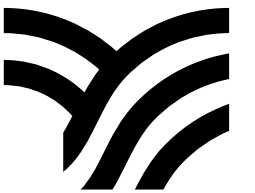


2025



Abdullah Eid El-Sayed Ammar



To-Do List Report

Color & Fonts



2025

Font: Galindo font ([Link](#))

Color:

- **Background:** #fff (White) 
 - **Texts:** #000 (Black) 
 - **Buttons:** #00f (Blue) 
 - **Checkbox-Background:** #2ecc7190 
 - **Checkbox-Border:** #2ecc7177 
 - **Finished-Task:** #cbd3d380 
 - **Finished-Task-Text:** #d3d3d3 (lightgray) 
- 

Screenshots



2025



you can add tasks by click
enter or add button

when there is no tasks

A mockup of a mobile application titled "To-Do List". It features a text input field with the placeholder "task 1", a blue "ADD" button, a large empty list container, and a footer message "you have finished all tasks". Green arrows point from external text labels to these specific UI elements.

To-Do List

task 1

ADD

you have finished all tasks

when there is no tasks

Tasks will appear here



2025



will change when he has not
finished tasks



**Please finish
your task, you
idiot**

ADD

☐ task 1

X

Total: 1

Finished: 0



task was added and it has
delete button and checkbox
to check it as finished



counter for total tasks added



counter for total tasks added



2025



will change when he has
more than 5 not finished
tasks



**I'm so
disappointed
you have more
than 5 task not
finished?!**

ADD

<input type="checkbox"/>	task 1	×
<input type="checkbox"/>	task2	×
<input type="checkbox"/>	task 3	×
<input type="checkbox"/>	task 4	×
<input type="checkbox"/>	task 5	×

Total: 6

Finished: 0



2025

checked task will changed
to appear like checked

Please finish your task, you idiot

ADD

<input checked="" type="checkbox"/>	task1	×
<input checked="" type="checkbox"/>	task2	×
<input type="checkbox"/>	task 3	×
<input type="checkbox"/>	task 4	×
<input type="checkbox"/>	task 5	×
<input type="checkbox"/>	task 6	×

Total: 6

Finished: 2

the not finished task is 4 and
less than or equal 5 so the
title will be changed

finished task counter will
increase

2025



it will notify the user that it
has finished all tasks

To-Do List

ADD

☐ task-1

×

☐ task2

×

☐ task-3

×

☐ task-4

×

☐ task-5

×

☐ task-6

×

Total: 6Finished: 6

you have finished all tasks



there is no not finished tasks
so the title will return to
default value



2025



**Please finish
your task, you
idiot**

ADD

☐ task 4

☒ task 5

☒ task 6

Total: 3Finished: 2



he can delete an unckeck
any task as he want



Code



2025



```
import { useFonts } from "expo-font";
import React, { useState } from "react";

import {
  View,
  Text,
  TextInput,
  TouchableOpacity,
  FlatList,
  StyleSheet,
} from "react-native";

function App() {
  const [task, setTask] = useState("");
  const [taskCounter, setTaskCounter] = useState(0);
  const [finishedTaskCounter, setFinishedTaskCounter] = useState(0);
  const [tasks, setTasks] = useState([]);
  const [loaded] = useFonts({
    Galindo: require("./assets/fonts/Galindo-Regular.ttf"),
  });
}
```

2025

```
function addTask() {
  if (task.trim() !== "") {
    setTaskCounter((taskCounter) => taskCounter + 1);
    setTasks([
      ...tasks,
      { id: Date.now().toString(), text: task, completed: false },
    ]);
    setTask("");
  }
}

function removeTask(id, completed) {
  console.log(completed);
  if (taskCounter !== 0) setTaskCounter((taskCounter) => taskCounter - 1);
  if (finishedTaskCounter !== 0 && completed)
    setFinishedTaskCounter((taskCounter) => taskCounter - 1);
  setTasks(tasks.filter((task) => task.id !== id));
}

function toggleTask(id) {
  setTasks(
    tasks.map((task) => {
      task.id === id
        ? task.completed
          ? finishedTaskCounter
            ? setFinishedTaskCounter((counter) => counter - 1)
            : {}
          : setFinishedTaskCounter((counter) => counter + 1)
        : {};
      return task.id === id ? { ...task, completed: !task.completed } : task;
    })
  );
}
```

2025

```
if (!loaded) {
  return null;
}

return (
  <View style={styles.container}>
    <View style={styles.header}>
      <Text style={styles.title}>
        {taskCounter - finishedTaskCounter > 0
          ? taskCounter - finishedTaskCounter > 5
            ? "I'm so disappointed you have more than 5 task not finished?!"
            : "Please finish your task, you idiot"
          : "To-Do List"}
      </Text>
      <View style={styles.headerComponents}>
        <TextInput
          style={styles.input}
          placeholder="Enter a task..."
          value={task}
          onChangeText={setTask}
          onSubmitEditing={addTask}
        />
        <TouchableOpacity style={styles.addButton} onPress={addTask}>
          <Text style={styles.addButtonText}>ADD</Text>
        </TouchableOpacity>
      </View>
    </View>
  </View>
)
```

2025

```
<View style={styles.body}>
  <FlatList
    data={tasks}
    keyExtractor={({item}) => item.id}
    renderItem={({ item }) => (
      <View
        style={[
          styles.taskContainer,
          item.completed && styles.completedTask,
        ]}
      >
        <View
          style={{
            flexDirection: "row",
            alignItems: "center",
            gap: 10,
          }}
        >
          <TouchableOpacity
            onPress={() => toggleTask(item.id)}
            style={[
              styles.checkbox,
              item.completed && styles.completedCheck,
            ]}
          ></TouchableOpacity>
          <Text
            style={[
              styles.taskText,
              item.completed && styles.taskCompleted,
            ]}
          >
            {item.text}
          </Text>
        </View>
        <TouchableOpacity
          onPress={() => removeTask(item.id, item.completed)}
        >
          <Text style={styles.deleteText}>✕</Text>
        </TouchableOpacity>
      </View>
    )}
  />
</View>
```

2025

```

{taskCounter !== 0 && (
  <View style={styles.taskCounterContainer}>
    <Text style={styles.counter}>`Total: ${taskCounter}`</Text>
    {taskCounter ? (
      <Text style={styles.counter}>
        `Finished: ${finishedTaskCounter}`
      </Text>
    ) : (
      <></>
    )}
  </View>
)}
{taskCounter - finishedTaskCounter === 0 && (
  <View>
    <Text style={styles.finish}>you have finished all tasks</Text>
  </View>
)}
</View>
);
}
```


2025

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 60,
    paddingHorizontal: 30,
    backgroundColor: "#fff",
  },
  title: {
    fontSize: 40,
    textAlign: "center",
    fontWeight: "bold",
    marginBottom: 30,
    fontFamily: "Galindo",
  },
  headerComponents: {
    flexDirection: "row",
    height: 45,
    gap: 20,
    paddingHorizontal: 10,
  },
  input: {
    borderWidth: 1,
    padding: 10,
    paddingLeft: 20,
    borderColor: "lightgray",
    borderRadius: 1000,
    height: "100%",
    width: "75%",
    placeholderTextColor: "lightgray",
    fontFamily: "Galindo",
  },
  addButton: {
    backgroundColor: "blue",
    padding: 10,
    borderRadius: 1000,
    alignItems: "center",
    width: "25%",
  },
});
```

2025



```
addButtonText: { color: "white", fontFamily: "Galindo", fontSize: 16 },
taskContainer: {
  flexDirection: "row",
  alignItems: "center",
  justifyContent: "space-between",
  paddingHorizontal: 20,
  paddingVertical: 15,
  borderBottomColor: "lightgray",
  borderBottomWidth: 1,
  marginHorizontal: 10,
  borderRadius: 5,
  fontFamily: "Galindo",
},
taskText: {
  fontSize: 16,
  flexWrap: "wrap",
  maxWidth: "70%",
  fontFamily: "Galindo",
},
deleteText: { color: "#f05550", fontFamily: "Galindo", fontWeight: "bold" },
header: { marginBottom: 30, width: "100%" },
body: {
  fontFamily: "Galindo",
  flex: 1,
  width: "100%",
  borderWidth: 3,
  borderRadius: 5,
  borderColor: "lightgray",
  marginBottom: 20,
},
counter: { fontSize: 18, fontFamily: "Galindo", fontWeight: "bold" },
finish: {
  fontFamily: "Galindo",
  fontSize: 18,
  textAlign: "center",
  fontWeight: "bold",
  marginTop: 20,
},
checkbox: {
  width: 25,
  height: 25,
  borderWidth: 2,
  borderColor: "lightgray",
  borderRadius: 5,
  alignItems: "center",
  justifyContent: "center",
  marginRight: 10,
},
```



2025

```
completedTask: {
  backgroundColor: "#cbded380",
},
completedCheck: {
  backgroundColor: "#2ECC7190",
  borderColor: "#2ECC7177",
},
taskCompleted: {
  textDecorationLine: "line-through",
  color: "gray",
},
taskCounterContainer: {
  flexDirection: "row",
  justifyContent: "space-evenly",
  alignItems: "center",
  columnGap: 40,
  fontFamily: "Galindo",
},
});

export default App;
```

2025



Thank
You!

