

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI – 590 018



**An Internship Project Report
on
*Plant App***

Submitted in partial fulfillment of the requirements for the VII Semester of degree
of **Bachelor of Engineering in Information Science and Engineering** of
Visvesvaraya Technological University, Belagavi

Submitted By

JyotsnaSingh
1RN18IS057

Under the Guidance of

Mrs. Chandana Rani
Assistant Professor
Department of ISE



Department of Information Science and Engineering

RNS Institute of Technology
Dr. Vishnuvardhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



CERTIFICATE

Certified that the Internship work entitled **Plant App** has been successfully completed by **Jyotsna Singh (1RN18IS057)** a Bonafede student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Chandana Rani

Internship Guide
Assistant Professor
Department of ISE

Dr. Suresh L

Professor and HoD
Department of ISE
RNSIT

Dr. M K Venkatesha

Principal
RNSIT

External Viva

Name of the Examiners

Signature with Date

1. _____

1. _____

2. _____

2. _____

DECLARATION

I, **Jyotsna Singh** [USN: **1RN18IS057**] student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Plant App*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

Jyotsna Singh (1RN18IS057)

ABSTRACT

We propose online plant exploration now which enables customers to straightforwardly know about all the different kinds of plants species available. It's practically not possible to go physically and explore these things. Our app gives an overall brief idea about the products available along with other stuffs such as sprays or pots etc. This makes life much easier for plant lovers .

There are many articles available currently on internet. But there are few Which provide better understanding of different species of plants ,seeds and fertilizers or sprayers used for the same all at one place. To develop a User friendly plant application which will contain description of products.

To develop a application which will contain solution to the above problems. By this application the user will come to know about varieties of plants and seeds etc and can learn additional knowledge. Also by this application a user can expand his/her knowledge among the world.

ACKNOWLEDGMENT

At the very onset we would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is, they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha**, for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mrs. ChandanaRani** Assistant Professor and HOD, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R, ENMAZ**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar**, Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

TABLE OF CONTENTS

CERTIFICATE	ii
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGMENT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	vii
ABBREVIATIONS	viii
1. INTRODUCTION	1
1.1 Introduction To Flutter	1
1.2 History	1
1.3 Frame Work Architecture	2
2. LITERATURE SURVEY	4
2.1 Plant App	4
2.1.1 Introduction	4
3. ANALYSIS	6
3.1 Hardware and Software Requirements	6
3.2 Tool/ Languages/ Platform	6
3.3 Functional Requirements	7
4. System Design	8
4.1 Greetings Page Widget	8
4.2 Login page Widget Tree	9
4 .3 Schema Design	10
5. IMPLEMENTATION DETAILS	11

5.1 main.dart	11
5.2 greetings.dart	11
5.3 login.dart	13
5.4 explore.dart	14
5.5 browse.dart	15
5.6 settings.dart	17
5.7 details.dart	19
5.8 color.dart	20
6. TESTING	22
6.1 Introduction	22
6.2 Levels Of Testing	22
6.2.1 Unit Testing	22
6.2.2 Integration Testing	23
6.2.3 System Testing	23
6.2.4 Validation Testing	23
6.2.5 Output Testing	23
6.2.6 User Validation Testing	24
7. DISCUSSION OF RESULTS	25
7.1 Home Page	25
7.2 Display Details Page	26
7.3 Exploration Page	27
8. CONCLUSION AND FUTURE	28
WORK 8.1 Conclusion	28
8.2 Future work	28
9. REFERNECES	29

LIST OF FIGURES

Figure. No.	Descriptions	Page
Figure. 4.1	Greetings Page Widget Tree	08
Figure. 4.2	Login Page Widget Tree	09
Figure. 4.3	Schema Design	10
Figure. 7.1	Browse Page	25
Figure. 7.2	Details Page	26
Figure 7.3	Settings Page	27

ABBREVIATIONS

Acronym	Description
ADO	Active X Data Object
SQL	Structured Query Language
MSSQL	Microsoft SQL Server
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
CLR	Common Language runtime
IE	Internet Explorer
VB	Visual Basics
ISO	International Organization of Standardization
ANSI	American National Standard Institutes

1.INTRODUCTION

1.1 Introduction to Flutter

[Flutter](#) is Google's Mobile SDK to build native iOS and Android, Desktop (Windows, Linux, macOS), Web apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. They are structural elements that ship with a bunch of material design-specific functionalities and new widgets can be composed out of existing ones too. The process of composing widgets together is called composition. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

1.2 History

Flutter launched as a project called Sky which at the beginning worked only on Android. Flutter's goal is enabling developers to compile for every platform using its own graphic layer rendered by the Skia engine. Here's a brief presentation of Flutter's relatively short history.

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, this allows you to create a native mobile application with only one code. It means that you can use one programming language and one codebase to create two different apps (IOS and Android).

The first version of Flutter was known by the codename "Sky" and ran on the [Android](#) operating system. It was unveiled at the 2015 [Dart](#) developer summit[6] with the stated intent of being able to [render](#) consistently at 120 [frames per second](#).[7] During the keynote of [Google Developer Days](#) in Shanghai in September 2018, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.[8]

On May 6, 2020, the Dart software development kit ([SDK](#)) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the [Metal API](#), improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with new CanvasKit renderer and web specific widgets, early-access desktop application support for [Windows](#), [macOS](#), and [Linux](#) and improved Add-to-App APIs.[9] This release included sound [null-safety](#), which caused many breaking changes and issues with many external packages, but the Flutter team included instructions to mitigate these changes as well.

On September 8th, 2021, the Dart SDK in version 2.14 and Flutter version 2.5 were released by Google. The update brought improvements to the Android Full-Screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, the newest lint conditions have been standardized and preset as the default conditions as well Dart for Apple Silicon is now stable.

1.1.1 Framework-Architecture

The major components of Flutter include:

- [Dart](#) platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter Development Tools (DevTools)

Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On [Windows](#), [macOS](#), and [Linux](#)[11] Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for [stateful](#) hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of [state](#).

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with ahead-of-time (AOT) compilation.

Flutter engine

Flutter's engine, written primarily in [C++](#), provides low-level [rendering](#) support using Google's [Skia](#) graphics library. Additionally, it interfaces with [platform-specific SDKs](#) such as those provided by [Android](#) and [iOS](#).^[10] The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

Foundation library

The Foundation library, written in [Dart](#), provides basic classes and functions that are used to construct applications using Flutter, such as [APIs](#) to communicate with the engine.

Design-specific widgets

The Flutter framework contains two sets of [widgets](#) that conform to specific design languages: [Material Design](#) widgets implement Google's [design language](#) of the same name, and Cupertino widgets implement Apple's [iOS Human interface guidelines](#).

2.LITERATURE SURVEY

2.1. FLUTTER PLANT APP

Species knowledge is essential for protecting biodiversity. The identification of plants by conventional keys is complex, time consuming, and due to the use of specific botanical terms frustrating for non-experts. This creates a hard to overcome hurdle for novices interested in acquiring species knowledge. Today, there is an increasing interest in automating the process of species identification. The availability and ubiquity of relevant technologies, such as, digital cameras and mobile devices, the remote access to databases, new techniques in image processing and pattern recognition let the idea of automated species identification become reality. This paper is the first systematic literature review with the aim of a thorough analysis and comparison of primary studies on computer vision approaches for plant species identification. We identified 120 peer-reviewed studies, selected through a multi-stage process, published in the last 10 years (2005--2015). After a careful analysis of these studies, we describe the applied methods categorized according to the studied plant organ, and the studied features, i.e., shape, texture, color, margin, and vein structure. Furthermore, we compare methods based on classification accuracy achieved on publicly available datasets. Our results are relevant to researches in ecology as well as computer vision for their ongoing research. The systematic and concise overview will also be helpful for beginners in those research fields, as they can use the comparable analyses of applied methods as a guide in this complex activity.

2.1.1 Introduction

Biodiversity is declining steadily throughout the world . The current rate of extinction is largely the result of direct and indirect human activities. Building accurate knowledge of the identity and the geographic distribution of plants is essential for future biodiversity conservation . Therefore, rapid and accurate plant identification is essential for effective study and management of biodiversity.

Our app helps people to know about different varieties of plants,seeds,fertilizers,flowers,sprayers,pots etc which is very useful as well as interesting for today's generation as we are very less aware with in this modern world.

3.ANALYSIS

3.1 Hardware and Software Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor : Pentium 4 Processor

Processor Speed : 2.4 GHz

RAM : 2 GB

Storage Space : 40 GB

The software requirements are very minimal and the program can be run on the machines with these requirements satisfied:

Editor : Visual Studio Code

Operating System : Windows/Mac OS

IDE : VS Code

Backend Tool : SQLite

3.2 Tools/ Languages/ Platform

Various tool used in making this project is given below:

Editor/IDE : Visual Studio Code

Operating System : Windows/Mac OS

Languages : Dart, Swift, SQLite

Backend Tool : SQLite

3.3 Functional Requirements

Flutter

Flutter is Google's Mobile SDK to build native iOS and Android apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

Compared to its contemporary technologies like React Native, Kotlin, and Java, Flutter is much better in regard to having a Single Codebase for Android and iOS, Reusable UI and Business Logic, high compatibility, performance, and productivity.

Dart

Dart is an open-source general-purpose programming language developed by Google. It supports application development in both client and server-side. But it is widely used for the development of android apps, iOS apps, IoT(Internet of Things), and web applications using the Flutter Framework.

Syntactically, Dart bears a strong resemblance to Java, C, and JavaScript. It is a dynamic object-oriented language with closure and lexical scope. The Dart language was released in 2011 but came into popularity after 2015 with Dart 2.0.

SQLite

SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. The SQLite database file format is cross-platform s

4.SYSTEM DESIGN

4.1 GREETINGS PAGE WIDGET TREE

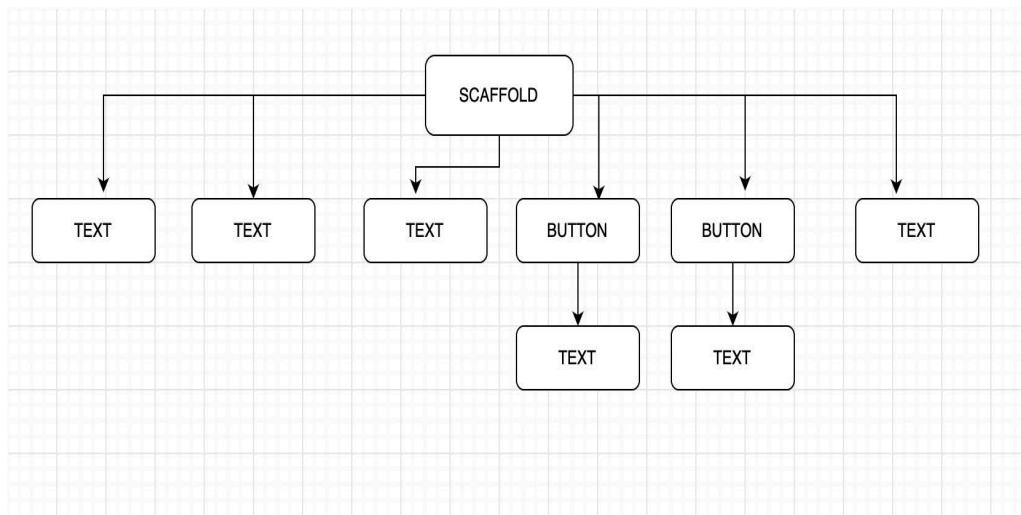


Figure 4.1:

4.2 LOGIN PAGE WIDGET PAGE

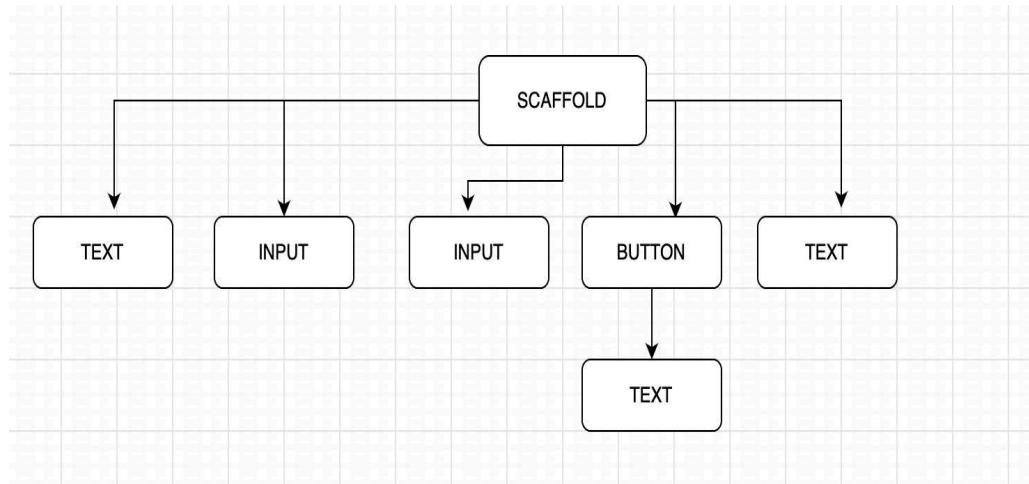
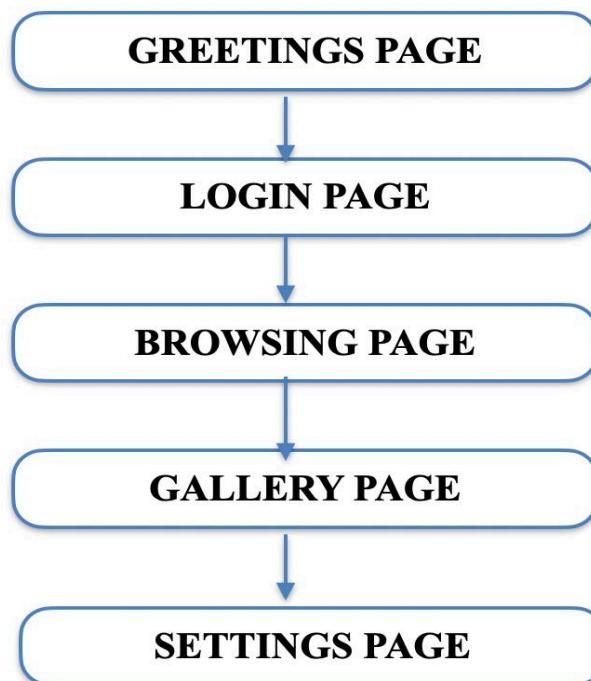


Figure 4.2:

4.3 SCHEMA DESIGN



5. IMPLEMENTATION

5.1 main.dart

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2
3 // SCREENS
4 import './screens/greeting_screen.dart';
5
6 void main() => runApp(MyApp());
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Flutter Demo',
13      theme: ThemeData(
14        primarySwatch: Colors.blue,
15      ), // ThemeData
16      home: GreetingScreen(),
17      debugShowCheckedModeBanner: false); // MaterialApp
18  }
19}
20
```

5.2 Greetings.dart

```
lib > screens > greeting_screen.dart > _GreetingScreenState > build
15
16 @override
17 Widget build(BuildContext context) {
18   return Scaffold(
19     resizeToAvoidBottomInset: false,
20     backgroundColor: Colors.white,
21     body: SingleChildScrollView(
22       child: Column(
23         mainAxisAlignment: MainAxisAlignment.center,
24         children: <Widget>[
25           SizedBox(
26             height: MediaQuery.of(context).size.height / 8,
27           ), // SizedBox
28           Row(
29             mainAxisAlignment: MainAxisAlignment.center,
30             children: <Widget>[
31               Text(
32                 "Your home. ",
33                 style: TextStyle(
34                   fontSize: 26.0,
35                   color: lightBlack,
36                   fontWeight: FontWeight.bold), // TextStyle
37               ), // Text
38               Text(
39                 "Greener",
40                 style: TextStyle(
41                   fontSize: 26.0,
42                   foreground: Paint()..shader = linearGradient,
43                   fontWeight: FontWeight.bold), // TextStyle
44               ), // Text
45             ], // <Widget>[]
46           ),
47           Text(
48             "Enjoy the experience",
49             style: TextStyle(color: lightGrey, fontSize: 18.0),
50           ), // Text
51           SizedBox(
52             height: 10.0,
53           ), // SizedBox
54           Image(
55             Image: AssetImage("assets/welcome_art.jpg"),
56           ), // Image
57           SizedBox(height: 20.0),
58           GestureDetector(
59             onTap: () {
60               Navigator.push(context,
```

```
creens > greeting_screen.dart > _GreetingScreenState > build
    ),
    ), // Row
    Text(
        "Enjoy the experience",
        style: TextStyle(color: lightGrey, fontSize: 18.0),
    ),
    ), // Text
    SizedBox(
        height: 10.0,
    ),
    ), // SizedBox
    Image(
        image: AssetImage("assets/welcome_art.jpg"),
    ),
    ), // Image
    SizedBox(height: 20.0),
    GestureDetector(
        onTap: () {
            Navigator.push(context,
                MaterialPageRoute(builder: (context) => LoginScreen()));
        },
        child: Container(
            width: MediaQuery.of(context).size.width - 75,
            height: 48,
            decoration: BoxDecoration(
                gradient: LinearGradient(
                    colors: [primaryColor1, secondaryColorGreen],
                    begin: Alignment.topLeft,
                    end: Alignment.bottomRight), // LinearGradient
                borderRadius: BorderRadius.circular(6.0),
                boxShadow: [
                    BoxShadow(
                        color: Color(0xFF9DA3B4).withOpacity(0.1),
                        blurRadius: 65.0,
                        offset: Offset(0.0, 15.0)) // BoxShadow
                ],
            ),
            child: Center(
                child: Text(
                    "Login",
                    style: TextStyle(
                        color: Color(0xFFFFFBFB),
                        fontSize: 14.0,
                        fontWeight: FontWeight.w500), // TextStyle
                ),
            ),
        ),
    ),
),
), // Container
),
), // GestureDetector
SizedBox(
    height: 20.0,
```

```
        style: TextStyle(
            color: Colors.black,
            fontSize: 14.0,
            fontWeight: FontWeight.w500), // TextStyle
        ), // Text
    ), // Center
), // Container
SizedBox(
    height: 20.0,
), // SizedBox
Text(
    "Terms of service",
    style: TextStyle(color: blueGrey, fontSize: 12),
) // Text
], // <Widget>[]
), // Column // SingleChildScrollView
); // Scaffold
}
}
```

5.3 Login_screen.dart

```
lib > screens > login_screen.dart > _LoginScreenState > build
  1 import 'package:flutter/material.dart';
  2 import './colors.dart';
  3 import './browse_screens.dart';
  4
  5 class LoginScreen extends StatefulWidget {
  6   LoginScreen({Key key}) : super(key: key);
  7
  8   _LoginScreenState createState() => _LoginScreenState();
  9 }
10
11 class _LoginScreenState extends State<LoginScreen> {
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       backgroundColor: Colors.white,
16       appBar: PreferredSize(
17         preferredSize: Size.fromHeight(140.0),
18         child: AppBar(
19           centerTitle: false,
20           flexibleSpace: Container(
21             margin: EdgeInsets.symmetric(vertical: 35.0, horizontal: 20.0),
22             child: Align(
23               alignment: Alignment.bottomLeft,
24               child: Text(
25                 "Login",
26                 style: TextStyle(
27                   color: Color(0xFF323643),
28                   fontSize: 26.0,
29                   fontWeight: FontWeight.w700), // TextStyle
30               ), // Text
31             ), // Align
32           ), // Container
33           elevation: 0.0,
34           backgroundColor: Colors.white,
35           iconTheme: IconThemeData(color: Color(0xFFC5CCD6)),
36         ), // AppBar
37       ), // PreferredSize
38       body: SingleChildScrollView(
39         child: Container(
40           width: MediaQuery.of(context).size.width - 40,
41           margin: EdgeInsets.symmetric(horizontal: 20.0),
42           child: Center(
43             child: Column(
44               mainAxisAlignment: MainAxisAlignment.center,
45               crossAxisAlignment: CrossAxisAlignment.center,
46               children: [
47                 Text("Email"),
48                 TextField(
49                   decoration: InputDecoration(
50                     hintText: "Email Address",
51                     border: OutlineInputBorder(),
52                   ),
53                 ),
54               ],
55             ),
56           ),
57         ),
58       ),
59     );
60   }
61 }
```

5.4 explore.dart

```
lib > screens > explore_screen.dart > ...
1 import 'package:flutter/material.dart';
2 import './colors.dart';
3
4 class ExploreScreen extends StatefulWidget {
5   ExploreScreen({Key key}) : super(key: key);
6
7   _ExploreScreenState createState() => _ExploreScreenState();
8 }
9
10 class _ExploreScreenState extends State<ExploreScreen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       backgroundColor: Colors.white,
15       appBar: PreferredSize(
16         preferredSize: Size.fromHeight(130.0),
17         child: AppBar(
18           centerTitle: false,
19           flexibleSpace: Container(
20             margin: EdgeInsets.only(left: 20.0, bottom: 30.0, right: 20.0),
21             child: Align(
22               alignment: Alignment.bottomLeft,
23               child: Row(
24                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
25                 children: <Widget>[
26                   Text(
27                     "Browse",
28                     style: TextStyle(
29                       color: Color(0xFF323643),
30                       fontSize: 26.0,
31                       fontWeight: FontWeight.w700), // TextStyle
32                   ), // Text
33                   Container(
34                     width: 160.0,
35                     height: 40,
36                     padding: EdgeInsets.all(5.0),
37                     decoration: BoxDecoration(
38                       color: Color(0xFF8E8E93).withOpacity(0.06),
39                       borderRadius: BorderRadius.circular(6.0)), // BoxDecoration
40                     child: TextField(
41                       cursorColor: Color(0xFF8E8E93).withOpacity(0.06),
42                       decoration: InputDecoration(
43                         labelText: "Search",
44                         labelStyle: TextStyle(
45                           color: Color(0xFFCC5CD6), fontSize: 14), // TextStyle
46                         suffixIcon: Icon(
```

```
> screens > 🌱 explore_screen.dart > ...
1           |   Image(image: AssetImage("assets/images/explore4.png")),
2           |   Image(image: AssetImage("assets/images/explore5.png")),
3           |   ], // <Widget>[]
4           |   ), // Row
5           |   SizedBox(height: 10.0),
6           |   Image(
7           |     image: AssetImage("assets/images/explore6.png"),
8           |   ), // Image
9           |   ], // <Widget>[]
10          |   ), // Column
11          |   ), // Center
12          |   ), // SingleChildScrollView
13          |   ), // Container
14          |   Positioned(
15          |     bottom: 0.0,
16          |     left: 0.0,
17          |     width: MediaQuery.of(context).size.width,
18          |     height: 100.0,
19          |     child: Container(
20          |       decoration: BoxDecoration(
21          |         gradient: LinearGradient(colors: [
22          |           Colors.white.withOpacity(0.01),
23          |           Colors.white.withOpacity(1)
24          |         ], begin: Alignment.topCenter, end: Alignment.bottomCenter)), // LinearGradient // BoxDecoration
25          |       child: Center(
26          |         child: Container(
27          |           width: 150.0,
28          |           height: 40,
29          |           decoration: BoxDecoration(
30          |             gradient: LinearGradient(
31          |               colors: [primaryColor1, secondaryColorGreen],
32          |               begin: Alignment.topLeft,
33          |               end: Alignment.bottomRight), // LinearGradient
34          |             borderRadius: BorderRadius.circular(6.0),
35          |             boxShadow: [
36          |               BoxShadow(
37          |                 color: Color(0xFF9DA3B4).withOpacity(0.1),
38          |                 blurRadius: 65.0,
39          |                 offset: Offset(0.0, 15.0)) // BoxShadow
40          |             ], // BoxDecoration
41          |             child: Center(
42          |               child: Text(
43          |                 "Filters",
44          |                 style: TextStyle(
45          |                   color: Color(0xFFFFFBFB),
46          |                 ),
47          |               ),
48          |             ),
49          |           ),
50          |         ),
51          |       ),
52          |     ),
53          |   ),
54        ),
55      ),
56    ),
57  ),
58),
59),
60),
61),
62),
63),
64),
65),
66),
67),
68),
69),
69),
70),
71),
72),
73),
74),
75),
76),
77),
78),
79),
79),
80),
81),
82),
83),
84),
85),
86),
87),
88),
89),
89),
90),
91),
92),
93),
94),
95),
96),
97),
98),
99),
99),
100),
101),
102),
103),
104),
105),
106),
107),
108),
109),
109),
110),
111),
112),
113),
114),
115),
116),
117),
118),
119),
119),
120),
121),
122),
123),
124),
125),
126),
127),
128),
129),
129),
130),
131),
132),
133),
134),
135),
136),
137),
138),
139),
139),
140),
141),
142),
143),
144),
145),
146),
147),
148),
149),
149),
150),
151),
152),
153),
154),
155),
156),
157),
158),
159),
159),
160),
161),
162),
163),
164),
165),
166),
167),
168),
169),
169),
170),
171),
172),
173),
174),
175),
176),
177),
178),
179),
179),
180),
181),
182),
183),
184),
185),
186),
187),
188),
189),
189),
190),
191),
192),
193),
194),
195),
196),
197),
198),
199),
199),
200),
201),
202),
203),
204),
205),
206),
207),
208),
209),
209),
210),
211),
212),
213),
214),
215),
216),
217),
218),
219),
219),
220),
221),
222),
223),
224),
225),
226),
227),
228),
229),
229),
230),
231),
232),
233),
234),
235),
236),
237),
238),
239),
239),
240),
241),
242),
243),
244),
245),
246),
247),
248),
249),
249),
250),
251),
252),
253),
254),
255),
256),
257),
258),
259),
259),
260),
261),
262),
263),
264),
265),
266),
267),
268),
269),
269),
270),
271),
272),
273),
274),
275),
276),
277),
278),
278),
279),
279),
280),
281),
282),
283),
284),
285),
286),
287),
288),
289),
289),
290),
291),
292),
293),
294),
295),
296),
297),
298),
299),
299),
300),
301),
302),
303),
304),
305),
306),
307),
308),
309),
309),
310),
311),
312),
313),
314),
315),
316),
317),
318),
319),
319),
320),
321),
322),
323),
324),
325),
326),
327),
328),
329),
329),
330),
331),
332),
333),
334),
335),
336),
337),
338),
339),
339),
340),
341),
342),
343),
344),
345),
346),
347),
348),
349),
349),
350),
351),
352),
353),
354),
355),
356),
357),
358),
359),
359),
360),
361),
362),
363),
364),
365),
366),
367),
368),
369),
369),
370),
371),
372),
373),
374),
375),
376),
377),
378),
378),
379),
379),
380),
381),
382),
383),
384),
385),
386),
387),
388),
388),
389),
389),
390),
391),
392),
393),
394),
395),
396),
397),
398),
398),
399),
399),
400),
401),
402),
403),
404),
405),
406),
407),
408),
409),
409),
410),
411),
412),
413),
414),
415),
416),
417),
418),
419),
419),
420),
421),
422),
423),
424),
425),
426),
427),
428),
429),
429),
430),
431),
432),
433),
434),
435),
436),
437),
438),
439),
439),
440),
441),
442),
443),
444),
445),
446),
447),
448),
449),
449),
450),
451),
452),
453),
454),
455),
456),
457),
458),
459),
459),
460),
461),
462),
463),
464),
465),
466),
467),
468),
469),
469),
470),
471),
472),
473),
474),
475),
476),
477),
478),
478),
479),
479),
480),
481),
482),
483),
484),
485),
486),
487),
488),
488),
489),
489),
490),
491),
492),
493),
494),
495),
496),
497),
497),
498),
498),
499),
499),
500),
501),
502),
503),
504),
505),
506),
507),
508),
509),
509),
510),
511),
512),
513),
514),
515),
516),
517),
518),
519),
519),
520),
521),
522),
523),
524),
525),
526),
527),
528),
529),
529),
530),
531),
532),
533),
534),
535),
536),
537),
538),
539),
539),
540),
541),
542),
543),
544),
545),
546),
547),
548),
549),
549),
550),
551),
552),
553),
554),
555),
556),
557),
558),
559),
559),
560),
561),
562),
563),
564),
565),
566),
567),
568),
569),
569),
570),
571),
572),
573),
574),
575),
576),
577),
578),
578),
579),
579),
580),
581),
582),
583),
584),
585),
586),
587),
588),
588),
589),
589),
590),
591),
592),
593),
594),
595),
596),
597),
597),
598),
598),
599),
599),
600),
601),
602),
603),
604),
605),
606),
607),
608),
609),
609),
610),
611),
612),
613),
614),
615),
616),
617),
618),
619),
619),
620),
621),
622),
623),
624),
625),
626),
627),
628),
629),
629),
630),
631),
632),
633),
634),
635),
636),
637),
638),
639),
639),
640),
641),
642),
643),
644),
645),
646),
647),
648),
649),
649),
650),
651),
652),
653),
654),
655),
656),
657),
658),
659),
659),
660),
661),
662),
663),
664),
665),
666),
667),
668),
669),
669),
670),
671),
672),
673),
674),
675),
676),
677),
678),
678),
679),
679),
680),
681),
682),
683),
684),
685),
686),
687),
688),
688),
689),
689),
690),
691),
692),
693),
694),
695),
696),
697),
697),
698),
698),
699),
699),
700),
701),
702),
703),
704),
705),
706),
707),
708),
709),
709),
710),
711),
712),
713),
714),
715),
716),
717),
718),
719),
719),
720),
721),
722),
723),
724),
725),
726),
727),
728),
729),
729),
730),
731),
732),
733),
734),
735),
736),
737),
738),
739),
739),
740),
741),
742),
743),
744),
745),
746),
747),
748),
749),
749),
750),
751),
752),
753),
754),
755),
756),
757),
758),
759),
759),
760),
761),
762),
763),
764),
765),
766),
767),
768),
769),
769),
770),
771),
772),
773),
774),
775),
776),
777),
778),
778),
779),
779),
780),
781),
782),
783),
784),
785),
786),
787),
788),
788),
789),
789),
790),
791),
792),
793),
794),
795),
796),
797),
797),
798),
798),
799),
799),
800),
801),
802),
803),
804),
805),
806),
807),
808),
809),
809),
810),
811),
812),
813),
814),
815),
816),
817),
818),
819),
819),
820),
821),
822),
823),
824),
825),
826),
827),
828),
829),
829),
830),
831),
832),
833),
834),
835),
836),
837),
838),
839),
839),
840),
841),
842),
843),
844),
845),
846),
847),
848),
849),
849),
850),
851),
852),
853),
854),
855),
856),
857),
858),
859),
859),
860),
861),
862),
863),
864),
865),
866),
867),
868),
869),
869),
870),
871),
872),
873),
874),
875),
876),
877),
878),
878),
879),
879),
880),
881),
882),
883),
884),
885),
886),
887),
888),
888),
889),
889),
890),
891),
892),
893),
894),
895),
896),
897),
897),
898),
898),
899),
899),
900),
901),
902),
903),
904),
905),
906),
907),
908),
909),
909),
910),
911),
912),
913),
914),
915),
916),
917),
918),
919),
919),
920),
921),
922),
923),
924),
925),
926),
927),
928),
929),
929),
930),
931),
932),
933),
934),
935),
936),
937),
938),
939),
939),
940),
941),
942),
943),
944),
945),
946),
947),
948),
949),
949),
950),
951),
952),
953),
954),
955),
956),
957),
958),
959),
959),
960),
961),
962),
963),
964),
965),
966),
967),
968),
969),
969),
970),
971),
972),
973),
974),
975),
976),
977),
978),
978),
979),
979),
980),
981),
982),
983),
984),
985),
986),
987),
988),
988),
989),
989),
990),
991),
992),
993),
994),
995),
995),
996),
996),
997),
997),
998),
998),
999),
999),
1000),
1001),
1002),
1003),
1004),
1005),
1006),
1007),
1008),
1009),
1009),
1010),
1011),
1012),
1013),
1014),
1015),
1016),
1017),
1018),
1019),
1019),
1020),
1021),
1022),
1023),
1024),
1025),
1026),
1027),
1028),
1029),
1029),
1030),
1031),
1032),
1033),
1034),
1035),
1036),
1037),
1038),
1039),
1039),
1040),
1041),
1042),
1043),
1044),
1045),
1046),
1047),
1048),
1049),
1049),
1050),
1051),
1052),
1053),
1054),
1055),
1056),
1057),
1058),
1059),
1059),
1060),
1061),
1062),
1063),
1064),
1065),
1066),
1067),
1068),
1069),
1069),
1070),
1071),
1072),
1073),
1074),
1075),
1076),
1077),
1078),
1078),
1079),
1079),
1080),
1081),
1082),
1083),
1084),
1085),
1086),
1087),
1088),
1088),
1089),
1089),
1090),
1091),
1092),
1093),
1094),
1095),
1096),
1097),
1097),
1098),
1098),
1099),
1099),
1100),
1101),
1102),
1103),
1104),
1105),
1106),
1107),
1108),
1109),
1109),
1110),
1111),
1112),
1113),
1114),
1115),
1116),
1117),
1118),
1119),
1119),
1120),
1121),
1122),
1123),
1124),
1125),
1126),
1127),
1128),
1129),
1129),
1130),
1131),
1132),
1133),
1134),
1135),
1136),
1137),
1138),
1139),
1139),
1140),
1141),
1142),
1143),
1144),
1145),
1146),
1147),
1148),
1149),
1149),
1150),
1151),
1152),
1153),
1154),
1155),
1156),
1157),
1158),
1159),
1159),
1160),
1161),
1162),
1163),
1164),
1165),
1166),
1167),
1168),
1169),
1169),
1170),
1171),
1172),
1173),
1174),
1175),
1176),
1177),
1178),
1178),
1179),
1179),
1180),
1181),
1182),
1183),
1184),
1185),
1186),
1187),
1187),
1188),
1188),
1189),
1189),
1190),
1191),
1192),
1193),
1194),
1195),
1196),
1197),
1197),
1198),
1198),
1199),
1199),
1200),
1201),
1202),
1203),
1204),
1205),
1206),
1207),
1208),
1209),
1209),
1210),
1211),
1212),
1213),
1214),
1215),
1216),
1217),
1218),
1219),
1219),
1220),
1221),
1222),
1223),
1224),
1225),
1226),
1227),
1228),
1229),
1229),
1230),
1231),
1232),
1233),
1234),
1235),
1236),
1237),
1238),
1239),
1239),
1240),
1241),
1242),
1243),
1244),
1245),
1246),
1247),
1248),
1249),
1249),
1250),
1251),
1252),
1253),
1254),
1255),
1256),
1257),
1258),
1259),
1259),
1260),
1261),
1262),
1263),
1264),
1265),
1266),
1267),
1268),
1269),
1269),
1270),
1271),
1272),
1273),
1274),
1275),
1276),
1277),
1278),
1278),
1279),
1279),
1280),
1281),
1282),
1283),
1284),
1285),
1286),
1287),
1287),
1288),
1288),
1289),
1289),
1290),
1291),
1292),
1293),
1294),
1295),
1296),
1297),
1297),
1298),
1298),
1299),
1299),
1300),
1301),
1302),
1303),
1304),
1305),
1306),
1307),
1308),
1309),
1309),
1310),
1311),
1312),
1313),
1314),
1315),
1316),
1317),
1318),
1319),
1319),
1320),
1321),
1322),
1323),
1324),
1325),
1326),
1327),
1328),
1329),
1329),
1330),
1331),
1332),
1333),
1334),
1335),
1336),
1337),
1338),
1339),
1339),
1340),
1341),
1342),
1343),
1344),
1345),
1346),
1347),
1348),
1349),
1349),
1350),
1351),
1352),
1353),
1354),
1355),
1356),
1357),
1358),
1359),
1359),
1360),
1361),
1362),
1363),
1364),
1365),
1366),
1367),
1368),
1369),
1369),
1370),
1371),
1372),
1373),
1374),
1375),
1376),
1377),
1378),
1378),
1379),
1379),
1380),
1381),
1382),
1383),
1384),
1385),
1386),
1387),
1387),
1388),
1388),
1389),
1389),
1390),
1391),
1392),
1393),
1394),
1395),
1396),
1397),
1397),
1398),
1398),
1399),
1399),
1400),
1401),
1402),
1403),
1404),
1405),
1406),
1407),
1408),
1409),
1409),
1410),
1411),
1412),
1413),
1414),
1415),
1416),
1417),
1418),
1419),
1419),
1420),
1421),
1422),
1423),
1424),
1425),
1426),
1427),
1428),
1429),
1429),
1430),
1431),
1432),
1433),
1434),
1435),
1436),
1437),
1438),
1439),
1439),
1440),
1441),
1442),
1443),
1444),
1445),
1446),
1447),
1448),
1449),
1449),
1450),
1451),
1452),
1453),
1454),
1455),
1456),
1457),
1458),
1459),
1459),
1460),
1461),
1462),
1463),
1464),
1465),
1466),
1467),
1468),
1469),
1469),
1470),
1471),
1472),
1473),
1474),
1475),
1476),
1477),
1478),
1478),
1479),
1479),
1480),
1481),
1482),
1483),
1484),
1485),
1486),
1487),
1487),
1488),
1488),
1489),
1489),
1490),
1491),
1492),
1493),
1494),
1495),
1496),
1497),
1497),
1498),
1498),
1499),
1499),
1500),
1501),
1502),
1503),
1504),
1505),
1506),
1507),
1508),
1509),
1509),
1510),
1511),
1512),
1513),
1514),
1515),
1516),
1517),
1518),
1519),
1519),
1520),
1521),
1522),
1523),
1524),
1525),
1526),
1527),
1528),
1529),
1529),
1530),
1531),
1532),
1533),
1534),
1535),
1536),
1537),
1538),
1539),
1539),
1540),
1541),
1542),
1543),
1544),
1545),
1546),
1547),
1548),
1549),
1549),
1550),
1551),
1552),
1553),
1554),
1555),
1556),
1557),
1558),
1559),
1559),
1560),
1561),
1562),
1563),
1564),
1565),
1566),
1567),
1568),
1569),
1569),
1570),
1571),
1572),
1573),
1574),
1575),
1576),
1577),
1578),
1578),
1579),
1579),
1580),
1581),
1582),
1583),
1584),
1585),
1586),
1587),
1587),
1588),
1588),
1589),
1589),
1590),
1591),
1592),
1593),
1594),
1595),
1596),
1597),
1597),
1598),
1598),
1599),
1599),
1600),
1601),
1602),
1603),
1604),
1605),
1606),
1607),
1608),
1609),
1609),
1610),
1611),
1612),
1613),
1614),
1615),
1616),
1617),
1618),
1619),
1619),
1620),
1621),
1622),
1623),
1624),
1625),
1626),
1627),
1628),
1629),
1629),
1630),
1631),
1632),
1633),
1634),
1635),
1636),
1637),
1638),
1639),
1639),
1640),
1641),
1642),
1643),
1644),
1645),
1646),
1647),
1648),
1649),
1649),
1650),
1651),
1652),
1653),
1654),
1655),
1656),
1657),
1658),
1659),
1659),
1660),
1661),
1662),
1663),
1664),
1665),
1666),
1667),
1668),
1669),
1669),
1670),
1671),
1672),
1673),
1674),
1675),
1676),
1677),
1678),
1678),
1679),
1679),
1680),
1681),
1682),
1683),
1684),
1685),
1686),
1687),
1687),
1688),
1688),
1689),
1689),
1690),
1691),
1692),
1693),
1694),
1695),
1696),
1697),
1697),
1698),
1698),
1699),
1699),
1700),
1701),
1702),
1703),
1704),
1705),
1706),
1707),
1708),
1709),
1709),
1710),
1711),
1712),
1713),
1714),
1715),
1716),
1717),
1718),
1719),
1719),
1720),
1721),
1722),
1723),
1724),
1725),
1726),
1727),
1728),
1729),
1729),
1730),
1731),
1732),
1733),
1734),
1735),
1736),
1737),
1738),
1739),
1739),
1740),
1741),
1742),
1743),
1744),
1745),
1746),
1747),
1748),
1749),
1749),
1750),
1751),
1752),
1753),
1754),
1755),
1756),
1757),
1758),
1759),
1759),
1760),
1761),
1762),
1763),
1764),
1765),
1766),
1767),
1768),
1769),
1769),
1770),
1771),
1772),
1773),
1774),
1775),
1776),
1777),
1778),
1778),
1779),
1779),
1780),
1781),
1782),
1783),
1784),
1785),
1786),
1787),
1787),
1788),
1788),
1789),
1789),
1790),
1791),
1792),
1793),
1794),
1795),
1796),
1797),
1797),
1798),
1798),
1799),
1799),
1800),
1801),
1802),
1803),
1804),
1805),
1806),
1807),
1808),
1809),
1809),
1810),
1811),
1812),
1813),
1814),
1815),
1816),
1817),
1818),
1819),
1819),
1820),
1821),
1822),
1823),
1824),
1825),
1826),
1827),
1828),
1829),
1829),
1830),
1831),
1832),
1833),
1834),
1835),
1836),
1837),
1838),
1839),
1839),
1840),
1841),
1842),
1843),
1844),
1845),
1846),
1847),
1848),
1849),
1849),
1850),
1851),
1852),
1853),
1854),
1855),
1856),
1857),
1858),
1859),
1859),
1860),
1861),
1862),
1863),
1864),
1865),
1866),
1867),
1868),
1869),
1869),
1870),
1871),
1872),
1873),
1874),
1875),
1876),
1877),
1878),
1878),
1879),
1879),
1880),
1881),
1882),
1883),
1884),
1885),
1886),
1887),
1887),
1888),
1888),
1889),
1889),
1890),
1891),
1892),
1893),
1894),
1895),
1896),
1897),
1897),
1898),
1898),
1899),
1899),
1900),
1901),
1902),
1903),
1904),
1905),
1906),
1907),
1908),
1909),
1909),
1910),
1911),
1912),
1913),
1914),
1915),
1916),
1917),
1918),
1919),
1919),
1920),
1921),
1922),
1923),
1924),
1925),
1926),
1927),
1928),
1929),
1929),
1930),
1931),
1932),
1933),
1934),
1935),
1936),
1937),
1938),
1939),
1939),
1940),
1941),
1942),
1943),
1944),
1945),
1946),
1947),
1948),
1949),
1949),
1950),
1951),
1952),
1953),
1954),
1955),
1956),
1957),
1958),
1959),
1959),
1960),
1961),
1962),
1963),
1964),
1965),
1966),
1967),
1968),
1969),
1969),
1970),
1971),
1972),
1973),
1974),
1975),
1976),
1977),
1978),
1978),
1979),
1979),
1980),
1981),
1982),
1983),
1984),
1985),
1986),
1987),
1988),
1989),
1989),
1990),
1991),
1992),
1993),
1994),
1995),
1996),
1997),
1997),
1998),
1998),
1999),
1999),
2000),
2001),
2002),
2003),
2004),
2005),
2006),
2007),
2008),
2009),
2009),
2010),
2011),
2012),
2013),
2014),
2015),
2016),
2017),
2018),
2019),
2019),
2020),
2021),
2022),
2023),
2024),
2025),
2026),
2027),
2028),
2029),
2029),
2030),
2031),
2032),
2033),
2034),
2035),
2036),
2037),
2038),
2039),
2039),
2040),
2041),
2042),
2043),
2044),
2045),
2046),
2047),
2048),
2049),
2049),
2050),
2051),
2052),
2053),
2054),
2055),
2056),
2057),
2058),
2059),
2059),
2060),
2061),
2062),
2063),
2064),
2065),
2066),
2067),
2068),
2069),
2069),
2070),
2071),
2072),
2073),
2074),
2075),
2076),
2077),
2078),
2078),
2079),
2079),
2080),
2081),
2082),
2083),
2084),
2085),
2086),
2087),
2088),
2089),
2089),
2090),
2091),
2092),
2093),
2094),
2
```

```
lib > screens > browse_screens.dart > products
  1 import 'package:flutter/material.dart';
  2 import './colors.dart';
  3 import './detail_screen.dart';
  4 import './settings_screen.dart';
  5 
  6 const products = [
  7   {'image': 'assets/plants.jpg', 'count': '147', 'name': 'Plants'},
  8   {'image': 'assets/seeds.jpg', 'count': '16', 'name': 'Seeds'},
  9   {'image': 'assets/flowers.jpg', 'count': '68', 'name': 'Flowers'},
 10   {'image': 'assets/sprayers.jpg', 'count': '17', 'name': 'Sprayers'},
 11   {'image': 'assets/pots.jpg', 'count': '47', 'name': 'Pots'},
 12   {'image': 'assets/fertilizers.jpg', 'count': '9', 'name': 'Fertilizers'},
 13 ];
 14 
 15 const inspirations = [
 16   {'image': 'assets/plants.jpg', 'count': '147', 'name': 'Plants'},
 17   {'image': 'assets/flowers.jpg', 'count': '68', 'name': 'Flowers'},
 18 ];
 19 
 20 const shop = [
 21   {'image': 'assets/seeds.jpg', 'count': '16', 'name': 'Seeds'},
 22   {'image': 'assets/sprayers.jpg', 'count': '17', 'name': 'Sprayers'},
 23   {'image': 'assets/pots.jpg', 'count': '47', 'name': 'Pots'},
 24   {'image': 'assets/fertilizers.jpg', 'count': '9', 'name': 'Fertilizers'},
 25 ];
 26 
 27 class BrowseScreen extends StatefulWidget {
 28   const BrowseScreen({Key key}) : super(key: key);
 29 
 30   @override
 31   _BrowseScreenState createState() => _BrowseScreenState();
 32 }
 33 
 34 class _BrowseScreenState extends State<BrowseScreen>
 35   with SingleTickerProviderStateMixin {
 36   TabController controller;
 37 
 38   final Shader linearGradient = LinearGradient(
 39     colors: <Color>[primaryColor1, secondaryColorGreen],
 40   ).createShader(new Rect.fromLTWH(0.0, 0.0, 200.0, 70.0)); // LinearGradient
 41 
 42   @override
 43   void initState() {
 44     super.initState();
 45     controller = TabController(vsync: this, length: 3);
 46   }
}
```

```

  > screens > browse_screens.dart > products
      Expanded(
    child: GridView.builder(
      padding: EdgeInsets.all(20.0),
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
        crossAxisSpacing: 30.0,
        mainAxisSpacing: 15.0), // SliverGridDelegateWithFixedCrossAxisCount
      itemCount: shop.length,
      itemBuilder: (BuildContext context, int index) {
        return Container(
          width: 150.0,
          height: 150.0,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(6.0),
            boxShadow: [
              BoxShadow(
                blurRadius: 55,
                offset: Offset(0.0, 15.0),
                color: Color(0xFF9DA3B4).withOpacity(0.16)) // BoxShadow
            ],
            color: Colors.white,
          ), // BoxDecoration
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                Image(
                  image: AssetImage(shop[index]['image']),
                  width: 60.0,
                  height: 60.0,
                ), // Image
                Text(
                  shop[index]['name'],
                  style: TextStyle(
                    color: Color(0xFF444444),
                    fontSize: 16,
                    fontWeight: FontWeight.w600), // TextStyle
                ), // Text
                Text(
                  "${shop[index]['count']} products",
                  style: TextStyle(
                    color: Color(0xFFC5CCD6),
                    fontSize: 14,
                    fontWeight: FontWeight.normal), // TextStyle
                ), // Text
              ], // <Widget>[]
            ),
          ),
        );
      }
    ),
  ),

```

5.6 settings.dart

```

import 'package:flutter/material.dart';
import 'package:get/get_state_manager/get_state_manager.dart';
import 'package:plant_app/controllers/question_controller.dart'; import 'package:flutter_svg/svg.dart';

import '../../../../../constants.dart';

class ProgressBar extends StatelessWidget {
  Key key,
}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      width: double.infinity,

```

```
class SettingsScreen extends StatefulWidget {
  SettingsScreen({Key key}) : super(key: key);

  _SettingsScreenState createState() => _SettingsScreenState();
}

class _SettingsScreenState extends State<SettingsScreen> {
  final Shader linearGradient = LinearGradient(
    colors: <Color>[primaryColor1, secondaryColorGreen],
  ).createShader(new Rect.fromLTWH(0.0, 0.0, 200.0, 70.0)); // LinearGradient

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: PreferredSize(
        preferredSize: Size.fromHeight(120.0),
        child: AppBar(
          centerTitle: false,
          flexibleSpace: Container(
            margin: EdgeInsets.only(left: 20.0, bottom: 20.0, right: 20.0),
            child: Align(
              alignment: Alignment.bottomLeft,
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                  Text(
                    "Settings",
                    style: TextStyle(
                      color: Color(0xFF323643),
                      fontSize: 26.0,
                      fontWeight: FontWeight.w700), // TextStyle
                ), // Text
                CircleAvatar(
                  backgroundColor: Colors.black,
                  backgroundImage: AssetImage("assets/user.jpg"),
                ) // CircleAvatar
              ], // <Widget>[]
            )), // Row // Align
        ), // Container
        elevation: 0.0,
        backgroundColor: Colors.white,
        iconTheme: IconThemeData(color: Color(0xFFECECCB)),
      ),
    );
  }
}
```

5.7 details.dart

```

lib> screens> detail_screen.dart> ...
1 import 'package:flutter/material.dart';
2 <import './explore_screen.dart';
3
4 class DetailScreen extends StatefulWidget {
5   DetailScreen({Key key}) : super(key: key);
6
7   _DetailScreenState createState() => _DetailScreenState();
8 }
9
10 class _DetailScreenState extends State<DetailScreen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         actions: <Widget>[
16           IconButton(
17             icon: Icon(Icons.more_horiz),
18             onPressed: () {},
19           ) // IconButton
20         ], // <Widget>[]
21         backgroundColor: Colors.white,
22         elevation: 0.0,
23         iconTheme: IconThemeData(color: Color(0xFFC5CCD6)),
24       ), // AppBar
25       body: SingleChildScrollView(
26         scrollDirection: Axis.vertical,
27         child: Column(
28           crossAxisAlignment: CrossAxisAlignment.start,
29           children: <Widget>[
30             Image(
31               image: AssetImage("assets/images/1.png"),
32             ), // Image
33             SizedBox(
34               height: 25.0,
35             ), // SizedBox
36             Container(
37               margin: EdgeInsets.only(left: 20.0, right: 20.0, bottom: 40.0),
38               child: Column(
39                 crossAxisAlignment: CrossAxisAlignment.start,
40                 children: <Widget>[
41                   Container(
42                     width: MediaQuery.of(context).size.width - 100,
43                     child: Text(
44                       "16 Best Plants That Thrive In Your Bedroom",
45                       style: TextStyle(
46                         color: Color(0xFF4444AA)
47                     )
48                   )
49                 ],
50               ),
51             )
52           ],
53         ),
54       ),
55     );
56   }
57 }

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Flutter: 2.5.3 Pixel_3a_API_31_arm64-v8a (android-arm64 emulator) ⚡ Prettier ⌂

```
    ),  
    ],  
    ),  
    );  
}  
}
```

5.8 color.dart

```
lib > screens > 🎨 colors.dart > ...  
1 import 'package:flutter/material.dart';  
2  
3 const primaryColor1 = Color(0xFF0AC4BA);  
4 const secondaryColorGreen = Color(0xFF2BDA8E);  
5 const lightBlack = Color(0xFF323643);  
6 const lightGrey = Color(0xFFC5CCD6);  
7 const blueGrey = Color(0xFF9DA3B4);  
8
```



6.TESTING

6.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

A software configuration that includes a software requirement specification, a design specification and source code.

A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

6.2 Levels of Testing

6.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working correctly or not. In this part the main intention is to check each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program is taking the inputs in the correct format or not.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

6.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs.

6.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

6.2.4 Validation Testing

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

6.1.1 Output Testing

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

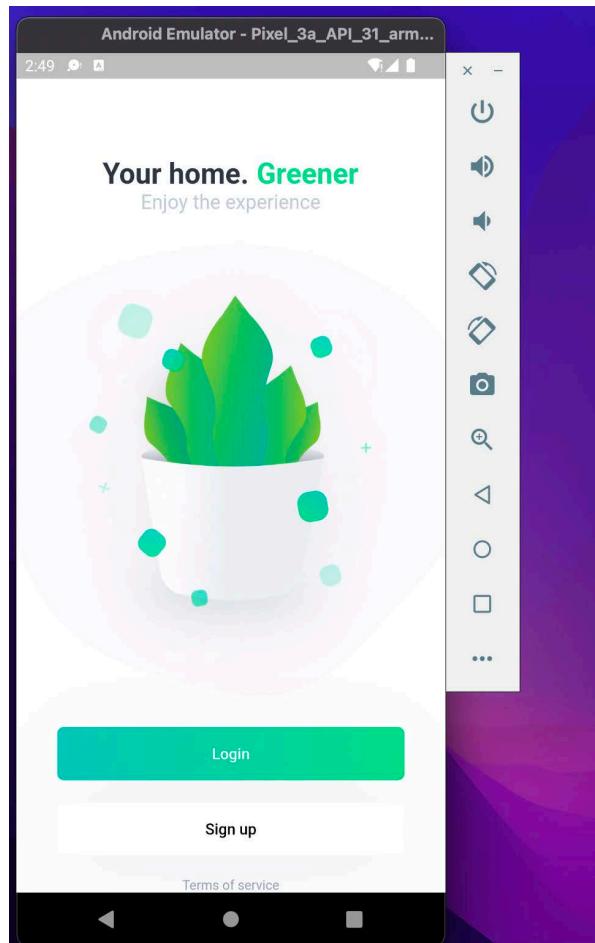
6.1.1 User Acceptance Testing

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment.

User Acceptance Testing is done in the final phase of testing.

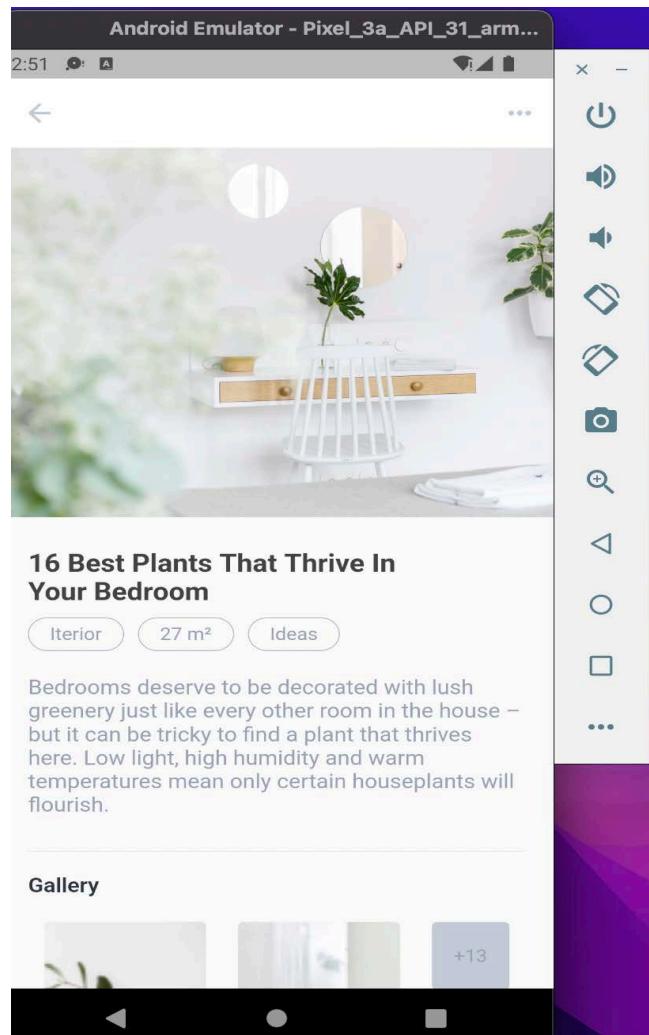
7.DISCUSSION OF RESULTS

7.1 Home page



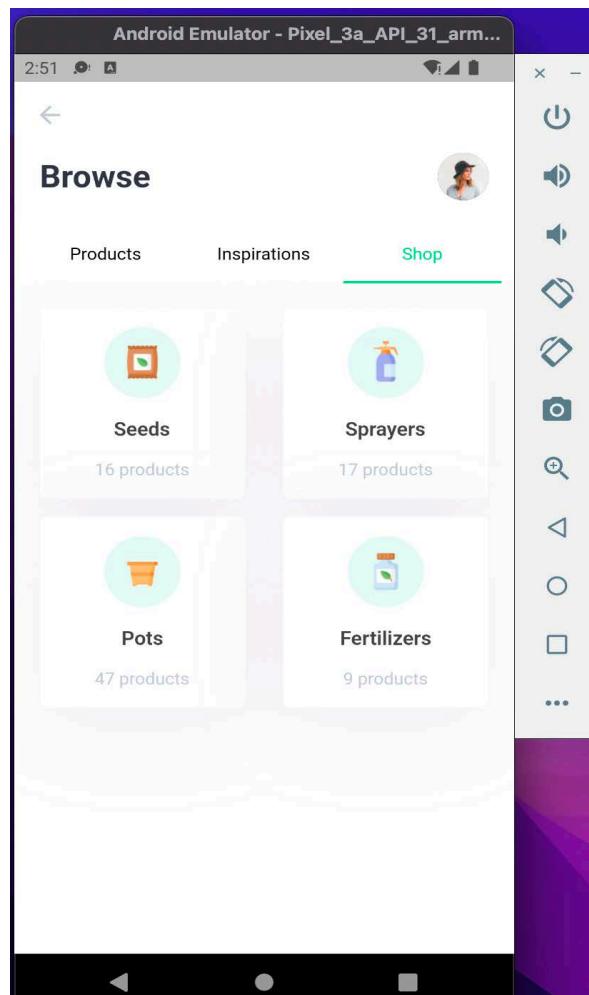
This is the landing page of the application where we can enter the details of the user and then it directs to the main plant app.

7.2 Display Details



This is the details page of app. There will be number of different varieties of plants and seeds etc . User will get to explore all the different varieties possible . User can select a particular plant and get the details for the same.

7.2 Explore Page



This is the explore page. In this page we will see all the sections in our app. You can navigate between these sections and choose the one according to preference.

8.CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

This plant app allows users to know about different plant varieties along with images. Along with that we also have flowers, seeds, sprayers, pots and fertilizers along with their details and images. Users just need to login and at one place they can get all updates of new varieties then they can think of buying it for themselves. It can simply be used to boost their general knowledge about plants and different species.

8.2 Future Enhancement

At the same time there is some scope for improvement in the future. It can be possible to make it more users friendly by adding more variety of functions to it. We can add backend and server. With that all different users will have their details saved. We can host our app on server like aws or heroku. We can add purchasing functionality for all products available in the app. We can do the payment gateway integration.

9. REFERENCES

- Microsoft Visual Studio Step by Step (Developer Reference Edition)
- RamezElmasri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.
- [Learn SQL Tutorial - javatpoint](#)
- [MS SQL Server Tutorial \(tutorialspoint.com\)](#)
- [Flutter Tutorial - GeeksforGeeks](#)
- [What is Dart Language? Explain Architecture & Components\(guru99.com\)](#)