



Report #1 - Todo List

– Screenshots

The first screenshot shows the initial state of the Todo List application. It has three sections: ACTIVE (1), CLEARED (1), and COMPLIANCE (50%). Below these are three tasks: "REPORT TO SECTOR-A FOR BIOMETRIC RECALIBRATION" (unchecked), "SUBMIT WEEKLY PRODUCTIVITY METRICS" (checked), and "SUBMIT MY FIRST TASK IN MOBIL-LAB" (checked). The second screenshot shows a new task "ENTER NEW TASK..." being added. The third screenshot shows a task being deleted, with a red trash icon appearing next to the "SUBMIT WEEKLY PRODUCTIVITY METRICS" task.

1.Initial State

2.Add Task

3.Delete Task

The left screenshot shows the Todo List application with two tasks: "Submit my second lab" (unchecked) and "REPORT TO SECTOR-A FOR BIOMETRIC RECALIBRATION" (checked). A green oval highlights the "Submit my second lab" task. The right screenshot shows the same application after editing, where the task has been updated to "SUBMIT MY SECOND LAB" (checked). A green oval highlights this edited task.

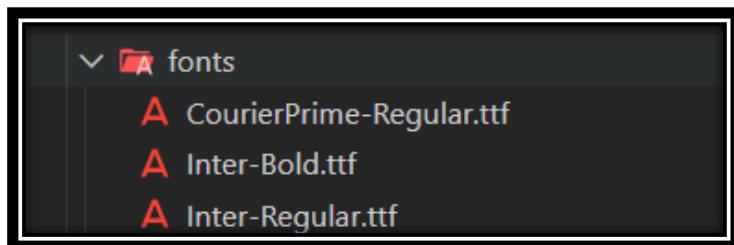
4.Edit Task



— Colors

```
export const Theme = {  
  colors: {  
    background: '#000000',  
    surface: '#050505',  
    primary: '#00d1a7',  
    text: '#FFFFFF',  
    logo: '#b3eade',  
    textMuted: '#333333',  
    border: '#181717',  
    glow: '#00ffc3f4',
```

— Fonts





— Code

```
src > components > Header.js > ...
1 import { View, Text, StyleSheet } from 'react-native';
2 import { Theme } from '../constants/Theme_dark';
3
4 export default function Header({ active, cleared }) {
5   const total = active + cleared;
6   const compliance = total > 0 ? Math.round((cleared / total) * 100) : 0;
7
8   return (
9     <View style={styles.container}>
10       <View style={styles.topRow}>
11         <Text style={styles.logo}>Todo List</Text>
12         <Text style={styles.version}>V1.7</Text>
13       </View>
14
15       <View style={styles.statsRow}>
16         <View style={styles.statBox}>
17           <Text style={styles.statLabel}>ACTIVE</Text>
18           <Text style={styles.statValue}>{active}</Text>
19         </View>
20         <View style={styles.statBox}>
21           <Text style={styles.statLabel}>CLEARED</Text>
22           <Text style={styles.statValue}>{cleared}</Text>
23         </View>
24         <View style={styles.statBox}>
25           <Text style={styles.statLabel}>COMPLIANCE</Text>
26           <Text style={[styles.statValue, { color: Theme.colors.primary }]}>{compliance}%</Text>
27         </View>
28       </View>
29       <View style={styles.divider} />
30     </View>
31   );
32 }
33
34 const styles = StyleSheet.create({
35   container: { padding: Theme.spacing.lg, backgroundColor: Theme.colors.background },
36   topRow: { flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', marginBottom: 40 },
37   logo: {
38     color: Theme.colors.logo, // Gold for brightness
39     fontSize: 36,
40     fontFamily: 'Inter-Regular',
41     letterSpacing: 3,
42     textShadowColor: Theme.colors.glow,
43     textShadowOffset: { width: 3, height: 1 },
44     textShadowRadius: 5,
45     elevation: 2,
46   },
47   version: { color: '#444', fontSize: 12, fontFamily: 'Courier' },
48   statsRow: { flexDirection: 'row', gap: 40 },
49   statLabel: { color: '#444', fontSize: 12, fontFamily: 'Courier', marginBottom: 8 },
50   statValue: { color: Theme.colors.text, fontSize: 24, fontFamily: 'Courier' },
51   divider: { height: 1, backgroundColor: '#222', marginTop: 30 }
52 });

```



```
src > components > TodoInput.js ...
1  import { useState } from 'react';
2  import { View, TextInput, TouchableOpacity, StyleSheet } from 'react-native';
3  import { Ionicons } from '@expo/vector-icons';
4  import { Theme } from '../constants/Theme_dark';
5
6  export default function TodoInput({ onAdd }) {
7    const [text, setText] = useState('');
8
9    return (
10      <View style={styles.container}>
11        <TextInput
12          style={styles.input}
13          placeholder="ENTER NEW TASK..."
14          placeholderTextColor="#222"
15          value={text}
16          onChangeText={setText}
17        />
18        <TouchableOpacity
19          style={styles.addButton}
20          onPress={() => { if (text) { onAdd(text); setText(''); } }}
21        >
22          <Ionicons name="add" size={24} color="black" />
23        </TouchableOpacity>
24      </View>
25    );
26  }
27
28 const styles = StyleSheet.create({
29   container: { flexDirection: 'row', padding: Theme.spacing.lg, gap: 10 },
30   input: {
31     flex: 1,
32     backgroundColor: '#0A0A0A',
33     borderWidth: 1,
34     borderColor: '#1A1A1A',
35     padding: 15,
36     color: Theme.colors.text,
37     fontFamily: 'Courier',
38     letterSpacing: 1,
39   },
40   addButton: {
41     backgroundColor: Theme.colors.primary,
42     width: 55,
43     justifyContent: 'center',
44     alignItems: 'center',
45   }
46 });

```



```
src > components > TodoItem.js > TodoItem
1  import { TouchableOpacity, Text, StyleSheet, View, TextInput } from 'react-native';
2  import { Ionicons } from '@expo/vector-icons';
3  import { Theme } from '../constants/Theme_dark';
4  import { Swipeable } from 'react-native-gesture-handler';
5  import { useState } from 'react';
6
7  export default function TodoItem({ item, onToggle, onDelete, onEdit }) {
8      const [editing, setEditing] = useState(false);
9      const [editText, setEditText] = useState(item.text);
10     const renderRightActions = () => (
11         <TouchableOpacity style={styles.deleteButton} onPress={() => onDelete(item.id)}>
12             <Ionicons name="trash" size={24} color="white" />
13         </TouchableOpacity>
14     );
15
16     const handleEditSubmit = () => {
17         setEditing(false);
18         if (editText.trim() && editText !== item.text) {
19             onEdit(item.id, editText);
20         }
21     };
22
23     return (
24         <Swipeable renderRightActions={renderRightActions}>
25             <View style={styles.container}>
26                 <TouchableOpacity onPress={() => onToggle(item.id)}>
27                     <View style={[styles.checkbox, item.completed && styles.checked]}>
28                         {item.completed && <Ionicons name="checkmark" size={16} color="black" /}
29                     </View>
30                 </TouchableOpacity>
31                 {editing ? (
32                     <TextInput
33                         style={[styles.text, item.completed && styles.textCompleted, styles.editInput]}
34                         value={editText}
35                         onChangeText={setEditText}
36                         autoFocus
37                         onSubmitEditing={handleEditSubmit}
38                         onBlur={handleEditSubmit}
39                         returnKeyType="done"
40                     />
41                 ) : (
42                     <View style={{ flexDirection: 'row', alignItems: 'center', flex: 1 }}>
43                         <Text style={[styles.text, item.completed && styles.textCompleted]}>
44                             {item.text.toUpperCase()}
45                         </Text>
46                         <TouchableOpacity style={styles.inlineEditIcon} onPress={() => setEditing(true)}>
47                             <Ionicons name="create" size={20} color={Theme.colors.primary} />
48                         </TouchableOpacity>
49                     </View>
50                 )}
51             </View>
52         </Swipeable>
53     );
54 }
```



Computer and Systems Engineering Department

الاسم / عمر حسين عبد الرضاى محمد

الفرقة / الرابعة



```
56 const styles = StyleSheet.create({
57   container: {
58     flexDirection: 'row',
59     alignItems: 'center',
60     paddingVertical: 20,
61     borderBottomWidth: 1,
62     borderBottomColor: Theme.colors.border,
63     paddingHorizontal: 20
64   },
65   checkbox: {
66     width: 24,
67     height: 24,
68     borderWidth: 1,
69     borderColor: Theme.colors.textMuted,
70     marginRight: 20,
71     justifyContent: 'center',
72     alignItems: 'center'
73   },
74   checked: { backgroundColor: Theme.colors.primary, borderColor: Theme.colors.primary },
75   text: { color: Theme.colors.text, fontFamily: 'Courier New', fontSize: 14, flex: 1, letterSpacing: 1 },
76   textCompleted: { color: Theme.colors.textMuted, textDecorationLine: 'line-through' },
77   deleteButton: {
78     backgroundColor: 'red',
79     justifyContent: 'center',
80     alignItems: 'center',
81     width: 60,
82     height: '100%',
83   },
84   inlineEditIcon: {
85     marginLeft: 10,
86     padding: 4,
87     borderRadius: 4,
88   },
89   editInput: {
90     backgroundColor: Theme.colors.surface,
91     color: Theme.colors.text,
92     borderRadius: 4,
93     paddingHorizontal: 8,
94     fontSize: 14,
95     flex: 1,
96   },
97 });
});
```



```
src > screens > HomeScreen.js > ...
1  import { useState } from 'react';
2  import { View, StyleSheet, FlatList, Text } from 'react-native';
3  import { SafeAreaView } from 'react-native-safe-area-context';
4  import { StatusBar } from 'expo-status-bar';
5  import { Theme } from '../constants/Theme_dark';
6  import Header from '../components/Header';
7  import TodoInput from '../components/TodoInput';
8  import TodoItem from '../components/TodoItem';
9
10 export default function HomeScreen() {
11   const [todos, setTodos] = useState([
12     { id: '1', text: 'REPORT TO SECTOR-A FOR BIOMETRIC RECALIBRATION', completed: false },
13     { id: '2', text: 'SUBMIT WEEKLY PRODUCTIVITY METRICS', completed: true },
14   ]);
15
16   const active = todos.filter(t => !t.completed).length;
17   const cleared = todos.filter(t => t.completed).length;
18
19   return (
20     <SafeAreaView style={styles.container}>
21       <StatusBar barStyle="light" hidden={true} />
22       <Header active={active} cleared={cleared} />
23       <TodoInput onAdd={(text) => setTodos([{ id: Date.now().toString(), text, completed: false }, ...todos])} />
24
25       <Flatlist
26         data={[...todos].sort((a, b) => {
27           if (a.completed !== b.completed) return a.completed - b.completed;
28           return b.id.localeCompare(a.id);
29         })}
30         keyExtractor={item => item.id}
31         renderItem={({ item }) => (
32           <TodoItem
33             item={item}
34             onToggle={() => setTodos(todos.map(t => t.id === id ? { ...t, completed: !t.completed } : t))}
35             onDelete={() => setTodos(todos.filter(t => t.id !== id))}
36             onEdit={(id, text) => setTodos(todos.map(t => t.id === id ? { ...t, text } : t))}
37           />
38         )}
39       />
40
41       <View style={styles.footer}>
42         <Text style={styles.footerText}>
43           | copyright © 2026 Omar Hussein.
44         </Text>
45       </View>
46     </SafeAreaView>
47   );
48 }
49
50 const styles = StyleSheet.create({
51   container: { flex: 1, backgroundColor: Theme.colors.background },
52   footer: { padding: 20, alignItems: 'center' },
53   footerText: { color: '#222', fontSize: 9, fontFamily: 'Courier', letterSpacing: 1 }
54});
```



```
JS App.js > App
1 import { useCallback } from 'react';
2 import { useFonts } from 'expo-font';
3 import * as SplashScreen from 'expo-splash-screen';
4 import { SafeAreaProvider } from 'react-native-safe-area-context';
5 import { GestureHandlerRootView } from 'react-native-gesture-handler';
6 import HomeScreen from './src/screens/HomeScreen';
7
8 SplashScreen.preventAutoHideAsync();
9
10 export default function App() {
11   const [fontsLoaded, fontError] = useFonts({
12     'Inter-Regular': require('./assets/fonts/Inter-Regular.ttf'),
13     'Inter-Bold': require('./assets/fonts/Inter-Bold.ttf'),
14     'Courier New': require('./assets/fonts/CourierPrime-Regular.ttf'),
15   });
16
17   const onLayoutRootView = useCallback(async () => {
18     if (fontsLoaded || fontError) {
19       // This tells the splash screen to hide immediately!
20       await SplashScreen.hideAsync();
21     }
22   }, [fontsLoaded, fontError]);
23
24   if (!fontsLoaded && !fontError) {
25     return null;
26   }
27
28   return (
29     <GestureHandlerRootView style={{ flex: 1 }}>
30       <SafeAreaProvider onLayout={onLayoutRootView}>
31         <HomeScreen />
32       </SafeAreaProvider>
33     </GestureHandlerRootView>
34   );
35 }
```