MOBILE PROGRAMMING

Tutorial 10

Activity 01: State Management with Redux in React Native

Create a React Native application that uses Redux to manage a counter state.

The user should be able to:

- Press the `Increment` button to increase the counter value by 1.
- Press the `Decrement` button to decrease the counter value by 1.
- View the `current counter value` displayed in the center of the screen.

Before you start implementing this activity, you need to install required libraries

```
npm install @reduxjs/toolkit react-redux
```

Project Structure

- Inside root app, create the following folders:
  - `redux/`: To store Redux-related files.
  - `components/`: To store React components.

Setting Up the Redux Store

Create a file named `./redux/store.js` to initialize the Redux store:

- Use `configureStore` from `@reduxjs/toolkit`.
- Integrate the counter reducer from the counter slice.

Create a file named `./redux/counterSlice.js`.

- Use `createSlice` from `@reduxjs/toolkit.`
- Define the initial state as `{ count: 0 }.`
- Create two actions: `increment` and `decrement`.

Update the `App.js` file.

- Use `Provider` from `react-redux` to wrap the entire application.
- Use the store you created.

Create a file named `./components/Counter.js` for User Interface

- Use `useSelector` to access the count value from the Redux store.
- Use `useDispatch` to trigger the increment and decrement actions.
- Design the UI with:
    - A title: "`Counter`".
    - Display the current count.
    - Two buttons: "`Increment (+)`" and "`Decrement (-)`".

Activity 02: Upgrade Product Management App using Redux

In this activity, you will modify an existing React Native application that implements a Product Management App with navigation. The goal is to refactor the app to use Redux for state management.

The existing application consists of two screens:

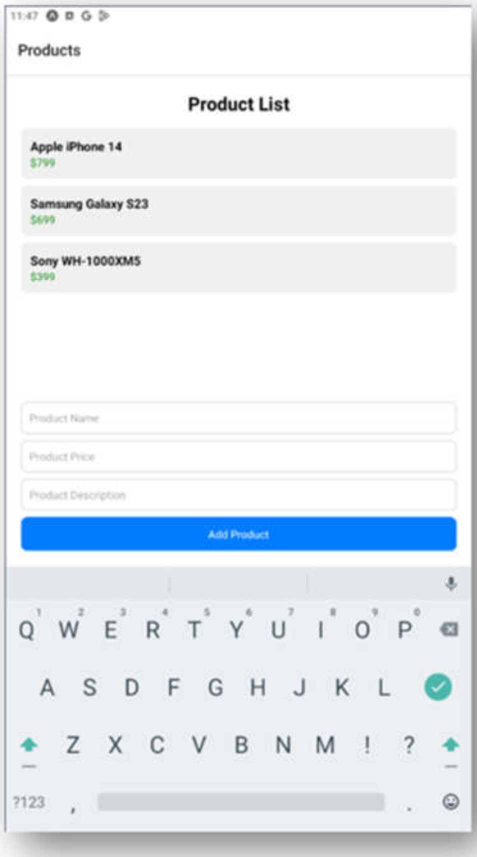
1. **Product List Screen:** Displays a list of products.
2. **Product Detail Screen:** Shows detailed information about a selected product.

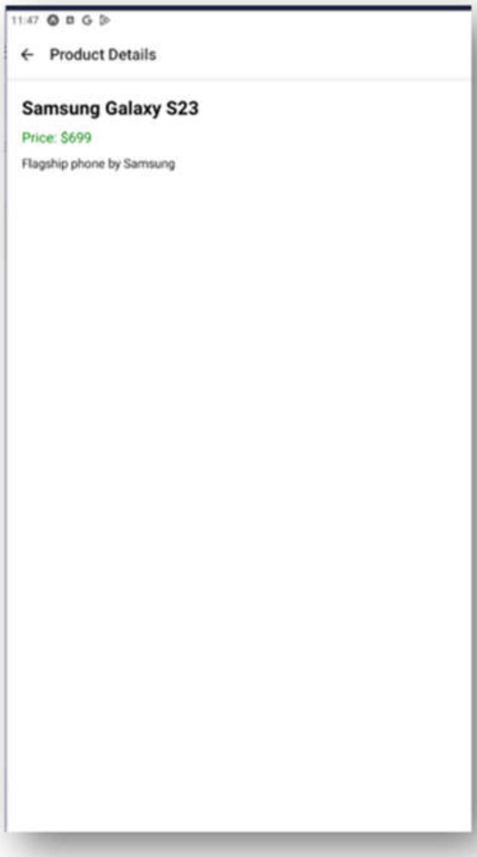
Currently, the state management is done using local component state. You will upgrade the app to manage the product list using Redux.

Objectives:

- Refactor the application to use Redux for managing the product list.
- Implement actions and reducers to handle product addition, update, and removal.

- Maintain the application's functionality and navigation after refactoring.


11:47
Products
Product List
Apple iPhone 14
$799
Samsung Galaxy S23
$699
Sony WH-1000XM5
$399
Product Name
Product Price
Product Description
Add Product


11:47
Product Details
Samsung Galaxy S23
Price: $699
Flagship phone by Samsung