Cross Platform App Development Lab Experiment No. 5

Aim: Adding user interactivity in the base components and customization of layout using functions provided by React Native and XAML.

Objectives:

1. Implement user interactivity in React Native components.
2. Customize layout using functions provided by React Native and XAML.

Theory:

- React Native Components:
  - Building blocks for mobile app development.
  - Includes interactive elements like buttons, input fields, etc.
  - Can be extended with user-defined logic.
- Layout Customization with Functions:
  - React Native provides functions for dynamic layout customization.
  - XAML offers layout customization through specific functions and attributes.
  - These functions enable responsiveness and adaptability.

Requirements:

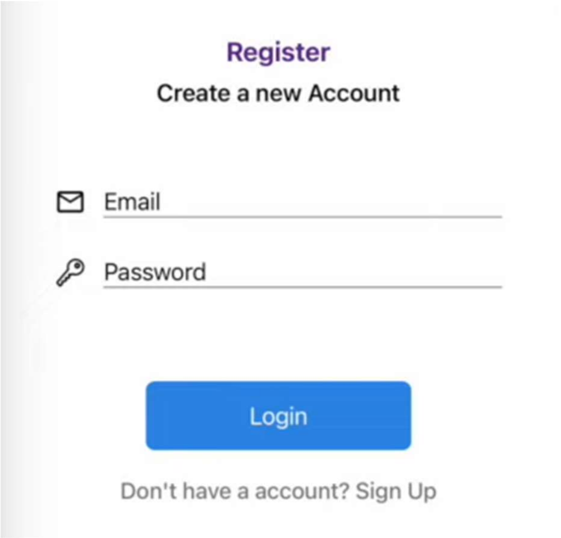
- React Native development environment set up.
- A text editor (e.g., Visual Studio Code).
- Understanding of JavaScript and React Native concepts.

Tools:

- React Native development environment.
- Text editor (e.g., Visual Studio Code).

Implementation/Code:-

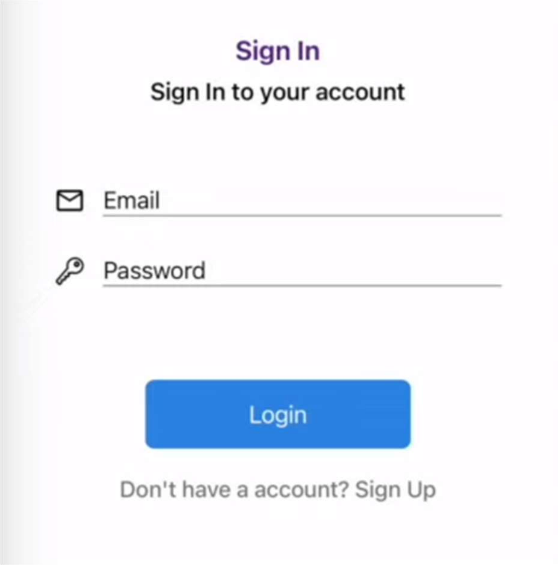
- User can register



```
<Pressable
        onPress={register}
          style={{
            width: 200,
            backgroundColor: "#318CE7",
            padding: 15,
            borderRadius: 7,
            marginTop: 50,
            marginLeft: "auto",
            marginRight: "auto",
          }}
        >
          <Text style={{ fontSize: 18, textAlign: "center", color: "white" }}>
            Register
          </Text>
        </Pressable>

        <Pressable onPress={() => navigation.goBack()} style={{ marginTop: 20
}}>
          <Text
            style={{
              textAlign: "center",
              fontSize: 17,
              color: "gray",
              fontWeight: "500",
            }}
          >
            Already have a account? Sign in
          </Text>
        </Pressable>
```

- User can sign in if they are already registered



Sign In

Sign In to your account

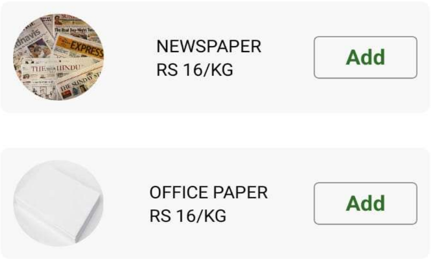
Don't have a account? Sign Up

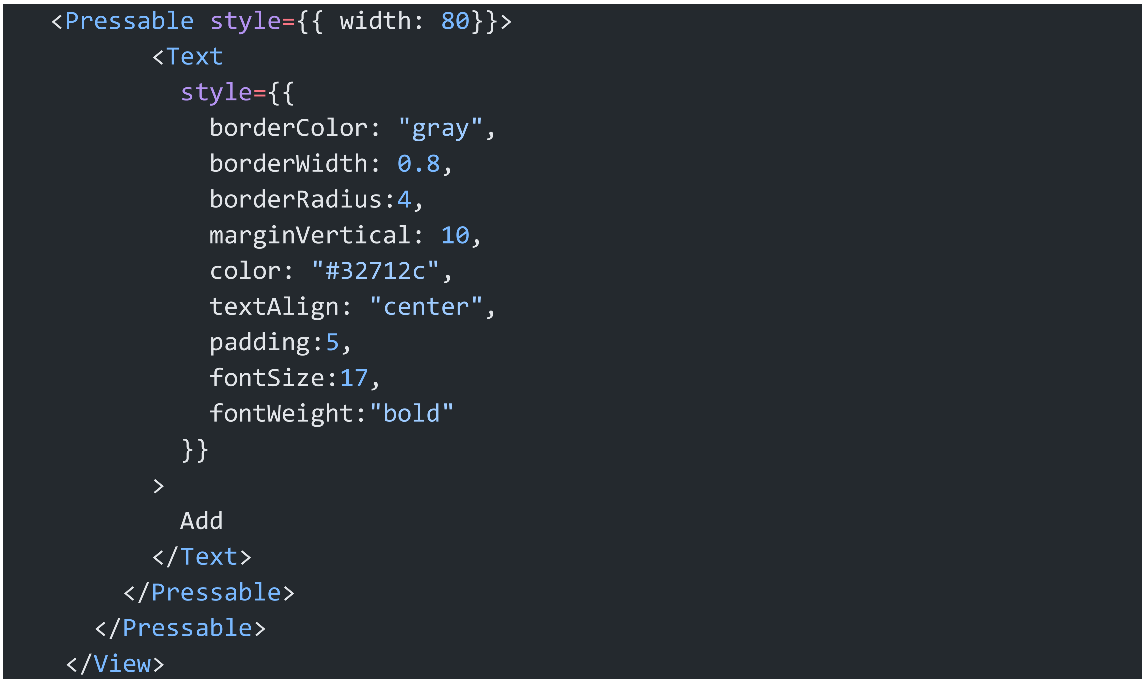
```
  <Pressable
     onPress={login}
       style={{
         width: 200,
         backgroundColor: "#318CE7",
         padding: 15,
         borderRadius: 7,
         marginTop: 50,
         marginLeft: "auto",
         marginRight: "auto",
       }}
     >
       <Text style={{ fontSize: 18, textAlign: "center", color: "white" }}>
         Login
       </Text>
     </Pressable>

     <Pressable onPress={() => navigation.navigate("Register")} style={{
marginTop: 20 }}>
```

```
  <Text
    style={{
      textAlign: "center",
      fontSize: 17,
      color: "gray",
      fontWeight: "500",
    }}
  >
    Don't have a account? Sign Up
  </Text>
</Pressable>
```

- User can add scrap type they want to add and sell to the seller





```
  <Pressable style={{ width: 80}}>
         <Text
           style={{
             borderColor: "gray",
             borderWidth: 0.8,
             borderRadius:4,
             marginVertical: 10,
             color: "#32712c",
             textAlign: "center",
             padding:5,
             fontSize:17,
             fontWeight:"bold"
           }}
         >
           Add
         </Text>
       </Pressable>
     </Pressable>
   </View>
```

Conclusion:

By adding user interactivity to base components and the layout customization functions provided by React Native and XAML, learned enhance the user experience and create more dynamic and adaptable interfaces for mobile applications.

References:

1. React Native Documentation: https://reactnative.dev/docs/getting-started

2. XAML Layout and Customization: https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml