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

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.Dharani

Head of the Department

Submitted for Viva voce Examination held on______________________

InternalExaminer **ExternalExaminer**

DECLARATION

I Palugulla Meghana here by declare that the Project Report entitled_CLUB HOUSE CLONE_done by me under the guidance of Dr.Ms.Manju C Nair 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

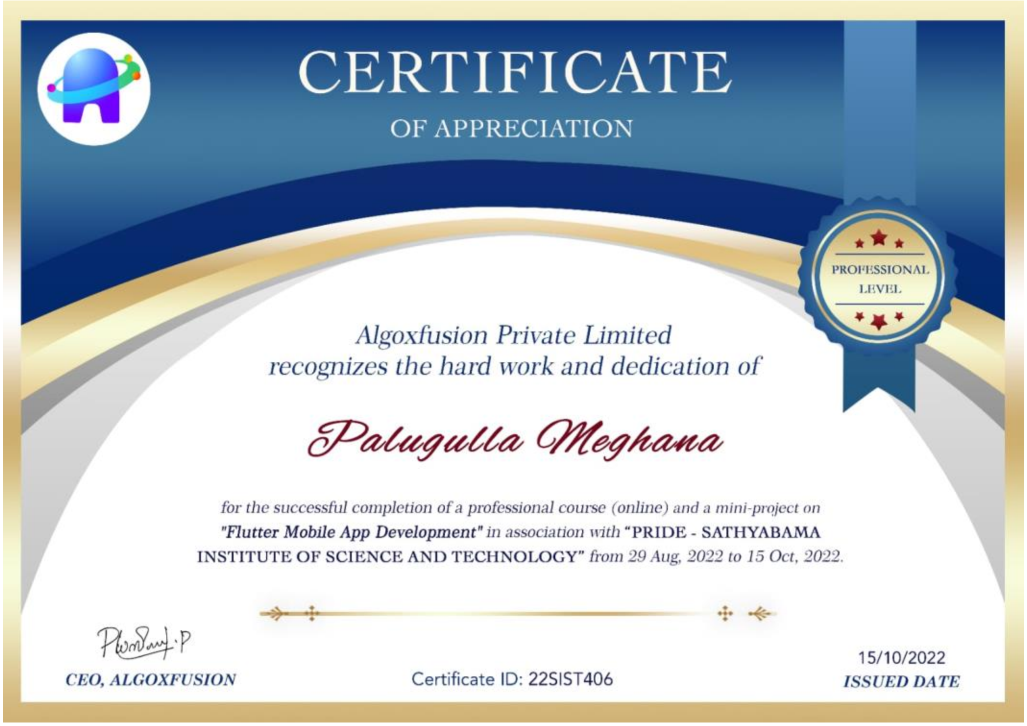
I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

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.Manju C Nair** 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



CERTIFICATE

OF APPRECIATION

PROFESSIONAL LEVEL

Algoxfusion Private Limited
recognizes the hard work and dedication of

Palugulla Meghana

for the successful completion of a professional course (online) and a mini-project on ***"Flutter Mobile App Development"*** *in association with* **"PRIDE - SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY"** *from 29 Aug, 2022 to 15 Oct, 2022.*

CEO, ALGOXFUSION

Certificate ID: 22SIST406

15/10/2022
ISSUED DATE

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

TABLE OF CONTENTS

CHAPTER NO **TITLE** **PAGE NO**

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:

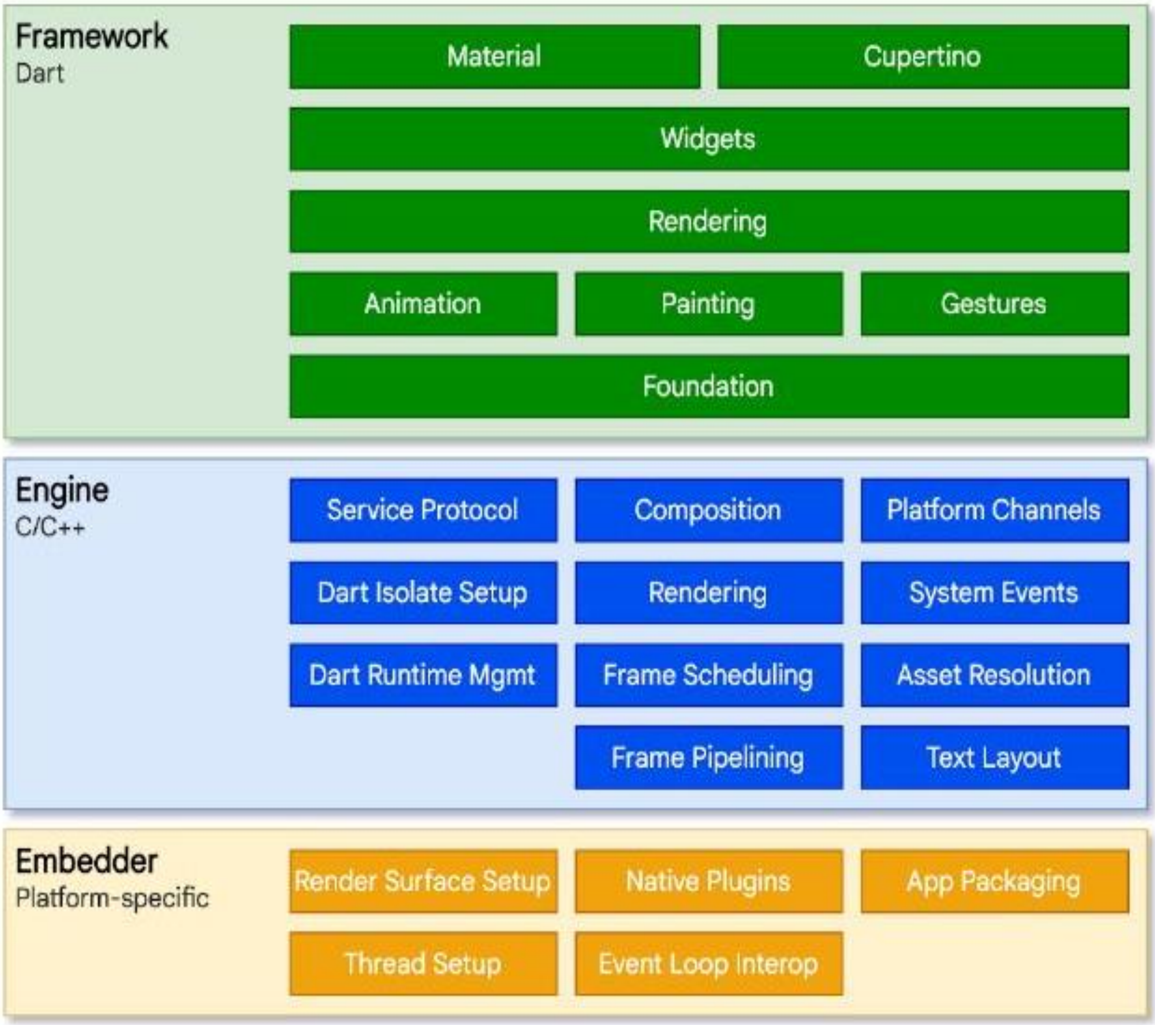
An SDK has a much wider scope as it’s a collection of tools, including libraries, documentation, APIs, sometimes frameworks, and more, giving you all you need for software development. And that’s the case with Flutter — it already contains everything necessary to build cross-platform applications.

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)

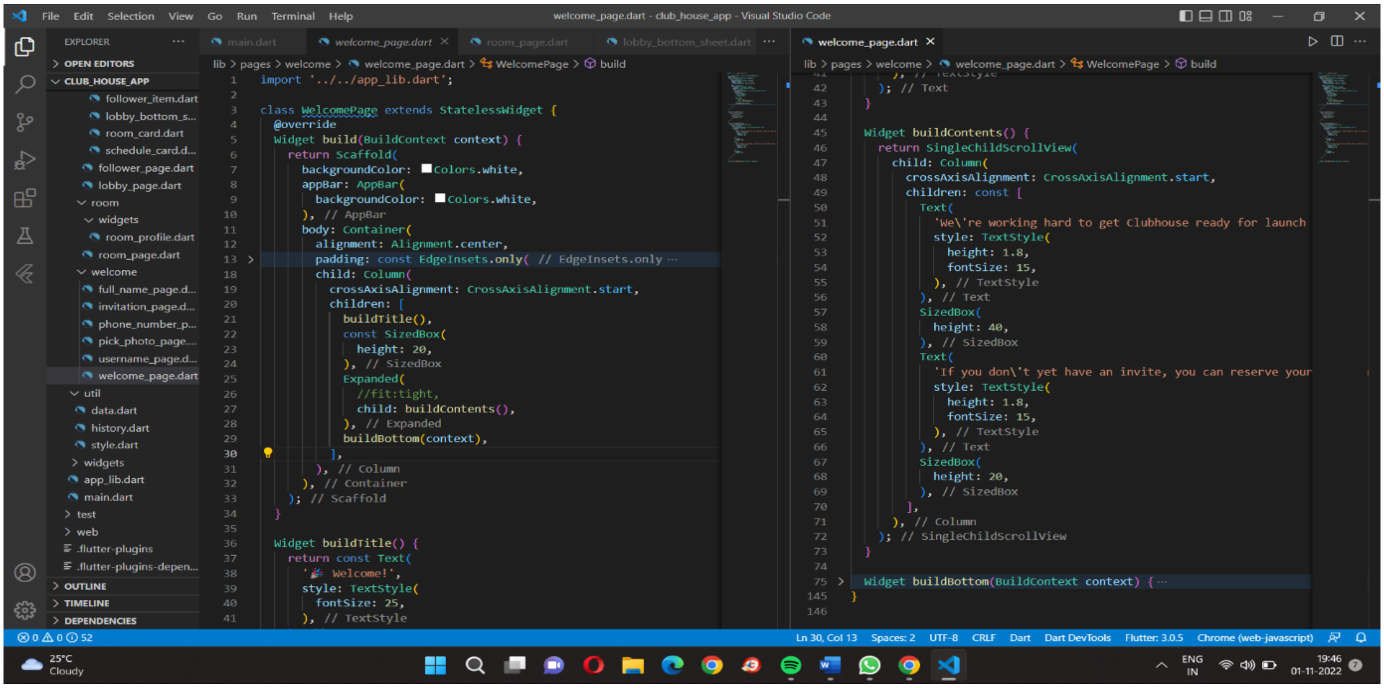
5. Implementation of the application

5.1 Welcome Page:

- Import Material Package to use the Built in Functionility

- Create a file welcomepage.dart And create a welcomepage class, It inherits The Properties of Stateless widget .
- Stateless widget: It is a static widget , which does not depend on Data or Behaviour change in runtime.
- Stateles widget has build() function which returns Scaffold
    - ▪ Scaffold contains:
        - ○ App bar
        - ○ Body

- Create A widget build contents() , buildtitle(),buildbottom().
- Call That Functions in Body Of Scaffold



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

class WelcomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
      ), // AppBar
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.only( // EdgeInsets.only ···
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            buildTitle(),
            const SizedBox(
              height: 20,
            ), // SizedBox
            Expanded(
              //fit:tight,
              child: buildContents(),
            ), // Expanded
            buildBottom(context),
          ],
        ), // Column
      ), // Container
    ); // Scaffold
  }

  Widget buildTitle() {
    return const Text(
      '🎉 Welcome!',
      style: TextStyle(
        fontSize: 25,
      ), // TextStyle
```

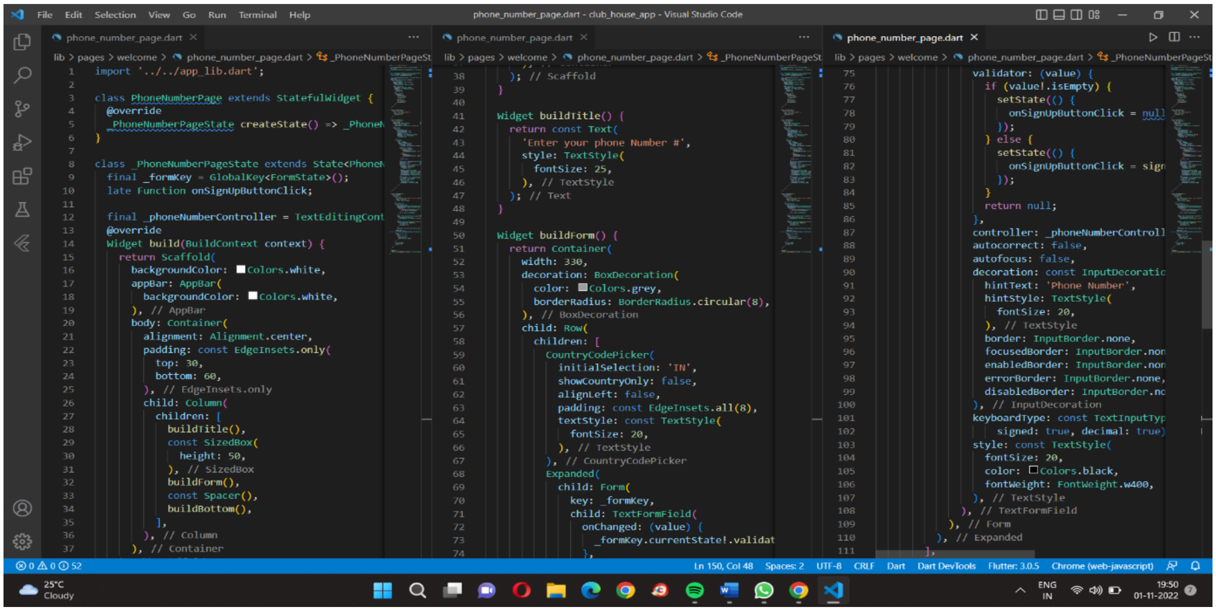
```
    ); // Text
  }

  Widget buildContents() {
    return SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: const [
          Text(
            'We\'re working hard to get Clubhouse ready for launch
            style: TextStyle(
              height: 1.8,
              fontSize: 15,
            ), // TextStyle
          ), // Text
          SizedBox(
            height: 40,
          ), // SizedBox
          Text(
            'If you don\'t yet have an invite, you can reserve your
            style: TextStyle(
              height: 1.8,
              fontSize: 15,
            ), // TextStyle
          ), // Text
          SizedBox(
            height: 20,
          ), // SizedBox
        ],
      ), // Column
    ); // SingleChildScrollView
  }

  Widget buildBottom(BuildContext context) {···
}
```

5.2 Phone Number Page:

- Create a file phonenumber.dart and Define a class phoneumber 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



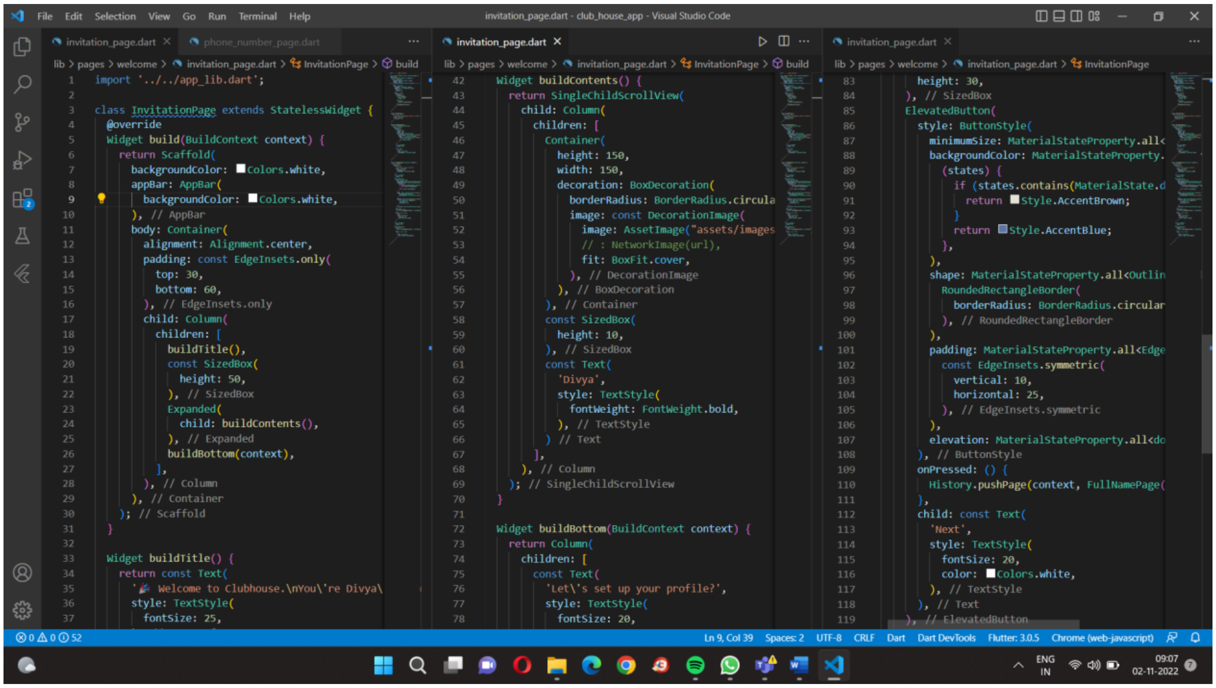
```dart
import '../../app_lib.dart';

class PhoneNumberPage extends StatefulWidget {
  @override
  _PhoneNumberPageState createState() => _PhoneN
}

class _PhoneNumberPageState extends State<PhoneN
  final _formKey = GlobalKey<FormState>();
  late Function onSignUpButtonClick;

  final _phoneNumberController = TextEditingCont
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
      ), // AppBar
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.only(
          top: 30,
          bottom: 60,
        ), // EdgeInsets.only
        child: Column(
          children: [
            buildTitle(),
            const SizedBox(
              height: 50,
            ), // SizedBox
            buildForm(),
            const Spacer(),
            buildBottom(),
          ],
        ), // Column
      ), // Container
```

```dart
    ); // Scaffold
  }

  Widget buildTitle() {
    return const Text(
      'Enter your phone Number #',
      style: TextStyle(
        fontSize: 25,
      ), // TextStyle
    ); // Text
  }

  Widget buildForm() {
    return Container(
      width: 330,
      decoration: BoxDecoration(
        color: Colors.grey,
        borderRadius: BorderRadius.circular(8),
      ), // BoxDecoration
      child: Row(
        children: [
          CountryCodePicker(
            initialSelection: 'IN',
            showCountryOnly: false,
            alignLeft: false,
            padding: const EdgeInsets.all(8),
            textStyle: const TextStyle(
              fontSize: 20,
            ), // TextStyle
          ), // CountryCodePicker
          Expanded(
            child: Form(
              key: _formKey,
              child: TextFormField(
                onChanged: (value) {
                  _formKey.currentState!.validat
                },
```

```dart
                validator: (value) {
                  if (value!.isEmpty) {
                    setState(() {
                      onSignUpButtonClick = null
                    });
                  } else {
                    setState(() {
                      onSignUpButtonClick = sign
                    });
                  }
                  return null;
                },
                controller: _phoneNumberControll
                autocorrect: false,
                autofocus: false,
                decoration: const InputDecoratio
                  hintText: 'Phone Number',
                  hintStyle: TextStyle(
                    fontSize: 20,
                  ), // TextStyle
                  border: InputBorder.none,
                  focusedBorder: InputBorder.non
                  enabledBorder: InputBorder.non
                  errorBorder: InputBorder.none,
                  disabledBorder: InputBorder.nc
                ), // InputDecoration
                keyboardType: const TextInputTyp
                    signed: true, decimal: true)
                style: const TextStyle(
                  fontSize: 20,
                  color: Colors.black,
                  fontWeight: FontWeight.w400,
                ), // TextStyle
              ), // TextFormField
            ), // Form
          ), // Expanded
        ],
```

- **5.3 Invitation Page:**
- Import Material Package to use the Built in Functionility

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



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

class InvitationPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Colors.white,
      ), // AppBar
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.only(
          top: 30,
          bottom: 60,
        ), // EdgeInsets.only
        child: Column(
          children: [
            buildTitle(),
            const SizedBox(
              height: 50,
            ), // SizedBox
            Expanded(
              child: buildContents(),
            ), // Expanded
            buildBottom(context),
          ],
        ), // Column
      ), // Container
    ); // Scaffold
  }

  Widget buildTitle() {
    return const Text(
      '🎉 Welcome to Clubhouse.\nYou\'re Divya\
      style: TextStyle(
        fontSize: 25,
```

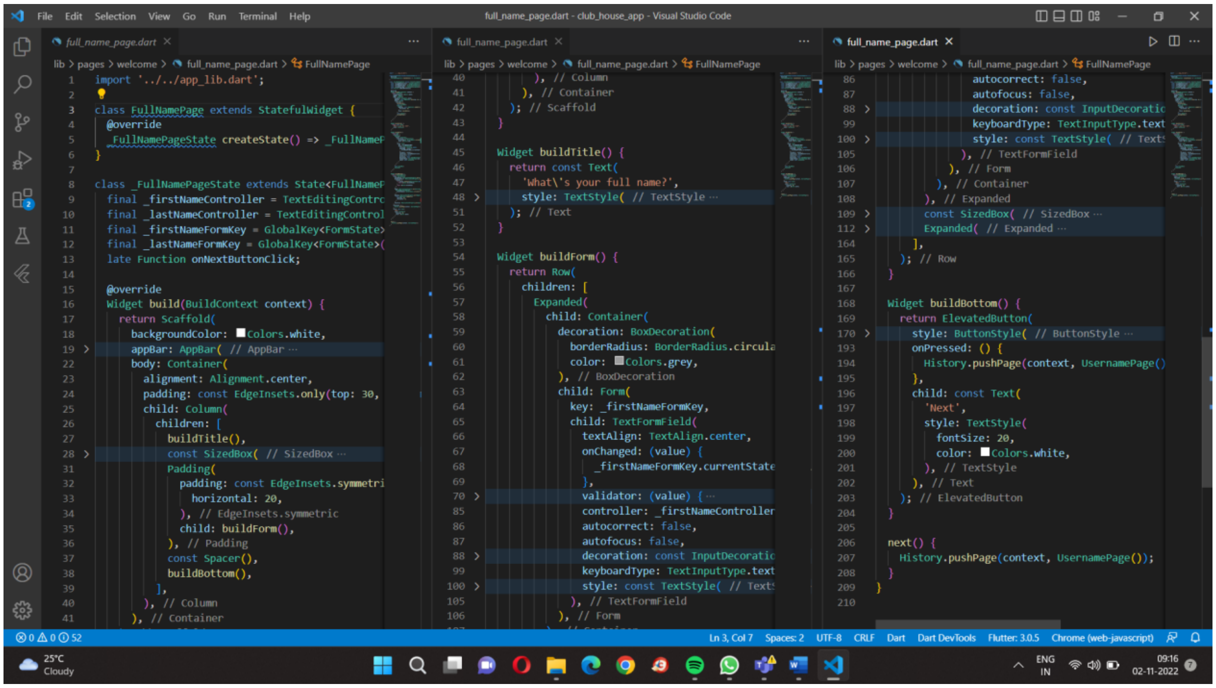
```
  Widget buildContents() {
    return SingleChildScrollView(
      child: Column(
        children: [
          Container(
            height: 150,
            width: 150,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circula
              image: const DecorationImage(
                image: AssetImage("assets/images
                // : NetworkImage(url),
                fit: BoxFit.cover,
              ), // DecorationImage
            ), // BoxDecoration
          ), // Container
          const SizedBox(
            height: 10,
          ), // SizedBox
          const Text(
            'Divya',
            style: TextStyle(
              fontWeight: FontWeight.bold,
            ), // TextStyle
          ) // Text
        ],
      ), // Column
    ); // SingleChildScrollView
  }

  Widget buildBottom(BuildContext context) {
    return Column(
      children: [
        const Text(
          'Let\'s set up your profile?',
          style: TextStyle(
            fontSize: 20,
```

```
          height: 30,
        ), // SizedBox
        ElevatedButton(
          style: ButtonStyle(
            minimumSize: MaterialStateProperty.all<
            backgroundColor: MaterialStateProperty.
              (states) {
                if (states.contains(MaterialState.d
                  return Style.AccentBrown;
                }
                return Style.AccentBlue;
              },
            ),
            shape: MaterialStateProperty.all<Outlin
              RoundedRectangleBorder(
                borderRadius: BorderRadius.circular
              ), // RoundedRectangleBorder
            ),
            padding: MaterialStateProperty.all<Edge
              const EdgeInsets.symmetric(
                vertical: 10,
                horizontal: 25,
              ), // EdgeInsets.symmetric
            ),
            elevation: MaterialStateProperty.all<do
          ), // ButtonStyle
          onPressed: () {
            History.pushPage(context, FullNamePage(
          },
          child: const Text(
            'Next',
            style: TextStyle(
              fontSize: 20,
              color: Colors.white,
            ), // TextStyle
          ), // Text
        ), // ElevatedButton
```

5.5 Full Name Page

Import Material Package to use the Built in Functionility

- Create a file fullNamePage.dart And create a class fullNamePage, It inherits The Properties of Stateless widget.
- Build() function returns Scaffold.scaffold consits 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.



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

class FullNamePage extends StatefulWidget {
  @override
  _FullNamePageState createState() => _FullNameP
}

class _FullNamePageState extends State<FullNameP
  final _firstNameController = TextEditingContro
  final _lastNameController = TextEditingControl
  final _firstNameFormKey = GlobalKey<FormState>
  final _lastNameFormKey = GlobalKey<FormState>(
  late Function onNextButtonClick;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar( // AppBar ···
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.only(top: 30,
        child: Column(
          children: [
            buildTitle(),
            const SizedBox( // SizedBox ···
            Padding(
              padding: const EdgeInsets.symmetri
                horizontal: 20,
              ), // EdgeInsets.symmetric
              child: buildForm(),
            ), // Padding
            const Spacer(),
            buildBottom(),
          ],
        ), // Column
      ), // Container
```

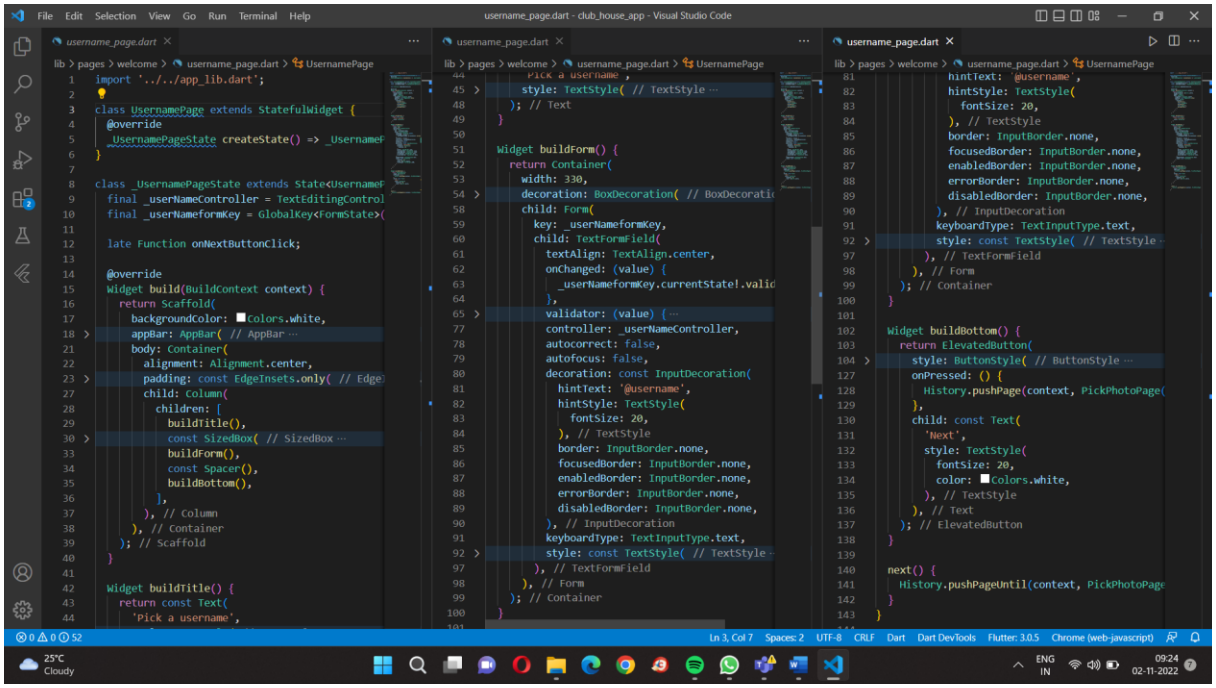
```dart
        ), // Column
      ), // Container
    ); // Scaffold
  }

  Widget buildTitle() {
    return const Text(
      'What\'s your full name?',
      style: TextStyle( // TextStyle ···
    ); // Text
  }

  Widget buildForm() {
    return Row(
      children: [
        Expanded(
          child: Container(
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circula
              color: Colors.grey,
            ), // BoxDecoration
            child: Form(
              key: _firstNameFormKey,
              child: TextFormField(
                textAlign: TextAlign.center,
                onChanged: (value) {
                  _firstNameFormKey.currentState
                },
                validator: (value) {···
                controller: _firstNameController
                autocorrect: false,
                autofocus: false,
                decoration: const InputDecoratio
                keyboardType: TextInputType.text
                style: const TextStyle( // TextS
              ), // TextFormField
            ), // Form
```

```dart
                autocorrect: false,
                autofocus: false,
                decoration: const InputDecoratio
                keyboardType: TextInputType.text
                style: const TextStyle( // TextS
              ), // TextFormField
            ), // Form
          ), // Container
        ), // Expanded
        const SizedBox( // SizedBox ···
        Expanded( // Expanded ···
      ],
    ); // Row
  }

  Widget buildBottom() {
    return ElevatedButton(
      style: ButtonStyle( // ButtonStyle ···
      onPressed: () {
        History.pushPage(context, UsernamePage()
      },
      child: const Text(
        'Next',
        style: TextStyle(
          fontSize: 20,
          color: Colors.white,
        ), // TextStyle
      ), // Text
    ); // ElevatedButton
  }

  next() {
    History.pushPage(context, UsernamePage());
  }
}
```

5.6 User Name Page

Import Material Package to use the Built in Functionility

- Create a file userNamePage.dart And create a class userNamePage, It inherits The Properties of Stateful widget.
- Build() function returns Scaffold.scaffold consits 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.


File Edit Selection View Go Run Terminal Help
username_page.dart - club_house_app - Visual Studio Code
username_page.dart
lib > pages > welcome > username_page.dart > UsernamePage
⊗ 0 ⚠ 0 ⓘ 52
Ln 3, Col 7 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 3.0.5 Chrome (web-javascript)
25°C Cloudy
ENG IN 09:24 02-11-2022

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

class UsernamePage extends StatefulWidget {
  @override
  _UsernamePageState createState() => _UsernameP
}

class _UsernamePageState extends State<UsernameP
  final _userNameController = TextEditingControl
  final _userNameformKey = GlobalKey<FormState>(

  late Function onNextButtonClick;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar( // AppBar ···
      body: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.only( // Edge
        child: Column(
          children: [
            buildTitle(),
            const SizedBox( // SizedBox ···
            buildForm(),
            const Spacer(),
            buildBottom(),
          ],
        ), // Column
      ), // Container
    ); // Scaffold
  }

  Widget buildTitle() {
    return const Text(
      'Pick a username',
```

```
      'Pick a username',
      style: TextStyle( // TextStyle ···
    ); // Text
  }

  Widget buildForm() {
    return Container(
      width: 330,
      decoration: BoxDecoration( // BoxDecoratio
      child: Form(
        key: _userNameformKey,
        child: TextFormField(
          textAlign: TextAlign.center,
          onChanged: (value) {
            _userNameformKey.currentState!.valid
          },
          validator: (value) {···
          controller: _userNameController,
          autocorrect: false,
          autofocus: false,
          decoration: const InputDecoration(
            hintText: '@username',
            hintStyle: TextStyle(
              fontSize: 20,
            ), // TextStyle
            border: InputBorder.none,
            focusedBorder: InputBorder.none,
            enabledBorder: InputBorder.none,
            errorBorder: InputBorder.none,
            disabledBorder: InputBorder.none,
          ), // InputDecoration
          keyboardType: TextInputType.text,
          style: const TextStyle( // TextStyle ···
        ), // TextFormField
      ), // Form
    ); // Container
  }
```

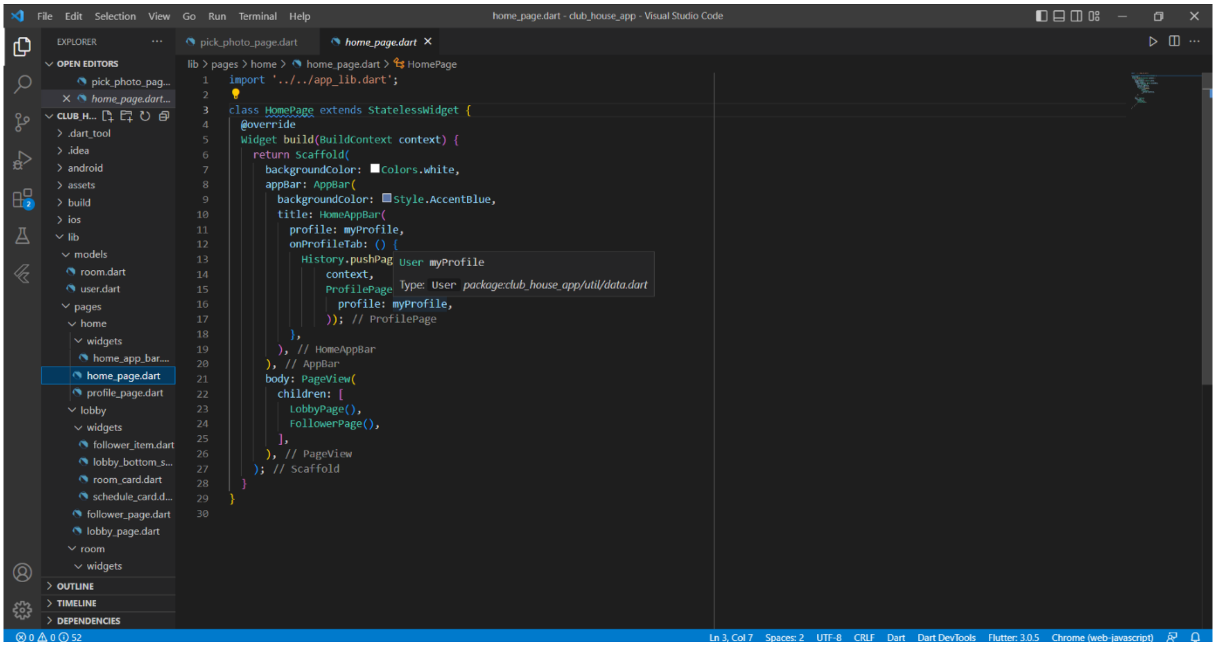
```
            hintText: '@username',
            hintStyle: TextStyle(
              fontSize: 20,
            ), // TextStyle
            border: InputBorder.none,
            focusedBorder: InputBorder.none,
            enabledBorder: InputBorder.none,
            errorBorder: InputBorder.none,
            disabledBorder: InputBorder.none,
          ), // InputDecoration
          keyboardType: TextInputType.text,
          style: const TextStyle( // TextStyle ···
        ), // TextFormField
      ), // Form
    ); // Container
  }

  Widget buildBottom() {
    return ElevatedButton(
      style: ButtonStyle( // ButtonStyle ···
      onPressed: () {
        History.pushPage(context, PickPhotoPage(
      },
      child: const Text(
        'Next',
        style: TextStyle(
          fontSize: 20,
          color: Colors.white,
        ), // TextStyle
      ), // Text
    ); // ElevatedButton
  }

  next() {
    History.pushPageUntil(context, PickPhotoPage
  }
}
```

-

5.8 Home Page

Import Material Package to use the Built in Functionility

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



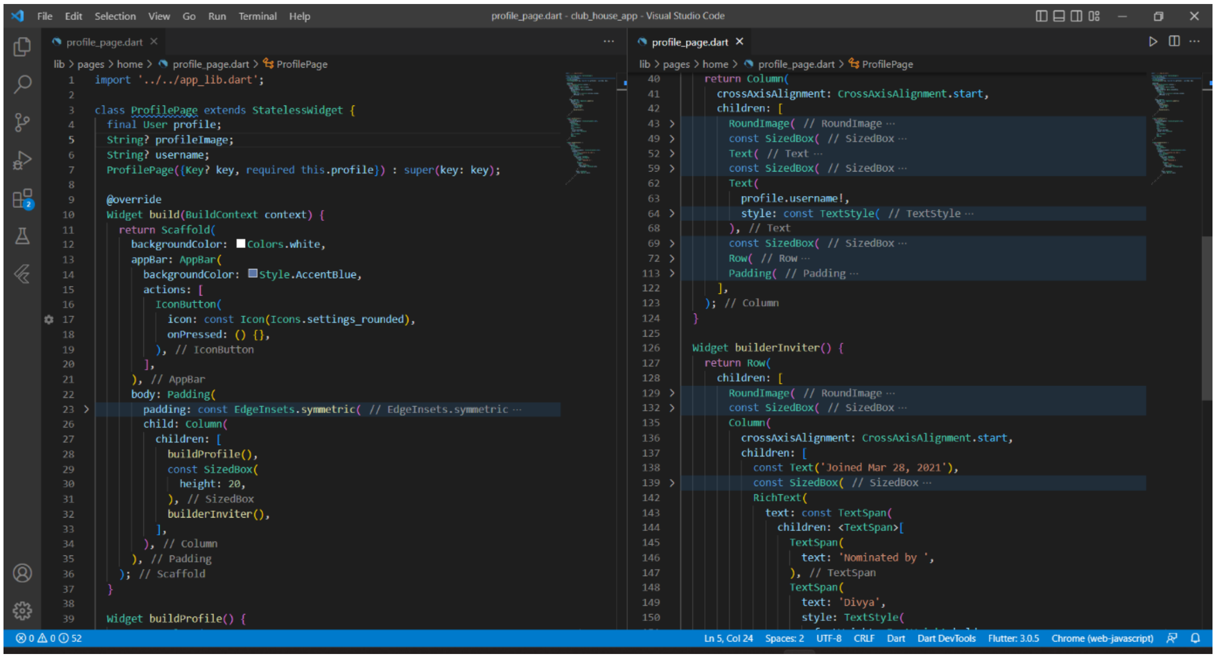
```
import '../../app_lib.dart';

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

```
User myProfile
Type: User package:club_house_app/util/data.dart
```

5.8.1 Profile Page

Import Material Package to use the Built in Functionility

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



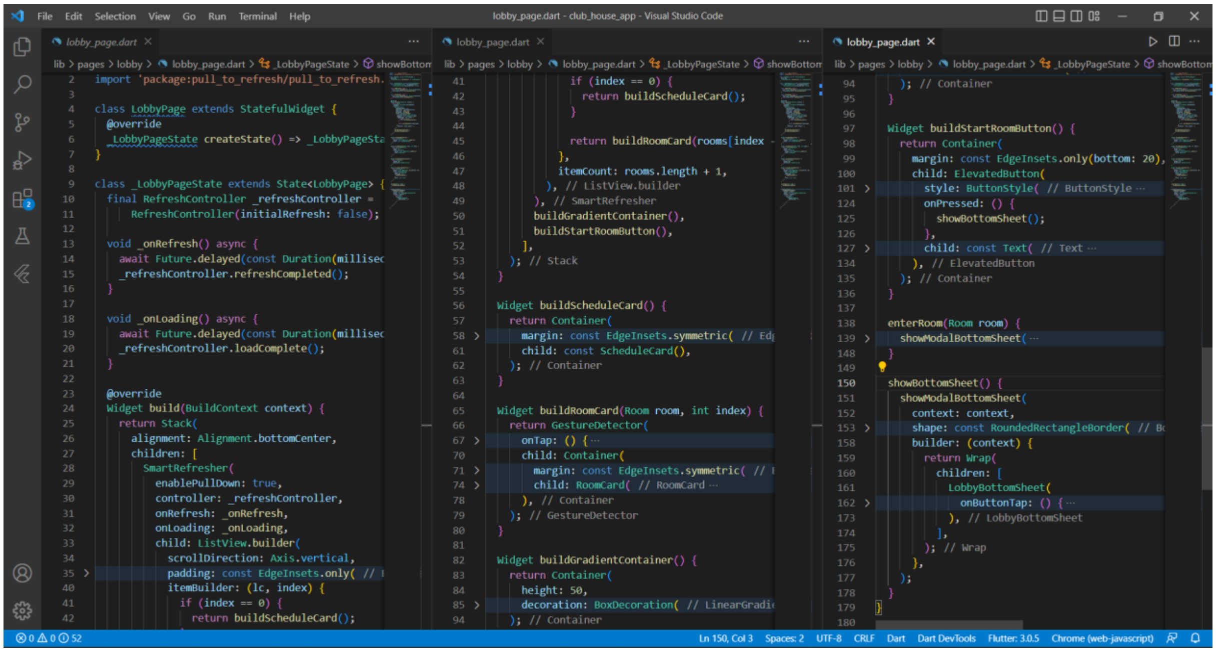
```dart
import '../../app_lib.dart';

class ProfilePage extends StatelessWidget {
  final User profile;
  String? profileImage;
  String? username;
  ProfilePage({Key? key, required this.profile}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        backgroundColor: Style.AccentBlue,
        actions: [
          IconButton(
            icon: const Icon(Icons.settings_rounded),
            onPressed: () {},
          ), // IconButton
        ],
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ···
        child: Column(
          children: [
            buildProfile(),
            const SizedBox(
              height: 20,
            ), // SizedBox
            builderInviter(),
          ],
        ), // Column
      ), // Padding
    ); // Scaffold
  }

  Widget buildProfile() {
```

```dart
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        RoundImage( // RoundImage ···
        const SizedBox( // SizedBox ···
        Text( // Text ···
        const SizedBox( // SizedBox ···
        Text(
          profile.username!,
          style: const TextStyle( // TextStyle ···
        ), // Text
        const SizedBox( // SizedBox ···
        Row( // Row ···
        Padding( // Padding ···
      ],
    ); // Column
  }

  Widget builderInviter() {
    return Row(
      children: [
        RoundImage( // RoundImage ···
        const SizedBox( // SizedBox ···
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text('Joined Mar 28, 2021'),
            const SizedBox( // SizedBox ···
            RichText(
              text: const TextSpan(
                children: <TextSpan>[
                  TextSpan(
                    text: 'Nominated by ',
                  ), // TextSpan
                  TextSpan(
                    text: 'Divya',
                    style: TextStyle(
```

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 Stateful widget.
- Build() function returns stack.stack consists of Body

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



```
import 'package:pull_to_refresh/pull_to_refresh.

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

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

  void _onRefresh() async {
    await Future.delayed(const Duration(millisec
    _refreshController.refreshCompleted();
  }

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

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

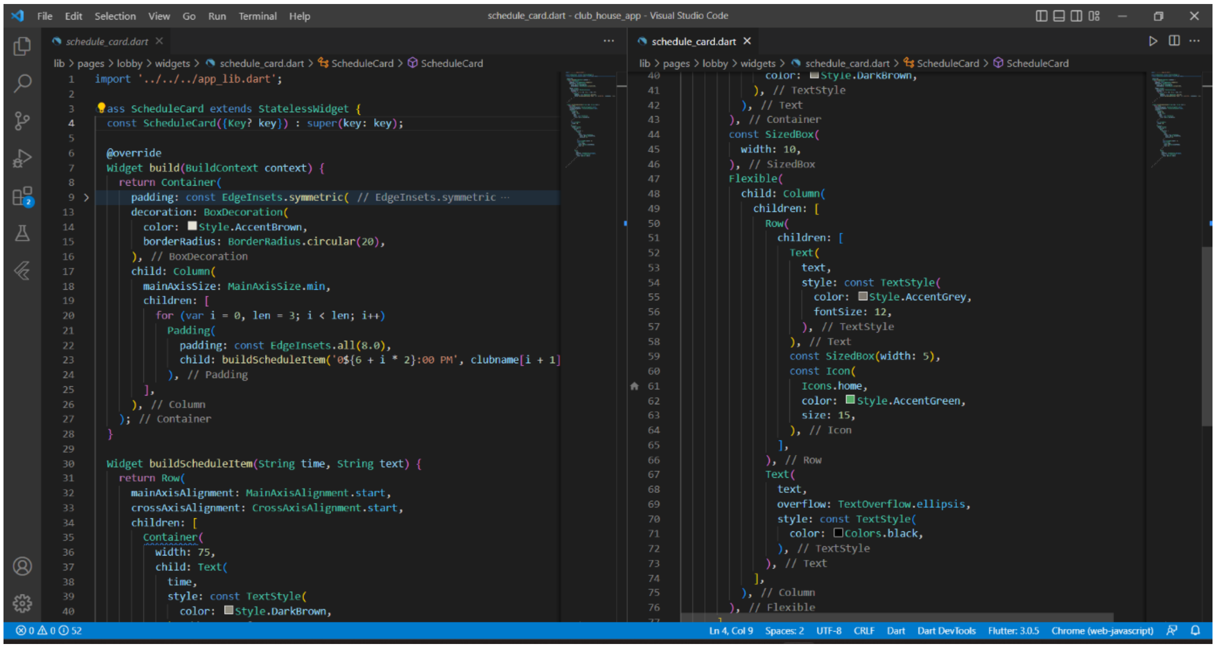
```
              if (index == 0) {
                return buildScheduleCard();
              }

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

  Widget buildScheduleCard() {
    return Container(
      margin: const EdgeInsets.symmetric( // Edg
      child: const ScheduleCard(),
    ); // Container
  }

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

  Widget buildGradientContainer() {
    return Container(
      height: 50,
      decoration: BoxDecoration( // LinearGradie
    ); // Container
```

```
    ); // Container
  }

  Widget buildStartRoomButton() {
    return Container(
      margin: const EdgeInsets.only(bottom: 20),
      child: ElevatedButton(
        style: ButtonStyle( // ButtonStyle ···
        onPressed: () {
          showBottomSheet();
        },
        child: const Text( // Text ···
      ), // ElevatedButton
    ); // Container
  }

  enterRoom(Room room) {
    showModalBottomSheet(···
  }

  showBottomSheet() {
    showModalBottomSheet(
      context: context,
      shape: const RoundedRectangleBorder( // Bo
      builder: (context) {
        return Wrap(
          children: [
            LobbyBottomSheet(
              onButtonTap: () {···
            ), // LobbyBottomSheet
          ],
        ); // Wrap
      },
    );
  }
}
```

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

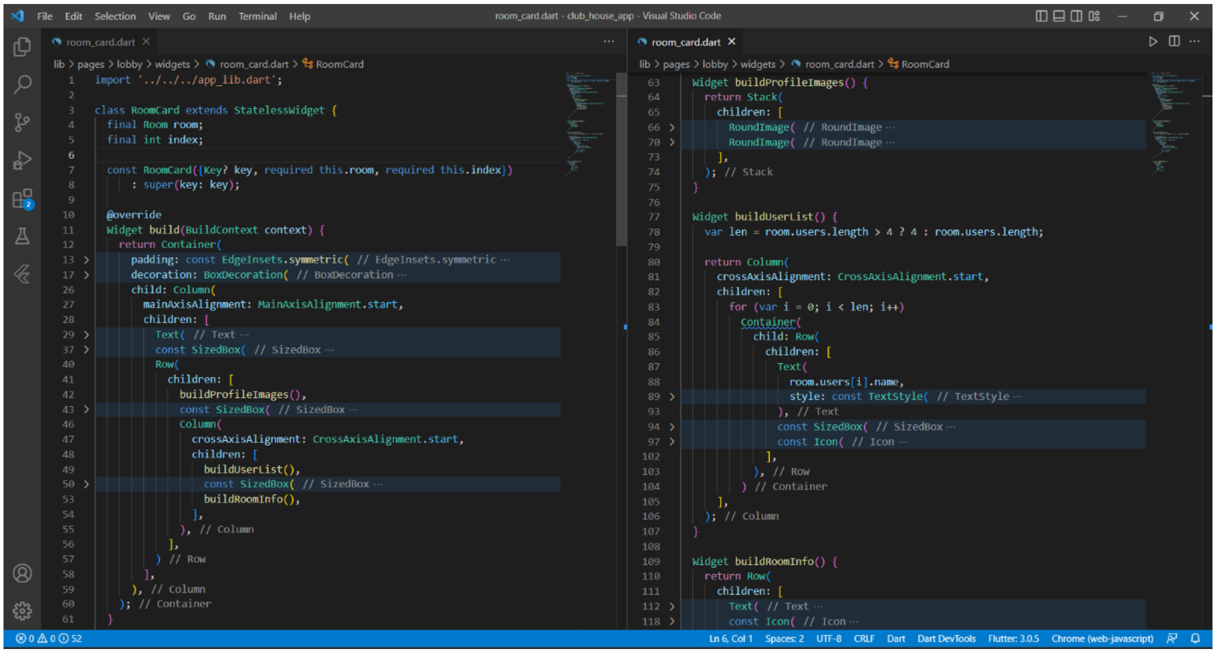

File Edit Selection View Go Run Terminal Help
schedule_card.dart - club_house_app - Visual Studio Code
schedule_card.dart
lib > pages > lobby > widgets > schedule_card.dart > ScheduleCard > ScheduleCard
⊗ 0 ⚠ 0 ⓘ 52
Ln 4, Col 9 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 3.0.5 Chrome (web-javascript)

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

ass ScheduleCard extends StatelessWidget {
  const ScheduleCard({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ···
      decoration: BoxDecoration(
        color: Style.AccentBrown,
        borderRadius: BorderRadius.circular(20),
      ), // BoxDecoration
      child: Column(
        mainAxisSize: MainAxisSize.min,
        children: [
          for (var i = 0, len = 3; i < len; i++)
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: buildScheduleItem('0${6 + i * 2}:00 PM', clubname[i + 1]
            ), // Padding
        ],
      ), // Column
    ); // Container
  }

  Widget buildScheduleItem(String time, String text) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Container(
          width: 75,
          child: Text(
            time,
            style: const TextStyle(
              color: Style.DarkBrown,
            ), // TextStyle
          ), // Text
        ), // Container
        const SizedBox(
          width: 10,
        ), // SizedBox
        Flexible(
          child: Column(
            children: [
              Row(
                children: [
                  Text(
                    text,
                    style: const TextStyle(
                      color: Style.AccentGrey,
                      fontSize: 12,
                    ), // TextStyle
                  ), // Text
                  const SizedBox(width: 5),
                  const Icon(
                    Icons.home,
                    color: Style.AccentGreen,
                    size: 15,
                  ), // Icon
                ],
              ), // Row
              Text(
                text,
                overflow: TextOverflow.ellipsis,
                style: const TextStyle(
                  color: Colors.black,
                ), // TextStyle
              ), // Text
            ],
          ), // Column
        ), // Flexible
```

5.8.2.2 Room Card

Import Material Package to use the Built in Functionility

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



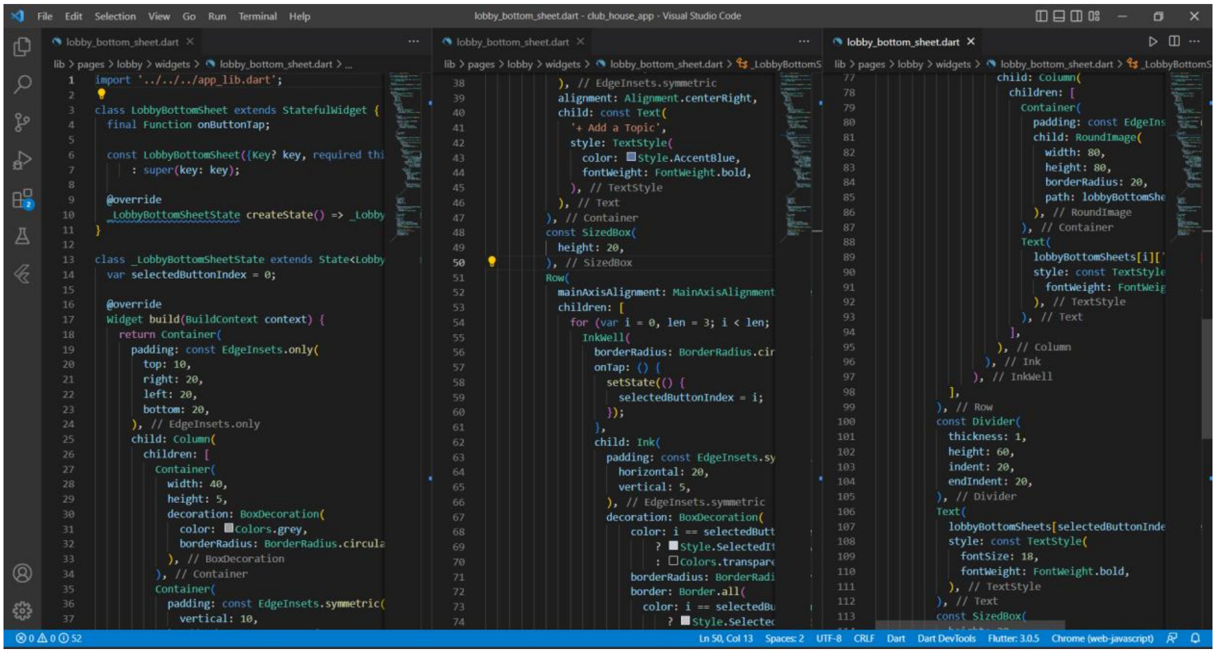
File
Edit
Selection
View
Go
Run
Terminal
Help
room_card.dart - club_house_app - Visual Studio Code
room_card.dart
lib > pages > lobby > widgets > room_card.dart > RoomCard
room_card.dart
lib > pages > lobby > widgets > room_card.dart > RoomCard
0 0 52
Ln 6, Col 1
Spaces: 2
UTF-8
CRLF
Dart
Dart DevTools
Flutter: 3.0.5
Chrome (web-javascript)

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

class RoomCard extends StatelessWidget {
  final Room room;
  final int index;

  const RoomCard({Key? key, required this.room, required this.index})
      : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.symmetric( // EdgeInsets.symmetric ···
      decoration: BoxDecoration( // BoxDecoration ···
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Text( // Text ···
          const SizedBox( // SizedBox ···
          Row(
            children: [
              buildProfileImages(),
              const SizedBox( // SizedBox ···
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  buildUserList(),
                  const SizedBox( // SizedBox ···
                  buildRoomInfo(),
                ],
              ), // Column
            ],
          ) // Row
        ],
      ), // Column
    ); // Container
  }
```

```
  Widget buildProfileImages() {
    return Stack(
      children: [
        RoundImage( // RoundImage ···
        RoundImage( // RoundImage ···
      ],
    ); // Stack
  }

  Widget buildUserList() {
    var len = room.users.length > 4 ? 4 : room.users.length;

    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        for (var i = 0; i < len; i++)
          Container(
            child: Row(
              children: [
                Text(
                  room.users[i].name,
                  style: const TextStyle( // TextStyle ···
                ), // Text
                const SizedBox( // SizedBox ···
                const Icon( // Icon ···
              ],
            ), // Row
          ) // Container
      ],
    ); // Column
  }

  Widget buildRoomInfo() {
    return Row(
      children: [
        Text( // Text ···
        const Icon( // Icon ···
```

5.8.2.3 LobbyBottomSheet

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


File Edit Selection View Go Run Terminal Help
lobby_bottom_sheet.dart - club_house_app - Visual Studio Code
lobby_bottom_sheet.dart
lib > pages > lobby > widgets > lobby_bottom_sheet.dart > ...
lib > pages > lobby > widgets > lobby_bottom_sheet.dart > _LobbyBottomS
Ln 50, Col 13 Spaces: 2 UTF-8 CRLF Dart Dart DevTools Flutter: 3.0.5 Chrome (web-javascript)
0 0 52

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

class LobbyBottomSheet extends StatefulWidget {
  final Function onButtonTap;

  const LobbyBottomSheet({Key? key, required thi
      : super(key: key);

  @override
  _LobbyBottomSheetState createState() => _Lobby
}

class _LobbyBottomSheetState extends State<Lobby
  var selectedButtonIndex = 0;

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.only(
        top: 10,
        right: 20,
        left: 20,
        bottom: 20,
      ), // EdgeInsets.only
      child: Column(
        children: [
          Container(
            width: 40,
            height: 5,
            decoration: BoxDecoration(
              color: ■Colors.grey,
              borderRadius: BorderRadius.circula
            ), // BoxDecoration
          ), // Container
          Container(
            padding: const EdgeInsets.symmetric(
              vertical: 10,
```

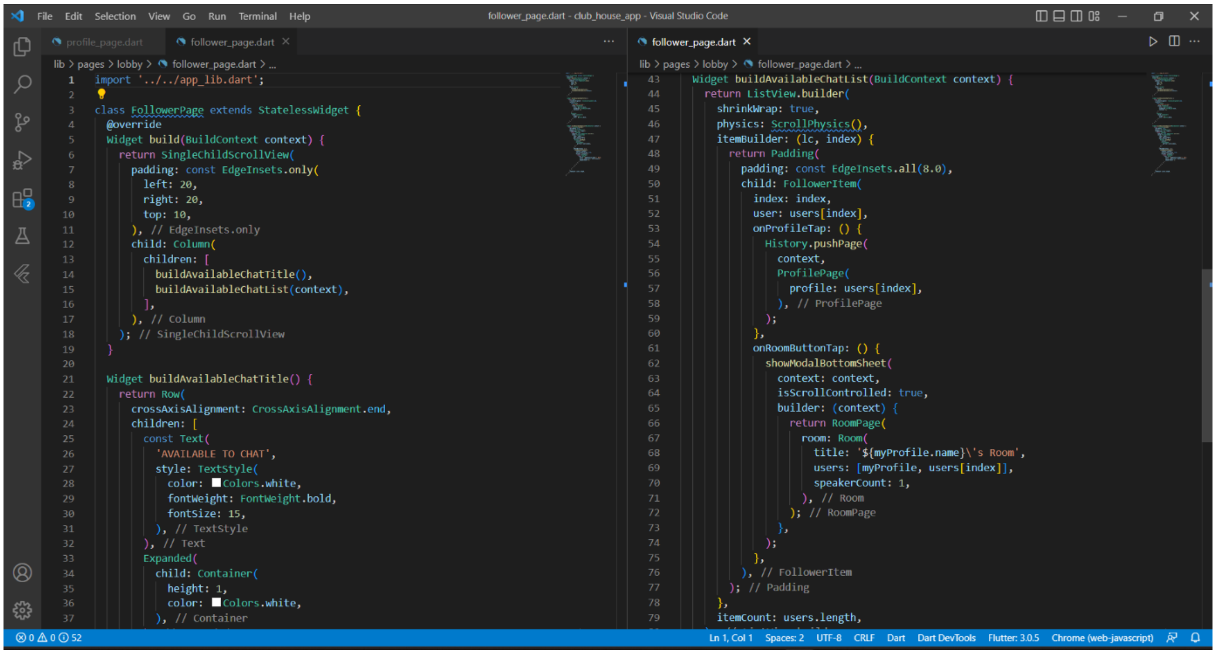
```
            ), // EdgeInsets.symmetric
            alignment: Alignment.centerRight,
            child: const Text(
              '+ Add a Topic',
              style: TextStyle(
                color: ■Style.AccentBlue,
                fontWeight: FontWeight.bold,
              ), // TextStyle
            ), // Text
          ), // Container
          const SizedBox(
            height: 20,
          ), // SizedBox
          Row(
            mainAxisAlignment: MainAxisAlignment
            children: [
              for (var i = 0, len = 3; i < len;
                InkWell(
                  borderRadius: BorderRadius.cir
                  onTap: () {
                    setState(() {
                      selectedButtonIndex = i;
                    });
                  },
                  child: Ink(
                    padding: const EdgeInsets.sy
                      horizontal: 20,
                      vertical: 5,
                    ), // EdgeInsets.symmetric
                    decoration: BoxDecoration(
                        color: i == selectedButt
                            ? ■Style.SelectedIt
                            : □Colors.transpare
                        borderRadius: BorderRadi
                        border: Border.all(
                          color: i == selectedBu
                              ? ■Style.Selectec
```

```
                    child: Column(
                      children: [
                        Container(
                          padding: const EdgeIns
                          child: RoundImage(
                            width: 80,
                            height: 80,
                            borderRadius: 20,
                            path: lobbyBottomShe
                          ), // RoundImage
                        ), // Container
                        Text(
                          lobbyBottomSheets[i]['
                          style: const TextStyle
                            fontWeight: FontWeig
                          ), // TextStyle
                        ), // Text
                      ],
                    ), // Column
                  ), // Ink
                ), // InkWell
            ],
          ), // Row
          const Divider(
            thickness: 1,
            height: 60,
            indent: 20,
            endIndent: 20,
          ), // Divider
          Text(
            lobbyBottomSheets[selectedButtonInde
            style: const TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ), // TextStyle
          ), // Text
          const SizedBox(
```

5.8.3 Follower Page

Import Material Package to use the Built in Functionility

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



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

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

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

```dart
  Widget buildAvailableChatList(BuildContext context) {
    return ListView.builder(
      shrinkWrap: true,
      physics: ScrollPhysics(),
      itemBuilder: (lc, index) {
        return Padding(
          padding: const EdgeInsets.all(8.0),
          child: FollowerItem(
            index: index,
            user: users[index],
            onProfileTap: () {
              History.pushPage(
                context,
                ProfilePage(
                  profile: users[index],
                ), // ProfilePage
              );
            },
            onRoomButtonTap: () {
              showModalBottomSheet(
                context: context,
                isScrollControlled: true,
                builder: (context) {
                  return RoomPage(
                    room: Room(
                      title: '${myProfile.name}\'s Room',
                      users: [myProfile, users[index]],
                      speakerCount: 1,
                    ), // Room
                  ); // RoomPage
                },
              );
            },
          ), // FollowerItem
        ); // Padding
      },
      itemCount: users.length,
```

6. SUMMARY AND CONCLUSIONS

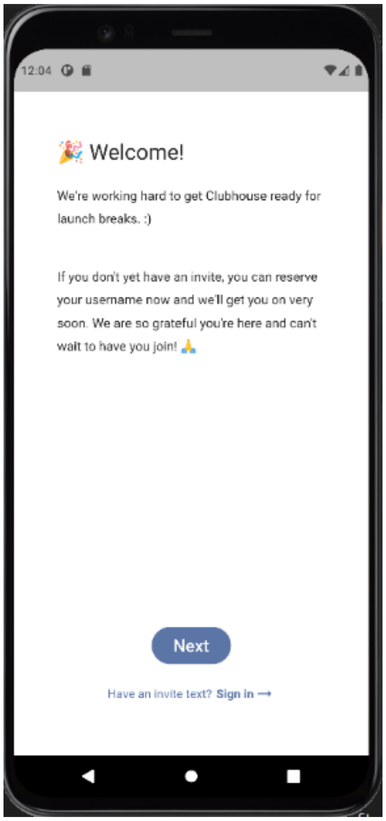
Summary:

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

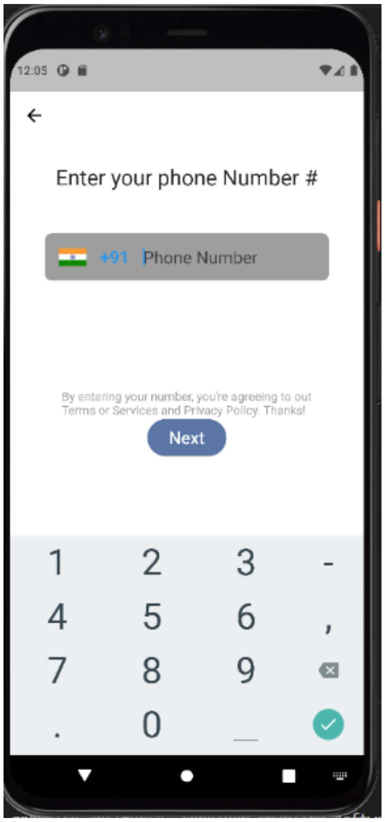


Welcome!

We're working hard to get Clubhouse ready for launch breaks. :)

If you don't yet have an invite, you can reserve your username now and we'll get you on very soon. We are so grateful you're here and can't wait to have you join! 🙏

Fig 1.1

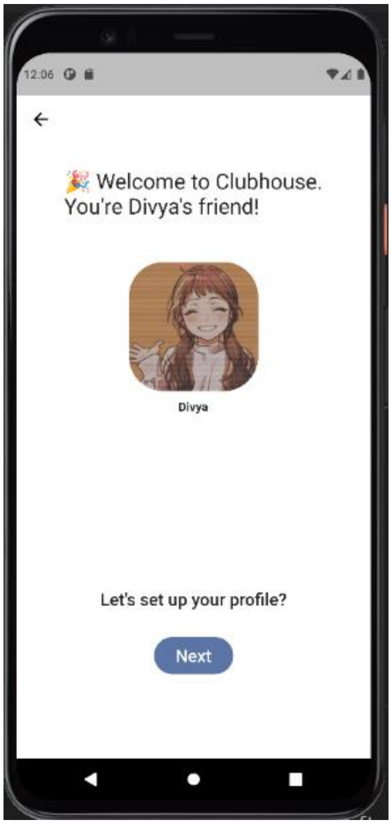


Enter your phone Number #

+91 Phone Number

By entering your number, you're agreeing to out Terms or Services and Privacy Policy. Thanks!

Fig 1.2

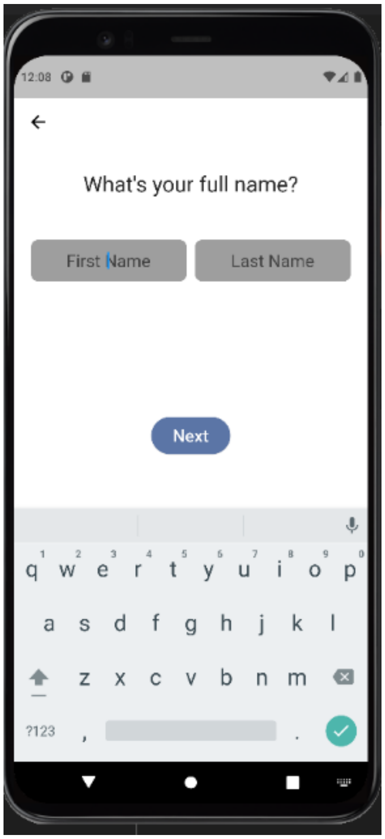


🎉 Welcome to Clubhouse.
You're Divya's friend!

Divya

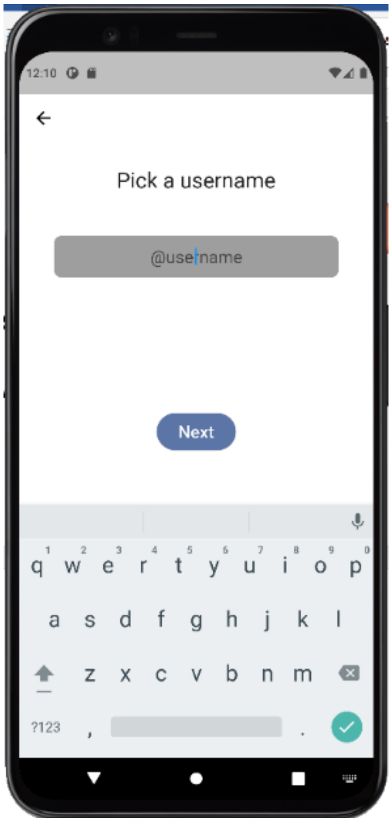
Let's set up your profile?

Fig 1.3



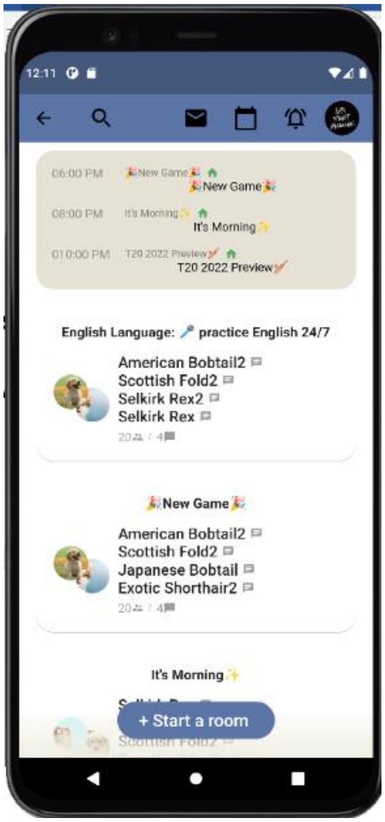
What's your full name?

Fig 1.4



Pick a username

Fig 1.5



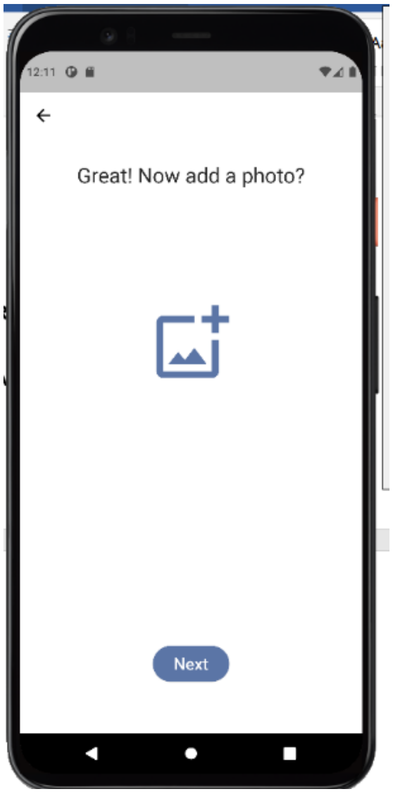
06:00 PM 🎉New Game🎉 🎉New Game🎉
08:00 PM It's Morning✨ It's Morning✨
010:00 PM T20 2022 Preview🏏 T20 2022 Preview🏏

English Language: 🎤 practice English 24/7
American Bobtail2
Scottish Fold2
Selkirk Rex2
Selkirk Rex
20 / 4

🎉New Game🎉
American Bobtail2
Scottish Fold2
Japanese Bobtail
Exotic Shorthair2
20 / 4

It's Morning✨
Scottish Fold2

Fig 1.6

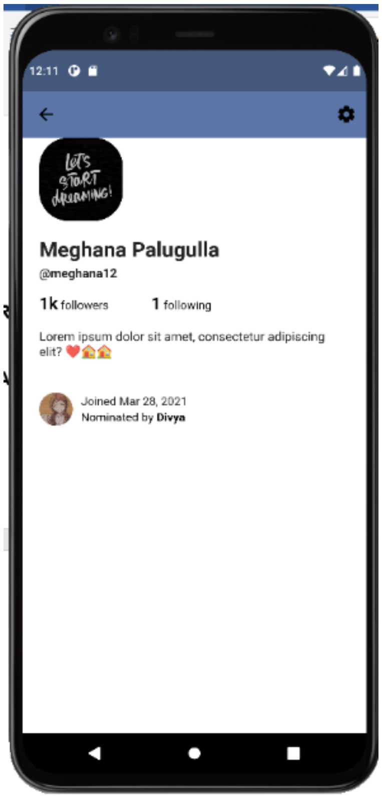


12:11

Great! Now add a photo?

Next

Fig 1.7



12:11

Let's START dreaming!

Meghana Palugulla

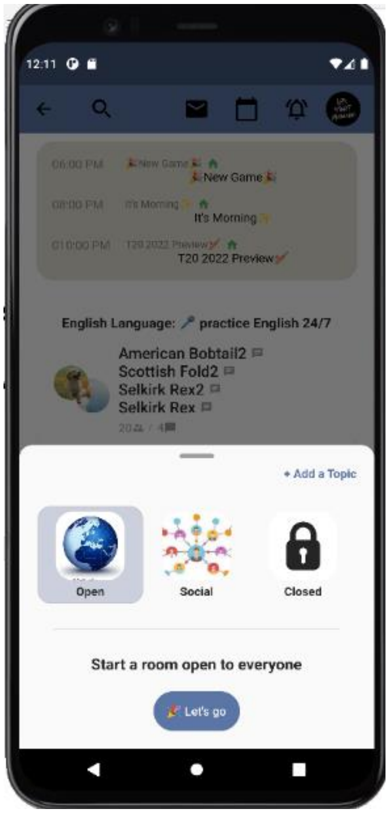
@meghana12

1k followers 1 following

Lorem ipsum dolor sit amet, consectetur adipiscing elit? ❤️🏠🏠

Joined Mar 28, 2021
Nominated by **Divya**

Fig 1.8



12:11

06:00 PM 🎉New Game🎉 🏠
🎉New Game🎉

08:00 PM It's Morning☀️ 🏠
It's Morning☀️

010:00 PM T20 2022 Preview✔️ 🏠
T20 2022 Preview✔️

English Language: 🎤 practice English 24/7

American Bobtail2
Scottish Fold2
Selkirk Rex2
Selkirk Rex

20 / 4

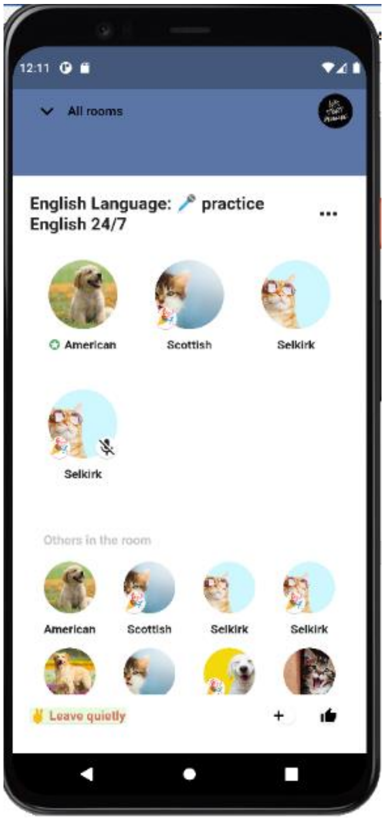
+ Add a Topic

Open Social Closed

Start a room open to everyone

🎉 Let's go

Fig 1.9



12:11

All rooms

English Language: 🎤 practice English 24/7

American Scottish Selkirk

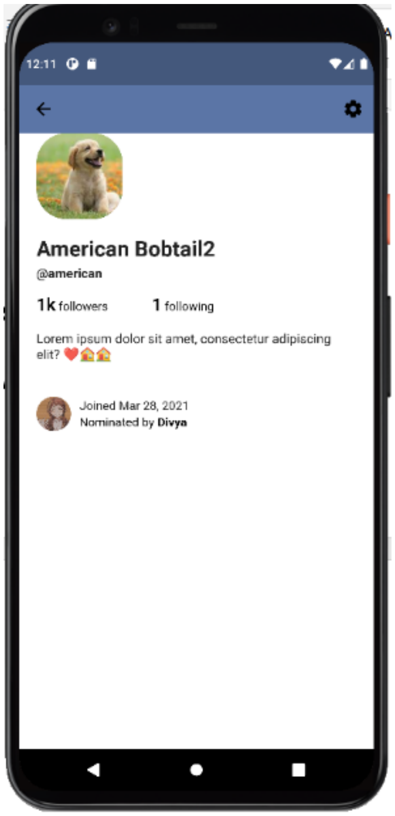
Selkirk

Others in the room

American Scottish Selkirk Selkirk

✌️ Leave quietly

Fig 2.0



12:11

American Bobtail2

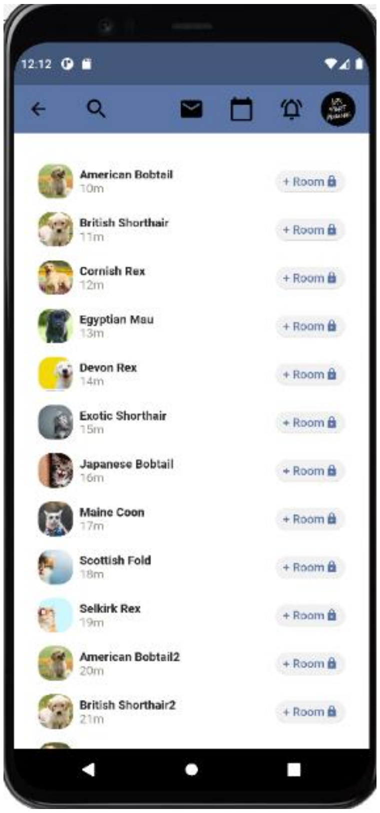
@american

1k followers 1 following

Lorem ipsum dolor sit amet, consectetur adipiscing elit? ❤️🏡🏡

Joined Mar 28, 2021
Nominated by **Divya**

Fig 2.1



12:12

American Bobtail 10m	+ Room 🔒
British Shorthair 11m	+ Room 🔒
Cornish Rex 12m	+ Room 🔒
Egyptian Mau 13m	+ Room 🔒
Devon Rex 14m	+ Room 🔒
Exotic Shorthair 15m	+ Room 🔒
Japanese Bobtail 16m	+ Room 🔒
Maine Coon 17m	+ Room 🔒
Scottish Fold 18m	+ Room 🔒
Selkirk Rex 19m	+ Room 🔒
American Bobtail2 20m	+ Room 🔒
British Shorthair2 21m	+ Room 🔒

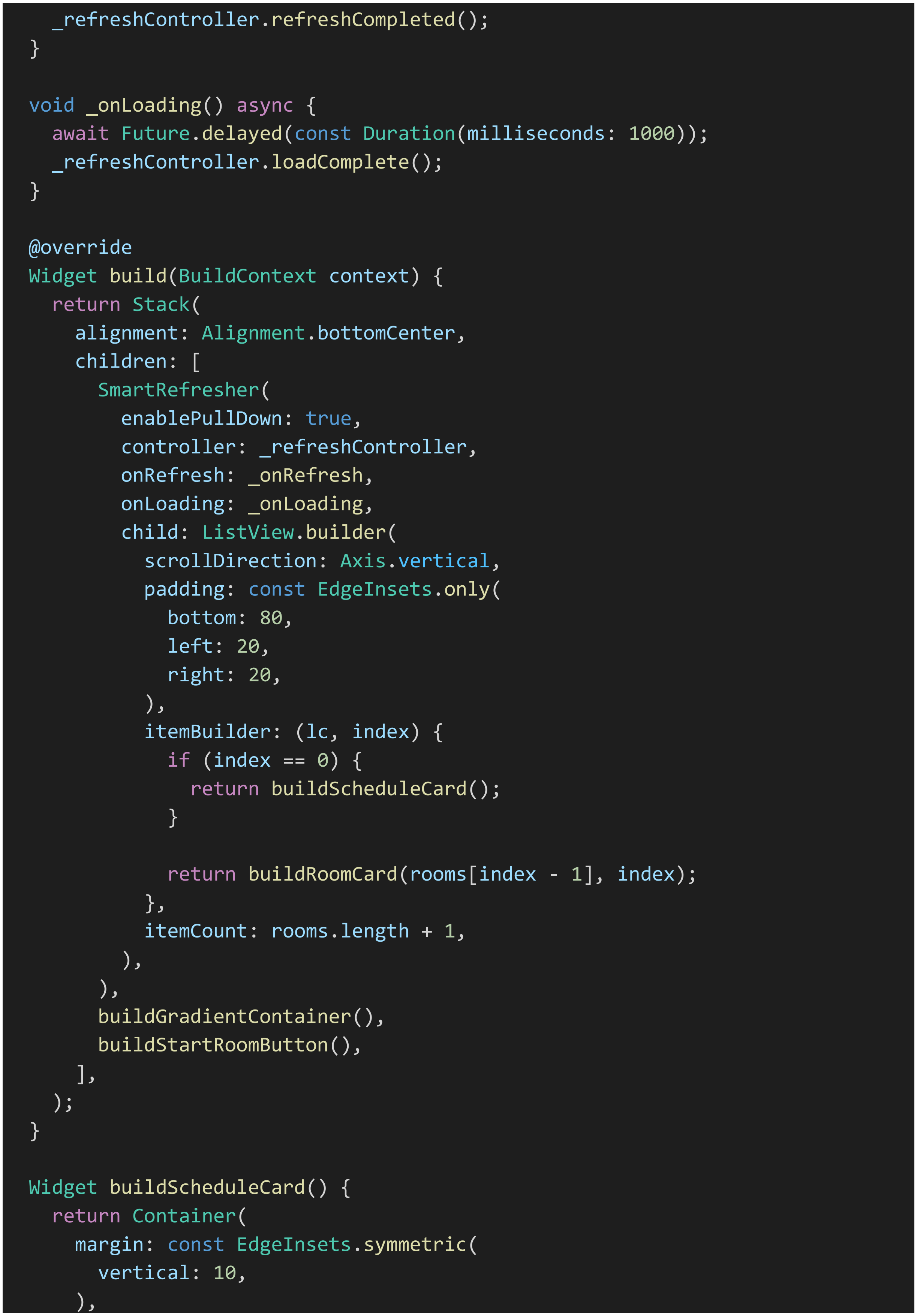
Fig 2.2

6.2 References:

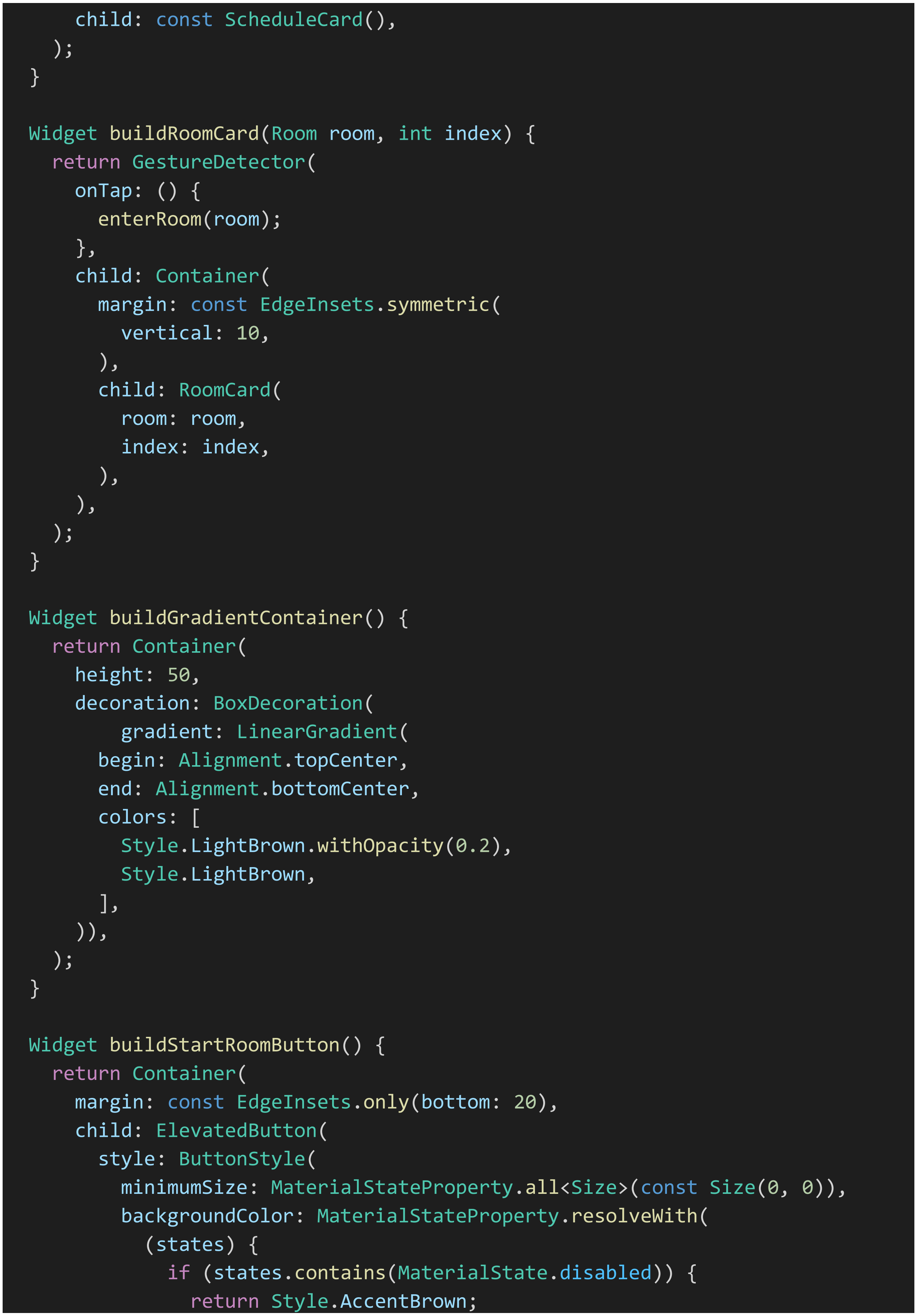
- docs.flutter.dev
- Beginning Flutter: A Hands On guide to App Development (Ebook)
- https://algofusion.org/course/flutter?tab=content
- https://pub.dev/
- https://dart.dev/
- https://flutter.dev/
- https://www.tutorialspoint.com/dart_programming/index.html

Source code:

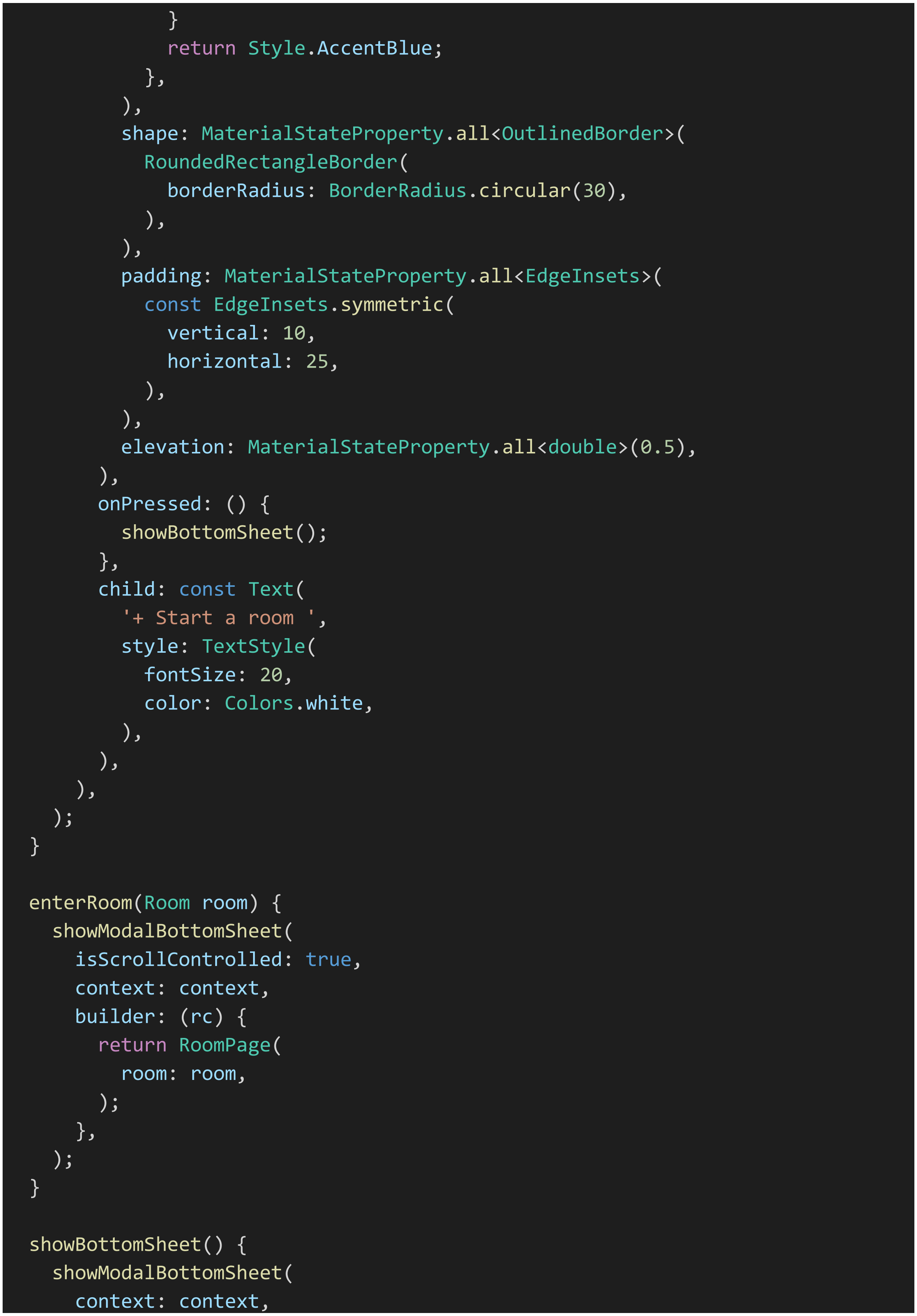
```
import '../../app_lib.dart';

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

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

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

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

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



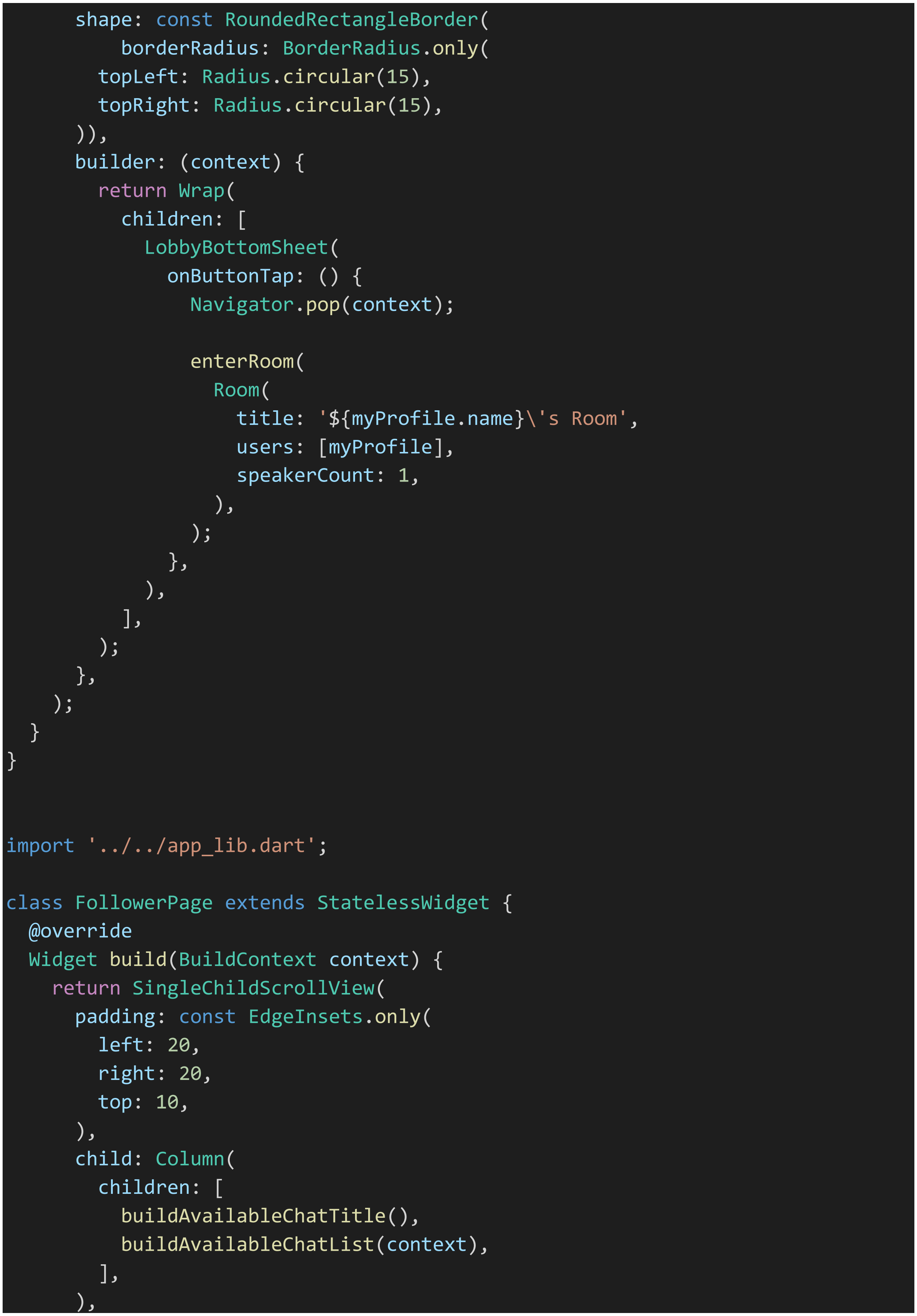
```
  _refreshController.refreshCompleted();
}

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

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

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

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



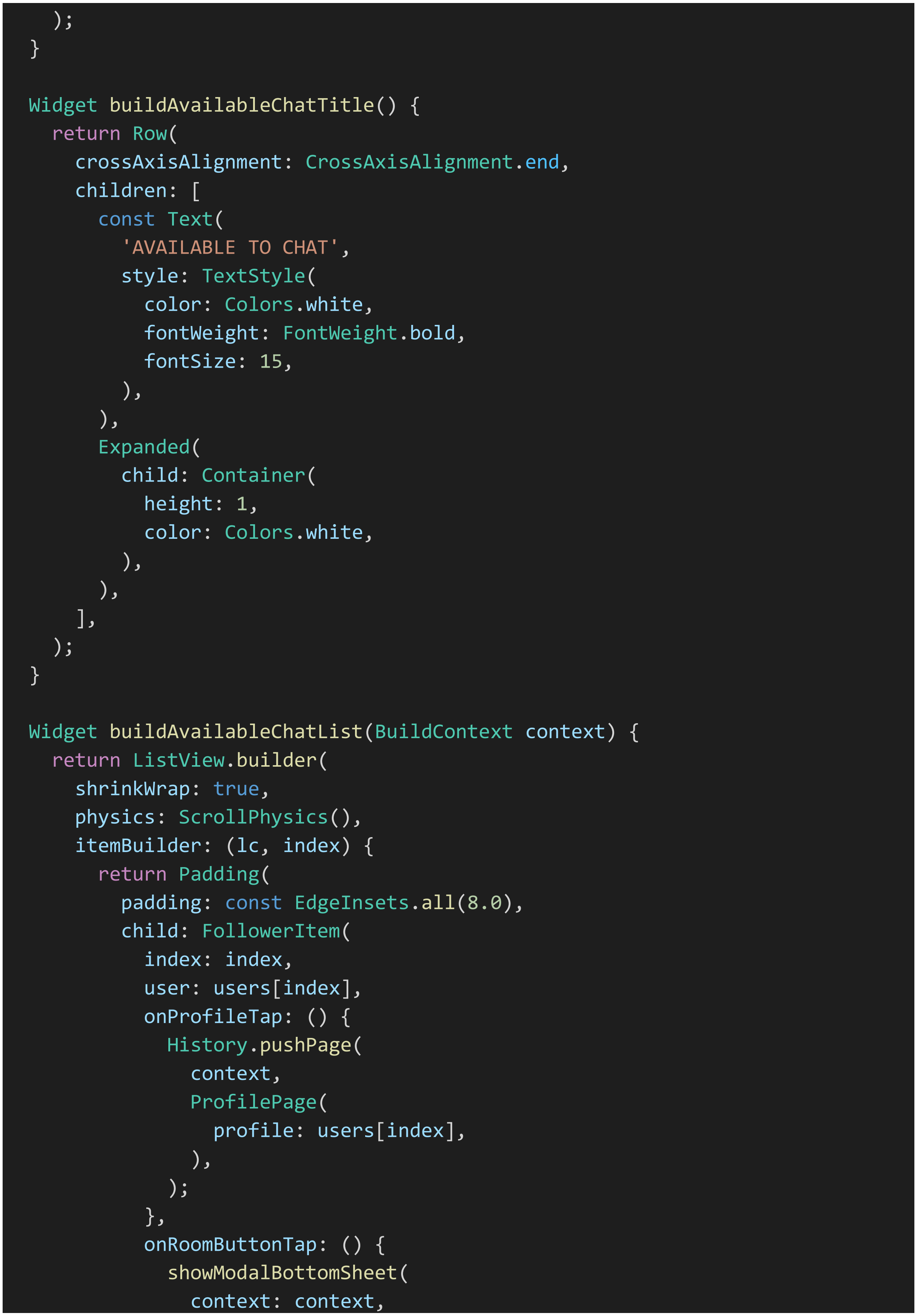
```
    child: const ScheduleCard(),
  );
}

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

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

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



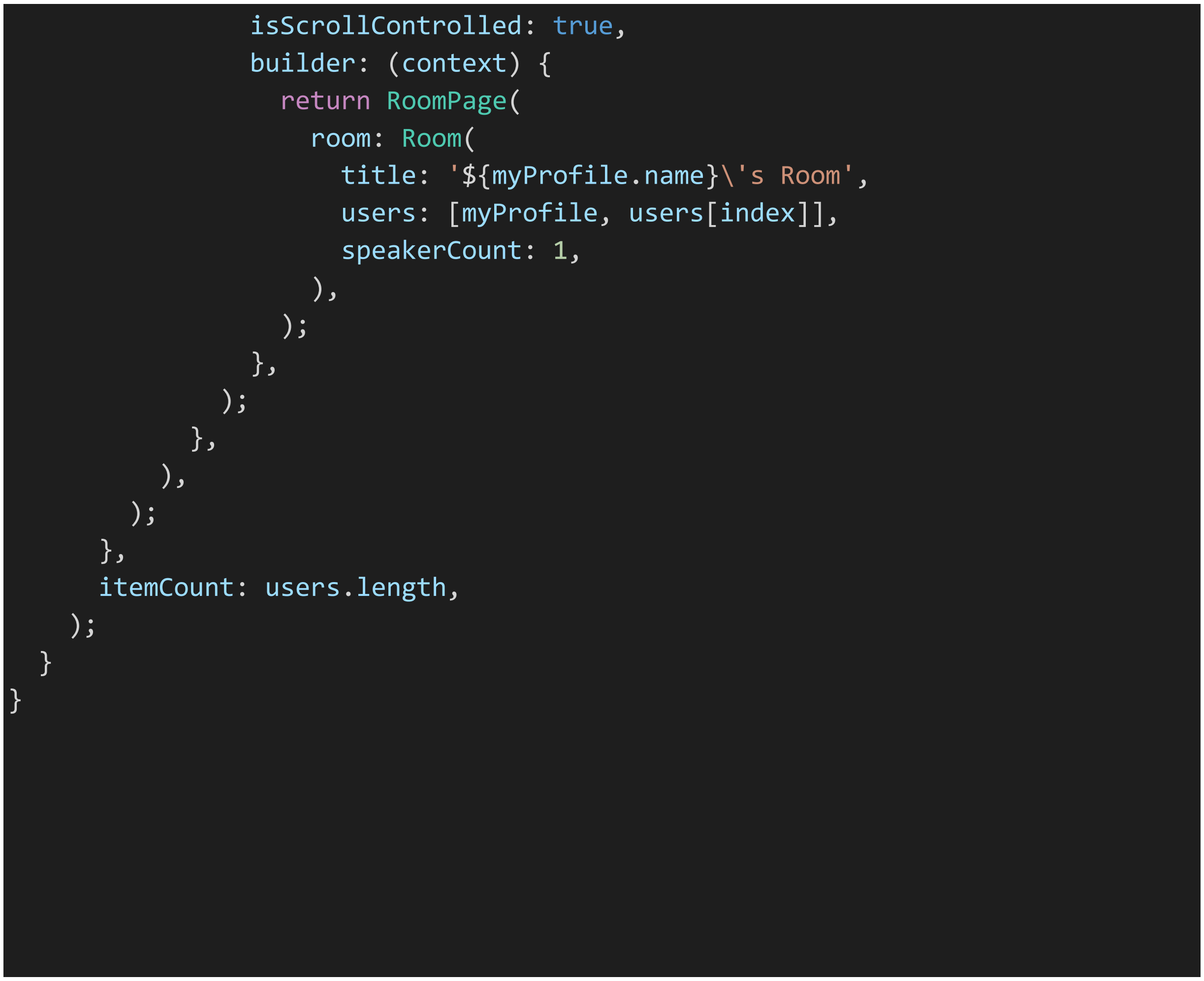
```
            }
            return Style.AccentBlue;
          },
        ),
        shape: MaterialStateProperty.all<OutlinedBorder>(
          RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(30),
          ),
        ),
        padding: MaterialStateProperty.all<EdgeInsets>(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

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

showBottomSheet() {
  showModalBottomSheet(
    context: context,
```



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

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


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

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



```
  );
}

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

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



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