

**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  
**Palugulla Meghana**  
**40110891**



**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](http://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 **Palugulla Meghana (40110891)** who carried out the project entitled “**CLUB HOUSE CLONE**” under my supervision from Aug 2022 to Oct 2022.

Internal Guide

Ms.Manju C Nair

Head of the Department

---

Submitted for Viva voce Examination held on \_\_\_\_\_

Internal Examiner

External Examiner

## **DECLARATION**

I \_Palugulla Meghana 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      |
| <b>1</b>   | <b>INTRODUCTION</b>  |         |
|            | 1.1 Introduction of club house                               | 10      |
| <b>2</b>   | <b>OVERVIEW OF THE APPLICATION</b>                           |         |
|            | 2.1 App specification  |         |
|            | 2.1.1 Rooms  | 11      |
|            | 2.1.2 Events   | 12      |
|            | 2.1.3 Clubs  | 12      |
| <b>3</b>   | <b>USED TECHNOLOGIES</b>                                     |         |
|            | 3.1 Flutter framework  | 13      |
|            | 3.2 Dart language  | 14,15   |
| <b>4</b>   | <b>SYSTEM ANALYSIS AND IMPLEMENTATION OF THE APPLICATION</b> |         |
|            | 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      |
| <b>5</b>   | <b>IMPLEMENTATION OF THE APPLICATION</b>                     |         |



|          |                                |           |
|----------|--------------------------------|-----------|
| 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        |
| <b>6</b> | <b>SUMMARY AND CONCLUSIONS</b> | <b>31</b> |
| 6.1      | Screenshots                    | 32-34     |
| 6.2      | References                     | 35        |

# **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.

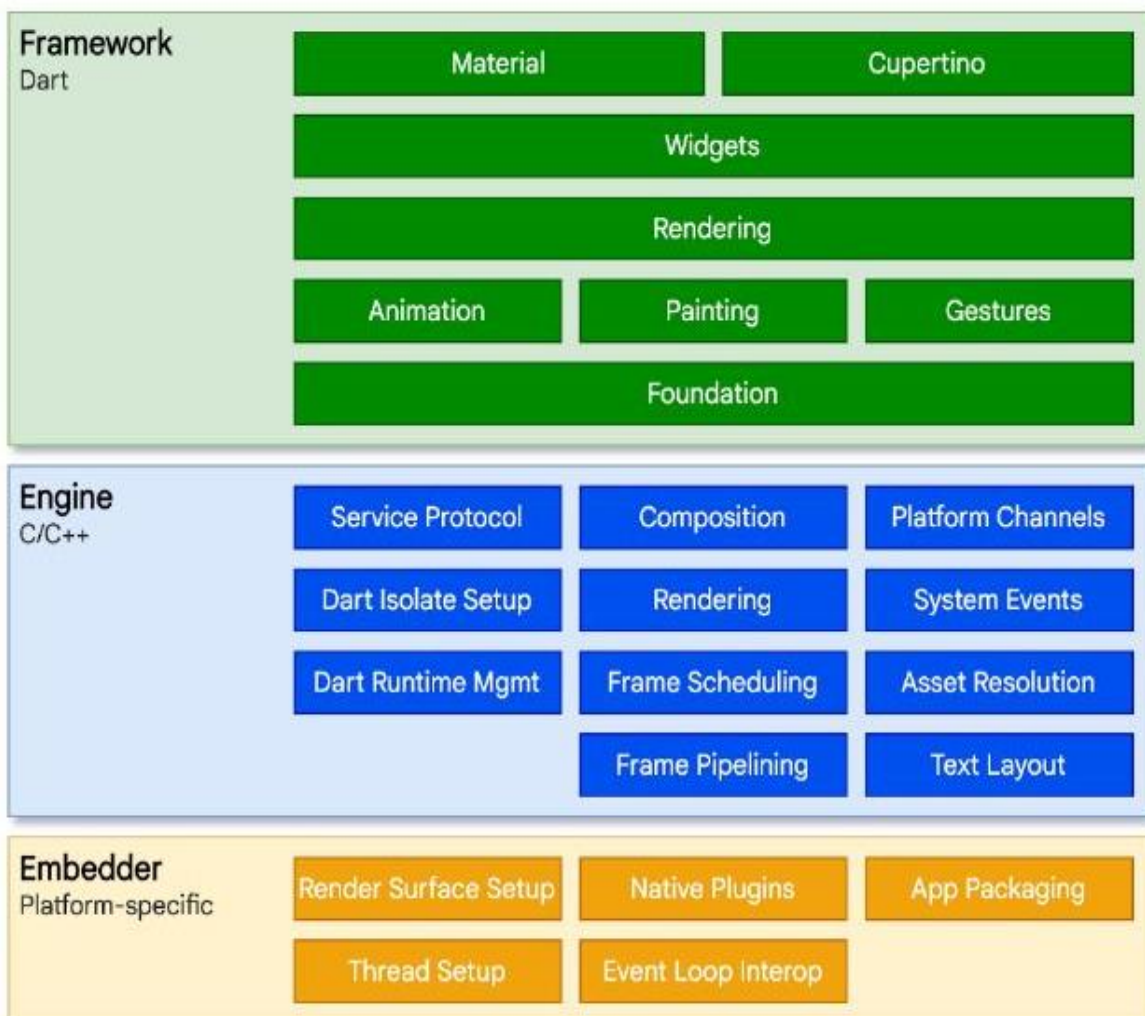
#### **Pros Of Flutter:**

- 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>

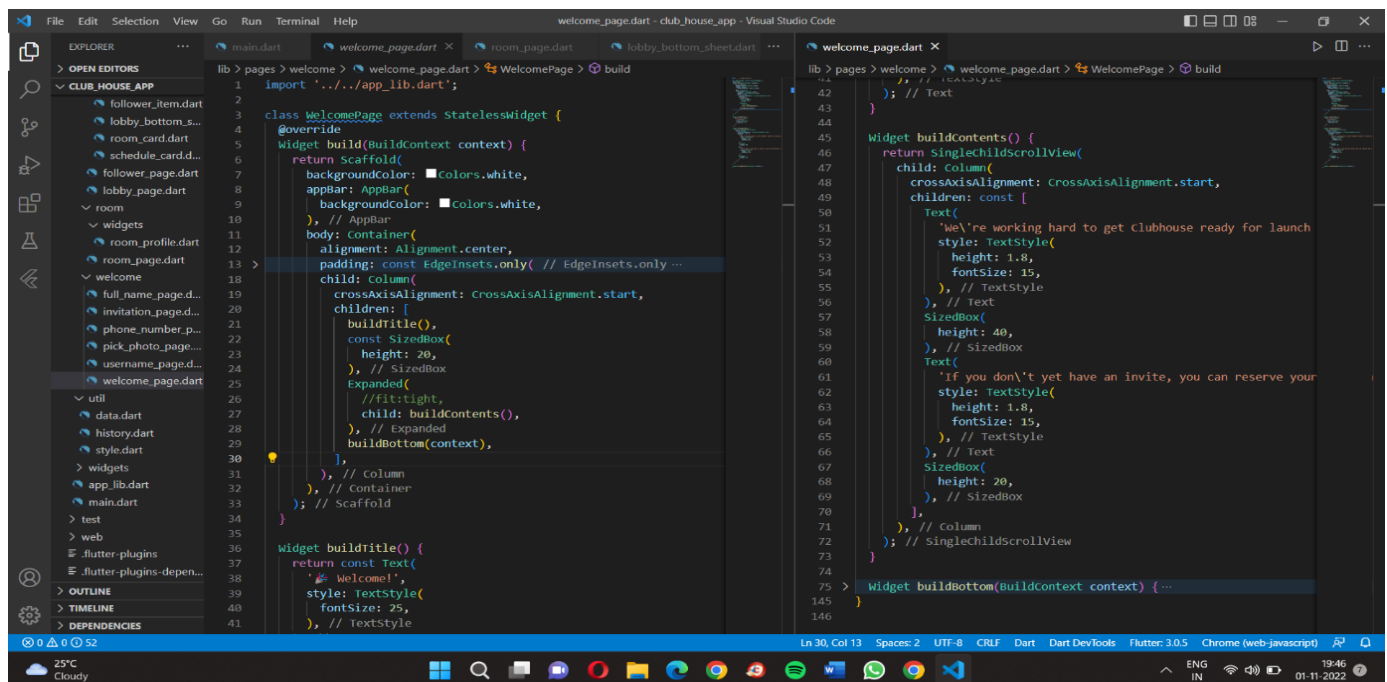
Check Installations:

- 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:

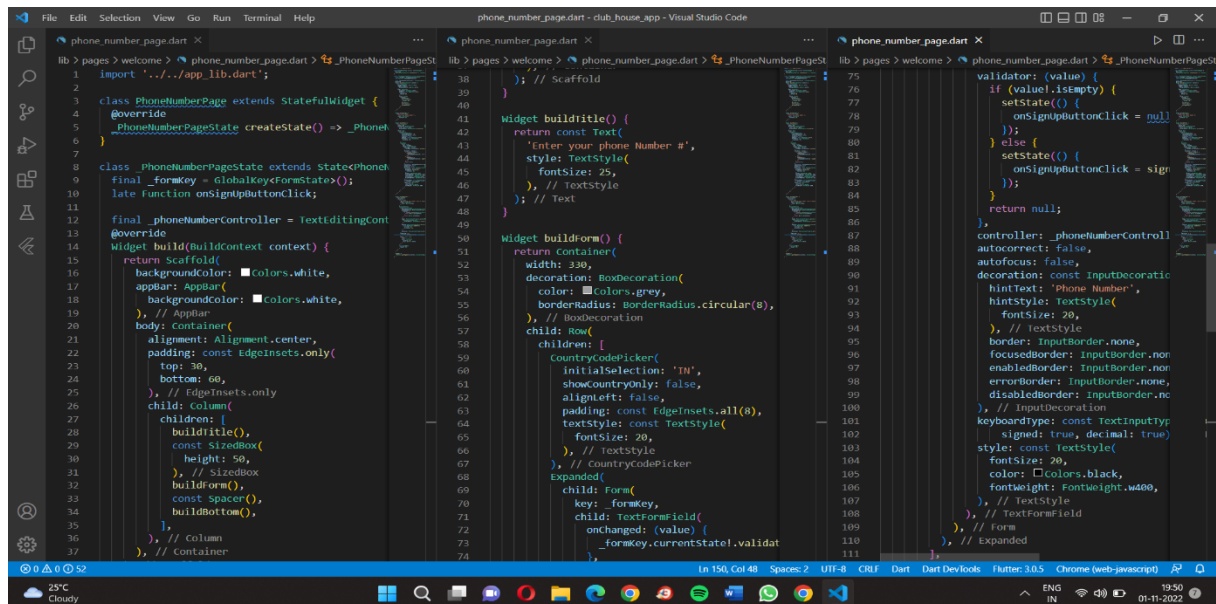
- Import Material Package to use the Built in Functionality
- Create a file welcomepage.dart And create a welcomepage class, It inherits The Properties of StatelessWidget .
- StatelessWidget: It is a static widget , which does not depend on Data or Behaviour change in runtime.
- StatelessWidget has build() 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        body: Container(
12          alignment: Alignment.center,
13          padding: const EdgeInsets.only( // EdgeInsets.only ...
14            child: Column(
15              crossAxisAlignment: CrossAxisAlignment.start,
16              children: [
17                buildTitle(),
18                const SizedBox(
19                  height: 20,
20                ), // SizedBox
21                Expanded(
22                  //fit:tight,
23                  child: buildContents(),
24                ), // Expanded
25                buildBottom(context),
26              ], // Column
27            ), // Container
28          ), // Scaffold
29        );
30      }
31
32      widget buildTitle() {
33        return const Text(
34          'Welcome!',
35          style: TextStyle(
36            fontSize: 25,
37          ), // TextStyle
38        );
39      }
40
41      widget buildContents() {
42        return SingleChildScrollView(
43          child: Column(
44            crossAxisAlignment: CrossAxisAlignment.start,
45            children: const [
46              Text(
47                'We\'re working hard to get Clubhouse ready for launch',
48                style: TextStyle(
49                  height: 1.8,
50                  fontSize: 15,
51                ), // TextStyle
52              ), // Text
53              SizedBox(
54                height: 40,
55              ), // SizedBox
56              Text(
57                'If you don\'t yet have an invite, you can reserve your',
58                style: TextStyle(
59                  height: 1.8,
60                  fontSize: 15,
61                ), // TextStyle
62              ), // Text
63              SizedBox(
64                height: 20,
65              ), // SizedBox
66            ], // Column
67          ); // SingleChildScrollView
68        );
69      }
70
71      widget buildBottom(BuildContext context) {
72        return const Text(
73          'Join the waitlist',
74          style: TextStyle(
75            color: Colors.blue,
76            fontWeight: FontWeight.bold,
77            fontSize: 18,
78          ), // TextStyle
79        );
80      }
81    );
82  }
83}
```

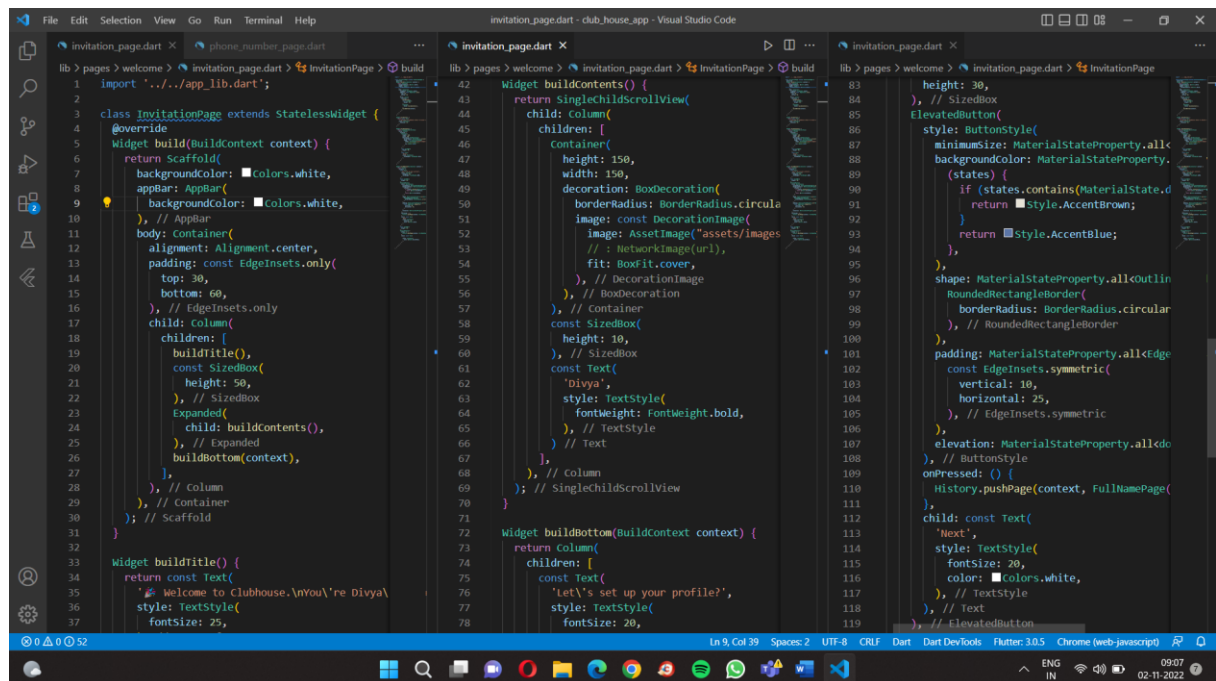
## 5.2 Phone Number Page:

- Create a file `phonenumber.dart` and Define a class `phoneumber` extends `StatefulWidget` class
- `StatefulWidget`: 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



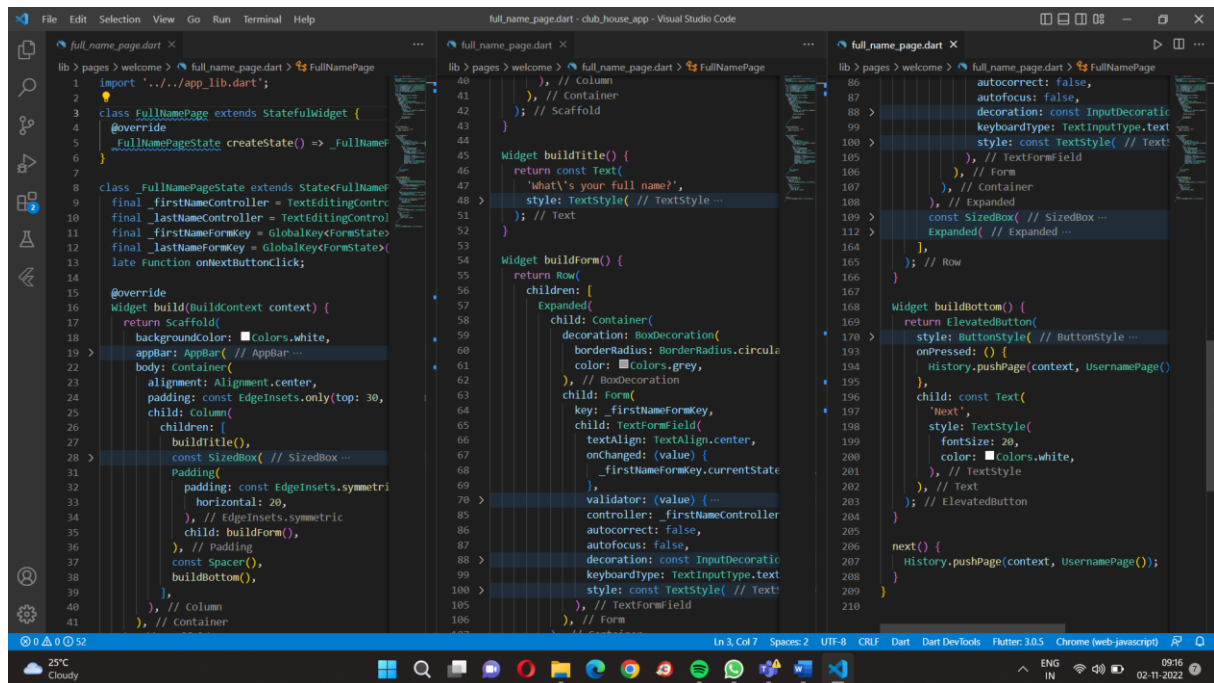
### 5.3 Invitation Page:

- 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.



## 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 userNamePage.dart And create a class userNamePage, It inherits The Properties of StatefulWidget.
- 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.

```
lib > pages > welcome > username_page.dart > UsernamePage
1 import '../app_lib.dart';
2
3 class UsernamePage extends StatefulWidget {
4   @override
5   UsernamePageState createState() => _UsernamePageState
6 }
7
8 class _UsernamePageState extends State<UsernamePage> {
9   final _usernameController = TextEditingController();
10  final _usernameFormKey = GlobalKey<FormState>();
11
12  late Function onNextButtonClick;
13
14  @override
15  Widget build(BuildContext context) {
16    return Scaffold(
17      backgroundColor: Colors.white,
18      appBar: AppBar( // AppBar ...
19        alignment: Alignment.center,
20        padding: const EdgeInsets.only( // EdgeInsets ...
21          child: Column(
22            children: [
23              buildTitle(),
24              const SizedBox( // SizedBox ...
25                child: buildForm(),
26              ),
27            ], // Column
28          ), // Container
29        ); // Scaffold
30
31    widget buildTitle() {
32      return const Text(
33        'Pick a username',
34      );
35    }
36  }
37
38  widget buildForm() {
39    return Container(
40      width: 330,
41      decoration: BoxDecoration( // BoxDecoration ...
42        child: Form(
43          key: _usernameFormKey,
44          child: TextFormField(
45            textAlign: TextAlign.center,
46            autofocus: false,
47            _usernameFormKey.currentState!.validate(),
48          ),
49        ),
50      ),
51    );
52  }
53
54  widget buildBottom() {
55    return ElevatedButton(
56      style: ButtonStyle( // ButtonStyle ...
57        onPressed: () {
58          History.pushPage(context, PickPhotoPage(
59            ),
60          );
61        },
62        child: const Text(
63          'Next',
64          style: TextStyle(
65            fontSize: 20,
66            color: Colors.white,
67          ), // TextStyle
68        ), // ElevatedButton
69      );
70    }
71
72    next() {
73      History.pushPageUntil(context, PickPhotoPage(
74        ),
75      );
76    }
77  }
78
79  widget buildTitle() {
80    return const Text(
81      'Pick a username',
82    );
83  }
84
85  widget buildForm() {
86    return Container(
87      width: 330,
88      decoration: BoxDecoration( // BoxDecoration ...
89        child: Form(
90          key: _usernameFormKey,
91          child: TextFormField(
92            textAlign: TextAlign.center,
93            autofocus: false,
94            _usernameFormKey.currentState!.validate(),
95          ),
96        ),
97      ),
98    );
99  }
100
101  widget buildBottom() {
102    return ElevatedButton(
103      style: ButtonStyle( // ButtonStyle ...
104        onPressed: () {
105          History.pushPage(context, PickPhotoPage(
106            ),
107          );
108        },
109        child: const Text(
110          'Next',
111          style: TextStyle(
112            fontSize: 20,
113            color: Colors.white,
114          ), // TextStyle
115        ), // ElevatedButton
116      );
117    }
118
119    next() {
120      History.pushPageUntil(context, PickPhotoPage(
121        ),
122      );
123    }
124  }
125
126  widget buildTitle() {
127    return const Text(
128      'Pick a username',
129    );
130  }
131
132  widget buildForm() {
133    return Container(
134      width: 330,
135      decoration: BoxDecoration( // BoxDecoration ...
136        child: Form(
137          key: _usernameFormKey,
138          child: TextFormField(
139            textAlign: TextAlign.center,
140            autofocus: false,
141            _usernameFormKey.currentState!.validate(),
142          ),
143        ),
144      ),
145    );
146  }
147
148  widget buildBottom() {
149    return ElevatedButton(
150      style: ButtonStyle( // ButtonStyle ...
151        onPressed: () {
152          History.pushPage(context, PickPhotoPage(
153            ),
154          );
155        },
156        child: const Text(
157          'Next',
158          style: TextStyle(
159            fontSize: 20,
160            color: Colors.white,
161          ), // TextStyle
162        ), // ElevatedButton
163      );
164    }
165
166    next() {
167      History.pushPageUntil(context, PickPhotoPage(
168        ),
169      );
170    }
171  }
172
173  widget buildTitle() {
174    return const Text(
175      'Pick a username',
176    );
177  }
178
179  widget buildForm() {
180    return Container(
181      width: 330,
182      decoration: BoxDecoration( // BoxDecoration ...
183        child: Form(
184          key: _usernameFormKey,
185          child: TextFormField(
186            textAlign: TextAlign.center,
187            autofocus: false,
188            _usernameFormKey.currentState!.validate(),
189          ),
190        ),
191      ),
192    );
193  }
194
195  widget buildBottom() {
196    return ElevatedButton(
197      style: ButtonStyle( // ButtonStyle ...
198        onPressed: () {
199          History.pushPage(context, PickPhotoPage(
200            ),
201          );
202        },
203        child: const Text(
204          'Next',
205          style: TextStyle(
206            fontSize: 20,
207            color: Colors.white,
208          ), // TextStyle
209        ), // ElevatedButton
210      );
211    }
212
213    next() {
214      History.pushPageUntil(context, PickPhotoPage(
215        ),
216      );
217    }
218  }
219
220  widget buildTitle() {
221    return const Text(
222      'Pick a username',
223    );
224  }
225
226  widget buildForm() {
227    return Container(
228      width: 330,
229      decoration: BoxDecoration( // BoxDecoration ...
230        child: Form(
231          key: _usernameFormKey,
232          child: TextFormField(
233            textAlign: TextAlign.center,
234            autofocus: false,
235            _usernameFormKey.currentState!.validate(),
236          ),
237        ),
238      ),
239    );
240  }
241
242  widget buildBottom() {
243    return ElevatedButton(
244      style: ButtonStyle( // ButtonStyle ...
245        onPressed: () {
246          History.pushPage(context, PickPhotoPage(
247            ),
248          );
249        },
250        child: const Text(
251          'Next',
252          style: TextStyle(
253            fontSize: 20,
254            color: Colors.white,
255          ), // TextStyle
256        ), // ElevatedButton
257      );
258    }
259
260    next() {
261      History.pushPageUntil(context, PickPhotoPage(
262        ),
263      );
264    }
265  }
266
267  widget buildTitle() {
268    return const Text(
269      'Pick a username',
270    );
271  }
272
273  widget buildForm() {
274    return Container(
275      width: 330,
276      decoration: BoxDecoration( // BoxDecoration ...
277        child: Form(
278          key: _usernameFormKey,
279          child: TextFormField(
280            textAlign: TextAlign.center,
281            autofocus: false,
282            _usernameFormKey.currentState!.validate(),
283          ),
284        ),
285      ),
286    );
287  }
288
289  widget buildBottom() {
290    return ElevatedButton(
291      style: ButtonStyle( // ButtonStyle ...
292        onPressed: () {
293          History.pushPage(context, PickPhotoPage(
294            ),
295          );
296        },
297        child: const Text(
298          'Next',
299          style: TextStyle(
300            fontSize: 20,
301            color: Colors.white,
302          ), // TextStyle
303        ), // ElevatedButton
304      );
305    }
306
307    next() {
308      History.pushPageUntil(context, PickPhotoPage(
309        ),
310      );
311    }
312  }
313
314  widget buildTitle() {
315    return const Text(
316      'Pick a username',
317    );
318  }
319
320  widget buildForm() {
321    return Container(
322      width: 330,
323      decoration: BoxDecoration( // BoxDecoration ...
324        child: Form(
325          key: _usernameFormKey,
326          child: TextFormField(
327            textAlign: TextAlign.center,
328            autofocus: false,
329            _usernameFormKey.currentState!.validate(),
330          ),
331        ),
332      ),
333    );
334  }
335
336  widget buildBottom() {
337    return ElevatedButton(
338      style: ButtonStyle( // ButtonStyle ...
339        onPressed: () {
340          History.pushPage(context, PickPhotoPage(
341            ),
342          );
343        },
344        child: const Text(
345          'Next',
346          style: TextStyle(
347            fontSize: 20,
348            color: Colors.white,
349          ), // TextStyle
350        ), // ElevatedButton
351      );
352    }
353
354    next() {
355      History.pushPageUntil(context, PickPhotoPage(
356        ),
357      );
358    }
359  }
360
361  widget buildTitle() {
362    return const Text(
363      'Pick a username',
364    );
365  }
366
367  widget buildForm() {
368    return Container(
369      width: 330,
370      decoration: BoxDecoration( // BoxDecoration ...
371        child: Form(
372          key: _usernameFormKey,
373          child: TextFormField(
374            textAlign: TextAlign.center,
375            autofocus: false,
376            _usernameFormKey.currentState!.validate(),
377          ),
378        ),
379      ),
380    );
381  }
382
383  widget buildBottom() {
384    return ElevatedButton(
385      style: ButtonStyle( // ButtonStyle ...
386        onPressed: () {
387          History.pushPage(context, PickPhotoPage(
388            ),
389          );
390        },
391        child: const Text(
392          'Next',
393          style: TextStyle(
394            fontSize: 20,
395            color: Colors.white,
396          ), // TextStyle
397        ), // ElevatedButton
398      );
399    }
400
401    next() {
402      History.pushPageUntil(context, PickPhotoPage(
403        ),
404      );
405    }
406  }
407
408  widget buildTitle() {
409    return const Text(
410      'Pick a username',
411    );
412  }
413
414  widget buildForm() {
415    return Container(
416      width: 330,
417      decoration: BoxDecoration( // BoxDecoration ...
418        child: Form(
419          key: _usernameFormKey,
420          child: TextFormField(
421            textAlign: TextAlign.center,
422            autofocus: false,
423            _usernameFormKey.currentState!.validate(),
424          ),
425        ),
426      ),
427    );
428  }
429
430  widget buildBottom() {
431    return ElevatedButton(
432      style: ButtonStyle( // ButtonStyle ...
433        onPressed: () {
434          History.pushPage(context, PickPhotoPage(
435            ),
436          );
437        },
438        child: const Text(
439          'Next',
440          style: TextStyle(
441            fontSize: 20,
442            color: Colors.white,
443          ), // TextStyle
444        ), // ElevatedButton
445      );
446    }
447
448    next() {
449      History.pushPageUntil(context, PickPhotoPage(
450        ),
451      );
452    }
453  }
454
455  widget buildTitle() {
456    return const Text(
457      'Pick a username',
458    );
459  }
460
461  widget buildForm() {
462    return Container(
463      width: 330,
464      decoration: BoxDecoration( // BoxDecoration ...
465        child: Form(
466          key: _usernameFormKey,
467          child: TextFormField(
468            textAlign: TextAlign.center,
469            autofocus: false,
470            _usernameFormKey.currentState!.validate(),
471          ),
472        ),
473      ),
474    );
475  }
476
477  widget buildBottom() {
478    return ElevatedButton(
479      style: ButtonStyle( // ButtonStyle ...
480        onPressed: () {
481          History.pushPage(context, PickPhotoPage(
482            ),
483          );
484        },
485        child: const Text(
486          'Next',
487          style: TextStyle(
488            fontSize: 20,
489            color: Colors.white,
490          ), // TextStyle
491        ), // ElevatedButton
492      );
493    }
494
495    next() {
496      History.pushPageUntil(context, PickPhotoPage(
497        ),
498      );
499    }
500  }
501
502  widget buildTitle() {
503    return const Text(
504      'Pick a username',
505    );
506  }
507
508  widget buildForm() {
509    return Container(
510      width: 330,
511      decoration: BoxDecoration( // BoxDecoration ...
512        child: Form(
513          key: _usernameFormKey,
514          child: TextFormField(
515            textAlign: TextAlign.center,
516            autofocus: false,
517            _usernameFormKey.currentState!.validate(),
518          ),
519        ),
520      ),
521    );
522  }
523
524  widget buildBottom() {
525    return ElevatedButton(
526      style: ButtonStyle( // ButtonStyle ...
527        onPressed: () {
528          History.pushPage(context, PickPhotoPage(
529            ),
530          );
531        },
532        child: const Text(
533          'Next',
534          style: TextStyle(
535            fontSize: 20,
536            color: Colors.white,
537          ), // TextStyle
538        ), // ElevatedButton
539      );
540    }
541
542    next() {
543      History.pushPageUntil(context, PickPhotoPage(
544        ),
545      );
546    }
547  }
548
549  widget buildTitle() {
550    return const Text(
551      'Pick a username',
552    );
553  }
554
555  widget buildForm() {
556    return Container(
557      width: 330,
558      decoration: BoxDecoration( // BoxDecoration ...
559        child: Form(
560          key: _usernameFormKey,
561          child: TextFormField(
562            textAlign: TextAlign.center,
563            autofocus: false,
564            _usernameFormKey.currentState!.validate(),
565          ),
566        ),
567      ),
568    );
569  }
570
571  widget buildBottom() {
572    return ElevatedButton(
573      style: ButtonStyle( // ButtonStyle ...
574        onPressed: () {
575          History.pushPage(context, PickPhotoPage(
576            ),
577          );
578        },
579        child: const Text(
580          'Next',
581          style: TextStyle(
582            fontSize: 20,
583            color: Colors.white,
584          ), // TextStyle
585        ), // ElevatedButton
586      );
587    }
588
589    next() {
590      History.pushPageUntil(context, PickPhotoPage(
591        ),
592      );
593    }
594  }
595
596  widget buildTitle() {
597    return const Text(
598      'Pick a username',
599    );
600  }
601
602  widget buildForm() {
603    return Container(
604      width: 330,
605      decoration: BoxDecoration( // BoxDecoration ...
606        child: Form(
607          key: _usernameFormKey,
608          child: TextFormField(
609            textAlign: TextAlign.center,
610            autofocus: false,
611            _usernameFormKey.currentState!.validate(),
612          ),
613        ),
614      ),
615    );
616  }
617
618  widget buildBottom() {
619    return ElevatedButton(
620      style: ButtonStyle( // ButtonStyle ...
621        onPressed: () {
622          History.pushPage(context, PickPhotoPage(
623            ),
624          );
625        },
626        child: const Text(
627          'Next',
628          style: TextStyle(
629            fontSize: 20,
630            color: Colors.white,
631          ), // TextStyle
632        ), // ElevatedButton
633      );
634    }
635
636    next() {
637      History.pushPageUntil(context, PickPhotoPage(
638        ),
639      );
640    }
641  }
642
643  widget buildTitle() {
644    return const Text(
645      'Pick a username',
646    );
647  }
648
649  widget buildForm() {
650    return Container(
651      width: 330,
652      decoration: BoxDecoration( // BoxDecoration ...
653        child: Form(
654          key: _usernameFormKey,
655          child: TextFormField(
656            textAlign: TextAlign.center,
657            autofocus: false,
658            _usernameFormKey.currentState!.validate(),
659          ),
660        ),
661      ),
662    );
663  }
664
665  widget buildBottom() {
666    return ElevatedButton(
667      style: ButtonStyle( // ButtonStyle ...
668        onPressed: () {
669          History.pushPage(context, PickPhotoPage(
670            ),
671          );
672        },
673        child: const Text(
674          'Next',
675          style: TextStyle(
676            fontSize: 20,
677            color: Colors.white,
678          ), // TextStyle
679        ), // ElevatedButton
680      );
681    }
682
683    next() {
684      History.pushPageUntil(context, PickPhotoPage(
685        ),
686      );
687    }
688  }
689
690  widget buildTitle() {
691    return const Text(
692      'Pick a username',
693    );
694  }
695
696  widget buildForm() {
697    return Container(
698      width: 330,
699      decoration: BoxDecoration( // BoxDecoration ...
700        child: Form(
701          key: _usernameFormKey,
702          child: TextFormField(
703            textAlign: TextAlign.center,
704            autofocus: false,
705            _usernameFormKey.currentState!.validate(),
706          ),
707        ),
708      ),
709    );
710  }
711
712  widget buildBottom() {
713    return ElevatedButton(
714      style: ButtonStyle( // ButtonStyle ...
715        onPressed: () {
716          History.pushPage(context, PickPhotoPage(
717            ),
718          );
719        },
720        child: const Text(
721          'Next',
722          style: TextStyle(
723            fontSize: 20,
724            color: Colors.white,
725          ), // TextStyle
726        ), // ElevatedButton
727      );
728    }
729
730    next() {
731      History.pushPageUntil(context, PickPhotoPage(
732        ),
733      );
734    }
735  }
736
737  widget buildTitle() {
738    return const Text(
739      'Pick a username',
740    );
741  }
742
743  widget buildForm() {
744    return Container(
745      width: 330,
746      decoration: BoxDecoration( // BoxDecoration ...
747        child: Form(
748          key: _usernameFormKey,
749          child: TextFormField(
750            textAlign: TextAlign.center,
751            autofocus: false,
752            _usernameFormKey.currentState!.validate(),
753          ),
754        ),
755      ),
756    );
757  }
758
759  widget buildBottom() {
760    return ElevatedButton(
761      style: ButtonStyle( // ButtonStyle ...
762        onPressed: () {
763          History.pushPage(context, PickPhotoPage(
764            ),
765          );
766        },
767        child: const Text(
768          'Next',
769          style: TextStyle(
770            fontSize: 20,
771            color: Colors.white,
772          ), // TextStyle
773        ), // ElevatedButton
774      );
775    }
776
777    next() {
778      History.pushPageUntil(context, PickPhotoPage(
779        ),
780      );
781    }
782  }
783
784  widget buildTitle() {
785    return const Text(
786      'Pick a username',
787    );
788  }
789
790  widget buildForm() {
791    return Container(
792      width: 330,
793      decoration: BoxDecoration( // BoxDecoration ...
794        child: Form(
795          key: _usernameFormKey,
796          child: TextFormField(
797            textAlign: TextAlign.center,
798            autofocus: false,
799            _usernameFormKey.currentState!.validate(),
800          ),
801        ),
802      ),
803    );
804  }
805
806  widget buildBottom() {
807    return ElevatedButton(
808      style: ButtonStyle( // ButtonStyle ...
809        onPressed: () {
810          History.pushPage(context, PickPhotoPage(
811            ),
812          );
813        },
814        child: const Text(
815          'Next',
816          style: TextStyle(
817            fontSize: 20,
818            color: Colors.white,
819          ), // TextStyle
820        ), // ElevatedButton
821      );
822    }
823
824    next() {
825      History.pushPageUntil(context, PickPhotoPage(
826        ),
827      );
828    }
829  }
830
831  widget buildTitle() {
832    return const Text(
833      'Pick a username',
834    );
835  }
836
837  widget buildForm() {
838    return Container(
839      width: 330,
840      decoration: BoxDecoration( // BoxDecoration ...
841        child: Form(
842          key: _usernameFormKey,
843          child: TextFormField(
844            textAlign: TextAlign.center,
845            autofocus: false,
846            _usernameFormKey.currentState!.validate(),
847          ),
848        ),
849      ),
850    );
851  }
852
853  widget buildBottom() {
854    return ElevatedButton(
855      style: ButtonStyle( // ButtonStyle ...
856        onPressed: () {
857          History.pushPage(context, PickPhotoPage(
858            ),
859          );
860        },
861        child: const Text(
862          'Next',
863          style: TextStyle(
864            fontSize: 20,
865            color: Colors.white,
866          ), // TextStyle
867        ), // ElevatedButton
868      );
869    }
870
871    next() {
872      History.pushPageUntil(context, PickPhotoPage(
873        ),
874      );
875    }
876  }
877
878  widget buildTitle() {
879    return const Text(
880      'Pick a username',
881    );
882  }
883
884  widget buildForm() {
885    return Container(
886      width: 330,
887      decoration: BoxDecoration( // BoxDecoration ...
888        child: Form(
889          key: _usernameFormKey,
890          child: TextFormField(
891            textAlign: TextAlign.center,
892            autofocus: false,
893            _usernameFormKey.currentState!.validate(),
894          ),
895        ),
896      ),
897    );
898  }
899
900  widget buildBottom() {
901    return ElevatedButton(
902      style: ButtonStyle( // ButtonStyle ...
903        onPressed: () {
904          History.pushPage(context, PickPhotoPage(
905            ),
906          );
907        },
908        child: const Text(
909          'Next',
910          style: TextStyle(
911            fontSize: 20,
912            color: Colors.white,
913          ), // TextStyle
914        ), // ElevatedButton
915      );
916    }
917
918    next() {
919      History.pushPageUntil(context, PickPhotoPage(
920        ),
921      );
922    }
923  }
924
925  widget buildTitle() {
926    return const Text(
927      'Pick a username',
928    );
929  }
930
931  widget buildForm() {
932    return Container(
933      width: 330,
934      decoration: BoxDecoration( // BoxDecoration ...
935        child: Form(
936          key: _usernameFormKey,
937          child: TextFormField(
938            textAlign: TextAlign.center,
939            autofocus: false,
940            _usernameFormKey.currentState!.validate(),
941          ),
942        ),
943      ),
944    );
945  }
946
947  widget buildBottom() {
948    return ElevatedButton(
949      style: ButtonStyle( // ButtonStyle ...
950        onPressed: () {
951          History.pushPage(context, PickPhotoPage(
952            ),
953          );
954        },
955        child: const Text(
956          'Next',
957          style: TextStyle(
958            fontSize: 20,
959            color: Colors.white,
960          ), // TextStyle
961        ), // ElevatedButton
962      );
963    }
964
965    next() {
966      History.pushPageUntil(context, PickPhotoPage(
967        ),
968      );
969    }
970  }
971
972  widget buildTitle() {
973    return const Text(
974      'Pick a username',
975    );
976  }
977
978  widget buildForm() {
979    return Container(
980      width: 330,
981      decoration: BoxDecoration( // BoxDecoration ...
982        child: Form(
983          key: _usernameFormKey,
984          child: TextFormField(
985            textAlign: TextAlign.center,
986            autofocus: false,
987            _usernameFormKey.currentState!.validate(),
988          ),
989        ),
990      ),
991    );
992  }
993
994  widget buildBottom() {
995    return ElevatedButton(
996      style: ButtonStyle( // ButtonStyle ...
997        onPressed: () {
998          History.pushPage(context, PickPhotoPage(
999            ),
1000          );
1001        },
1002        child: const Text(
1003          'Next',
1004          style: TextStyle(
1005            fontSize: 20,
1006            color: Colors.white,
1007          ), // TextStyle
1008        ), // ElevatedButton
1009      );
1010    }
1011
1012    next() {
1013      History.pushPageUntil(context, PickPhotoPage(
1014        ),
1015      );
1016    }
1017  }
1018
1019  widget buildTitle() {
1020    return const Text(
1021      'Pick a username',
1022    );
1023  }
1024
1025  widget buildForm() {
1026    return Container(
1027      width: 330,
1028      decoration: BoxDecoration( // BoxDecoration ...
1029        child: Form(
1030          key: _usernameFormKey,
1031          child: TextFormField(
1032            textAlign: TextAlign.center,
1033            autofocus: false,
1034            _usernameFormKey.currentState!.validate(),
1035          ),
1036        ),
1037      ),
1038    );
1039  }
1040
1041  widget buildBottom() {
1042    return ElevatedButton(
1043      style: ButtonStyle( // ButtonStyle ...
1044        onPressed: () {
1045          History.pushPage(context, PickPhotoPage(
1046            ),
1047          );
1048        },
1049        child: const Text(
1050          'Next',
1051          style: TextStyle(
1052            fontSize: 20,
1053            color: Colors.white,
1054          ), // TextStyle
1055        ), // ElevatedButton
1056      );
1057    }
1058
1059    next() {
1060      History.pushPageUntil(context, PickPhotoPage(
1061        ),
1062      );
1063    }
1064  }
1065
1066  widget buildTitle() {
1067    return const Text(
1068      'Pick a username',
1069    );
1070  }
1071
1072  widget buildForm() {
1073    return Container(
1074      width: 330,
1075      decoration: BoxDecoration( // BoxDecoration ...
1076        child: Form(
1077          key: _usernameFormKey,
1078          child: TextFormField(
1079            textAlign: TextAlign.center,
1080            autofocus: false,
1081            _usernameFormKey.currentState!.validate(),
1082          ),
1083        ),
1084      ),
1085    );
1086  }
1087
1088  widget buildBottom() {
1089    return ElevatedButton(
1090      style: ButtonStyle( // ButtonStyle ...
1091        onPressed: () {
1092          History.pushPage(context, PickPhotoPage(
1093            ),
1094          );
1095        },
1096        child: const Text(
1097          'Next',
1098          style: TextStyle(
1099            fontSize: 20,
1100            color: Colors.white,
1101          ), // TextStyle
1102        ), // ElevatedButton
1103      );
1104    }
1105
1106    next() {
1107      History.pushPageUntil(context, PickPhotoPage(
1108        ),
1109      );
1110    }
1111  }
1112
1113  widget buildTitle() {
1114    return const Text(
1115      'Pick a username',
1116    );
1117  }
1118
1119  widget buildForm() {
1120    return Container(
1121      width: 330,
1122      decoration: BoxDecoration( // BoxDecoration ...
1123        child: Form(
1124          key: _usernameFormKey,
1125          child: TextFormField(
1126            textAlign: TextAlign.center,
1127            autofocus: false,
1128            _usernameFormKey.currentState!.validate(),
1129          ),
1130        ),
1131      ),
1132    );
1133  }
1134
1135  widget buildBottom() {
1136    return ElevatedButton(
1137      style: ButtonStyle( // ButtonStyle ...
1138        onPressed: () {
1139          History.pushPage(context, PickPhotoPage(
1140            ),
1141          );
1142        },
1143        child: const Text(
1144          'Next',
1145          style: TextStyle(
1146            fontSize: 20,
1147            color: Colors.white,
1148          ), // TextStyle
1149        ), // ElevatedButton
1150      );
1151    }
1152
1153    next() {
1154      History.pushPageUntil(context, PickPhotoPage(
1155        ),
1156      );
1157    }
1158  }
1159
1160  widget buildTitle() {
1161    return const Text(
1162      'Pick a username',
1163    );
1164  }
1165
1166  widget buildForm() {
1167    return Container(
1168      width: 330,
1169      decoration: BoxDecoration( // BoxDecoration ...
1170        child: Form(
1171          key: _usernameFormKey,
1172          child: TextFormField(
1173            textAlign: TextAlign.center,
1174            autofocus: false,
1175            _usernameFormKey.currentState!.validate(),
1176          ),
1177        ),
1178      ),
1179    );
1180  }
1181
1182  widget buildBottom() {
1183    return ElevatedButton(
1184      style: ButtonStyle( // ButtonStyle ...
1185        onPressed: () {
1186          History.pushPage(context, PickPhotoPage(
1187            ),
1188          );
1189        },
1190        child: const Text(
1191          'Next',
1192          style: TextStyle(
1193            fontSize: 20,
1194            color: Colors.white,
1195          ), // TextStyle
1196        ), // ElevatedButton
1197      );
1198    }
1199
1200    next() {
1201      History.pushPageUntil(context, PickPhotoPage(
1202        ),
1203      );
1204    }
1205  }
1206
1207  widget buildTitle() {
1208    return const Text(
1209      'Pick a username',
1210    );
1211  }
1212
1213  widget buildForm() {
1214    return Container(
1215      width: 330,
1216      decoration: BoxDecoration( // BoxDecoration ...
1217        child: Form(
1218          key: _usernameFormKey,
1219          child: TextFormField(
1220            textAlign: TextAlign.center,
1221            autofocus: false,
1222            _usernameFormKey.currentState!.validate(),
1223          ),
1224        ),
1225      ),
1226    );
1227  }
1228
1229  widget buildBottom() {
1230    return ElevatedButton(
1231      style: ButtonStyle( // ButtonStyle ...
1232        onPressed: () {
1233          History.pushPage(context, PickPhotoPage(
1234            ),
1235          );
1236        },
1237        child: const Text(
1238          'Next',
1239          style: TextStyle(
1240            fontSize: 20,
1241            color: Colors.white,
1242          ), // TextStyle
1243        ), // ElevatedButton
1244      );
1245    }
1246
1247    next() {
1248      History.pushPageUntil(context, PickPhotoPage(
1249        ),
1250      );
1251    }
1252  }
1253
1254  widget buildTitle() {
1255    return const Text(
1256      'Pick a username',
1257    );
1258  }
1259
1260  widget buildForm() {
1261    return Container(
1262      width: 330,
1263      decoration: BoxDecoration( // BoxDecoration ...
1264        child: Form(
1265          key: _usernameFormKey,
1266          child: TextFormField(
1267            textAlign: TextAlign.center,
1268            autofocus: false,
1269            _usernameFormKey.currentState!.validate(),
1270          ),
1271        ),
1272      ),
1273    );
1274  }
1275
1276  widget buildBottom() {
1277    return ElevatedButton(
1278      style: ButtonStyle( // ButtonStyle ...
1279        onPressed: () {
1280          History.pushPage(context, PickPhotoPage(
1281            ),
1282          );
1283        },
1284        child: const Text(
1285          'Next',
1286          style: TextStyle(
1287            fontSize: 20,
1288            color: Colors.white,
1289          ), // TextStyle
1290        ), // ElevatedButton
1291      );
1292    }
1293
1294    next() {
1295      History.pushPageUntil(context, PickPhotoPage(
1296        ),
1297      );
1298    }
1299  }
1300
1301  widget buildTitle() {
1302    return const Text(
1303      'Pick a username',
1304    );
1305  }
1306
1307  widget buildForm() {
1308    return Container(
1309      width: 330,
1310      decoration: BoxDecoration( // BoxDecoration ...
1311        child: Form(
1312          key: _usernameFormKey,
1313          child: TextFormField(
1314            textAlign: TextAlign.center,
1315            autofocus: false,
1316            _usernameFormKey.currentState!.validate(),
1317          ),
1318        ),
1319      ),
1320    );
1321  }
1322
1323  widget buildBottom() {
1324    return ElevatedButton(
1325      style: ButtonStyle( // ButtonStyle ...
1326        onPressed: () {
1327          History.pushPage(context, PickPhotoPage(
1328            ),
1329          );
1330        },
1331        child: const Text(
1332          'Next',
1333          style: TextStyle(
1334            fontSize: 20,
1335            color: Colors.white,
1336          ), // TextStyle
1337        ), // ElevatedButton
1338      );
1339    }
1340
1341    next() {
1342      History.pushPageUntil(context, PickPhotoPage(
1343        ),
1344      );
1345    }
1346  }
1347
1348  widget buildTitle() {
1349    return const Text(
1350      'Pick a username',
1351    );
1352  }
1353
1354  widget buildForm() {
1355    return Container(
1356      width: 330,
1357      decoration: BoxDecoration( // BoxDecoration ...
1358        child: Form(
1359          key: _usernameFormKey,
1360          child: TextFormField(
1361            textAlign: TextAlign.center,
1362            autofocus: false,
1363            _usernameFormKey.currentState!.validate(),
1364          ),
1365        ),
1366      ),
1367    );
1368  }
1369
1370  widget buildBottom() {
1371    return ElevatedButton(
1372      style: ButtonStyle( // ButtonStyle ...
1373        onPressed: () {
1374          History.pushPage(context, PickPhotoPage(
1375            ),
1376          );
1377        },
1378        child: const Text(
1379          'Next',
1380          style: TextStyle(
1381            fontSize: 20,
1382            color: Colors.white,
1383          ), // TextStyle
1384        ), // ElevatedButton
1385      );
1386    }
1387
1388    next() {
1389      History.pushPageUntil(context, PickPhotoPage(
1390        ),
1391      );
1392    }
1393  }
1394
1395  widget buildTitle() {
1396    return const Text(
1397      'Pick a username',
1398    );
1399  }
1400
1401  widget buildForm() {
1402    return Container(
1403      width: 330,
1404      decoration: BoxDecoration( // BoxDecoration ...
1405        child: Form(
1406          key: _usernameFormKey,
1407          child: TextFormField(
1408            textAlign: TextAlign.center,
1409            autofocus: false,
1410            _usernameFormKey.currentState!.validate(),
1411          ),
1412
```



- 

## 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

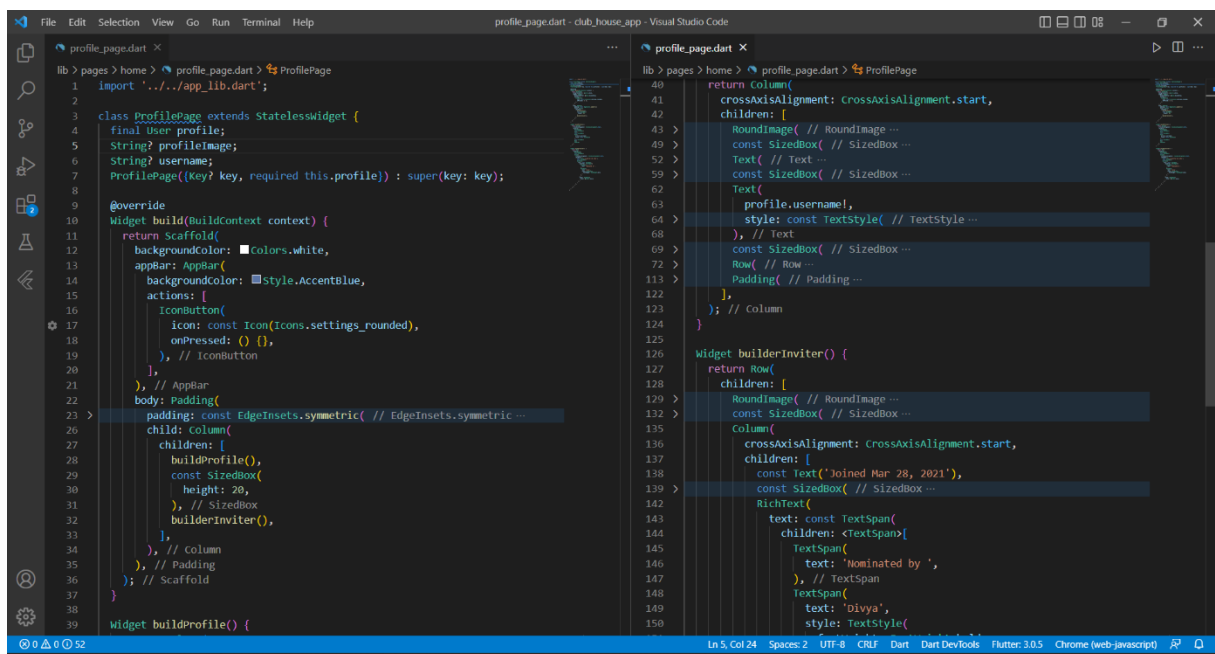
```

lib > pages > home > home_page.dart > HomePage
1  import '../app_lib.dart';
2
3  class HomePage extends StatelessWidget {
4    @override
5    widget build(BuildContext context) {
6      return Scaffold(
7        backgroundColor: Colors.white,
8        appBar: AppBar(
9          backgroundColor: Style.AccentBlue,
10         title: HomeAppBar(
11           profile: myProfile,
12           onProfileTab: () {
13             History.pushPage( User: myProfile
14               context,
15               ProfilePage Type: User package:club_house_app/util/data.dart
16               profile: myProfile,
17             ); // ProfilePage
18           },
19         ), // HomeAppBar
20       ), // AppBar
21       body: PageView(
22         children: [
23           LobbyPage(),
24           FollowerPage(),
25         ],
26       ), // PageView
27     ); // Scaffold
28   }
29 }
30

```

### 5.8.1 Profile Page

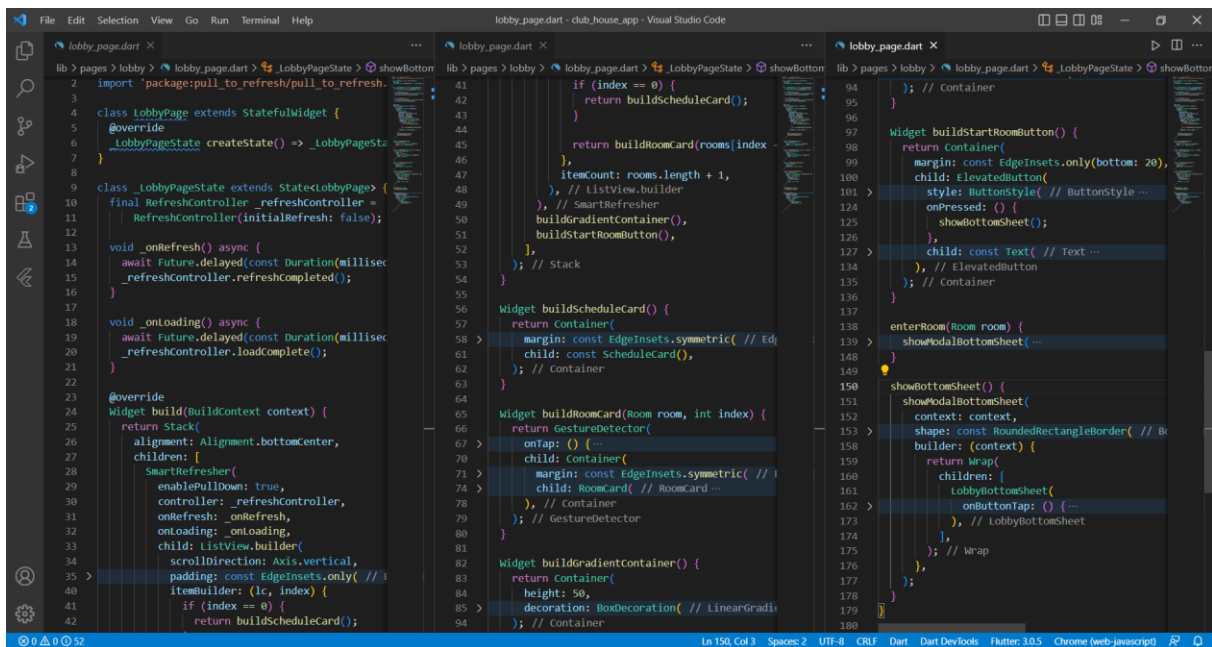
- Import Material Package to use the Built in Functionality
- Create a file ProfilePage.dart And create a class ProfilePage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body



### 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

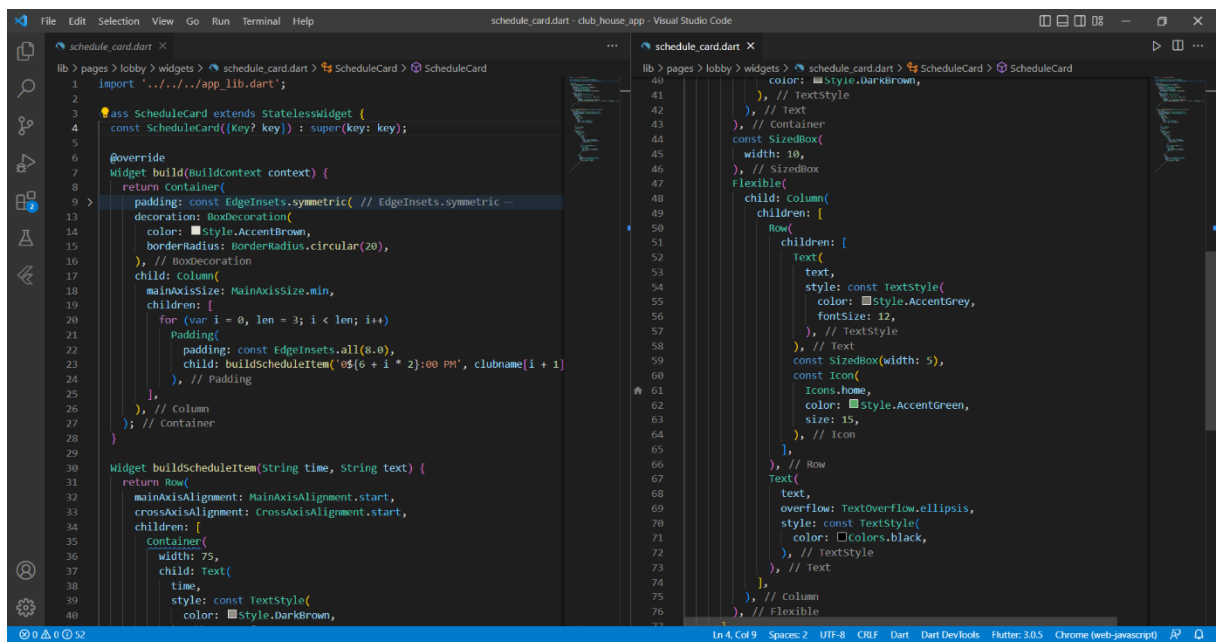
- Body consists of
  - SmartReferesher
    - ScheduleCard
    - RoomCard
    - LobbyBottomSheet
  - buildGradientContainer()-(userDefined Function)
  - buildStartRoomButton()-(userDefined Function)



### 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 Stateless widget.

- It will display the list of scheduled meetings



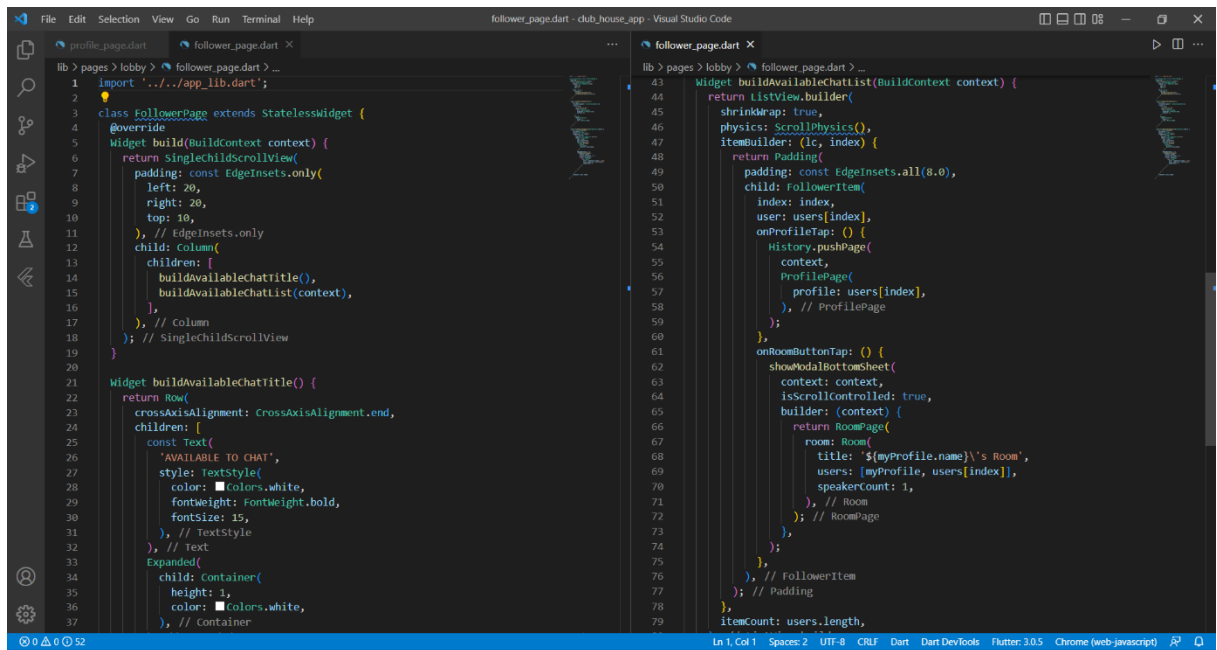
### 5.8.2.2 Room Card

- Import Material Package to use the Built in Functionality
- Create a file `RoomCard.dart` And create a class `RoomCard`, It inherits The Properties of `Stateless widget`.
- It displays the currently running meetings.
- Users can join the meetings by clicking the room card.



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

## 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

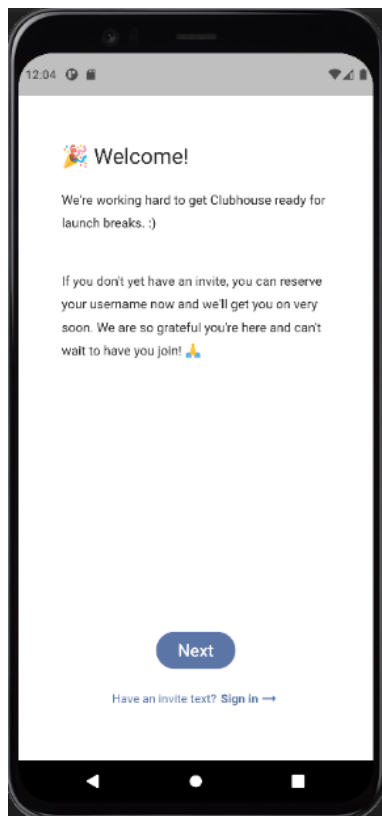


Fig 1.1

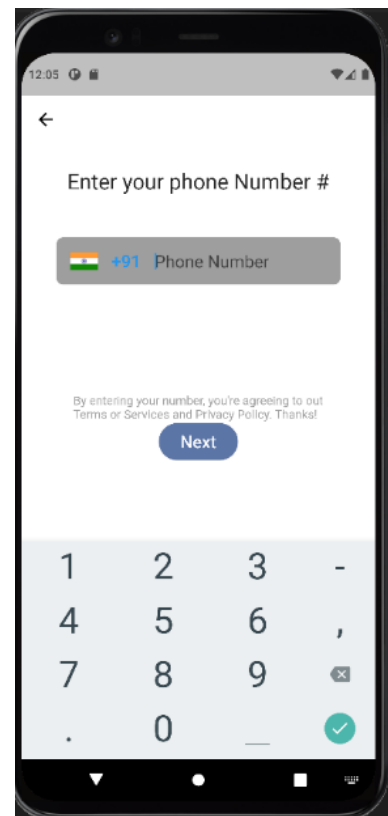


Fig 1.2



Fig 1.3



Fig 1.4



Fig 1.5



Fig 1.6



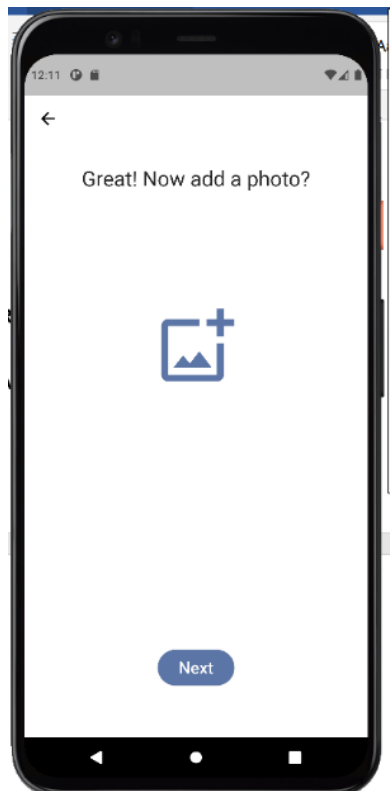


Fig 1.7

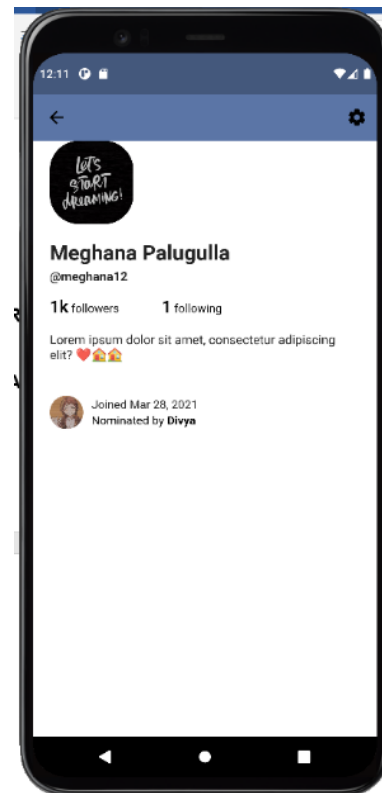


Fig 1.8

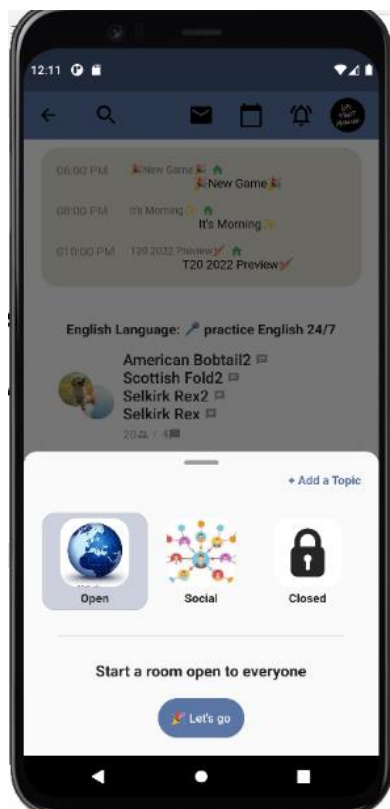


Fig 1.9

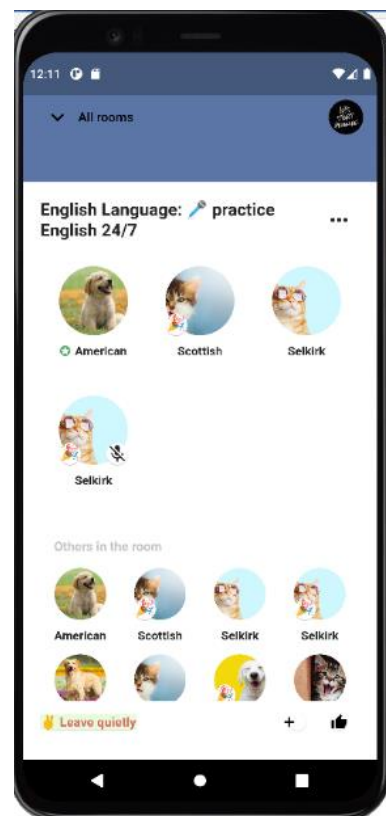


Fig 2.0

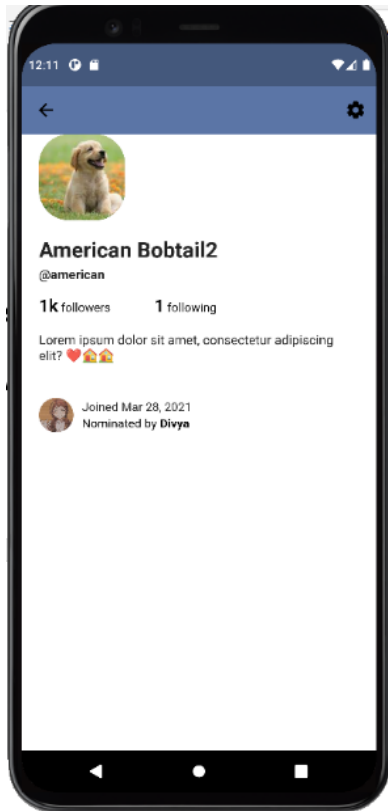


Fig 2.1

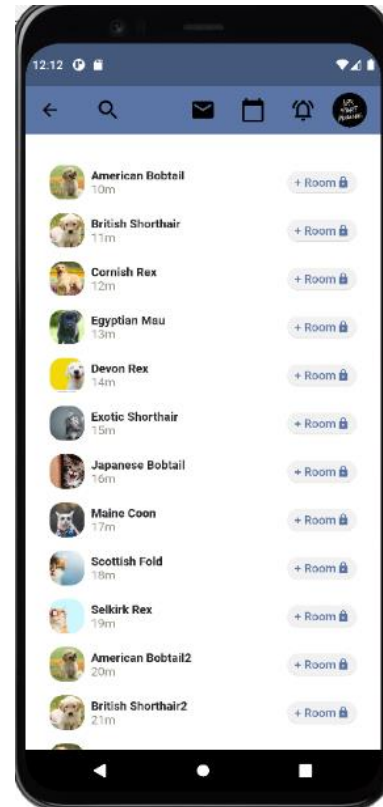


Fig 2.2

## 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](https://www.tutorialspoint.com/dart_programming/index.html)