Program Code:

main.dart

import 'package:flutter/material.dart';

import 'screens/home\_screen.dart';

import 'screens/weather\_screen.dart';

import 'screens/tasks\_screen.dart';

import 'screens/contact\_admin\_screen.dart';

import 'screens/about\_screen.dart';

import 'screens/login\_screen.dart';

import 'screens/create\_account\_screen.dart';

import 'screens/logout\_screen.dart';

void main() => runApp(StudentConnectApp());

class StudentConnectApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Student Connect',

theme: ThemeData(

useMaterial3: true,

colorSchemeSeed: Colors.teal,

textTheme: Theme.of(context).textTheme.apply(fontFamily: 'Poppins'),

appBarTheme: AppBarTheme(

backgroundColor: const Color.fromARGB(255, 233, 125, 43),

foregroundColor: Colors.white,

titleTextStyle: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),

),

),

initialRoute: '/',

routes: {

'/': (context) => LoginScreen(),

'/create': (context) => CreateAccountScreen(),

'/home': (context) => HomeScreen(),

'/weather': (context) => WeatherScreen(),

'/tasks': (context) => TasksScreen(),

'/contact': (context) => ContactAdminScreen(),

'/about': (context) => AboutScreen(),

'/logout': (context) => LogoutScreen(),

},

);

}

}

about\_screen.dart

import 'package:flutter/material.dart';

class AboutScreen extends StatelessWidget {

const AboutScreen({super.key});

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

appBar: AppBar(title: Text('About')),

body: SingleChildScrollView(

padding: const EdgeInsets.all(24),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Student Connect',

style: theme.textTheme.headlineMedium?.copyWith(fontWeight: FontWeight.bold),

),

SizedBox(height: 12),

Text(

'Student Connect is a Flutter-based mobile app designed to streamline campus life for students at MSRIT. It offers a clean UI, intuitive navigation, and essential features to stay organized and informed.',

style: theme.textTheme.bodyLarge,

),

SizedBox(height: 24),

Divider(),

SizedBox(height: 12),

Text('Features', style: theme.textTheme.titleLarge),

SizedBox(height: 12),

\_buildFeatureTile(Icons.wb\_sunny, 'Weather Updates',

'Get daily weather forecasts for Bengaluru to plan your day better.'),

\_buildFeatureTile(Icons.task, 'Task Manager',

'Add, edit, and delete personal tasks to stay productive and organized.'),

\_buildFeatureTile(Icons.support\_agent, 'Contact Admin',

'Reach out to campus administration for queries or support.'),

\_buildFeatureTile(Icons.design\_services, 'Material 3 Design',

'Modern UI components with smooth transitions and adaptive styling.'),

\_buildFeatureTile(Icons.navigation, 'Named Navigation',

'Efficient screen transitions using Flutter’s named routes.'),

SizedBox(height: 24),

Divider(),

SizedBox(height: 12),

Text('About MSRIT', style: theme.textTheme.titleLarge),

SizedBox(height: 12),

Text(

'M.S. Ramaiah Institute of Technology (MSRIT) is one of India’s premier engineering colleges, located in Bengaluru. Known for academic excellence, innovation, and vibrant campus life, MSRIT fosters a culture of learning and leadership.',

style: theme.textTheme.bodyLarge,

),

SizedBox(height: 24),

Center(

child: Text(

'Made with ❤️ in Flutter',

style: theme.textTheme.bodyMedium?.copyWith(color: theme.colorScheme.primary),

),

),

],

),

),

);

}

Widget \_buildFeatureTile(IconData icon, String title, String subtitle) {

return ListTile(

leading: Icon(icon, size: 32, color: Colors.teal),

title: Text(title, style: TextStyle(fontWeight: FontWeight.bold)),

subtitle: Text(subtitle),

contentPadding: EdgeInsets.symmetric(vertical: 4),

);

}

}

contact\_admin\_screen.dart

import 'package:flutter/material.dart';

import 'package:url\_launcher/url\_launcher.dart';

class ContactAdminScreen extends StatefulWidget {

@override

State<ContactAdminScreen> createState() => \_ContactAdminScreenState();

}

class \_ContactAdminScreenState extends State<ContactAdminScreen> {

final nameController = TextEditingController();

final emailController = TextEditingController();

final subjectController = TextEditingController();

final messageController = TextEditingController();

void sendEmail() async {

final uri = Uri(

scheme: 'mailto',

path: 'admin@example.com',

query: 'subject=${subjectController.text}&body=From: ${nameController.text} (${emailController.text})\n\n${messageController.text}',

);

if (await canLaunchUrl(uri)) {

await launchUrl(uri);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Contact Admin')),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(children: [

TextField(controller: nameController, decoration: InputDecoration(labelText: 'Your Name')),

TextField(controller: emailController, decoration: InputDecoration(labelText: 'Your Email')),

TextField(controller: subjectController, decoration: InputDecoration(labelText: 'Subject')),

TextField(controller: messageController, decoration: InputDecoration(labelText: 'Message')),

SizedBox(height: 20),

ElevatedButton(onPressed: sendEmail, child: Text('Send')),

]),

),

);

}

}

create\_account\_screen.dart

import 'package:flutter/material.dart';

class CreateAccountScreen extends StatefulWidget {

@override

State<CreateAccountScreen> createState() => \_CreateAccountScreenState();

}

class \_CreateAccountScreenState extends State<CreateAccountScreen> {

final nameController = TextEditingController();

final dobController = TextEditingController();

final branchController = TextEditingController();

final yearController = TextEditingController();

final emailController = TextEditingController();

String selectedCity = 'Bengaluru';

final cities = ['Bengaluru', 'Mumbai', 'Delhi', 'Chennai', 'Hyderabad'];

void createAccount() {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Account created for ${nameController.text}')),

);

Navigator.pop(context);

}

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

appBar: AppBar(title: Text('Create Account')),

body: SingleChildScrollView(

padding: const EdgeInsets.symmetric(horizontal: 32, vertical: 24),

child: Column(

children: [

Text('Let’s Get You Started', style: theme.textTheme.headlineMedium),

SizedBox(height: 24),

\_buildTextField(nameController, 'Name'),

\_buildTextField(dobController, 'Date of Birth', hint: 'DD/MM/YYYY'),

\_buildTextField(branchController, 'Branch'),

\_buildTextField(yearController, 'Year of Study'),

\_buildTextField(emailController, 'Email'),

SizedBox(height: 16),

DropdownButtonFormField<String>(

value: selectedCity,

items: cities.map((city) => DropdownMenuItem(value: city, child: Text(city))).toList(),

onChanged: (val) => setState(() => selectedCity = val!),

decoration: InputDecoration(

labelText: 'Place',

filled: true,

border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),

),

),

SizedBox(height: 24),

ElevatedButton.icon(

onPressed: createAccount,

icon: Icon(Icons.check\_circle),

label: Text('Create Account'),

style: ElevatedButton.styleFrom(

minimumSize: Size(double.infinity, 48),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

),

),

],

),

),

);

}

Widget \_buildTextField(TextEditingController controller, String label, {String? hint}) {

return Padding(

padding: const EdgeInsets.only(bottom: 16),

child: TextField(

controller: controller,

decoration: InputDecoration(

labelText: label,

hintText: hint,

filled: true,

border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),

),

),

);

}

}

home\_screen.dart

import 'package:flutter/material.dart';

import 'package:lottie/lottie.dart';

import '../widgets/drawer\_widget.dart';

class HomeScreen extends StatelessWidget {

final List<\_FeatureCard> features = [

\_FeatureCard('Weather', Icons.sunny, '/weather', Colors.orange),

\_FeatureCard('Tasks', Icons.check\_circle, '/tasks', Colors.green),

\_FeatureCard('Contact Admin', Icons.support\_agent, '/contact', Colors.blue),

\_FeatureCard('About', Icons.info\_outline, '/about', Colors.purple),

];

@override

Widget build(BuildContext context) {

return Scaffold(

drawer: AppDrawer(),

appBar: AppBar(

title: Text('Student Connect'),

backgroundColor: Colors.teal,

),

body: Stack(

children: [

// 🖼️ Background Image

Positioned.fill(

child: Image.asset(

'bg.jpg', // Make sure this image exists in your assets folder

fit: BoxFit.cover,

),

),

// 🌫️ Overlay for readability

Container(color: Colors.black.withValues(alpha: 0.05)),

// 🧱 Main Content

Column(

children: [

\_buildBanner(),

SizedBox(height: 16),

Expanded(

child: GridView.count(

crossAxisCount: 2,

padding: EdgeInsets.all(16),

crossAxisSpacing: 16,

mainAxisSpacing: 16,

children: features.map((f) => \_buildFeatureCard(context, f)).toList(),

),

),

],

),

],

),

);

}

// 🎉 Welcome Banner with Lottie

Widget \_buildBanner() {

return Container(

height: 180,

width: double.infinity,

decoration: BoxDecoration(

gradient: LinearGradient(colors: [Colors.teal, Colors.deepPurple]),

borderRadius: BorderRadius.only(

bottomLeft: Radius.circular(24),

bottomRight: Radius.circular(24),

),

),

child: Stack(

children: [

Positioned(

right: 0,

bottom: 0,

child: Lottie.asset('assets/student.json', height: 140),

),

Padding(

padding: const EdgeInsets.all(24.0),

child: Text(

'Welcome to Student Connect 👋',

style: TextStyle(

color: Colors.white,

fontSize: 22,

fontWeight: FontWeight.bold,

),

),

),

],

),

);

}

// 🧱 Feature Card

Widget \_buildFeatureCard(BuildContext context, \_FeatureCard feature) {

return GestureDetector(

onTap: () => Navigator.pushNamed(context, feature.route),

child: Card(

elevation: 6,

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(16)),

child: Container(

padding: EdgeInsets.all(16),

decoration: BoxDecoration(

color: feature.color.withValues(alpha: 0.1),

borderRadius: BorderRadius.circular(16),

),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Icon(feature.icon, size: 40, color: feature.color),

SizedBox(height: 12),

Text(feature.label, style: TextStyle(fontSize: 16)),

],

),

),

),

);

}

}

// 📦 Feature Card Model

class \_FeatureCard {

final String label;

final IconData icon;

final String route;

final Color color;

\_FeatureCard(this.label, this.icon, this.route, this.color);

}

login\_screen.dart

import 'package:flutter/material.dart';

class LoginScreen extends StatefulWidget {

@override

State<LoginScreen> createState() => \_LoginScreenState();

}

class \_LoginScreenState extends State<LoginScreen> {

final emailController = TextEditingController();

final dobController = TextEditingController();

void login() {

// Optional: validate email and DOB before navigating

if (emailController.text.isNotEmpty && dobController.text.isNotEmpty) {

Navigator.pushReplacementNamed(context, '/home');

} else {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Please enter both email and date of birth')),

);

}

}

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

backgroundColor: theme.colorScheme.surface,

body: Center(

child: SingleChildScrollView(

padding: const EdgeInsets.symmetric(horizontal: 32, vertical: 24),

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

CircleAvatar(

radius: 50,

backgroundColor: theme.colorScheme.primaryContainer,

child: Icon(Icons.person, size: 50, color: theme.colorScheme.onPrimaryContainer),

),

SizedBox(height: 24),

Text('Welcome Back', style: theme.textTheme.headlineMedium),

SizedBox(height: 16),

TextField(

controller: emailController,

decoration: InputDecoration(

labelText: 'Email',

filled: true,

border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),

),

),

SizedBox(height: 16),

TextField(

controller: dobController,

decoration: InputDecoration(

labelText: 'Date of Birth',

hintText: 'DD/MM/YYYY',

filled: true,

border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),

),

),

SizedBox(height: 24),

ElevatedButton.icon(

onPressed: login,

icon: Icon(Icons.login),

label: Text('Login'),

style: ElevatedButton.styleFrom(

minimumSize: Size(double.infinity, 48),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

),

),

SizedBox(height: 12),

OutlinedButton(

onPressed: () => Navigator.pushNamed(context, '/create'),

child: Text('Create Account'),

style: OutlinedButton.styleFrom(

minimumSize: Size(double.infinity, 48),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

),

),

],

),

),

),

);

}

}

logout\_screen.dart

import 'package:flutter/material.dart';

class LogoutScreen extends StatelessWidget {

const LogoutScreen({super.key});

void \_logout(BuildContext context) {

// Replace current screen with login

Navigator.pushReplacementNamed(context, '/');

}

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

appBar: AppBar(title: Text('Logout')),

body: Center(

child: Padding(

padding: const EdgeInsets.all(24.0),

child: Column(

mainAxisSize: MainAxisSize.min,

children: [

Icon(Icons.logout, size: 64, color: theme.colorScheme.primary),

SizedBox(height: 24),

Text('Are you sure you want to logout?', style: theme.textTheme.titleMedium),

SizedBox(height: 24),

ElevatedButton.icon(

onPressed: () => \_logout(context),

icon: Icon(Icons.check),

label: Text('Yes, Logout'),

style: ElevatedButton.styleFrom(

minimumSize: Size(double.infinity, 48),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

),

),

SizedBox(height: 12),

OutlinedButton(

onPressed: () => Navigator.pop(context),

child: Text('Cancel'),

style: OutlinedButton.styleFrom(

minimumSize: Size(double.infinity, 48),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

),

),

],

),

),

),

);

}

}

tasks\_screen.dart

import 'package:flutter/material.dart';

import '../services/storage\_service.dart';

import '../widgets/task\_tile.dart';

class TasksScreen extends StatefulWidget {

@override

State<TasksScreen> createState() => \_TasksScreenState();

}

class \_TasksScreenState extends State<TasksScreen> {

List<Map<String, String>> tasks = [];

@override

void initState() {

super.initState();

loadTasks();

}

Future<void> loadTasks() async {

tasks = await StorageService.getTasks();

setState(() {});

}

Future<void> addTask(String title, String note) async {

await StorageService.addTask(title, note);

await loadTasks();

}

Future<void> deleteTask(int index) async {

await StorageService.deleteTask(index);

await loadTasks();

}

Future<void> editTask(int index, String newTitle, String newNote) async {

await StorageService.editTask(index, newTitle, newNote);

await loadTasks();

}

void showTaskDialog({int? index}) {

String title = index != null ? tasks[index]['title']! : '';

String note = index != null ? tasks[index]['note']! : '';

showDialog(

context: context,

builder: (\_) => AlertDialog(

title: Text(index == null ? 'Add Task' : 'Edit Task'),

content: Column(

mainAxisSize: MainAxisSize.min,

children: [

TextField(

controller: TextEditingController(text: title),

onChanged: (val) => title = val,

decoration: InputDecoration(labelText: 'Title'),

),

TextField(

controller: TextEditingController(text: note),

onChanged: (val) => note = val,

decoration: InputDecoration(labelText: 'Note'),

),

],

),

actions: [

TextButton(onPressed: () => Navigator.pop(context), child: Text('Cancel')),

ElevatedButton(

onPressed: () {

if (index == null) {

addTask(title, note);

} else {

editTask(index, title, note);

}

Navigator.pop(context);

},

child: Text(index == null ? 'Save' : 'Update'),

),

],

),

);

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Tasks')),

body: tasks.isEmpty

? Center(child: Text('No tasks yet. Tap + to add one.'))

: ListView.builder(

itemCount: tasks.length,

itemBuilder: (\_, i) => Dismissible(

key: Key('$i'),

direction: DismissDirection.endToStart,

background: Container(

color: Colors.red,

alignment: Alignment.centerRight,

padding: EdgeInsets.symmetric(horizontal: 20),

child: Icon(Icons.delete, color: Colors.white),

),

onDismissed: (\_) => deleteTask(i),

child: TaskTile(

title: tasks[i]['title']!,

note: tasks[i]['note']!,

onEdit: () => showTaskDialog(index: i),

onDelete: () => deleteTask(i),

)

),

),

floatingActionButton: FloatingActionButton(

onPressed: () => showTaskDialog(),

child: Icon(Icons.add),

),

);

}

}

weather\_screen.dart

import 'package:flutter/material.dart';

import '../services/weather\_service.dart';

class WeatherScreen extends StatelessWidget {

const WeatherScreen({super.key});

IconData \_getWeatherIcon(String forecast) {

forecast = forecast.toLowerCase();

if (forecast.contains('sunny')) return Icons.wb\_sunny;

if (forecast.contains('rain')) return Icons.beach\_access;

if (forecast.contains('cloud')) return Icons.cloud;

if (forecast.contains('storm')) return Icons.flash\_on;

if (forecast.contains('snow')) return Icons.ac\_unit;

return Icons.wb\_cloudy;

}

@override

Widget build(BuildContext context) {

final theme = Theme.of(context);

return Scaffold(

appBar: AppBar(title: Text('Weather Forecast')),

body: FutureBuilder<List<String>>(

future: WeatherService.getForecast(),

builder: (context, snapshot) {

if (snapshot.connectionState == ConnectionState.waiting)

return Center(child: CircularProgressIndicator());

if (snapshot.hasError || !snapshot.hasData || snapshot.data!.isEmpty)

return Center(child: Text('Failed to load weather', style: theme.textTheme.bodyLarge));

return ListView.separated(

padding: EdgeInsets.all(16),

itemCount: snapshot.data!.length,

separatorBuilder: (\_, \_\_) => SizedBox(height: 12),

itemBuilder: (context, index) {

final forecast = snapshot.data![index];

final icon = \_getWeatherIcon(forecast);

return Card(

elevation: 3,

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),

child: ListTile(

leading: Icon(icon, size: 32, color: theme.colorScheme.primary),

title: Text(forecast, style: theme.textTheme.titleMedium),

subtitle: Text('Day ${index + 1}', style: theme.textTheme.bodySmall),

),

);

},

);

},

),

);

}

}

storage\_device.dart

import 'dart:convert';

import 'package:shared\_preferences/shared\_preferences.dart';

class StorageService {

static const String \_key = 'tasks';

/// Retrieve all tasks from local storage

static Future<List<Map<String, String>>> getTasks() async {

final prefs = await SharedPreferences.getInstance();

final jsonString = prefs.getString(\_key);

if (jsonString == null) return [];

final List decoded = jsonDecode(jsonString);

return decoded.map((e) => Map<String, String>.from(e)).toList();

}

/// Save the updated task list to local storage

static Future<void> \_saveTasks(List<Map<String, String>> tasks) async {

final prefs = await SharedPreferences.getInstance();

final jsonString = jsonEncode(tasks);

await prefs.setString(\_key, jsonString);

}

/// Add a new task

static Future<void> addTask(String title, String note) async {

final tasks = await getTasks();

tasks.add({'title': title, 'note': note});

await \_saveTasks(tasks);

}

/// Delete a task by index

static Future<void> deleteTask(int index) async {

final tasks = await getTasks();

if (index >= 0 && index < tasks.length) {

tasks.removeAt(index);

await \_saveTasks(tasks);

}

}

/// Edit a task by index

static Future<void> editTask(int index, String newTitle, String newNote) async {

final tasks = await getTasks();

if (index >= 0 && index < tasks.length) {

tasks[index] = {'title': newTitle, 'note': newNote};

await \_saveTasks(tasks);

}

}

}

weather\_service.dart

mport 'dart:convert';

import 'package:http/http.dart' as http;

class WeatherService {

static Future<List<String>> getForecast() async {

final url = Uri.parse(

'https://api.open-meteo.com/v1/forecast?latitude=12.97&longitude=77.59&daily=temperature\_2m\_max,temperature\_2m\_min,weathercode&timezone=auto'

);

try {

final response = await http.get(url);

if (response.statusCode == 200) {

final data = jsonDecode(response.body);

final dates = data['daily']['time'];

final maxTemps = data['daily']['temperature\_2m\_max'];

final minTemps = data['daily']['temperature\_2m\_min'];

final weatherCodes = data['daily']['weathercode'];

List<String> forecast = [];

for (int i = 0; i < dates.length; i++) {

final condition = \_mapWeatherCode(weatherCodes[i]);

forecast.add(

'${dates[i]}: $condition • Max: ${maxTemps[i]}°C • Min: ${minTemps[i]}°C'

);

}

return forecast;

} else {

throw Exception('Failed to load weather data');

}

} catch (e) {

print('Error fetching weather: $e');

return ['Unable to fetch weather data'];

}

}

static String \_mapWeatherCode(int code) {

// Simplified mapping based on Open-Meteo weather codes

if (code == 0) return 'Clear';

if (code == 1 || code == 2) return 'Partly Cloudy';

if (code == 3) return 'Overcast';

if (code >= 45 && code <= 48) return 'Fog';

if (code >= 51 && code <= 67) return 'Drizzle';

if (code >= 71 && code <= 77) return 'Snow';

if (code >= 80 && code <= 82) return 'Showers';

if (code >= 95) return 'Thunderstorm';

return 'Unknown';

}

}

drawer\_widget.dart

import 'package:flutter/material.dart';

class AppDrawer extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Drawer(

child: ListView(

padding: EdgeInsets.zero,

children: [

DrawerHeader(

decoration: BoxDecoration(

gradient: LinearGradient(colors: [Colors.teal, Colors.deepPurple]),

),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

CircleAvatar(radius: 30, backgroundColor: Colors.white,backgroundImage: AssetImage('person.jpg'),),

SizedBox(height: 10),

Text('Welcome, Student', style: TextStyle(color: Colors.white, fontSize: 16)),

],

),

),

\_buildDrawerItem(context, Icons.home, 'Home', '/home'),

\_buildDrawerItem(context, Icons.cloud, 'Weather', '/weather'),

\_buildDrawerItem(context, Icons.task, 'Tasks', '/tasks'),

\_buildDrawerItem(context, Icons.email, 'Contact Admin', '/contact'),

\_buildDrawerItem(context, Icons.info, 'About', '/about'),

Divider(),

\_buildDrawerItem(context, Icons.logout, 'Logout', '/logout'),

],

),

);

}

Widget \_buildDrawerItem(BuildContext context, IconData icon, String label, String route) {

return ListTile(

leading: Icon(icon),

title: Text(label),

onTap: () {

Navigator.pop(context); // Close drawer

Navigator.pushNamed(context, route);

},

);

}

}

task\_tile.dart

import 'package:flutter/material.dart';

class TaskTile extends StatelessWidget {

final String title;

final String note;

final VoidCallback onEdit;

final VoidCallback onDelete;

const TaskTile({

required this.title,

required this.note,

required this.onEdit,

required this.onDelete,

Key? key,

}) : super(key: key);

@override

Widget build(BuildContext context) {

return Card(

margin: EdgeInsets.symmetric(horizontal: 12, vertical: 6),

elevation: 3,

child: ListTile(

title: Text(title, style: TextStyle(fontWeight: FontWeight.bold)),

subtitle: Text(note),

trailing: Wrap(

spacing: 8,

children: [

IconButton(

icon: Icon(Icons.edit, color: Colors.teal),

onPressed: onEdit,

tooltip: 'Edit Task',

),

IconButton(

icon: Icon(Icons.delete, color: Colors.red),

onPressed: onDelete,

tooltip: 'Delete Task',

),

],

),

),

);

}

}

Output

















