**1. Web Application (React.js)**

**App.js**

jsx

Copy code

import React, { useState } from 'react';

import './App.css';

function App() {

const [tasks, setTasks] = useState([]);

const [task, setTask] = useState('');

const [dueDate, setDueDate] = useState('');

const [priority, setPriority] = useState('Low');

const addTask = () => {

if (task && dueDate) {

setTasks([...tasks, { task, dueDate, priority }]);

setTask('');

setDueDate('');

}

};

const deleteTask = (index) => {

const newTasks = tasks.filter((\_, i) => i !== index);

setTasks(newTasks);

};

return (

<div className="App">

<h1>To-Do List</h1>

<input

type="text"

placeholder="Task Name"

value={task}

onChange={(e) => setTask(e.target.value)}

/>

<input

type="date"

value={dueDate}

onChange={(e) => setDueDate(e.target.value)}

/>

<select value={priority} onChange={(e) => setPriority(e.target.value)}>

<option value="Low">Low</option>

<option value="Medium">Medium</option>

<option value="High">High</option>

</select>

<button onClick={addTask}>Add Task</button>

<ul>

{tasks.map((task, index) => (

<li key={index}>

<span>{task.task} - Due: {task.dueDate} - Priority: {task.priority}</span>

<button onClick={() => deleteTask(index)}>Delete</button>

</li>

))}

</ul>

</div>

);

}

export default App;

**App.css**

css

Copy code

.App {

font-family: Arial, sans-serif;

text-align: center;

}

input, select {

margin: 10px;

}

button {

padding: 10px;

cursor: pointer;

}

ul {

list-style-type: none;

padding: 0;

}

li {

display: flex;

justify-content: space-between;

align-items: center;

margin: 10px 0;

}

**2. Mobile Application (Flutter)**

**main.dart**

dart

Copy code

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'To-Do List App',

home: ToDoListScreen(),

);

}

}

class ToDoListScreen extends StatefulWidget {

@override

\_ToDoListScreenState createState() => \_ToDoListScreenState();

}

class \_ToDoListScreenState extends State<ToDoListScreen> {

final List<Map<String, dynamic>> \_tasks = [];

final \_taskController = TextEditingController();

final \_dueDateController = TextEditingController();

String \_priority = 'Low';

void \_addTask() {

if (\_taskController.text.isNotEmpty && \_dueDateController.text.isNotEmpty) {

setState(() {

\_tasks.add({

'task': \_taskController.text,

'dueDate': \_dueDateController.text,

'priority': \_priority,

});

});

\_taskController.clear();

\_dueDateController.clear();

}

}

void \_deleteTask(int index) {

setState(() {

\_tasks.removeAt(index);

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('To-Do List')),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

children: [

TextField(

controller: \_taskController,

decoration: InputDecoration(hintText: 'Task Name'),

),

TextField(

controller: \_dueDateController,

decoration: InputDecoration(hintText: 'Due Date (YYYY-MM-DD)'),

),

DropdownButton<String>(

value: \_priority,

onChanged: (String? newValue) {

setState(() {

\_priority = newValue!;

});

},

items: <String>['Low', 'Medium', 'High']

.map<DropdownMenuItem<String>>((String value) {

return DropdownMenuItem<String>(

value: value,

child: Text(value),

);

}).toList(),

),

ElevatedButton(

onPressed: \_addTask,

child: Text('Add Task'),

),

Expanded(

child: ListView.builder(

itemCount: \_tasks.length,

itemBuilder: (context, index) {

final task = \_tasks[index];

return ListTile(

title: Text('${task['task']} - Due: ${task['dueDate']}'),

subtitle: Text('Priority: ${task['priority']}'),

trailing: IconButton(

icon: Icon(Icons.delete),

onPressed: () => \_deleteTask(index),

),

);

},

),

),

],

),

),

);

}

}