

**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](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 **THOKALA DIVYA (40111339)** who carried out the project entitled "**CLUB HOUSE CLONE**" under my supervision from Aug 2022 to Oct 2022.

Internal Guide

Dr.Ms.Dharani

Head of the Department

---

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

Internal Examiner

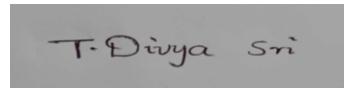
External Examiner

## DECLARATION

I DIVYA SRI THOKALA (Name/s of the individual Candidate) 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:

A rectangular box containing a handwritten signature in dark ink. The signature appears to read "T. Divya Sri".

SIGNATURE OF THE CANDIDATE

## 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.

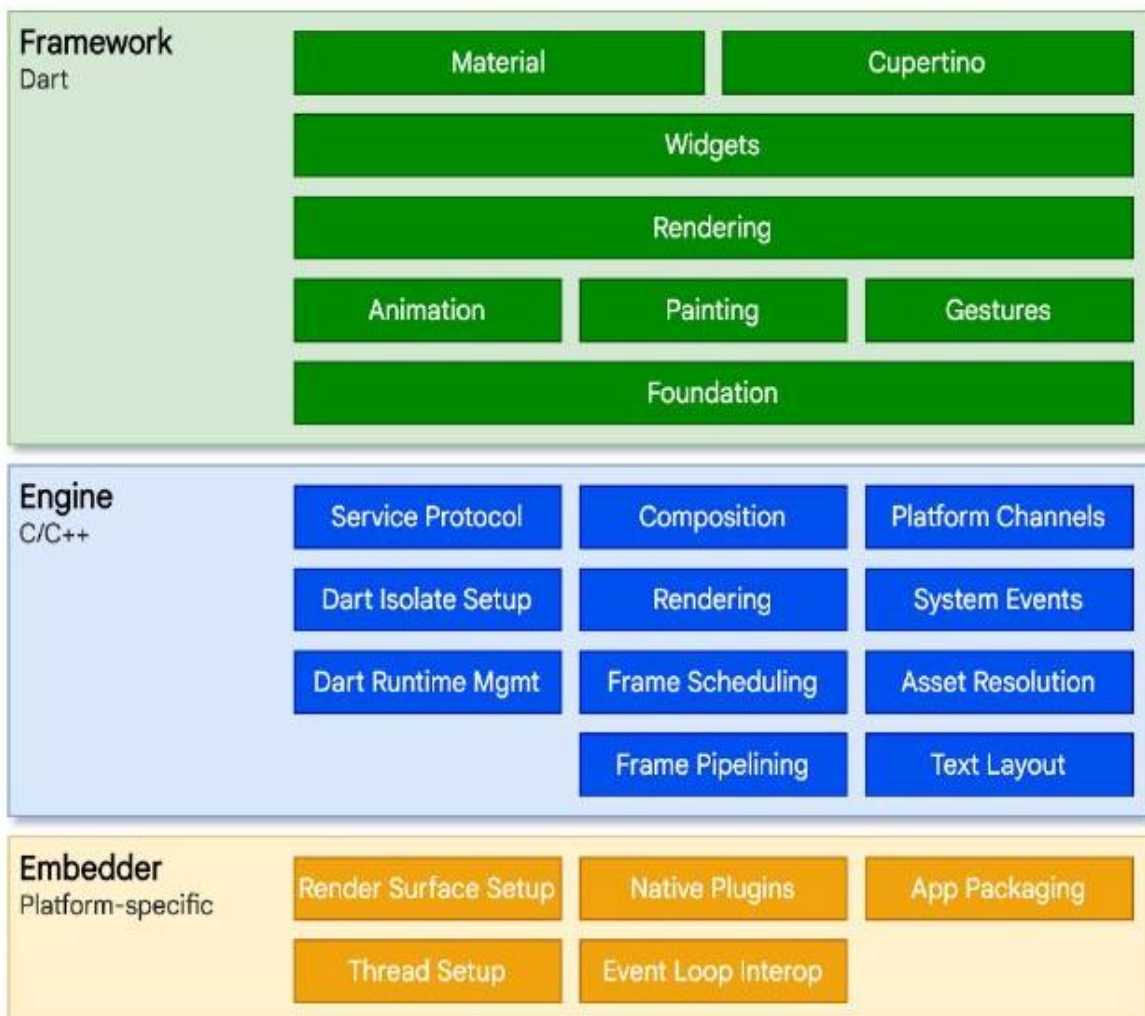
#### **Pros Of Flutter:**

- 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>

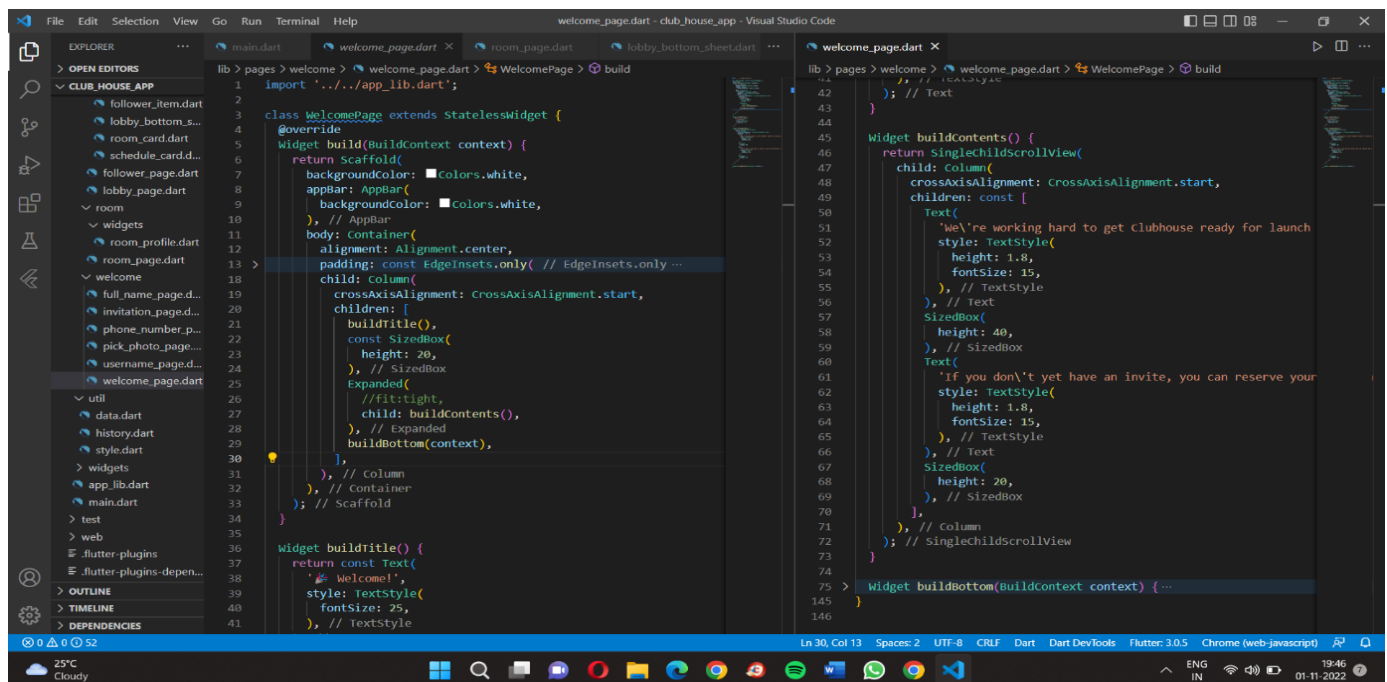
Check Installations:

- 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        // ...
73      }
74    }
75  }
76 }
```

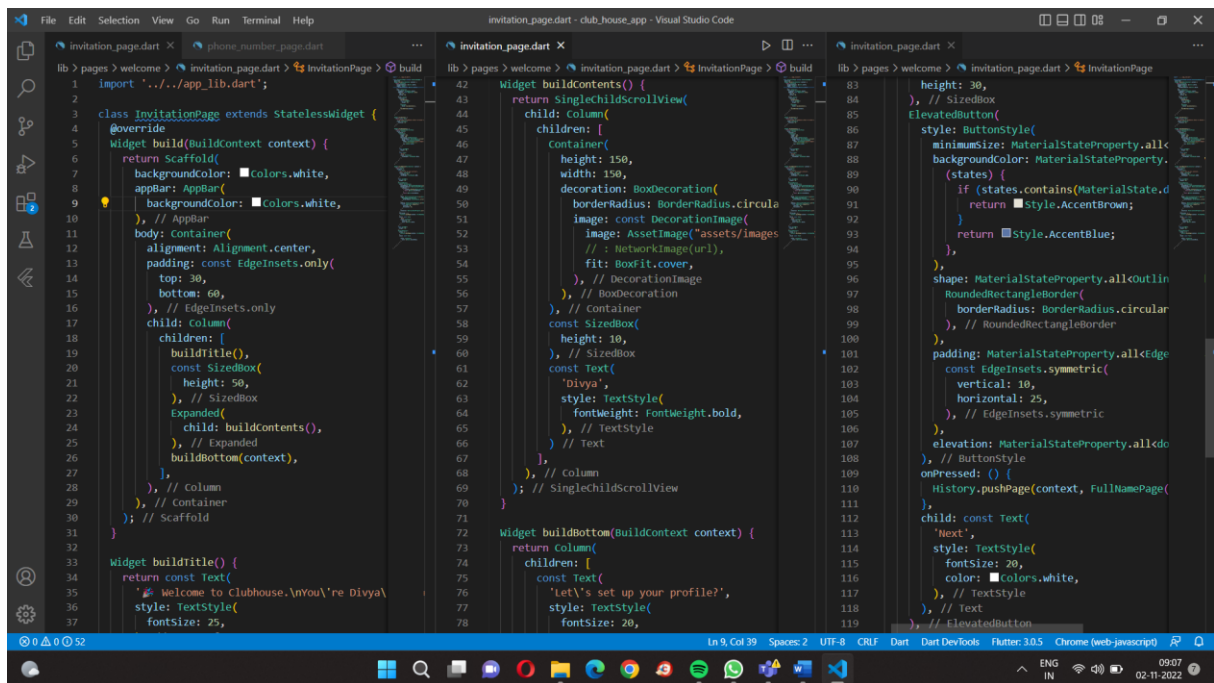
## 5.2 Phone Number Page:

- 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 PhoneNumberPage extends StatefulWidget {
4   @override
5   _PhoneNumberPageState createState() -> _Phone
6
7
8 class _PhoneNumberPageState extends State<Phone
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      ), // Column
37    ); // Container
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
```

### 5.3 Invitation Page:

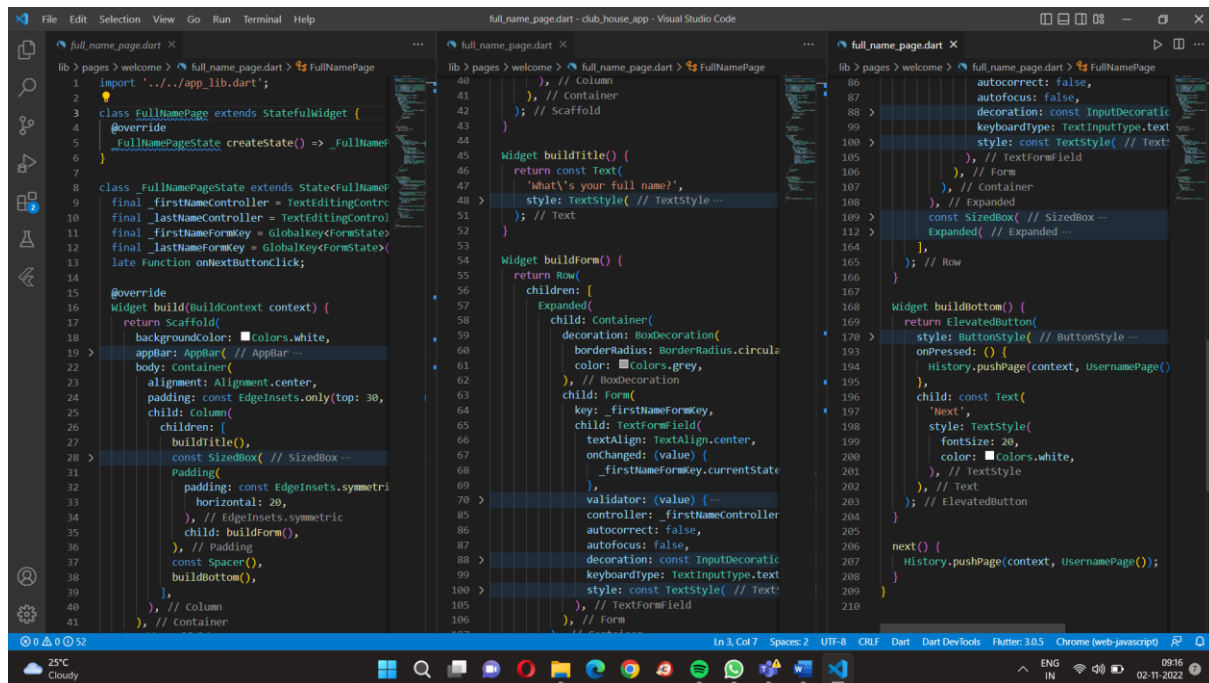
- Import Material Package to use the Built in Functionality
- Create a file invitationPage.dart And create a class welcomepage, It inherits The Properties of Stateless widget.
- Build() function returns Scaffold.scaffold consists of appBar and body
- Create widget functions buildTitle(),buildContents(),buildBottom()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



```
lib > pages > welcome > invitation_page.dart > InvitationPage > build
1  import 'package:flutter/material.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: const 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            ], // Column
28          ), // Container
29        ); // Scaffold
30      }
31
32  Widget buildTitle() {
33    return const Text(
34      "Welcome to Clubhouse, You're Divya",
35      style: TextStyle(
36        fontSize: 25,
37
38  lib > pages > welcome > invitation_page.dart > InvitationPage > build
39  Widget buildContents() {
40    return SingleChildScrollView(
41      child: Column(
42        children: [
43          Container(
44            height: 150,
45            width: 150,
46            decoration: BoxDecoration(
47              borderRadius: BorderRadius.circular(
48                10,
49              ), // BoxDecoration
50              image: const DecorationImage(
51                image: AssetImage("assets/images/
52                  profile.jpg"), // AssetImage
53                fit: BoxFit.cover,
54              ), // DecorationImage
55            ), // Container
56            const SizedBox(
57              height: 10,
58            ), // SizedBox
59            const Text(
60              "Divya",
61              style: TextStyle(
62                fontWeight: FontWeight.bold,
63                fontSize: 20,
64              ), // TextStyle
65            ), // Text
66          ], // Column
67        ), // SingleChildScrollView
68      );
69
70  Widget buildBottom(BuildContext context) {
71    return Column(
72      children: [
73        const Text(
74          "Let's set up your profile?",
75          style: TextStyle(
76            fontSize: 20,
77
78  lib > pages > welcome > invitation_page.dart > InvitationPage
79  height: 30,
80  ), // SizedBox
81  ElevatedButton(
82    style: ButtonStyle(
83      textStyle: MaterialStateProperty.all(
84        TextStyle(
85          color: Colors.white,
86          backgroundColor: MaterialStateProperty(
87            (states) {
88              if (states.contains(MaterialState.d
89                isPressed)) {
90                return Colors.brown;
91              }
92              return Colors.blue;
93            },
94          ), // ButtonStyle
95          shape: MaterialStateProperty.all(Outline
96            RoundedRectangleBorder(
97              borderRadius: BorderRadius.circular(
98                10,
99            ), // RoundedRectangleBorder
100          ), // ElevatedButton
101          padding: MaterialStateProperty.all(Edge
102            insets: EdgeInsets.symmetric(
103              vertical: 10,
104              horizontal: 25,
105            ), // EdgeInsets.symmetric
106          ), // ElevatedButton
107          elevation: MaterialStateProperty.all(0),
108          ), // ButtonStyle
109          onPressed: () {
110            History.pushPage(context, FullNamePage(
111              context,
112            ), child: const Text(
113              "Next",
114              style: TextStyle(
115                fontSize: 20,
116                color: Colors.white,
117              ), // TextStyle
118            ), // Text
119          ), // ElevatedButton
```

## 5.5 Full Name Page

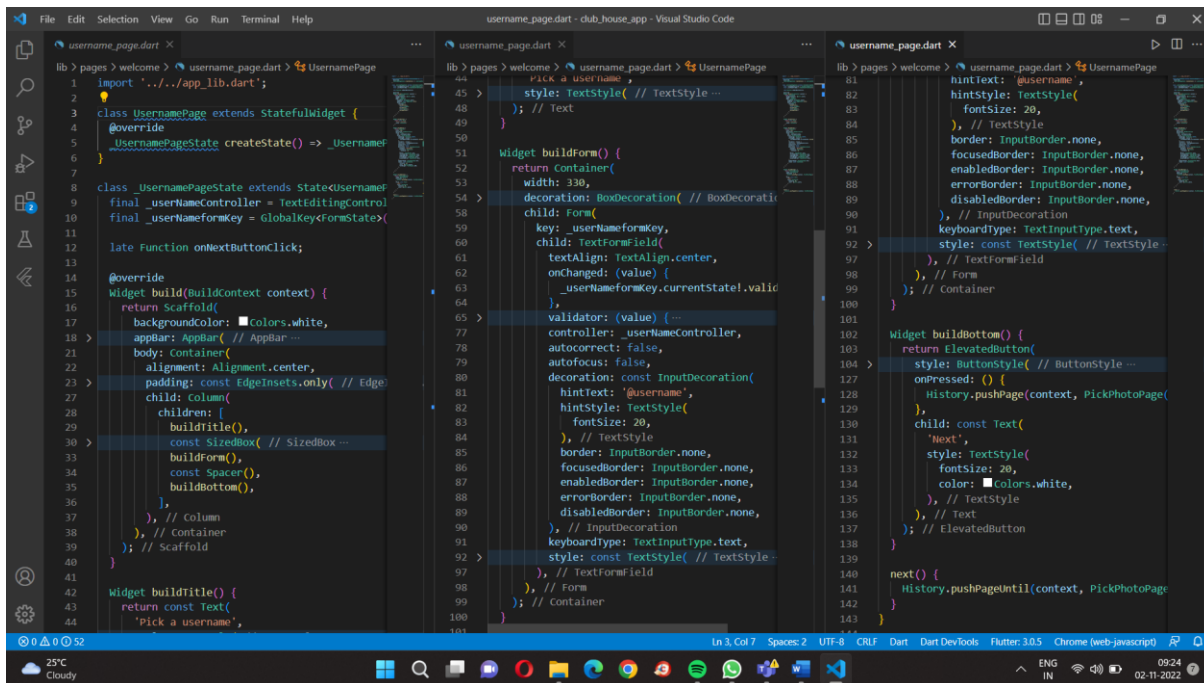
- Import Material Package to use the Built in Functionality
- Create a file fullNamePage.dart And create a class fullNamePage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body
- Create widget functions buildTitle(),buildForm(),buildBottom()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



```
lib > pages > welcome > full_name_page.dart > FullNamePage
1 import '../app_lib.dart';
2
3 class FullNamePage extends StatelessWidget {
4   @override
5   FullNamePageState createState() => _FullNamePageState
6 }
7
8 class FullNamePageState extends State<FullNamePage> {
9   final _firstNameController = TextEditingController();
10  final _lastNameController = TextEditingController();
11  final _firstNameFormKey = GlobalKey<FormState>();
12  final _lastNameFormKey = GlobalKey<FormState>();
13  late Function onTextButtonClick;
14
15  @override
16  Widget build(BuildContext context) {
17    return Scaffold(
18      backgroundColor: Colors.white,
19      appBar: AppBar( // AppBar ...
20        // ...
21      ),
22      body: Container(
23        alignment: Alignment.center,
24        padding: EdgeInsets.only(top: 30),
25        child: Column(
26          children: [
27            buildTitle(),
28            const SizedBox( // SizedBox ...
29              // ...
30            ),
31            Padding(
32              padding: EdgeInsets.symmetric(
33                horizontal: 20,
34              ), // EdgeInsets.symmetric
35              child: buildForm(),
36            ), // Padding
37            const Spacer(),
38            buildBottom(),
39          ], // Column
40        ), // container
41      ),
42    ); // Scaffold
43  }
44
45  Widget buildTitle() {
46    return const Text(
47      'What's your full name?',
48      style: TextStyle( // TextStyle ...
49        // ...
50      ), // Text
51    ); // Text
52  }
53
54  Widget buildForm() {
55    return Row(
56      children: [
57        Expanded(
58          child: Container(
59            decoration: BoxDecoration(
60              borderRadius: BorderRadius.circular(
61                10, // Colors.grey
62              ), // BoxDecoration
63            ), // Form(
64              key: _firstNameFormKey,
65              child: TextFormField(
66                textAlignment: TextAlign.center,
67                onChanged: (value) {
68                  _firstNameFormKey.currentState?.
69                    validator = (value) {
70                      // ...
71                    }, // validator
72                controller: _firstNameController,
73                autocorrect: false,
74                autofocus: false,
75                decoration: const InputDecoration(
76                  keyboardType: TextInputType.text,
77                  style: const TextStyle( // Text
78                    // ...
79                  ), // Form
80                ), // TextFormField
81              ), // Form
82            ), // Container
83          ), // Expanded
84        ), // Row
85      ), // Row
86    ); // Row
87  }
88
89  Widget buildBottom() {
90    return ElevatedButton(
91      style: ButtonStyle( // ButtonStyle ...
92        // ...
93      ), // ElevatedButton
94      onPressed: () {
95        History.pushPage(context, UsernamePage());
96      }, // ElevatedButton
97      child: const Text(
98        'Next',
99        style: TextStyle(
100          fontSize: 20,
101          color: Colors.white,
102        ), // Text
103      ), // ElevatedButton
104    ); // ElevatedButton
105  }
106
107  next() {
108    History.pushPage(context, UsernamePage());
109  }
110}
```

## 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.



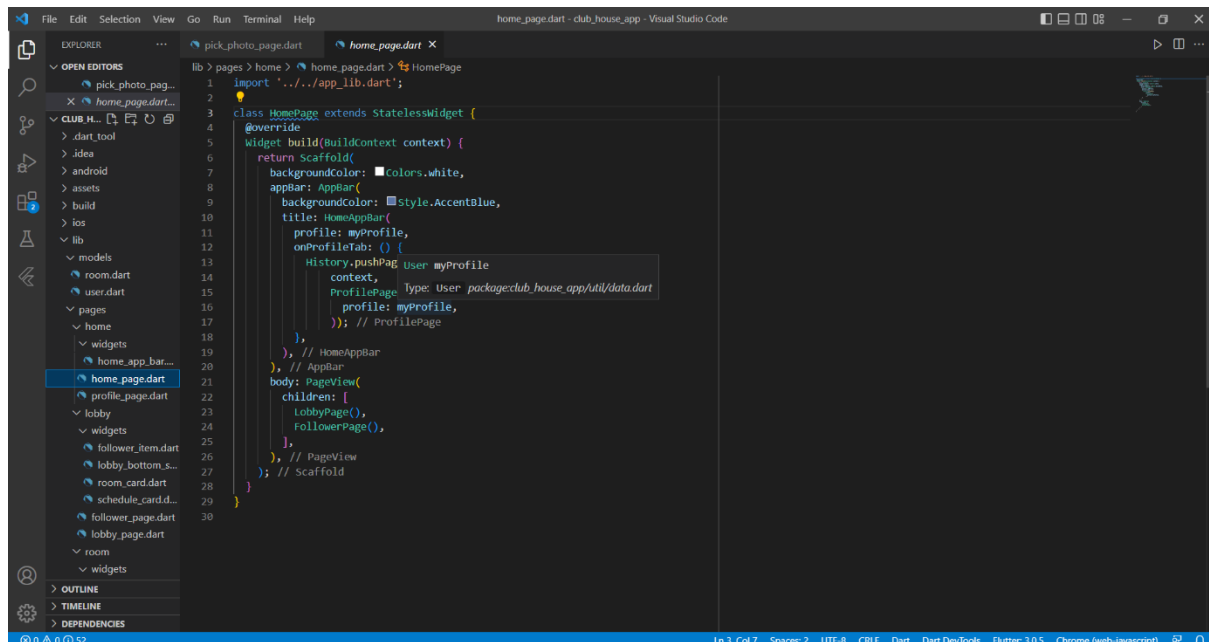
```
lib > pages > welcome > username_page.dart > UsernamePage
1 import '../app_lib.dart';
2
3 class UsernamePage extends StatefulWidget {
4   @override
5   UsernamePageState createState() => _UsernamePageState
6 }
7
8 class _UsernamePageState extends State<UsernamePage> {
9   final _usernameController = 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                buildForm(),
26                const Spacer(),
27                buildBottom(),
28              ], // Column
29            ), // Container
30          ), // Scaffold
31        );
32
33    widget buildTitle() {
34      return const Text(
35        'Pick a username',
36      );
37    }
38
39    widget buildForm() {
40      return Container(
41        width: 330,
42        decoration: BoxDecoration( // BoxDecoration ...
43          child: Form(
44            key: _usernameFormKey,
45            child: TextFormField(
46              textAlign: TextAlign.center,
47              autofocus: false,
48              _usernameFormKey.currentState!.valid
49            ),
50          ),
51          validator: (value) {
52            controller: _usernameController,
53            autocorrect: false,
54            autofocus: false,
55            decoration: const InputDecoration(
56              hintText: '@username',
57              hintStyle: TextStyle(
58                fontSize: 20,
59              ), // TextStyle
60            ), // InputDecoration
61            border: InputBorder.none,
62            focusedBorder: InputBorder.none,
63            enabledBorder: InputBorder.none,
64            errorBorder: InputBorder.none,
65            disabledBorder: InputBorder.none,
66          ), // InputDecoration
67          keyboardType: TextInputType.text,
68          style: const TextStyle( // TextStyle ...
69            ), // TextFormField
70          ), // Form
71        ); // Container
72      );
73    }
74
75    widget buildBottom() {
76      return ElevatedButton(
77        style: ButtonStyle( // ButtonStyle ...
78          onPressed: () {
79            History.pushPage(context, PickPhotoPage(
80              child: const Text(
81                'Next',
82                style: TextStyle(
83                  fontSize: 20,
84                  color: Colors.white,
85                ), // TextStyle
86              ), // Text
87            ); // ElevatedButton
88          },
89        );
90
91    next() {
92      History.pushPageUntil(context, PickPhotoPage(
93      );
94    }
95  }
96 }
```



- 

## 5.8 Home Page

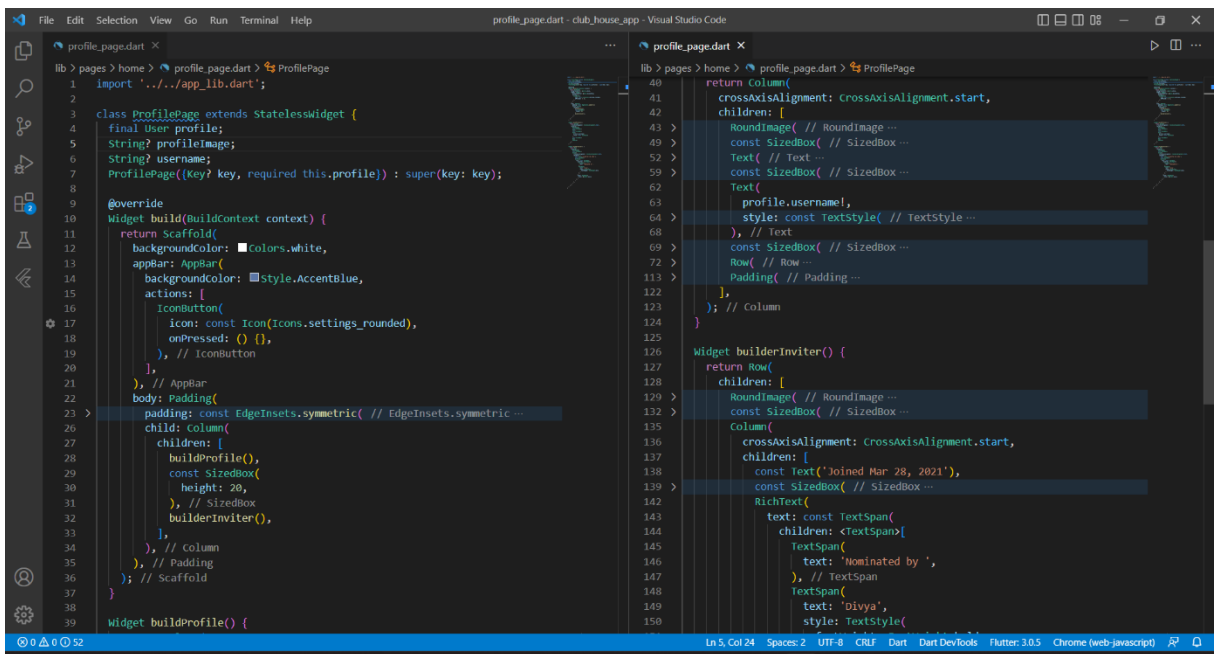
- Import Material Package to use the Built in Functionality
- Create a file HomePage.dart And create a class HomePage, It inherits The Properties of StatelessWidget.
- Build() function returns Scaffold.scaffold consists of appBar and body
- Body consists of two classes
  - Lobby Page
  - Follower Page



```
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(
14              context,
15              ProfilePage(
16                type: User package:club_house_app/util/data.dart
17                profile: myProfile,
18              ); // ProfilePage
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



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

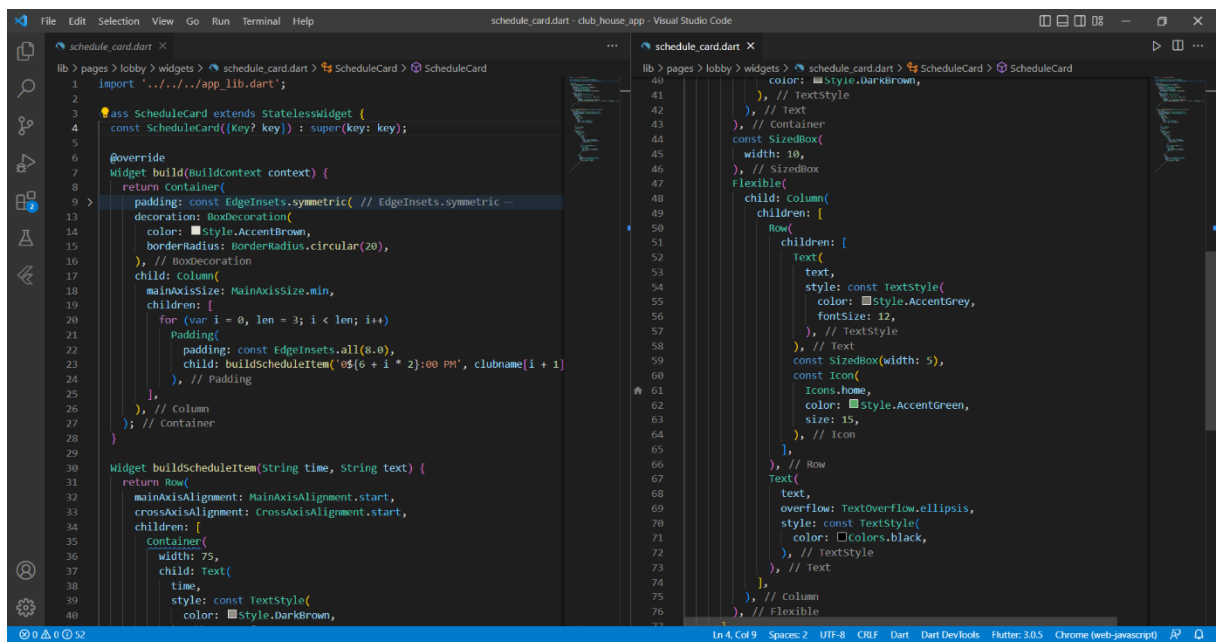
```

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() => lobbyPageSta...
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(millise...
15     _refreshController.refreshCompleted();
16   }
17
18   void _onLoading() async {
19     await Future.delayed(const Duration(millise...
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( // 1
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               }
49             ); // Stack
50           ); // Stack
51         Widget buildScheduleCard() {
52           return Container(
53             margin: const EdgeInsets.symmetric( // Ed
54               child: const ScheduleCard(),
55             ); // Container
56           );
57         }
58         Widget buildRoomCard(Room room, int index) {
59           return GestureDetector(
60             onTap: () { ...
61             child: Container(
62               margin: const EdgeInsets.symmetric( // 1
63               child: RoomCard( // RoomCard ...
64             ); // Container
65           ); // GestureDetector
66         }
67         Widget buildGradientContainer() {
68           return Container(
69             height: 50,
70             decoration: BoxDecoration( // LinearGradi
71           ); // Container
72         }
73         Widget buildStartRoomButton() {
74           return Container(
75             margin: const EdgeInsets.only(bottom: 20),
76             child: ElevatedButton(
77               style: ButtonStyle( // ButtonStyle ...
78               onPressed: () {
79                 showBottomSheet();
80               },
81               child: const Text( // Text ...
82             ); // ElevatedButton
83           ); // Container
84         }
85         enterRoom(Room room) {
86           showModalBottomSheet( ...
87         }
88         showBottomSheet() {
89           showModalBottomSheet(
90             context: context,
91             shape: const RoundedRectangleBorder( // B
92             builder: (context) {
93               return Wrap(
94                 children: [
95                   lobbyBottomSheet(
96                     onPressed: () { ...
97                   ); // LobbyBottomSheet
98                 ],
99               ); // Wrap
100             },
101           );
102         }
103       ],
104     );
105   }
106 }
  
```

### 5.8.2.1 ScheduleCard

- Import Material Package to use the Built in Functionility
- 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

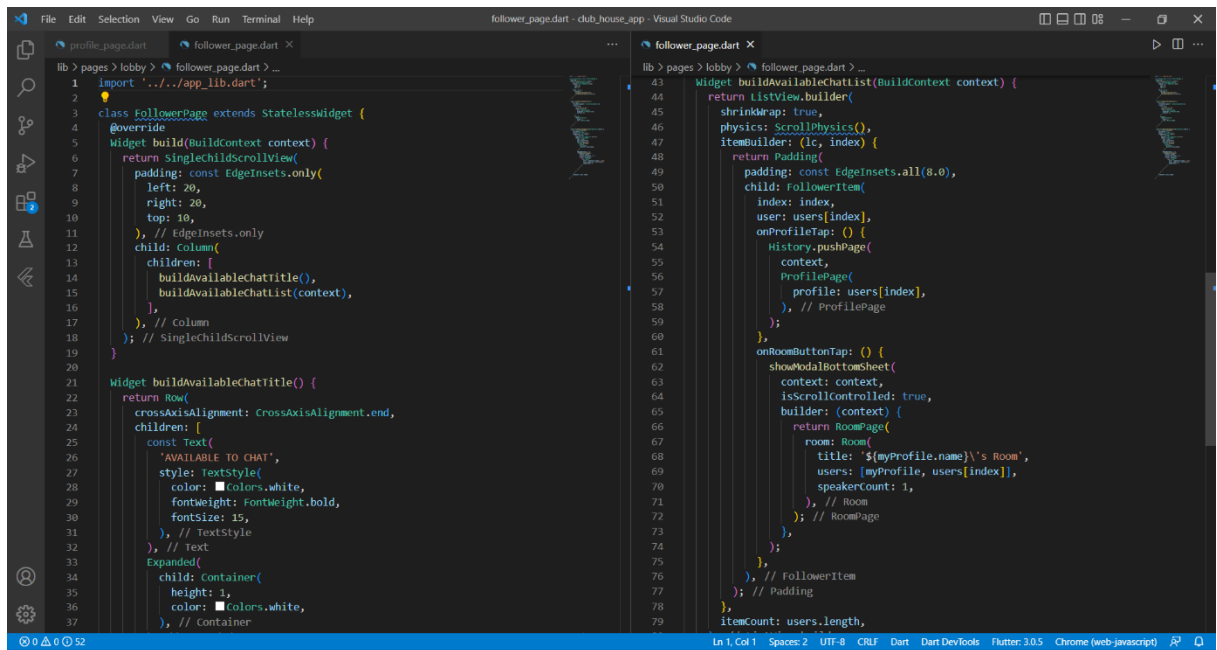
- Import Material Package to use the Built in Functionality
- Create a file `RoomCard.dart` And create a class `RoomCard`, It inherits The Properties of `Stateless widget`.
- It displays the currently running meetings.
- Users can join the meetings by clicking the room card.

```

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                  ], // Container
34                ], // Row
35              ), // Column
36            ), // Container
37          ], // Row
38        ), // Column
39      ), // Container
40    ); // Container
41  }
42
43  Widget buildProfileImages() {
44    return Stack(
45      children: [
46        RoundImage( // RoundImage ...
47        RoundImage( // RoundImage ...
48      ], // Stack
49    ); // Stack
50  }
51
52  Widget buildUserList() {
53    var len = room.users.length > 4 ? 4 : room.users.length;
54
55    return Column(
56      crossAxisAlignment: CrossAxisAlignment.start,
57      children: [
58        for (var i = 0; i < len; i++)
59          Container(
60            child: Row(
61              children: [
62                Text(
63                  room.users[i].name,
64                  style: const TextStyle( // TextStyle ...
65                ), // Text
66                const SizedBox( // SizedBox ...
67                const Icon( // Icon ...
68              ], // Row
69            ), // Container
70          ], // Column
71        ); // Column
72  }
73
74  Widget buildRoomInfo() {
75    return Row(
76      children: [
77        Text( // Text ...
78        const Icon( // Icon ...
79      ], // Row
80    ); // Row
81  }
82
83  Widget buildRoomInfo() {
84    return Row(
85      children: [
86        Text( // Text ...
87        const Icon( // Icon ...
88      ], // Row
89    ); // Row
90  }
91
92  Widget buildRoomInfo() {
93    return Row(
94      children: [
95        Text( // Text ...
96        const Icon( // Icon ...
97      ], // Row
98    ); // Row
99  }
100
101  Widget buildRoomInfo() {
102    return Row(
103      children: [
104        Text( // Text ...
105        const Icon( // Icon ...
106      ], // Row
107    ); // Row
108  }
109
110  Widget buildRoomInfo() {
111    return Row(
112      children: [
113        Text( // Text ...
114        const Icon( // Icon ...
115      ], // Row
116    ); // Row
117  }
118
119  Widget buildRoomInfo() {
120    return Row(
121      children: [
122        Text( // Text ...
123        const Icon( // Icon ...
124      ], // Row
125    ); // Row
126  }
127
128  Widget buildRoomInfo() {
129    return Row(
130      children: [
131        Text( // Text ...
132        const Icon( // Icon ...
133      ], // Row
134    ); // Row
135  }
136
137  Widget buildRoomInfo() {
138    return Row(
139      children: [
140        Text( // Text ...
141        const Icon( // Icon ...
142      ], // Row
143    ); // Row
144  }
145
146  Widget buildRoomInfo() {
147    return Row(
148      children: [
149        Text( // Text ...
150        const Icon( // Icon ...
151      ], // Row
152    ); // Row
153  }
154
155  Widget buildRoomInfo() {
156    return Row(
157      children: [
158        Text( // Text ...
159        const Icon( // Icon ...
160      ], // Row
161    ); // Row
162  }
163
164  Widget buildRoomInfo() {
165    return Row(
166      children: [
167        Text( // Text ...
168        const Icon( // Icon ...
169      ], // Row
170    ); // Row
171  }
172
173  Widget buildRoomInfo() {
174    return Row(
175      children: [
176        Text( // Text ...
177        const Icon( // Icon ...
178      ], // Row
179    ); // Row
180  }
181
182  Widget buildRoomInfo() {
183    return Row(
184      children: [
185        Text( // Text ...
186        const Icon( // Icon ...
187      ], // Row
188    ); // Row
189  }
190
191  Widget buildRoomInfo() {
192    return Row(
193      children: [
194        Text( // Text ...
195        const Icon( // Icon ...
196      ], // Row
197    ); // Row
198  }
199
200  Widget buildRoomInfo() {
201    return Row(
202      children: [
203        Text( // Text ...
204        const Icon( // Icon ...
205      ], // Row
206    ); // Row
207  }
208
209  Widget buildRoomInfo() {
210    return Row(
211      children: [
212        Text( // Text ...
213        const Icon( // Icon ...
214      ], // Row
215    ); // Row
216  }
217
218  Widget buildRoomInfo() {
219    return Row(
220      children: [
221        Text( // Text ...
222        const Icon( // Icon ...
223      ], // Row
224    ); // Row
225  }
226
227  Widget buildRoomInfo() {
228    return Row(
229      children: [
230        Text( // Text ...
231        const Icon( // Icon ...
232      ], // Row
233    ); // Row
234  }
235
236  Widget buildRoomInfo() {
237    return Row(
238      children: [
239        Text( // Text ...
240        const Icon( // Icon ...
241      ], // Row
242    ); // Row
243  }
244
245  Widget buildRoomInfo() {
246    return Row(
247      children: [
248        Text( // Text ...
249        const Icon( // Icon ...
250      ], // Row
251    ); // Row
252  }
253
254  Widget buildRoomInfo() {
255    return Row(
256      children: [
257        Text( // Text ...
258        const Icon( // Icon ...
259      ], // Row
260    ); // Row
261  }
262
263  Widget buildRoomInfo() {
264    return Row(
265      children: [
266        Text( // Text ...
267        const Icon( // Icon ...
268      ], // Row
269    ); // Row
270  }
271
272  Widget buildRoomInfo() {
273    return Row(
274      children: [
275        Text( // Text ...
276        const Icon( // Icon ...
277      ], // Row
278    ); // Row
279  }
280
281  Widget buildRoomInfo() {
282    return Row(
283      children: [
284        Text( // Text ...
285        const Icon( // Icon ...
286      ], // Row
287    ); // Row
288  }
289
290  Widget buildRoomInfo() {
291    return Row(
292      children: [
293        Text( // Text ...
294        const Icon( // Icon ...
295      ], // Row
296    ); // Row
297  }
298
299  Widget buildRoomInfo() {
300    return Row(
301      children: [
302        Text( // Text ...
303        const Icon( // Icon ...
304      ], // Row
305    ); // Row
306  }
307
308  Widget buildRoomInfo() {
309    return Row(
310      children: [
311        Text( // Text ...
312        const Icon( // Icon ...
313      ], // Row
314    ); // Row
315  }
316
317  Widget buildRoomInfo() {
318    return Row(
319      children: [
320        Text( // Text ...
321        const Icon( // Icon ...
322      ], // Row
323    ); // Row
324  }
325
326  Widget buildRoomInfo() {
327    return Row(
328      children: [
329        Text( // Text ...
330        const Icon( // Icon ...
331      ], // Row
332    ); // Row
333  }
334
335  Widget buildRoomInfo() {
336    return Row(
337      children: [
338        Text( // Text ...
339        const Icon( // Icon ...
340      ], // Row
341    ); // Row
342  }
343
344  Widget buildRoomInfo() {
345    return Row(
346      children: [
347        Text( // Text ...
348        const Icon( // Icon ...
349      ], // Row
350    ); // Row
351  }
352
353  Widget buildRoomInfo() {
354    return Row(
355      children: [
356        Text( // Text ...
357        const Icon( // Icon ...
358      ], // Row
359    ); // Row
360  }
361
362  Widget buildRoomInfo() {
363    return Row(
364      children: [
365        Text( // Text ...
366        const Icon( // Icon ...
367      ], // Row
368    ); // Row
369  }
370
371  Widget buildRoomInfo() {
372    return Row(
373      children: [
374        Text( // Text ...
375        const Icon( // Icon ...
376      ], // Row
377    ); // Row
378  }
379
380  Widget buildRoomInfo() {
381    return Row(
382      children: [
383        Text( // Text ...
384        const Icon( // Icon ...
385      ], // Row
386    ); // Row
387  }
388
389  Widget buildRoomInfo() {
390    return Row(
391      children: [
392        Text( // Text ...
393        const Icon( // Icon ...
394      ], // Row
395    ); // Row
396  }
397
398  Widget buildRoomInfo() {
399    return Row(
400      children: [
401        Text( // Text ...
402        const Icon( // Icon ...
403      ], // Row
404    ); // Row
405  }
406
407  Widget buildRoomInfo() {
408    return Row(
409      children: [
410        Text( // Text ...
411        const Icon( // Icon ...
412      ], // Row
413    ); // Row
414  }
415
416  Widget buildRoomInfo() {
417    return Row(
418      children: [
419        Text( // Text ...
420        const Icon( // Icon ...
421      ], // Row
422    ); // Row
423  }
424
425  Widget buildRoomInfo() {
426    return Row(
427      children: [
428        Text( // Text ...
429        const Icon( // Icon ...
430      ], // Row
431    ); // Row
432  }
433
434  Widget buildRoomInfo() {
435    return Row(
436      children: [
437        Text( // Text ...
438        const Icon( // Icon ...
439      ], // Row
440    ); // Row
441  }
442
443  Widget buildRoomInfo() {
444    return Row(
445      children: [
446        Text( // Text ...
447        const Icon( // Icon ...
448      ], // Row
449    ); // Row
450  }
451
452  Widget buildRoomInfo() {
453    return Row(
454      children: [
455        Text( // Text ...
456        const Icon( // Icon ...
457      ], // Row
458    ); // Row
459  }
460
461  Widget buildRoomInfo() {
462    return Row(
463      children: [
464        Text( // Text ...
465        const Icon( // Icon ...
466      ], // Row
467    ); // Row
468  }
469
470  Widget buildRoomInfo() {
471    return Row(
472      children: [
473        Text( // Text ...
474        const Icon( // Icon ...
475      ], // Row
476    ); // Row
477  }
478
479  Widget buildRoomInfo() {
480    return Row(
481      children: [
482        Text( // Text ...
483        const Icon( // Icon ...
484      ], // Row
485    ); // Row
486  }
487
488  Widget buildRoomInfo() {
489    return Row(
490      children: [
491        Text( // Text ...
492        const Icon( // Icon ...
493      ], // Row
494    ); // Row
495  }
496
497  Widget buildRoomInfo() {
498    return Row(
499      children: [
500        Text( // Text ...
501        const Icon( // Icon ...
502      ], // Row
503    ); // Row
504  }
505
506  Widget buildRoomInfo() {
507    return Row(
508      children: [
509        Text( // Text ...
510        const Icon( // Icon ...
511      ], // Row
512    ); // Row
513  }
514
515  Widget buildRoomInfo() {
516    return Row(
517      children: [
518        Text( // Text ...
519        const Icon( // Icon ...
520      ], // Row
521    ); // Row
522  }
523
524  Widget buildRoomInfo() {
525    return Row(
526      children: [
527        Text( // Text ...
528        const Icon( // Icon ...
529      ], // Row
530    ); // Row
531  }
532
533  Widget buildRoomInfo() {
534    return Row(
535      children: [
536        Text( // Text ...
537        const Icon( // Icon ...
538      ], // Row
539    ); // Row
540  }
541
542  Widget buildRoomInfo() {
543    return Row(
544      children: [
545        Text( // Text ...
546        const Icon( // Icon ...
547      ], // Row
548    ); // Row
549  }
550
551  Widget buildRoomInfo() {
552    return Row(
553      children: [
554        Text( // Text ...
555        const Icon( // Icon ...
556      ], // Row
557    ); // Row
558  }
559
560  Widget buildRoomInfo() {
561    return Row(
562      children: [
563        Text( // Text ...
564        const Icon( // Icon ...
565      ], // Row
566    ); // Row
567  }
568
569  Widget buildRoomInfo() {
570    return Row(
571      children: [
572        Text( // Text ...
573        const Icon( // Icon ...
574      ], // Row
575    ); // Row
576  }
577
578  Widget buildRoomInfo() {
579    return Row(
580      children: [
581        Text( // Text ...
582        const Icon( // Icon ...
583      ], // Row
584    ); // Row
585  }
586
587  Widget buildRoomInfo() {
588    return Row(
589      children: [
590        Text( // Text ...
591        const Icon( // Icon ...
592      ], // Row
593    ); // Row
594  }
595
596  Widget buildRoomInfo() {
597    return Row(
598      children: [
599        Text( // Text ...
600        const Icon( // Icon ...
601      ], // Row
602    ); // Row
603  }
604
605  Widget buildRoomInfo() {
606    return Row(
607      children: [
608        Text( // Text ...
609        const Icon( // Icon ...
610      ], // Row
611    ); // Row
612  }
613
614  Widget buildRoomInfo() {
615    return Row(
616      children: [
617        Text( // Text ...
618        const Icon( // Icon ...
619      ], // Row
620    ); // Row
621  }
622
623  Widget buildRoomInfo() {
624    return Row(
625      children: [
626        Text( // Text ...
627        const Icon( // Icon ...
628      ], // Row
629    ); // Row
630  }
631
632  Widget buildRoomInfo() {
633    return Row(
634      children: [
635        Text( // Text ...
636        const Icon( // Icon ...
637      ], // Row
638    ); // Row
639  }
640
641  Widget buildRoomInfo() {
642    return Row(
643      children: [
644        Text( // Text ...
645        const Icon( // Icon ...
646      ], // Row
647    ); // Row
648  }
649
650  Widget buildRoomInfo() {
651    return Row(
652      children: [
653        Text( // Text ...
654        const Icon( // Icon ...
655      ], // Row
656    ); // Row
657  }
658
659  Widget buildRoomInfo() {
660    return Row(
661      children: [
662        Text( // Text ...
663        const Icon( // Icon ...
664      ], // Row
665    ); // Row
666  }
667
668  Widget buildRoomInfo() {
669    return Row(
670      children: [
671        Text( // Text ...
672        const Icon( // Icon ...
673      ], // Row
674    ); // Row
675  }
676
677  Widget buildRoomInfo() {
678    return Row(
679      children: [
680        Text( // Text ...
681        const Icon( // Icon ...
682      ], // Row
683    ); // Row
684  }
685
686  Widget buildRoomInfo() {
687    return Row(
688      children: [
689        Text( // Text ...
690        const Icon( // Icon ...
691      ], // Row
692    ); // Row
693  }
694
695  Widget buildRoomInfo() {
696    return Row(
697      children: [
698        Text( // Text ...
699        const Icon( // Icon ...
700      ], // Row
701    ); // Row
702  }
703
704  Widget buildRoomInfo() {
705    return Row(
706      children: [
707        Text( // Text ...
708        const Icon( // Icon ...
709      ], // Row
710    ); // Row
711  }
712
713  Widget buildRoomInfo() {
714    return Row(
715      children: [
716        Text( // Text ...
717        const Icon( // Icon ...
718      ], // Row
719    ); // Row
720  }
721
722  Widget buildRoomInfo() {
723    return Row(
724      children: [
725        Text( // Text ...
726        const Icon( // Icon ...
727      ], // Row
728    ); // Row
729  }
730
731  Widget buildRoomInfo() {
732    return Row(
733      children: [
734        Text( // Text ...
735        const Icon( // Icon ...
736      ], // Row
737    ); // Row
738  }
739
740  Widget buildRoomInfo() {
741    return Row(
742      children: [
743        Text( // Text ...
744        const Icon( // Icon ...
745      ], // Row
746    ); // Row
747  }
748
749  Widget buildRoomInfo() {
750    return Row(
751      children: [
752        Text( // Text ...
753        const Icon( // Icon ...
754      ], // Row
755    ); // Row
756  }
757
758  Widget buildRoomInfo() {
759    return Row(
760      children: [
761        Text( // Text ...
762        const Icon( // Icon ...
763      ], // Row
764    ); // Row
765  }
766
767  Widget buildRoomInfo() {
768    return Row(
769      children: [
770        Text( // Text ...
771        const Icon( // Icon ...
772      ], // Row
773    ); // Row
774  }
775
776  Widget buildRoomInfo() {
777    return Row(
778      children: [
779        Text( // Text ...
780        const Icon( // Icon ...
781      ], // Row
782    ); // Row
783  }
784
785  Widget buildRoomInfo() {
786    return Row(
787      children: [
788        Text( // Text ...
789        const Icon( // Icon ...
790      ], // Row
791    ); // Row
792  }
793
794  Widget buildRoomInfo() {
795    return Row(
796      children: [
797        Text( // Text ...
798        const Icon( // Icon ...
799      ], // Row
800    ); // Row
801  }
802
803  Widget buildRoomInfo() {
804    return Row(
805      children: [
806        Text( // Text ...
807        const Icon( // Icon ...
808      ], // Row
809    ); // Row
810  }
811
812  Widget buildRoomInfo() {
813    return Row(
814      children: [
815        Text( // Text ...
816        const Icon( // Icon ...
817      ], // Row
818    ); // Row
819  }
820
821  Widget buildRoomInfo() {
822    return Row(
823      children: [
824        Text( // Text ...
825        const Icon( // Icon ...
826      ], // Row
827    ); // Row
828  }
829
830  Widget buildRoomInfo() {
831    return Row(
832      children: [
833        Text( // Text ...
834        const Icon( // Icon ...
835      ], // Row
836    ); // Row
837  }
838
839  Widget buildRoomInfo() {
840    return Row(
841      children: [
842        Text( // Text ...
843        const Icon( // Icon ...
844      ], // Row
845    ); // Row
846  }
847
848  Widget buildRoomInfo() {
849    return Row(
850      children: [
851        Text( // Text ...
852        const Icon( // Icon ...
853      ], // Row
854    ); // Row
855  }
856
857  Widget buildRoomInfo() {
858    return Row(
859      children: [
860        Text( // Text ...
861        const Icon( // Icon ...
862      ], // Row
863    ); // Row
864  }
865
866  Widget buildRoomInfo() {
867    return Row(
868      children: [
869        Text( // Text ...
870        const Icon( // Icon ...
871      ], // Row
872    ); // Row
873  }
874
875  Widget buildRoomInfo() {
876    return Row(
877      children: [
878        Text( // Text ...
879        const Icon( // Icon ...
880      ], // Row
881    ); // Row
882  }
883
884  Widget buildRoomInfo() {
885    return Row(
886      children: [
887        Text( // Text ...
888        const Icon( // Icon ...
889      ], // Row
890    ); // Row
891  }
892
893  Widget buildRoomInfo() {
894    return Row(
895      children: [
896        Text( // Text ...
897        const Icon( // Icon ...
898      ], // Row
899    ); // Row
900  }
901
902  Widget buildRoomInfo() {
903    return Row(
904      children: [
905        Text( // Text ...
906        const Icon( // Icon ...
907      ], // Row
908    ); // Row
909  }
910
911  Widget buildRoomInfo() {
912    return Row(
913      children: [
914        Text( // Text ...
915        const Icon( // Icon ...
916      ], // Row
917    ); // Row
918  }
919
920  Widget buildRoomInfo() {
921    return Row(
922      children: [
923        Text( // Text ...
924        const Icon( // Icon ...
925      ], // Row
926    ); // Row
927  }
928
929  Widget buildRoomInfo() {
930    return Row(
931      children: [
932        Text( // Text ...
933        const Icon( // Icon ...
934      ], // Row
935    ); // Row
936  }
937
938  Widget buildRoomInfo() {
939    return Row(
940      children: [
941        Text( // Text ...
942        const Icon( // Icon ...
943      ], // Row
944    ); // Row
945  }
946
947  Widget buildRoomInfo() {
948    return Row(
949      children: [
950        Text( // Text ...
951        const Icon( // Icon ...
952      ], // Row
953    ); // Row
954  }
955
956  Widget buildRoomInfo() {
957    return Row(
958      children: [
959        Text( // Text ...
960        const Icon( // Icon ...
961      ], // Row
962    ); // Row
963  }
964
965  Widget buildRoomInfo() {
966    return Row(
967      children: [
968        Text( // Text ...
969        const Icon( // Icon ...
970      ], // Row
971    ); // Row
972  }
973
974  Widget buildRoomInfo() {
975    return Row(
976      children: [
977        Text( // Text ...
978        const Icon( // Icon ...
979      ], // Row
980    ); // Row
981  }
982
983  Widget buildRoomInfo() {
984    return Row(
985      children: [
986        Text( // Text ...
987        const Icon( // Icon ...
988      ], // Row
989    ); // Row
990  }
991
992  Widget buildRoomInfo() {
993    return Row(
994      children: [
995        Text( // Text ...
996        const Icon( // Icon ...
997      ], // Row
998    ); // Row
999  }
1000
1001  Widget buildRoomInfo() {
1002    return Row(
1003      children: [
1004        Text( // Text ...
1005        const Icon( // Icon ...
1006      ], // Row
1007    ); // Row
1008  }
1009
1010  Widget buildRoomInfo() {
1011    return Row(
1012      children: [
1013        Text( // Text ...
1014        const Icon( // Icon ...
1015      ], // Row
1016    ); // Row
1017  }
1018
1019  Widget buildRoomInfo() {
1020    return Row(
1021      children: [
1022        Text( // Text ...
1023        const Icon( // Icon ...
1024      ], // Row
1025    ); // Row
1026  }
1027
1028  Widget buildRoomInfo() {
1029    return Row(
1030      children: [
1031        Text( // Text ...
1032        const Icon( // Icon ...
1033      ], // Row
1034    ); // Row
1035  }
1036
1037  Widget buildRoomInfo() {
1038    return Row(
1039      children: [
1040        Text( // Text ...
1041        const Icon( // Icon ...
1042      ], // Row
1043    ); // Row
1044  }
1045
1046  Widget buildRoomInfo() {
1047    return Row(
1048      children: [
1049        Text( // Text ...
1050        const Icon( // Icon ...
1051      ], // Row
1052    ); // Row
1053  }
1054
1055  Widget buildRoomInfo() {
1056    return Row(
1057      children: [
1058        Text( // Text ...
1059        const Icon( // Icon ...
1060      ], // Row
1061    ); // Row
1062  }
1063
1064  Widget buildRoomInfo() {
1065    return Row(
1066      children: [
1067        Text( // Text ...
1068        const Icon( // Icon ...
1069      ], // Row
1070    ); // Row
1071  }
1072
1073  Widget buildRoomInfo() {
1074    return Row(
1075      children: [
1076        Text( // Text ...
1077        const Icon( // Icon ...
1078      ], // Row
1079    ); // Row
1080  }
1081
1082  Widget buildRoomInfo() {
1083    return Row(
1084      children: [
1085        Text( // Text ...
1086        const Icon( // Icon ...
1087      ], // Row
1088    ); // Row
1089  }
1090
1091  Widget buildRoomInfo() {
1092    return Row(
1093      children: [
1094        Text( // Text ...
1095        const Icon( // Icon ...
1096      ], // Row
1097    ); // Row
1098  }
1099
1100  Widget buildRoomInfo() {
1101    return Row(
1102      children: [
1103        Text( // Text ...
1104        const Icon( // Icon ...
1105      ], // Row
1106    ); // Row
1107  }
1108
1109  Widget buildRoomInfo() {
1110    return Row(
1111      children: [
1112        Text( // Text ...
1113        const Icon( // Icon ...
1114      ], // Row
1115    ); // Row
1116  }
1117
1118  Widget buildRoomInfo() {
1119    return Row(
1120      children: [
1121        Text( // Text ...
1122        const Icon( // Icon ...
1123      ], // Row
1124    ); // Row
1125  }
1126
1127  Widget buildRoomInfo() {
1128    return Row(
1129      children: [
1130        Text( // Text ...
1131        const Icon( // Icon ...
1132      ], // Row
1133    ); // Row
1134  }
1135
1136  Widget buildRoomInfo() {
1137    return Row(
1138      children: [
1139        Text( // Text ...
1140        const Icon( // Icon ...
1141      ], // Row
1142    ); // Row
1143  }
1144
1145  Widget buildRoomInfo() {
1146    return Row(
1147      children: [
1148        Text( // Text ...
1149        const Icon( // Icon ...
1150      ], // Row
1151    ); // Row
1152  }
1153
1154  Widget buildRoomInfo() {
1155    return Row(
1156      children: [
1157        Text( // Text ...
1158        const Icon( // Icon ...
1159      ], // Row
1160    ); // Row
1161  }
1162
1163  Widget buildRoomInfo() {
1164    return Row(
1165      children: [
1166        Text( // Text ...
1167        const Icon( // Icon ...
1168      ], // Row
1169    ); // Row
1170  }
1171
1172  Widget buildRoomInfo() {
1173    return Row(
1174      children: [
1175        Text( // Text ...
1176        const Icon( // Icon ...
1177      ], // Row
1178    ); // Row
1179  }
1180
1181  Widget buildRoomInfo() {
1182    return Row(
1183      children: [
1184        Text( // Text ...
1185        const Icon( // Icon ...
1186      ], // Row
1187    ); // Row
1188  }
1189
1190  Widget buildRoomInfo() {
1191    return Row(
1192      children: [
1193        Text( // Text ...
1194        const Icon( // Icon ...
1195      ], // Row
1196    ); // Row
1197  }
1198
1199  Widget buildRoomInfo() {
1200    return Row(
1201      children: [
1202        Text( // Text ...
1203        const Icon( // Icon ...
1204      ], // Row
1205    ); // Row
1206  }
1207
1208  Widget buildRoomInfo() {
1209    return Row(
1210      children: [
1211        Text( // Text ...
1212        const Icon( // Icon ...
1213      ], // Row
1214    ); // Row
1215  }
1216
1217  Widget buildRoomInfo() {
1218    return Row(
1219      children: [
1220        Text( // Text ...
1221        const Icon( // Icon ...
1222      ], // Row
1223    ); // Row
1224  }
1225
1226  Widget buildRoomInfo() {
1227    return Row(
1228      children: [
1229        Text( // Text ...
1230        const Icon( // Icon ...
1231      ], // Row
1232    ); // Row
1233  }
1234
1235  Widget buildRoomInfo() {
1236    return Row(
1237      children: [
1238        Text( // Text ...
1239        const Icon( // Icon ...
1240      ], // Row
1241    ); // Row
1242  }
1243
1244  Widget buildRoomInfo() {
1245    return Row(
1246      children: [
1247        Text( // Text ...
1248        const Icon( // Icon ...
1249      ], // Row
1250    ); // Row
1251  }
1252
1253  Widget buildRoomInfo() {
1254    return Row(
1255      children: [
1256        Text( // Text ...
1257        const Icon( // Icon ...
1258      ], // Row
1259    ); // Row
1260  }
1261
1262  Widget buildRoomInfo() {
1263    return Row(
1264      children: [
1265        Text( // Text ...
1266        const Icon( // Icon ...
1267      ], // Row
1268    ); // Row
1269  }
1270
1271  Widget buildRoomInfo() {
1272    return Row(
1273      children: [
1274        Text( // Text ...
1275        const Icon( // Icon ...
1276      ], // Row
1277    ); // Row
1278  }
1279
1280  Widget buildRoomInfo() {
1281    return Row(
1282      children: [
1283        Text( // Text ...
1284        const Icon( // Icon ...
1285      ], // Row
1286    ); // Row
1287  }
1288
1289  Widget buildRoomInfo() {
1290    return Row(
1291      children: [
1292        Text( // Text ...
1293        const Icon( // Icon ...
1294      ], // Row
1295    ); // Row
1296  }
1297
1298  Widget buildRoomInfo() {
1299    return Row(
1300      children: [
1301        Text( // Text ...
1302        const Icon( // Icon ...
1303      ], // Row
1304    ); // Row
1305  }
1306
1307  Widget buildRoomInfo() {
1308    return Row(
1309      children: [
1310        Text( // Text ...
1311        const Icon( // Icon ...
1312      ], // Row
1313    ); // Row
1314  }
1315
1316  Widget buildRoomInfo() {
1317    return Row(
1318      children: [
1319        Text( // Text ...
1320        const Icon( // Icon ...
1321      ], // Row
1322    ); // Row
1323  }
1324
1325  Widget buildRoomInfo() {
1326    return Row(
1327      children: [
1328        Text( // Text ...
1329        const Icon( // Icon ...
1330      ], // Row
1331    ); // Row
1332  }
1333
1334  Widget buildRoomInfo() {
1335    return Row(
1336      children: [
1337        Text( // Text ...
1338        const Icon( // Icon ...
1339      ], // Row
1340    ); // Row
1341  }
1342
1343  Widget buildRoomInfo() {
1344    return Row(
1345      children: [
1346        Text( // Text ...
1347        const Icon( // Icon ...
1348      ], // Row
1349    ); // Row
1350  }
1351
1352  Widget buildRoomInfo() {
1353    return Row(
1354      children: [
1355        Text( // Text ...
1356        const Icon( // Icon ...
1357      ], // Row
1358    ); // Row
1359  }
1360
1361  Widget buildRoomInfo() {
1362    return Row(
1363      children: [
1364        Text( // Text ...
1365        const Icon( // Icon ...
1366      ], // Row
1367    ); // Row
1368  }
1369
1370  Widget buildRoomInfo() {
1371    return Row(
1372      children: [
1373        Text( // Text ...
1374        const Icon( // Icon ...
1375      ], // Row
1376    ); // Row
1377  }
1378
1379  Widget buildRoomInfo() {
1380    return Row(
1381      children: [
1382        Text( // Text ...
1383        const Icon( // Icon ...
1384      ], // Row
1385    ); // Row
1386  }
1387
1388  Widget buildRoomInfo() {
1389    return Row(
1390      children: [
1391        Text( // Text ...
1392        const Icon( // Icon ...
1393      ], // Row
1394    ); // Row
1395  }
1396
1397  Widget buildRoomInfo() {
1398    return Row(
1399      children: [
1400        Text( // Text ...
1401        const Icon( // Icon ...
1402      ], // Row
1403    ); // Row
1404  }
1405
1406  Widget buildRoomInfo() {
1407    return Row(
1408      children: [
1409        Text( // Text ...
1410        const Icon( // Icon ...
1411      ], // Row
1412    ); // Row
1413  }
1414
1415  Widget buildRoomInfo() {
1416    return Row(
1417      children: [
1418        Text( // Text ...
1419        const Icon( // Icon ...
1420      ], // Row
1421    ); // Row
1422  }
1423
1424  Widget buildRoomInfo() {
1425    return Row(
1426      children: [
1427        Text( // Text ...
1428        const Icon( // Icon ...
1429      ], // Row
1430    ); // Row
1431  }
1432
1433  Widget buildRoomInfo() {
1434    return Row(
1435      children: [
1436        Text( // Text ...
1437        const Icon( // Icon ...
1438      ], // Row
1439    ); // Row
1440  }
1441
1442  Widget buildRoomInfo() {
1443    return Row(
1444      children: [
144
```

### 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.



The screenshot displays two Dart files in a code editor. The left file, `profile_page.dart`, defines the `ProfilePage` class, which extends `StatelessWidget`. It includes a `build` method that returns a `SingleChildScrollView` containing a `Column` of widgets: a chat title, a list of chat items, and a text input field. The right file, `follower_page.dart`, defines the `FollowerPage` class, which also extends `StatelessWidget`. It includes a `build` method that returns a `SingleChildScrollView` containing a `Column` of widgets: a chat title, a list of chat items, and a text input field. The `buildAvailableChatList` method is also defined in this file, which returns a `ListView.builder` widget.

```
lib > pages > lobby > profile_page.dart > ...
1 import '../app_lib.dart';
2
3 class ProfilePage 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 }

lib > pages > lobby > follower_page.dart > ...
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

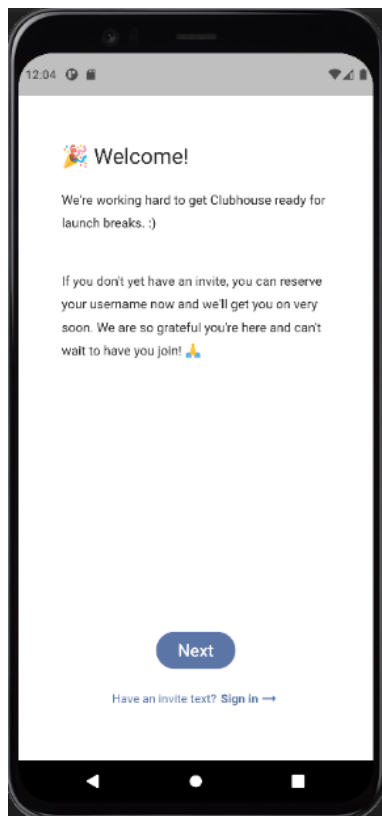


Fig 1.1

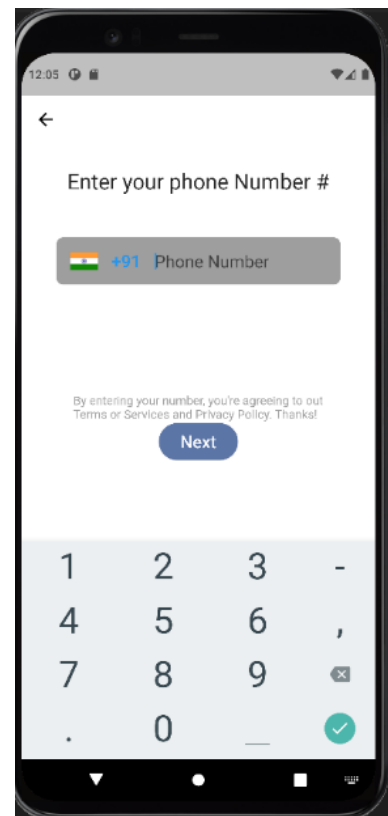


Fig 1.2



Fig 1.3



Fig 1.4



Fig 1.5



Fig 1.6



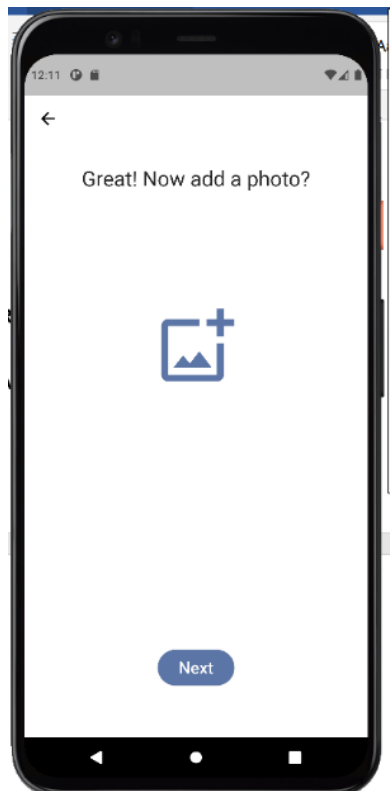


Fig 1.7

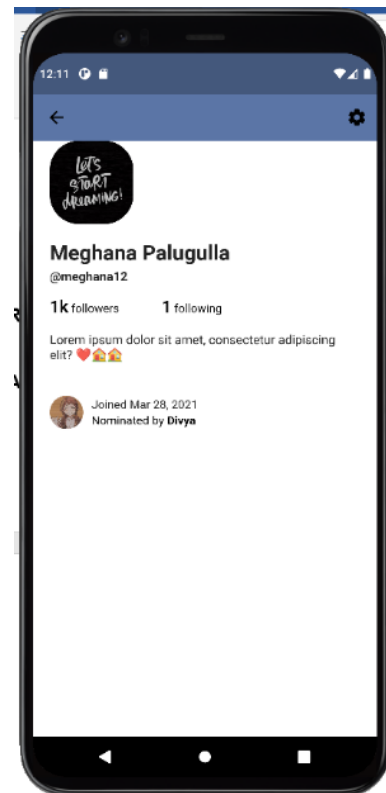


Fig 1.8

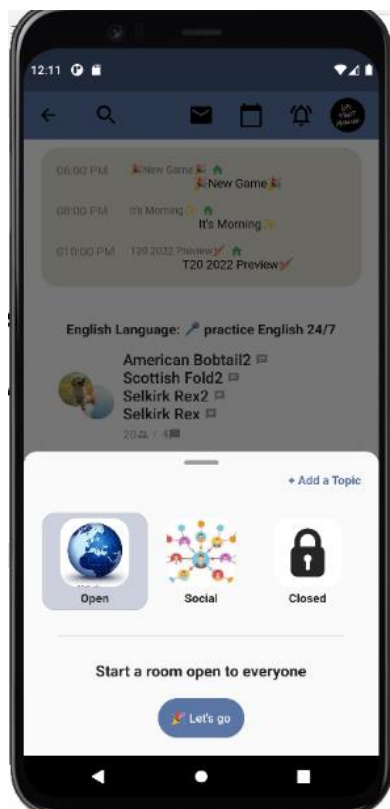


Fig 1.9

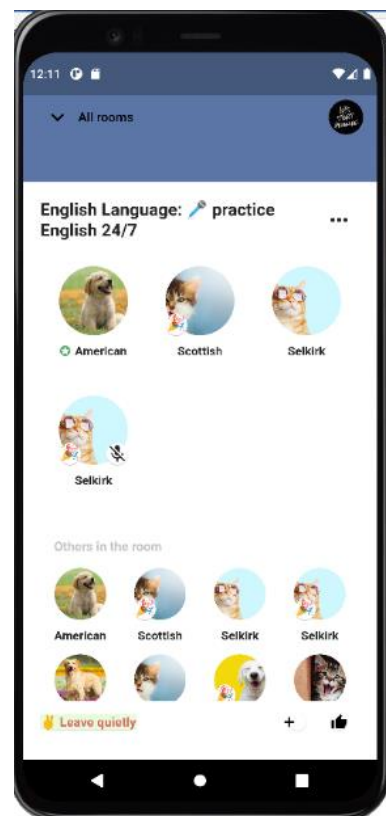


Fig 2.0

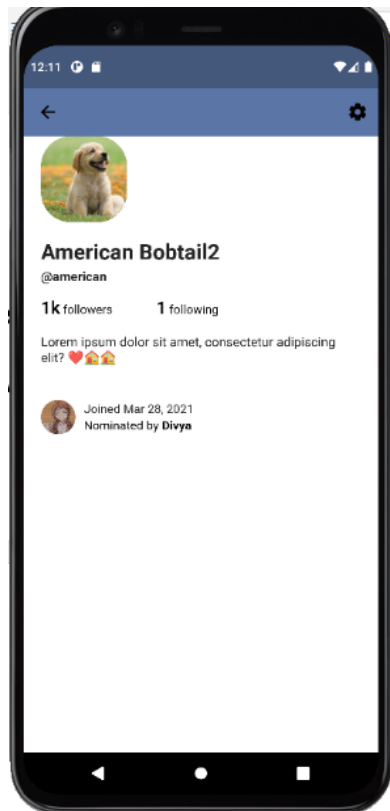


Fig 2.1

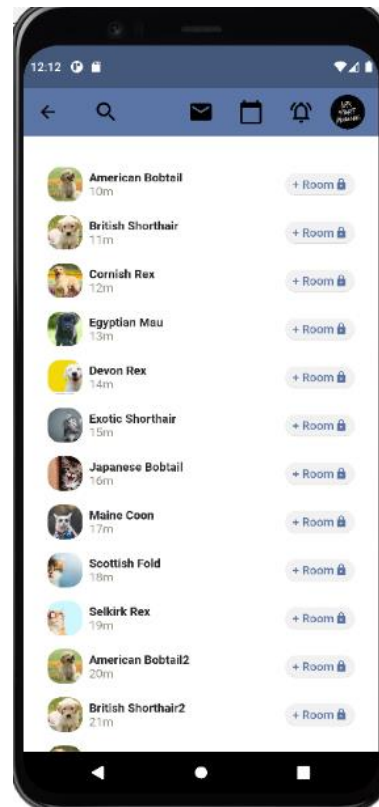


Fig 2.2

## 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)