Design** a responsive and attractive navigation bar.
2. **Create** a `NavBar` component with links to Home, About, and Contact Us pages.
3. **Include** links to different sections of the landing page (Home, About, Contact Us) using React Router.
4. **Style** the navigation bar using style components and Material UI.

State Management:

1. **Use `useState`** to manage form inputs and a theme toggle feature.
2. **In the Home component**, use `useState` to manage a welcome message that updates based on user input.

Side Effects:

1. **Use `useEffect`** to handle simple side effects such as setting the document title or logging to the console when a component mounts or updates.

Scenario 1: `useEffect` with an Empty Dependency Array

- **Initialize Local State:** Set initial values for local state variables, such as setting a welcome message based on the time of day.

Scenario 2: useEffect without a Dependency Array

- **Animate Elements on Render:** Trigger animations for specific elements each time the component re-renders.
- **Update Real-Time Data:** Update a clock or countdown timer displayed on the landing page every second.

Scenario 3: useEffect with Dependencies

- **Form Validation Feedback:** Show real-time validation feedback as the user types in a contact form.
- **Update Local Storage:** Save user preferences or input (e.g., theme, form data) to local storage whenever they change.
- **Toggle Class Based on State:** Apply dynamic styles or classes based on a state variable, such as toggling a CSS class for a dark/light theme.

Context:

1. **Create a ThemeContext** to manage and toggle between light and dark themes across the landing page.
2. **Create a ThemeContext** with useState to manage the theme.
3. **Provide** the ThemeContext to the entire application.
4. **Add** a toggle button in the NavBar to switch between light and dark themes.

Form Handling:

1. **Create a ContactForm component.**
2. **Create** a contact form with validation (name, email, message).
3. **Use useState** to manage form inputs.
4. **Implement** validation logic (e.g., required fields, email format).
5. **Show** error messages for invalid inputs and success messages upon form submission.

React Routing:

1. **Implement routing** to navigate between different sections (Home, About, Contact Us) using BrowserRouter, Routes, Route, and NavLink from react-router-dom.
2. **Set up nested routes** for additional sections within About (e.g., History, Team, Values).
3. **Use route parameters** in the Contact Us page to display a specific contact detail based on the URL parameter.
4. **Ensure** that the navigation links in the NavBar update the URL and display the corresponding components.

Additional Requirements for Page UIs:

Home Page:

- **Design** an engaging landing page with a welcoming message, brief introduction, and a call-to-action button.
- **Example:** A hero section with a background image, a heading, a subheading, and a call-to-action button.

About Page:

- **Design** a professional and informative About Us page with sections such as the company's history, mission, team members, and values.
- **Example:** A section with an image and text side by side, team member profiles with pictures and descriptions.

Contact Us Page:

- **Design** a user-friendly Contact Us page with a contact form and additional contact information.
- **Example:** A contact form with fields for name, email, and message, followed by a section with the company's address, phone number, and email.