

Basic layout

Create New Project with expo

```
expo init -rn-complete-guide
```

choose blank template

1. Basic layout - 1

```
import { StatusBar } from 'expo-status-bar';
import React from 'react';
import { StyleSheet, Text, View, TextInput, Button } from 'react-native';

export default function App() {
  return (
    <View style={{padding: 50}}>
      <View style={{flexDirection: 'row', justifyContent: 'space-between',
        alignItems: 'center'}}>
        <TextInput placeholder="Course Goal"
          style={{width: '80%', borderColor: 'black', borderBottomWidth: 1,
            padding: 10}}/>
        <Button title="Add"/>
      </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Flexbox

Ref:

<https://medium.com/wix-engineering/the-full-react-native-layout-cheat-sheet-a4147802405c>

<https://reactnative.dev/docs/flexbox>

```
// each box
<View style={{
  backgroundColor:'red',
  width:100,
  height:100,
  justifyContent:'center',
  alignItems:'center'
}}>
  <Text>1</Text>
</View>
```

```
// For padding with each border
<View style={{padding:50}}>
</View>
```

```
// flex box direction
<View style={{padding:50, flexDirection:'row'}}>
</View>
```

```
// add parent view this style
// remove child view width and height
<View style={{
  padding:50,
  flexDirection:'row',
  width:'80%',
  height:300,
  justifyContent:'space-around',
  alignItems:'stretch'
}}>
</View>
```

```
// add flex:1 at child view

<View style={{
  backgroundColor:'red',
  flex:1,
  justifyContent:'center',
  alignItems:'center'
}}>
  <Text>1</Text>
</View>
```

```
// add flex:2 for second child view
<View style={{
  backgroundColor:'green',
  flex:2,
  justifyContent:'center',
  alignItems:'center'
}}>
  <Text>2</Text>
</View>
```

Basic layout - 2

```
// create a style with style sheet object

<View style={styles.screen}>
  <View style={styles.inputContainer}>
    <TextInput placeholder="Course Goal" style={styles.input} />
    <Button title="Add" />
  </View>
</View>

const styles = StyleSheet.create({
  screen: {
    padding: 50,
  },
  inputContainer: {
    flexDirection: "row",
    justifyContent: "space-between",
    alignItems: "center",
  },
  input: {
    width: "80%",
    borderColor: "black",
    borderBottomWidth: 1,
    padding: 10,
  },
});
```

Working with Stage & Events

```

//1. import useStage
import React,{useState} from "react";

//2. set for null value for text box
const [enteredGoal,setEnteredGoal] = useState('');

//3. create function goInputHandler
function goInputHandler(enterText) {
  setEnteredGoal(enterText);
}

//3.1 arrow function
const goInputHandler = enterText => setEnteredGoal(enterText);

//4. add onChangeText events
<TextInput
  placeholder="Course Goal"
  style={styles.input}
  onChangeText={goInputHandler}
/>

//5. add value property
<TextInput
  placeholder="Course Goal"
  style={styles.input}
  onChangeText={goInputHandler}
  value={enteredGoal}
/>

//6. add onPress Events at button
<Button title="Add" onPress={addGoHandler} />

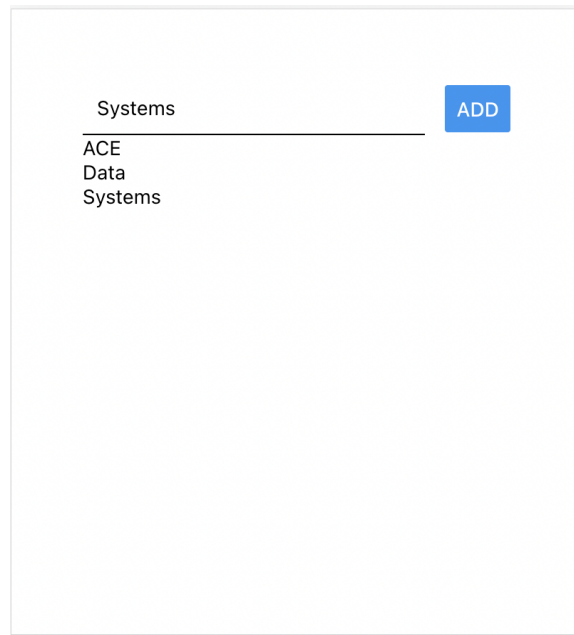
//7. implement addGoHandler function
const addGoHandler = () => {
  console.log(enteredGoal);
}

//8
const [courseGoals, setCourseGoals] = useState([]);

// courseGoal is previous stage, enterGoal is update stage
const addGoHandler = () => {
  setCourseGoals(currentGoals => [...currentGoals,enteredGoal]);
}
// 9 Bind the new view
<View>
  {courseGoals.map((goal) => <Text key={goal}>{goal}</Text>)}
</View>

```

Testing the app

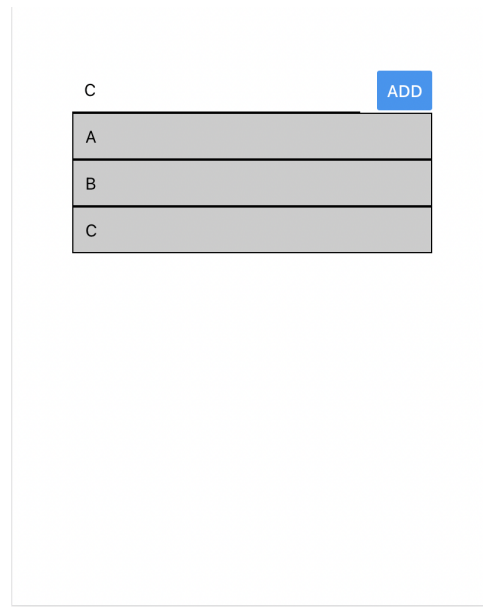


2. Styling the items

```
// Text component to wrap with View component
// View component is more style option
<View>
  <View style={styles.listItem}>
    {courseGoals.map((goal) => (
      <Text key={goal}>{goal}</Text>
    ))}
  </View>
</View>

// list item style create

listItem:{
  padding:10,
  backgroundColor:'#ccc',
  borderColor:'black',
  borderWidth:1
}
```



Git:Repo

<https://github.com/arkarhtetmyint/rn-complete-guide.git>