

To-Do List App Report

1. Introduction

This report documents the design and implementation of a simple To-Do List app built using React Native. The app allows users to:

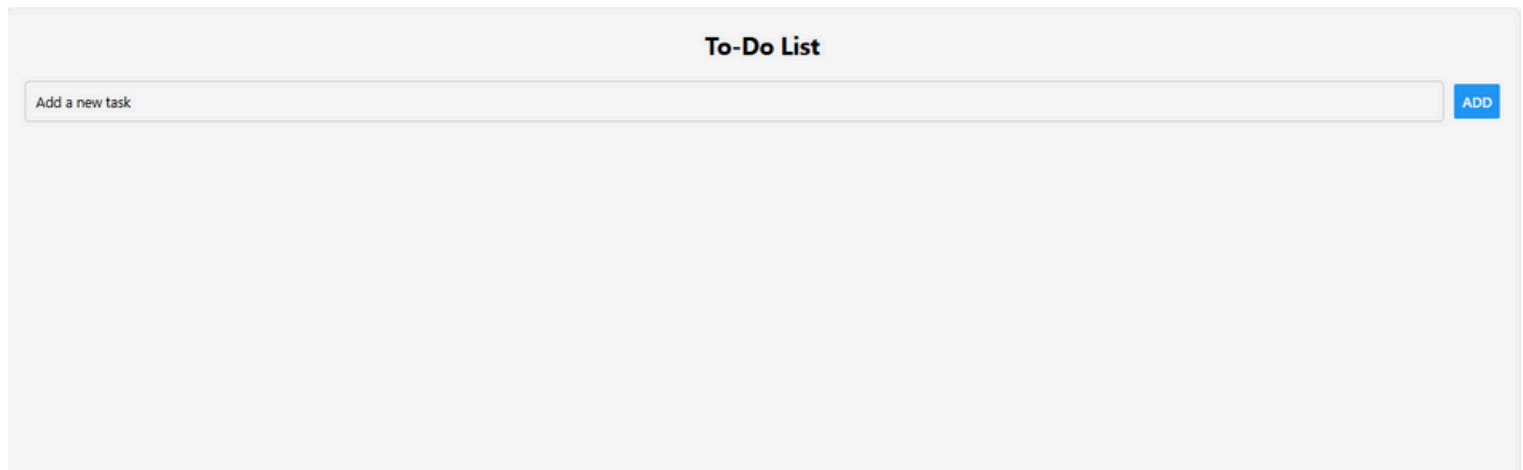
- Add new tasks.
- Mark tasks as completed.
- Delete tasks.

The app is designed to be intuitive and user-friendly, with a clean and modern interface.

2. Screenshots of the Design

Below are screenshots of the app, showcasing different actions and their results.

2.1 Home Screen



- Displays the title "To-Do List."
- Shows an input field and an "Add" button for adding new tasks.
- Lists all to-do items.

2.2 Adding a New Task

To-Do List

Task-01

ADD

To-Do List

Add a new task

ADD

Task-01

Delete

- The user types a task in the input field and clicks "Add."
- The new task appears in the list below.

2.3 Marking a Task as Completed

To-Do List

Add a new task

ADD

Task-01

Delete

Task-02

Delete

Task-03

Delete

Task-04

Delete

Task-05

Delete

Task-06

Delete

Task-07

Delete

Task-08

Delete

- The user taps on a task to mark it as completed.
- Completed tasks are displayed with a strikethrough and grayed-out text.

2.4 Deleting a Task

To-Do List

Add a new task

ADD

Task-01

Delete

Task-02

Delete

To-Do List

Add a new task

ADD

Task-02

Delete

- The user taps the "Delete" button next to a task.
- The task is removed from the list.

3. Colors Used in the Project

The app uses a simple and clean color scheme to ensure readability and a pleasant user experience. Below are the colors used:

Primary Colors

- Background Color: #f5f5f5 (Light Gray)
Used for the main container to provide a neutral background.
- Task Item Background: #ffffff (White)
Used for individual to-do items to make them stand out against the background.

Text Colors

- Title Text: #000000 (Black)
Used for the app title ("To-Do List").
- Task Text: #000000 (Black)
Used for uncompleted tasks.
- Completed Task Text: #888888 (Gray)
Used for completed tasks to indicate they are done.
- Delete Button Text: #ff0000 (Red)
Used for the "Delete" button to draw attention.

Input Field Colors

- Input Border: #cccccc (Light Gray)
Used for the border of the input field.
- Input Text: #000000 (Black)
Used for the text entered by the user.

4. Fonts Used in the Project

The app uses system fonts for simplicity and consistency across platforms. Below are the font styles:

Title Font

- Font Size: 24px
- Font Weight: bold
- Alignment: Centered
- Example: "To-Do List"

Task Font

- Font Size: 16px
- Font Weight: Normal
- Example: "Task-01"

Completed Task Font

- Font Size: 16px
- Font Weight: Normal
- Text Decoration: line-through
- Example: "~~Task-01~~"

Delete Button Font

- Font Size: 14px
- Font Weight: Normal
- Color: #ff0000 (Red)
- Example: "Delete"

5. Code

App.js

```
import React, { useState } from 'react';
import { View, Text, StyleSheet, FlatList, Button } from 'react-native';
import TodoItem from './components/TodoItem';
import AddTodo from './components/AddTodo';

const App = () => {
  const [todos, setTodos] = useState([]);

  // Add a new task
  const addTodoHandler = (task) => {
    setTodos((currentTodos) => [
      ...currentTodos,
      { id: Math.random().toString(), text: task, completed: false },
    ]);
  };

  // Mark a task as completed
  const toggleTodoHandler = (id) => {
    setTodos((currentTodos) =>
      currentTodos.map((todo) =>
        todo.id === id ? { ...todo, completed: !todo.completed } : todo
      )
    );
  };

  // Delete a task
  const deleteTodoHandler = (id) => {
    setTodos((currentTodos) => currentTodos.filter((todo) => todo.id !== id));
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>To-Do List</Text>
      <AddTodo onAddTodo={addTodoHandler} />
      <FlatList
        data={todos}
        keyExtractor={(item) => item.id}
        renderItem={({ item }) => (
          <TodoItem
            item={item}
            onToggle={toggleTodoHandler}
            onDelete={deleteTodoHandler}
          />
        )}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    backgroundColor: '#f5f5f5',
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 20,
    textAlign: 'center',
  },
});

export default App;
```

AddTodo.js

```
import React, { useState } from 'react';
import { View, TextInput, Button, StyleSheet } from 'react-native';

const AddTodo = ({ onAddTodo }) => {
  const [task, setTask] = useState('');

  const handleAddTodo = () => {
    if (task.trim()) {
      onAddTodo(task);
      setTask('');
    }
  };

  return (
    <View style={styles.container}>
      <TextInput
        style={styles.input}
        placeholder="Add a new task"
        value={task}
        onChangeText={setTask}
      />
      <Button title="Add" onPress={handleAddTodo} />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    marginBottom: 20,
  },
  input: {
    flex: 1,
    borderColor: '#ccc',
    borderWidth: 1,
    borderRadius: 5,
    padding: 10,
    marginRight: 10,
  },
});

export default AddTodo;
```

TodoItem.js

```
import React from 'react';
import { View, Text, StyleSheet, TouchableOpacity } from 'react-native';

const TodoItem = ({ item, onToggle, onDelete }) => {
  return (
    <View style={styles.todoItem}>
      <TouchableOpacity onPress={() => onToggle(item.id)}>
        <Text
          style={[
            styles.todoText,
            item.completed && styles.completedText,
          ]}
        >
          {item.text}
        </Text>
      </TouchableOpacity>
      <TouchableOpacity onPress={() => onDelete(item.id)}>
        <Text style={styles.deleteText}>Delete</Text>
      </TouchableOpacity>
    </View>
  );
};

const styles = StyleSheet.create({
  todoItem: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    padding: 10,
    marginVertical: 5,
    backgroundColor: '#fff',
    borderRadius: 5,
    elevation: 2,
  },
  todoText: {
    fontSize: 16,
  },
  completedText: {
    textDecorationLine: 'line-through',
    color: '#888',
  },
  deleteText: {
    color: 'red',
    fontSize: 14,
  },
});

export default TodoItem;
```