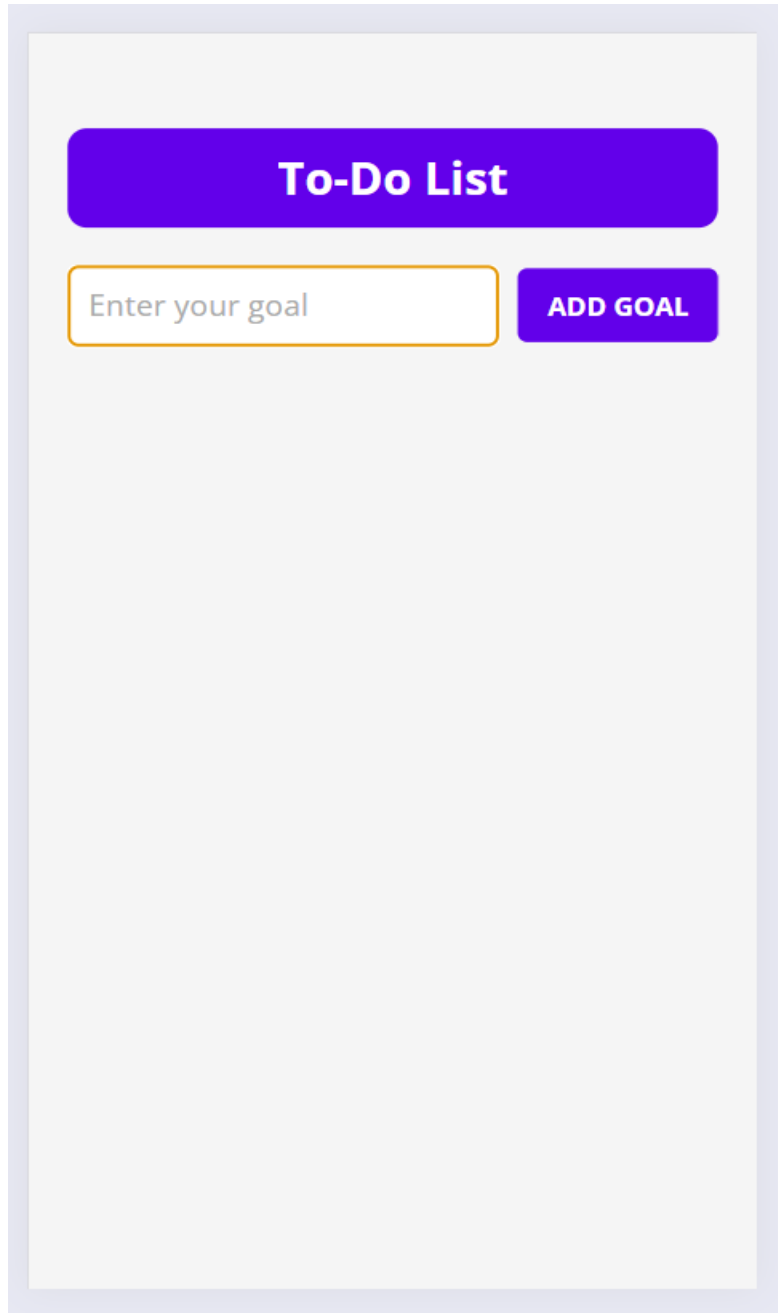


Name: **مريم اشرف عطيه ذكي**

Section : 2

React Native To-Do List App Report

Screenshots of my design:



To-Do List

Task

ADD GOAL

To-Do List

Enter your goal

ADD GOAL

Task

To-Do List

ADD GOAL

Task

Task1

Task2

Task3

Task4

Task5

Task6

Task7

Task8

To-Do List

ADD GOAL

Task1

Task2

Task3

Task4

Task5

Task6

Task7

Task8

Task9

Task10

Colors and Fonts Used:

Colors:

- **Primary Color:** #6200EA (Deep Purple)
- **Background Color:** #f5f5f5 (Light Gray)
- **Text Placeholder Color:** #aaa (Gray)
- **Button Text Color:** #fff (White)

Fonts:

- **Regular Font:** OpenSans-Regular
- **Bold Font:** OpenSans-Bold

Code Breakdown

1. Components and Logic

App Component

The main component manages:

- State for storing goals (goal and goals array)
- Font loading using expo-font
- Handling user input and adding goals

```
import React, { useState, useEffect } from 'react';
import { Text, View, TextInput, TouchableOpacity, FlatList, StyleSheet, ActivityIndicator } from
'react-native';
import * as Font from 'expo-font';

export default function App() {
  const [goal, setGoal] = useState('');
  const [goals, setGoals] = useState([]);
  const [fontsLoaded, setFontsLoaded] = useState(false);

  useEffect(() => {
    async function loadFonts() {
      await Font.loadAsync({
        'OpenSans-Regular': require('./assets/fonts/OpenSans-Regular.ttf'),
        'OpenSans-Bold': require('./assets/fonts/OpenSans-Bold.ttf'),
      });
      setFontsLoaded(true);
    }
    loadFonts();
  }, []);

  if (!fontsLoaded) {
    return <ActivityIndicator size="large" color="#6200EA" />;
  }

  const addGoalHandler = () => {
    if (goal.trim().length > 0) {
      setGoals([...goals, goal]);
      setGoal('');
    }
  };
}
```



```
return (  
  <View style={styles.container}>  
    <Text style={styles.header}>To-Do List</Text>  
    <View style={styles.inputContainer}>  
      <TextInput  
        style={styles.input}  
        placeholder="Enter your goal"  
        placeholderTextColor="#aaa"  
        value={goal}  
        onChangeText={setGoal}  
      />  
      <TouchableOpacity style={styles.addButton} onPress={addGoalHandler}>  
        <Text style={styles.addButtonText}>ADD GOAL</Text>  
      </TouchableOpacity>  
    </View>  
    <FlatList  
      data={goals}  
      keyExtractor={({item, index}) => index.toString()}  
      renderItem={({ item }) => (  
        <View style={styles.goalItem}>  
          <Text style={styles.goalText}>{item}</Text>  
        </View>  
      )}  
    />  
  </View>  
}
```


2. Styling

All styles are defined in a StyleSheet object.

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
    paddingTop: 50,
    paddingHorizontal: 20,
  },
  header: {
    fontSize: 24,
    fontFamily: 'OpenSans-Bold',
    textAlign: 'center',
    marginBottom: 20,
    backgroundColor: '#6200EA',
    color: 'fff',
    paddingVertical: 10,
    borderRadius: 10,
  },
  inputContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    marginBottom: 20,
  },
});
```

```
input: {
  flex: 1,
  borderWidth: 1,
  borderColor: '#6200EA',
  padding: 10,
  borderRadius: 5,
  fontSize: 16,
  fontFamily: 'OpenSans-Regular',
  backgroundColor: '#fff',
},
addButton: {
  marginLeft: 10,
  backgroundColor: '#6200EA',
  paddingVertical: 10,
  paddingHorizontal: 15,
  borderRadius: 5,
},
addButtonText: {
  color: '#fff',
  fontSize: 14,
  fontFamily: 'OpenSans-Bold',
},
goalItem: {
  backgroundColor: '#6200EA',
  padding: 10,
  marginVertical: 5,
  borderRadius: 5,
},
goalText: {
  color: '#fff',
  fontSize: 16,
  fontFamily: 'OpenSans-Regular',
},
});
```