MOBILE PROGRAMMING

Tutorial 05

Activity 01: Implementing a basic Stack Navigator in React Native

Create a simple React Native app using `@react-navigation/native-stack` to navigate between two screens: `Home` and `Details`.

Requirements:

1. Install the necessary dependencies:

```
npm install @react-navigation/native-stack
```

2. Implement the navigation between two screens:
    - **Home** screen: Displays a button that navigates to the Details screen.
    - **Details** screen**:** Displays a button to go back to the Home screen.
3. Customize the screen title using the options property.
4. Set the Home screen as the initial route.

Activity 02: Implement a Product Management App with Navigation in React Native

Create a React Native application with two screens:

1. A **Product List** screen that displays a list of products.
2. A **Product Detail** screen that shows detailed information about a selected product.

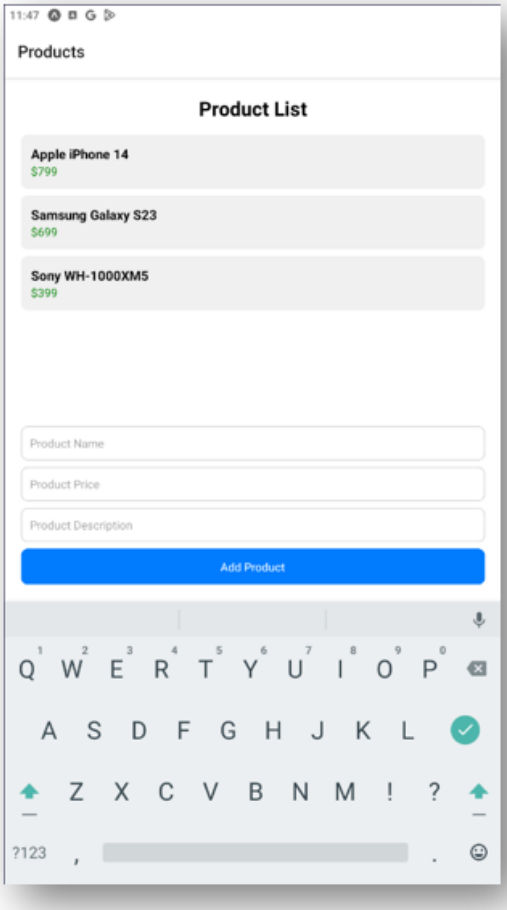
You need to implement the following functionalities:

Requirements:

1. **Product List** screen**:**
    - Display a list of products using FlatList.
    - Each product should show the name and price.
    - When a user taps on a product, navigate to the **Product Detail** screen and pass the selected product's details to it.
    - Include a form to add new products with fields: name, price, and description.

- When the "Add Product" button is pressed, the new product should be added to the list.

2. **Product Detail** screen**:**
   - This screen should display the name, price, and description of the selected product.
3. **Navigation:**
   - Use React Navigation to navigate between the two screens (Product List and Product Detail).
   - The **Product List** screen should allow the user to go to the **Product Detail** screen by tapping on a product.
4. **Styling:**
   - Style the application to make it user-friendly, with clear distinctions between different sections (product list, form, etc.).



Activity 03: Refactor Product Management App to add a separate Add Product screen

Refactor your existing Product Management App by splitting the **Add Product** functionality into a new screen and implementing navigation to and from that screen.

Requirements:

1. **Create a new Screen for adding Products:**

- Move the **Add Product** form from the main screen to a new screen called **Add Product.**
- The **Add Product** screen should have:
    - Text fields for **name**, **price**, and **description** of the product.
    - A button to submit the new product.

2. **Navigation:**

- Set up navigation between the following screens using React Navigation:
    - **Product List** screen (displays all products).
    - **Add Product** screen (to add a new product).
    - When the user adds a product, navigate back to the **Product List** screen.

3. **Passing data:**

- Once the product is added, ensure that it appears in the **Product List** screen.
- Use state management to persist the list of products (you can use `useState` for local state).

Submission

Compress your relevant project files (`App.js`, components, `assets` folder, etc.) into `zip` files (1 `zip` file per activity) and put all `zip` files into a single `zip` file.

Submit the final `zip` file to this tutorial's submission box on FIT LMS.