Flutter Assesment question and desired out to get

Date:11/08/2025

Title: Student Connect – Multi-Screen Flutter App with Named Navigation, API Integration & Local Storage Scenario: Your college wants a small mobile app to help students check weather updates, manage their personal tasks, and contact the admin. You can check the demo of the Flutter application here: ⍰ ⍰ https://brunda-contact-app.web.app/#/home Note: This is just an example to help you understand the features and flow of the application. You are not limited to this exact design.

Requirements:

1. Features

● Must have at least 15 widgets in total (e.g., Text, Image, Drawer, ListView, Card, TextField, Button, Slider, Switch, etc.).

● Home Screen with Drawer Widget for navigation. ● Minimum 5–10 screens, including:

1. Home Screen

2. Weather Screen (External API)

3. Tasks Screen (CRUD operations with local storage)

4. Contact Admin Form (Profile Screen)

5. About Screen

● Named Navigation must be used for moving between screens.

● At least one screen must take user input (Contact Admin form).

● CRUD functionality: Add, Edit, Delete tasks.

● Background color and banner image must be used on the Home Screen. 2. Storage

● Use SharedPreferences for storing and retrieving data locally inside the app. ● Store the task list in local storage so that added tasks remain even after the app is closed and reopened.

3. Design

● Consistent color scheme (e.g., Indigo shades).

● Use Material 3 design style. ● Keep UI clean and mobile-friendly.

4. External API

● Integrate the Open-Meteo API to display the current temperature for Bengaluru in the Weather Screen. Expected Output: The finished app should allow:

● Navigating between Home, Weather, Tasks, Contact Admin, and About screens using a Drawer.

● Viewing live weather data via API.

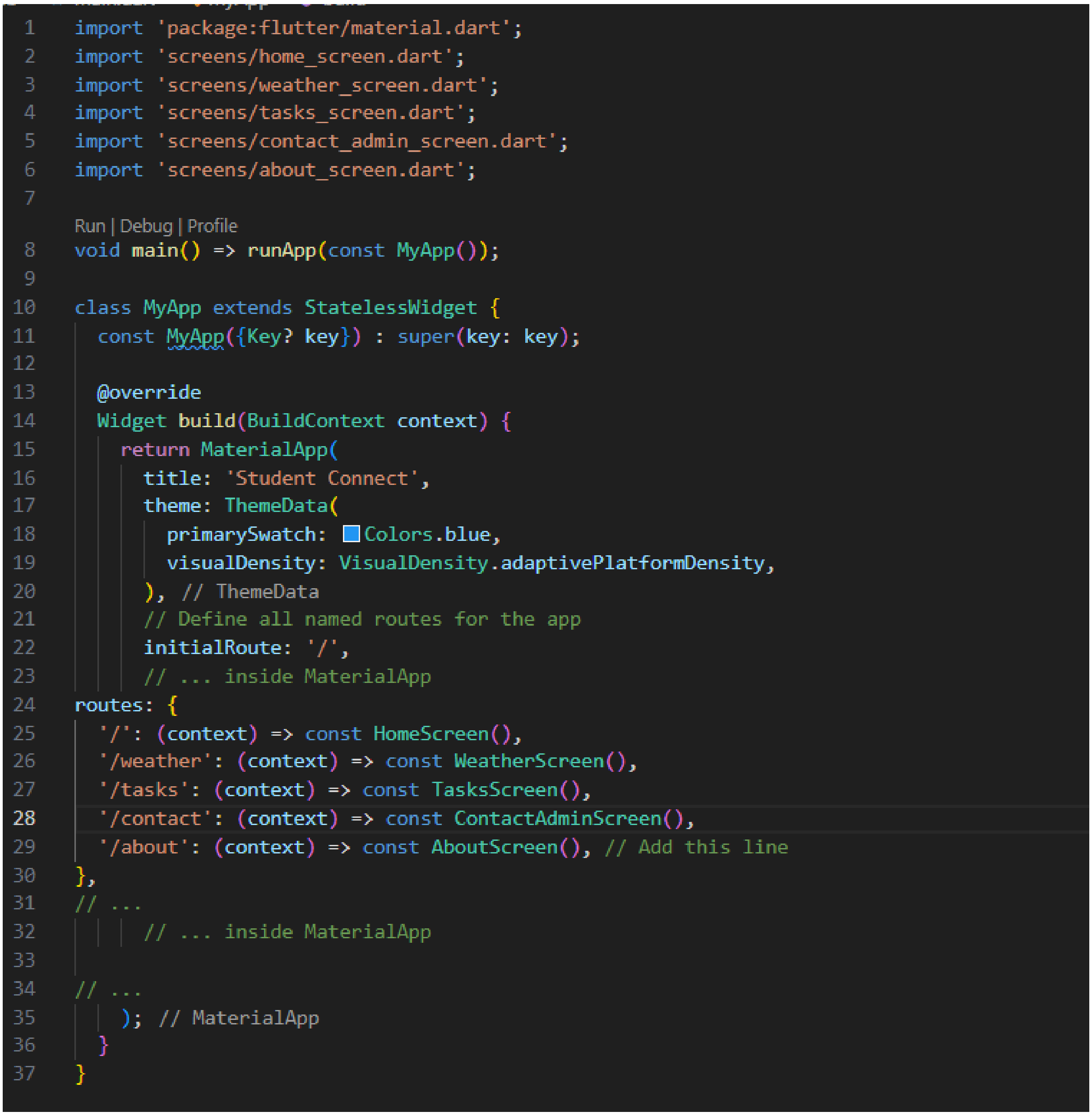
● Adding, editing, and deleting tasks with data stored locally in SharedPreferences.

● Filling and submitting a Contact Admin form, which opens the mail app with the details pre-filled.

● Viewing app information and features in the About screen.

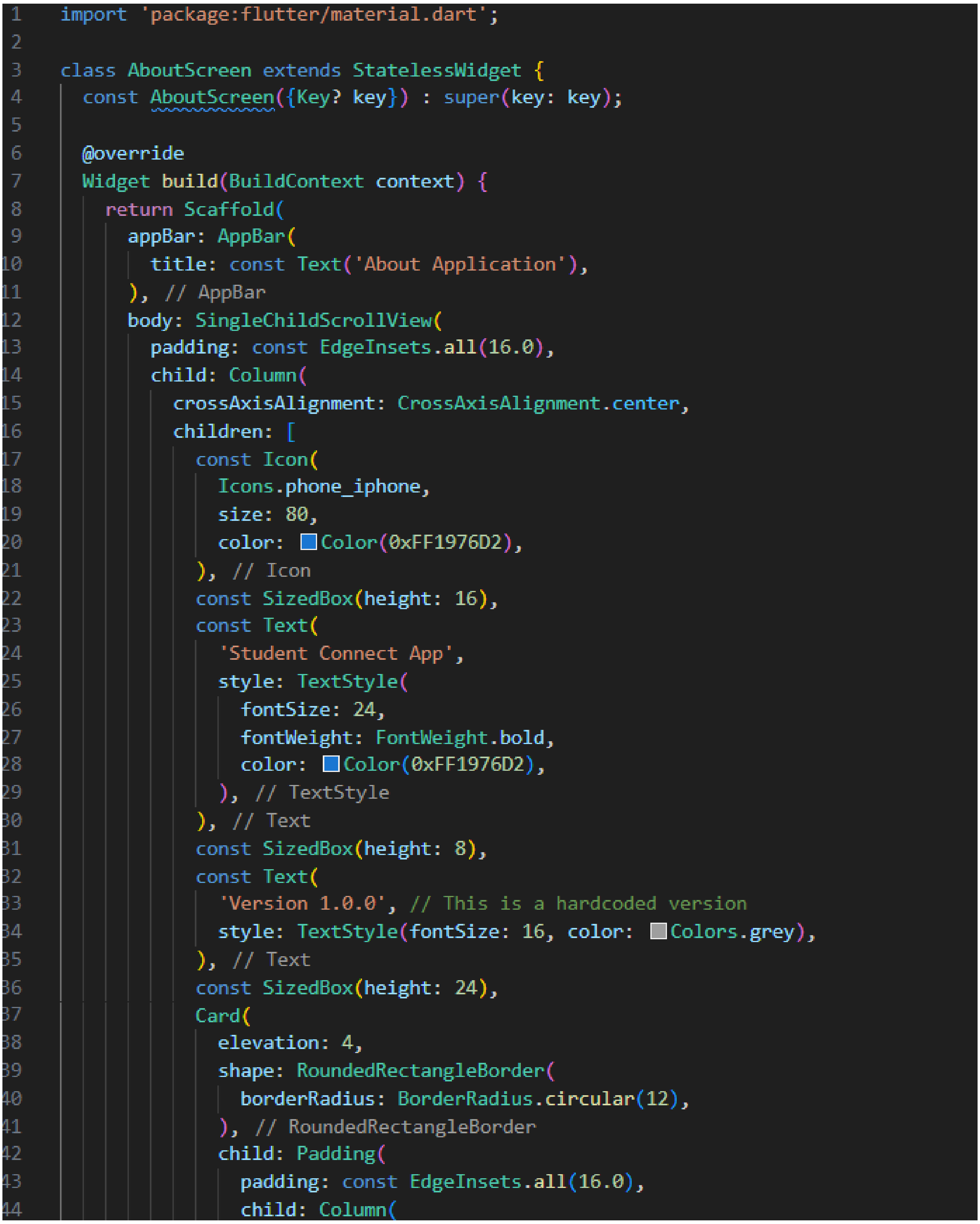
Program:-

In main.dart file

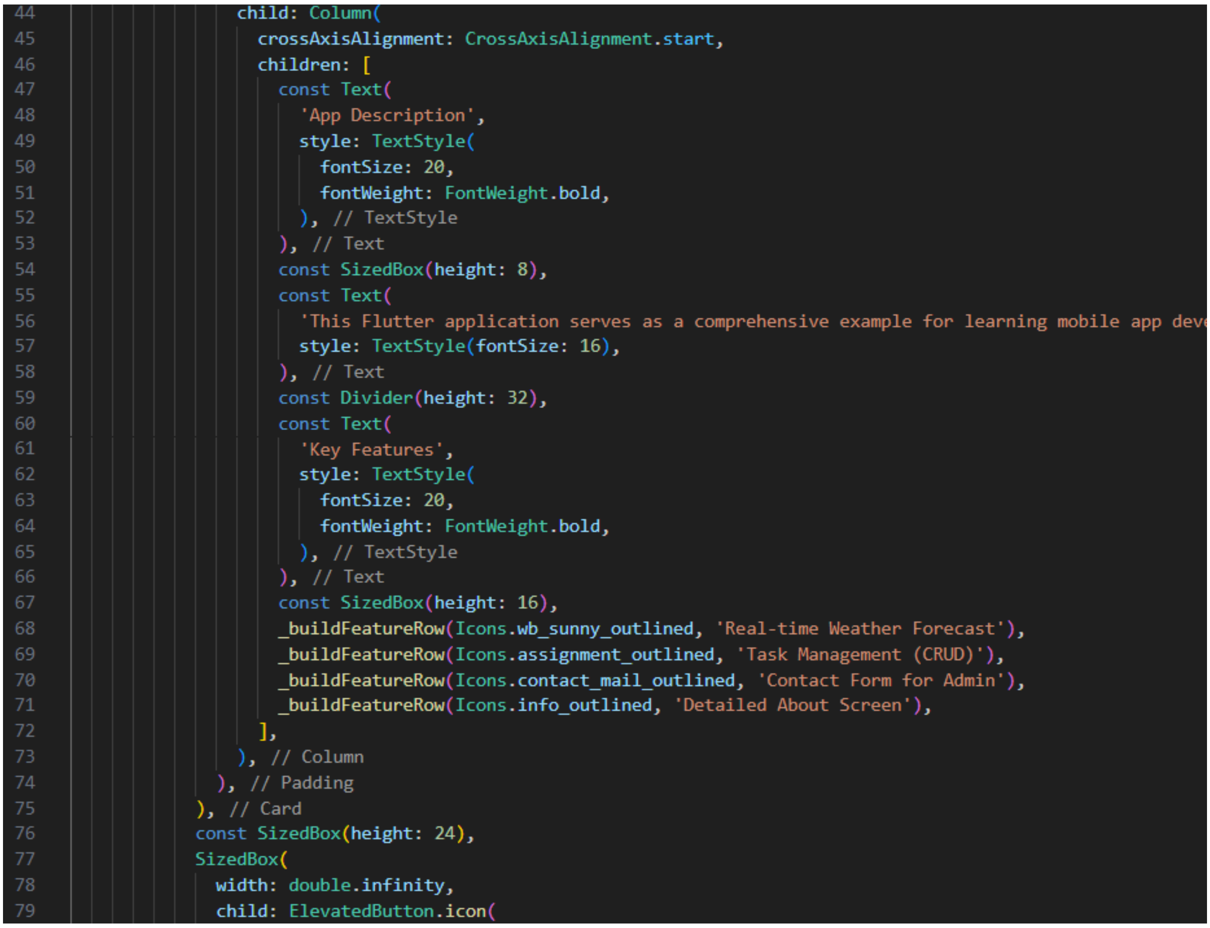


```dart
import 'package:flutter/material.dart';
import 'screens/home_screen.dart';
import 'screens/weather_screen.dart';
import 'screens/tasks_screen.dart';
import 'screens/contact_admin_screen.dart';
import 'screens/about_screen.dart';

Run | Debug | Profile
void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Student Connect',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ), // ThemeData
      // Define all named routes for the app
      initialRoute: '/',
      // ... inside MaterialApp
routes: {
  '/': (context) => const HomeScreen(),
  '/weather': (context) => const WeatherScreen(),
  '/tasks': (context) => const TasksScreen(),
  '/contact': (context) => const ContactAdminScreen(),
  '/about': (context) => const AboutScreen(), // Add this line
},
// ...
      // ... inside MaterialApp

// ...
    ); // MaterialApp
  }
}
```

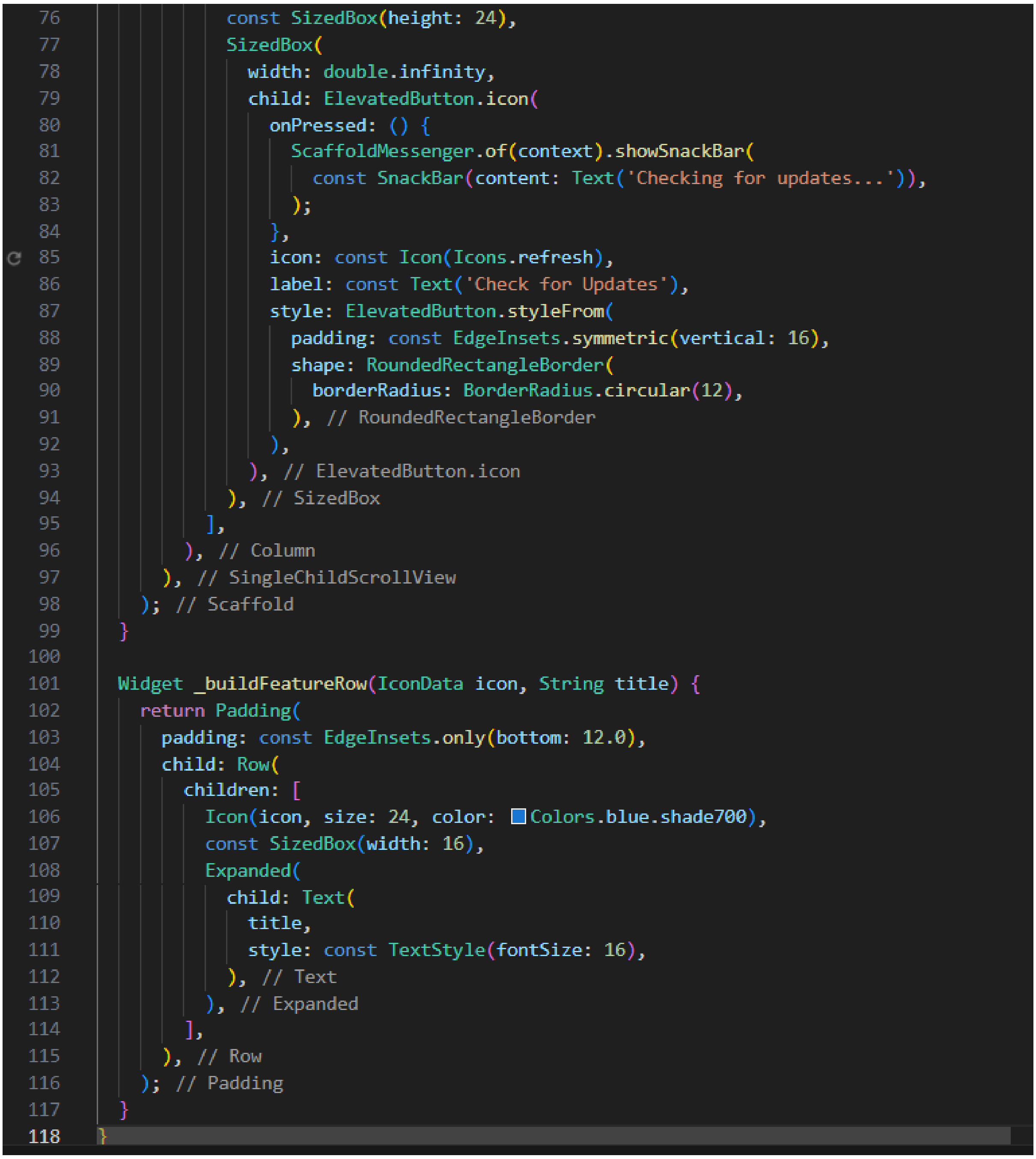
In about_screen.dart :-



```
import 'package:flutter/material.dart';

class AboutScreen extends StatelessWidget {
  const AboutScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('About Application'),
      ), // AppBar
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            const Icon(
              Icons.phone_iphone,
              size: 80,
              color: Color(0xFF1976D2),
            ), // Icon
            const SizedBox(height: 16),
            const Text(
              'Student Connect App',
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
                color: Color(0xFF1976D2),
              ), // TextStyle
            ), // Text
            const SizedBox(height: 8),
            const Text(
              'Version 1.0.0', // This is a hardcoded version
              style: TextStyle(fontSize: 16, color: Colors.grey),
            ), // Text
            const SizedBox(height: 24),
            Card(
              elevation: 4,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(12),
              ), // RoundedRectangleBorder
              child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
```

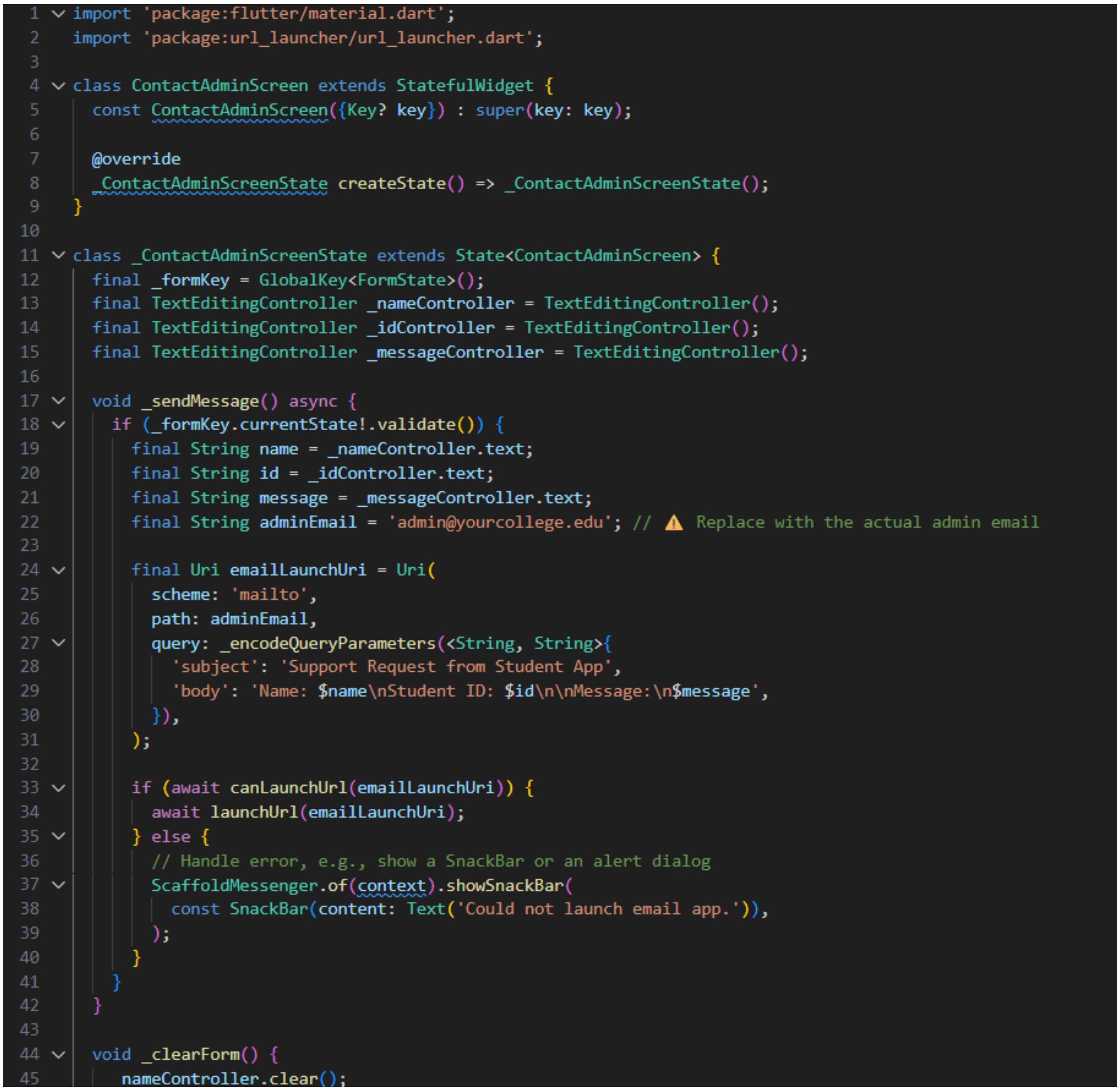


```
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    const Text(
                      'App Description',
                      style: TextStyle(
                        fontSize: 20,
                        fontWeight: FontWeight.bold,
                      ), // TextStyle
                    ), // Text
                    const SizedBox(height: 8),
                    const Text(
                      'This Flutter application serves as a comprehensive example for learning mobile app deve
                      style: TextStyle(fontSize: 16),
                    ), // Text
                    const Divider(height: 32),
                    const Text(
                      'Key Features',
                      style: TextStyle(
                        fontSize: 20,
                        fontWeight: FontWeight.bold,
                      ), // TextStyle
                    ), // Text
                    const SizedBox(height: 16),
                    _buildFeatureRow(Icons.wb_sunny_outlined, 'Real-time Weather Forecast'),
                    _buildFeatureRow(Icons.assignment_outlined, 'Task Management (CRUD)'),
                    _buildFeatureRow(Icons.contact_mail_outlined, 'Contact Form for Admin'),
                    _buildFeatureRow(Icons.info_outlined, 'Detailed About Screen'),
                  ],
                ), // Column
              ), // Padding
            ), // Card
            const SizedBox(height: 24),
            SizedBox(
              width: double.infinity,
              child: ElevatedButton.icon(
```

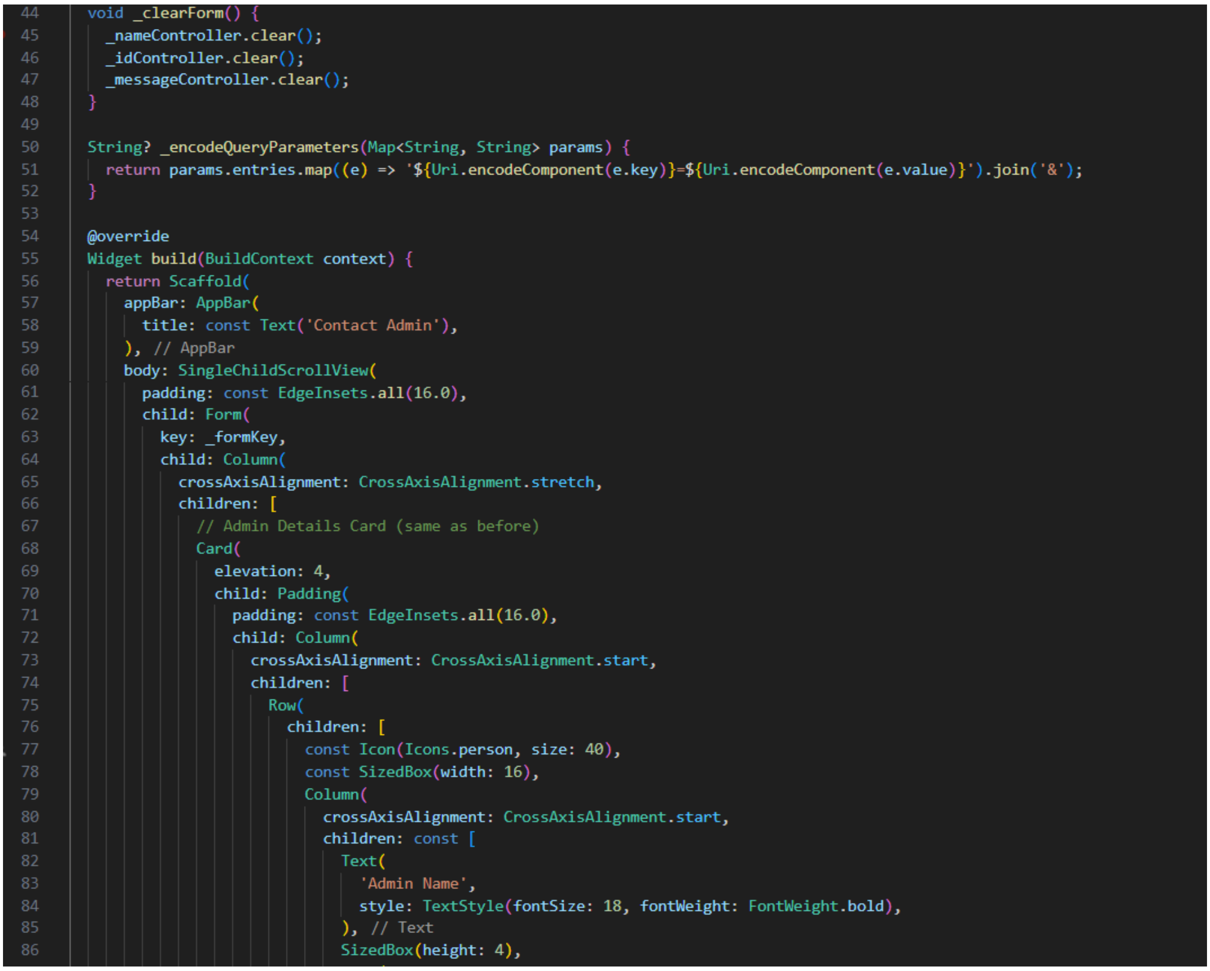


```
            const SizedBox(height: 24),
            SizedBox(
              width: double.infinity,
              child: ElevatedButton.icon(
                onPressed: () {
                  ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Checking for updates...')),
                  );
                },
                icon: const Icon(Icons.refresh),
                label: const Text('Check for Updates'),
                style: ElevatedButton.styleFrom(
                  padding: const EdgeInsets.symmetric(vertical: 16),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ), // RoundedRectangleBorder
                ),
              ), // ElevatedButton.icon
            ), // SizedBox
          ],
        ), // Column
      ), // SingleChildScrollView
    ); // Scaffold
  }

  Widget _buildFeatureRow(IconData icon, String title) {
    return Padding(
      padding: const EdgeInsets.only(bottom: 12.0),
      child: Row(
        children: [
          Icon(icon, size: 24, color: Colors.blue.shade700),
          const SizedBox(width: 16),
          Expanded(
            child: Text(
              title,
              style: const TextStyle(fontSize: 16),
            ), // Text
          ), // Expanded
        ],
      ), // Row
    ); // Padding
  }
}
```

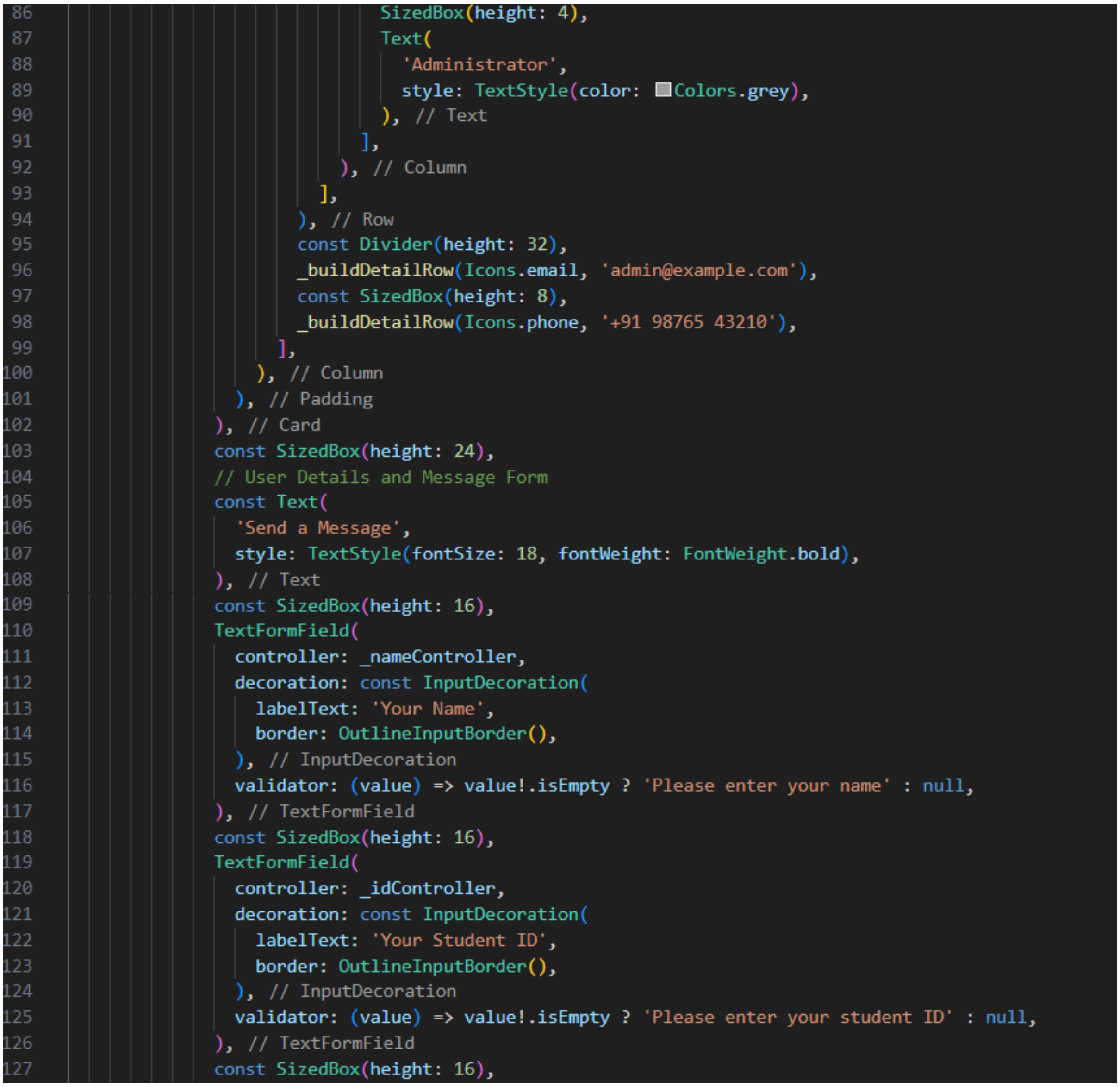
In contact_admin_screen.dart file :-



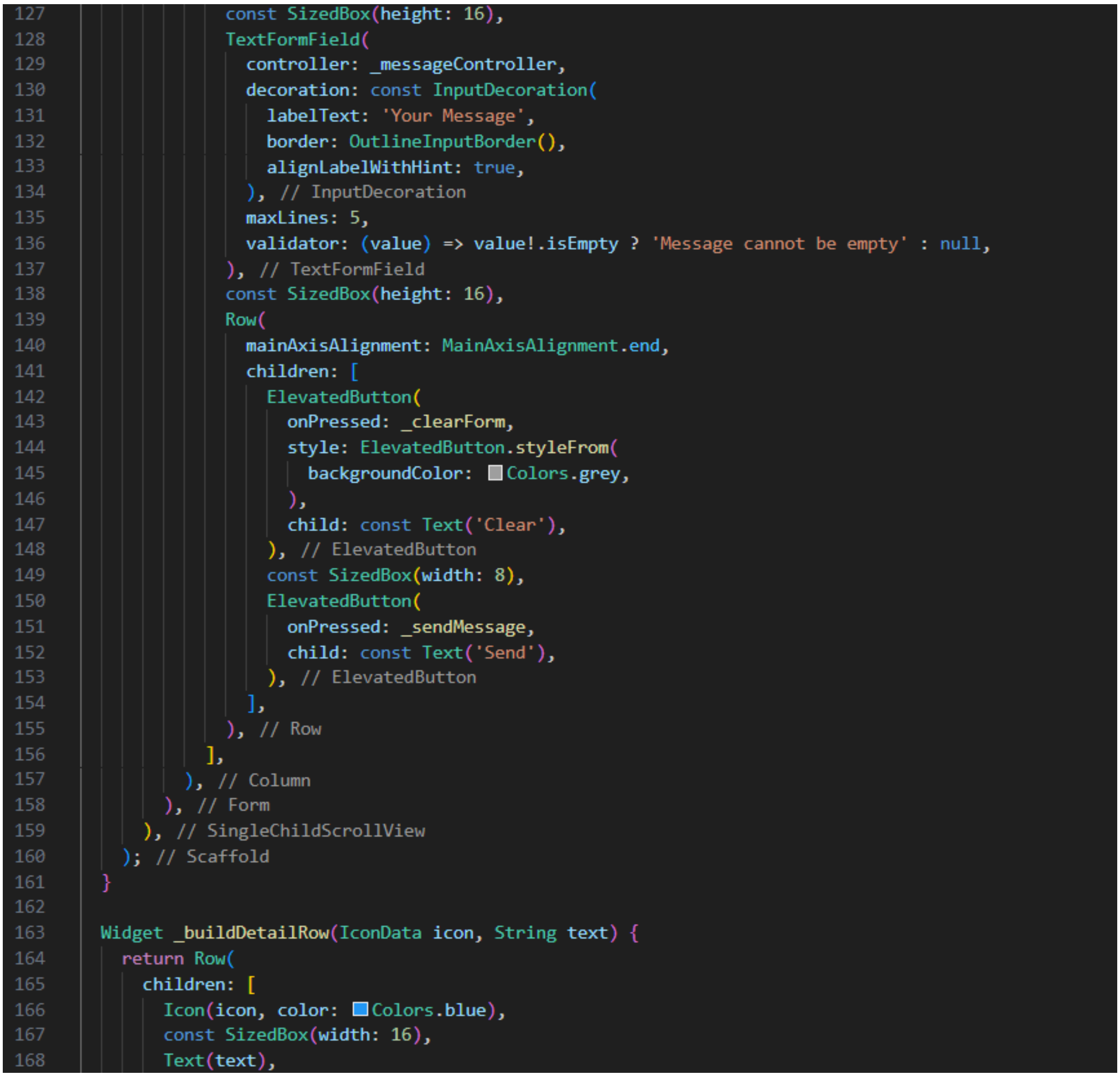
```dart
import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';

class ContactAdminScreen extends StatefulWidget {
  const ContactAdminScreen({Key? key}) : super(key: key);

  @override
  _ContactAdminScreenState createState() => _ContactAdminScreenState();
}

class _ContactAdminScreenState extends State<ContactAdminScreen> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _idController = TextEditingController();
  final TextEditingController _messageController = TextEditingController();

  void _sendMessage() async {
    if (_formKey.currentState!.validate()) {
      final String name = _nameController.text;
      final String id = _idController.text;
      final String message = _messageController.text;
      final String adminEmail = 'admin@yourcollege.edu'; // ⚠ Replace with the actual admin email

      final Uri emailLaunchUri = Uri(
        scheme: 'mailto',
        path: adminEmail,
        query: _encodeQueryParameters(<String, String>{
          'subject': 'Support Request from Student App',
          'body': 'Name: $name\nStudent ID: $id\n\nMessage:\n$message',
        }),
      );

      if (await canLaunchUrl(emailLaunchUri)) {
        await launchUrl(emailLaunchUri);
      } else {
        // Handle error, e.g., show a SnackBar or an alert dialog
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('Could not launch email app.')),
        );
      }
    }
  }

  void _clearForm() {
     nameController.clear();
```



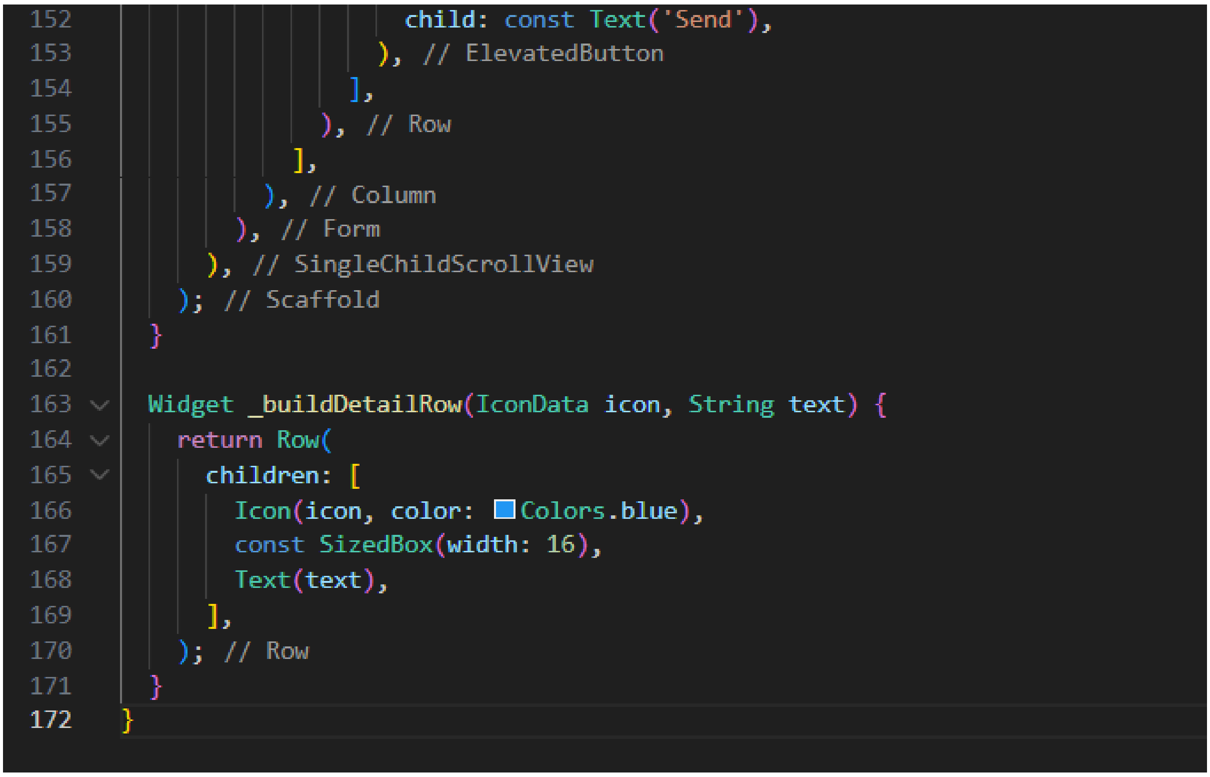
```dart
  void _clearForm() {
    _nameController.clear();
    _idController.clear();
    _messageController.clear();
  }

  String? _encodeQueryParameters(Map<String, String> params) {
    return params.entries.map((e) => '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}').join('&');
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Contact Admin'),
      ), // AppBar
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              // Admin Details Card (same as before)
              Card(
                elevation: 4,
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Row(
                        children: [
                          const Icon(Icons.person, size: 40),
                          const SizedBox(width: 16),
                          Column(
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: const [
                              Text(
                                'Admin Name',
                                style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                              ), // Text
                              SizedBox(height: 4),
```



```
                                SizedBox(height: 4),
                                Text(
                                  'Administrator',
                                  style: TextStyle(color: ■Colors.grey),
                                ), // Text
                              ],
                            ), // Column
                          ],
                        ), // Row
                        const Divider(height: 32),
                        _buildDetailRow(Icons.email, 'admin@example.com'),
                        const SizedBox(height: 8),
                        _buildDetailRow(Icons.phone, '+91 98765 43210'),
                      ],
                    ), // Column
                  ), // Padding
                ), // Card
                const SizedBox(height: 24),
                // User Details and Message Form
                const Text(
                  'Send a Message',
                  style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                ), // Text
                const SizedBox(height: 16),
                TextFormField(
                  controller: _nameController,
                  decoration: const InputDecoration(
                    labelText: 'Your Name',
                    border: OutlineInputBorder(),
                  ), // InputDecoration
                  validator: (value) => value!.isEmpty ? 'Please enter your name' : null,
                ), // TextFormField
                const SizedBox(height: 16),
                TextFormField(
                  controller: _idController,
                  decoration: const InputDecoration(
                    labelText: 'Your Student ID',
                    border: OutlineInputBorder(),
                  ), // InputDecoration
                  validator: (value) => value!.isEmpty ? 'Please enter your student ID' : null,
                ), // TextFormField
                const SizedBox(height: 16),
```

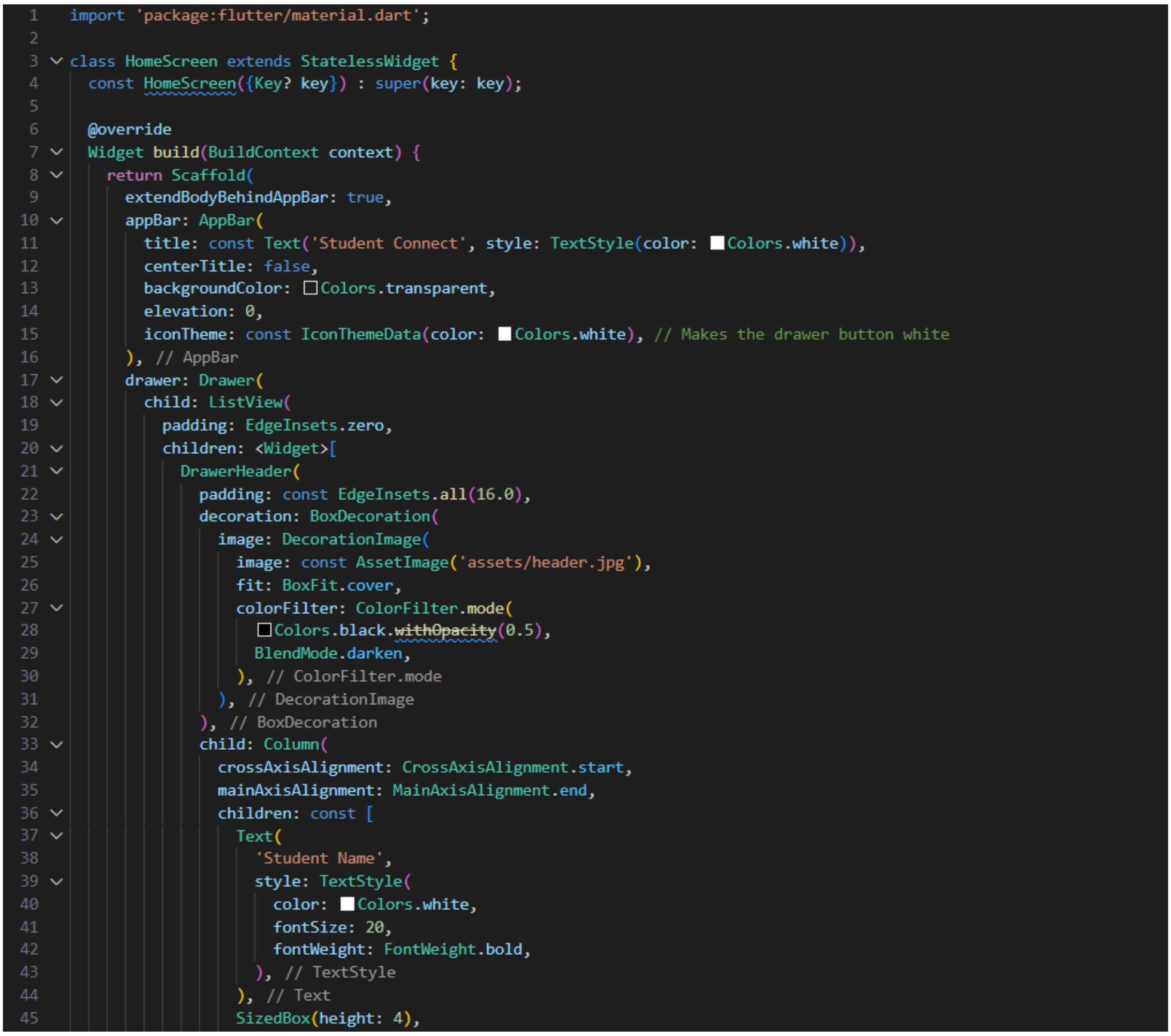


```dart
              const SizedBox(height: 16),
              TextFormField(
                controller: _messageController,
                decoration: const InputDecoration(
                  labelText: 'Your Message',
                  border: OutlineInputBorder(),
                  alignLabelWithHint: true,
                ), // InputDecoration
                maxLines: 5,
                validator: (value) => value!.isEmpty ? 'Message cannot be empty' : null,
              ), // TextFormField
              const SizedBox(height: 16),
              Row(
                mainAxisAlignment: MainAxisAlignment.end,
                children: [
                  ElevatedButton(
                    onPressed: _clearForm,
                    style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.grey,
                    ),
                    child: const Text('Clear'),
                  ), // ElevatedButton
                  const SizedBox(width: 8),
                  ElevatedButton(
                    onPressed: _sendMessage,
                    child: const Text('Send'),
                  ), // ElevatedButton
                ],
              ), // Row
            ],
          ), // Column
        ), // Form
      ), // SingleChildScrollView
    ); // Scaffold
  }

  Widget _buildDetailRow(IconData icon, String text) {
    return Row(
      children: [
        Icon(icon, color: Colors.blue),
        const SizedBox(width: 16),
        Text(text),
```

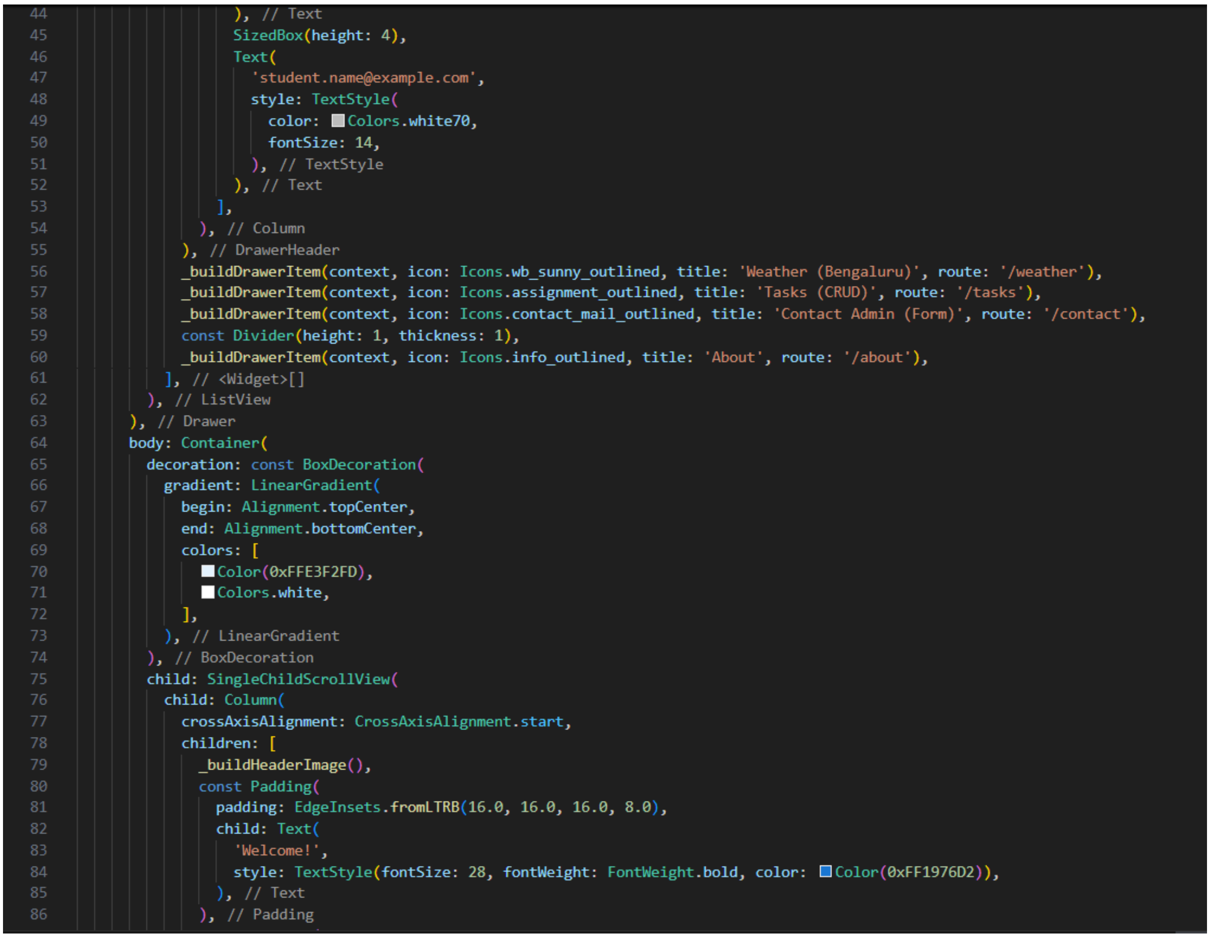


```dart
                    child: const Text('Send'),
                  ), // ElevatedButton
                ],
              ), // Row
            ],
          ), // Column
        ), // Form
      ), // SingleChildScrollView
    ); // Scaffold
  }

  Widget _buildDetailRow(IconData icon, String text) {
    return Row(
      children: [
        Icon(icon, color: Colors.blue),
        const SizedBox(width: 16),
        Text(text),
      ],
    ); // Row
  }
}
```

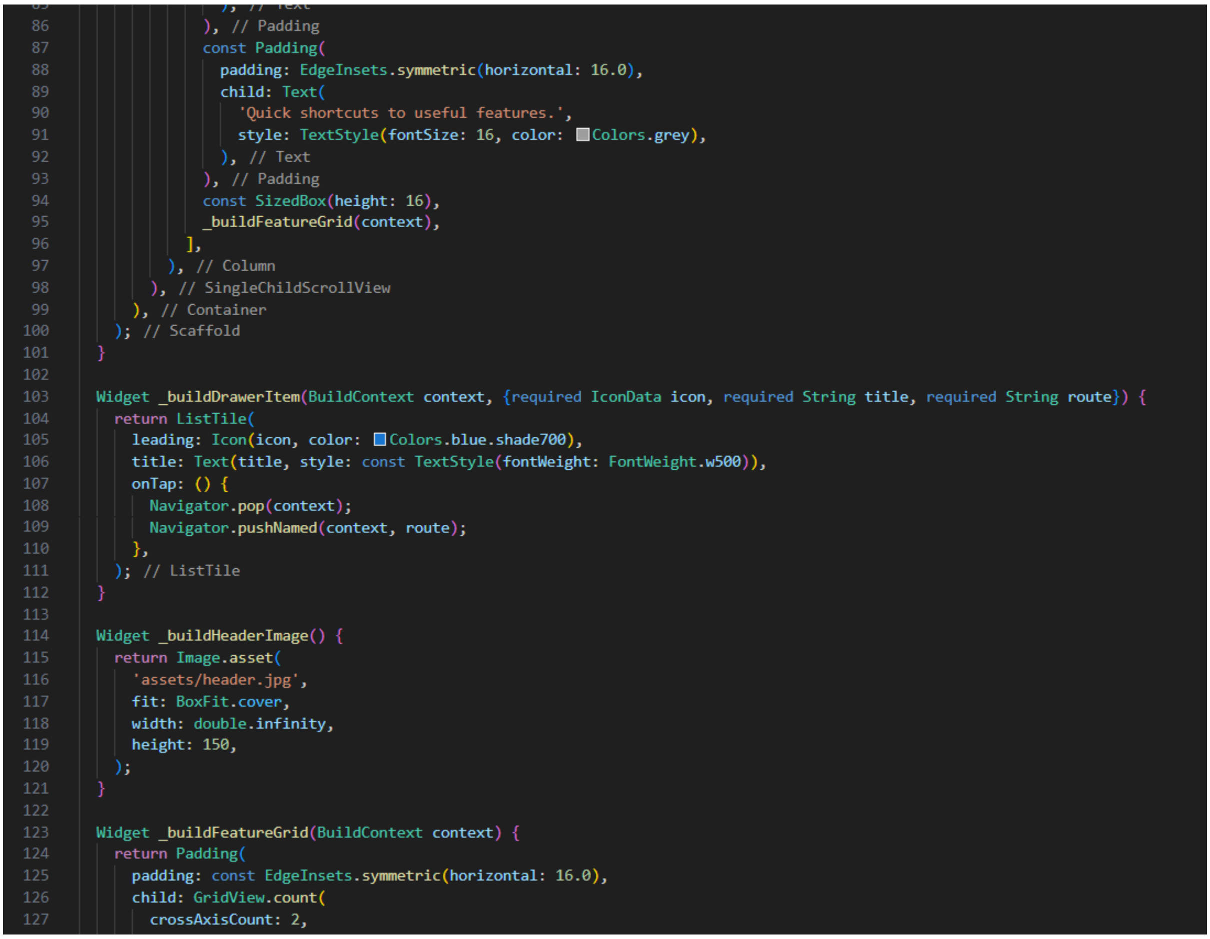
In home_screen.dart file :-



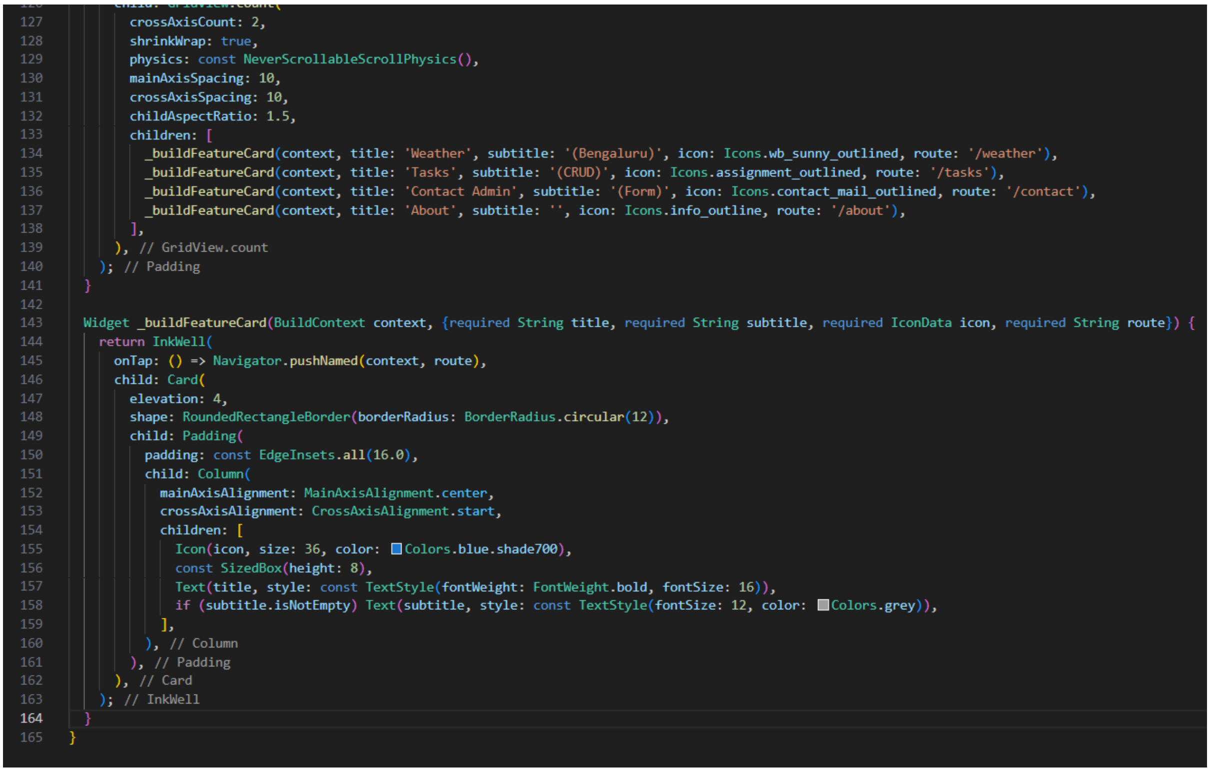
```dart
import 'package:flutter/material.dart';

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBodyBehindAppBar: true,
      appBar: AppBar(
        title: const Text('Student Connect', style: TextStyle(color: ■Colors.white)),
        centerTitle: false,
        backgroundColor: □Colors.transparent,
        elevation: 0,
        iconTheme: const IconThemeData(color: ■Colors.white), // Makes the drawer button white
      ), // AppBar
      drawer: Drawer(
        child: ListView(
          padding: EdgeInsets.zero,
          children: <Widget>[
            DrawerHeader(
              padding: const EdgeInsets.all(16.0),
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: const AssetImage('assets/header.jpg'),
                  fit: BoxFit.cover,
                  colorFilter: ColorFilter.mode(
                    □Colors.black.withOpacity(0.5),
                    BlendMode.darken,
                  ), // ColorFilter.mode
                ), // DecorationImage
              ), // BoxDecoration
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                mainAxisAlignment: MainAxisAlignment.end,
                children: const [
                  Text(
                    'Student Name',
                    style: TextStyle(
                      color: ■Colors.white,
                      fontSize: 20,
                      fontWeight: FontWeight.bold,
                    ), // TextStyle
                  ), // Text
                  SizedBox(height: 4),
```



```dart
                    ), // Text
                    SizedBox(height: 4),
                    Text(
                      'student.name@example.com',
                      style: TextStyle(
                        color: Colors.white70,
                        fontSize: 14,
                      ), // TextStyle
                    ), // Text
                  ],
                ), // Column
              ), // DrawerHeader
              _buildDrawerItem(context, icon: Icons.wb_sunny_outlined, title: 'Weather (Bengaluru)', route: '/weather'),
              _buildDrawerItem(context, icon: Icons.assignment_outlined, title: 'Tasks (CRUD)', route: '/tasks'),
              _buildDrawerItem(context, icon: Icons.contact_mail_outlined, title: 'Contact Admin (Form)', route: '/contact'),
              const Divider(height: 1, thickness: 1),
              _buildDrawerItem(context, icon: Icons.info_outlined, title: 'About', route: '/about'),
            ], // <Widget>[]
          ), // ListView
        ), // Drawer
        body: Container(
          decoration: const BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
              colors: [
                Color(0xFFE3F2FD),
                Colors.white,
              ],
            ), // LinearGradient
          ), // BoxDecoration
          child: SingleChildScrollView(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                _buildHeaderImage(),
                const Padding(
                  padding: EdgeInsets.fromLTRB(16.0, 16.0, 16.0, 8.0),
                  child: Text(
                    'Welcome!',
                    style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold, color: Color(0xFF1976D2)),
                  ), // Text
                ), // Padding
```

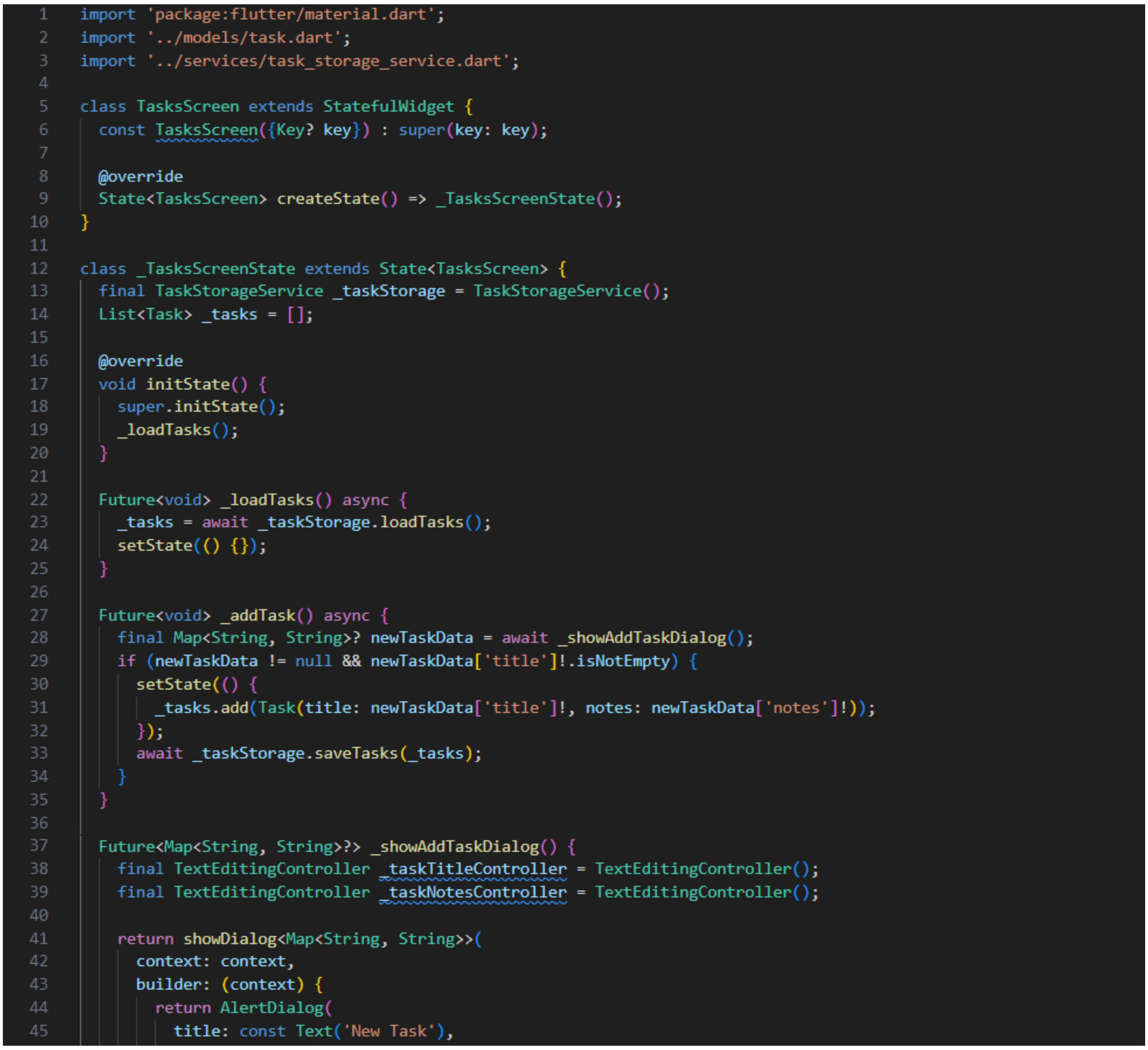


```dart
                ), // Text
              ), // Padding
              const Padding(
                padding: EdgeInsets.symmetric(horizontal: 16.0),
                child: Text(
                  'Quick shortcuts to useful features.',
                  style: TextStyle(fontSize: 16, color: ■Colors.grey),
                ), // Text
              ), // Padding
              const SizedBox(height: 16),
              _buildFeatureGrid(context),
            ],
          ), // Column
        ), // SingleChildScrollView
      ), // Container
    ); // Scaffold
  }

  Widget _buildDrawerItem(BuildContext context, {required IconData icon, required String title, required String route}) {
    return ListTile(
      leading: Icon(icon, color: 🟦Colors.blue.shade700),
      title: Text(title, style: const TextStyle(fontWeight: FontWeight.w500)),
      onTap: () {
        Navigator.pop(context);
        Navigator.pushNamed(context, route);
      },
    ); // ListTile
  }

  Widget _buildHeaderImage() {
    return Image.asset(
      'assets/header.jpg',
      fit: BoxFit.cover,
      width: double.infinity,
      height: 150,
    );
  }

  Widget _buildFeatureGrid(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.symmetric(horizontal: 16.0),
      child: GridView.count(
        crossAxisCount: 2,
```

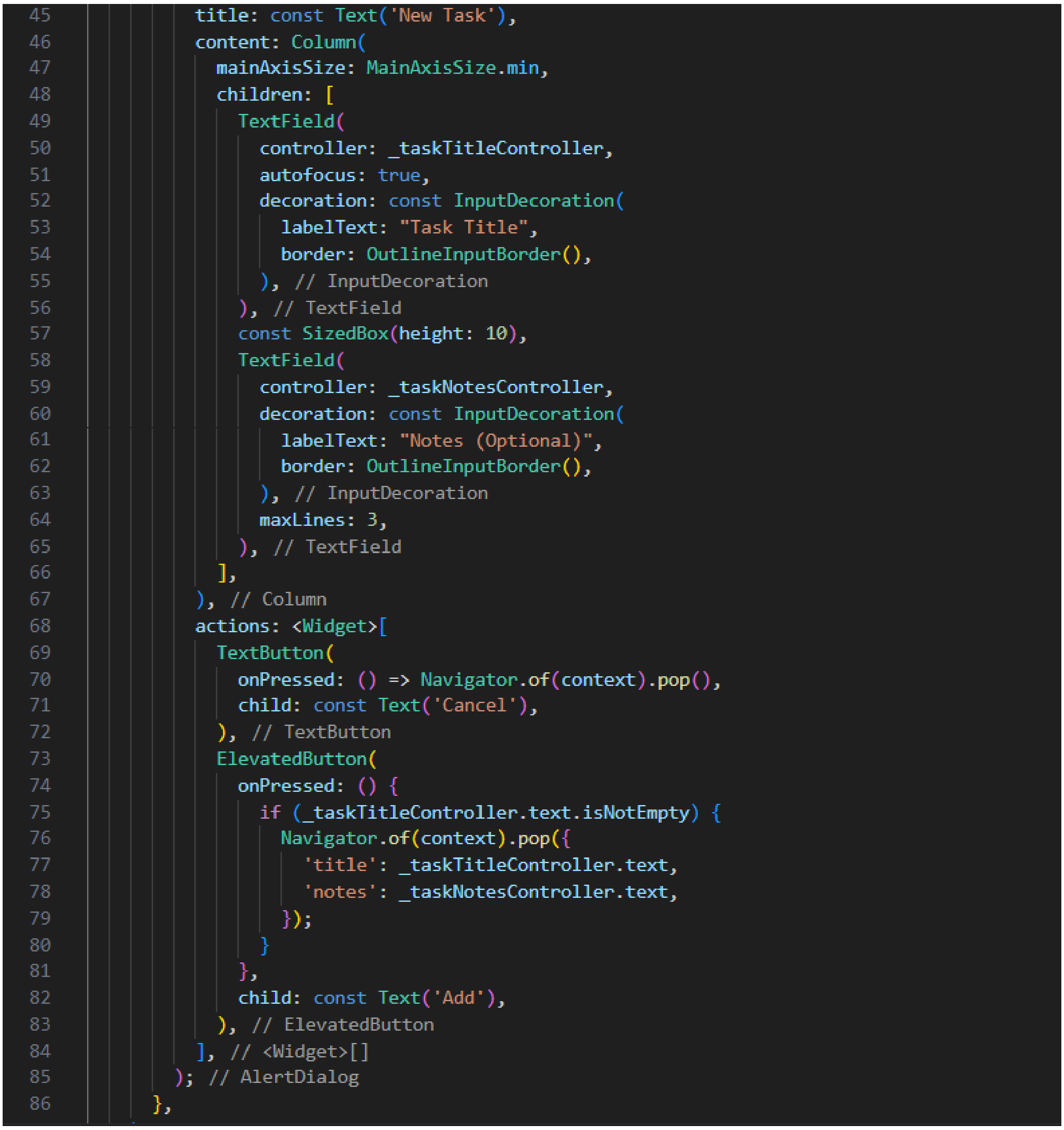


```dart
      child: GridView.count(
        crossAxisCount: 2,
        shrinkWrap: true,
        physics: const NeverScrollableScrollPhysics(),
        mainAxisSpacing: 10,
        crossAxisSpacing: 10,
        childAspectRatio: 1.5,
        children: [
          _buildFeatureCard(context, title: 'Weather', subtitle: '(Bengaluru)', icon: Icons.wb_sunny_outlined, route: '/weather'),
          _buildFeatureCard(context, title: 'Tasks', subtitle: '(CRUD)', icon: Icons.assignment_outlined, route: '/tasks'),
          _buildFeatureCard(context, title: 'Contact Admin', subtitle: '(Form)', icon: Icons.contact_mail_outlined, route: '/contact'),
          _buildFeatureCard(context, title: 'About', subtitle: '', icon: Icons.info_outline, route: '/about'),
        ],
      ), // GridView.count
    ); // Padding
  }

  Widget _buildFeatureCard(BuildContext context, {required String title, required String subtitle, required IconData icon, required String route}) {
    return InkWell(
      onTap: () => Navigator.pushNamed(context, route),
      child: Card(
        elevation: 4,
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Icon(icon, size: 36, color: 🟦Colors.blue.shade700),
              const SizedBox(height: 8),
              Text(title, style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 16)),
              if (subtitle.isNotEmpty) Text(subtitle, style: const TextStyle(fontSize: 12, color: ■Colors.grey)),
            ],
          ), // Column
        ), // Padding
      ), // Card
    ); // InkWell
  }
}
```

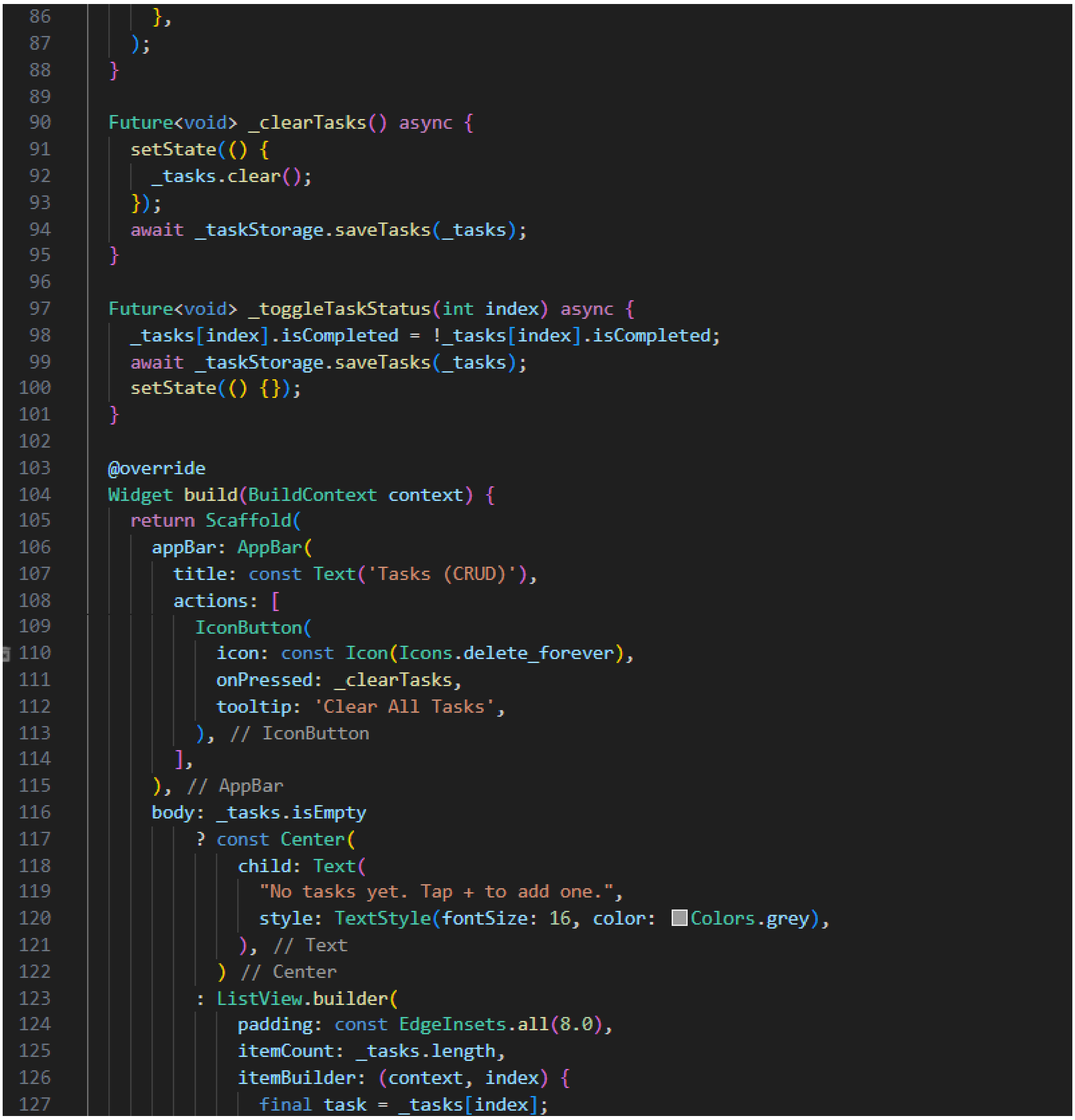
In task_screen.dart file :-



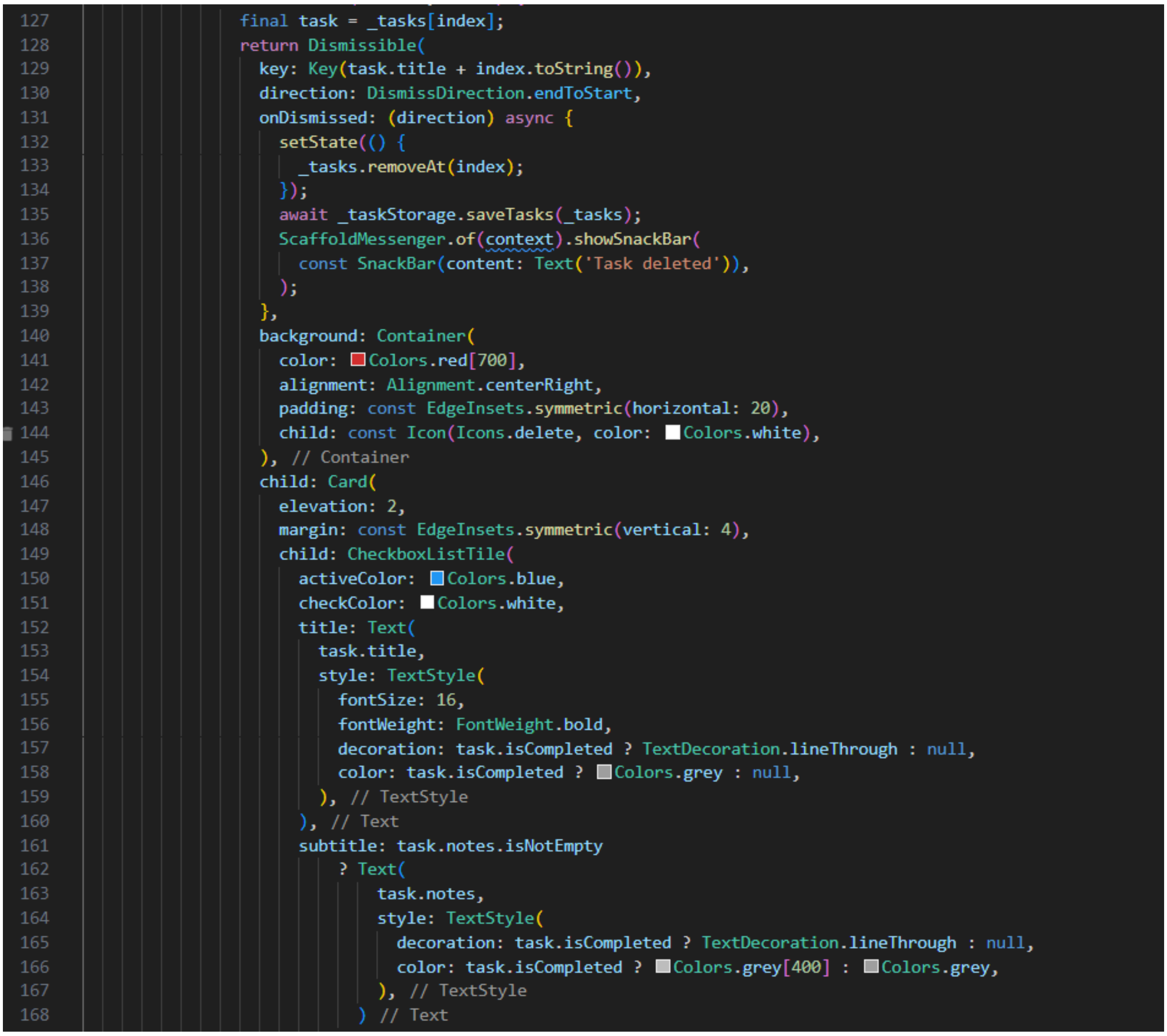
```dart
import 'package:flutter/material.dart';
import '../models/task.dart';
import '../services/task_storage_service.dart';

class TasksScreen extends StatefulWidget {
  const TasksScreen({Key? key}) : super(key: key);

  @override
  State<TasksScreen> createState() => _TasksScreenState();
}

class _TasksScreenState extends State<TasksScreen> {
  final TaskStorageService _taskStorage = TaskStorageService();
  List<Task> _tasks = [];

  @override
  void initState() {
    super.initState();
    _loadTasks();
  }

  Future<void> _loadTasks() async {
    _tasks = await _taskStorage.loadTasks();
    setState(() {});
  }

  Future<void> _addTask() async {
    final Map<String, String>? newTaskData = await _showAddTaskDialog();
    if (newTaskData != null && newTaskData['title']!.isNotEmpty) {
      setState(() {
        _tasks.add(Task(title: newTaskData['title']!, notes: newTaskData['notes']!));
      });
      await _taskStorage.saveTasks(_tasks);
    }
  }

  Future<Map<String, String>?> _showAddTaskDialog() {
    final TextEditingController _taskTitleController = TextEditingController();
    final TextEditingController _taskNotesController = TextEditingController();

    return showDialog<Map<String, String>>(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: const Text('New Task'),
```



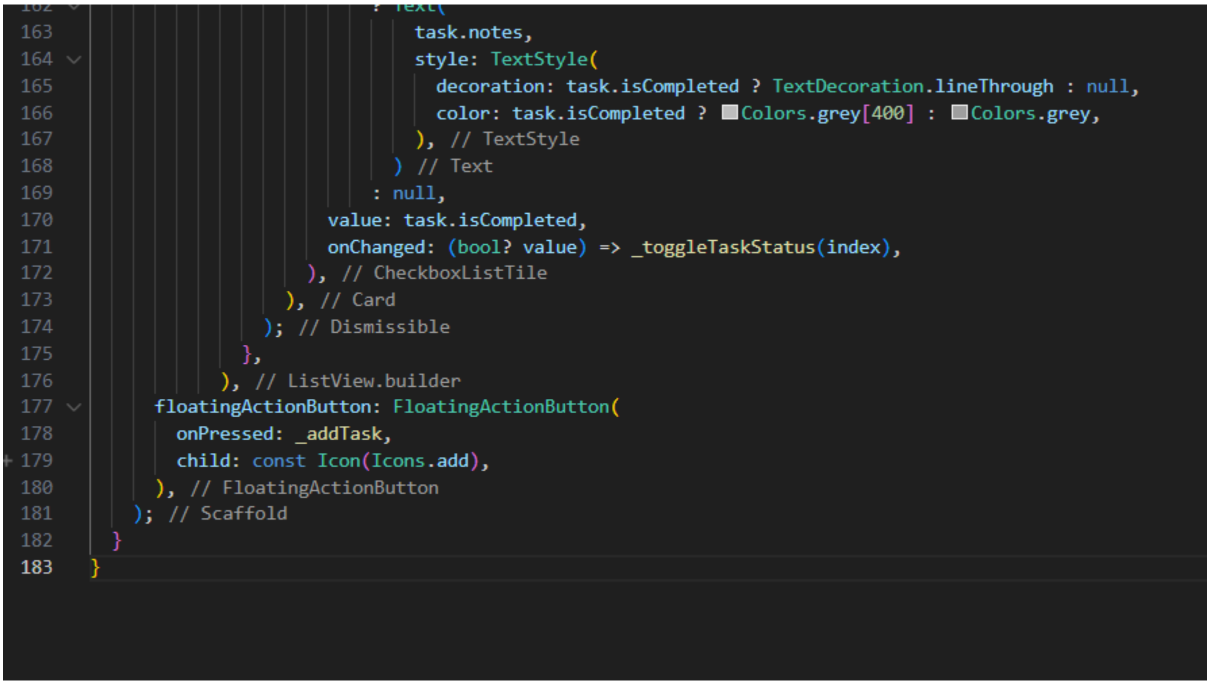
```
          title: const Text('New Task'),
          content: Column(
            mainAxisSize: MainAxisSize.min,
            children: [
              TextField(
                controller: _taskTitleController,
                autofocus: true,
                decoration: const InputDecoration(
                  labelText: "Task Title",
                  border: OutlineInputBorder(),
                ), // InputDecoration
              ), // TextField
              const SizedBox(height: 10),
              TextField(
                controller: _taskNotesController,
                decoration: const InputDecoration(
                  labelText: "Notes (Optional)",
                  border: OutlineInputBorder(),
                ), // InputDecoration
                maxLines: 3,
              ), // TextField
            ],
          ), // Column
          actions: <Widget>[
            TextButton(
              onPressed: () => Navigator.of(context).pop(),
              child: const Text('Cancel'),
            ), // TextButton
            ElevatedButton(
              onPressed: () {
                if (_taskTitleController.text.isNotEmpty) {
                  Navigator.of(context).pop({
                    'title': _taskTitleController.text,
                    'notes': _taskNotesController.text,
                  });
                }
              },
              child: const Text('Add'),
            ), // ElevatedButton
          ], // <Widget>[]
        ); // AlertDialog
      },
```



```
      },
    );
  }

  Future<void> _clearTasks() async {
    setState(() {
      _tasks.clear();
    });
    await _taskStorage.saveTasks(_tasks);
  }

  Future<void> _toggleTaskStatus(int index) async {
    _tasks[index].isCompleted = !_tasks[index].isCompleted;
    await _taskStorage.saveTasks(_tasks);
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Tasks (CRUD)'),
        actions: [
          IconButton(
            icon: const Icon(Icons.delete_forever),
            onPressed: _clearTasks,
            tooltip: 'Clear All Tasks',
          ), // IconButton
        ],
      ), // AppBar
      body: _tasks.isEmpty
          ? const Center(
              child: Text(
                "No tasks yet. Tap + to add one.",
                style: TextStyle(fontSize: 16, color: Colors.grey),
              ), // Text
            ) // Center
          : ListView.builder(
              padding: const EdgeInsets.all(8.0),
              itemCount: _tasks.length,
              itemBuilder: (context, index) {
                final task = _tasks[index];
```

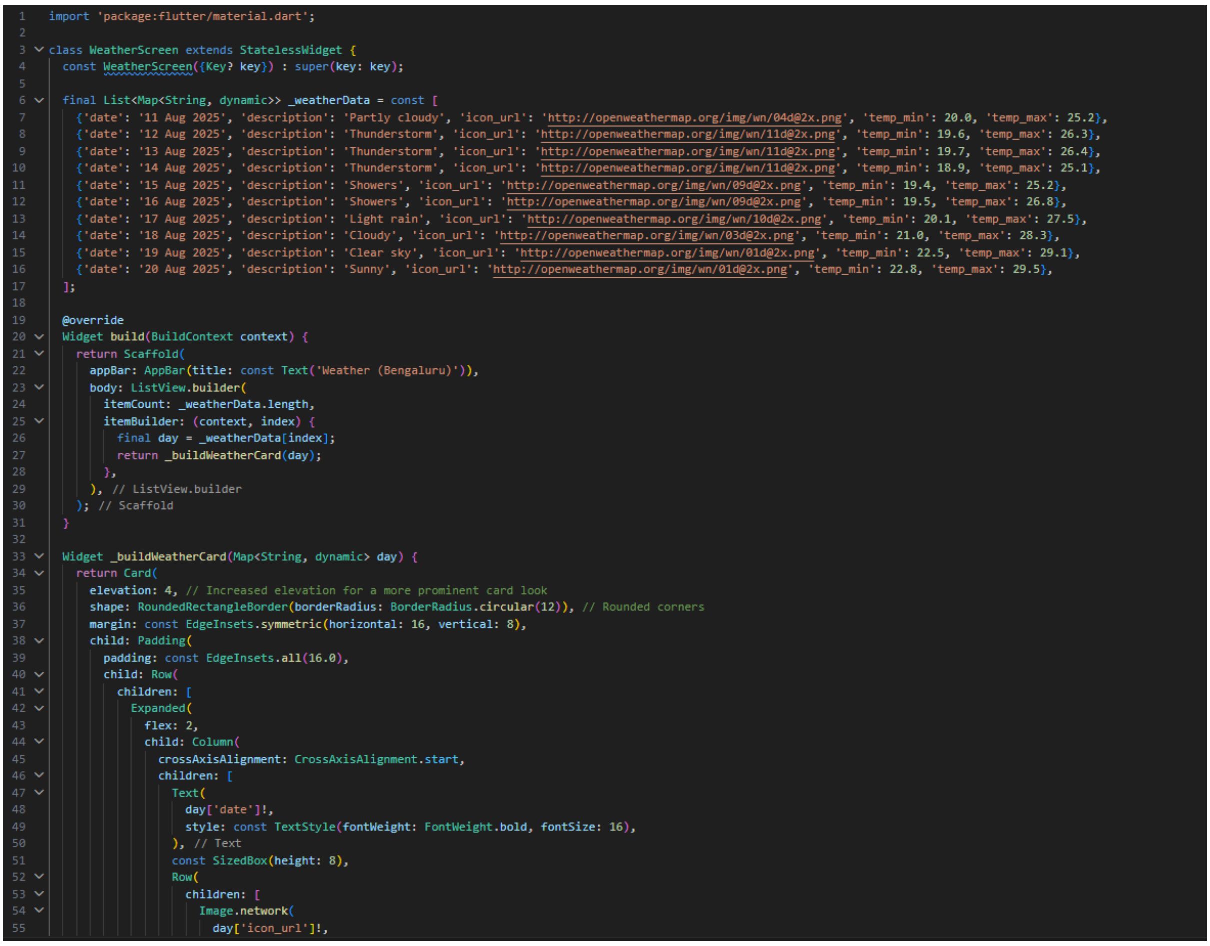


```
                final task = _tasks[index];
                return Dismissible(
                  key: Key(task.title + index.toString()),
                  direction: DismissDirection.endToStart,
                  onDismissed: (direction) async {
                    setState(() {
                      _tasks.removeAt(index);
                    });
                    await _taskStorage.saveTasks(_tasks);
                    ScaffoldMessenger.of(context).showSnackBar(
                      const SnackBar(content: Text('Task deleted')),
                    );
                  },
                  background: Container(
                    color: Colors.red[700],
                    alignment: Alignment.centerRight,
                    padding: const EdgeInsets.symmetric(horizontal: 20),
                    child: const Icon(Icons.delete, color: Colors.white),
                  ), // Container
                  child: Card(
                    elevation: 2,
                    margin: const EdgeInsets.symmetric(vertical: 4),
                    child: CheckboxListTile(
                      activeColor: Colors.blue,
                      checkColor: Colors.white,
                      title: Text(
                        task.title,
                        style: TextStyle(
                          fontSize: 16,
                          fontWeight: FontWeight.bold,
                          decoration: task.isCompleted ? TextDecoration.lineThrough : null,
                          color: task.isCompleted ? Colors.grey : null,
                        ), // TextStyle
                      ), // Text
                      subtitle: task.notes.isNotEmpty
                          ? Text(
                              task.notes,
                              style: TextStyle(
                                decoration: task.isCompleted ? TextDecoration.lineThrough : null,
                                color: task.isCompleted ? Colors.grey[400] : Colors.grey,
                              ), // TextStyle
                            ) // Text
```

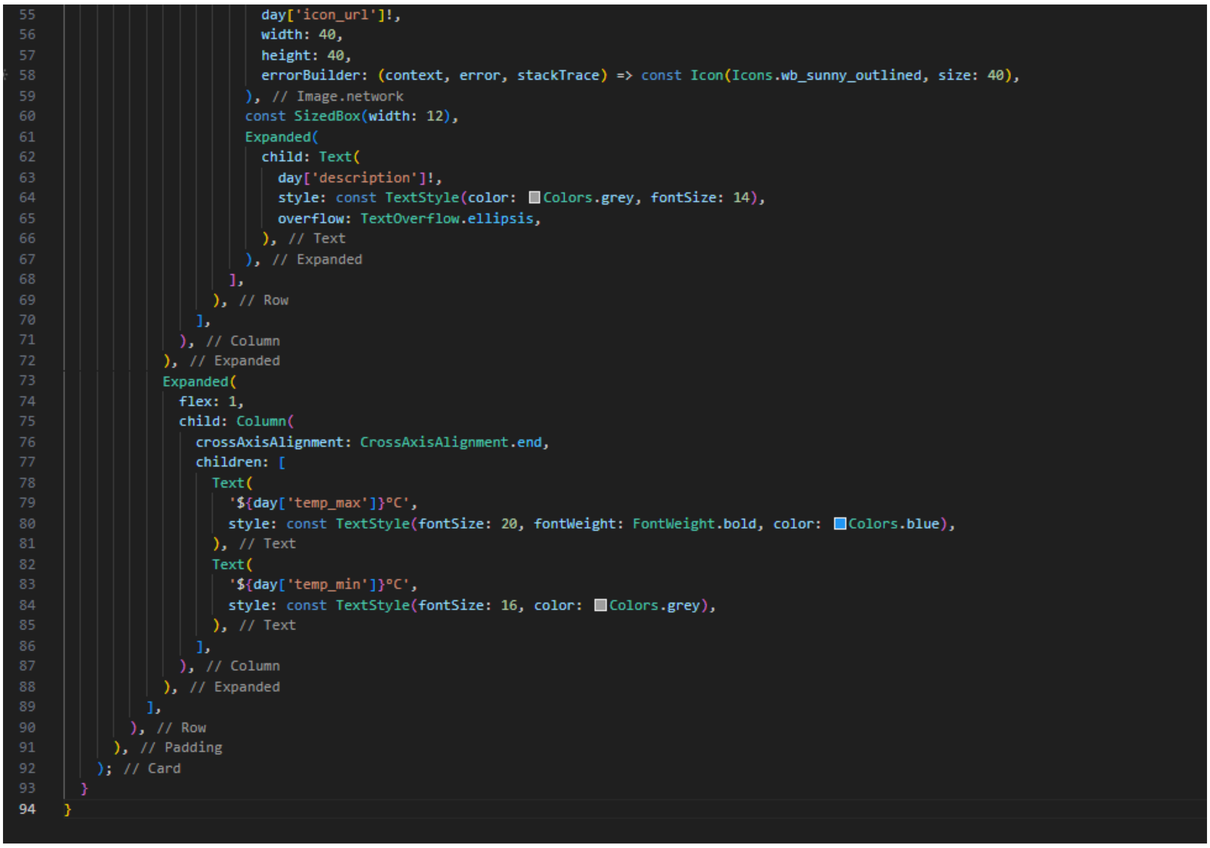


```
                              task.notes,
                              style: TextStyle(
                                decoration: task.isCompleted ? TextDecoration.lineThrough : null,
                                color: task.isCompleted ? Colors.grey[400] : Colors.grey,
                              ), // TextStyle
                            ) // Text
                          : null,
                      value: task.isCompleted,
                      onChanged: (bool? value) => _toggleTaskStatus(index),
                    ), // CheckboxListTile
                  ), // Card
                ); // Dismissible
              },
            ), // ListView.builder
      floatingActionButton: FloatingActionButton(
        onPressed: _addTask,
        child: const Icon(Icons.add),
      ), // FloatingActionButton
    ); // Scaffold
  }
}
```

In   weather_screen.dart file :-

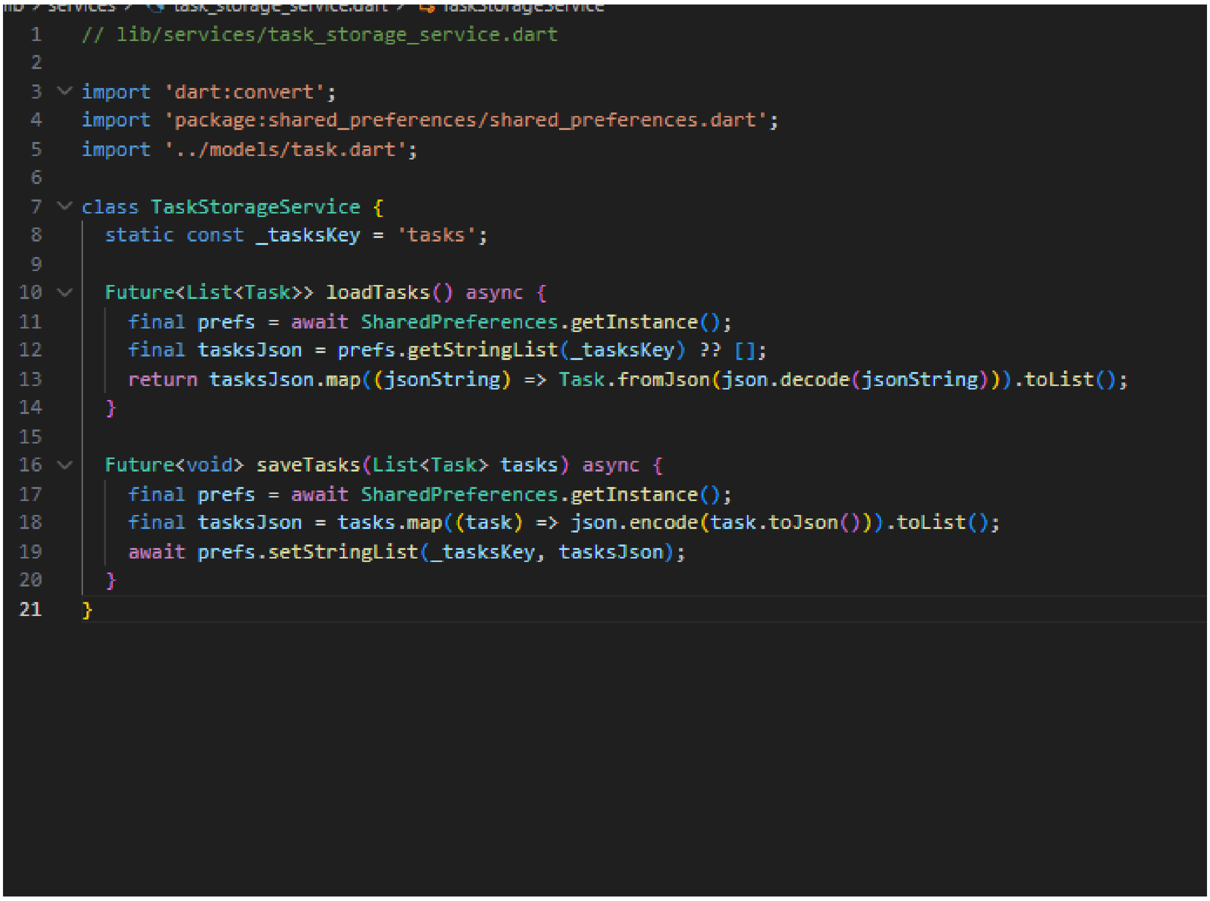


```dart
import 'package:flutter/material.dart';

class WeatherScreen extends StatelessWidget {
  const WeatherScreen({Key? key}) : super(key: key);

  final List<Map<String, dynamic>> _weatherData = const [
    {'date': '11 Aug 2025', 'description': 'Partly cloudy', 'icon_url': 'http://openweathermap.org/img/wn/04d@2x.png', 'temp_min': 20.0, 'temp_max': 25.2},
    {'date': '12 Aug 2025', 'description': 'Thunderstorm', 'icon_url': 'http://openweathermap.org/img/wn/11d@2x.png', 'temp_min': 19.6, 'temp_max': 26.3},
    {'date': '13 Aug 2025', 'description': 'Thunderstorm', 'icon_url': 'http://openweathermap.org/img/wn/11d@2x.png', 'temp_min': 19.7, 'temp_max': 26.4},
    {'date': '14 Aug 2025', 'description': 'Thunderstorm', 'icon_url': 'http://openweathermap.org/img/wn/11d@2x.png', 'temp_min': 18.9, 'temp_max': 25.1},
    {'date': '15 Aug 2025', 'description': 'Showers', 'icon_url': 'http://openweathermap.org/img/wn/09d@2x.png', 'temp_min': 19.4, 'temp_max': 25.2},
    {'date': '16 Aug 2025', 'description': 'Showers', 'icon_url': 'http://openweathermap.org/img/wn/09d@2x.png', 'temp_min': 19.5, 'temp_max': 26.8},
    {'date': '17 Aug 2025', 'description': 'Light rain', 'icon_url': 'http://openweathermap.org/img/wn/10d@2x.png', 'temp_min': 20.1, 'temp_max': 27.5},
    {'date': '18 Aug 2025', 'description': 'Cloudy', 'icon_url': 'http://openweathermap.org/img/wn/03d@2x.png', 'temp_min': 21.0, 'temp_max': 28.3},
    {'date': '19 Aug 2025', 'description': 'Clear sky', 'icon_url': 'http://openweathermap.org/img/wn/01d@2x.png', 'temp_min': 22.5, 'temp_max': 29.1},
    {'date': '20 Aug 2025', 'description': 'Sunny', 'icon_url': 'http://openweathermap.org/img/wn/01d@2x.png', 'temp_min': 22.8, 'temp_max': 29.5},
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Weather (Bengaluru)')),
      body: ListView.builder(
        itemCount: _weatherData.length,
        itemBuilder: (context, index) {
          final day = _weatherData[index];
          return _buildWeatherCard(day);
        },
      ), // ListView.builder
    ); // Scaffold
  }

  Widget _buildWeatherCard(Map<String, dynamic> day) {
    return Card(
      elevation: 4, // Increased elevation for a more prominent card look
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)), // Rounded corners
      margin: const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Row(
          children: [
            Expanded(
              flex: 2,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    day['date']!,
                    style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 16),
                  ), // Text
                  const SizedBox(height: 8),
                  Row(
                    children: [
                      Image.network(
                        day['icon_url']!,
```



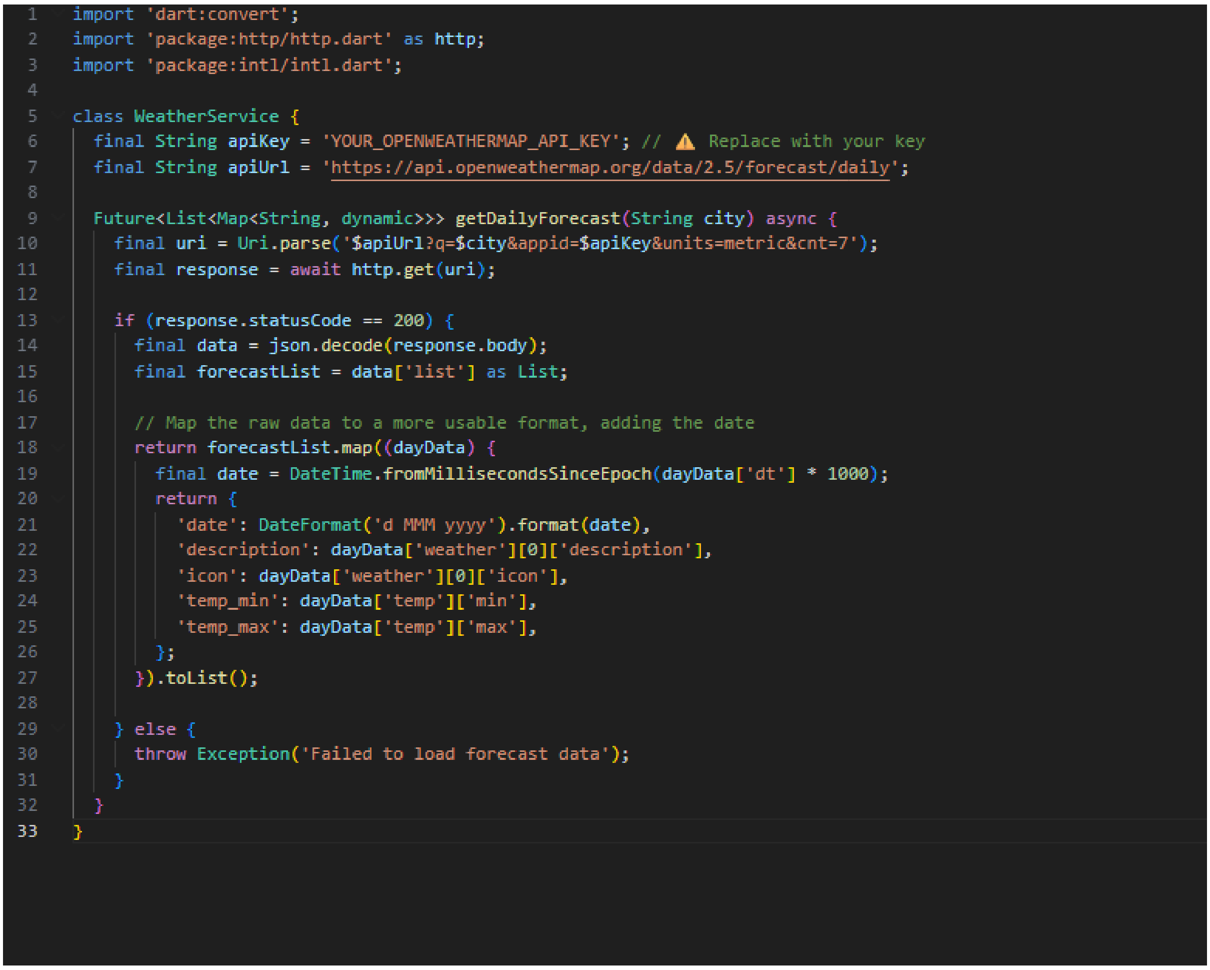
```dart
                        day['icon_url']!,
                        width: 40,
                        height: 40,
                        errorBuilder: (context, error, stackTrace) => const Icon(Icons.wb_sunny_outlined, size: 40),
                      ), // Image.network
                      const SizedBox(width: 12),
                      Expanded(
                        child: Text(
                          day['description']!,
                          style: const TextStyle(color: Colors.grey, fontSize: 14),
                          overflow: TextOverflow.ellipsis,
                        ), // Text
                      ), // Expanded
                    ],
                  ), // Row
                ],
              ), // Column
            ), // Expanded
            Expanded(
              flex: 1,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.end,
                children: [
                  Text(
                    '${day['temp_max']}°C',
                    style: const TextStyle(fontSize: 20, fontWeight: FontWeight.bold, color: Colors.blue),
                  ), // Text
                  Text(
                    '${day['temp_min']}°C',
                    style: const TextStyle(fontSize: 16, color: Colors.grey),
                  ), // Text
                ],
              ), // Column
            ), // Expanded
          ],
        ), // Row
      ), // Padding
    ); // Card
  }
}
```

In task_storage_service.dart file :-


lib > services > task_storage_service.dart > TaskStorageService

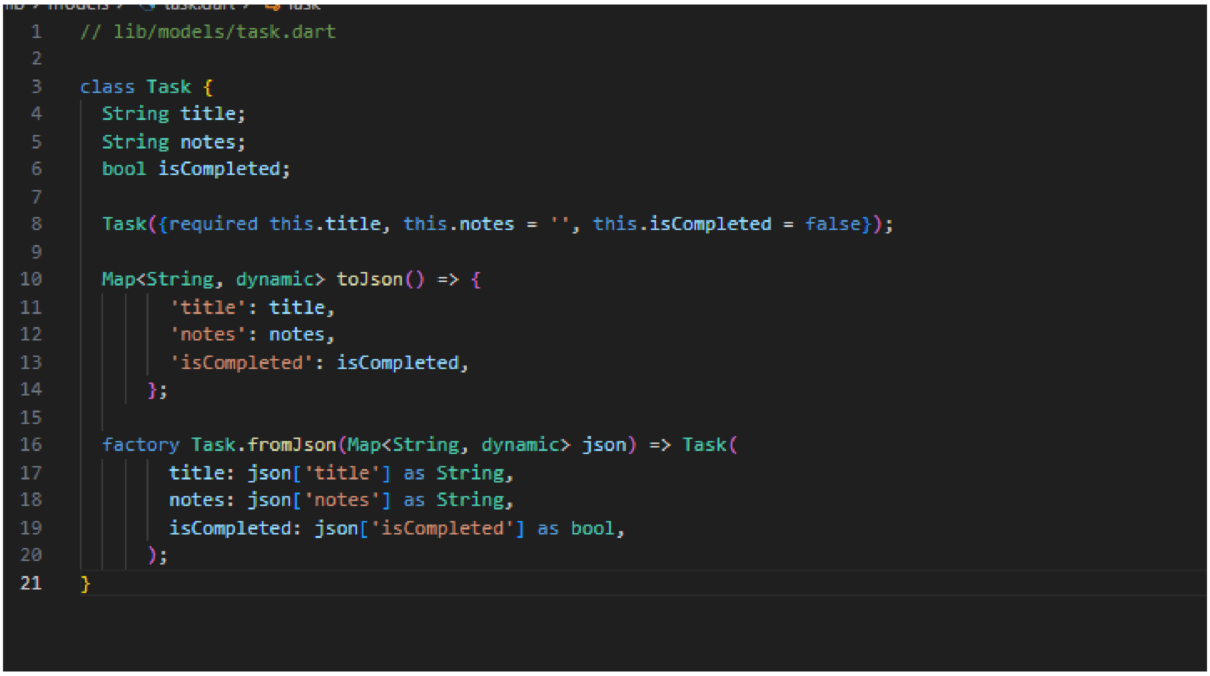
```dart
// lib/services/task_storage_service.dart

import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';
import '../models/task.dart';

class TaskStorageService {
  static const _tasksKey = 'tasks';

  Future<List<Task>> loadTasks() async {
    final prefs = await SharedPreferences.getInstance();
    final tasksJson = prefs.getStringList(_tasksKey) ?? [];
    return tasksJson.map((jsonString) => Task.fromJson(json.decode(jsonString))).toList();
  }

  Future<void> saveTasks(List<Task> tasks) async {
    final prefs = await SharedPreferences.getInstance();
    final tasksJson = tasks.map((task) => json.encode(task.toJson())).toList();
    await prefs.setStringList(_tasksKey, tasksJson);
  }
}
```

In weather_service.dart file :-



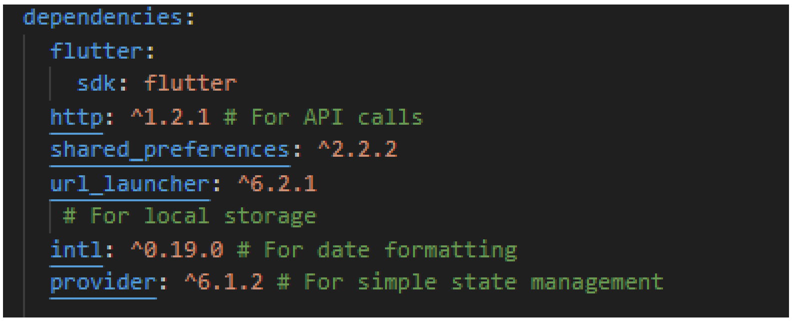
```dart
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:intl/intl.dart';

class WeatherService {
  final String apiKey = 'YOUR_OPENWEATHERMAP_API_KEY'; // ⚠ Replace with your key
  final String apiUrl = 'https://api.openweathermap.org/data/2.5/forecast/daily';

  Future<List<Map<String, dynamic>>> getDailyForecast(String city) async {
    final uri = Uri.parse('$apiUrl?q=$city&appid=$apiKey&units=metric&cnt=7');
    final response = await http.get(uri);

    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      final forecastList = data['list'] as List;

      // Map the raw data to a more usable format, adding the date
      return forecastList.map((dayData) {
        final date = DateTime.fromMillisecondsSinceEpoch(dayData['dt'] * 1000);
        return {
          'date': DateFormat('d MMM yyyy').format(date),
          'description': dayData['weather'][0]['description'],
          'icon': dayData['weather'][0]['icon'],
          'temp_min': dayData['temp']['min'],
          'temp_max': dayData['temp']['max'],
        };
      }).toList();

    } else {
      throw Exception('Failed to load forecast data');
    }
  }
}
```

In module/task.dart file :-

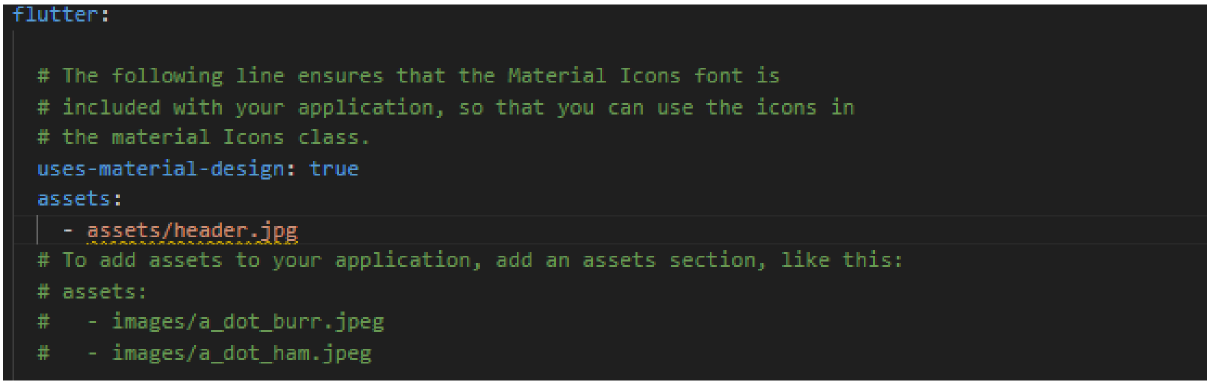


```dart
// lib/models/task.dart

class Task {
  String title;
  String notes;
  bool isCompleted;

  Task({required this.title, this.notes = '', this.isCompleted = false});

  Map<String, dynamic> toJson() => {
        'title': title,
        'notes': notes,
        'isCompleted': isCompleted,
      };

  factory Task.fromJson(Map<String, dynamic> json) => Task(
        title: json['title'] as String,
        notes: json['notes'] as String,
        isCompleted: json['isCompleted'] as bool,
      );
}
```

In pubspec.yaml file :-



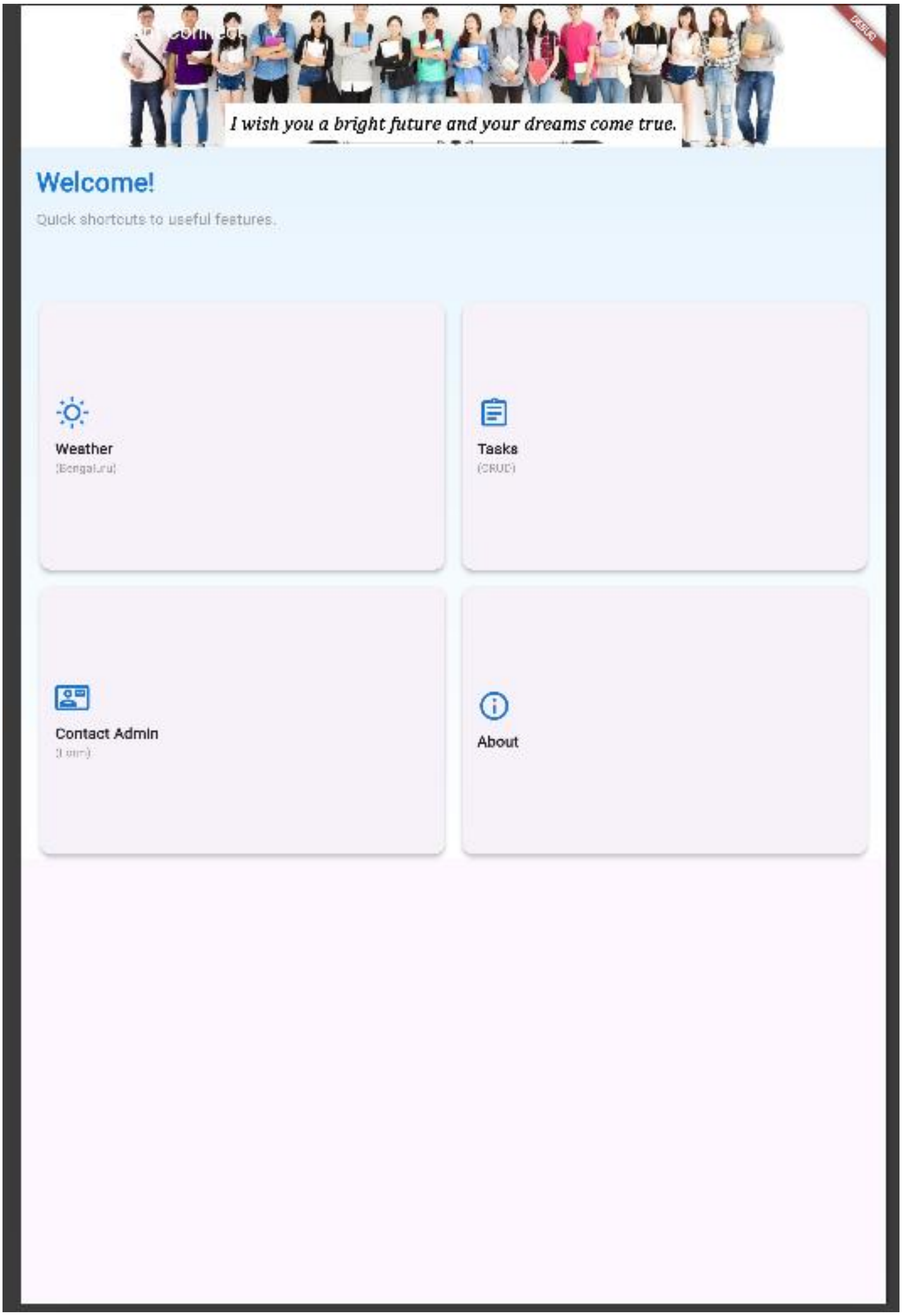
```yaml
dependencies:
  flutter:
    sdk: flutter
  http: ^1.2.1 # For API calls
  shared_preferences: ^2.2.2
  url_launcher: ^6.2.1
   # For local storage
  intl: ^0.19.0 # For date formatting
  provider: ^6.1.2 # For simple state management
```



```yaml
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true
  assets:
    - assets/header.jpg
  # To add assets to your application, add an assets section, like this:
  # assets:
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg
```

Output :-

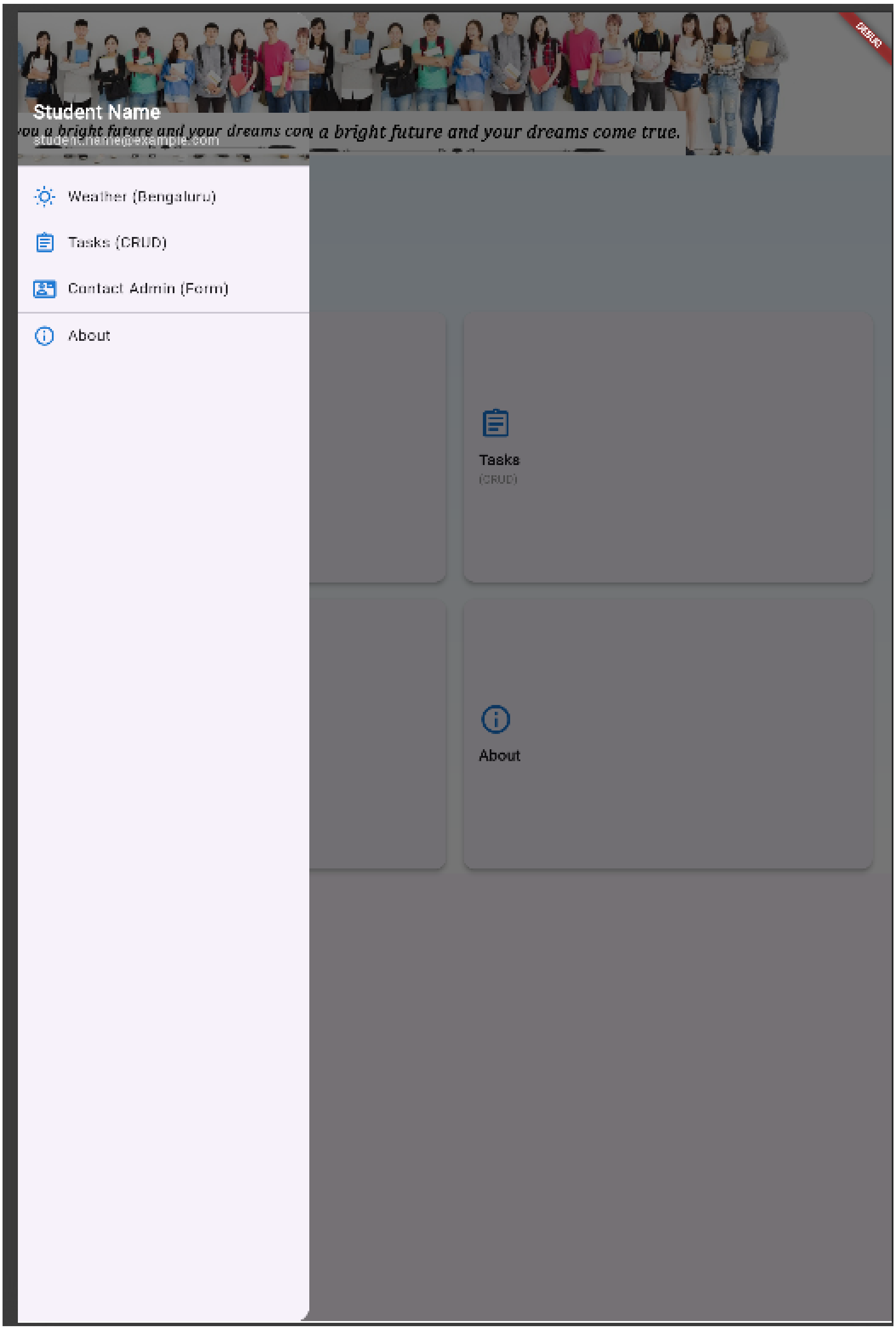
Home screen



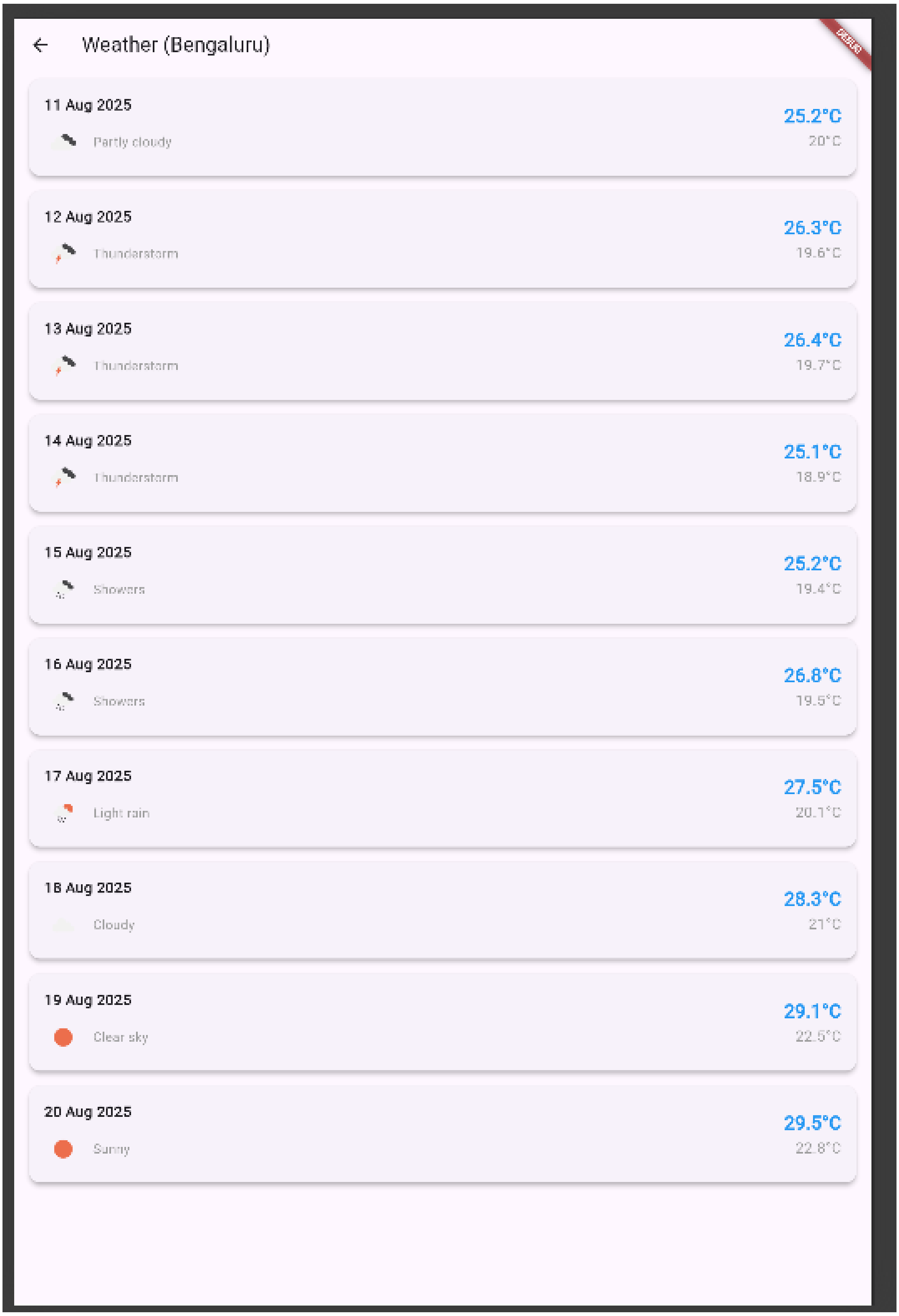
Welcome!

Quick shortcuts to useful features.

When draw menu is opened :-



When weather button is clicked from draw menu or from home screen :-



Weather (Bengaluru)

11 Aug 2025
Partly cloudy
25.2°C
20°C

12 Aug 2025
Thunderstorm
26.3°C
19.6°C

13 Aug 2025
Thunderstorm
26.4°C
19.7°C

14 Aug 2025
Thunderstorm
25.1°C
18.9°C

15 Aug 2025
Showers
25.2°C
19.4°C

16 Aug 2025
Showers
26.8°C
19.5°C

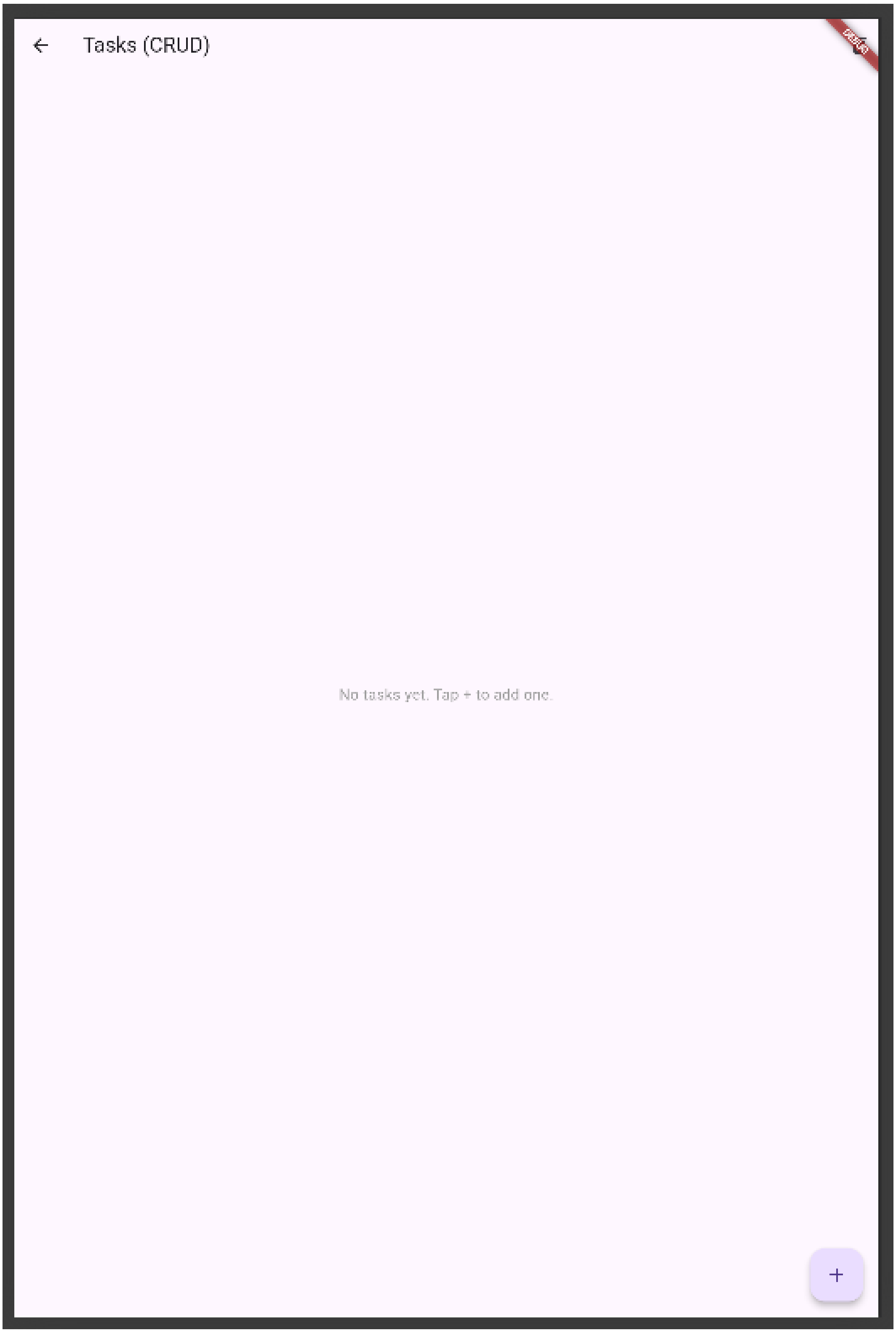
17 Aug 2025
Light rain
27.5°C
20.1°C

18 Aug 2025
Cloudy
28.3°C
21°C

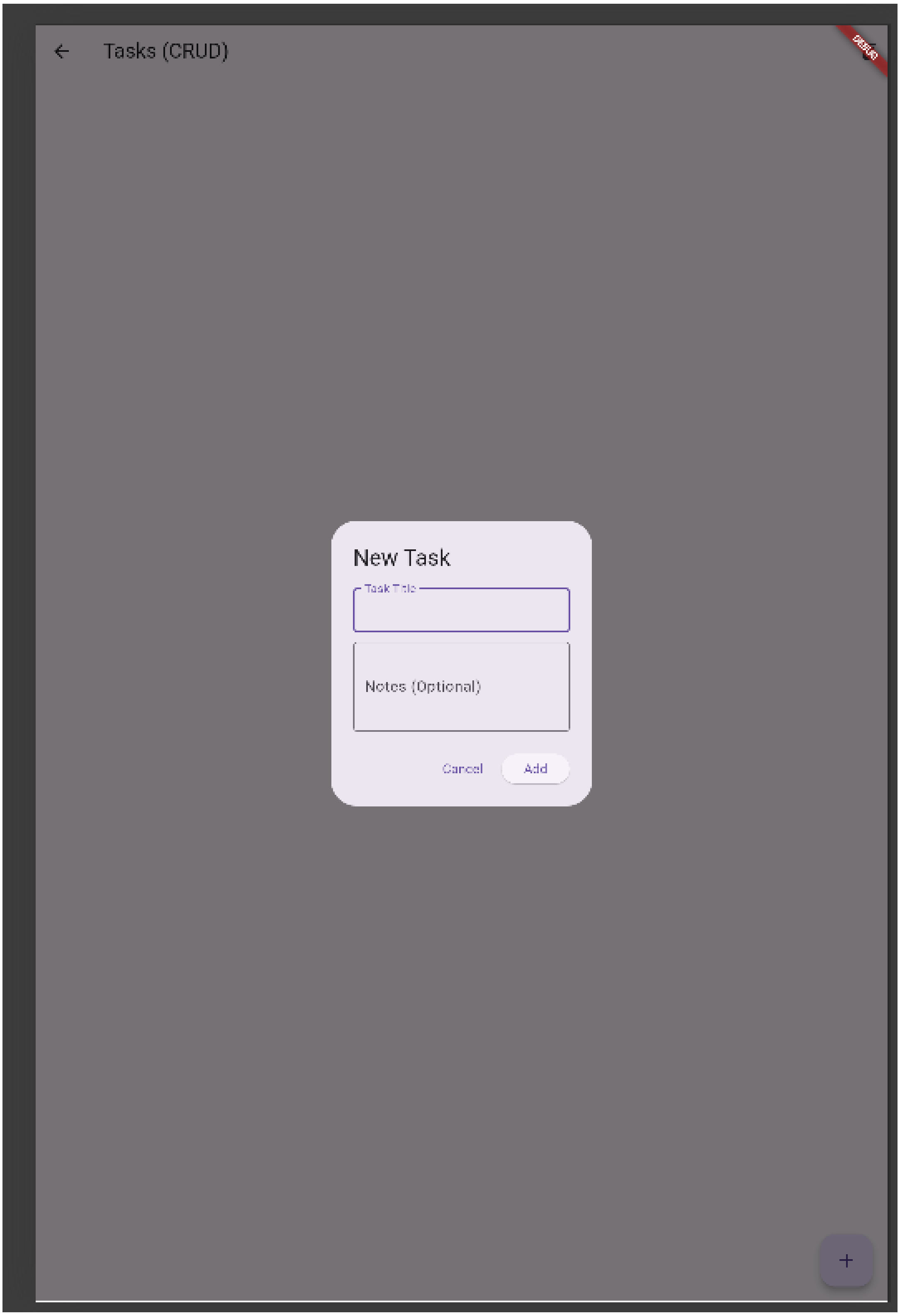
19 Aug 2025
Clear sky
29.1°C
22.5°C

20 Aug 2025
Sunny
29.5°C
22.8°C

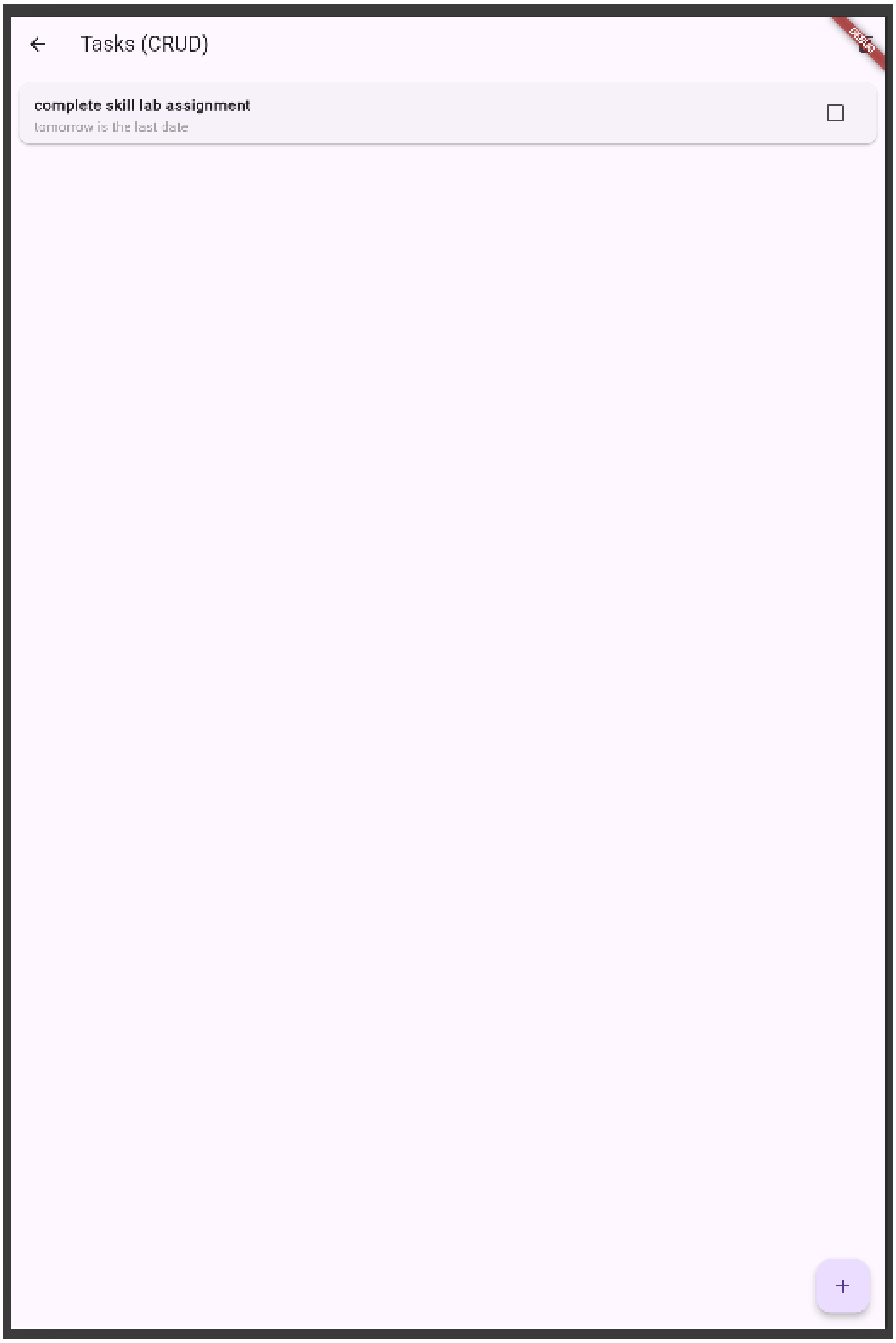
When task(crud) from draw menu clicked :-



When we clicked on add task button :-



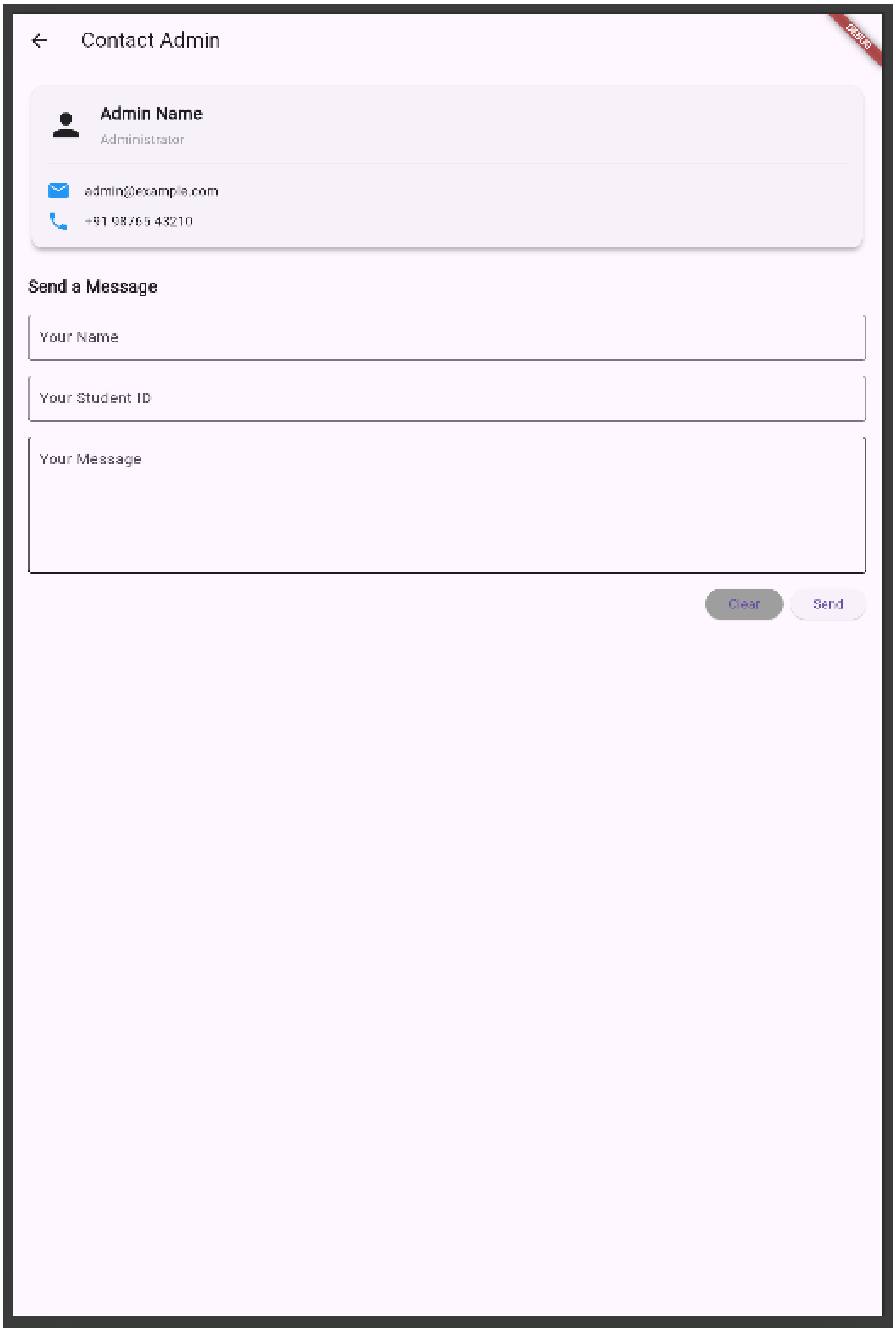
After task is added :-



complete skill lab assignment

tomorrow is the last date □

When we click on contact admin button in home screen :-



Contact Admin

Admin Name

Administrator

admin@example.com

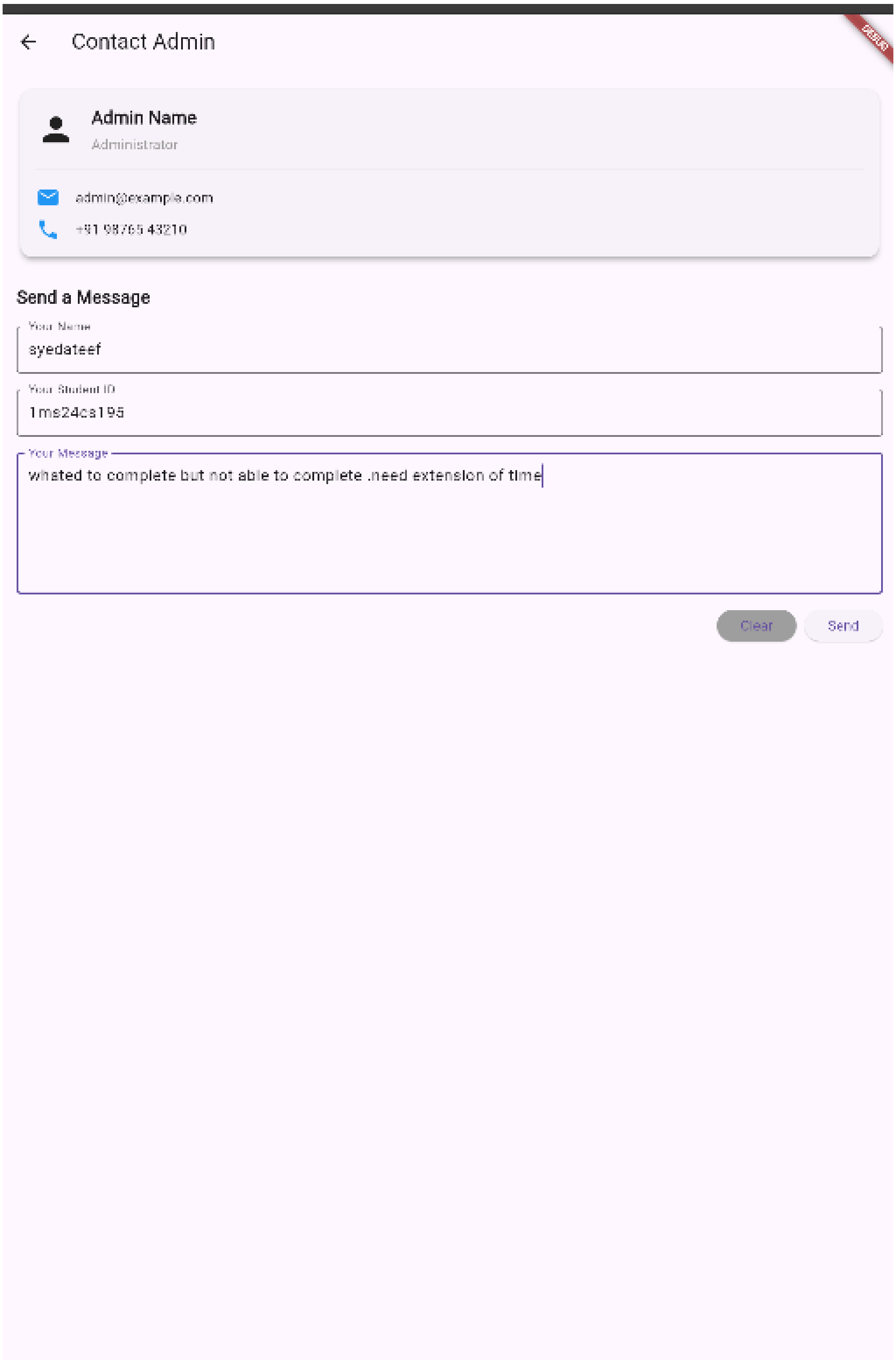
+91 98765 43210

Send a Message

Your Name

Your Student ID

Your Message



Contact Admin

Admin Name
Administrator

admin@example.com
+91 98765 43210

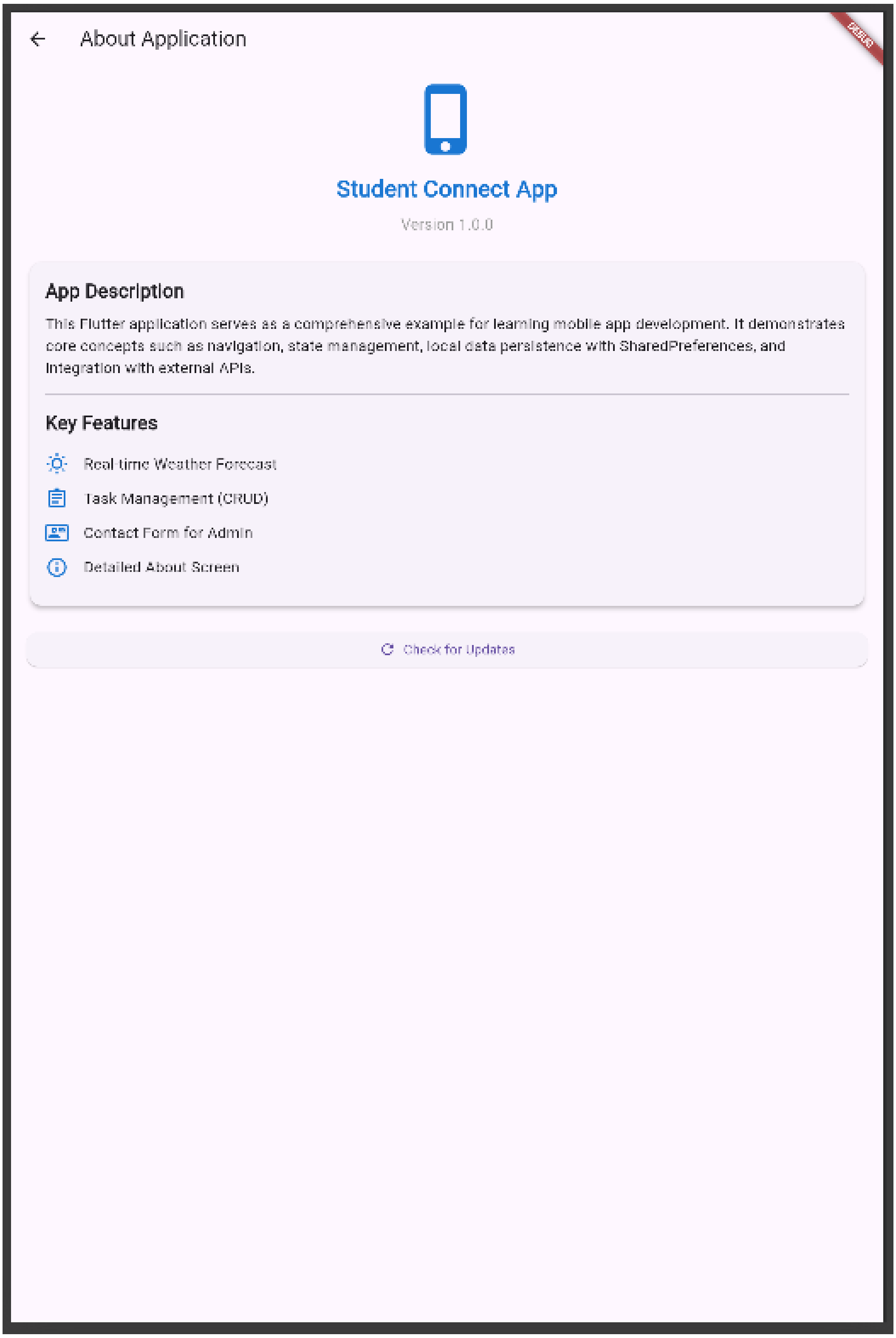
Send a Message

Your Name: syedateef

Your Student ID: 1ms24cs195

Your Message: whated to complete but not able to complete .need extension of time

When you click on about button :-



← About Application

Student Connect App

Version 1.0.0

App Description

This Flutter application serves as a comprehensive example for learning mobile app development. It demonstrates core concepts such as navigation, state management, local data persistence with SharedPreferences, and integration with external APIs.

Key Features

- Real-time Weather Forecast
- Task Management (CRUD)
- Contact Form for Admin
- Detailed About Screen

Check for Updates