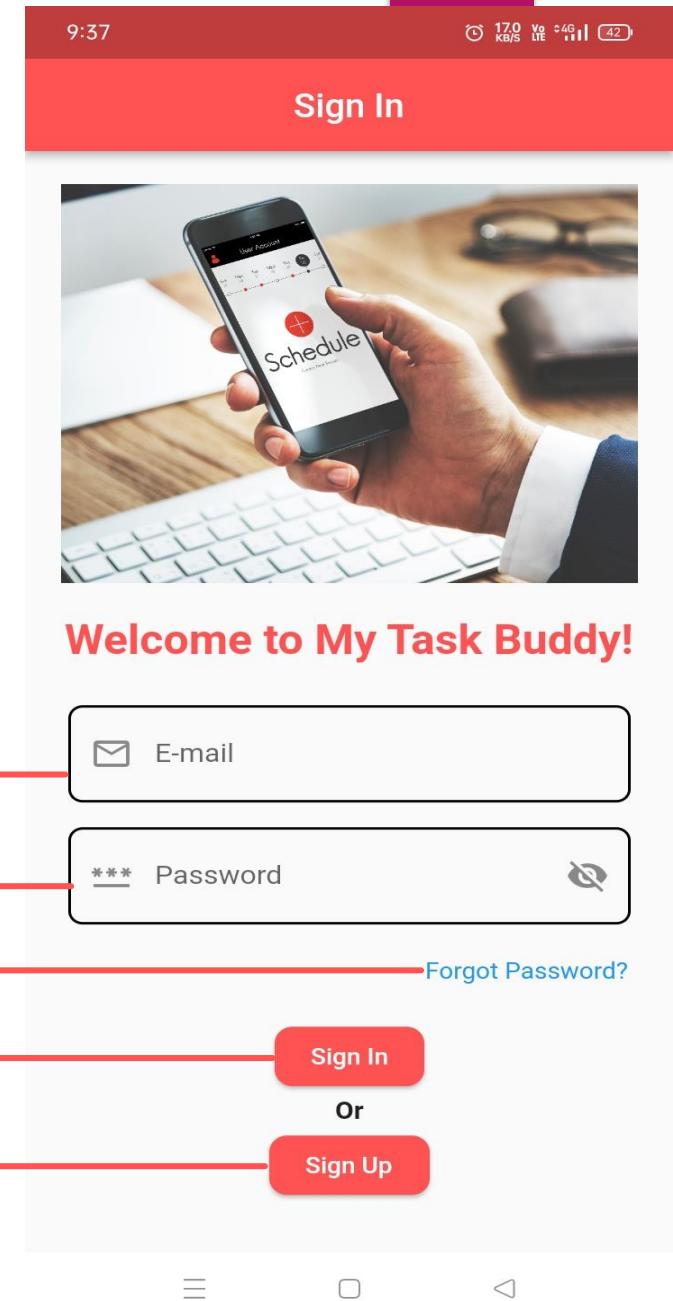
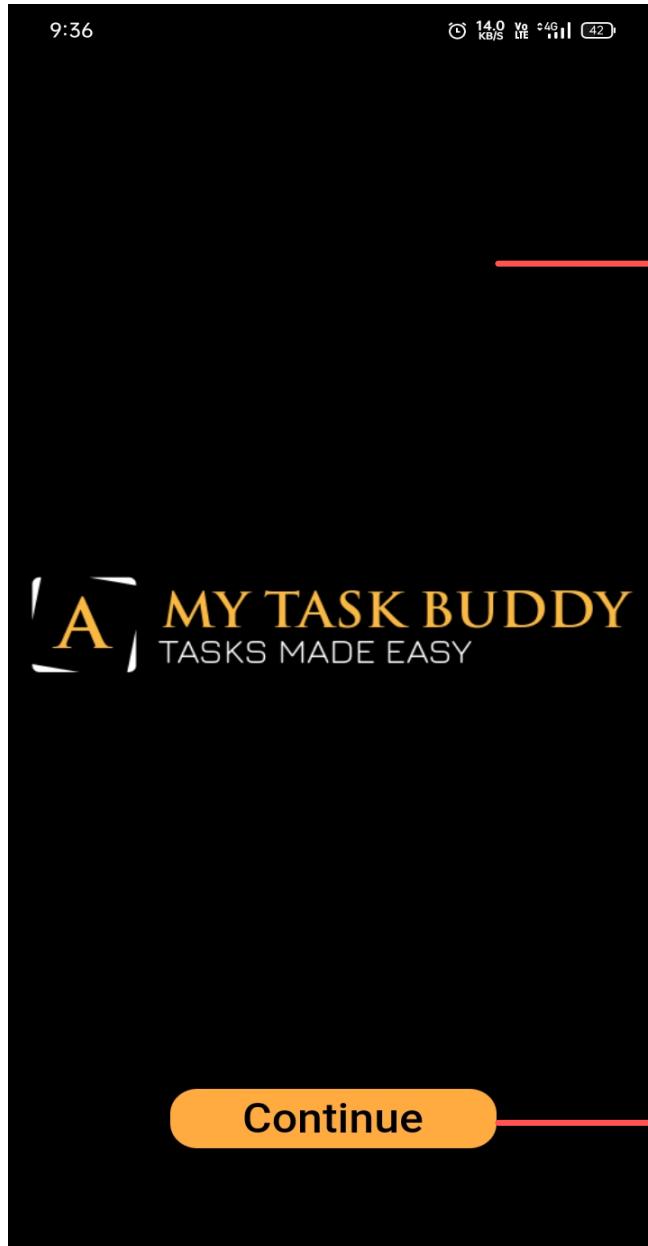


# ANDROID APP PROJECT

APP NAME: MY TASK BUDDY

BY ANSHU JHA





## Sign Up



**Join My Task Buddy!**

 Name Phone Number E-mail Password**Sign Up****Or****Already have an account? Sign In****Enter your details****Click to sign up****Click to sign in**

## Forgot Password

**Enter Your E-mail ID** E-mail**Enter your email id****Click to change password****Forgot Password**

[← Reset Password](#)

Please Enter a New Password

\*\*\* Password



Show/Hide password

\*\*\* Confirm Password



Enter details

Reset Password

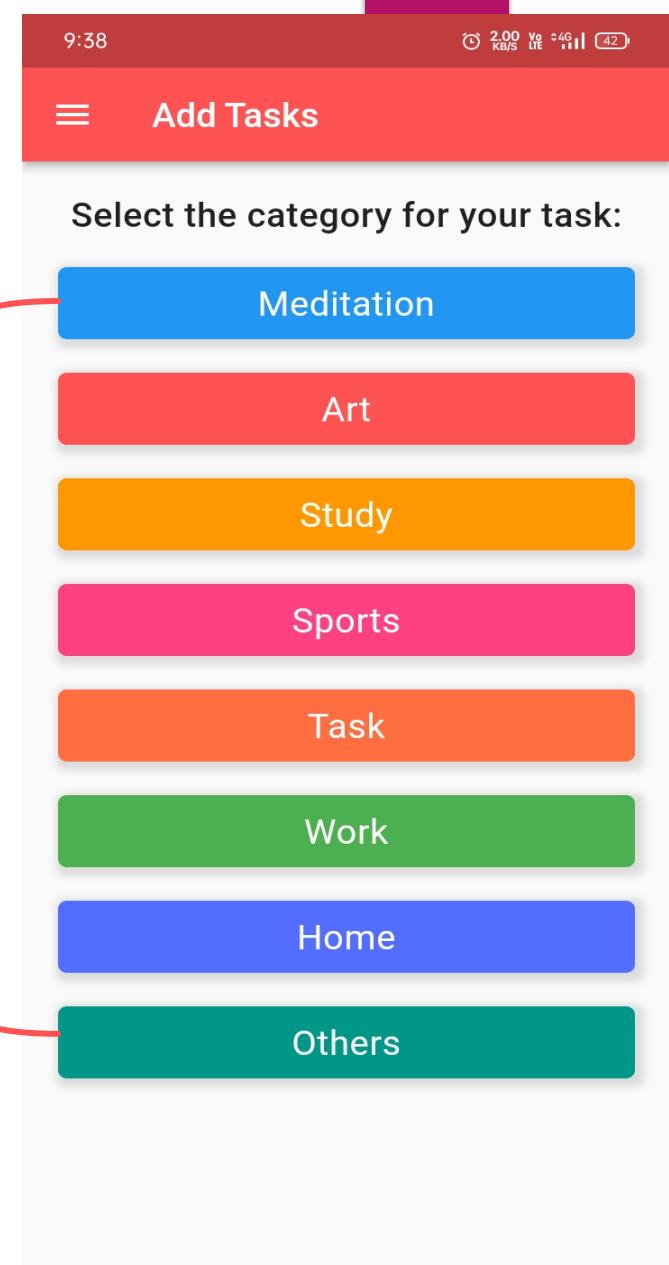
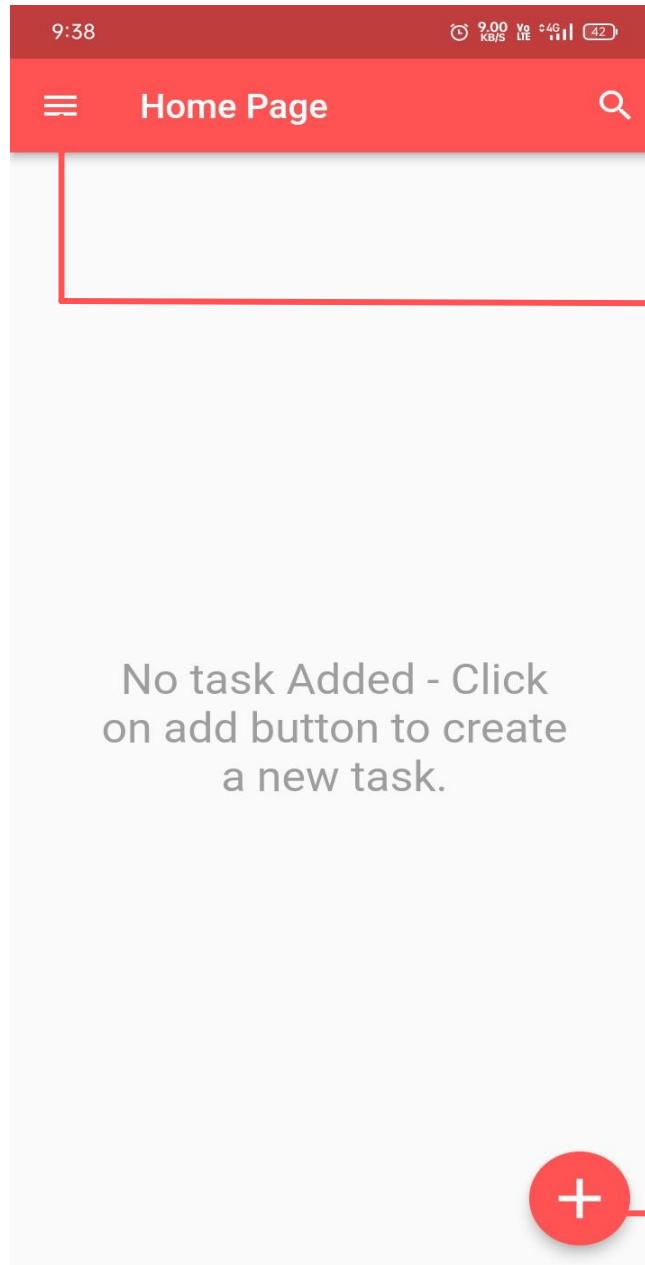
Click to reset password



Reset Done

Go to sign in page

Return to SignIn



9:38

0.96 KB/S LTE 4G 42%

### Meditation

#### Create New Task

Task Name

Task Description (Optional)

Task Date

Task Time

Priority

Medium

Category

Meditation

Reminder

Add Task

Search tasks by name, priority or category

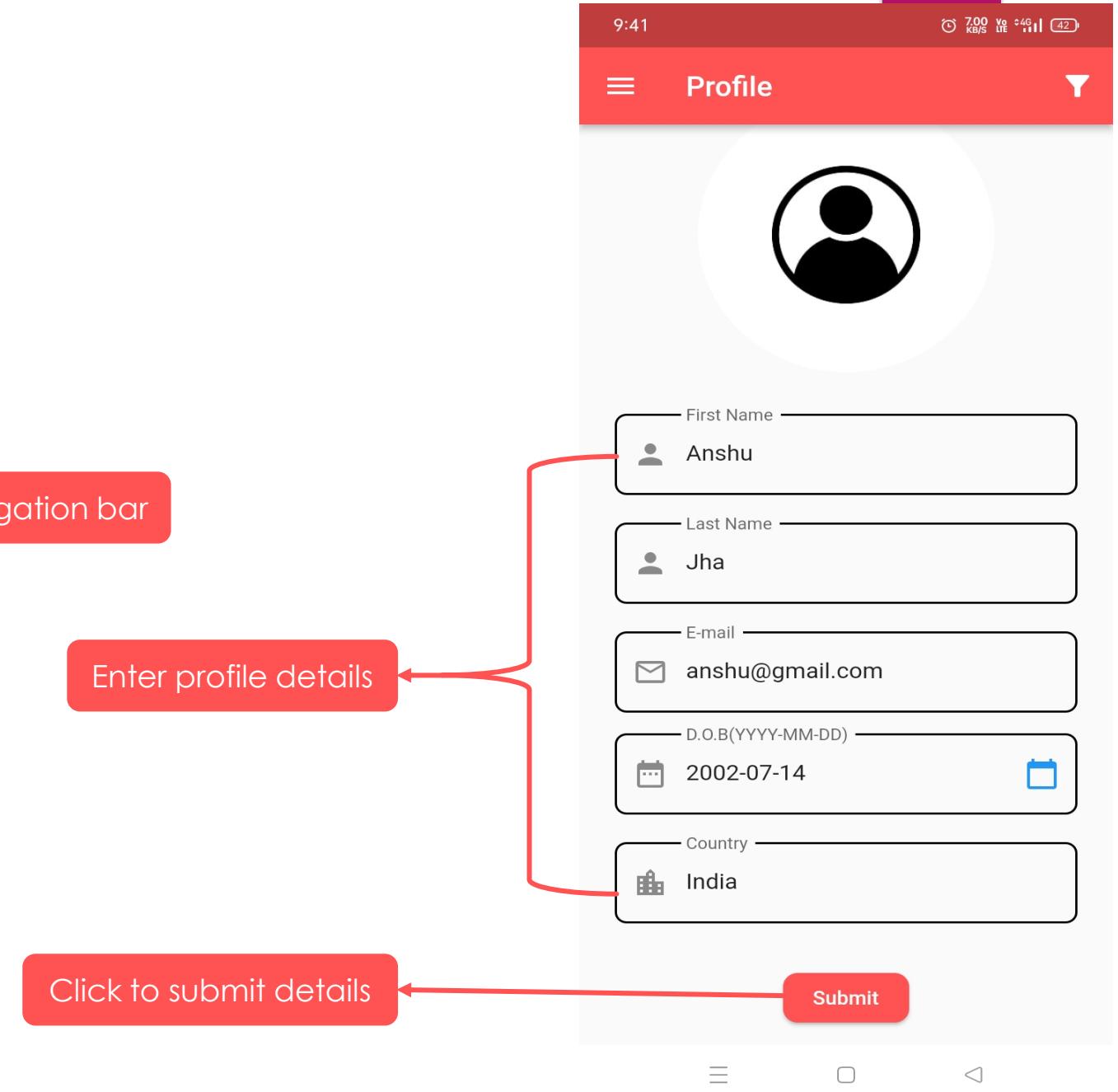
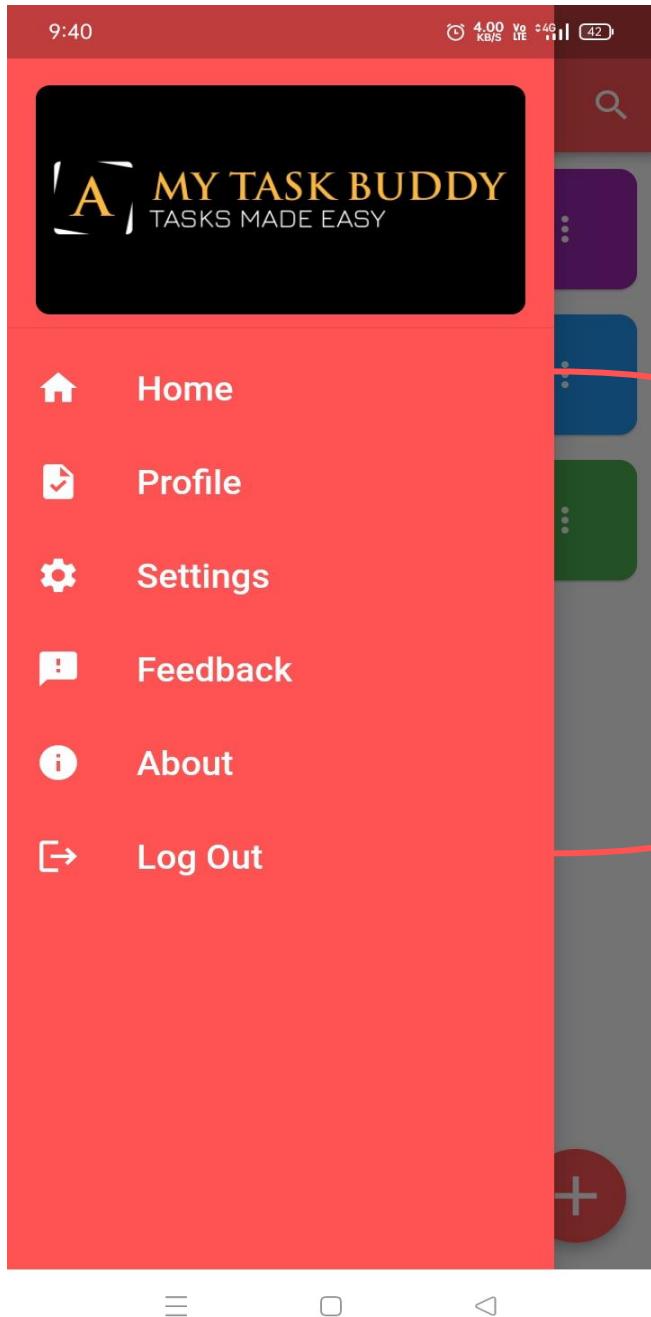
Open calendar

Enter details

Open clock

Click to add task

- 9:40
- 6.00 KB/S VoLTE 4G 42%
- ### Search
- MEDITATE  
High | Meditation
- DRAW PAINTING  
Medium | Art
- GROCERY SHOPPING  
Low | Task
- Edit
- Delete
- +
- Task done
- Edit task
- Delete task



9:41

7.00 4G 42

## About 'My Task Buddy'

### Dark Mode

Switch to dark themes



### Offline Mode

Use app features Offline



### Auto-Sync

Enable automatic synchronization



### Notification Sound

Turn on notifications



Control settings

Notification

Dark mode enabled

Notification enabled

my\_task\_buddy

3.00 4G 42

My Task Buddy  
Notification Sound Enabled

Dark Mode  
Switch to dark themes

Offline Mode  
Use app features Offline

Auto-Sync  
Enable automatic synchronization

Notification Sound  
Turn on notifications





## Feedback



**Your Feedback is Important!**

How do you rate this app?



Please leave your feedback below:

Enter your feedback here...(Optional)

Submit feedback  
for app

About  
app



## About 'My Task Buddy'



To do list , Routine Planner, and Reminders. All in just One app!

**Introducing Task Buddy - Your Smart Task Manager**

Your Daily Task Manager. Simplify your life with our intuitive to-do list app. Plan tasks, set reminders, and boost your productivity. Download now and make every day more organized and efficient

*Developed in 2023  
- by Anshu Jha*

# SOME PUBSPEC DEPENDENCIES USED IN PROJECT

- ▶ FLUTTER
- ▶ FLUTTER RATING BAR: FOR THE FEEDBACK PAGE
- ▶ FILE PICKER: FOR PROFILE PAGE
- ▶ INTL: FOR DATE FORMAT
- ▶ FLUTTER LOCAL NOTIFICATIONS: FOR THE SETTINGS PAGE
- ▶ TIME ZONE
- ▶ SHARED PREFERENCES
- ▶ PATH PROVIDER

# PUBSEC FILE:

The screenshot shows the Android Studio interface with the project 'my\_task\_buddy' open. The left sidebar displays the project structure, including files like drawee\_image.dart, notification\_service.dart, profile.dart, settings.dart, task\_model.dart, task\_provider.dart, theme\_provider.dart, view\_task.dart, and various platform-specific folders (linux, macos, test). The main editor window shows the contents of the pubspec.yaml file. The file defines a version of 1.0.0+1 and specifies dependencies for flutter, flutter\_rating\_bar (~4.0.0), file\_picker (~5.3.3), intl (~0.18.1), cupertino\_icons (~1.0.2), provider (~6.0.5), flutter\_local\_notifications (~15.1.0+1), timezone (~0.9.2), shared\_preferences (~2.2.0), and path\_provider (~2.1.0). The code is color-coded for readability.

```
version: 1.0.0+1
environment:
  sdk: '>=3.0.5 <4.0.0'

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.

dependencies:
  flutter:
    sdk: flutter
  flutter_rating_bar: ^4.0.0
  file_picker: ^5.3.3
  intl: ^0.18.1

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
provider: ^6.0.5
flutter_local_notifications: ^15.1.0+1
timezone: ^0.9.2
shared_preferences: ^2.2.0
path_provider: ^2.1.0
```

Document 1/1 > dependencies: > timezone: > ^0.9.2

42:19 CRLF UTF-8 2 spaces Schema: pubspec.json

1 30°C Haze

Search

Layout Inspector

ENG IN

20:46

14-08-2023

# MAIN PAGE

The screenshot shows the Android Studio interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** my\_task\_buddy - main.dart [my\_task\_buddy]
- Toolbars:** Device Manager, Notifications, Flutter Inspector, Flutter Performance, Flutter Outline, Device File Explorer.
- Left Sidebar:** Resource Manager, Project, Bookmarks, Build Variants, Structure.
- Code Editor:** The main.dart file is open, showing Dart code for a Flutter application. The code includes imports for ChangeNotifierProvider and TaskProvider, and defines a MultiProvider widget with providers for these two classes. It also includes a Consumer widget for ThemeProvider and a MaterialApp with a light theme.
- Bottom Navigation:** Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis.
- System Tray:** Shows weather (31°C Haze), search bar, system icons (B, S, M, F, E, A, P), network, battery, and date/time (20:06, 14-08-2023).

```
13
14
15 @override
16 Widget build(BuildContext context) {
17   return MultiProvider(
18     providers: [
19       ChangeNotifierProvider(create: (_) => ThemeProvider()), //ChangeNotifierProvider is allowing other widgets to listen to and interact with the provider
20       ChangeNotifierProvider(create: (_) => TaskProvider()), //Create is responsible for creating the class that you want to make available to other parts of your app
21     ],
22     child: Consumer<ThemeProvider>(()</Consumer> this widget listens to changes in the state of ThemeProvider and rebuilds its child widget when the state changes {
23       builder: (context, themeProvider, child) {
24         return MaterialApp(
25           debugShowCheckedModeBanner: false,
26           title: 'My Task Buddy',
27           theme: ThemeData.light(),
28           darkTheme: ThemeData.dark(),
29           themeMode: themeProvider.getThemeMode() == ThemeModeType.dark
30             ? ThemeMode.dark
31             : ThemeMode.light,
32           home: const Splash(),
33         ); // MaterialApp
34       }, // Consumer
35     ); // MultiProvider
36   }
37 }
38 }
```

# SPLASH PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file is "splash.dart" located in the "lib" directory. The code defines a "Splash" StatefulWidget and its corresponding "\_SplashState" state class. The splash screen features a black background with a centered logo image and a floating action button at the bottom right.

```
import 'package:flutter/material.dart';
import 'package:my_task_buddy/home_page.dart';
import 'package:my_task_buddy/sign_in.dart';
import 'package:shared_preferences/shared_preferences.dart';

class Splash extends StatefulWidget {
    const Splash({super.key});

    @override
    State<Splash> createState() => _SplashState();
}

class _SplashState extends State<Splash> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.black,
            body: const Center(
                child: Image(
                    image: AssetImage("assets/images/logo.png"), // Image
                ), // Center
            floatingActionButton: Padding(
                padding: const EdgeInsets.only(bottom: 40, left: 33),
                child: Align(
                    alignment: Alignment.bottomCenter,
                    child: ElevatedButton(
                        onPressed: () async {
                            _handleButtonClick();
                        },
                        style: ElevatedButton.styleFrom(

```

The IDE interface includes a Project tree on the left, a toolbar with various icons at the top, and several toolbars at the bottom for Version Control, Profiler, App Quality Insights, TODO, Problems, Terminal, Services, Dart Analysis, Layout Inspector, and system status indicators like battery and connectivity.

# SIGN IN PAGE

The screenshot shows a development environment for a Flutter application named "my\_task\_buddy". The main window displays the code for the "sign\_in.dart" file, which contains the implementation for a sign-in screen. The code includes a stateful widget "SignIn" with a nested state class "\_SignInState". The build method of "SignIn" returns a "WillPopScope" widget, which handles a dialog for quitting the app. The "assets/images" directory is shown containing files like feedback.png, forgotPassword.jpg, logo.png, no-task-1.jpg, profile-logo.png, resetDone.jpg, and task.png.

```
const SignIn({super.key});  
@override  
State<SignIn> createState() => _SignInState();  
  
class _SignInState extends State<SignIn> {  
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();  
  TextEditingController emailController = TextEditingController();  
  TextEditingController passwordController = TextEditingController();  
  bool _showPassword = false;  
  
  @override  
  Widget build(BuildContext context) {  
    return WillPopScope(  
      onWillPop: () async {  
        showDialog(  
          context: context,  
          builder: (context) => AlertDialog(  
            title: const Text("My Task Buddy"),  
            content: const Text("Do you want to quit the app?"),  
            shape: RoundedRectangleBorder(  
              borderRadius: BorderRadius.circular(10.0),  
            ),  
            actions: [  
              ElevatedButton(  
                onPressed: () {  
                  Navigator.pop(context);  
                },  
                style: ElevatedButton.styleFrom(  
                  backgroundColor: Colors.redAccent,  
                ),  
              ),  
            ],  
          );  
      },  
    );  
  }  
}  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40
```

The bottom status bar shows system information including temperature (30°C Haze), battery level, network connectivity, and the current date and time (14-08-2023).

# SIGN UP PAGE

The screenshot shows a Flutter project in an IDE. The project structure on the left includes files like `about.dart`, `add_new_task.dart`, `category_page.dart`, `drawer.dart`, `edit_task.dart`, `feedback.dart`, `forgot_password.dart`, `home_page.dart`, `main.dart`, `notification_service.dart`, `profile.dart`, `reset_done.dart`, `reset_password.dart`, `settings.dart`, `sign_in.dart`, `sign_up.dart`, `splash.dart`, `task_model.dart`, `task_provider.dart`, `theme_provider.dart`, and `view_task.dart`. The `sign_up.dart` file is open in the editor, showing the code for the `SignUp` stateful widget. The code handles form key management and a will-pop scope for a dialog.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:my_task_buddy/sign_in.dart';
import 'package:my_task_buddy/home_page.dart';
import 'package:shared_preferences/shared_preferences.dart';

class SignUp extends StatefulWidget {
    const SignUp({super.key});

    @override
    State<SignUp> createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
    final _formKey = GlobalKey<FormState>();
    TextEditingController nameController = TextEditingController();
    TextEditingController numberController = TextEditingController();
    TextEditingController emailController = TextEditingController();
    TextEditingController passwordController = TextEditingController();
    bool _showPassword = false;

    @override
    Widget build(BuildContext context) {
        return WillPopScope(
            onWillPop: () async {
                showDialog(
                    context: context,
                    builder: (context) => AlertDialog(
                        title: const Text("My Task Buddy"),
                        content: const Text("Do you want to quit the app?"),
                        shape: RoundedRectangleBorder(

```

At the bottom, the status bar shows the weather (30°C Haze), system icons, and the date/time (14-08-2023).

# FORGOT PASSWORD PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file is "forgot\_password.dart" located in the "lib" directory. The code defines a "ForgotPassword" stateful widget and its corresponding state class, "\_ForgotPasswordState". The build method returns a Scaffold with an AppBar titled "Forgot Password" and a body containing a Column with a Container holding an image.

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

class ForgotPassword extends StatefulWidget {
    const ForgotPassword({super.key});

    @override
    State<ForgotPassword> createState() => _ForgotPasswordState();
}

class _ForgotPasswordState extends State<ForgotPassword> {
    final _formKey = GlobalKey<FormState>();
    TextEditingController emailController = TextEditingController();

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text("Forgot Password"),
                backgroundColor: Colors.redAccent,
                foregroundColor: Colors.white,
            ),
            body: SingleChildScrollView(
                child: Column(
                    children: [
                        Container(
                            padding: EdgeInsets.all(20),
                            child: const Image(
                                image: AssetImage("assets/images/forgotPassword.jpg"),
                            ),
                        ),
                    ],
                ),
            ),
        );
    }
}
```

The IDE interface includes a Project tree on the left, a toolbar at the top with tabs like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a status bar at the bottom showing system information and connectivity.

# RESET PASSWORD PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file is "reset\_password.dart" located in the "lib" directory. The code implements a "ResetPassword" stateful widget that displays a "Reset Password" screen with an AppBar, a Column body containing a Container with padding, and an Image child.

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

class ResetPassword extends StatefulWidget {
    const ResetPassword({super.key});

    @override
    State<ResetPassword> createState() => _ResetPasswordState();
}

class _ResetPasswordState extends State<ResetPassword> {
    final _formKey = GlobalKey<FormState>();
    TextEditingController passwordController = TextEditingController();
    TextEditingController verifyPasswordController = TextEditingController();
    bool _showPassword = false;
    bool _showVerifyPassword = false;

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text("Reset Password"),
                backgroundColor: Colors.redAccent,
                foregroundColor: Colors.white,
            ), // AppBar
            body: SingleChildScrollView(
                child: Column(
                    children: [
                        Container(
                            padding: EdgeInsets.all(20),
                            child: const Image(
                                image: AssetImage("assets/icon.png"),
                            ),
                        ),
                    ],
                ),
            ),
        );
    }
}
```

The IDE interface includes a Project sidebar, a central code editor with syntax highlighting, and various toolbars and panels for device management, notifications, and performance analysis.

# RESET DONE PAGE

The screenshot shows the Android Studio interface with a Flutter project titled "my\_task\_buddy". The "lib" directory contains several Dart files, including "reset\_done.dart", which is currently open in the code editor. The code implements a "ResetDone" stateful widget that handles a will-pop event to show a confirmation dialog before exiting the app.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:my_task_buddy/sign_in.dart';

class ResetDone extends StatefulWidget {
    const ResetDone({super.key});

    @override
    State<ResetDone> createState() => _ResetDoneState();
}

class _ResetDoneState extends State<ResetDone> {
    @override
    Widget build(BuildContext context) {
        return WillPopScope(
            onWillPop: () async {
                showDialog(
                    context: context,
                    builder: (context) => AlertDialog(
                        title: const Text("My Task Buddy"),
                        content: const Text("Do you want to quit the app?"),
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10.0),
                        ),
                        actions: [
                            ElevatedButton(
                                onPressed: () {
                                    Navigator.pop(context);
                                },
                                style: ElevatedButton.styleFrom(
                                    backgroundColor: Colors.redAccent,
                                )
                            )
                        ],
                    )
                );
            }
        );
    }
}
```

The bottom status bar shows system information like battery level (30°C Haze), network connectivity, and the current date and time (14-08-2023).

# DRAWER PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file being edited is "drawer.dart" located in the "lib" directory. The code implements a Drawer widget with a ListView containing a DrawerHeader and several ListTile items.

```
import 'package:flutter/material.dart';
import 'package:my_task_buddy/about.dart';
import 'package:my_task_buddy/feedback.dart';
import 'package:my_task_buddy/home_page.dart';
import 'package:my_task_buddy/profile.dart';
import 'package:my_task_buddy/settings.dart';
import 'package:my_task_buddy/sign_in.dart';
import 'package:shared_preferences/shared_preferences.dart';

Widget appDrawer(context) {
  return Drawer(
    child: Container(
      color: Colors.redAccent,
      child: ListView(
        children: [
          DrawerHeader(
            child: Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(8),
                color: Colors.black,
                image: const DecorationImage(
                  image: AssetImage("assets/images/logo.png"))), // DecorationImage, BoxDecoration
          ), // Container, DrawerHeader
          ListTile(
            leading: const Icon(Icons.home, color: Colors.white, size: 25),
            title: const Text('Home',
              style: TextStyle(
                fontSize: 20,
                color: Colors.white,
            )), // TextStyle, Text
            onTap: () {

```

The Project sidebar shows other files like main.dart, home\_page.dart, and category\_page.dart. The Build Variants sidebar shows configurations for linux, macos, and web. The bottom status bar includes a weather icon (30°C Haze), system icons, and a timestamp (14-08-2023).

# THEME PROVIDER PAGE

The screenshot shows the Android Studio interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** my\_task\_buddy > lib > theme\_provider.dart
- Toolbars:** Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis.
- Left Sidebar:** Resource Manager (Build, iOS, lib), Project (about.dart, add\_new\_task.dart, category\_page.dart, drawer.dart, edit\_task.dart, feedback.dart, forgot\_password.dart, home\_page.dart, main.dart, notification\_service.dart, profile.dart, reset\_done.dart, reset\_password.dart, settings.dart, sign\_in.dart, sign\_up.dart, splash.dart, task\_model.dart, task\_provider.dart, theme\_provider.dart, view\_task.dart), Bookmarks, Build Variants, Structure.
- Central Area:** Code editor showing theme\_provider.dart. The code defines an enum `ThemeModeType` and a class `ThemeProvider` that extends `ChangeNotifier`. It includes methods to get and set the theme mode.
- Right Sidebar:** Device Manager, Notifications, Flutter Inspector, Flutter Performance, Flutter Outline, Device File Explorer.
- Bottom Bar:** Layout Inspector, 18:4, CRLF, UTF-8, 2 spaces, 30°C Haze, ENG IN, 14-08-2023, 20:24.

# HOME PAGE

The screenshot shows the Android Studio interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** my\_task\_buddy - home\_page.dart [my\_task\_buddy].
- Toolbars:** Resource Manager, Project, Bookmarks, Build Variants, Structure, Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis.
- Code Editor:** The code for `home_page.dart` is displayed. It imports various packages and defines two classes: `HomePage` and `_HomePageState`.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:my_task_buddy/category_page.dart';
import 'package:my_task_buddy/drawer.dart';
import 'package:my_task_buddy/task_model.dart';
import 'package:my_task_buddy/task_provider.dart';
import 'package:my_task_buddy/view_task.dart';
import 'package:provider/provider.dart';
import 'edit_task.dart';

class HomePage extends StatefulWidget {
    const HomePage({super.key});

    @override
    State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
    bool _isSearchBarVisible = false;
    final TextEditingController _searchController = TextEditingController();

    void _toggleSearchBar() {
        setState(() {
            _isSearchBarVisible = !_isSearchBarVisible;
            if (!_isSearchBarVisible) {
                _searchController.clear();
            }
        });
    }

    @override
```
- Tool Windows:** Device Manager, Notifications, Flutter Inspector, Flutter Performance, Flutter Outline, Device File Explorer.
- System Bar:** Shows weather (30°C Haze), search bar, system icons, and system status (20:21, ENG IN, 14-08-2023).

# CATEGORY PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file is "category\_page.dart" located in the "lib" directory. The code defines a "CategorySelect" StatefulWidget and its corresponding state class, "\_CategorySelectState". The state class contains a build method that returns a Scaffold with a red AppBar titled "Add Tasks" and a white body containing a Column with a Container and a Text widget asking to select a category.

```
import 'package:flutter/material.dart';
import 'drawer.dart';
import 'package:my_task_buddy/add_new_task.dart';

class CategorySelect extends StatefulWidget {
    const CategorySelect({super.key});

    @override
    State<CategorySelect> createState() => _CategorySelectState();
}

class _CategorySelectState extends State<CategorySelect> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text("Add Tasks"),
                backgroundColor: Colors.redAccent,
                foregroundColor: Colors.white,
            ),
            drawer: appDrawer(context),
            body: SingleChildScrollView(
                child: Column(
                    children: [
                        Container(
                            padding: const EdgeInsets.only(top: 20),
                            child: const Text(
                                "Select the category for your task:",
                                style: TextStyle(
                                    fontSize: 20,
                                ),
                            ),
                        )
                    ],
                ),
            ),
        );
    }
}
```

The IDE interface includes a Project tree on the left, a toolbar at the top with tabs like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a bottom navigation bar with icons for Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis, and Layout Inspector. The status bar at the bottom shows system information like battery level, network, and date.

# TASK MODEL

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The "task\_model.dart" file is the active code editor. The code defines a "Task" class with properties like id, name, description, date, priority, reminders, isDone, selectedTime, and category. It also includes a constructor and a toJson() method. The IDE interface includes a Project sidebar, a Resource Manager, and various toolbars and panels.

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

class Task {
    final String id;
    final String name;
    final String description;
    final DateTime date;
    final String priority;
    final bool reminders;
    bool isDone;
    TimeOfDay selectedTime;
    final String category;

    Task({
        required this.id,
        required this.name,
        required this.description,
        required this.date,
        required this.priority,
        required this.reminders,
        required this.isDone,
        required this.selectedTime,
        required this.category,
    });

    // Convert Task to JSON
    Map<String, dynamic> toJson() {
        return {
            'id': id,
            'name': name,
            'description': description,
        };
    }
}
```

IDE UI Elements:

- File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
- my\_task\_buddy - task\_model.dart [my\_task\_buddy]
- Project Manager: build, ios, lib (selected), about.dart, add\_new\_task.dart, category\_page.dart, drawer.dart, edit\_task.dart, feedback.dart, forgot\_password.dart, home\_page.dart, main.dart, notification\_service.dart, profile.dart, reset\_done.dart, reset\_password.dart, settings.dart, sign\_in.dart, sign\_up.dart, splash.dart, task\_model.dart (selected), task\_provider.dart, theme\_provider.dart, view\_task.dart, linux, macos, test, widget\_test.dart
- Resource Manager: .flutter-plugins, .flutter-plugins-dependencies, .gitignore, .metadata
- Device Manager: <no device selected>, main.dart, Pixel 5 API 30
- Notifications, Flutter Inspector, Flutter Performance, Flutter Outline, Device File Explorer
- Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis
- Layout Inspector
- System tray: 30°C Haze, Search, Microsoft Edge, File Explorer, Task View, Taskbar icons, Language: ENG IN, Date: 14-08-2023, Time: 20:24

# TASK PROVIDER

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The code editor displays the file `task_provider.dart` under the `lib` directory. The code implements a `TaskProvider` class that uses `ChangeNotifier`. It handles task storage using `SharedPreferences`, loading tasks from JSON and saving them back to JSON. The IDE interface includes a Project sidebar, a Resource Manager, and various toolbars and status bars at the bottom.

```
import 'package:flutter/material.dart';
import 'package:my_task_buddy/task_model.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:convert';

class TaskProvider with ChangeNotifier {
    final List<Task> _tasks = [];
    SharedPreferences? _prefs;

    TaskProvider() {
        _loadTasks();
    }

    Future<void> _initSharedPreferences() async {
        _prefs = await SharedPreferences.getInstance();
    }

    Future<void> _loadTasks() async {
        await _initSharedPreferences();
        final tasksJson = _prefs?.getString('tasks');
        if (tasksJson != null) {
            final List<dynamic> taskListJson = jsonDecode(tasksJson);
            _tasks.addAll(taskListJson.map((taskJson) => Task.fromJson(taskJson)));
            notifyListeners();
        }
    }

    Future<void> _saveTasks() async {
        await _initSharedPreferences();
        final taskListJson = _tasks.map((task) => task.toJson()).toList();
        _prefs?.setString('tasks', jsonEncode(taskListJson));
    }
}
```

# ADD TASK PAGE

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help my\_task\_buddy - add\_new\_task.dart [my\_task\_buddy]

my\_task\_buddy > lib > add\_new\_task.dart

Project Manager

Resource Manager

Project

Bookmarks

Build Variants

Structure

Version Control Profiler Logcat App Quality Insights

TODO Problems Terminal Services App Inspection Dart Analysis

48:62 CRLF UTF-8 2 spaces Layout Inspector

30°C Haze

Search

Device Manager Notifications Flutter Inspector Flutter Performance Flutter Outline Device File Explorer

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:my_task_buddy/drawer.dart';
import 'package:my_task_buddy/task_model.dart';
import 'package:my_task_buddy/task_provider.dart';
import 'package:provider/provider.dart';
import 'home_page.dart';

class AddTask extends StatefulWidget {
    final String categoryName;

    const AddTask({required this.categoryName, Key? key}) : super(key: key);

    @override
    State<AddTask> createState() => _AddTaskState();
}

class _AddTaskState extends State<AddTask> {
    late String _selectedCategory = '';
    final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

    TextEditingController taskNameController = TextEditingController();
    TextEditingController dateController = TextEditingController();
    TextEditingController timeController = TextEditingController();
    TextEditingController descriptionController = TextEditingController();
    TextEditingController categoryController = TextEditingController();
    String _selectedPriority = 'Medium';
    DateTime _selectedDate = DateTime.now();
    TimeOfDay _selectedTime = TimeOfDay.now();
    bool _reminders = true;
}
```

14-08-2023 20:21

# EDIT TASK PAGE

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help my\_task\_buddy - edit\_task.dart [my\_task\_buddy]

my\_task\_buddy > lib > edit\_task.dart

Project Manager

Resource Manager

Project

Bookmarks

Build Variants

Structure

edit\_task.dart

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:my_task_buddy/task_model.dart';
import 'package:my_task_buddy/task_provider.dart';
import 'package:provider/provider.dart';

import 'home_page.dart';

class EditTaskPage extends StatefulWidget {
    final Task task;

    const EditTaskPage({required this.task, Key? key}) : super(key: key);

    @override
    State<EditTaskPage> createState() => _EditTaskPageState();
}

class _EditTaskPageState extends State<EditTaskPage> {
    late TextEditingController taskNameController;
    late TextEditingController descriptionController;
    late TextEditingController dateController;
    late TextEditingController timeController;
    late TextEditingController categoryController;
    late String _selectedPriority;
    late DateTime _selectedDate;
    late TimeOfDay _selectedTime;
    late bool _reminders;

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

Version Control Profiler Logcat App Quality Insights TODO Problems Terminal Services Dart Analysis

228:35 CRLF UTF-8 2 spaces Layout Inspector

30°C Haze

Search

Device Manager Notifications Flutter Inspector Flutter Performance Flutter Outline Device File Explorer

ENG IN 20:23 14-08-2023

# VIEW TASK PAGE

The screenshot shows a Flutter project named 'my\_task\_buddy' open in an IDE. The 'view\_task.dart' file is the active code editor. The code defines a StatelessWidget named 'TaskDetailsScreen' that displays task details with a red-themed Scaffold.

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:my_task_buddy/task_model.dart';

class TaskDetailsScreen extends StatelessWidget {
    final Task task;

    const TaskDetailsScreen({required this.task, Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text("Task Details"),
                backgroundColor: Colors.redAccent,
                foregroundColor: Colors.white,
            ),
            body: Padding(
                padding: const EdgeInsets.all(20),
                child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        _buildRow('Task Name:', task.name),
                        const SizedBox(height: 20),
                        _buildRow('Date:', DateFormat('yyyy-MM-dd').format(task.date)),
                        const SizedBox(height: 20),
                        _buildRow('Time:', formatTime(task.selectedTime)),
                        const SizedBox(height: 20),
                        _buildRow('Priority:', task.priority),
                        const SizedBox(height: 20),
                        _buildRow('Category', task.category),
                    ],
                ),
            ),
        );
    }

    Row _buildRow(String label, String value) {
        return Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Text(label),
                Text(value),
            ],
        );
    }
}
```

The IDE interface includes a Project sidebar, a central code editor, and various toolbars and panels for device management, notifications, and performance monitoring.

# PROFILE PAGE

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help my\_task\_buddy - profile.dart [my\_task\_buddy]

my\_task\_buddy > lib > profile.dart

Project Manager Device Manager

Resource Manager Project Notifications

Bookmarks Flutter Inspector

Build Variants Flutter Performance

Structure Flutter Outline

Version Control .gitignore Layout Inspector

Profiler Logcat App Quality Insights TODO Problems Terminal Services App Inspection Dart Analysis

30°C ENG IN 14-08-2023

Search

Pixel 5 API 30

Profile Page

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:intl/intl.dart';
import 'package:my_task_buddy/drawer.dart';
import 'package:my_task_buddy/home_page.dart';
import 'package:shared_preferences/shared_preferences.dart';

class ProfilePage extends StatefulWidget {
    const ProfilePage({super.key});

    @override
    State<ProfilePage> createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
    TextEditingController firstNameController = TextEditingController();
    TextEditingController lastNameController = TextEditingController();
    TextEditingController emailController = TextEditingController();
    TextEditingController dateOfBirthController = TextEditingController();
    TextEditingController countryController = TextEditingController();
    final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
    DateTime _selectedDate = DateTime.now();
    late File? selectedImage;
    late String? imagePath;
    late SharedPreferences _prefs;

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

# ABOUT PAGE

The screenshot shows a development environment with the following details:

- Project Structure:** The project is named "my\_task\_buddy". The `lib` directory contains several files: `about.dart` (selected), `add_new_task.dart`, `category_page.dart`, `drawer.dart`, `edit_task.dart`, `feedback.dart`, `forgot_password.dart`, `home_page.dart`, `main.dart`, `notification_service.dart`, `profile.dart`, `reset_done.dart`, `reset_password.dart`, `settings.dart`, `sign_in.dart`, `sign_up.dart`, `splash.dart`, `task_model.dart`, `task_provider.dart`, `theme_provider.dart`, and `view_task.dart`. There are also `build`, `ios`, and `test` directories.
- Code Editor:** The `about.dart` file is open. It defines a `AboutPage` widget that extends `StatefulWidget`. The `build` method returns a `WillPopScope` widget containing an `AlertDialog` for quitting the app. The `actions` field of the `AlertDialog` includes an `ElevatedButton` with a red background color.
- Toolbars and Status Bar:** The top bar includes standard menu items like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and tabs for `main.dart` and `Pixel 5 API 30`. The bottom bar shows various developer tools: Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, Dart Analysis, Layout Inspector, and system status indicators for battery, signal, and time (6:32 CRLF UTF-8 2 spaces).
- System Tray:** The bottom left shows a weather icon (30°C Haze) and a date/time stamp (14-08-2023 20:23).

# FEEDBACK PAGE

The screenshot shows a Flutter project named "my\_task\_buddy" open in an IDE. The current file is "feedback.dart" located in the "lib" directory. The code implements a "FeedbackPage" stateful widget that extends "StatefulWidget". It contains a "createState" method returning a "\_FeedbackPageState" object. The "build" method returns a "WillPopScope" widget with an "onWillPop" handler that shows an "AlertDialog" asking if the user wants to quit the app. The "AlertDialog" has a rounded rectangle border and a circular border radius of 10.0. The "actions" field is currently empty.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:my_task_buddy/drawer.dart';
import 'package:flutter_rating_bar/flutter_rating_bar.dart';
import 'package:my_task_buddy/home_page.dart';

class FeedbackPage extends StatefulWidget {
    const FeedbackPage({super.key});

    @override
    State<FeedbackPage> createState() => _FeedbackPageState();
}

class _FeedbackPageState extends State<FeedbackPage> {
    double _rating = 0.0;
    final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
    bool _showRatingError = false;

    @override
    Widget build(BuildContext context) {
        return WillPopScope(
            onWillPop: () async {
                showDialog(
                    context: context,
                    builder: (context) => AlertDialog(
                        title: const Text("My Task Buddy"),
                        content: const Text("Do you want to quit the app?"),
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(10.0),
                        ),
                        actions: [
                    ),
            ),
        );
    }
}
```

The IDE interface includes a Project tree on the left, a toolbar at the top, and various inspection and analysis tools along the bottom.

# SETTINGS PAGE

The screenshot shows the Android Studio interface with a Flutter project named "my\_task\_buddy". The "lib" directory contains several Dart files, and the "settings.dart" file is currently selected and open in the editor. The code implements a StatefulWidget for the settings page, initializing notification service providers and handling theme mode.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:my_task_buddy/theme_provider.dart';
import 'package:provider/provider.dart';
import 'notification_service.dart';

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

    @override
    State<SettingsPage> createState() => _SettingsPageState();
}

class _SettingsPageState extends State<SettingsPage> {
    bool _offlineModeEnabled = true;
    bool _autoSyncEnabled = false;
    bool _notificationSoundEnabled = false;
    late bool _darkModeEnabled;

    @override
    void initState() {
        super.initState();
        final themeProvider = Provider.of<ThemeProvider>(context, listen: false);
        _darkModeEnabled = themeProvider.getThemeMode() == ThemeModeType.dark;
        _initializeNotificationSettings();
    }

    Future<void> _initializeNotificationSettings() async {
        await NotificationService().init();
    }
}
```

The status bar at the bottom shows system information like battery level (30°C Haze), network, and date (14-08-2023).

# NOTIFICATION SERVICE

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar displays the project structure for "my\_task\_buddy". It includes the "lib" directory containing files like "about.dart", "add\_new\_task.dart", "category\_page.dart", "drawer.dart", "edit\_task.dart", "feedback.dart", "forgot\_password.dart", "home\_page.dart", "main.dart", "notification\_service.dart", "profile.dart", "reset\_done.dart", "reset\_password.dart", "settings.dart", "sign\_in.dart", "sign\_up.dart", "splash.dart", "task\_model.dart", "task\_provider.dart", "theme\_provider.dart", and "view\_task.dart". Other directories like "build", "ios", "linux", "macos", "test", "web", "windows", and ".flutter-plugins" are also visible.
- Code Editor:** The main editor window shows the content of "notification\_service.dart". The code imports the necessary Flutter Local Notifications plugin and defines a static factory method to return an instance of the NotificationService. It then initializes the plugin and creates an Android notification channel named "My Task Buddy".

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';

final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin(); //An instance used to interact with the local notification system.

class NotificationService {
    static final NotificationService _notificationService =
        NotificationService._internal();

    factory NotificationService() {
        return _notificationService;
    }

    NotificationService._internal();

    Future<void> init() async {
        const AndroidInitializationSettings initializationSettingsAndroid =
            AndroidInitializationSettings('logo1');
        const InitializationSettings initializationSettings =
            InitializationSettings(android: initializationSettingsAndroid);

        await flutterLocalNotificationsPlugin.initialize(initializationSettings);

        const AndroidNotificationChannel channel = AndroidNotificationChannel(
            'My Task Buddy',
            'Tasks Made Easy',
            importance: Importance.max,
            playSound: true,
        );
        await flutterLocalNotificationsPlugin
            .resolvePlatformSpecificImplementation<
```
- Toolbars and Status Bar:** The top bar shows tabs for "my\_task\_buddy - notification\_service.dart [my\_task\_buddy]", "main.dart", and "Pixel 5 API 30". The bottom bar includes icons for Version Control, Profiler, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, App Inspection, and Dart Analysis. The status bar at the bottom right shows the date (14-08-2023), time (10:34), and battery level (20:23).