MOBILE PROGRAMMING

Tutorial 03

Activity 01: Display Product List with Scroll and FlatList

Objective: Create a React Native app that displays a default list of products using a FlatList.

1. Create a default list of products. Each product should have the following fields:
    - **Name**: The name of the product (e.g., "Apple iPhone 14").
    - **Price**: The price of the product in USD (e.g., "$799").
    - **Description**: A short description of the product (e.g., "Smartphone by Apple").

```
[

    {
      id: "1",
      name: "Apple iPhone 14",
      price: "799",
      description: "Smartphone by Apple",
    },
    {
      id: "2",
      name: "Samsung Galaxy S23",
      price: "699",
      description: "Flagship phone by Samsung",
    },
    {
      id: "3",
      name: "Sony WH-1000XM5",
      price: "399",
      description: "Noise-canceling headphones",
    },
];
```

2. Use a FlatList to display the product list.

3. Each product item should display:
    - Product Name (bold text).
    - Product Price (e.g., "$799").
    - Product Description (small text).

Expected Output:

- When the app starts, a scrollable list of products is displayed.

Activity 02: Add New Products to the List

Objective: Extend the app to allow users to add new products to the list.

1. Add a form at the bottom of the screen with the following input fields:
    - **Product Name**: A text input field for the product name.
    - **Product Price**: A numeric input field for the product price.
    - **Product Description**: A text input field for the product description (optional).
2. Add a button labeled **"Add Product"** below the input fields.
3. When the button is clicked:
    - Validate that the **Product Name** and **Product Price** fields are not empty.
    - Add the new product to the list.
    - Clear the input fields after adding the product.

Expected Output:

- Users can type in the form and click **"Add Product"** to add new items to the product list.
- The list updates immediately to include the new product.

Activity 03: Handle Keyboard and Styling

Objective: Improve the app's user experience by handling the keyboard and adding styling.

1. **Keyboard Handling**:
    - Use KeyboardAvoidingView to ensure the keyboard does not cover the input fields when typing.
2. **Styling**:
    - Add a title at the top: **"Product Management App"**.
    - Style the FlatList items with the following:
        - Rounded corners.
        - Light background color.
        - Padding for spacing.
    - Style the input fields and button:
        - Add borders and padding for the input fields.
        - Use a button with a clear label and proper spacing.

Expected Output:

- The app should maintain a clean layout when the keyboard is open.
- The app should look visually appealing with consistent styling.

Submission

Submit a zip file containing all Java programs to this tutorial's submission box in the course website on FIT Portal.