

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By
Thokala Divya sri
40111339



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

OCT 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Thokala Divya Sri (40111339)** who carried out the project entitled "**CLUB HOUSE CLONE**" under my supervision from Aug 2022 to Oct 2022.

Internal Guide

Ms.Dharani

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I Thokala_Divya Sri here by declare that the Project Report entitled CLUB HOUSE CLONE_done by me under the guidance of Dr.Ms.Dharani is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Ms.Dharani** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my projectwork.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

CLUB HOUSE APP is an Audio-based Social Network App,Engaging Informative ,Platform Of Making new Connections. Where users Host Meetings and Discuss Various Topics .It can be Customized based on the Client's Requirements. This project is built by using FLUTTER FRAMEWORK and implementing by DART PROGRAMMING LANGUAGE.

Initially Application Displays The Welcome page and Agree th terms and Conditions Policy By User . And User should enter the details such as Phone Number , Full name , and Username in their Respective page and Set the profile photo. Here comes the main part of the application it displays the Scheduled meetings in the top and followed by the Currently running Meetings .Users can join in any meeting or User can Create A Room and Can Share the link to anyone.

Hence , Club house clone is a Most Flexible Conversative Application ,Fun Way To Learn , Teach and Connect , That You Can join a Club room and Host a Room Freely . Each Room Is like a live Interactive And Unfiltered Podcast.

List Of Figures

- Fig 1.1 Welcome page
- Fig 1.2 PhoneNumber page
- Fig 1.3 Invitation page
- Fig 1.4 Full Name page
- Fig 1.5 User name page
- Fig 1.6 Home Page
- Fig 1.7 Pick photo page
- Fig 1.8 Profile photo page
- Fig 1.9 Lobby bottom page
- Fig 2.0 Room card
- Fig 2.1 Other users page
- Fig 2.2 Followers page

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	
	1.1 Introduction of club house	10
2	OVERVIEW OF THE APPLICATION	
	2.1 App specification	
	2.1.1 Rooms	11
	2.1.2 Events	12
	2.1.3 Clubs	12
3	USED TECHNOLOGIES	
	3.1 Flutter framework	13
	3.2 Dart language	15
4	SYSTEM ANALYSIS AND IMPLEMENTATION OF THE APPLICATION	
	4.1 Installation of app requirements	
	4.1.1 Flutter	16
	4.1.2 Dart	17
	4.1.3 Visual studio code	18
	4.1.4 Android Studio	18
	4.1.5 JDK	19
	4.2 Environmental Setup	19
5	IMPLEMENTATION OF THE APPLICATION	

5.1	Welcome page	20
5.2	Phone number page	21
5.3	Invitation page	22
5.4	Fullname page	23
5.5	Username page	24
5.6	Home page	25
5.6.1	Profile page	26
5.6.2	Lobby page	27
5.6.2.1	Schedule card	28
5.6.2.2	Room card	29
5.6.2.3	LobbyBottomSheet	30
5.6.3	Follower page	30
6	SUMMARY AND CONCLUSIONS	31
6.1	Screenshots	32-34
6.2	References	35
6.3	Source Code	36-43

1.Introduction

1.1 Introduction of club house

Nowadays, a person can hardly imagine a normal life without socializing online. Due to such a need, the Clubhouse became a way to exchange information. Briefly speaking, it's an audio chat room where people gather and discuss specific topics. In the app's early days, hundreds of millions of listeners used it as a substitute for real-life meetups. Usually, a so-called room has two groups: speakers and listeners. Participants can also look through the list of people participating in the debate. Unlike the rest of the social networks, the Clubhouse has a strict hierarchy: you can join a room as an invite-only user. Moreover, the moderator has a right to exclude some participants.

CLUB HOUSE APP is an Audio-based Social Network App,Engaging Informative ,Platform Of Making new Connections. Where users Host Meetings and Discuss Various Topics .It can be Customized based on the Client's Requirements.This project is built by using FLUTTER FRAMEWORK and implementing by DART PROGRAMMING LANGUAGE.

Club House is a closed heirarchial Platform. A moderator oversees discussions and has the ability to let someone chime in or to kick out the unruly. In addition to the "clubs" sorted by topic, two or more users can join together and start their own chat room

1.2 Overview of the application

Clubhouse is a social audio app for iOS and Android where users can communicate in audio chat rooms that accommodate groups of thousands of people.

Clubhouse led to the emergence of a new social media segment known as social audio or drop-in audio. Soon realizing the potential of this segment, a handful of companies came out with their social audio solutions as standalone products or as an expansion to their current products. Clubhouse being the pioneer in this segment, all competitors eventually adapted its features to their products. Club House is a space for casual, drop-in audio conversations—with friends and other interesting people around the world

2.1 App Specification

2.1.1 Rooms: The main feature of Clubhouse is real-time virtual “rooms” in which users can communicate with each other via audio. Rooms are categorized based on differing levels of privacy. “Open rooms” can be joined by anyone on Clubhouse, and all rooms default to this setting on creation. In “social rooms,” only users followed by the moderators are allowed to join. Users need to receive an invite from the moderators to join “closed rooms.” Within a room, there are three sections: the “stage,” “followed by the speakers,” and “others in the room.”

The profile picture and name of each user present in a room are displayed in the appropriate section. When a user creates a room, they are assigned the role of “moderator” which gives them the power to call users to the stage, mute users, and remove speakers from the stage. The moderator role is denoted by a green star that appears next to the user's name. When a user joins a room, they are initially assigned to the role of a “listener” and cannot unmute themselves. Listeners can notify the moderators of their intent to join the stage and speak by clicking on the “raise hand” icon. Users who are invited to the stage become

“speakers,” and gain the ability to unmute themselves. Users can exit a room by tapping the “leave quietly” button or the peace sign emoji

2.1.2 Events: A lot of conversations in Clubhouse happen spontaneously, but users can schedule conversations by creating events. When scheduling an event, users can first name the event and then set the date and time that the conversation will begin. Users can also add co-hosts to help moderate the event. Once an event is created, it is added to the Clubhouse “bulletin.” The bulletin displays upcoming scheduled events and allows users to set notifications for events by clicking the bell icon corresponding to the event. Users can access the bulletin by clicking on the calendar icon at the top of the home page.

2.1.3 Clubs: At the Clubhouse, clubs are user communities that regularly discuss a common interest. Many clubs are present in Clubhouse representing a wide array of topics and users can find clubs by name under the search tab. A club consists of three categories of users - “Admin”, “Leader”, and “Member”. Members can create private rooms and invite more users into the club. Leaders have all the privileges of a member, also authorized to create/schedule club-branded open rooms. An admin can modify club settings, add/delete users, change user privileges and create/schedule any type of room. There are three types of clubs - “Open”, “By Approval”, and “Closed” for membership. Any user can join an open club by pressing the "Join The Club" button on the club profile.

In case of approval, users need to apply and wait for membership by pressing the "Apply To Join" button on the club profile. The admins of the respective club are privileged to accept or reject the user's request. In a closed club, membership is limited to users selected by the club admin. All users of a club will be notified when a public room within the club is created.

3.Used Technologies

3.1 Flutter Framework

Flutter is Google's open-source technology for creating mobile, desktop, and web apps with a single codebase. Unlike other popular solutions, Flutter is not a framework or library; it's a complete SDK – software development kit.

A library is basically a reusable piece of code that you put in your application to perform a particular function.

A framework is a structure that provides you with a skeleton architecture for building software. It's a set of tools that serves as a foundation for your app, requiring you to fill in the blanks with your code to complete the entire structure and get the desired functionality.

Flutter SDK:

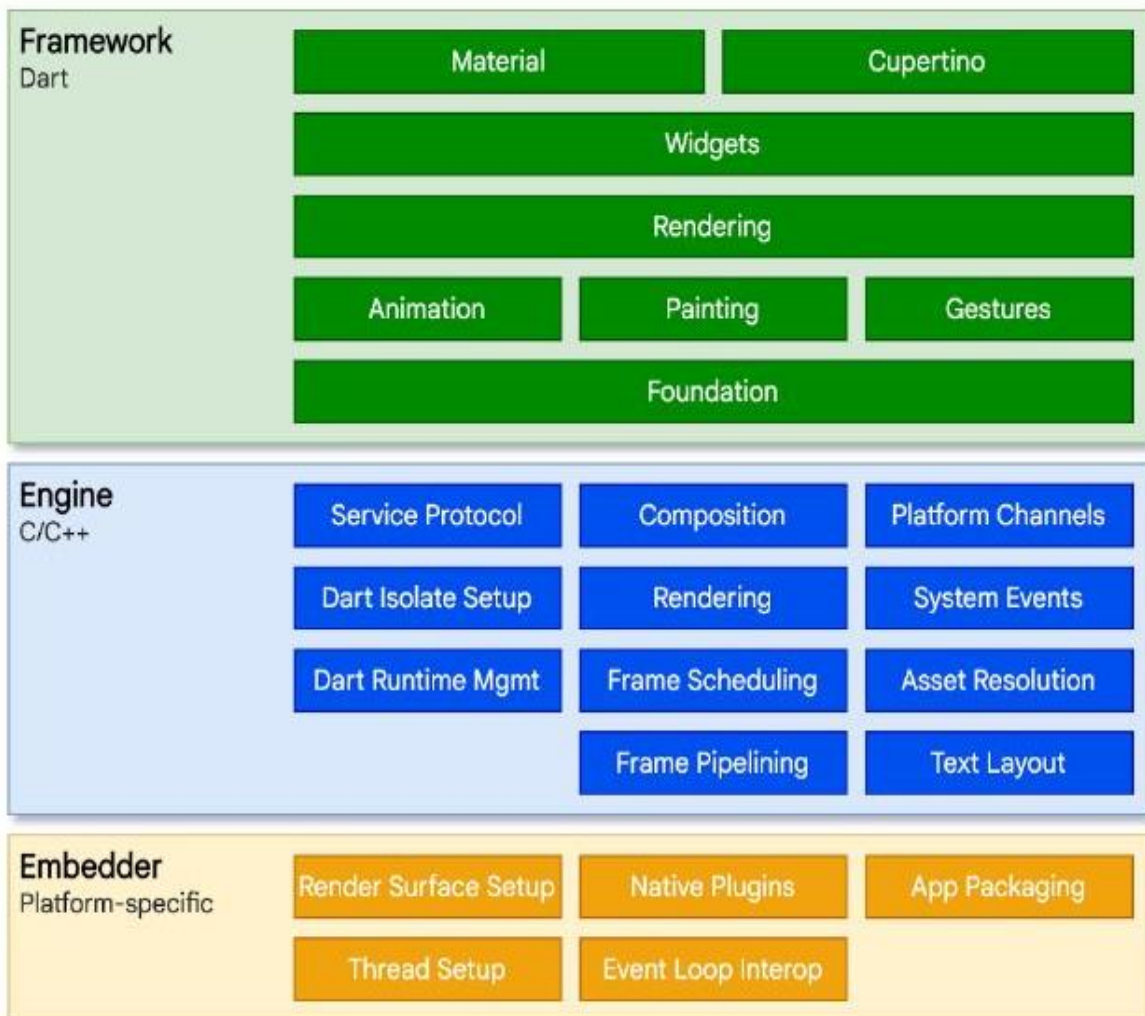
An SDK has a much wider scope as it's a collection of tools, including libraries, documentation, APIs, sometimes frameworks, and more, giving you all you need for software development. And that's the case with Flutter — it already contains everything necessary to build cross-platform applications.

Pros Of Flutter:

- Dart programming language – a simple and effective tool
- Ahead-of-Time (AOT) and Just-in-Time (JIT) compilation types
- No need for XML files
- No need for intermediate bridges

Cons Of Flutter:

- Lack of third-party libraries
- Relatively low adoption of Dart
- Flutter app size



3.2 Dart Language

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

The Dart language is type safe; it uses static type checking to ensure that a variable's value always matches the variable's static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during

experimentation or for code that needs to be especially dynamic.

Dart offers sound null safety, meaning that values can't be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is always non-nullable. If you inspect your running code in the debugger, you'll see that non-nullability is retained at runtime (hence sound null safety).

4.System analysis and implementation of the application

4.1 Installation of app requirements

Requirements:

- 8GB Ram
- SSD (HDD - replace) : 400mb/s - 4000mb/s
- Windows, MacOS, Linux

4.1.1 Flutter

- ZIP file
- (<https://flutter.dev>)Flutter for Windows (<https://docs.flutter.dev/get-started/install/windows>)
- Downloading the file - extract it in the following directory (C:\src\flutter)

4.1.2 Dart

- Zip file
- (<https://dart.dev/get-dart/archive>) dart archive
- Extract it into the C directory
- Setting up the environment variable (in windows)

4.1.3 Visual studio code

Visual Studio Code is the part of visual studio family which is developed

by Microsoft in November 2015. It is based-on Electron framework which is used for Node.js (node java script). It is written in TypeScript, JavaScript and CSS /15/16/.

Features of Visual Studio Code:

- It is open-source and Freeware text editor for private and commercial purposes
- It is cross-platform source code editor debugger
- It supports many different programming languages while only proper installation of the extension is required for React JS, Java, JavaScript, C++, C#, Python etc.
- GitHub is built-in
- Products availability as it explains the definition and show opening and closing of the brackets etc. 19
- It provided unique customizations
- Provides Visual Studio Keymap extension for using Key binding
- It provides in the portable mode that means it keeps data and settings in the same location of installation is possible even on a USB drive
- It is available in many different language services
- It is also available in Remote Development mode /15/16/. Installation of Visual Studio Code: All platforms installer is available, and the developer can pick the proper one for the project.

VS Code (exe file):

- <https://code.visualstudio.com/download>

4.1.4 Android Studio

- (<https://developer.android.com/studio>)
- .exe file - install it as per the procedure
- After installation, we need to add plugins Flutter & Dart
- Restart your IDE

4.1.5 JDK

- Downloading the JDK Installer
- Running the JDK Installer
- Setting the PATH Environment Variable.

4.2 Environmental setup:

Emulator Set Up:

- Create Virtual Device (with recent Android OS - v11.0)
- AVD (Android Virtual Device)

SDK set up:

- Set up the SDK path and update it in environment variable
- Choose the target Android OS

VS Code (exe file):

- <https://code.visualstudio.com/download>

Check Installations:

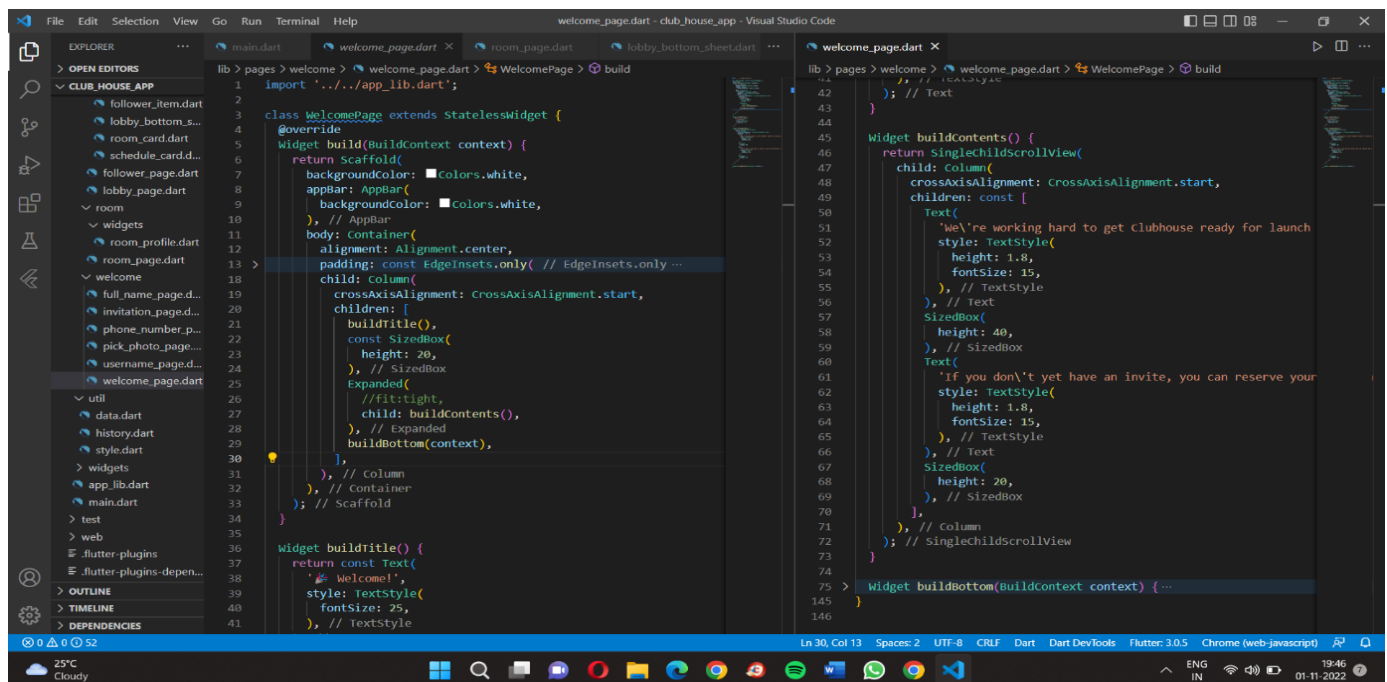
- where.exe flutter dart
- flutter doctor (command to help you with the installation)

- flutter doctor --android-licenses (to accept android licenses)

5. Implementation of the application

5.1 Welcome Page:

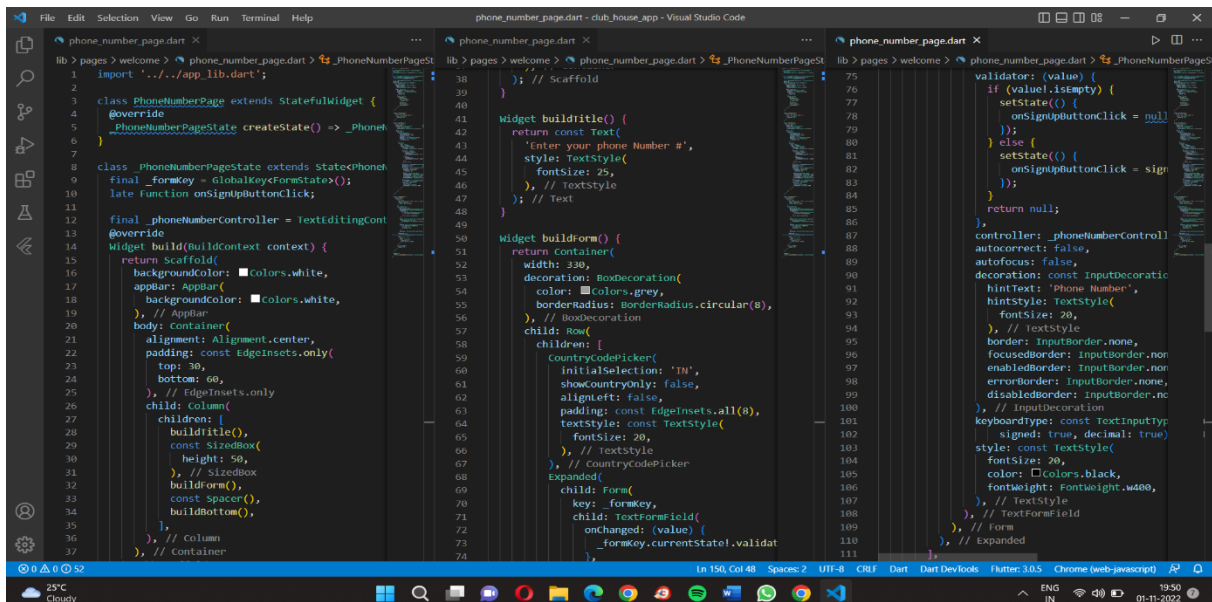
- Import Material Package to use the Built in Functionality
- Create a file welcome_page.dart And create a welcomepage class, It inherits The Properties of StatelessWidget .
- StatelessWidget: It is a static widget , which does not depend on Data or Behaviour change in runtime.
- StatelessWidget has build() function which returns Scaffold
 - Scaffold contains:
 - AppBar
 - Body
- Create A widget build contents() , buildtitle(),buildbottom().
- Call That Functions in Body Of Scaffold



```
lib > pages > welcome > welcome_page.dart > WelcomePage > build
1 import '../app_lib.dart';
2
3 class WelcomePage extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       backgroundColor: Colors.white,
8       appBar: AppBar(
9         backgroundColor: Colors.white,
10        // AppBar
11      ),
12      body: Container(
13        alignment: Alignment.center,
14        padding: const EdgeInsets.only( // EdgeInsets.only ...
15          child: Column(
16            crossAxisAlignment: CrossAxisAlignment.start,
17            children: [
18              buildTitle(),
19              const SizedBox(
20                height: 20,
21              ), // SizedBox
22              Expanded(
23                //fit:tight,
24                child: buildContents(),
25              ), // Expanded
26              buildBottom(context),
27            ],
28          ), // Column
29        ), // Container
30      ); // Scaffold
31    }
32
33    widget buildTitle() {
34      return const Text(
35        'Welcome!',
36        style: TextStyle(
37          fontSize: 25,
38        ), // TextStyle
39      );
40    }
41
42    widget buildContents() {
43      return SingleChildScrollView(
44        child: Column(
45          crossAxisAlignment: CrossAxisAlignment.start,
46          children: const [
47            Text(
48              'We\'re working hard to get Clubhouse ready for launch',
49              style: TextStyle(
50                height: 1.8,
51                fontSize: 15,
52              ), // TextStyle
53            ), // Text
54            SizedBox(
55              height: 40,
56            ), // SizedBox
57            Text(
58              'If you don\'t yet have an invite, you can reserve your',
59              style: TextStyle(
60                height: 1.8,
61                fontSize: 15,
62              ), // TextStyle
63            ), // Text
64            SizedBox(
65              height: 20,
66            ), // SizedBox
67          ],
68        ), // Column
69      ); // SingleChildScrollView
70    }
71
72    widget buildBottom(BuildContext context) {
73      return const Text(
74        'Join the waitlist',
75        style: TextStyle(
76          color: Colors.blue,
77          fontWeight: FontWeight.bold,
78          fontSize: 18,
79        ), // TextStyle
80      );
81    }
82  }
83}
```

5.2 Phone Number Page:

- Create a file `phonenumber.dart` and Define a class `phonenumber` extends `StatefulWidget` class
- `Statefullwidget`: It is a Dynamic widget ,It is mutable.
- `StatefulWidget` has `build()` function which returns `Scaffold`
- Scaffold Contains:
 - App Bar
 - Body
- Create a widget `buildTitle()`,`buildForm()`,`buildBottom()` functions
- Call That Functions in the body of scaffold.
- Here,user can select the country code to enter his/her number



```
lib > pages > welcome > phone_number_page.dart > _PhoneNumberPageSt
1 import '../app_lib.dart';
2
3 class PhoneNumbersPage extends StatefulWidget {
4   @override
5   _PhoneNumberPageState createState() -> _PhoneN
6
7
8 class _PhoneNumberPageState extends State<PhoneN
9   final _formKey = GlobalKey<FormState>();
10  late function onSignUpButtonClick;
11
12  final _phoneNumberController = TextEditingController
13
14  @override
15  Widget build(BuildContext context) {
16    return Scaffold(
17      backgroundColor: Colors.white,
18      appBar: AppBar(
19        backgroundColor: Colors.white,
20      ), // AppBar
21      body: Container(
22        alignment: Alignment.center,
23        padding: const EdgeInsets.only(
24          top: 30,
25          bottom: 60,
26        ), // EdgeInsets.only
27        children: [
28          buildTitle(),
29          const SizedBox(
30            height: 50,
31          ), // SizedBox
32          buildForm(),
33          const Spacer(),
34          buildBottom(),
35        ], // Column
36      ), // Container
37    );
```

- **5.3 Invitation Page:**
- Import Material Package to use the Built in Functionality
- Create a file invitationPage.dart And create a class welcomepage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body
- Create widget functions buildTitle(),buildContents(),buildBottom() and call these functions in body
- This Page displays about the inviter.

```

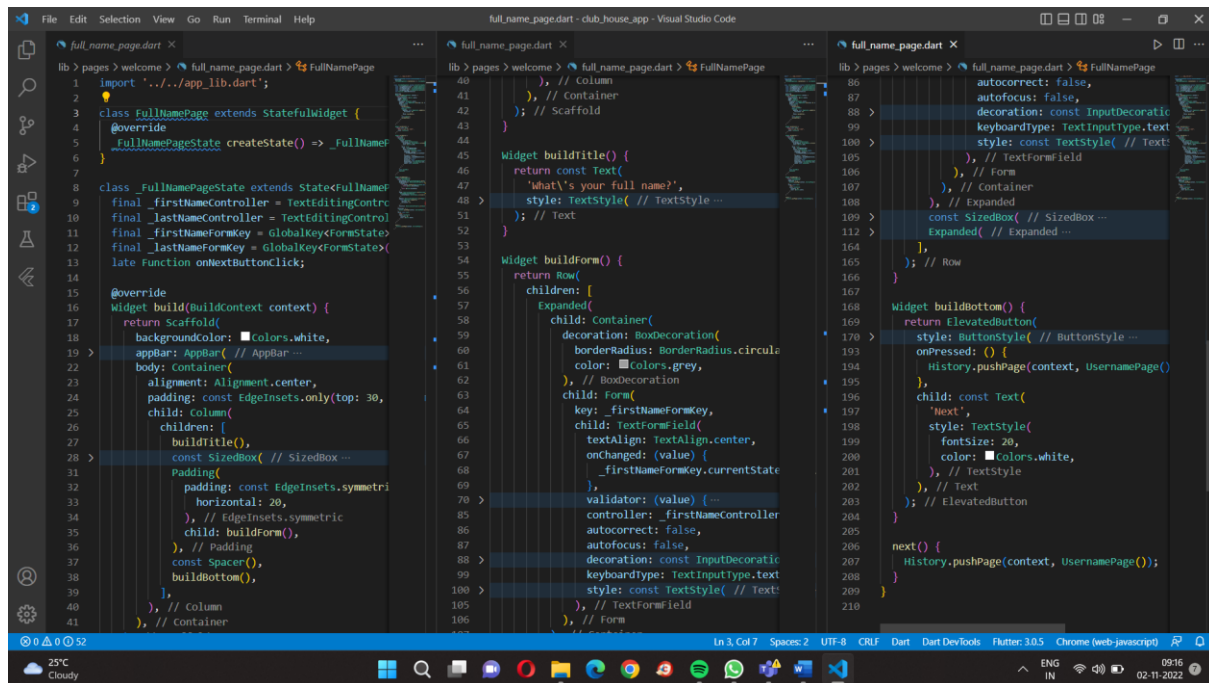
lib > pages > welcome > invitation_page.dart > InvitationPage > build
1 import '../app_lib.dart';
2
3 class InvitationPage extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       backgroundColor: Colors.white,
8       appBar: AppBar(
9         backgroundColor: Colors.white,
10      ), // AppBar
11      body: Container(
12        alignment: Alignment.center,
13        padding: EdgeInsets.only(
14          top: 30,
15          bottom: 60,
16        ), // EdgeInsets.only
17        child: Column(
18          children: [
19            buildTitle(),
20            const SizedBox(
21              height: 50,
22            ), // SizedBox
23            Expanded(
24              child: buildContents(),
25            ), // Expanded
26            buildBottom(context),
27          ],
28        ), // Column
29      ), // Container
30    ), // Scaffold
31  }
32
33  Widget buildTitle() {
34    return const Text(
35      'Welcome to Clubhouse, You're Divya',
36      style: TextStyle(
37        fontSize: 25,
38      ),
39    );
40  }
41
42  Widget buildContents() {
43    return SingleChildScrollView(
44      child: Column(
45        children: [
46          Container(
47            height: 150,
48            width: 150,
49            decoration: BoxDecoration(
50              borderRadius: BorderRadius.circular(
51                15,
52              ),
53              image: const DecorationImage(
54                image: AssetImage('assets/images/
55                // : NetworkImage(url),
56                fit: BoxFit.cover,
57              ), // DecorationImage
58            ), // Container
59            const SizedBox(
60              height: 10,
61            ), // SizedBox
62            const Text(
63              'Divya',
64              style: TextStyle(
65                fontWeight: FontWeight.bold,
66              ), // TextStyle
67            ), // Text
68          ], // Column
69        ), // SingleChildScrollView
70      );
71    }
72
73    Widget buildBottom(BuildContext context) {
74      return Column(
75        children: [
76          const Text(
77            'let's set up your profile',
78            style: TextStyle(
79              fontSize: 20,
80            ),
81          ),
82          const SizedBox(
83            height: 30,
84          ), // SizedBox
85          ElevatedButton(
86            style: ButtonStyle(
87              minimumSize: MaterialStateProperty.all(
88                Size(150, 50),
89              ),
90              backgroundColor: MaterialStateProperty(
91                (states) {
92                  if (states.contains(MaterialState.d
93                    return Colors.accentBrown;
94                  }
95                  return Colors.accentBlue;
96                ),
97              shape: MaterialStateProperty.all(Outline
98                RoundedRectangleBorder(
99                  borderRadius: BorderRadius.circular(
100                  15,
101                ), // RoundedRectangleBorder
102              ),
103              padding: MaterialStateProperty.all(Edge
104                EdgeInsets.symmetric(
105                  vertical: 10,
106                  horizontal: 25,
107                ), // EdgeInsets.symmetric
108              ),
109              elevation: MaterialStateProperty.all(0
110            ), // ButtonStyle
111            onPressed: () {
112              History.pushPage(context, FullNamePage(
113            ),
114            child: const Text(
115              'Next',
116              style: TextStyle(
117                fontSize: 20,
118                color: Colors.white,
119              ), // TextStyle
120            ), // Text
121          ), // ElevatedButton

```

5.5 Full Name Page

Import Material Package to use the Built in Functionality

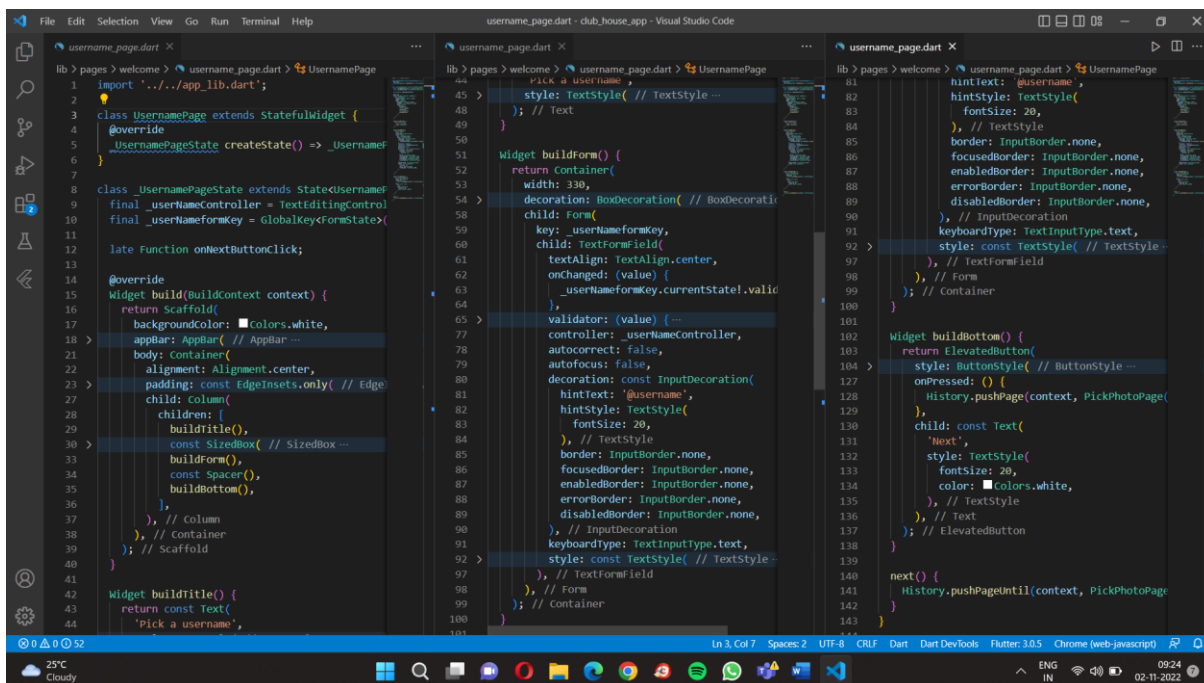
- Create a file `fullNamePage.dart` And create a class `fullNamePage`, It inherits The Properties of `Stateless widget`.
- `Build()` function returns `Scaffold.scaffold` consists of `appBar` and `body`
- Create widget functions `buildTitle()`,`buildForm()`,`buildBottom()` and call these functions in `body`
- In this page, User enter his/her full name.



5.6 User Name Page

Import Material Package to use the Built in Functionality

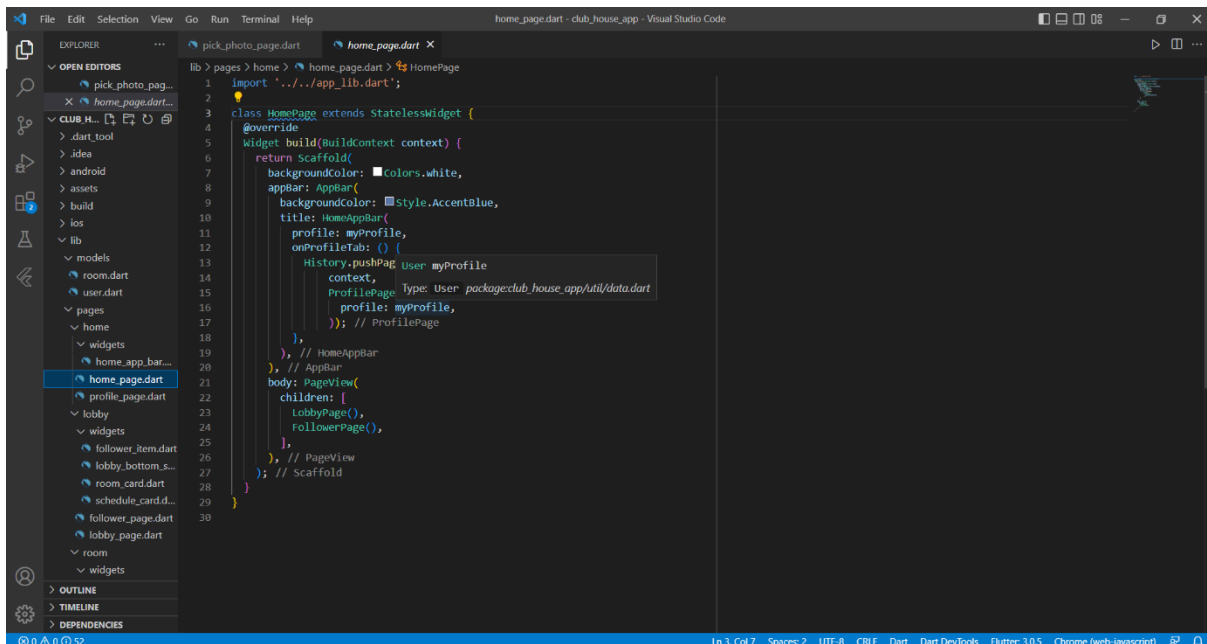
- Create a file `username_page.dart` And create a class `usernamePage`, It inherits The Properties of `Stateful widget`.
- `Build()` function returns `Scaffold.scaffold` consists of `AppBar` and `body`
- Create widget functions `buildTitle()`, `buildForm()`, `buildBottom()` and call these functions in `body`
- In this page, user enter the username that should be unique.



5.8 Home Page

Import Material Package to use the Built in Functionality

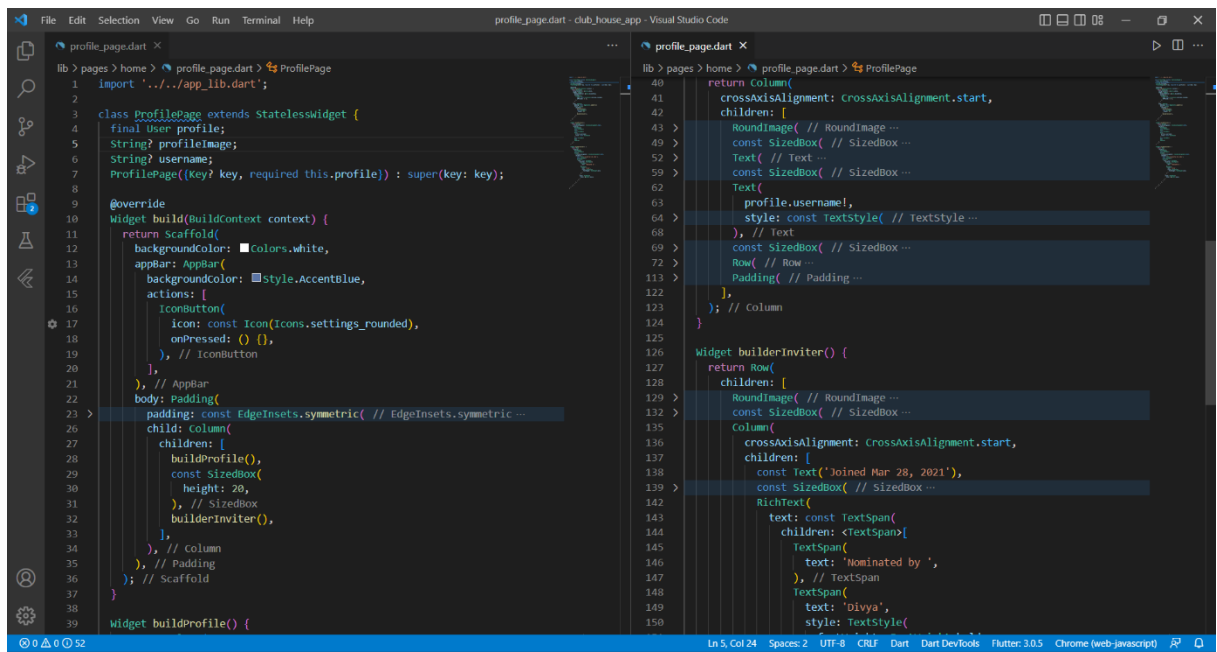
- Create a file HomePage.dart And create a class HomePage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body
- Body consists of two classes
 - Lobby Page
 - Follower Page



5.8.1 Profile Page

Import Material Package to use the Built in Functionality

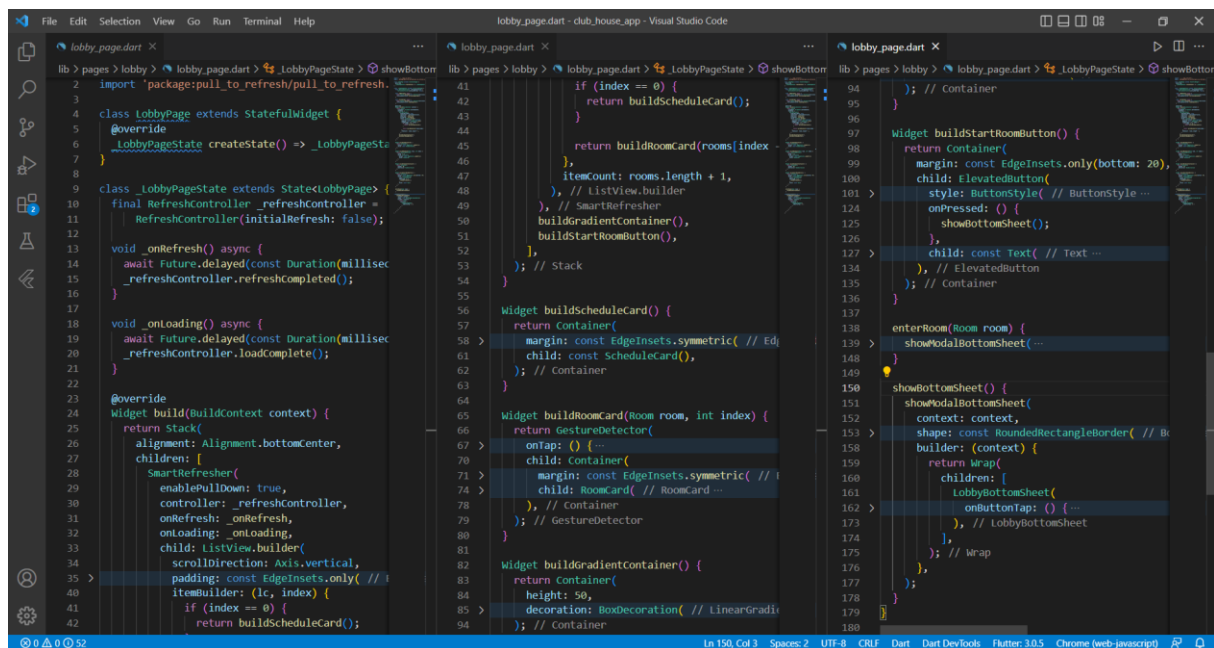
- Create a file ProfilePage.dart And create a class ProfilePage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body



```
lib > pages > home > profile_page.dart > ProfilePage
1 import '../app_lib.dart';
2
3 class ProfilePage extends StatelessWidget {
4   final User profile;
5   String? profileImage;
6   String? username;
7   ProfilePage({key? key, required this.profile}) : super(key: key);
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      backgroundColor: Colors.white,
13      appBar: AppBar(
14        backgroundColor: style.AccentBlue,
15        actions: [
16          IconButton(
17            icon: const Icon(Icons.settings_rounded),
18            onPressed: () {},
19          ), // IconButton
20        ],
21      ), // AppBar
22      body: Padding(
23        padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ...
24          children: [
25            buildProfile(),
26            const SizedBox(
27              height: 20,
28            ), // SizedBox
29            builderInviter(),
30          ],
31        ), // Column
32      ), // Scaffold
33    );
34  }
35
36  Widget buildProfile() {
37
38
39
40
41    return Column(
42      crossAxisAlignment: CrossAxisAlignment.start,
43      children: [
44        RoundImage( // RoundImage ...
45          const SizedBox( // SizedBox ...
46            Text( // Text ...
47              const SizedBox( // Sizedbox ...
48                Text(
49                  profile.username!,
50                  style: const TextStyle( // TextStyle ...
51                ), // Text
52              ), // Text
53            ), // Text
54          ), // Text
55        ), // Text
56        const SizedBox( // Sizedbox ...
57        Row( // Row ...
58          Padding( // Padding ...
59        ],
60      ], // Column
61    ); // Column
62  }
63
64  Widget builderInviter() {
65    return Row(
66      children: [
67        RoundImage( // RoundImage ...
68        const SizedBox( // Sizedbox ...
69        Column(
70          crossAxisAlignment: CrossAxisAlignment.start,
71          children: [
72            const Text("Joined Mar 28, 2021"),
73            const SizedBox( // Sizedbox ...
74            RichText(
75              text: const TextSpan(
76                children: <textSpan>[
77                  TextSpan(
78                    text: 'Nominated by ',
79                  ), // TextSpan
80                  TextSpan(
81                    text: 'Divya',
82                    style: TextStyle(
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

5.8.2 Lobby Page

- Import Material and pull_to_refresh package.
- Create a file LobbyPage.dart And create a class LobbyPage, It inherits The Properties of StatefulWidget.
- Build() function returns stack.stack consists of Body
 - Body consists of
 - SmartRefereshier
 - ScheduleCard
 - RoomCard
 - LobbyBottomSheet
 - buildGradientContainer()-(userDefined Function)
 - buildStartRoomButton()-(userDefined Function)

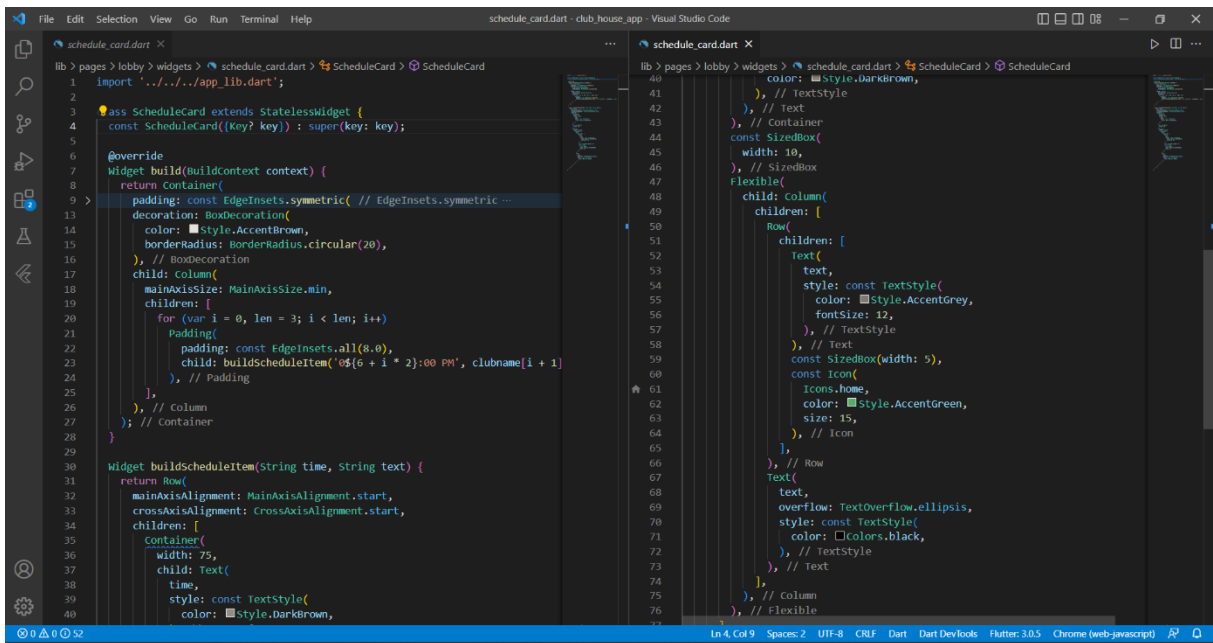


```
lib > pages > lobby > lobby_page.dart > LobbyPageState > showBottom
2 import 'package:pull_to_refresh/pull_to_refresh.';
3
4 class LobbyPage extends StatefulWidget {
5   @override
6   LobbyPageState createState() => _LobbyPageState;
7 }
8
9 class _LobbyPageState extends State<LobbyPage> {
10   final RefreshController _refreshController =
11     RefreshController(initialRefresh: false);
12
13   void _onRefresh() async {
14     await Future.delayed(const Duration(milliseconds: 2000));
15     _refreshController.refreshCompleted();
16   }
17
18   void _onLoading() async {
19     await Future.delayed(const Duration(milliseconds: 2000));
20     _refreshController.loadComplete();
21   }
22
23   @override
24   Widget build(BuildContext context) {
25     return Stack(
26       alignment: Alignment.bottomCenter,
27       children: [
28         SmartRefereshier(
29           enablePullDown: true,
30           controller: _refreshController,
31           onRefresh: _onRefresh,
32           onLoading: _onLoading,
33           child: ListView.builder(
34             scrollDirection: Axis.vertical,
35             padding: const EdgeInsets.only(top: 10),
36             itemBuilder: (lc, index) {
37               if (index == 0) {
38                 return buildScheduleCard();
39               }
40               if (index == 1) {
41                 return buildRoomCard(rooms[index],
42                   itemCount: rooms.length + 1,
43                   // SmartRefereshier
44                   buildGradientContainer(),
45                   buildStartRoomButton(),
46                 ); // Stack
47               }
48             }, // Container
49           ), // Container
50         ), // Container
51         buildStartRoomButton(),
52       ], // Stack
53     );
54   }
55
56   Widget buildScheduleCard() {
57     return Container(
58       margin: const EdgeInsets.symmetric(horizontal: 10),
59       child: const ScheduleCard(),
60     ); // Container
61   }
62
63   Widget buildRoomCard(Room room, int index) {
64     return GestureDetector(
65       onTap: () { ... },
66       child: Container(
67         margin: const EdgeInsets.symmetric(horizontal: 10),
68         child: RoomCard( // RoomCard ...
69       ), // Container
70     ); // GestureDetector
71   }
72
73   Widget buildGradientContainer() {
74     return Container(
75       height: 50,
76       decoration: BoxDecoration( // LinearGradient
77     ); // Container
78   }
79
80   Widget buildStartRoomButton() {
81     return Container(
82       margin: const EdgeInsets.only(bottom: 20),
83       child: ElevatedButton(
84         style: ButtonStyle( // ButtonStyle ...
85         onPressed: () {
86           showBottomSheet();
87         },
88         child: const Text( // Text ...
89       ), // ElevatedButton
90     ); // Container
91   }
92
93   enterRoom(Room room) {
94     showModalBottomSheet(
95       context: context,
96       shape: const RoundedRectangleBorder( // B
97       builder: (context) {
98         return Wrap(
99           children: [
100             LobbyBottomSheet(
101               onPressed: () { ... }, // LobbyBottomSheet
102             ), // Wrap
103           ],
104         );
105       },
106     );
107   }
108
109   showBottomSheet() {
110     showModalBottomSheet(
111       context: context,
112       shape: const RoundedRectangleBorder( // B
113       builder: (context) {
114         return Wrap(
115           children: [
116             LobbyBottomSheet(
117               onPressed: () { ... }, // LobbyBottomSheet
118             ), // Wrap
119           ],
120         );
121       },
122     );
123   }
124 }
```

5.8.2.1 ScheduleCard

Import Material Package to use the Built in Functionality

- Create a file ScheduleCard.dart And create a class ScheduleCard, It inherits The Properties of StatelessWidget.
- It will display the list of scheduled meetings

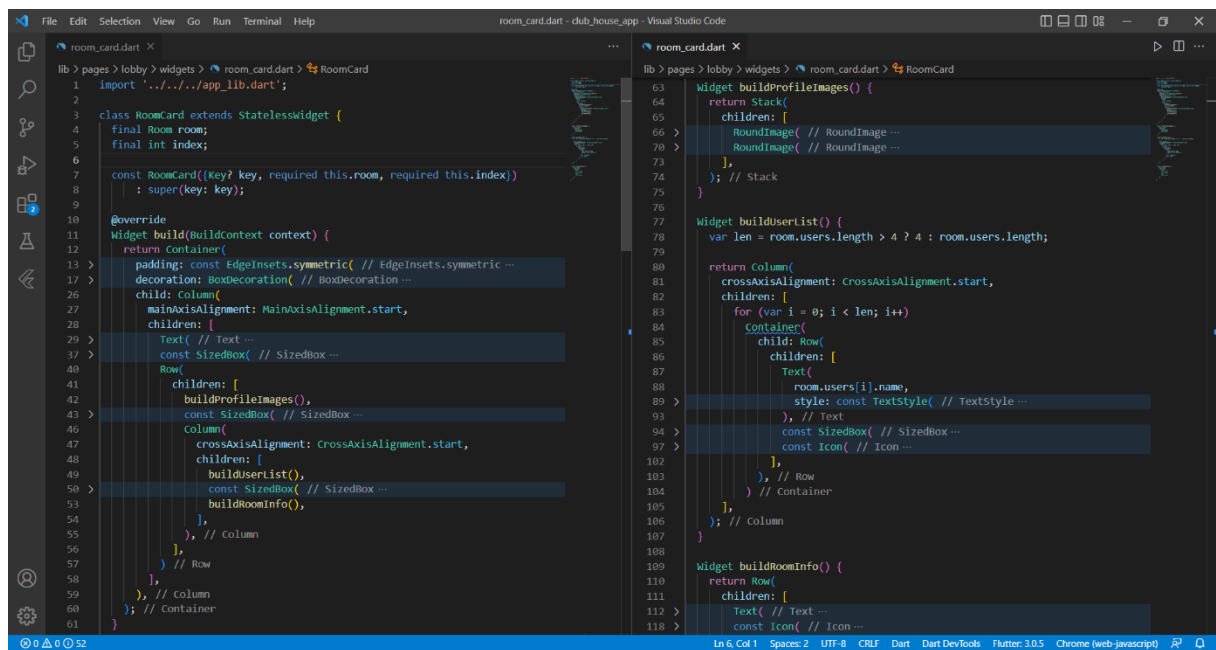


```
lib > pages > lobby > widgets > schedule_card.dart > ScheduleCard > ScheduleCard
1 import '../..../app_lib.dart';
2
3 class ScheduleCard extends StatelessWidget {
4   const ScheduleCard({key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Container(
9       padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ...
10     ),
11     decoration: BoxDecoration(
12       color: ColorsAccentBrown,
13       borderRadius: BorderRadius.circular(20),
14     ), // BoxDecoration
15     child: Column(
16       mainAxisAlignment: MainAxisAlignment.min,
17       children: [
18         for (var i = 0, len = 3; i < len; i++)
19           Padding(
20             padding: const EdgeInsets.all(8.0),
21             child: buildScheduleItem('${6 + i * 2}:00 PM', clubname[i + 1])
22           ), // Padding
23       ], // Column
24     ); // Container
25
26   Widget buildScheduleItem(String time, String text) {
27     return Row(
28       mainAxisAlignment: MainAxisAlignment.start,
29       crossAxisAlignment: CrossAxisAlignment.start,
30       children: [
31         Container(
32           width: 75,
33           child: Text(
34             time,
35             style: const TextStyle(
36               color: ColorsDarkBrown,
37             ), // TextStyle
38           ), // Text
39         ), // Container
40         const SizedBox(width: 10), // SizedBox
41         Flexible(
42           child: Column(
43             children: [
44               Row(
45                 children: [
46                   Text(
47                     text,
48                     style: const TextStyle(
49                       color: ColorsAccentGrey,
50                       fontSize: 12,
51                     ), // TextStyle
52                   ), // Text
53                   const SizedBox(width: 5),
54                   Icon(
55                     Icons.home,
56                     color: ColorsAccentGreen,
57                     size: 15,
58                   ), // Icon
59                 ], // Row
60               ), // Row
61               Text(
62                 text,
63                 overflow: TextOverflow.ellipsis,
64                 style: const TextStyle(
65                   color: ColorsBlack,
66                 ), // TextStyle
67               ), // Text
68             ], // Column
69           ), // Flexible
70         ), // Flexible
71       ], // Row
72     ); // Row
73   }
74 }
```

5.8.2.2 Room Card

Import Material Package to use the Built in Functionality

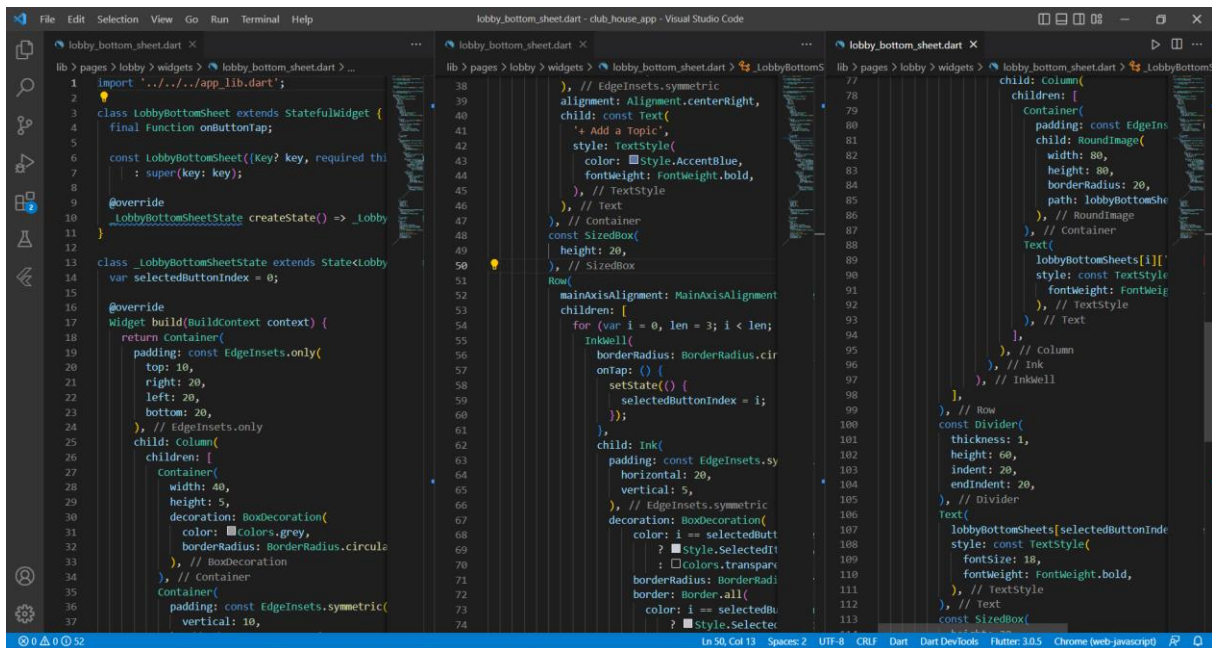
- Create a file RoomCard.dart And create a class RoomCard, It inherits The Properties of StatelessWidget.
- It displays the currently running meetings.
- Users can join the meetings by clicking the room card.



```
lib > pages > lobby > widgets > room_card.dart > RoomCard
1 import '../..../app_lib.dart';
2
3 class RoomCard extends StatelessWidget {
4   final Room room;
5   final int index;
6
7   const RoomCard({Key? key, required this.room, required this.index})
8     : super(key: key);
9
10  @override
11  Widget build(BuildContext context) {
12    return Container(
13      padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ...
14        decoration: BoxDecoration( // BoxDecoration ...
15      child: Column(
16        mainAxisAlignment: MainAxisAlignment.start,
17        children: [
18          Text( // Text ...
19            const SizedBox( // SizedBox ...
20              Row(
21                children: [
22                  buildProfileImages(),
23                  const SizedBox( // SizedBox ...
24                    Column(
25                      crossAxisAlignment: CrossAxisAlignment.start,
26                      children: [
27                        buildUserList(),
28                        const SizedBox( // SizedBox ...
29                          buildRoomInfo(),
30                        ], // Column
31                      ], // Row
32                    ), // Column
33                  ], // Row
34                ), // Column
35              ], // Container
36            ); // Container
37          }
38
39  widget buildProfileImages() {
40    return Stack(
41      children: [
42        RoundImage( // RoundImage ...
43        RoundImage( // RoundImage ...
44      ], // Stack
45    ); // Stack
46  }
47
48  widget buildUserList() {
49    var len = room.users.length > 4 ? 4 : room.users.length;
50
51    return Column(
52      crossAxisAlignment: CrossAxisAlignment.start,
53      children: [
54        for (var i = 0; i < len; i++)
55          container(
56            child: Row(
57              children: [
58                Text(
59                  room.users[i].name,
60                  style: const TextStyle( // TextStyle ...
61                ), // Text
62                const SizedBox( // SizedBox ...
63                const Icon( // Icon ...
64              ], // Row
65            ), // Container
66          ], // Column
67        ); // Column
68      }
69
70  widget buildRoomInfo() {
71    return Row(
72      children: [
73        Text( // Text ...
74        const Icon( // Icon ...
75      ], // Row
76    ); // Row
77  }
```

5.8.2.3 LobbyBottomSheet

- Import Material and pull_to_refresh package.
- Create a file LobbyBottomSheet.dart And create a class LobbyBottomSheet, It inherits The Properties of StatefulWidget.
- It displays the options to create the room with anyone or with specified people.

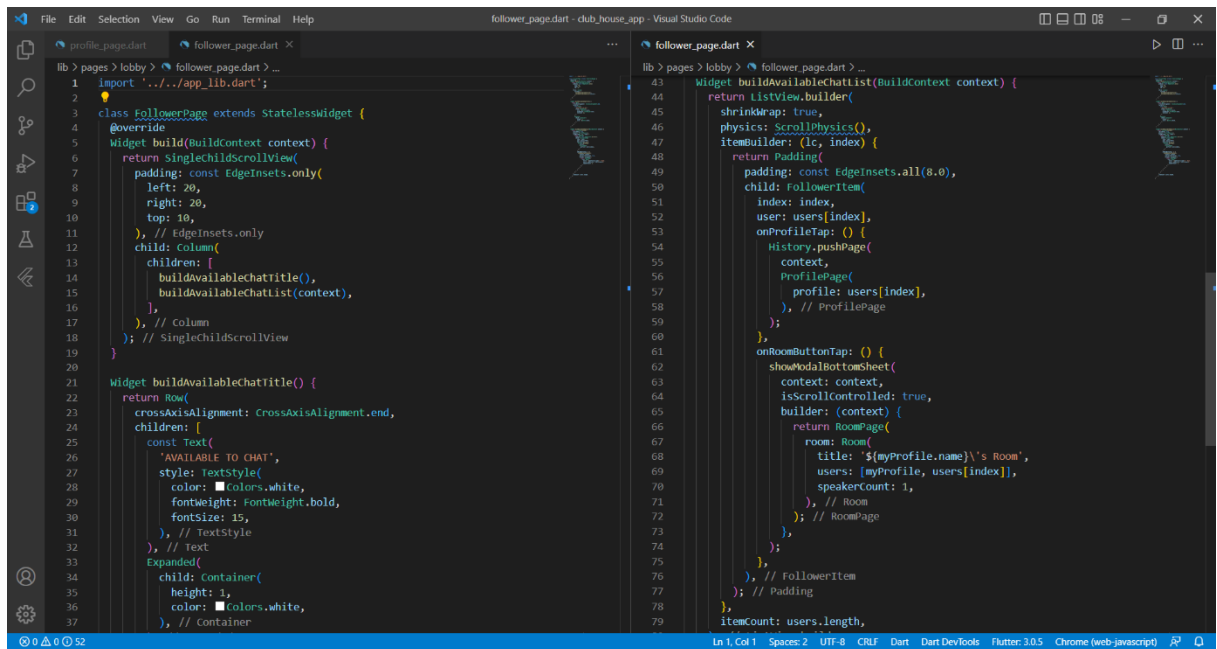


```
lib > pages > lobby > widgets > lobby_bottom_sheet.dart > ...
1 import '../..../app_lib.dart';
2
3 class LobbyBottomSheet extends StatefulWidget {
4   final Function onPressed;
5
6   const LobbyBottomSheet({Key? key, required this.onPressed,
7     : super(key: key)});
8
9   @override
10  LobbyBottomSheetState createState() => _LobbyBottomSheetState;
11
12 }
13
14 class _LobbyBottomSheetState extends State<LobbyBottomSheet> {
15   var selectedIndex = 0;
16
17   @override
18   Widget build(BuildContext context) {
19     return Container(
20       padding: const EdgeInsets.only(
21         top: 10,
22         right: 20,
23         left: 20,
24         bottom: 20,
25       ), // EdgeInsets.only
26       child: Column(
27         children: [
28           Container(
29             width: 40,
30             height: 5,
31             decoration: BoxDecoration(
32               color: Colors.grey,
33               borderRadius: BorderRadius.circular(10),
34             ), // BoxDecoration
35           ), // Container
36           Container(
37             padding: const EdgeInsets.symmetric(
38               vertical: 10,
39             ), // EdgeInsets.symmetric
40             alignment: Alignment.centerRight,
41             child: const Text(
42               '+ Add a Topic',
43               style: TextStyle(
44                 color: Colors.accentBlue,
45                 fontWeight: FontWeight.bold,
46               ), // TextStyle
47             ), // Text
48           ), // Container
49           const SizedBox(
50             height: 20,
51           ), // SizedBox
52           Row(
53             mainAxisAlignment: MainAxisAlignment.spaceBetween,
54             children: [
55               InkWell(
56                 borderRadius: BorderRadius.circular(10),
57                 onTap: () {
58                   setState(() {
59                     selectedIndex = 1;
60                   });
61                 },
62               ),
63               InkWell(
64                 padding: const EdgeInsets.symmetric(
65                   horizontal: 20,
66                   vertical: 5,
67                 ), // EdgeInsets.symmetric
68                 decoration: BoxDecoration(
69                   color: selectedIndex == 0 ? Colors.transparent : Colors.transparent,
70                   borderRadius: BorderRadius.circular(10),
71                   border: Border.all(
72                     color: selectedIndex == 0 ? Colors.transparent : Colors.transparent,
73                   ), // Border.all
74                 ), // BoxDecoration
75               ), // InkWell
76             ], // Row
77           ), // Container
78           const Column(
79             children: [
80               Container(
81                 padding: const EdgeInsets(
82                   width: 80,
83                   height: 80,
84                   borderRadius: 20,
85                 ), // Container
86               ), // RoundImage
87             ], // Container
88           ), // Text
89           lobbyBottomSheets[i]['
90             style: const TextStyle(
91               fontWeight: FontWeight
92             ), // TextStyle
93           ), // Text
94         ], // Column
95       ), // Ink
96     ), // InkWell
97   ], // Row
98   const Divider(
99     thickness: 1,
100     height: 60,
101     indent: 20,
102     endIndent: 20,
103   ), // Divider
104   Text(
105     lobbyBottomSheets[selectedIndex]
106     style: const TextStyle(
107       fontSize: 18,
108       fontWeight: FontWeight.bold,
109     ), // TextStyle
110   ), // Text
111   const SizedBox(
112     height: 20,
113   ), // SizedBox
114 }
```

5.8.3 Follower Page

Import Material Package to use the Built in Functionality

- Create a file FollowerPage.dart And create a class FollowerPage, It inherits The Properties of StatelessWidget.
- Build() returns SingleChildScrollView.It contains Child.



```
lib > pages > lobby > follower_page.dart > ...
1 import '../app_lib.dart';
2
3 class FollowerPage extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return SingleChildScrollView(
7       padding: const EdgeInsets.only(
8         left: 20,
9         right: 20,
10        top: 10,
11      ), // EdgeInsets.only
12       child: Column(
13         children: [
14           buildAvailableChatTitle(),
15           buildAvailableChatlist(context),
16         ], // Column
17     ); // SingleChildScrollView
18   }
19
20   Widget buildAvailableChatTitle() {
21     return Row(
22       crossAxisAlignment: CrossAxisAlignment.end,
23       children: [
24         const Text(
25           'AVAILABLE TO CHAT',
26           style: TextStyle(
27             color: Colors.white,
28             fontWeight: FontWeight.bold,
29             fontSize: 15,
30           ), // TextStyle
31         ), // Text
32         Expanded(
33           child: Container(
34             height: 1,
35             color: Colors.white,
36           ), // Container
37         ),
38       ],
39     );
40   }
41
42   Widget buildAvailableChatlist(BuildContext context) {
43     return ListView.builder(
44       shrinkWrap: true,
45       physics: ScrollPhysics(),
46       itemBuilder: (lc, index) {
47         return Padding(
48           padding: const EdgeInsets.all(8.0),
49           child: FollowerItem(
50             index: index,
51             user: users[index],
52             onProfileTap: () {
53               History.pushPage(
54                 context,
55                 ProfilePage(
56                   profile: users[index],
57                 ), // ProfilePage
58               );
59             },
60             onRoomButtonTap: () {
61               showModalBottomSheet(
62                 context: context,
63                 isScrollControlled: true,
64                 builder: (context) {
65                   return RoomPage(
66                     room: Room(
67                       title: '${myProfile.name}'s Room',
68                       users: [myProfile, users[index]],
69                       speakerCount: 1,
70                     ), // Room
71                   ); // RoomPage
72                 },
73               );
74             },
75           ), // FollowerItem
76         ); // Padding
77       },
78     );
79     itemCount: users.length,
80   }
81 }
```

6. SUMMARY AND CONCLUSIONS

Summary:

Clubhouse represents a shift in how we use social media. The discussions that occur between users are meant to be non-permanent, and the audio-only format creates a different type of conversation compared to those you see on image and text-based social media such as Instagram or Twitter. However, Clubhouse also has serious privacy concerns, and because it is such a young company, it is unclear how it uses the data it collects.

Conclusion:

Clubhouse is built on the idea of connecting people to discuss topics that interest them, and the best way to find things to listen to is to find communities first, then see what rooms you can join within these communities.

6.1 Screenshots

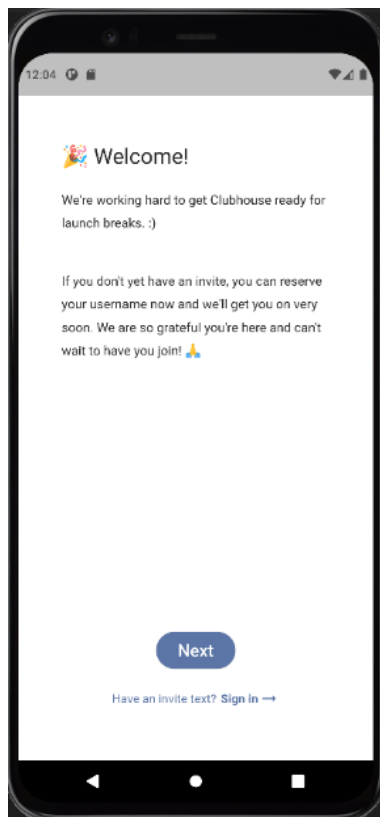


Fig 1.1

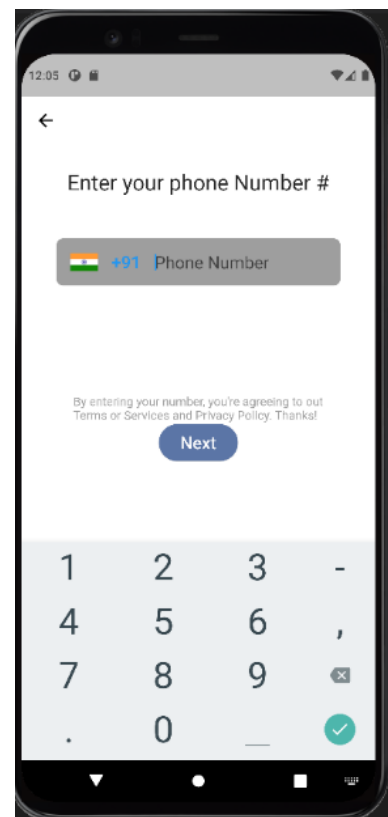


Fig 1.2



Fig 1.3



Fig 1.4



Fig 1.5

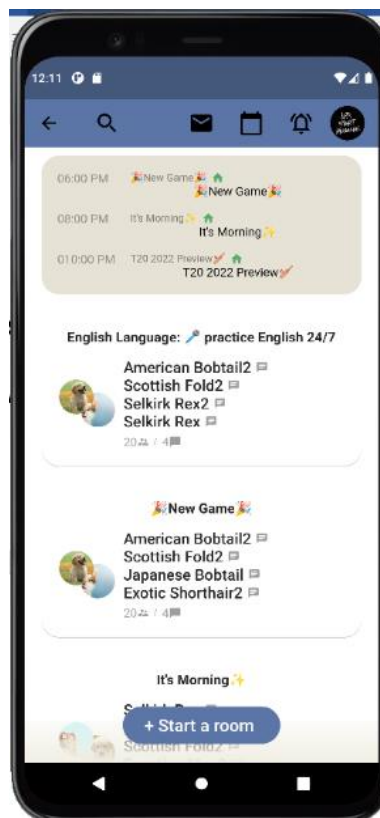


Fig 1.6

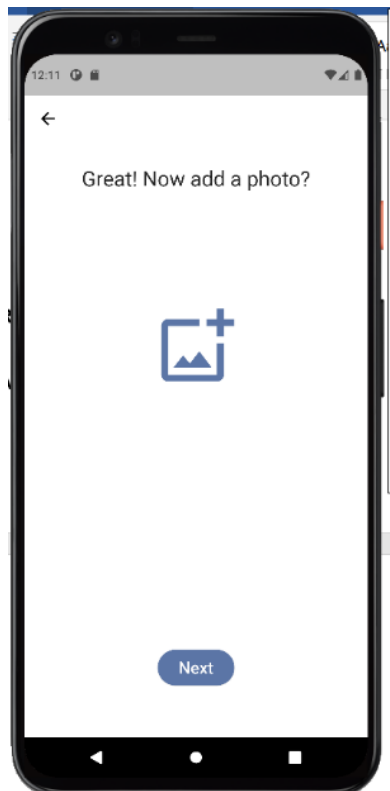


Fig 1.7

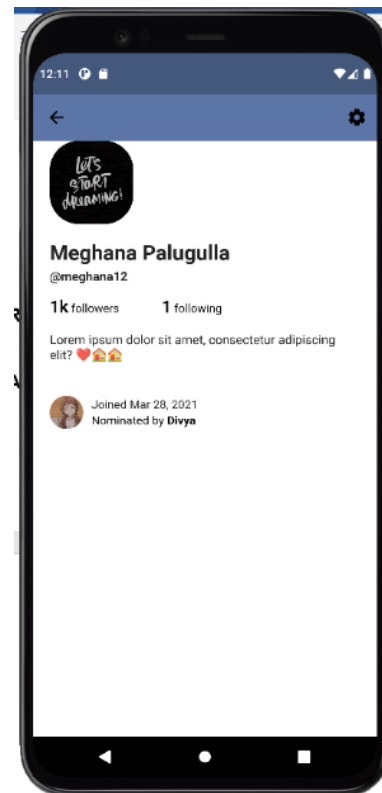


Fig 1.8



Fig 1.9

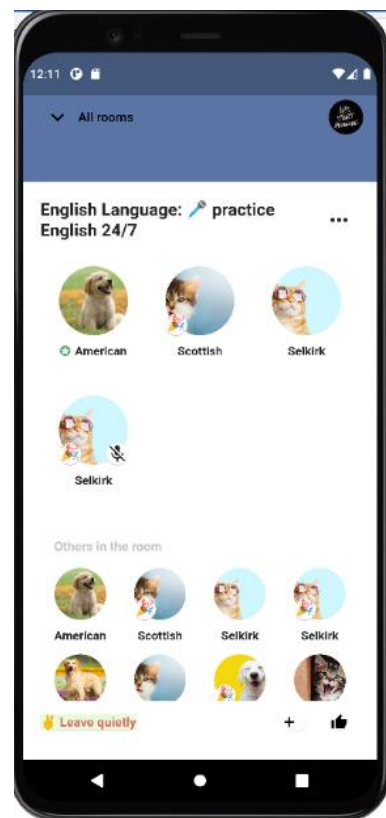


Fig 2.0

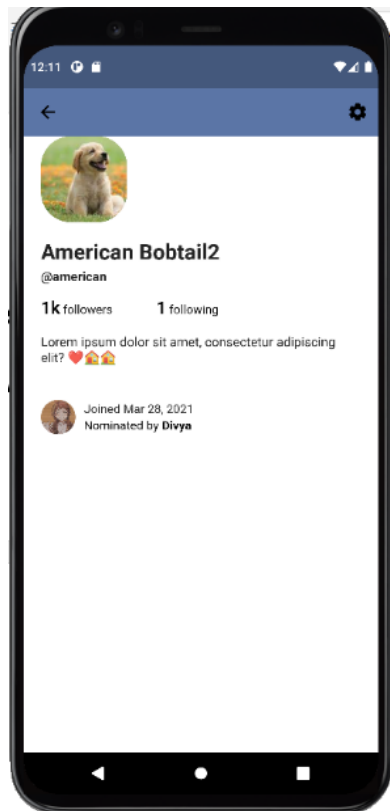


Fig 2.1

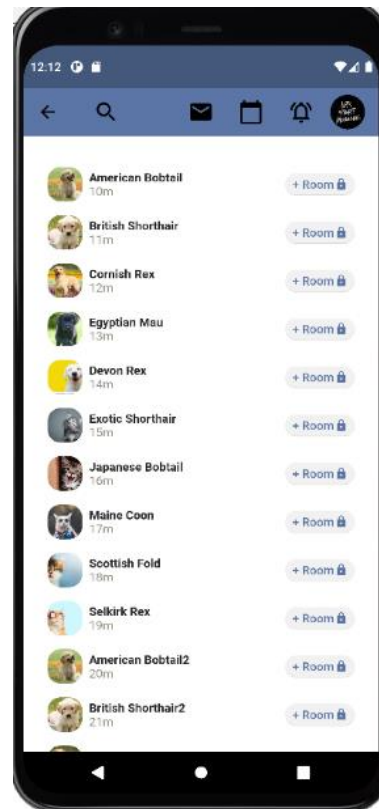


Fig 2.2

6.2 References:

- docs.flutter.dev
- Beginning Flutter: A Hands On guide to App Development (Ebook)
- <https://algofusion.org/course/flutter?tab=content>
- <https://pub.dev/>
- <https://dart.dev/>
- <https://flutter.dev/>
- https://www.tutorialspoint.com/dart_programming/index.html

Source code:

```
import '../..//app_lib.dart';

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Style.AccentBlue,
        title: HomeAppBar(
          profile: myProfile,
          onProfileTab: () {
            History.pushPage(
              context,
              ProfilePage(
                profile: myProfile,
              ));
          },
        ),
      ),
      body: PageView(
        children: [
          LobbyPage(),
          FollowerPage(),
        ],
      ),
    );
  }
}

import '../..//app_lib.dart';
import 'package:pull_to_refresh/pull_to_refresh.dart';

class LobbyPage extends StatefulWidget {
  @override
  _LobbyPageState createState() => _LobbyPageState();
}

class _LobbyPageState extends State<LobbyPage> {
  final RefreshController _refreshController =
    RefreshController(initialRefresh: false);

  void _onRefresh() async {
    await Future.delayed(const Duration(milliseconds: 1000));
  }
}
```

```

    _refreshController.refreshCompleted();
  }

  void _onLoading() async {
    await Future.delayed(const Duration(milliseconds: 1000));
    _refreshController.loadComplete();
  }

  @override
  Widget build(BuildContext context) {
    return Stack(
      alignment: Alignment.bottomCenter,
      children: [
        SmartRefresher(
          enablePullDown: true,
          controller: _refreshController,
          onRefresh: _onRefresh,
          onLoading: _onLoading,
          child: ListView.builder(
            scrollDirection: Axis.vertical,
            padding: const EdgeInsets.only(
              bottom: 80,
              left: 20,
              right: 20,
            ),
            itemBuilder: (lc, index) {
              if (index == 0) {
                return buildScheduleCard();
              }

              return buildRoomCard(rooms[index - 1], index);
            },
            itemCount: rooms.length + 1,
          ),
        ),
        buildGradientContainer(),
        buildStartRoomButton(),
      ],
    );
  }

  Widget buildScheduleCard() {
    return Container(
      margin: const EdgeInsets.symmetric(
        vertical: 10,
      ),
    ),
  }

```

```

        child: const ScheduleCard(),
      );
    }

Widget buildRoomCard(Room room, int index) {
  return GestureDetector(
    onTap: () {
      enterRoom(room);
    },
    child: Container(
      margin: const EdgeInsets.symmetric(
        vertical: 10,
      ),
      child: RoomCard(
        room: room,
        index: index,
      ),
    ),
  );
}

Widget buildGradientContainer() {
  return Container(
    height: 50,
    decoration: BoxDecoration(
      gradient: LinearGradient(
        begin: Alignment.topCenter,
        end: Alignment.bottomCenter,
        colors: [
          Style.LightBrown.withOpacity(0.2),
          Style.LightBrown,
        ],
      ),
    ),
  );
}

Widget buildStartRoomButton() {
  return Container(
    margin: const EdgeInsets.only(bottom: 20),
    child: ElevatedButton(
      style: ButtonStyle(
        minimumSize: MaterialStateProperty.all<Size>(const Size(0, 0)),
        backgroundColor: MaterialStateProperty.resolveWith(
          (states) {
            if (states.contains(MaterialState.disabled)) {
              return Style.AccentBrown;
            }
          },
        ),
      ),
    ),
  );
}

```

```

        }
        return Style.AccentBlue;
    },
),
shape: MaterialStateProperty.all<OutlinedBorder>(
    RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
    ),
),
padding: MaterialStateProperty.all<EdgeInsets>(
    const EdgeInsets.symmetric(
        vertical: 10,
        horizontal: 25,
    ),
),
elevation: MaterialStateProperty.all<double>(0.5),
),
onPressed: () {
    showBottomSheet();
},
child: const Text(
    '+ Start a room ',
    style: TextStyle(
        fontSize: 20,
        color: Colors.white,
    ),
),
),
),
);
}

enterRoom(Room room) {
    showModalBottomSheet(
        isScrollControlled: true,
        context: context,
        builder: (rc) {
            return RoomPage(
                room: room,
            );
        },
    );
}

showBottomSheet() {
    showModalBottomSheet(
        context: context,

```



```

        shape: const RoundedRectangleBorder(
          borderRadius: BorderRadius.only(
            topLeft: Radius.circular(15),
            topRight: Radius.circular(15),
          )),
        builder: (context) {
          return Wrap(
            children: [
              LobbyBottomSheet(
                onTap: () {
                  Navigator.pop(context);

                  enterRoom(
                    Room(
                      title: '${myProfile.name}'s Room',
                      users: [myProfile],
                      speakerCount: 1,
                    ),
                  );
                },
              ),
            ],
          );
        },
      );
    }
  }
}

```

```
import '../..app_lib.dart';
```

```

class FollowerPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
      padding: const EdgeInsets.only(
        left: 20,
        right: 20,
        top: 10,
      ),
      child: Column(
        children: [
          buildAvailableChatTitle(),
          buildAvailableChatList(context),
        ],
      ),
    ),
  },
}

```

```

    );
}

Widget buildAvailableChatTitle() {
  return Row(
    crossAxisAlignment: CrossAxisAlignment.end,
    children: [
      const Text(
        'AVAILABLE TO CHAT',
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
          fontSize: 15,
        ),
      ),
      Expanded(
        child: Container(
          height: 1,
          color: Colors.white,
        ),
      ),
    ],
  );
}

Widget buildAvailableChatList(BuildContext context) {
  return ListView.builder(
    shrinkWrap: true,
    physics: ScrollPhysics(),
    itemBuilder: (lc, index) {
      return Padding(
        padding: const EdgeInsets.all(8.0),
        child: FollowerItem(
          index: index,
          user: users[index],
          onProfileTap: () {
            History.pushPage(
              context,
              ProfilePage(
                profile: users[index],
              ),
            );
          },
          onRoomButtonTap: () {
            showModalBottomSheet(
              context: context,

```

```

        isScrollControlled: true,
        builder: (context) {
          return RoomPage(
            room: Room(
              title: '${myProfile.name}\''s Room',
              users: [myProfile, users[index]],
              speakerCount: 1,
            ),
          );
        },
      );
    },
  ),
);
},
itemCount: users.length,
);
}
}

```