Cross Platform App Development Lab Experiment No.4

Aim: Basic user interface design using react components, Stylesheet, Flexbox and XAML, manipulation of components using States and Props.

Objectives:

1. Understand the basics of React components.
2. Learn how to use Stylesheet for styling.
3. Grasp the concepts of Flexbox for layout design.
4. Familiarize yourself with XAML for markup.
5. Explore the manipulation of components using React States and Props.

Theory:

- React Components:

- Building blocks of a React application.
- Encapsulate reusable code.
- Can be class components or functional components.

- Stylesheet:

- Used for styling React components.
- Helps in maintaining a consistent look and feel.
- Can include CSS or other styling languages.

- Flexbox:

- A layout model for designing complex layouts.
- Provides an efficient way to distribute space among items in a container.
- Simplifies the design of responsive and dynamic layouts.

- XAML:

- Extensible Application Markup Language.
- Used for designing user interfaces in .NET applications.
- A declarative XML-based language.

- States and Props:

- States:

- Manage the internal state of a component.
- Allow components to change their output over time.

- Props:
  - Short for properties.
  - Enable the passing of data from parent to child components.

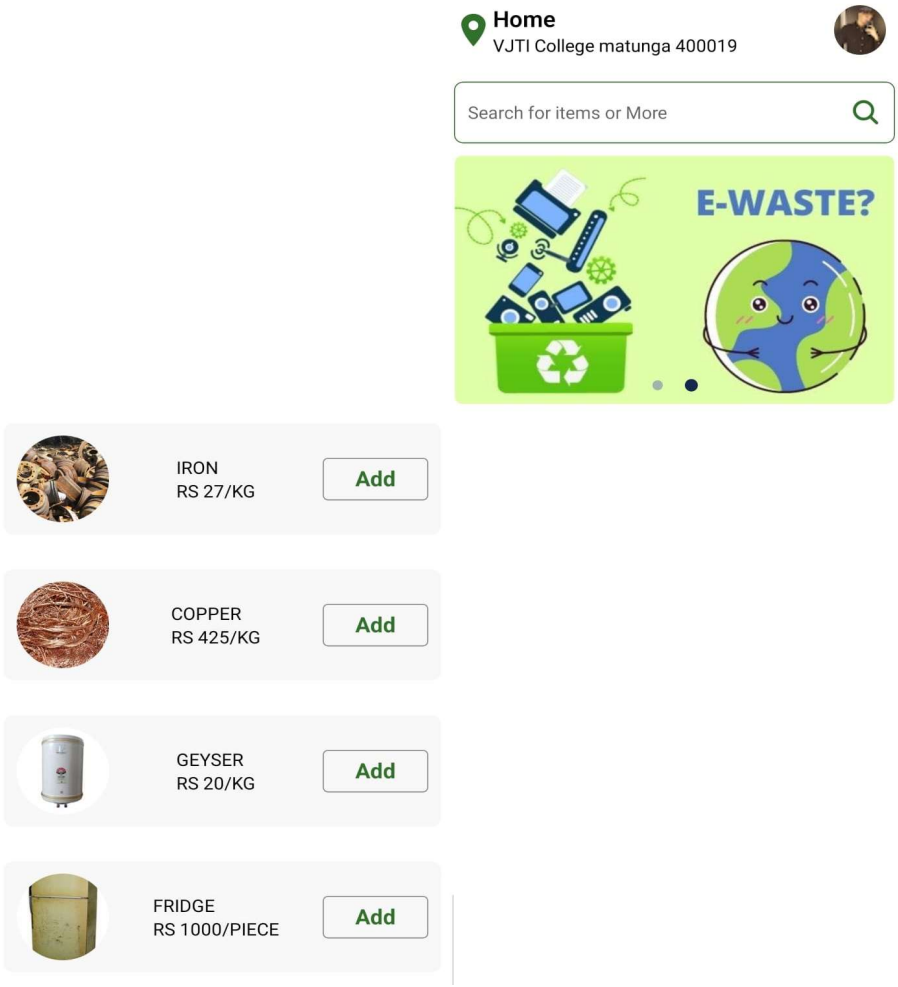
Requirements:

- A text editor (e.g., Visual Studio Code).
- Node.js and npm installed.
- Basic understanding of HTML and JavaScript.
- Familiarity with the React library.

Tools:

- Text editor (e.g., Visual Studio Code).
- Node.js and npm.

Implementation/Code:-



Code:

- Using Flexbox for designing

```
 return (
    // Location and profile
    <ScrollView style={{backgroundColor: "F0F0F0", flex:1,marginTop:1}}>
      <StatusBar hidden />

      <View style={{ flexDirection: "row", alignItems: "center", padding: 10
}}>
        <Ionicons name="location-sharp" size={30} color="#32712c" />
        <View>
          <Text style={{ fontSize: 18, fontWeight: "600" }}>Home</Text>
          <Text>{displayCurrentAddress}</Text>
        </View>

        <Pressable style={{ marginLeft: "auto", marginRight: 7 }}>
          <Image
            style={{ width: 40, height: 40, borderRadius: 20 }}
            source={{
              uri:
"https://lh3.googleusercontent.com/ogw/AKPQZvzhvGMWRESqI4jU33yjRU876j-
tzbgWV0948GYgVw=s32-c-mo",
            }}
          />
        </Pressable>
      </View>
      {/* Search bar */}
```

- Using in style toh ise CCS

```
 <View
        style={{
          padding: 10,
          margin: 10,
          flexDirection: "row",
          alignItems: "center",
          justifyContent: "space-between",
          borderWidth: 0.8,
          borderColor: "#32712c",
          borderRadius: 7,
        }}
      >
        <TextInput placeholder="Search for items or More" />
```

- Using inline style to style list of scrap items.

```
const ScrapItem = ({ item }) => {
  return (
    <View>
      <Pressable
        style={{
          backgroundColor: "#F8F8F8",
          borderRadius: 8,
          padding: 10,
          flexDirection: "row",
          alignItems: "center",
          justifyContent: "space-between",
          margin: 14,
        }}
      >
        <View>
          <Image
            style={{ width: 70, height: 70, borderRadius:50}}
            source={{ uri: item.image }}
          />
        </View>
        <View>
          <Text>{item.name}</Text>
          <Text>{item.price}</Text>
        </View>
```

Conclusion:

In this project, we learnt the fundamentals of React components, Stylesheet for styling, Flexbox for layout, XAML for markup, and the usage of States and Props for component manipulation. This provides a solid foundation for building interactive and well-styled user interfaces using React.

References:

1. **React Documentation**: https://reactjs.org/docs/getting-started.html
2. **Flexbox Guide:** https://css-tricks.com/snippets/css/a-guide-to-flexbox/

3. **XAML Overview:** https://docs.microsoft.com/en-us/dotnet/desktop-wpf/fundamentals/xaml