# Exercise: Build a Multi-Feature Blogging Application

## Objective:

Create a blogging application where users can create, edit, delete, and view blog posts. The application should use React concepts like Redux, React Router, useState, useReducer, useEffect, useRef, and Axios.

## Requirements:

**Homepage:**

- Display a list of blog posts with a summary of each post.
- Each post should have a "Read More" button that navigates to the detailed view of the post.

**Post Detail Page:**

- Display the full content of the blog post.
- Provide options to edit or delete the post.

**Create/Edit Post Page:**

- Include a form to create a new blog post or edit an existing one.
- The form should include fields for the title, content, and tags.
- Use useRef to focus on the title input when the page loads.
- Manage form state with useState and handle form submission using useEffect.

**Global State Management (Redux):**

- Use Redux to manage the state of the blog posts across the application.
- Include actions for adding, editing, and deleting posts.

**Navigation (React Router):**

- Implement routing to navigate between the homepage, post detail page, and create/edit post page.
- Use React Router to handle the navigation and URL parameters.

**API Integration (Axios):**

- Fetch the list of blog posts from a mock API when the application loads (use useEffect).
- Use Axios to handle the API requests for CRUD operations (Create, Read, Update, Delete).

**Custom Hooks:**

- Implement a custom hook that encapsulates the logic for form validation and can be reused across different forms in the application.

**Side Effects (useEffect):**

- Use useEffect to handle side effects like fetching data, updating the document title with the post title on the Post Detail page, and saving form data.

**Complex State Management (useReducer):**

- Implement useReducer to manage complex state transitions within the form, such as handling form input changes, validation errors, and submission status.