

A Project Report on

**“KUVA”**

Submitted by [Group-14]

**1891004 – Akshit Jayant Chheda 1891007 – Ronik Mehul Dedhia 1891014 – Tanuj Rajendra Joshi**

Guided By

**Mrs. N. G. Kadukar**

Term – January’21 to May’21

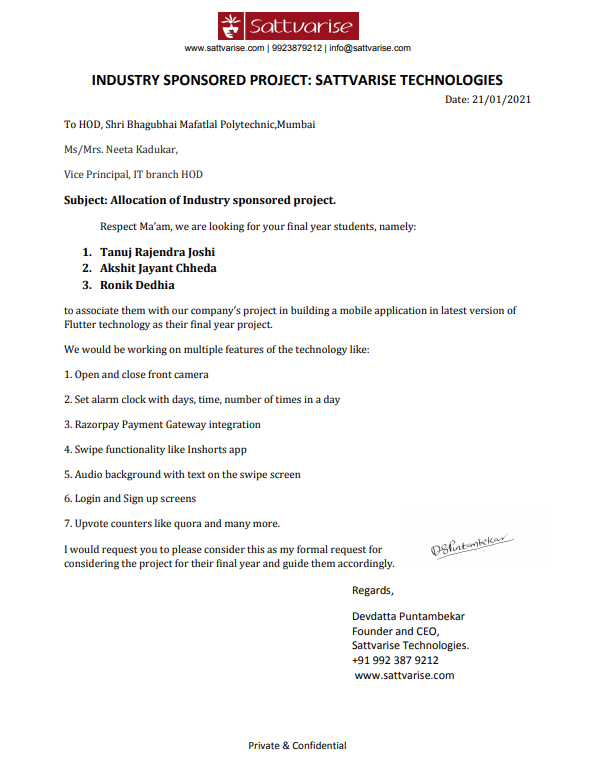
Department of Information Technology

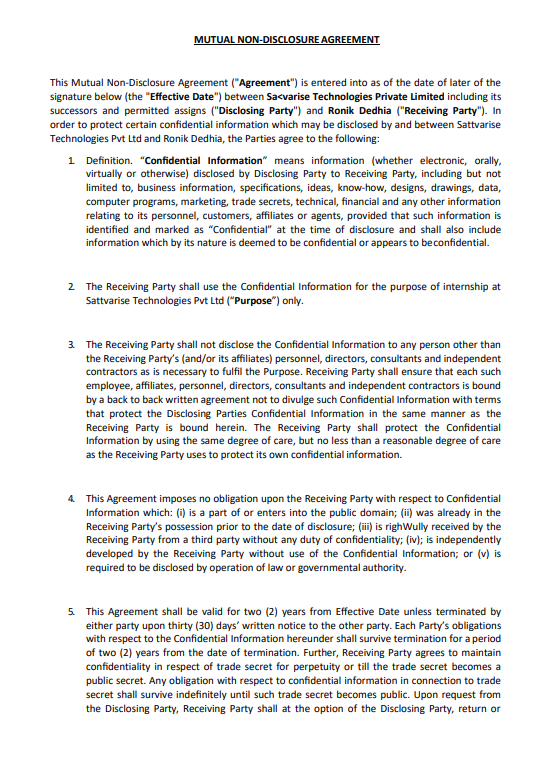
SVKM’s

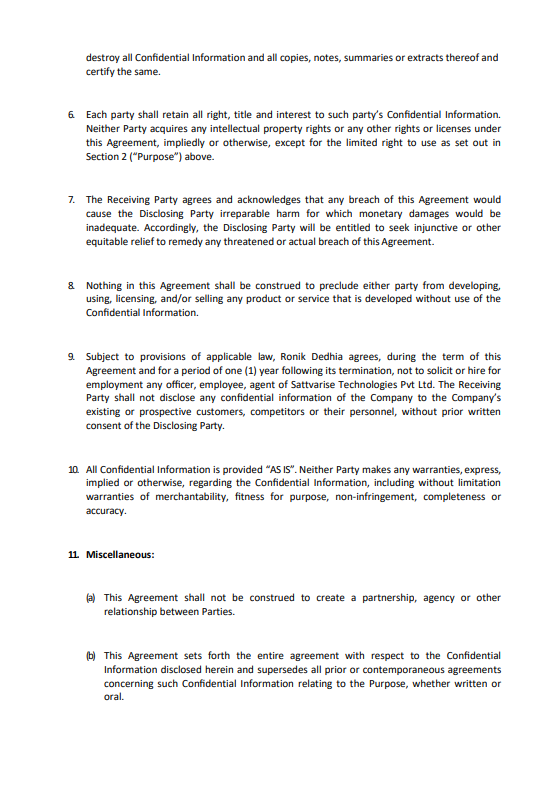
Shri Bhagubhai Mafatlal Polytechnic,

Irla, N.R.G. Marg, Vile Parle (W), Mumbai – 400056.

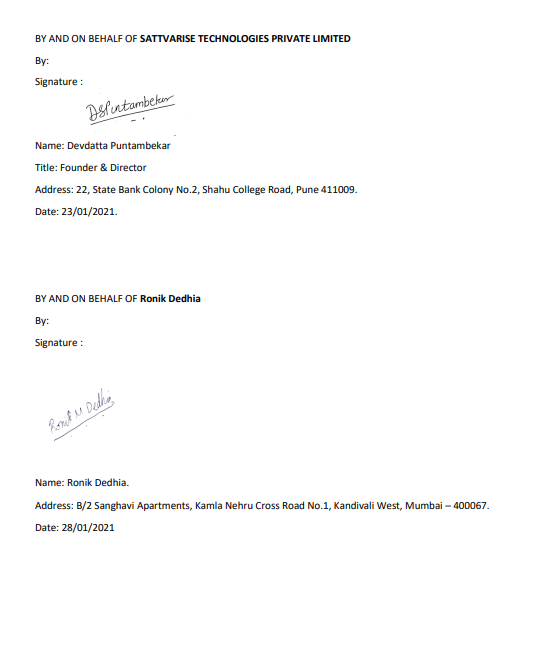


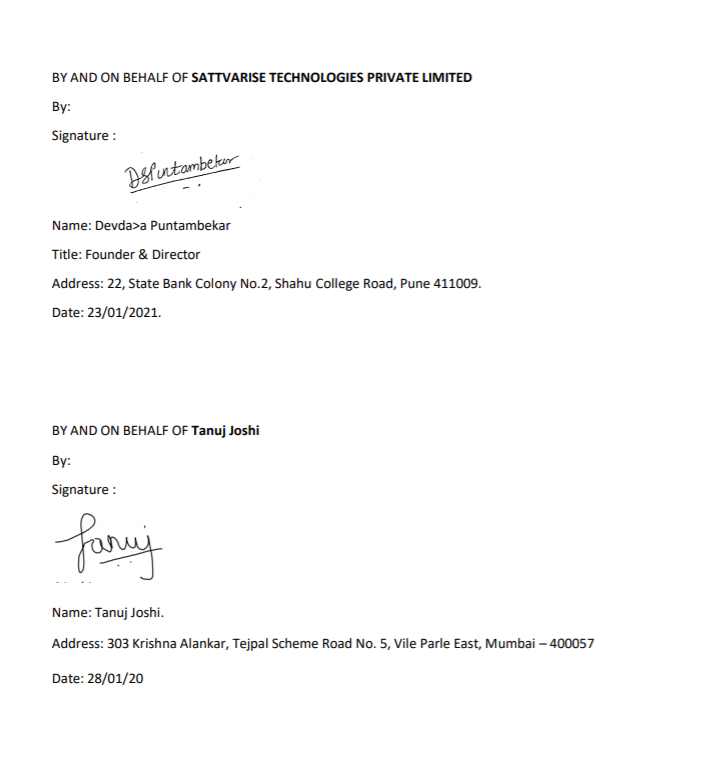


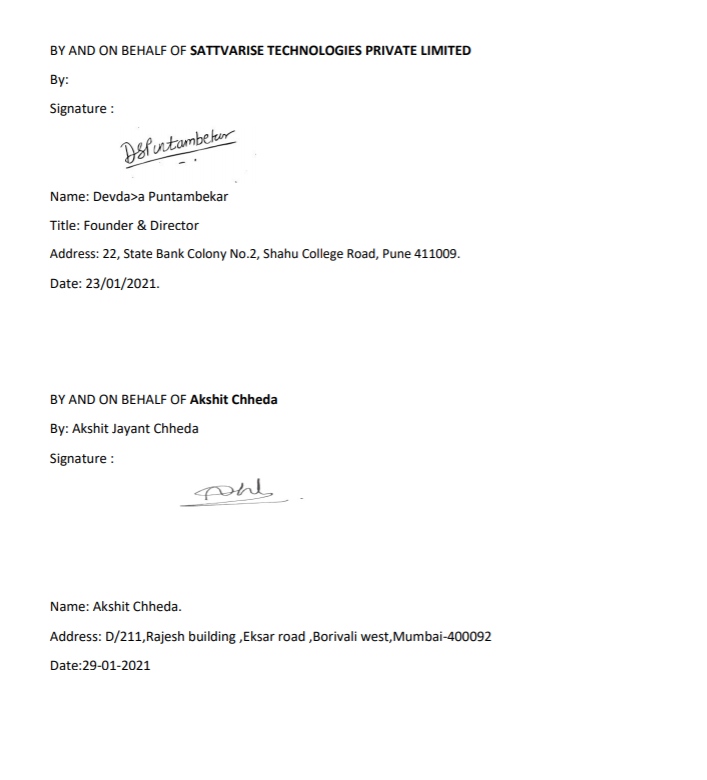












**Index**

1. Acknowledgement
2. Problem definition
3. Abstract
4. Introduction
   1. Concept
   2. Objectives

5 Software requirements 6 Platforms

7 Timeline chart

8 Process Flow

* 1. FlowChat
  2. UseCase

1. Literature Survey
   1. Flutter
   2. Dart
   3. Amazon Web Service
   4. Razorpay Paymenet Gateway
2. Kuva App
   1. Main.dart File
   2. Login.dart File
   3. Signup.dart File
   4. SplashScreen.dart File
   5. TabScreen.dart File
   6. CategoriesScreen.dart File
   7. CategoriesItem.dart File
   8. CategoryAffirmationScreen.dart File
   9. AffirmationItem.dart
   10. AffirmationDetailScreen.dart
   11. FirstScreen.dart
   12. Settings.dart
   13. HomePage.dart
   14. AlarmPage.dart
   15. ClockPage.dart
   16. ClockView.dart
   17. AlarmHelper.dart
   18. Data.dart
   19. Enums.dart
   20. Task.dart
   21. TaskRow.dart
   22. AlarmInfo.dart
   23. ThemeData.dart
   24. Affirmations.dart
   25. Categories.dart
   26. DummyCategories.dart
3. Limitations
4. Applications
5. Future scope
6. Conclusion
7. References

# Acknowledgement

First and foremost, we would like to thank our supervisor/guide of this project, **Mrs. Neeta Kadukar** for the valuable guidance and advice. She inspired us greatly to work in this project. Her willingness to motivate us contributed tremendously to our project.

Besides, we would like to thank the authority of **Sattvarise Technologies** for providing us with a good environment and facilities to complete this project. Also, we would like to take this opportunity to thank the Sattvarise Technologies for offering this subject - Affirmation App. It gave us an opportunity to participate and learn about the real- world app development with industry.

We are highly indebted to ***Mr. Devdatta Puntambekar*** *(Director and Co- Founder of Sattvarise Technologies)* for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to industry persons for giving us such attention and time. Finally, an honorable mention goes to our **families and friends** for their understanding and support for us in completing this project. Without the help of the that mentioned above, we would face many difficulties while doing this.

Thanking you,

*1891004 – Akshit Jayant Chheda, 1891007 – Ronik Mehul Dedhia, 1891014 – Tanuj Rajendra Joshi*

# Problem definition

The global pandemic caused by corona virus has not only affected people physically but also have increased the mental breakdown among the people. Mental Health Study conducted by Economic Times suggests that these individuals are the most affected – frontline workers who are constantly exposed to high-risk environments, working professionals, college students, people with existing chronic conditions, and many more who are confined to their home. Making a person feel better and reducing his mental stress is not an easy task for anyone so we decided to develop a solution.

An individual today faces concerns in many facets of his/her life like Family, Wealth, Health, Relationship and many more. The stress in people’s lives has increased manifold after losing their loved ones or losing their jobs. He needs to hear something positive daily to boost up his morale and carry on with his work effectively and happily. Keeping the above problems in mind, the company came up a with a unique and innovative solution to serve the society and help making people’s lives easier.

We plan to create an interface that is easily usable by a novice user. The user can create an account and access it whenever required allowing him to organize the affirmations and listen to it whenever required. With the goal of users’ convenience and to make the app well-structured, the app allows the users to choose from multiple categories at the same time and select the affirmations within them and parallelly listen to those affirmations.

# Abstract

The Affirmation Reminder App is focused on improving the mental health of the user and filling them with positivity and happiness. In this high-speed world you can easily lose sight of how large the small things really are. When healthy you may forget to think of how much you appreciate it. A simple morning affirmation sentence of “I am healthy” can go a long way. Daily affirmations not only help keep you surrounded by the things you want in your life, but they help bring about more blessings and gifts. Affirmations are powerful self-help tools to influence changes in your moods, feelings, thoughts, and habits. Daily affirmations influence your subconscious mind to access new beliefs.

Our aim is to build an app that is easy to use and motivates the user. It allows the user to set up the categories for the affirmation. The user can set his choices on the time he wants to hear the affirmations and the app notifies him on that time. The user can choose the number of times and set the alarms when he wants the app to notify them much like taking an appointment from a therapist. This will assist the user to get mental help even if he cannot attend a therapist physically. Just as a therapist helps his patients to come out of mental trauma, the app also strives to do the same and help the user with various functionalities.

Categories involved for Affirmations :-

Diagram

Description automatically generated

# Introduction

When a person is facing mental problems then this app helps them to change their negative thought patterns into positive ones. A recent study shows that optimistic people have healthier hearts, and affirmations help you to stay positive. ‘Kuva’ solves the following problems:

• For the users who are not comfortable with human interaction, this app will provide the interface to improve themselves on their own.

• Rising anxiety, stress, and defensiveness associated with threats to our sense of self.

• Best for people who prefer their own company and do not like to share their problems with anyone anytime.

• Clients not feeling like talking to someone, can turn on the Text-to-Speech mode, wherein the API will dictate the affirmations to the client, almost making it feel like they are talking to someone in person or talking to a therapist.

• Reduces the amount which the client pays to therapist.

• Saves the time of the client as it is always easy to open an app rather than talking with someone physically.

### Objectives:

• Helps you feel positive about yourself and boosts your self-confidence.

• Makes us more resilient to difficulties when they arise.

• Influence your subconscious mind to access new beliefs.

• You notice more synchronicities in your life, which serves to encourage and motivate you to act and keep up the practice.

• Keep you in a constant state of gratitude.

• Helps to keep the small things in perspective.

• Provide cross-platform app for fulfiling all above objectives for improving the health of human beings.

# Software Requirements

• Operating Systems: Windows 7 or later (64-bit), x86-64 based

• Disk Space: 1.64 GB (does not include disk space for IDE/tools).

• Tools: Flutter depends on these tools being available in your environment.

a) Windows PowerShell 5.0 or newer.

b) Git for Windows 2.x, with the Use Git from the Windows Command Prompt option.

• IDE: Android Studio

• SDK: Flutter SDK for Windows

• Virtual Device – Android Emulator (optional)

# Platforms

• Language – Dart

• Toolkit - Flutter

• Text Editor - VS Code

• Database – AWS, SQLite

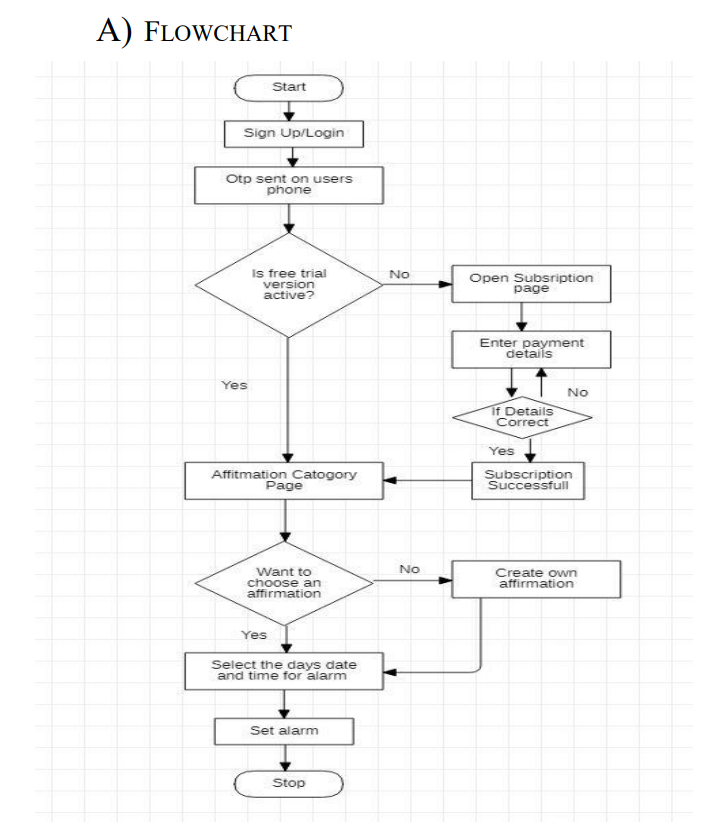
• Payment API – RazorPay

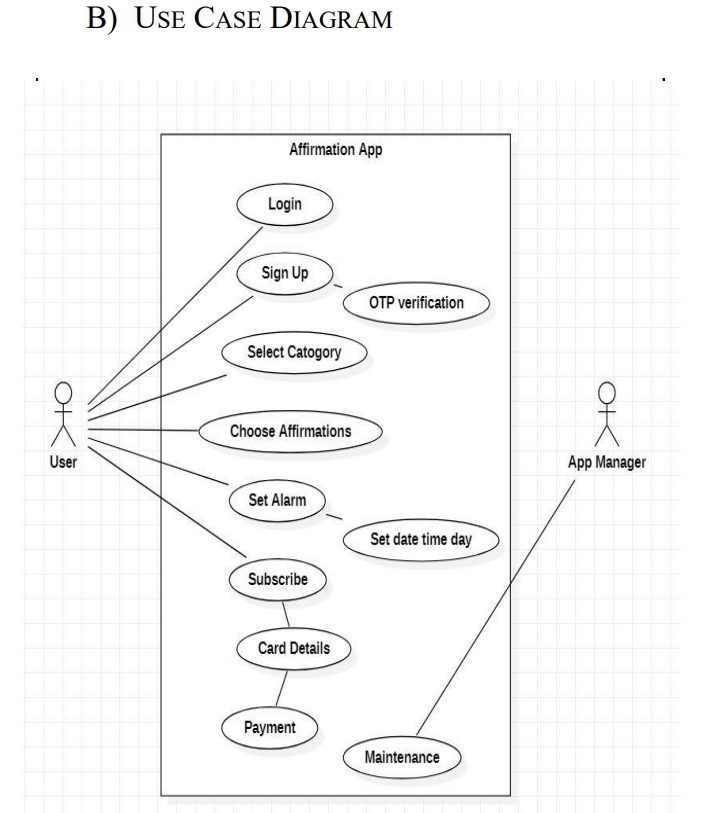
• Admin Panel - Laravel

# Timeline Chart

****

**8. Process Flow**

****

****

**9. Literature Survey**

### Flutter:



Flutter is Google’s open-source technology for creating native Android and iOS apps with a single codebase. Unlike other popular solutions, Flutter is not a framework; it is a complete SDK – software development kit – which already contains everything you will need to build cross-platform applications. This includes a rendering engine, ready-made widgets, testing and integration APIs, and command-line tools.

#### 

#### Why to choose Flutter?

• Same UI and business logic in all platforms.

• Similar to native app performance.

• Custom, animated UI of any complexity available.

• Hot reloading feature facilitates designer developer cooperation.

• High app responsiveness.

### Dart:



Dart is a programming language designed for client development such as for the web and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). It is developed by [Google](https://en.wikipedia.org/wiki/Google) and is used to build server and desktop applications. Dart is an open-source programming language which is widely used to develop the mobile application, modern web-applications, desktop application, and the Internet of Things (IoT) using by Flutter framework.

#### Why to choose Dart?

• It is an object-oriented programming language and supports all features of OOPS such as inheritance, interfaces, and optional type features.

• It is an open-source language, which means it available free for everyone.

• Dart is a platform-independent language and supports all operating systems such as Windows, Mac, Linux, etc.

• Dart is very useful in building real-time applications because of its stability.

Dart has rich tooling support. Almost every major Text Editor and IDE has great support for the Dart language.

Dart as a language, is very robust.

A screenshot of a computer

Description automatically generated with medium confidence

* 1. **Amazon Web Services:**



Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 175 AWS services are available.

#### Why to choose AWS?

• New services can be provisioned quickly, without the upfront capital expense.

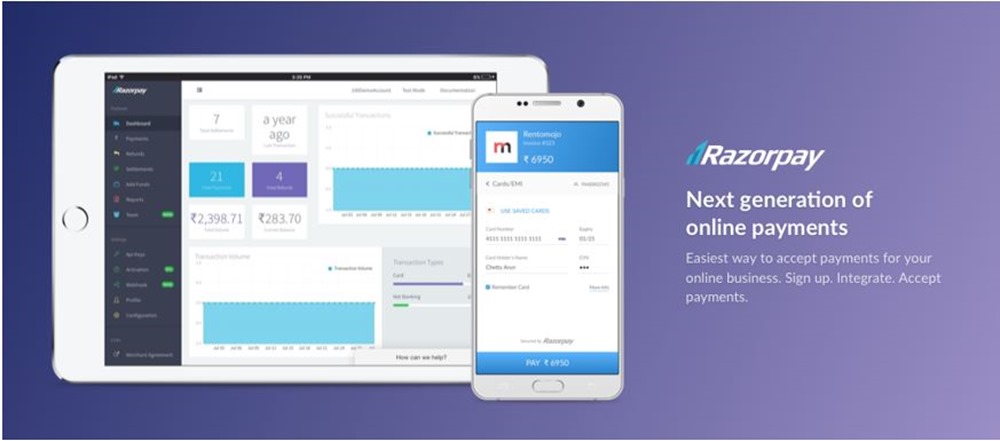
• Helps to respond quickly to changing business requirements.

• Designed to host your applications quickly and securely.

AWS has always been the most reliable and consistent service.

Provides the flexibility to scale up or down as required to increase or decrease server space in a matter of milliseconds.

**9.4 RazorPay Payment Gateway**



Razorpay is the only converged payments solution company in India that allows your business to accept, process, and disburse payments via its product suite. With Razorpay, you have access to all payment modes, including credit and debit cards, UPI, and popular mobile wallets. The most crucial quality of a payment solution is the ability to scale itself to meet your growing demands. The company offers a fast, secure, hassle-free, and affordable way for merchants, ecommerce retailers, schools and other businesses to accept and disburse payments digitally.

#### Why to choose Razorpay?

• Supports wide range of payment solutions and services to support your versatile business.

• Target your efforts towards specific areas of your business based on data and insights.

• Manage and track money movements to vendors, customers, employees.

• Opens new revenue streams by expanding your business on a global scale.

**10. Kuva App**

**10.1 Main.dart File**

import 'package:flutter/material.dart';

import 'package:kuva/SpashScreen.dart';

import './dummy\_data.dart';

import './screen/tabs\_screen.dart';

import './screen/affirmation\_detail\_screen.dart';

import './screen/category\_affirmation\_screen.dart';

import './screen/login.dart';

import 'screen/signup.dart';

import './screen/categories\_screen.dart';

import './models/affirmations.dart';

import 'package:flutter\_local\_notifications/flutter\_local\_notifications.dart';

final **FlutterLocalNotificationsPlugin** flutterLocalNotificationsPlugin =

**FlutterLocalNotificationsPlugin**();

void **main**() async {

**binding**();

**runApp**(**MyApp**());

}

class **MyApp** extends **StatefulWidget** {

  @override

**\_MyAppState** **createState**() => **\_MyAppState**();

}

**binding**() async

{

**WidgetsFlutterBinding**.**ensureInitialized**();

  var initializationSettingsAndroid =

**AndroidInitializationSettings**('codex\_logo');

  var initializationSettingsIOS = **IOSInitializationSettings**(

      requestAlertPermission: true,

      requestBadgePermission: true,

      requestSoundPermission: true,

      onDidReceiveLocalNotification:

          (**int** id, **String** title, **String** body, **String** payload) async {});

  var initializationSettings = **InitializationSettings**(

      initializationSettingsAndroid, initializationSettingsIOS);

  await flutterLocalNotificationsPlugin.**initialize**(initializationSettings,

      onSelectNotification: (**String** payload) async {

    if (payload != null) {

**debugPrint**('notification payload: ' + payload);

    }

  });

}

class **\_MyAppState** extends **State**<**MyApp**> {

@override

**List**<**Affirmations**> \_availableAffirmations = **DUMMY\_Affirmations**;

**List**<**Affirmations**> \_favoriteAffirmations = [];

  void **\_toggleFavorite**(**String** affId) {

    final existingIndex =

        \_favoriteAffirmations.**indexWhere**((aff) => aff.id == affId);

    if (existingIndex >= 0) {

**setState**(() {

        \_favoriteAffirmations.**removeAt**(existingIndex);

      });

    } else {

**setState**(() {

        \_favoriteAffirmations.**add**(

**DUMMY\_Affirmations**.**firstWhere**((meal) => meal.id == affId),

        );

      });

    }

  }

**bool** **\_isAffFavorite**(**String** id) {

    return \_favoriteAffirmations.**any**((meal) => meal.id == id);

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **MaterialApp**(

      title: 'Kuva',

      darkTheme: **ThemeData**.**dark**(),

      theme: **ThemeData**(

        primarySwatch: **Colors**.pink,

        accentColor: **Colors**.amber,

        canvasColor: **Color**.**fromRGBO**(255, 254, 229, 1),

        fontFamily: 'Raleway',

        textTheme: **ThemeData**.**dark**().textTheme.**copyWith**(

            body1: **TextStyle**(

              color: **Color**.**fromRGBO**(20, 51, 51, 1),

            ),

            body2: **TextStyle**(

              color: **Color**.**fromRGBO**(20, 51, 51, 1),

            ),

            title: **TextStyle**(

              fontSize: 20,

              fontFamily: 'RobotoCondensed',

              fontWeight: **FontWeight**.bold,

            )),

      ),

      initialRoute: '/splash',

      routes: {

        '/splash':(context)=>**SScreen**(),

         '/login': (context) => **LoginPage**(),

        '/signup' :(context) => **SignupPage**(),

        '/': (ctx) => **TabsScreen**(),

**CategoryAffirmationScreen**.routeName: (ctx) =>

**CategoryAffirmationScreen**(\_availableAffirmations),

**AffirmationDetailScreen**.routeName: (ctx) => **AffirmationDetailScreen**(\_toggleFavorite, \_isAffFavorite),

      },

      onGenerateRoute: (settings) {

**print**(settings.arguments);

      },

      onUnknownRoute: (settings) {

        return **MaterialPageRoute**(

          builder: (ctx) => **CategoriesScreen**(),

        );

      },

    );

  }

}

**10.2 Login.dart File**

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

class **LoginPage** extends **StatefulWidget** {

**\_submitAuthForm**(

**String** phn,

**String** email,

**String** password,

**String** username,

**BuildContext** ctx,

  ) async {

    try {

        if(username=="akshitchheda23@gmail.com" && password=="akshitchheda")

        {  **print**('akshit2');

**Navigator**.**pushNamed**(ctx, '/');

        }

        else {

**print**(username);

**print**(password);

**Navigator**.**pushNamed**(ctx, '/signup');

      }

    } on **PlatformException** catch (err) {

      var message = 'An error occurred, pelase check your credentials!';

      if (err.message != null) {

        message = err.message;

      }

**Scaffold**.**of**(ctx).**showSnackBar**(

**SnackBar**(

          content: **Text**(message),

          backgroundColor: **Theme**.**of**(ctx).errorColor,

        ),

      );

    } catch (err) {

**print**(err);

    }

  }

  @override

**State**<**StatefulWidget**> **createState**() {

    return **\_LoginPageState**();

  }

}

class **\_LoginPageState** extends **State**<**LoginPage**> {

   final \_formKey = **GlobalKey**<**FormState**>();

  var \_userEmail = '';

  var \_phn='';

  var \_userName = '';

  var \_userPassword = '';

  void **\_trySubmit**() {

    final isValid = \_formKey.currentState.**validate**();

**FocusScope**.**of**(context).**unfocus**();

    if (isValid) {

      \_formKey.currentState.**save**();

      widget.**\_submitAuthForm**(\_userEmail.**trim**(),\_phn.**trim**(), \_userPassword.**trim**(), \_userName.**trim**(),

           context);

    }

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **Scaffold**(

      body: **Container**(

        child: **ListView**(

          children: <**Widget**>[

**Container**(

              width: **MediaQuery**.**of**(context).size.width,

              height: **MediaQuery**.**of**(context).size.height/3.5,

              decoration: **BoxDecoration**(

                gradient: **LinearGradient**(

                  begin: **Alignment**.topCenter,

                  end: **Alignment**.bottomCenter,

                  colors: [

**Color**(0xff6bceff),

**Color**(0xff6bceff)

                  ],

                ),

                borderRadius: **BorderRadius**.**only**(

                  bottomLeft: **Radius**.**circular**(90)

                )

              ),

              child: **Column**(

                mainAxisAlignment: **MainAxisAlignment**.center,

                children: <**Widget**>[

**Spacer**(),

**Align**(

                    alignment: **Alignment**.center,

                    child: **Icon**(**Icons**.person,

                      size: 90,

                      color: **Colors**.white,

                    ),

                  ),

**Spacer**(),

**Align**(

                    alignment: **Alignment**.bottomRight,

                      child: **Padding**(

                        padding: const **EdgeInsets**.**only**(

                          bottom: 32,

                          right: 32

                        ),

                        child: **Text**('Login',

                          style: **TextStyle**(

                            color: **Colors**.white,

                            fontSize: 18

                          ),

                        ),

                      ),

                  ),

                ],

              ),

            ),

**Container**(

              height: **MediaQuery**.**of**(context).size.height/2,

              width: **MediaQuery**.**of**(context).size.width,

              padding: **EdgeInsets**.**only**(top: 62),

              child:**Form**(key: \_formKey,child: **Column**(

                children: <**Widget**>[

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 55,

                    padding: **EdgeInsets**.**only**(

                      top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                      borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                      ),

                      color: **Colors**.white,

                      boxShadow: [

**BoxShadow**(

                          color: **Colors**.black12,

                          blurRadius: 5

                        )

                      ]

                    ),

                    child: **TextFormField**(

                      style: **TextStyle**(

                  color: **Colors**.black

                ),

                       key:**ValueKey**('username'),

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                        icon: **Icon**(**Icons**.person,

                            color: **Color**(0xff6bceff),

                        ),

                          hintText: 'Username',

                          hintStyle: **TextStyle**(color:**Colors**.black),

                      ),

                       validator: (value) {

                        if (value.isEmpty || value.length < 7) {

                          return 'Password must be at least 7 characters long.';

                        }

                        return null;

                      },

                      onSaved: (value) {

                          \_userName = value;

**print**(\_userName);

                        },

                    ),

                  ),

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 55,

                    margin: **EdgeInsets**.**only**(top: 32),

                    padding: **EdgeInsets**.**only**(

                        top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                        borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                        ),

                        color: **Colors**.white,

                        boxShadow: [

**BoxShadow**(

                              color: **Colors**.black12,

                              blurRadius: 5

                          )

                        ]

                    ),

                    child: **TextFormField**(  style: **TextStyle**(

                  color: **Colors**.black

                ),key: **ValueKey**('password'), validator: (value) {

                        if (value.isEmpty || value.length < 7) {

                          return 'Password must be at least 7 characters long.';

                        }

                        return null;

                      },

                      obscureText: true,

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                        icon: **Icon**(**Icons**.vpn\_key,

                          color: **Color**(0xff6bceff),

                        ),

                        hintText: 'Password',

                        hintStyle: **TextStyle**(color:**Colors**.black),

                      ),

                      onSaved: (value) {

                        \_userPassword = value;

                      },

                    ),

                  ),

**Align**(

                    alignment: **Alignment**.centerRight,

                    child: **Padding**(

                      padding: const **EdgeInsets**.**only**(

                        top: 16, right: 32

                      ),

                      child: **Text**('Forgot Password ?',

                        style: **TextStyle**(

                          color: **Colors**.grey

                        ),

                      ),

                    ),

                  ),

**Spacer**(),

**InkWell**(

                      onTap:

                       \_trySubmit

                      ,

                      child: **Container**(

                      height: 45,

                      width: **MediaQuery**.**of**(context).size.width/1.2,

                      decoration: **BoxDecoration**(

                        gradient: **LinearGradient**(

                          colors: [

**Color**(0xff6bceff),

**Color**(0xFF00abff),

                          ],

                        ),

                        borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                        )

                      ),

                      child: **Center**(

                        child: **Text**('Login'.**toUpperCase**(),

                          style: **TextStyle**(

                            color: **Colors**.white,

                            fontWeight: **FontWeight**.bold

                          ),

                        ),

                      ),

                    ),

                  ),

                ],

              ),)

            ),

**SizedBox**(

              height: 50,

            ),

**InkWell**(

              child: **Row**(

                mainAxisAlignment: **MainAxisAlignment**.center,

                children: <**Widget**>[

**Text**("Don't have an account ?"),

**Text**("Sign Up",style: **TextStyle**(color: **Color**(0xff6bceff)),),

                ],

              ),

              onTap: (){

**Navigator**.**pushNamed**(context, '/signup');

              },

            ),

          ],

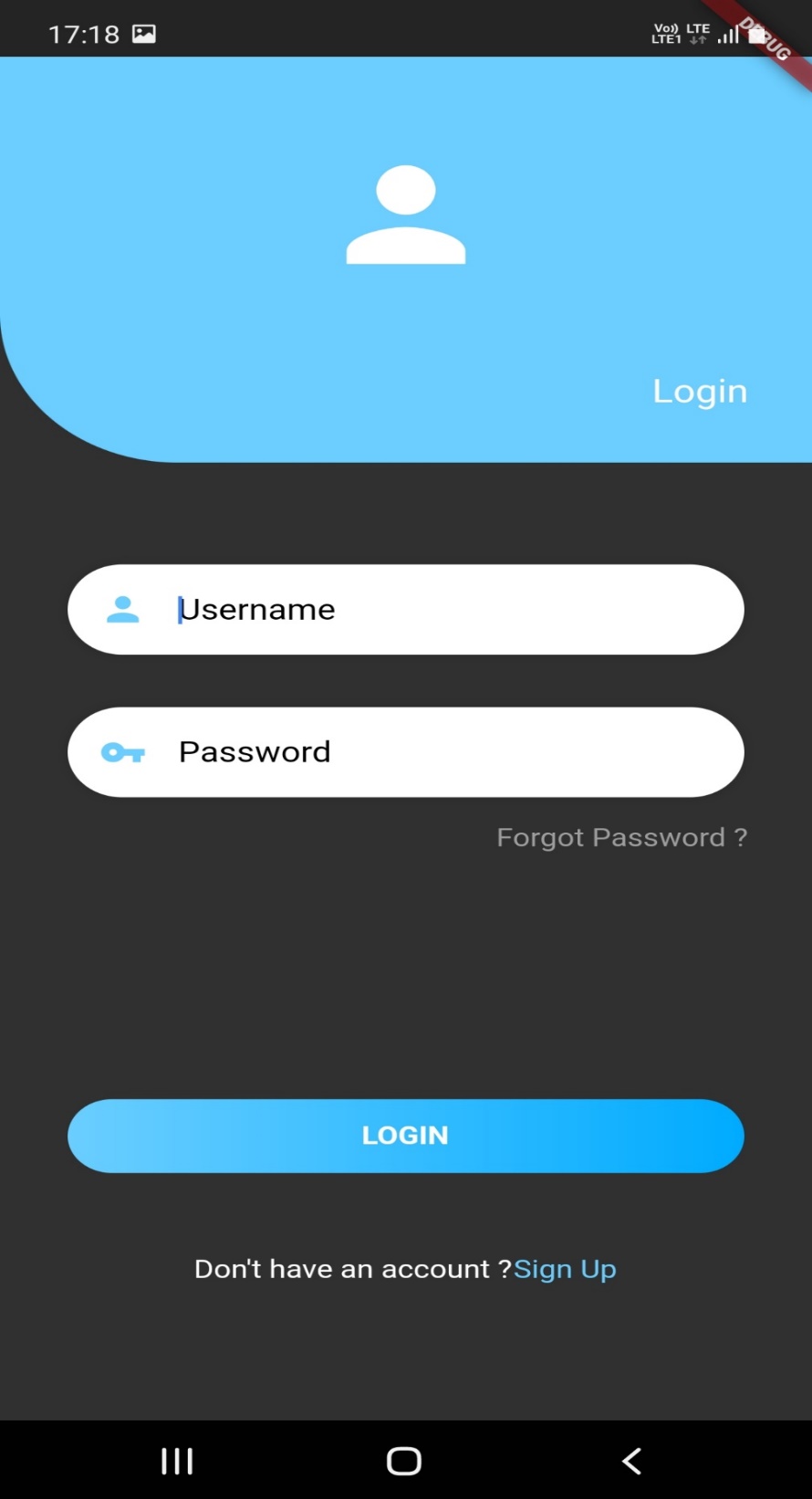
        ),

      ),

    );

  }

}



**10.3 Signup.dart File**

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

class **SignupPage** extends **StatefulWidget** {

**\_submitAuthForm2**(

**String** phn,

**String** email,

**String** password,

**String** username,

**String** fname,

**BuildContext** ctx,

  ) async {

    try {

**Navigator**.**pushNamed**(ctx, '/');

    } on **PlatformException** catch (err) {

      var message = 'An error occurred, pelase check your credentials!';

      if (err.message != null) {

        message = err.message;

      }

**Scaffold**.**of**(ctx).**showSnackBar**(

**SnackBar**(

          content: **Text**(message),

          backgroundColor: **Theme**.**of**(ctx).errorColor,

        ),

      );

    } catch (err) {

**print**(err);

    }

  }

  @override

**\_SignupPageState** **createState**() => **\_SignupPageState**();

}

class **\_SignupPageState** extends **State**<**SignupPage**> {

   final \_formKey = **GlobalKey**<**FormState**>();

  var \_userEmail = '';

  var \_phn='';

  var \_userName = '';

  var \_userPassword = '';

  var \_userFname='';

**bool** \_checked = false;

**bool** showErrorMessage = false;

  void **\_trySubmit**() {

    final isValid = \_formKey.currentState.**validate**();

**FocusScope**.**of**(context).**unfocus**();

    if (isValid) {

      \_formKey.currentState.**save**();

      widget.**\_submitAuthForm2**(\_userEmail.**trim**(),\_phn.**trim**(), \_userPassword.**trim**(), \_userName.**trim**(),\_userFname.**trim**(),

           context);

    }

  void **displayMessage**(){

**showDialog**(context: context,

  builder: (**BuildContext** context){

**AlertDialog** dialog = **AlertDialog**(

      content: **Text**('You have accepted the terms'),

      actions: [

**FlatButton**(

          child: **Text**('OK'),

          onPressed: (){

**Navigator**.**of**(context).**pop**();

          },

        )

      ],

    );

    return dialog;

  }

  );

}

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **Scaffold**(

      body: **Container**(

        child: **ListView**(

          children: <**Widget**>[

**Container**(

              width: **MediaQuery**.**of**(context).size.width,

              height: **MediaQuery**.**of**(context).size.height/3.5,

              decoration: **BoxDecoration**(

                gradient: **LinearGradient**(

                  begin: **Alignment**.topCenter,

                  end: **Alignment**.bottomCenter,

                  colors: [

**Color**(0xff6bceff),

**Color**(0xff6bceff)

                  ],

                ),

                borderRadius: **BorderRadius**.**only**(

                  bottomLeft: **Radius**.**circular**(90)

                )

              ),

              child: **Column**(

                mainAxisAlignment: **MainAxisAlignment**.center,

                children: <**Widget**>[

**Spacer**(),

**Align**(

                    alignment: **Alignment**.center,

                    child: **Icon**(**Icons**.person,

                      size: 90,

                      color: **Colors**.white,

                    ),

                  ),

**Spacer**(),

**Align**(

                    alignment: **Alignment**.bottomRight,

                      child: **Padding**(

                        padding: const **EdgeInsets**.**only**(

                          bottom: 32,

                          right: 32

                        ),

                        child: **Text**('Sign Up',

                          style: **TextStyle**(

                            color: **Colors**.white,

                            fontSize: 18

                          ),

                        ),

                      ),

                  ),

                ],

              ),

            ),

**Container**(

              height: **MediaQuery**.**of**(context).size.height/2,

              width: **MediaQuery**.**of**(context).size.width,

              padding: **EdgeInsets**.**only**(top: 62),

              child: **Form**(key:\_formKey,child:**Column**(

                children: <**Widget**>[

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 55,

                    padding: **EdgeInsets**.**only**(

                      top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                      borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                      ),

                      color: **Colors**.white,

                      boxShadow: [

**BoxShadow**(

                          color: **Colors**.black12,

                          blurRadius: 5

                        )

                      ]

                    ),

                    child: **TextFormField**( style: **TextStyle**(

                  color: **Colors**.black

                ),

                      key: **ValueKey**('Full name'),validator: (value) {

                        if (value.isEmpty || value.length < 7) {

                          return 'Full Name must be at least 7 characters long.';

                        }

                        return null;

                      },

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                          hintText: 'Full Name',

                          hintStyle: **TextStyle**(color:**Colors**.black),

                      ), onSaved: (value) {

                        \_userFname = value;

                      }

                    ),

                  ),

**SizedBox**(

                    height: 5,

                  ),

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 50,

                    padding: **EdgeInsets**.**only**(

                      top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                      borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                      ),

                      color: **Colors**.white,

                      boxShadow: [

**BoxShadow**(

                          color: **Colors**.black12,

                          blurRadius: 5

                        )

                      ]

                    ),

                    child: **TextFormField**( style: **TextStyle**(

                  color: **Colors**.black

                ),

                       key: **ValueKey**('Email') ,validator: (value) {

                        if (value.isEmpty || !value.**contains**('@')) {

                          return 'Please enter a valid email address.';

                        }

                        return null;

                      },

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                          hintText: 'Email',

                          hintStyle: **TextStyle**(color:**Colors**.black),

                      ),onSaved: (value) {

                        \_userEmail = value;

                      },

                    ),

                  ),

**SizedBox**(

                    height: 5,

                  ),

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 50,

                    padding: **EdgeInsets**.**only**(

                      top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                      borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                      ),

                      color: **Colors**.white,

                      boxShadow: [

**BoxShadow**(

                          color: **Colors**.black12,

                          blurRadius: 5

                        )

                      ]

                    ),

                    child: **TextFormField**(

                      keyboardType: **TextInputType**.number,

                      inputFormatters: [

                        new **LengthLimitingTextInputFormatter**(10),

                      ],

                      style: **TextStyle**(

                      color: **Colors**.black,

                ),

                       key: **ValueKey**('Username'),validator: (value) {

                        if (value.isEmpty || value.length < 7) {

                          return 'Phone no must be at least 10 characters only.';

                        }

                        return null;

                      },

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                          hintText: 'Phone no',

                          hintStyle: **TextStyle**(color:**Colors**.black),

                      ),

                     onSaved: (value) {

                          \_userName = value;

                        },

                    ),

                  ),

**SizedBox**(

                    height: 5,

                  ),

**Container**(

                    width: **MediaQuery**.**of**(context).size.width/1.2,

                    height: 50,

                    padding: **EdgeInsets**.**only**(

                      top: 4,left: 16, right: 16, bottom: 4

                    ),

                    decoration: **BoxDecoration**(

                      borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                      ),

                      color: **Colors**.white,

                      boxShadow: [

**BoxShadow**(

                          color: **Colors**.black12,

                          blurRadius: 5

                        )

                      ]

                    ),

                    child: **TextFormField**( style: **TextStyle**(

                  color: **Colors**.black

                ),

                       key: **ValueKey**('Password'),validator: (value) {

                        if (value.isEmpty || value.length < 7) {

                          return 'Password must be at least 7 characters long.';

                        }

                        return null;

                      },

                      obscureText: true,

                      decoration: **InputDecoration**(

                        border: **InputBorder**.none,

                          hintText: 'Password',

                          hintStyle: **TextStyle**(color:**Colors**.black),

                      ),

                       onSaved: (value) {

                        \_userPassword = value;

                      }

                    ),

                  ),

**SizedBox**(

                    height: 15,

                  ),

**Padding**(

                  padding: const **EdgeInsets**.**fromLTRB**(15, 0, 0, 0),

                  child: **Padding**(

                    padding: const **EdgeInsets**.**fromLTRB**(10, 0, 0, 0),

                    child: **Container**(

                            margin: **EdgeInsets**.**only**(top: 5),

                            child: **Row**(

                              mainAxisAlignment: **MainAxisAlignment**.start,

                              children: <**Widget**>[

**Checkbox**(

                                  value: \_checked,

                                  onChanged: (**bool** b){

**setState**(() {

                                      \_checked = b;

                                    });

                                  },

                                ),

**Container**(

                                  margin: **EdgeInsets**.**only**(left: 5),

                                  child: **Text**("Accept Terms Conditions"),

                                ),

                              ],

                            ),

                          ),

                  )

                              ),

**SizedBox**(

                    height: 15,

                  ),

**InkWell**(

                    onTap: \_trySubmit,

                    child: **Container**(

                      child: **ElevatedButton**(

                                  onPressed: \_checked ? \_trySubmit : null,

                                  child: **Text**('REGISTER')

                                ),

                      height: 50,

                      width: **MediaQuery**.**of**(context).size.width/1.2,

                      decoration: **BoxDecoration**(

                        gradient: **LinearGradient**(

                          colors: [

**Color**(0xff6bceff),

**Color**(0xFF00abff),

                          ],

                        ),

                        borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(50)

                        )

                      ),

                      ),

                    ),

                ],

              ),)

            ),

**InkWell**(

              child: **Row**(

                mainAxisAlignment: **MainAxisAlignment**.center,

                children: <**Widget**>[

**Text**("Have an account ?"),

**Text**("Login",style: **TextStyle**(color: **Color**(0xff6bceff)),),

                ],

              ),

              onTap: (){

**Navigator**.**pushNamed**(context, '/login');

              },

            ),

          ],

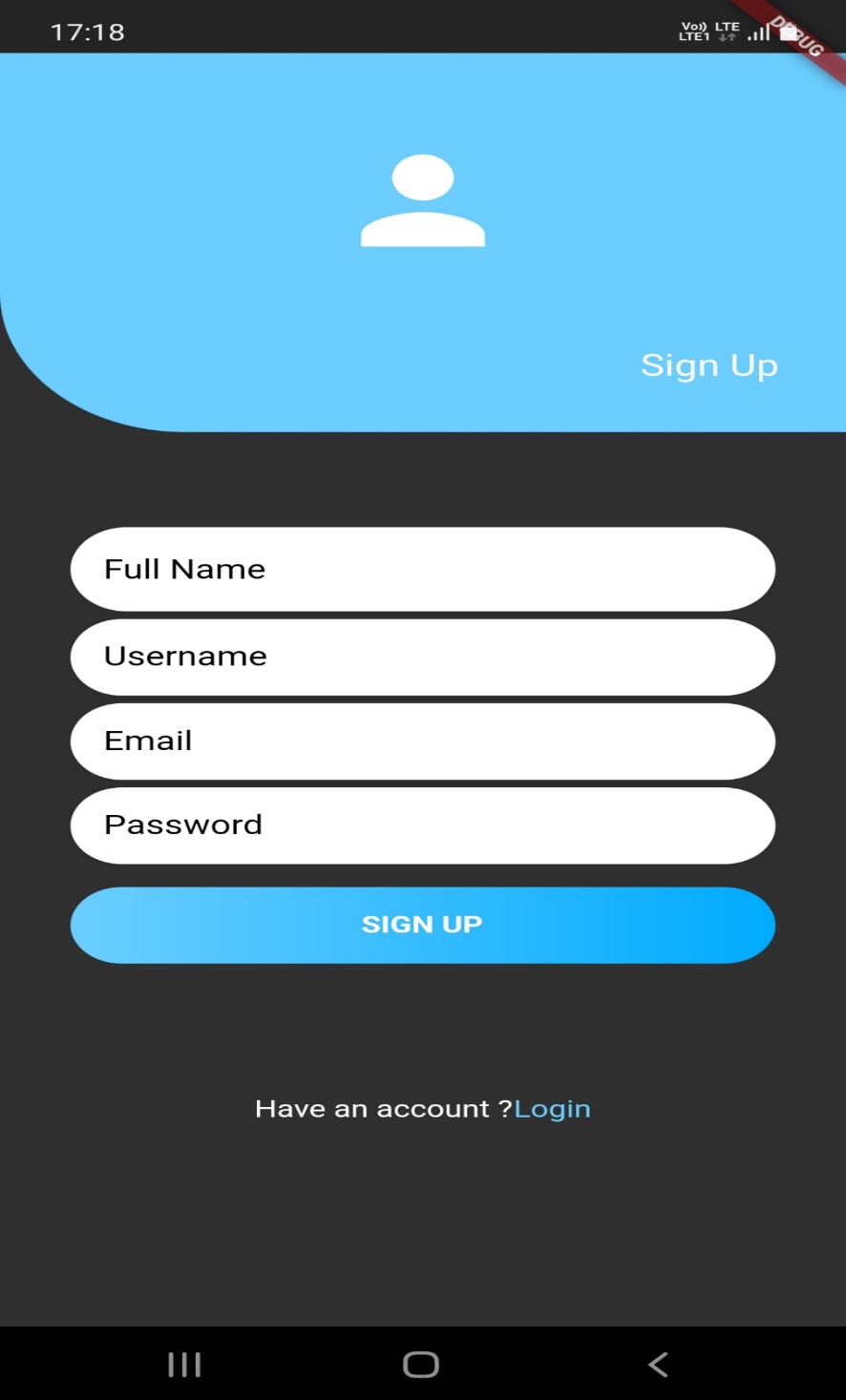
        ),

      ),

    );

  }

}



**10.4 SplashScreen.dart File**

import 'package:flutter/material.dart';

import 'package:kuva/screen/login.dart';

import 'package:splashscreen/splashscreen.dart';

class **SScreen** extends **StatefulWidget** {

  @override

**\_SScreenState** **createState**() => **\_SScreenState**();

}

class **\_SScreenState** extends **State**<**SScreen**> {

  @override

**Widget** **build**(**BuildContext** context) {

    return **SplashScreen**(

      seconds: 5,

      backgroundColor: **Colors**.black,

*// image: Image.asset('images/kuva2.jpg'),*

      loaderColor: **Colors**.white,

      photoSize: 100,

      navigateAfterSeconds: **LoginPage**(),

    );

  }

}



**10.5 TabScreen.dart File**

import 'package:flutter/material.dart';

import 'package:kuva/models/affirmations.dart';

import 'package:kuva/screen/affirmation\_detail\_screen.dart';

import 'package:kuva/screen/categories\_screen.dart';

import 'package:kuva/widgets/category\_item.dart';

import 'Category\_affirmation\_screen.dart';

import 'favorites\_screen.dart';

class **TabsScreen** extends **StatefulWidget**

{

   static var len1;

**int** akflag=0;

**TabsScreen**();

@override

**\_TabsScreen** **createState**()=>**\_TabsScreen**();

}

class **\_TabsScreen** extends **State**<**TabsScreen**>

{

  var akid;

  var aktt;

**int** \_selectedPageindex=0;

**List**<**Map**<**String**,**Object**>>\_pages;

  @override

  void **initState**() {

    \_pages=[{'page':**CategoriesScreen**(),'title':'Categories'},

  ];

    super.**initState**();

  }

  void **\_selectPage**(**int** index)

  {

**setState**(() {

  \_selectedPageindex=index;

});

  }

  @override

**Widget** **build**(**BuildContext** context)

  {

    return  **WillPopScope**(

      onWillPop: () async => false,

          child: **Scaffold**(

     appBar: **AppBar**(title: **Text**(\_pages[\_selectedPageindex]['title']), actions: <**Widget**>[

**RaisedButton**(

        child: **Text**("Submit"),

        onPressed:(){

**CategoryItem**.tlist2.**toSet**().**toList**();

**CategoryItem**.tlist2=**CategoryItem**.tlist2.**toSet**().**toList**();

**CategoryItem**.idlist2.**toSet**().**toList**();

**CategoryItem**.idlist2=**CategoryItem**.idlist2.**toSet**().**toList**();

**print**(**CategoryItem**.tlist2);

           aktt=**CategoryItem**.tlist2[1];

           akid=**CategoryItem**.idlist2[1];

**print**('AK id is');

**print**(akid);

**print**(aktt);

**print**('Id are:');

**print**(**CategoryItem**.idlist2);

**print**(**CategoryItem**.tlist2[0]);

**print**(**CategoryItem**.tlist2[1]);

**TabsScreen**.len1=(**CategoryItem**.tlist2.length)-1;

**Navigator**.**of**(context).**pushNamed**(

**CategoryAffirmationScreen**.routeName,

       arguments: {

         'id':**CategoryItem**.idlist2[**AffirmationDetailScreen**.roncount].**toString**() ,

         'title': **CategoryItem**.tlist2[**AffirmationDetailScreen**.roncount].**toString**(),

       },

     );

        },

        color: **Colors**.pink,

        textColor: **Colors**.white,

        splashColor: **Colors**.grey,

        padding: **EdgeInsets**.**fromLTRB**(10, 10, 10, 10),

      ),

**SizedBox**(

              width: 20.0,

            ),

          ],)

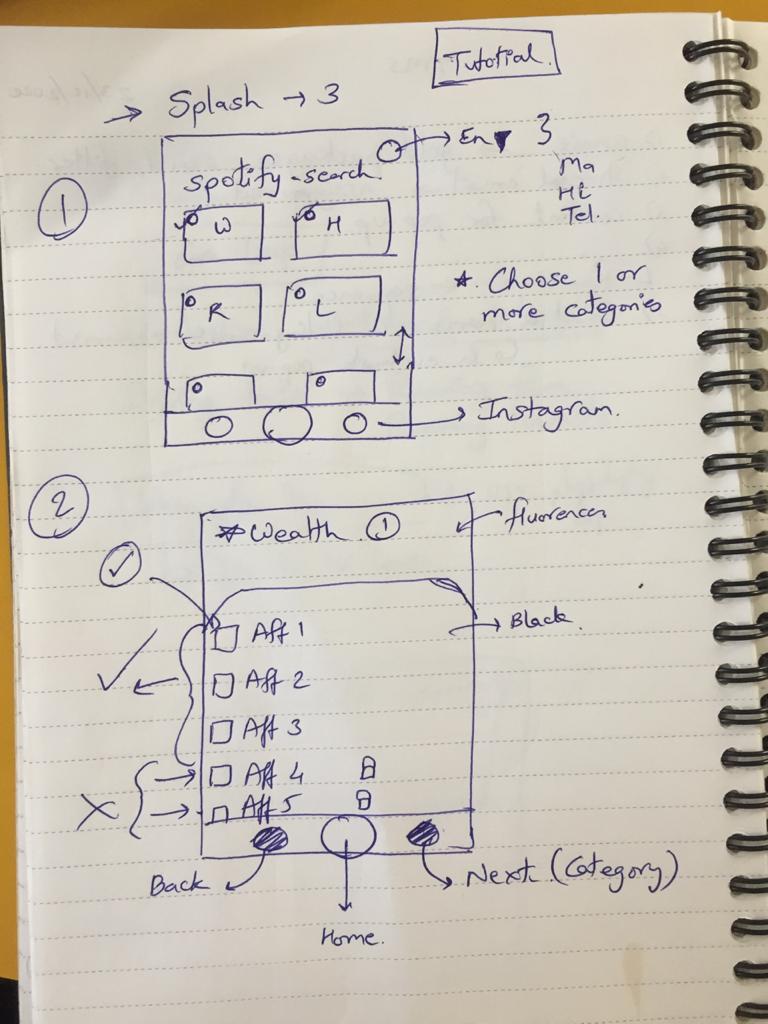
       , body: \_pages[\_selectedPageindex]['page'],

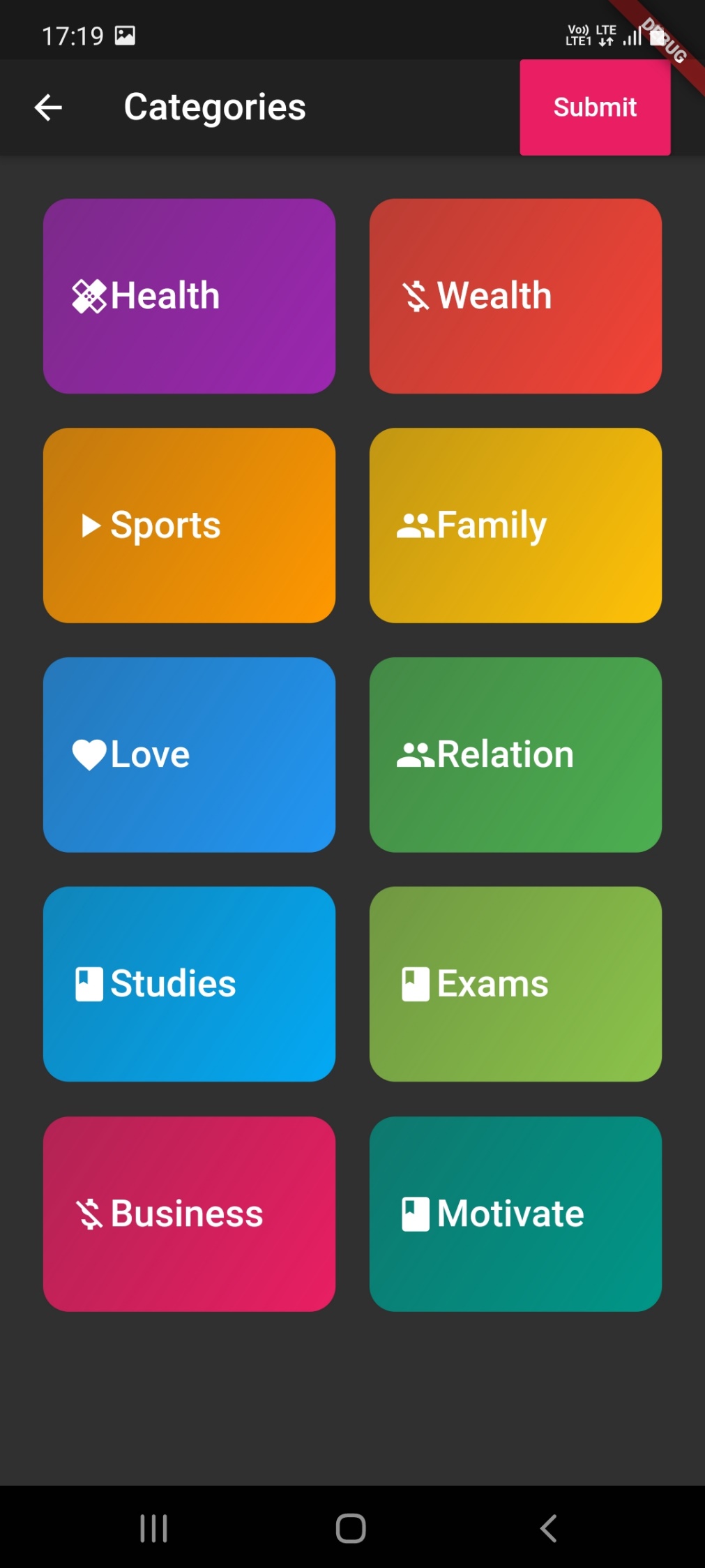
      ),

    );

  }

}





**10.5 CategoriesScreen.dart**

import 'package:flutter/material.dart';

import '../dummy\_data.dart';

import '../widgets/category\_item.dart';

class **CategoriesScreen** extends **StatelessWidget** {

  static **List** items;

  final nameRoute='/category-affirmation';

  @override

**Widget** **build**(**BuildContext** context) {

    return **GridView**(

        padding:

         const **EdgeInsets**.**all**(25)

        ,children:

**DUMMY\_CATEGORIES**.**map**((catData)=>**CategoryItem**(catData.id,catData.title,catData.color,catData.color,catData.iconname)).**toList**(),gridDelegate:

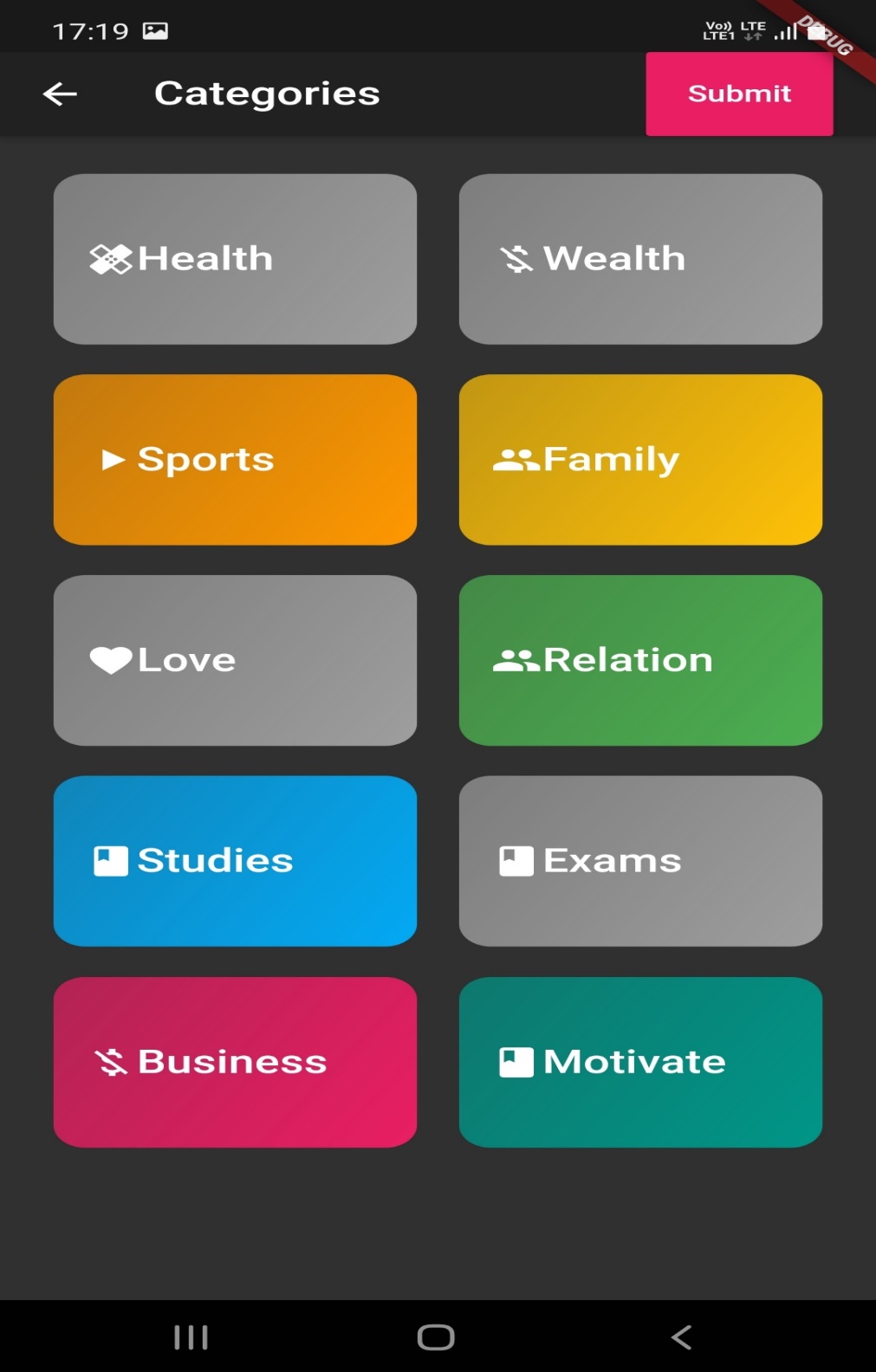
**SliverGridDelegateWithMaxCrossAxisExtent**(

      maxCrossAxisExtent: 200,childAspectRatio: 3/2,crossAxisSpacing: 20,mainAxisSpacing: 20)

      );

  }

}



**10.6 CategoriesItem.dart File**

import 'package:flutter/material.dart';

import 'package:kuva/screen/categories\_screen.dart';

import 'package:kuva/screen/tabs\_screen.dart';

import 'package:kuva/widgets/Items.dart';

import '../screen/Category\_affirmation\_screen.dart';

class **CategoryItem** extends **StatefulWidget** {

  final **String** id;

  final **String** title;

**Color** color;

**Color** color2;

  final **Icon** iconname;

static **List** tlist2=[""];

static **List** idlist2=[""];

**CategoryItem**(this.id, this.title, this.color,this.color2,this.iconname);

  @override

**\_CategoryItemState** **createState**() => **\_CategoryItemState**();

}

class **\_CategoryItemState** extends **State**<**CategoryItem**> {

 final **List** tlist1=[""];

 final **List** idlist1=[""];

  var flag=0;

   void **selectCategory**(**BuildContext** ctx) {

**Navigator**.**of**(ctx).**pushNamed**(

**CategoryAffirmationScreen**.routeName,

      arguments: {

        'id': widget.id,

        'title': widget.title,

      },

    );

  }

**building**()

{

**setState**(() {

     flag==1?widget.color=**Colors**.amber:widget.color=**Colors**.blue;

  });

}

  @override

**Widget** **build**(**BuildContext** context) {

    return **GestureDetector**(

      onTap: ()

      {

        flag==0?flag=1:flag=0;

**print**(widget.title+" flag is ");

**print**(flag);

**setState**(() {

          if(flag==0){

           widget.color=widget.color2;

**CategoryItem**.tlist2.**remove**(widget.title);

**CategoryItem**.idlist2.**remove**(widget.id);

             }

             else{

                widget.color=**Colors**.grey;

**print**(widget.title);

                 tlist1.**add**(widget.title);

                 idlist1.**add**(widget.id);

**CategoryItem**.tlist2=**CategoryItem**.tlist2+tlist1;

**CategoryItem**.idlist2=**CategoryItem**.idlist2+idlist1;

             }

        });

      },

          child: **InkWell**(

        splashColor: **Theme**.**of**(context).primaryColor,

        borderRadius: **BorderRadius**.**circular**(15),

        child: **Container**(

          padding: const **EdgeInsets**.**all**(15),

          child: **Row**(

            children: <**Widget**>[

              widget.iconname,

**Text**(

                widget.title,

                style: **Theme**.**of**(context).textTheme.title,

              ),

            ],

          ),

          decoration: **BoxDecoration**(

            gradient: **LinearGradient**(

              colors: [

                widget.color.**withOpacity**(0.7),

                widget.color,

              ],

              begin: **Alignment**.topLeft,

              end: **Alignment**.bottomRight,

            ),

            borderRadius: **BorderRadius**.**circular**(15),

          ),

        ),

      ),

    );

  }

}

**10.7 CategoryAffirmationScreen.dart**

import 'package:flutter/material.dart';

import 'package:kuva/models/affirmations.dart';

import '../widgets/affirmation.dart';

import '../models/affirmations.dart';

class **CategoryAffirmationScreen** extends **StatefulWidget** {

  final **List**<**Affirmations**> availableAffirmation;

**CategoryAffirmationScreen**(this.availableAffirmation);

  static const routeName = '/category-affirmation';

  @override

**\_CategoryAffirmationScreenState** **createState**() => **\_CategoryAffirmationScreenState**();

}

class **\_CategoryAffirmationScreenState** extends **State**<**CategoryAffirmationScreen**> {

**String** categoryTitle;

**List**<**Affirmations**> displayedAffirmations;

  var \_loadedInitData = false;

  @override

  void **initState**() {

    super.**initState**();

  }

  @override

  void **didChangeDependencies**() {

    if (!\_loadedInitData) {

      final routeArgs =

**ModalRoute**.**of**(context).settings.arguments as **Map**<**String**, **String**>;

      categoryTitle = routeArgs['title'];

      final categoryId = routeArgs['id'];

      displayedAffirmations = widget.availableAffirmation.**where**((aff) {

        return aff.categories.**contains**(categoryId);

      }).**toList**();

      \_loadedInitData = true;

    }

    super.**didChangeDependencies**();

  }

  void **\_removeAffirmation**(**String** affId) {

**setState**(() {

      displayedAffirmations.**removeWhere**((aff) => aff.id == affId);

    });

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **Scaffold**(

      appBar: **AppBar**(

        title: **Text**(categoryTitle),

      ),

      body: **ListView**.**builder**(

        itemBuilder: (ctx, index) {

          return **AffirmationItem**(

            id: displayedAffirmations[index].id,

            title: displayedAffirmations[index].title,

            imageUrl: displayedAffirmations[index].imageUrl,

          );

        },

        itemCount: displayedAffirmations.length,

      ),

    );

  }

}

**10.8 AffirmationItem.dart**

import 'package:flutter/material.dart';

import '../screen/affirmation\_detail\_screen.dart';

import '../models/affirmations.dart';

class **AffirmationItem** extends **StatelessWidget** {

  final **String** id;

  final **String** title;

  final **String** imageUrl;

**AffirmationItem**({

    @required this.id,

    @required this.title,

    @required this.imageUrl,

  });

  void **selectMeal**(**BuildContext** context) {

**Navigator**.**of**(context).**pushNamed**(

**AffirmationDetailScreen**.routeName,

      arguments: id,

    ).**then**((result){

    });

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **InkWell**(

      onTap: () => **selectMeal**(context),

      child: **Card**(

        shape: **RoundedRectangleBorder**(

          borderRadius: **BorderRadius**.**circular**(15),

        ),

        elevation: 4,

        margin: **EdgeInsets**.**all**(10),

        child: **Column**(

          children: <**Widget**>[

**Stack**(

              children: <**Widget**>[

**ClipRRect**(

                  borderRadius: **BorderRadius**.**only**(

                    topLeft: **Radius**.**circular**(15),

                    topRight: **Radius**.**circular**(15),

                  ),

                  child: **Image**.**network**(

                    imageUrl,

                    height: 250,

                    width: **double**.infinity,

                    fit: **BoxFit**.cover,

                  ),

                ),

**Positioned**(

                  bottom: 20,

                  right: 10,

                  child: **Container**(

                    width: 300,

                    color: **Colors**.black54,

                    padding: **EdgeInsets**.**symmetric**(

                      vertical: 5,

                      horizontal: 20,

                    ),

                    child: **Text**(

                      title,

                      style: **TextStyle**(

                        fontSize: 26,

                        color: **Colors**.white,

                      ),

                      softWrap: true,

                      overflow: **TextOverflow**.fade,

                    ),

                  ),

                )

              ],

            ),

**Padding**(

              padding: **EdgeInsets**.**all**(20),

              child: **Row**(

                mainAxisAlignment: **MainAxisAlignment**.spaceAround,

                children: <**Widget**>[

**Row**(

                    children: <**Widget**>[

**SizedBox**(

                        width: 6,

                      ),

                    ],

                  ),

**Row**(

                    children: <**Widget**>[

**SizedBox**(

                        width: 6,

                      ),                      **Text**("Affirmations ",style: **TextStyle**(color: **Colors**.white,fontSize: 30,),)

                    ],

                  ),

**Row**(

                    children: <**Widget**>[

**SizedBox**(

                        width: 6,

                      ),

                    ],

                  ),

                ],

              ),

            ),

          ],

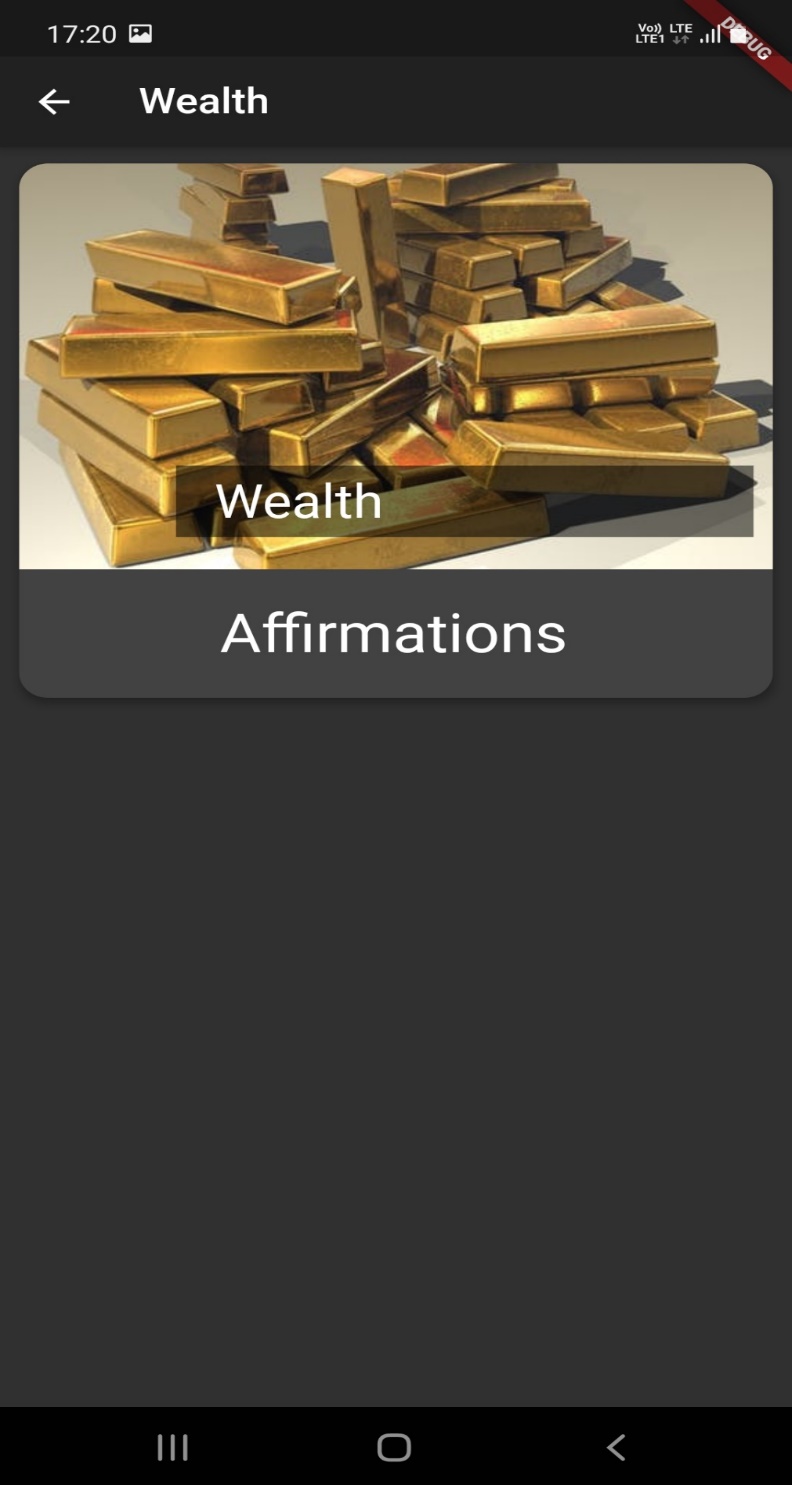
        ),

      ),

    );

  }

}



**10.9 AffirmationDetailScreen.dart**

import 'package:flutter/material.dart';

import 'package:kuva/screen/FirstScreen.dart';

import 'package:assets\_audio\_player/assets\_audio\_player.dart';

import 'package:kuva/models/affirmations.dart';

import 'package:kuva/screen/tabs\_screen.dart';

import 'package:kuva/widgets/category\_item.dart';

import '../dummy\_data.dart';

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:razorpay\_flutter/razorpay\_flutter.dart';

import 'package:slider\_button/slider\_button.dart';

import 'package:fluttertoast/fluttertoast.dart';

import 'Category\_affirmation\_screen.dart';

class **AffirmationDetailScreen** extends **StatefulWidget** {

  var count=1;

  static var holder\_3=[];

  static var ttlist=[];

  static var imglist=[];

  static const routeName = '/meal-detail';

  final **Function** toggleFavorite;

  final **Function** isFavorite;

 static var flag=1;

 static var roncount=1;

 var hl;

**AffirmationDetailScreen**(this.toggleFavorite, this.isFavorite);

  @override

**\_AffirmationDetailScreenState** **createState**() => **\_AffirmationDetailScreenState**();

}

class **\_AffirmationDetailScreenState** extends **State**<**AffirmationDetailScreen**> {

**Map**<**String**,**bool**> map2 = {};

 var  count=1;

**implicit**()

 {

   final mealId = **ModalRoute**.**of**(context).settings.arguments as **String**;

    final selectedAff = **DUMMY\_Affirmations**.**firstWhere**((meal) => meal.id == mealId);

selectedAff.affirmations.**forEach**((selectedAff) => map2[selectedAff] = false);

count=count+1;

 }

var holder\_1 = [''];

var holder\_2=[''];

var tt\_2=[''];

var img2=[''];

**getItems**(**String** title,**String** imgurl){

    map2.**forEach**((key, value) {

      if(value == true)

      {

        holder\_1.**add**(key);

      }

    });

**print**("Holder1::");

**print**(holder\_1);

**AffirmationDetailScreen**.holder\_3.**addAll**(holder\_1);

  widget.hl=holder\_1.length-1;

**print**('Holder1 length');

**print**(widget.hl);

holder\_1.**clear**();

   for(var i=0;**i**<widget.hl;i++)

   {

     tt\_2.**insert**(i,title);

   }

**print**("tt2 is ");

**print**(tt\_2);

**AffirmationDetailScreen**.ttlist.**addAll**(tt\_2);

**print**('Title list here is');

**print**(**AffirmationDetailScreen**.ttlist);

for(var i=0;**i**<widget.hl;i++)

   {

     img2.**insert**(i,imgurl);

   }

**print**("img2 is ");

**print**(img2);

**AffirmationDetailScreen**.imglist.**addAll**(img2);

**print**('Title list here is');

**print**(**AffirmationDetailScreen**.imglist);

  }

**Razorpay** \_razorpay;

**Widget** **buildSectionTitle**(**BuildContext** context, **String** text) {

    return **Container**(

      margin: **EdgeInsets**.**symmetric**(vertical: 10),

      child: **Text**(

        text,

        style: **Theme**.**of**(context).textTheme.title,

      ),

    );

  }

**Widget** **buildContainer**(**Widget** child) {

    return **Container**(

      decoration: **BoxDecoration**(

        border: **Border**.**all**(color: **Colors**.grey),

        borderRadius: **BorderRadius**.**circular**(10),

      ),

      margin: **EdgeInsets**.**all**(10),

      padding: **EdgeInsets**.**all**(10),

      height: 350,

      width: 350,

      child: child,

    );

  }

**onPlayAudio**(**String** name) async{

**setState**(() {

**AssetsAudioPlayer** assetsAudioPlayer = **AssetsAudioPlayer**();

  assetsAudioPlayer.**open**(

**Audio**(name)

  );

  });

}

  @override

**Widget** **build**(**BuildContext** context) {

    final mealId = **ModalRoute**.**of**(context).settings.arguments as **String**;

    final selectedAff = **DUMMY\_Affirmations**.**firstWhere**((meal) => meal.id == mealId);

     var count2=1;

    if (count==1){

**print**(count);

**implicit**();

    }

    return **WillPopScope**(

          child: **Scaffold**(

        appBar: **AppBar**(

          title: **Text**('${selectedAff.title}'),

        ),

        body: **SingleChildScrollView**(

          child: **Material**(

                    child: **Column**(

              children: <**Widget**>[

**Container**(

                  height: 300,

                  width: **double**.infinity,

                  child: **Image**.**network**(

                    selectedAff.imageUrl,

                    fit: **BoxFit**.cover,

                  ),

                ),**RaisedButton**(

        child: **Text**("Submit"),

        onPressed:(){

**getItems**(selectedAff.title,selectedAff.imageUrl);

**print**('Title list');

**print**(**AffirmationDetailScreen**.ttlist);

**print**('Holder1:');

**print**(holder\_1);

**print**('holder3:');

**AffirmationDetailScreen**.holder\_3.**removeWhere**((value) => value == "");

**print**(**AffirmationDetailScreen**.holder\_3);

**AffirmationDetailScreen**.ttlist.**removeWhere**((value) => value == "");

**print**(**AffirmationDetailScreen**.ttlist);

**AffirmationDetailScreen**.imglist.**removeWhere**((value) => value == "");

**print**(**AffirmationDetailScreen**.imglist);

**AffirmationDetailScreen**.roncount=**AffirmationDetailScreen**.roncount+1;

          if(**AffirmationDetailScreen**.roncount>**TabsScreen**.len1)

     {

**setState**(() {

**Navigator**.**push**(context, **MaterialPageRoute**(builder: (context) => **FirstScreen**(selectedAff.imageUrl,selectedAff.title,**AffirmationDetailScreen**.holder\_3,**AffirmationDetailScreen**.ttlist,**AffirmationDetailScreen**.imglist)));

       });

     }

**Navigator**.**of**(context).**pushNamed**(

**CategoryAffirmationScreen**.routeName,

       arguments: {

         'id':**CategoryItem**.idlist2[**AffirmationDetailScreen**.roncount].**toString**() ,

         'title': **CategoryItem**.tlist2[**AffirmationDetailScreen**.roncount].**toString**(),

       },

     );

        },

        color: **Colors**.pink,

        textColor: **Colors**.white,

        splashColor: **Colors**.grey,

        padding: **EdgeInsets**.**fromLTRB**(10, 10, 10, 10),

      ),

**buildSectionTitle**(context, 'Affirmations'),

**buildContainer**(

**ListView**(

          children: map2.keys.**map**((**String** key) {

          if (**count2**<3) {

               count2=count2+1;

            return new **CheckboxListTile**(

              title: new **Text**(key),

              value: map2[key],

              activeColor: **Colors**.deepPurple[400],

              checkColor: **Colors**.white,

              onChanged: (**bool** value) {

**setState**(() {

                  map2[key] = value;

                });

              },

            );

          }

           else {

            count2=count2+1;

                 return new  **CheckboxListTile**(

             secondary: const **Icon**(**Icons**.lock),

                  title: new **Text**(key),

                  value: map2[key],

                  activeColor: **Colors**.deepPurple[400],

                  checkColor: **Colors**.white,

                  onChanged: (**bool** value) {

**showModalBottomSheet**(

                              context: context,

                              builder: (**BuildContext** bContext) {

                                return **Container**(

                                    decoration: **BoxDecoration**(

                                        borderRadius: **BorderRadius**.**only**(

                                      topLeft: const **Radius**.**circular**(10),

                                      topRight: const **Radius**.**circular**(10),

                                    )),

                                    child: **Material**(

                                                                    child: **Column**(

                                        children: [

**Padding**(

                                            padding: const **EdgeInsets**.**all**(3.0),

                                            child: **Text**(

                                              "Become a Premium Member",

                                              style: **TextStyle**(

                                                  fontWeight: **FontWeight**.bold,

                                                  fontSize: 24,

                                                  color: **Color**(0xffF8BC24)),

                                            ),

                                          ),

**Material**(

                      child: **Column**(children: [

**Container**(

                                                child: **Row**(

                                                  children: [

**Icon**(**Icons**.star\_border),

**Padding**(

                                                      padding: const **EdgeInsets**.**all**(5.0)

                                                      child: **Text**(

                                                        "   Mirror",

                                                       style: **TextStyle**(fontSize: 18.0),

                                                      ),

                                                    ),

                                                  ],

                                                ),

                                              ),

**Container**(

                                                child: **Row**(

                                                  children: [

**Icon**(**Icons**.star\_border),

**Padding**(

                                                     padding: const **EdgeInsets**.**all**(5.0),

                                                      child: **Text**(

                                                        "   Multi-Language",

                                                       style: **TextStyle**(fontSize: 18.0),

                                                      ),

                                                    ),

                                                  ],

                                                ),

                                              ),

**Container**(

                                                child: **Row**(

                                                  children: [

**Icon**(**Icons**.star\_border),

**Text**(

                                                      "    List of affirmations",

                                                      style: **TextStyle**(fontSize: 18.0),

                                                    ),

                                                  ],

                                                ),

                                              ),

                                            ]),

                                          ),

**Material**(

    child: **Column**(children: [

**Row**(children: [

**Card**(

                                           child: **InkWell**(

                                                                                          child: **Container**(

                                                        width: **MediaQuery**.**of**(context)

                                                                .size

                                                                .width /

                                                            3,

                                                       height: **MediaQuery**.**of**(context)

                                                                .size

                                                                .height /

                                                            4.5,

                                                     margin: const **EdgeInsets**.**all**(30.0),

                                                    padding: const **EdgeInsets**.**all**(10.0),

                                                        decoration: **BoxDecoration**(

                                                            color: **Colors**.orange,

                                                            border: **Border**.**all**(

                                                     color: **Colors**.pink[800],

                                                                width:

                                                                    3.0),

                                                            borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(

                                                                    4.0)),

                                                            boxShadow: [

**BoxShadow**(

                                                                  blurRadius: 10,

                                                                  color: **Colors**.black,

                                                                  offset: **Offset**(1, 3))

                                                            ]

                                                            ),

                                                        child: **Column**(children: [                                                          **Center**(

                                                            child: **Text**(

                                                              "Rs. 99/Month",

                                                          textAlign: **TextAlign**.center,

                                                              style:

**TextStyle**(fontSize: 24.0),

                                                            ),

                                                          )

                                                        ]

                                                      )

                                                    ),

                                                      onTap:

                                                       openCheckout

                                                  ),

                                                ),

**Expanded**(

                                                  child: **Card**(

                                                                                          child: **InkWell**(

                                                                                              child: **Container**(

                                                        width: **MediaQuery**.**of**(context)

                                                                .size

                                                                .width /

                                                            3,

                                                        height: **MediaQuery**.**of**(context)

                                                                .size

                                                                .height /

                                                            4.5,

                                                     margin: const **EdgeInsets**.**all**(30.0),

                                                    padding: const **EdgeInsets**.**all**(10.0),

                                                        decoration: **BoxDecoration**(

                                                            color: **Colors**.orange,

                                                            border: **Border**.**all**(

                                                                color: **Colors**.pink[

                                                                    800],

                                                                width:

                                                                    3.0),

                                                     borderRadius: **BorderRadius**.**all**(

**Radius**.**circular**(

                                                                    4.0

                                                            boxShadow: [

**BoxShadow**(

                                                                  blurRadius: 10,

                                                                  color: **Colors**.black,

                                                                  offset: **Offset**(1, 3))

                                                            ]

                                                            ),

                                                        child: **Column**(

                                                          children: [

**Text**(

                                                              "Rs. 1188",

                                                              style: **TextStyle**(

                                                                fontSize: 17.0,

                                                 decoration: **TextDecoration**lineThrough,

                                                              ),

                                                            ),

**Text**(

                                                              "30% off!",

                                                              style: **TextStyle**(

                                                                fontSize: 11.0,

                                                              ),

                                                            ),

**Text**(

                                                              "Rs. 730/",

                                                              style: **TextStyle**(

                                                                fontSize: 24.0,

                                                              ),

                                                            ),

**Text**(

                                                              "Year",

                                                              style: **TextStyle**(

                                                                fontSize: 14.0,

                                                              ),

                                                            ),

**Text**(

                            "(ie Rs. 2/day - for self improvement)",

                                                        textAlign: **TextAlign**.center,

                                                              style: **TextStyle**(

                       fontStyle: **FontStyle**.italic,

                                                                fontSize: 8.0,

                                                              ),

                                                            )

                                                          ],

                                                        ),

                                                      ),

                                                      onTap: openCheckout2

                                                    ),

                                                  ),

                                                ),

                                              ]),

                                            ]),

                                          ),

**Expanded**(

                                              child: **SliderButton**(

                                            backgroundColor: **Colors**.amber,

                                            width: 200,

                                            action:openCheckout,

                                            label: **Text**(

                                              "Pay Now",

                                              style: **TextStyle**(

                                                  color: **Color**(0xff1A4645),

                                                  fontWeight: **FontWeight**.w200,

                                                  fontSize: 24),

                                            ),

                                            icon: **Text**(

                                              "x",

                                              style: **TextStyle**(

                                                color: **Colors**.black,

                                                fontWeight: **FontWeight**.w100,

                                                fontSize: 34,

                                              ),

                                            ),

                                          )),

                                        ],

                                      ),

                                    ));

                              });

                  },

            );

          }

          }).**toList**(),

        ),

                )],

            ),

          ),

        ),

        floatingActionButton: **FloatingActionButton**(

          child: **Icon**(

             widget.**isFavorite**(mealId) ? **Icons**.star : **Icons**.star\_border,

          ),

          onPressed: () => widget.**toggleFavorite**(mealId),

        ),

      ),

    );

  }

  void **initState**() {

    super.**initState**();

    \_razorpay = **Razorpay**();

    \_razorpay.on(**Razorpay**.**EVENT\_PAYMENT\_SUCCESS**, \_handlePaymentSuccess);

    \_razorpay.on(**Razorpay**.**EVENT\_PAYMENT\_ERROR**, \_handlePaymentError);

    \_razorpay.on(**Razorpay**.**EVENT\_EXTERNAL\_WALLET**, \_handleExternalWallet);

  }

  @override

  void **dispose**() {

    super.**dispose**();

    \_razorpay.**clear**();

  }

void **showSheet**()

{

}

  void **openCheckout**() async {

    var options = {

      'key': 'rzp\_live\_HSO9hrKe80xgAm',

      'amount': 9999,

      'name': 'Company.',

      'description': 'Affirmation',

      'prefill': {'contact': '9137028272', 'email': 'test@razorpay.com'},

      'external': {

        'wallets': ['paytm']

      }

    };

    try {

      \_razorpay.**open**(options);

    } catch (e) {

**debugPrint**(e);

    }

  }

 void **openCheckout2**() async {

    var options = {

      'key': 'rzp\_live\_HSO9hrKe80xgAm',

      'amount':73000,

      'name': 'Company.',

      'description': 'Affirmation',

      'prefill': {'contact': '9137028272', 'email': 'test@razorpay.com'},

      'external': {

        'wallets': ['paytm']

      }

    };

    try {

      \_razorpay.**open**(options);

    } catch (e) {

**debugPrint**(e);

    }

  }

  void **\_handlePaymentSuccess**(**PaymentSuccessResponse** response) {

**Fluttertoast**.**showToast**(

        msg: "SUCCESS: " + response.paymentId, timeInSecForIos: 4);

  }

**\_handlePaymentError**(**PaymentFailureResponse** response) {

   msg: "Fail";

  }

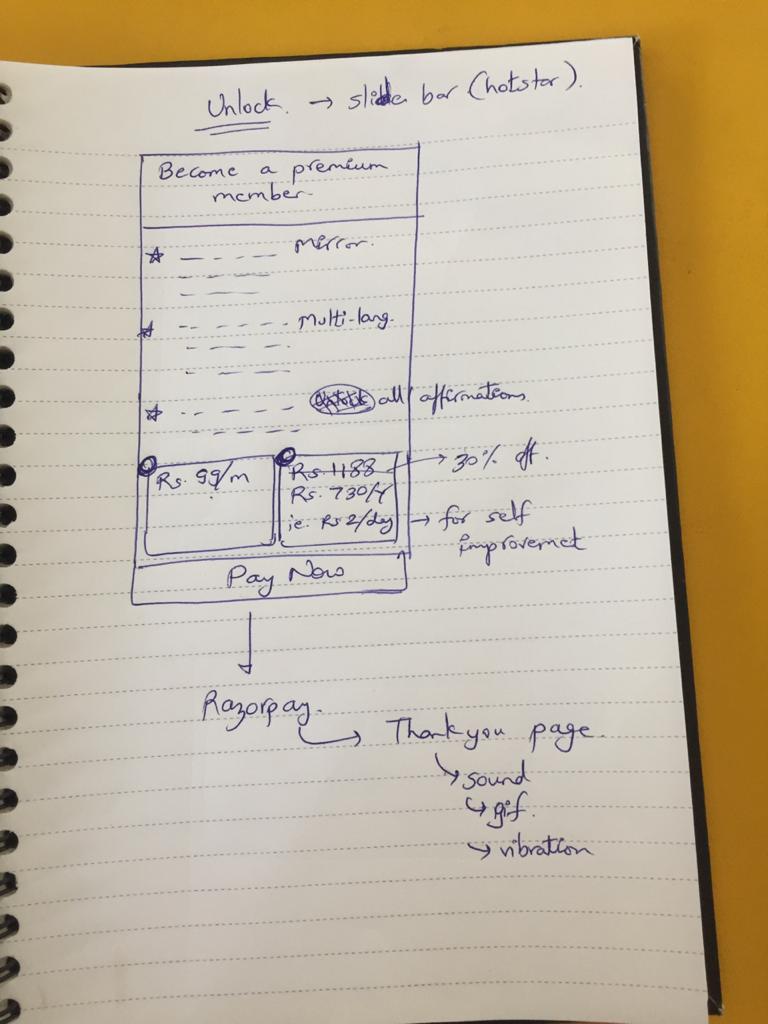
  void **\_handleExternalWallet**(**ExternalWalletResponse** response) {

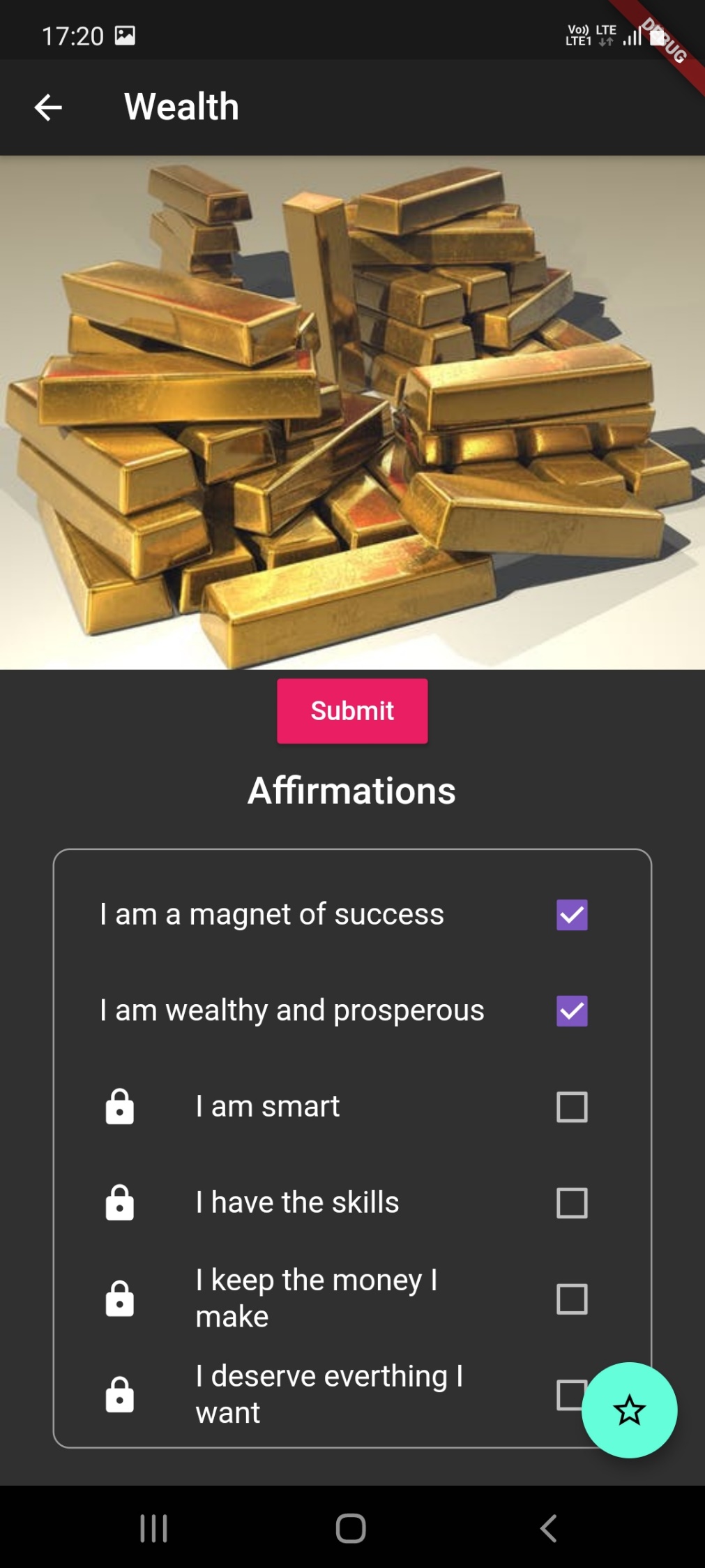
**Fluttertoast**.**showToast**(

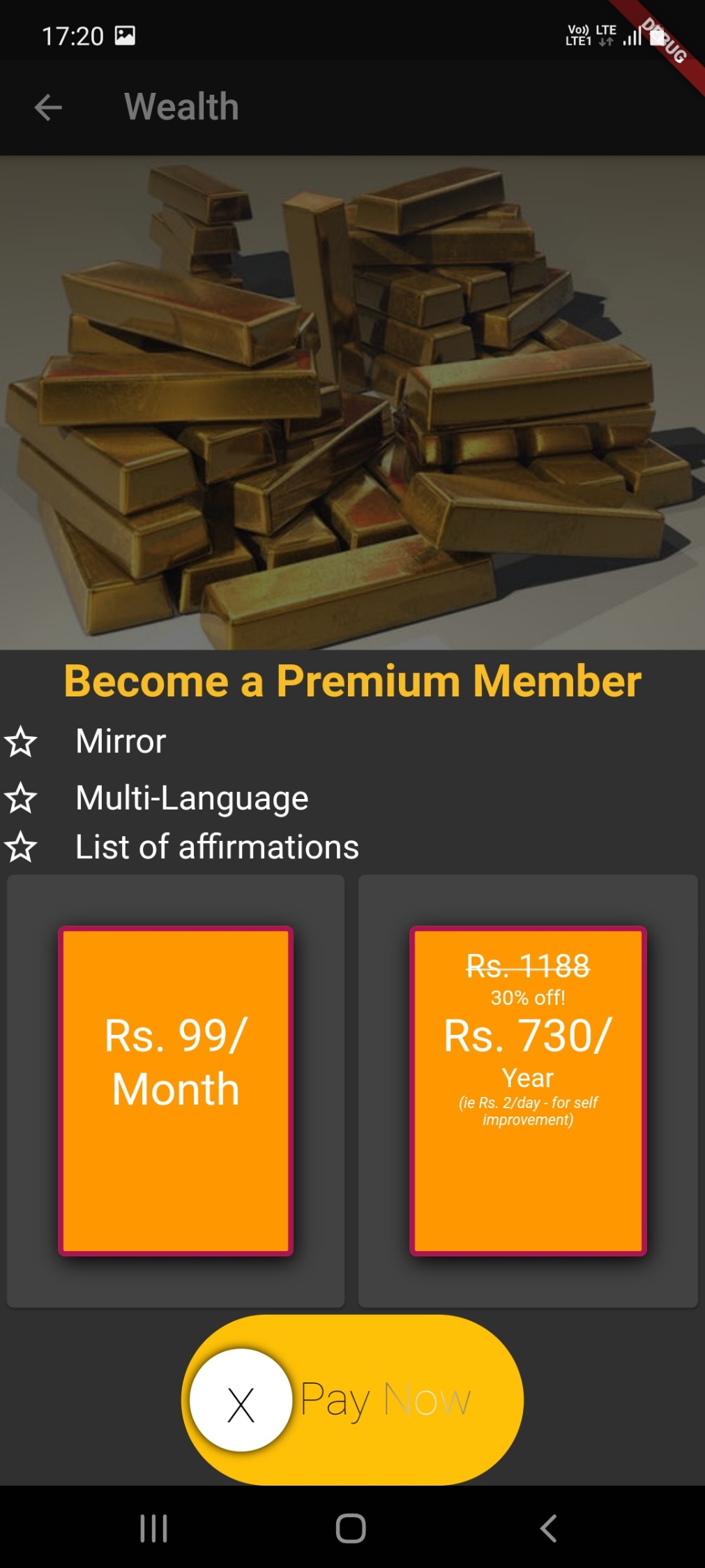
        msg: "EXTERNAL\_WALLET: " + response.walletName, timeInSecForIos: 4);

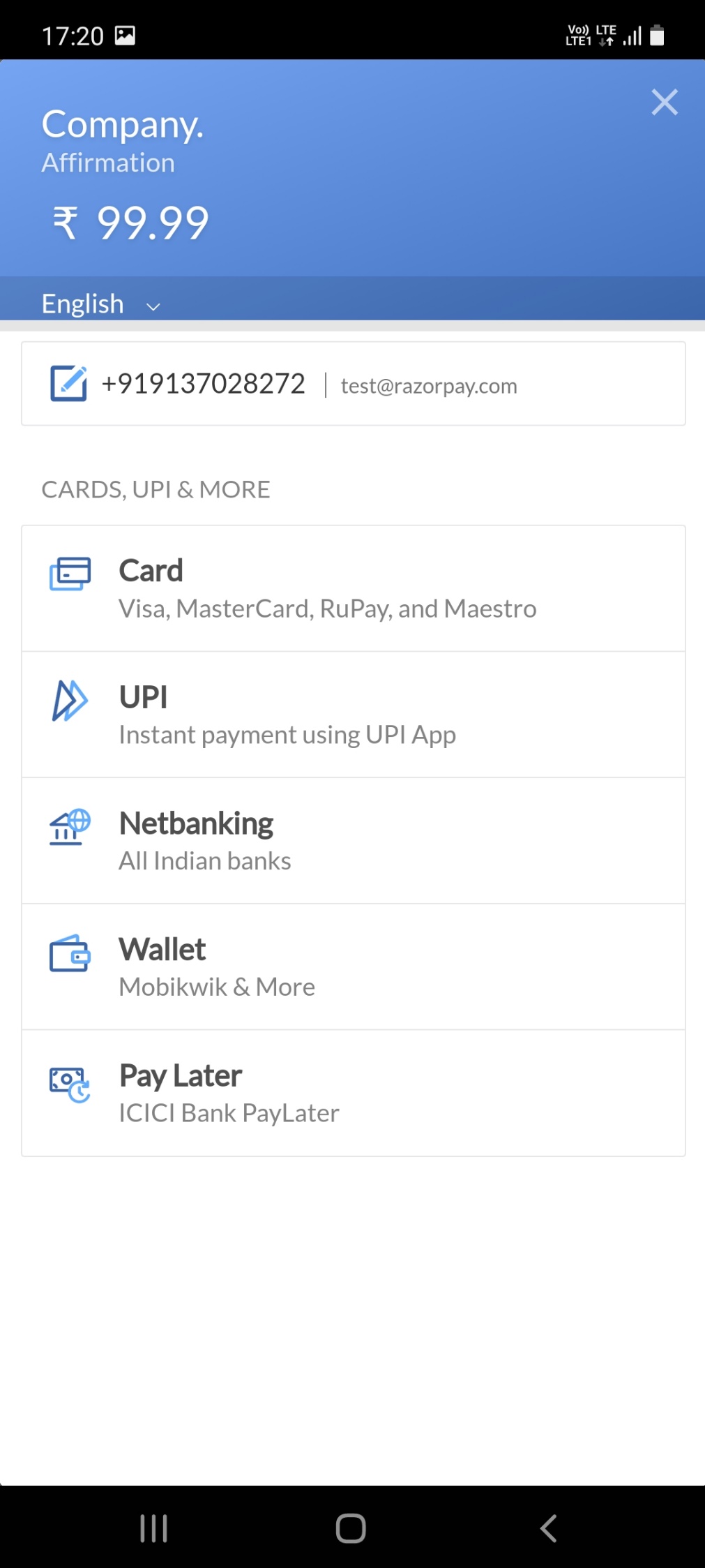
  }

}









**10.9 FirstScreen.dart**

import 'package:flutter/material.dart';

import 'package:kuva/enums.dart';

import 'package:kuva/models/menu\_info.dart';

import 'package:kuva/screen/settings\_screen.dart';

import 'package:kuva/views/homepage.dart';

import 'package:provider/provider.dart';

import 'package:assets\_audio\_player/assets\_audio\_player.dart';

class **FirstScreen** extends **StatefulWidget** {

**String** image;

**String** title;

**List** holder;

**List** ttlist;

**List** imgList;

**FirstScreen**(this.image, this.title, this.holder, this.ttlist, this.imgList);

  @override

**State**<**StatefulWidget**> **createState**() {

    return **\_FirstScreen**();

  }

}

class **\_FirstScreen** extends **State**<**FirstScreen**> {

**PageController** pageController = **PageController**(initialPage: 0);

**int** index = 4;

**bool** \_isFavorited = true;

  void **\_toggleFavorite**() {

**setState**(() {

      \_isFavorited = !\_isFavorited;

    });

  }

**onPlayAudio**(**String** name) async {

**setState**(() {

**AssetsAudioPlayer** assetsAudioPlayer = **AssetsAudioPlayer**();

      assetsAudioPlayer.**open**(**Audio**(name));

    });

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **MaterialApp**(

      home: **Scaffold**(

          appBar: **AppBar**(

            title: **Text**('Affirmations'),

            elevation: 0,

            backgroundColor: **Color**.**fromRGBO**(102, 70, 75, 1),

            actions: <**Widget**>[

**IconButton**(

                icon: const **Icon**(

**Icons**.settings,

                  color: **Colors**.white,

                ),

                onPressed: () {

**Navigator**.**push**(

                    context,

**MaterialPageRoute**(builder: (context) => **SettingsScreen**()),

                  );

                },

              ),

**SizedBox**(

                width: 20.0,

              ),

            ],

          ),

          backgroundColor: **Color**.**fromRGBO**(75, 53, 56, 1),

          body: **Container**(

            child: **PageView**(

                controller: pageController,

                scrollDirection: **Axis**.vertical,

                children: <**Widget**>[

                  for (var i = 0; i < widget.holder.length; i++)

**Column**(

                      crossAxisAlignment: **CrossAxisAlignment**.center,

                      mainAxisSize: **MainAxisSize**.max,

                      mainAxisAlignment: **MainAxisAlignment**.end,

                      children: <**Widget**>[

**Text**(

                          widget.ttlist[i],

                          style: **TextStyle**(

                              color: **Colors**.white,

                              fontSize: 30.0,

                              fontWeight: **FontWeight**.bold),

                        ),

**Image**.**network**(

                          widget.imgList[i],

                          height: 500,

                          width: 350,

                          fit: **BoxFit**.cover,

                        ),

**SizedBox**(

                          height: 16.0,

                        ),

**Container**(

                          decoration: **BoxDecoration**(

                              color: **Color**.**fromRGBO**(45, 35, 36, 1),

                              borderRadius: new **BorderRadius**.**only**(

                                topLeft: const **Radius**.**circular**(20.0),

                                topRight: const **Radius**.**circular**(20.0),

                              )),

                          height: 200,

                          width: 500,

                          child: **Align**(

                            alignment: **Alignment**.bottomLeft,

                            child: **Padding**(

                              padding: **EdgeInsets**.**all**(20.0),

                              child: **Column**(

                                crossAxisAlignment: **CrossAxisAlignment**.start,

                                children: <**Widget**>[

**SizedBox**(

                                    height: 5.0,

                                  ),

**Text**(

                                    widget.holder[i],

                                    style: **TextStyle**(

                                        color: **Colors**.white,

                                        letterSpacing: 2.0,

                                        fontSize: 30.0),

                                  ),

**Row**(

                                    children: [

**FloatingActionButton**(

                                        heroTag: "btn1",

                                        onPressed: () =>

**onPlayAudio**("assets/audio3.mp3"),

                                        child: **Icon**(**Icons**.music\_note),

                                        backgroundColor:

**Color**.**fromRGBO**(102, 70, 75, 1),

                                        foregroundColor: **Colors**.white,

                                      ),

**Spacer**(flex: 2),

**FloatingActionButton**.**extended**(

                                        heroTag: "btn2",

                                        onPressed: () {

**Navigator**.**push**(

                                            context,

**MaterialPageRoute**(

                                                builder: (context) =>

**ChangeNotifierProvider**<

**MenuInfo**>(

                                                      create: (context) =>

**MenuInfo**(

**MenuType**.clock),

                                                      child: **HomePage**(),

                                                    )),

                                          );

                                        },

                                        icon: **Icon**(**Icons**.alarm),

                                        backgroundColor:

**Color**.**fromRGBO**(102, 70, 75, 1),

                                        foregroundColor: **Colors**.white,

                                        label: **Text**("Alarm"),

                                      ),],

                                  ),

                                ],),

                            ),

                          ),

                        ),

                      ],

                    ),

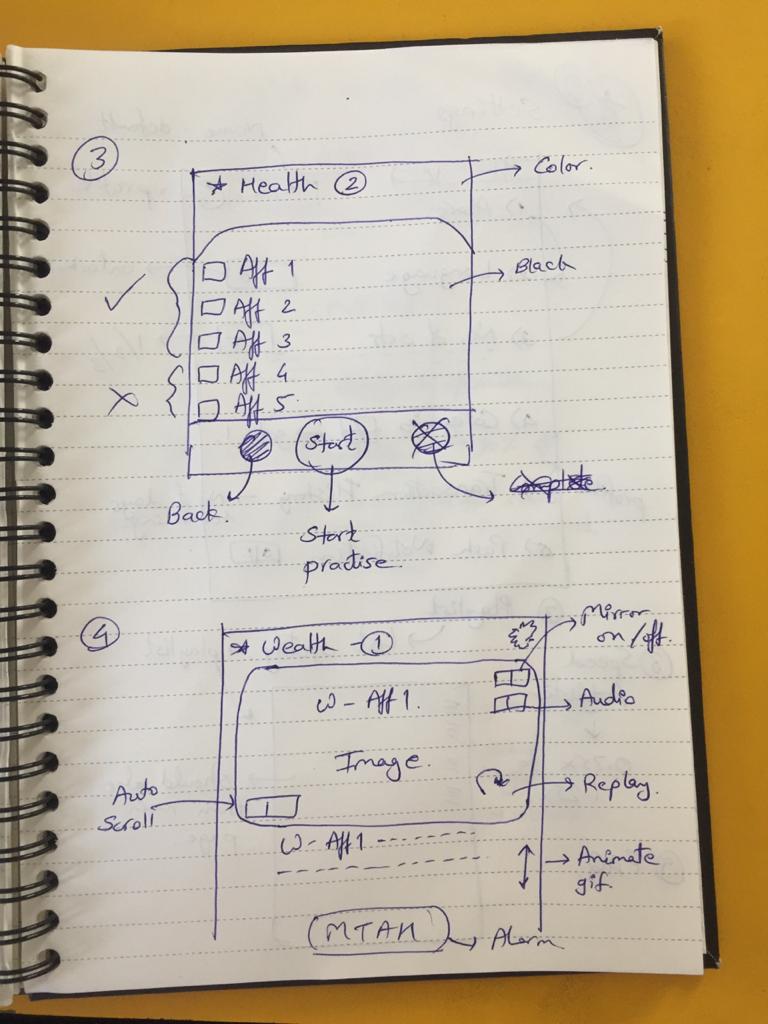
                ]),

          )),

    );

  }

}





**10.10 Settings.dart**

import 'package:flutter/material.dart';

import 'package:settings\_ui/settings\_ui.dart';

import 'set\_screen.dart';

import 'languages\_screen.dart';

class **SettingsScreen** extends **StatefulWidget** {

  @override

**\_SettingsScreenState** **createState**() => **\_SettingsScreenState**();

}

class **\_SettingsScreenState** extends **State**<**SettingsScreen**> {

**bool** lockInBackground = true;

**bool** notificationsEnabled = true;

  @override

**Widget** **build**(**BuildContext** context) {

    return **Scaffold**(

      appBar: **AppBar**(

        title: **Text**('Profile'),

        actions: [**Text**("Ronik")],

      ),

      body: **buildSettingsList**(),

    );

  }

**Widget** **buildSettingsList**() {

    return **SettingsList**(

      sections: [

**SettingsSection**(

          title: 'Settings',

          tiles: [

**SettingsTile**(

              title: 'Alarm',

              subtitle: 'Set Alarm',

              leading: **Icon**(**Icons**.alarm),

              onPressed: (context) {

              },

            ),

**SettingsTile**(

              title: 'Language',

              subtitle: 'English',

              leading: **Icon**(**Icons**.language),

              onPressed: (context) {

**Navigator**.**of**(context).**push**(**MaterialPageRoute**(

                  builder: (\_) => **LanguagesScreen**(),

                ));

              },

            ),

**SettingsTile**(

              title: 'Number of Sets',

              subtitle: 'Select Number of Sets',

              leading: **Icon**(**Icons**.add),

              onPressed: (context) {

**Navigator**.**of**(context).**push**(**MaterialPageRoute**(

                  builder: (\_) => **SetScreen**(),

                ));

              },

            ),

**SettingsTile**(

              title: 'Categories',

              subtitle: 'Goto Categories Page',

              leading: **Icon**(**Icons**.category),

              onPressed: (context) {

**Navigator**.**pushNamed**(context, '/');

              },

            ),

**SettingsTile**(

              title: 'Transaction History',

              subtitle: 'Number of Days Remaining in Subscription',

              leading: **Icon**(**Icons**.history),

              onPressed: (context) {

              },

            ),

**SettingsTile**.**switchTile**(

              title: 'Push Notification',

              leading: **Icon**(**Icons**.notifications),

              switchValue: lockInBackground,

              onToggle: (**bool** value) {

**setState**(() {

                  lockInBackground = value;

                  notificationsEnabled = value;

                });

              },

            ),

**SettingsTile**(

              title: 'Playlist',

              subtitle: 'Your Customized Playlist',

              leading: **Icon**(**Icons**.playlist\_play),

              onPressed: (context) {

              },

            ),

**SettingsTile**(

              title: 'Playback Speed',

              subtitle: 'Suit your style',

              leading: **Icon**(**Icons**.shutter\_speed),

              onPressed: (context) {

              },

            ),

**SettingsTile**(

              title: 'FAQ',

              subtitle: 'Answering Frequently Asked Questions',

              leading: **Icon**(**Icons**.question\_answer),

              onPressed: (context) {

              },

            ),

          ],

        ),

**CustomSection**(

          child: **Column**(

            children: [

**Padding**(

                padding: const **EdgeInsets**.**only**(top: 22, bottom: 8),

              ),

**Text**(

                'Kuva',

                style: **TextStyle**(color: **Color**(0xFF777777)),

              ),

            ],

          ),

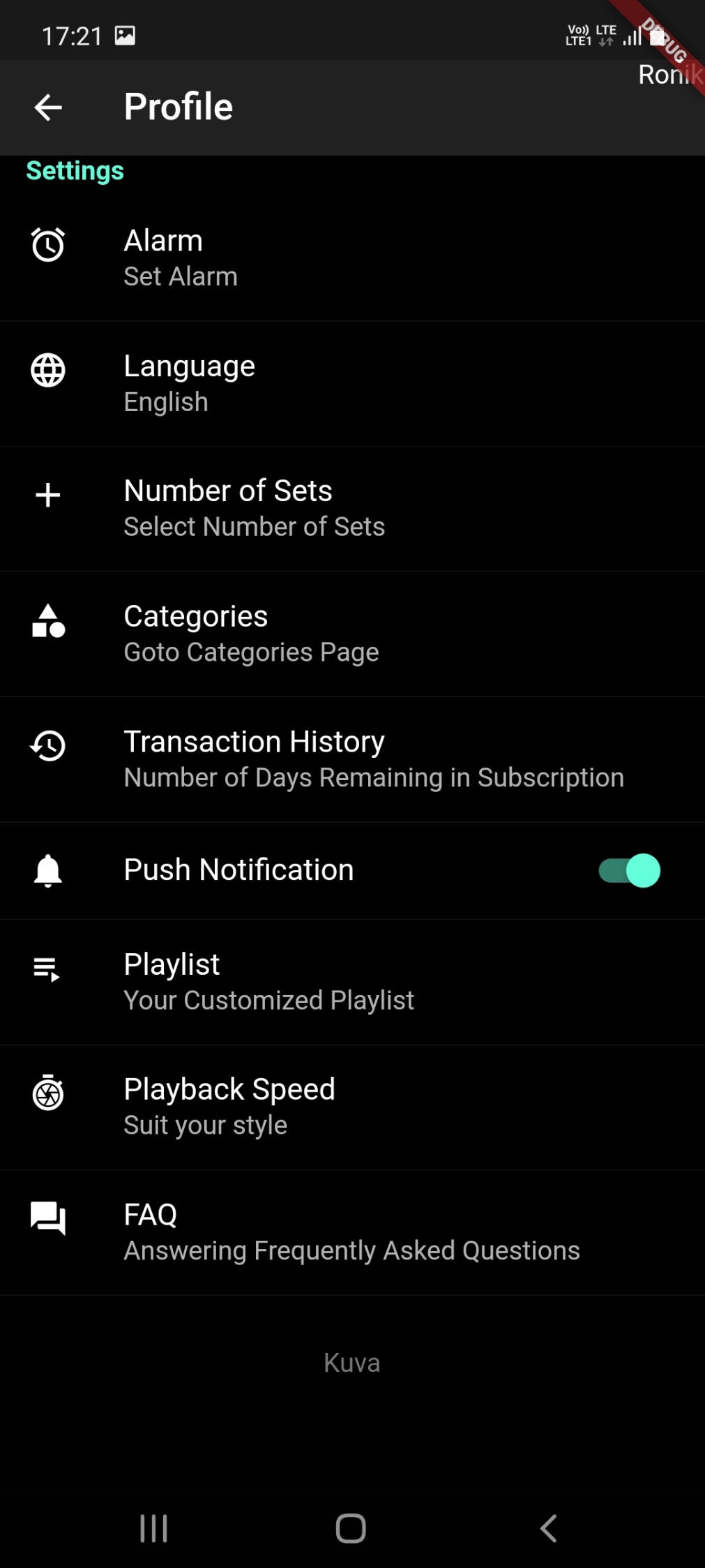
        ),

      ],

    );

  }

}





**10.11 HomePage.dart**

import 'package:kuva/constants/theme\_data.dart';

import 'package:kuva/data.dart';

import 'package:kuva/enums.dart';

import 'package:kuva/models/menu\_info.dart';

import 'package:kuva/views/alarm\_page.dart';

import 'package:kuva/views/clock\_page.dart';

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

class **HomePage** extends **StatefulWidget** {

  @override

**\_HomePageState** **createState**() => **\_HomePageState**();

}

class **\_HomePageState** extends **State**<**HomePage**> {

  @override

**Widget** **build**(**BuildContext** context) {

    return **Scaffold**(

      backgroundColor: **CustomColors**.pageBackgroundColor,

      body: **Row**(

        children: <**Widget**>[

**Column**(

            mainAxisAlignment: **MainAxisAlignment**.center,

            children: menuItems

                .**map**((currentMenuInfo) => **buildMenuButton**(currentMenuInfo))

                .**toList**(),

          ),

**VerticalDivider**(

            color: **CustomColors**.dividerColor,

            width: 1,

          ),

**Expanded**(

            child: **Consumer**<**MenuInfo**>(

              builder: (**BuildContext** context, **MenuInfo** value, **Widget** child) {

                if (value.menuType == **MenuType**.clock)

                  return **ClockPage**();

                else if (value.menuType == **MenuType**.alarm)

                  return **AlarmPage**();

                else

                  return **Container**(

                    child: **RichText**(

                      text: **TextSpan**(

                        style: **TextStyle**(fontSize: 20),

                        children: <**TextSpan**>[

**TextSpan**(text: 'Upcoming Tutorial\n'),

**TextSpan**(

                            text: value.title,

                            style: **TextStyle**(fontSize: 48),

                          ),

                        ],

                      ),

                    ),

                  );

              },

            ),

          ),

        ],

      ),

    );

  }

**Widget** **buildMenuButton**(**MenuInfo** currentMenuInfo) {

    return **Consumer**<**MenuInfo**>(

      builder: (**BuildContext** context, **MenuInfo** value, **Widget** child) {

        return **FlatButton**(

          shape: **RoundedRectangleBorder**(

              borderRadius: **BorderRadius**.**only**(topRight: **Radius**.**circular**(32))),

          padding: const **EdgeInsets**.**symmetric**(vertical: 16.0, horizontal: 0),

          color: currentMenuInfo.menuType == value.menuType

              ? **CustomColors**.menuBackgroundColor

              : **Colors**.transparent,

          onPressed: () {

            var menuInfo = **Provider**.**of**<**MenuInfo**>(context, listen: false);

            menuInfo.**updateMenu**(currentMenuInfo);

          },

          child: **Column**(

            children: <**Widget**>[

**Image**.**asset**(

                currentMenuInfo.imageSource,

                scale: 1.5,

              ),

**SizedBox**(height: 16),

**Text**(

                currentMenuInfo.title ?? '',

                style: **TextStyle**(

                    fontFamily: 'avenir',

                    color: **CustomColors**.primaryTextColor,

                    fontSize: 14),

              ),

            ],

          ),

        );

      },

    );

  }

}

**10.12 AlarmPage.dart**

import 'package:kuva/alarm\_helper.dart';

import 'package:kuva/constants/theme\_data.dart';

import 'package:kuva/data.dart';

import 'package:kuva/models/alarm\_info.dart';

import 'package:dotted\_border/dotted\_border.dart';

import 'package:flutter/material.dart';

import 'package:flutter\_local\_notifications/flutter\_local\_notifications.dart';

import 'package:intl/intl.dart';

import '../main.dart';

class **AlarmPage** extends **StatefulWidget** {

  @override

**\_AlarmPageState** **createState**() => **\_AlarmPageState**();

}

class **\_AlarmPageState** extends **State**<**AlarmPage**> {

**DateTime** \_alarmTime;

**String** \_alarmTimeString;

**AlarmHelper** \_alarmHelper = **AlarmHelper**();

**Future**<**List**<**AlarmInfo**>> \_alarms;

**List**<**AlarmInfo**> \_currentAlarms;

  @override

  void **initState**() {

    \_alarmTime = **DateTime**.**now**();

    \_alarmHelper.**initializeDatabase**().**then**((value) {

**print**('------database intialized');

**loadAlarms**();

    });

    super.**initState**();

  }

  void **loadAlarms**() {

    \_alarms = \_alarmHelper.**getAlarms**();

    if (mounted) **setState**(() {});

  }

  var selectedYear;

  @override

**Widget** **build**(**BuildContext** context) {

    return **Container**(

      padding: **EdgeInsets**.**symmetric**(horizontal: 32, vertical: 64),

      child: **Column**(

        crossAxisAlignment: **CrossAxisAlignment**.start,

        children: <**Widget**>[

**Text**(

            'Alarm',

            style: **TextStyle**(

                fontFamily: 'avenir',

                fontWeight: **FontWeight**.w700,

                color: **CustomColors**.primaryTextColor,

                fontSize: 24),

          ),

**Expanded**(

            child: **FutureBuilder**<**List**<**AlarmInfo**>>(

              future: \_alarms,

              builder: (context, snapshot) {

                if (snapshot.hasData) {

                  \_currentAlarms = snapshot.data;

                  return **ListView**(

                    children: snapshot.data.**map**<**Widget**>((alarm) {

                      var alarmTime =

**DateFormat**('dd-MM-yyyy/hh:mm aa').**format**(alarm.alarmDateTime);

                      var gradientColor = **GradientTemplate**

                          .gradientTemplate[alarm.gradientColorIndex].colors;

                      return **Container**(

                        margin: const **EdgeInsets**.**only**(bottom: 32),

                        padding: const **EdgeInsets**.**symmetric**(

                            horizontal: 16, vertical: 8),

                        decoration: **BoxDecoration**(

                          gradient: **LinearGradient**(

                            colors: gradientColor,

                            begin: **Alignment**.centerLeft,

                            end: **Alignment**.centerRight,

                          ),

                          boxShadow: [

**BoxShadow**(

                              color: gradientColor.last.**withOpacity**(0.4),

                              blurRadius: 8,

                              spreadRadius: 2,

                              offset: **Offset**(4, 4),

                            ),],

                          borderRadius: **BorderRadius**.**all**(**Radius**.**circular**(24)),

                        ),

                        child: **Column**(

                          crossAxisAlignment: **CrossAxisAlignment**.start,

                          children: <**Widget**>[

**Row**(

                              mainAxisAlignment: **MainAxisAlignment**.spaceBetween,

                              children: <**Widget**>[

**Row**(

                                  children: <**Widget**>[

**Icon**(

**Icons**.label,

                                      color: **Colors**.white,

                                      size: 24,

                                    ),

**SizedBox**(width: 8),

**Text**(

                                      alarm.title,

                                      style: **TextStyle**(

                                          color: **Colors**.white,

                                          fontFamily: 'avenir'),

                                    ),

                                  ],

                                ),

**Switch**(

                                  onChanged: (**bool** value) {},

                                  value: true,

                                  activeColor: **Colors**.white,

                                ),],

                            ),

**Text**(

                              alarmTime.**substring**(0,10),

                              style: **TextStyle**(

                                  color: **Colors**.white, fontFamily: 'avenir'),

                            ),

**Row**(

                              mainAxisAlignment: **MainAxisAlignment**.spaceBetween,

                              children: <**Widget**>[

**Text**(

                                  alarmTime.**substring**(11,),

                                  style: **TextStyle**(

                                      color: **Colors**.white,

                                      fontFamily: 'avenir',

                                      fontSize: 24,

                                      fontWeight: **FontWeight**.w700),

                                ),

**IconButton**(

                                    icon: **Icon**(**Icons**.delete),

                                    color: **Colors**.white,

                                    onPressed: () {

**deleteAlarm**(alarm.id);

                                    }),

                              ],),

                          ],),

                      );

                    }).**followedBy**([

                      if (\_currentAlarms.length < 5)

**DottedBorder**(

                          strokeWidth: 2,

                          color: **CustomColors**.clockOutline,

                          borderType: **BorderType**.**RRect**,

                          radius: **Radius**.**circular**(24),

                          dashPattern: [5, 4],

                          child: **Container**(

                            width: **double**.infinity,

                            decoration: **BoxDecoration**(

                              color: **CustomColors**.clockBG,

                              borderRadius:

**BorderRadius**.**all**(**Radius**.**circular**(24)),

                            ),

                            child: **FlatButton**(

                              padding: const **EdgeInsets**.**symmetric**(

                                  horizontal: 32, vertical: 16),

                              onPressed: () {

                                \_alarmTimeString =

**DateFormat**('HH:mm').**format**(**DateTime**.**now**());

**showModalBottomSheet**(

                                  useRootNavigator: true,

                                  context: context,

                                  clipBehavior: **Clip**.antiAlias,

                                  shape: **RoundedRectangleBorder**(

                                    borderRadius: **BorderRadius**.**vertical**(

                                      top: **Radius**.**circular**(24),

                                    ),

                                  ),

                                  builder: (context) {

                                    return **StatefulBuilder**(

                                      builder: (context, setModalState) {

                                        return **Container**(

                                          padding: const **EdgeInsets**.**all**(32),

                                          child: **Column**(

                                            children: [

**FlatButton**(

                                                onPressed: () async {

                                                   selectedYear=await

**showDatePicker**(context: context, initialDate:**DateTime**.**now**(), firstDate: **DateTime**.**now**(),

                                                  lastDate: **DateTime**(2030));

                                                  var selectedTime =

                                                      await **showTimePicker**(

                                                    context: context,

                                                    initialTime:

**TimeOfDay**.**now**(),

                                                  );

                                                  if (selectedTime != null) {

                                                    final now = **DateTime**.**now**();

                                                    var selectedDateTime =

**DateTime**(

                                                            selectedYear.year,

                                                            selectedYear.month,

                                                            selectedYear.day,

                                                            selectedTime.hour,

                                                            selectedTime

                                                                .minute);

                                                    \_alarmTime =

                                                        selectedDateTime;

**setModalState**(() {

                                                      \_alarmTimeString =

**DateFormat**('HH:mm')

                                                              .**format**(

                                                                  selectedDateTime);

                                                    });

                                                  }

                                                },

                                                child: **Text**(

                                                  \_alarmTimeString,

                                                  style:

**TextStyle**(fontSize: 32),

                                                ),),

**ListTile**(

                                                title: **Text**('Repeat'),

                                                trailing: **Icon**(

**Icons**.arrow\_forward\_ios),

                                              ),

**ListTile**(

                                                title: **Text**('Sound'),

                                                trailing: **Icon**(

**Icons**.arrow\_forward\_ios),

                                              ),

**ListTile**(

                                                title: **Text**('Title'),

                                                trailing: **Icon**(

**Icons**.arrow\_forward\_ios),

                                              ),

**FloatingActionButton**.**extended**(

                                                onPressed: onSaveAlarm,

                                                icon: **Icon**(**Icons**.alarm),

                                                label: **Text**('Save'),

                                              ),],

                                          ),   );

                                      },

                                    ); },

                                );

                              },

                              child: **Column**(

                                children: <**Widget**>[

**Image**.**asset**(

                                    'assets/add\_alarm.png',

                                    scale: 1.5,

                                  ),

**SizedBox**(height: 8),

**Text**(

                                    'Add Alarm',

                                    style: **TextStyle**(

                                        color: **Colors**.white,

                                        fontFamily: 'avenir'),

                                  ),

                                ],),

                            ),),

                        )

                      else

**Center**(

                            child: **Text**(

                          'Only 5 alarms allowed!',

                          style: **TextStyle**(color: **Colors**.white),

                        )),

                    ]).**toList**(),

                  );

                }

                return **Center**(

                  child: **Text**(

                    'Loading..',

                    style: **TextStyle**(color: **Colors**.white),

                  ),

                ); },

            ),

          ),

        ],),

    );

  }

  void **scheduleAlarm**(

**DateTime** scheduledNotificationDateTime, **AlarmInfo** alarmInfo) async {

    var androidPlatformChannelSpecifics = **AndroidNotificationDetails**(

      'alarm\_notif',

      'alarm\_notif',

      'Channel for Alarm notification',

      icon: 'codex\_logo',

      sound: **RawResourceAndroidNotificationSound**('a\_long\_cold\_sting'),

      largeIcon: **DrawableResourceAndroidBitmap**('codex\_logo'),

    );

    var iOSPlatformChannelSpecifics = **IOSNotificationDetails**(

        sound: 'a\_long\_cold\_sting.wav',

        presentAlert: true,

        presentBadge: true,

        presentSound: true);

    var platformChannelSpecifics = **NotificationDetails**(

        androidPlatformChannelSpecifics, iOSPlatformChannelSpecifics);

    await flutterLocalNotificationsPlugin.**schedule**(0, 'Office', alarmInfo.title,

        scheduledNotificationDateTime, platformChannelSpecifics);

  }

  void **onSaveAlarm**() {

**DateTime** scheduleAlarmDateTime;

    if (\_alarmTime.**isAfter**(**DateTime**.**now**()))

      scheduleAlarmDateTime = \_alarmTime;

    else

      scheduleAlarmDateTime = \_alarmTime.**add**(**Duration**(days: 1));

    var alarmInfo = **AlarmInfo**(

      alarmDateTime: scheduleAlarmDateTime,

      gradientColorIndex: \_currentAlarms.length,

      title: 'alarm',

    );

    \_alarmHelper.**insertAlarm**(alarmInfo);

**scheduleAlarm**(scheduleAlarmDateTime, alarmInfo);

**Navigator**.**pop**(context);

**loadAlarms**();

  }

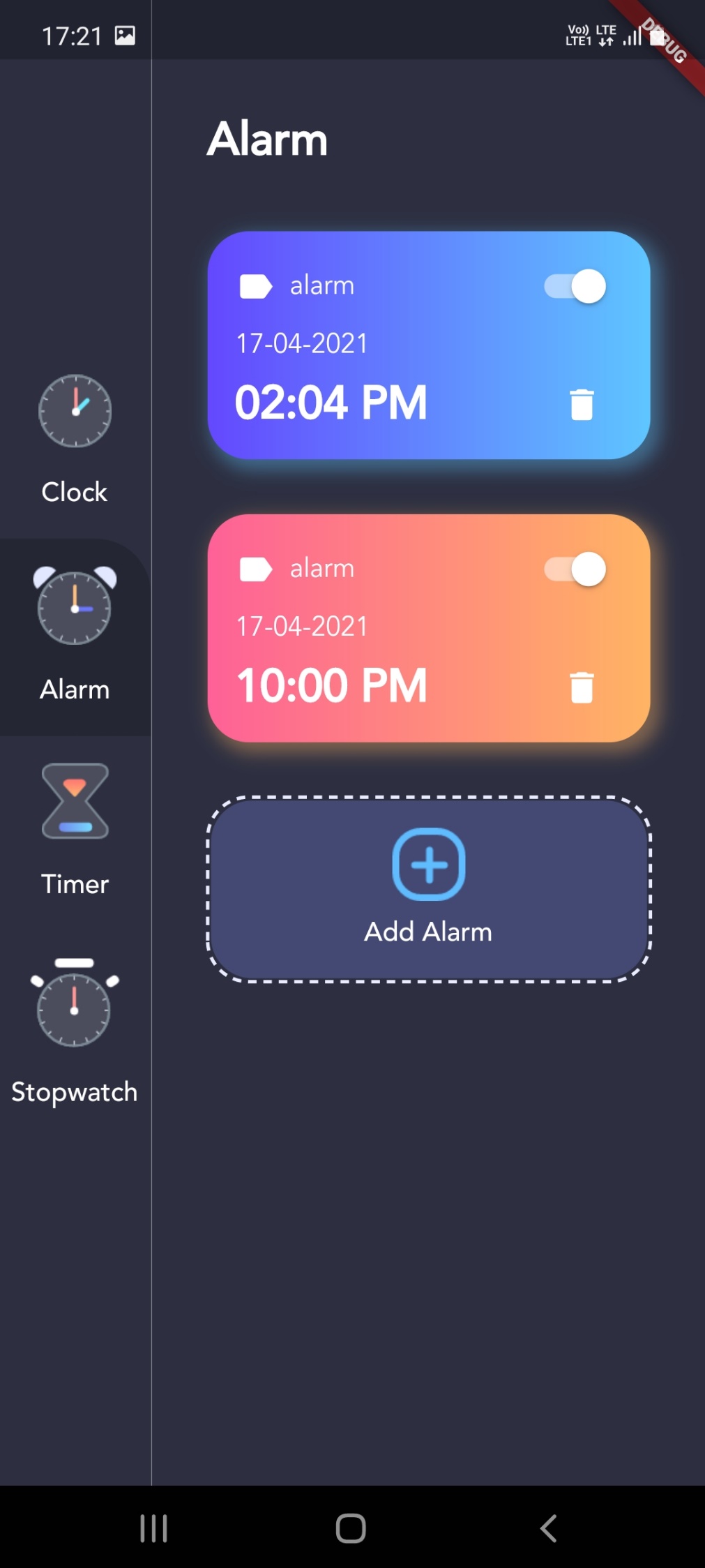
  void **deleteAlarm**(**int** id) {

    \_alarmHelper.**delete**(id);

**loadAlarms**();

  }

}



**10.13 ClockPage.dart**

import 'dart:async';

import 'package:kuva/constants/theme\_data.dart';

import 'package:flutter/material.dart';

import 'package:intl/intl.dart';

import 'clockview.dart';

class **ClockPage** extends **StatefulWidget** {

  @override

**\_ClockPageState** **createState**() => **\_ClockPageState**();

}

class **\_ClockPageState** extends **State**<**ClockPage**> {

  @override

**Widget** **build**(**BuildContext** context) {

    var now = **DateTime**.**now**();

    var formattedDate = **DateFormat**('EEE, d MMM').**format**(now);

    var timezoneString = now.timeZoneOffset.**toString**().**split**('.').first;

    var offsetSign = '';

    if (!timezoneString.**startsWith**('-')) offsetSign = '+';

    return **Container**(

      padding: **EdgeInsets**.**symmetric**(horizontal: 32, vertical: 64),

      child: **Column**(

        crossAxisAlignment: **CrossAxisAlignment**.start,

        children: <**Widget**>[

**Flexible**(

            flex: 1,

            fit: **FlexFit**.tight,

            child: **Text**(

              'Clock',

              style: **TextStyle**(

                  fontFamily: 'avenir',

                  fontWeight: **FontWeight**.w700,

                  color: **CustomColors**.primaryTextColor,

                  fontSize: 24),

            ),

          ),

**Flexible**(

            flex: 2,

            child: **Column**(

              crossAxisAlignment: **CrossAxisAlignment**.start,

              children: <**Widget**>[

**DigitalClockWidget**(),

**Text**(

                  formattedDate,

                  style: **TextStyle**(

                      fontFamily: 'avenir',

                      fontWeight: **FontWeight**.w300,

                      color: **CustomColors**.primaryTextColor,

                      fontSize: 20),

                ),

              ],

            ),

          ),

**Flexible**(

            flex: 4,

            fit: **FlexFit**.tight,

            child: **Align**(

              alignment: **Alignment**.center,

              child: **ClockView**(

                size: **MediaQuery**.**of**(context).size.height / 4,

              ),

            ),

          ),

**Flexible**(

            flex: 2,

            fit: **FlexFit**.tight,

            child: **Column**(

              crossAxisAlignment: **CrossAxisAlignment**.start,

              children: <**Widget**>[

**Text**(

                  'Timezone',

                  style: **TextStyle**(

                      fontFamily: 'avenir',

                      fontWeight: **FontWeight**.w500,

                      color: **CustomColors**.primaryTextColor,

                      fontSize: 24),

                ),

**SizedBox**(height: 16),

**Row**(

                  children: <**Widget**>[

**Icon**(

**Icons**.language,

                      color: **CustomColors**.primaryTextColor,

                    ),

**SizedBox**(width: 16),

**Text**(

                      'UTC' + offsetSign + timezoneString,

                      style: **TextStyle**(

                          fontFamily: 'avenir',

                          color: **CustomColors**.primaryTextColor,

                          fontSize: 14),

                    ),

                  ],

                ),

              ],

            ),

          ),

        ],

      ),

    );

  }

}

class **DigitalClockWidget** extends **StatefulWidget** {

  const **DigitalClockWidget**({

**Key** key,

  }) : super(key: key);

  @override

**State**<**StatefulWidget**> **createState**() {

    return **DigitalClockWidgetState**();

  }

}

class **DigitalClockWidgetState** extends **State**<**DigitalClockWidget**> {

  var formattedTime = **DateFormat**('HH:mm').**format**(**DateTime**.**now**());

  @override

  void **initState**() {

**Timer**.**periodic**(**Duration**(seconds: 1), (timer) {

      var perviousMinute = **DateTime**.**now**().**add**(**Duration**(seconds: -1)).minute;

      var currentMinute = **DateTime**.**now**().minute;

      if (perviousMinute != currentMinute)

**setState**(() {

          formattedTime = **DateFormat**('HH:mm').**format**(**DateTime**.**now**());

        });

    });

    super.**initState**();

  }

  @override

**Widget** **build**(**BuildContext** context) {

**print**('=====>digital clock updated');

    return **Text**(

      formattedTime,

      style: **TextStyle**(

          fontFamily: 'avenir',

          color: **CustomColors**.primaryTextColor,

          fontSize: 64),

    );

  }

}



**10.14 ClockView.dart**

import 'dart:async';

import 'dart:math';

import 'package:kuva/constants/theme\_data.dart';

import 'package:flutter/material.dart';

class **ClockView** extends **StatefulWidget** {

  final **double** size;

  const **ClockView**({**Key** key, this.size}) : super(key: key);

  @override

**\_ClockViewState** **createState**() => **\_ClockViewState**();

}

class **\_ClockViewState** extends **State**<**ClockView**> {

  @override

  void **initState**() {

**Timer**.**periodic**(**Duration**(seconds: 1), (timer) {

**setState**(() {});

    });

    super.**initState**();

  }

  @override

**Widget** **build**(**BuildContext** context) {

    return **Align**(

      alignment: **Alignment**.topCenter,

      child: **Container**(

        width: widget.size,

        height: widget.size,

        child: **Transform**.**rotate**(

          angle: -pi / 2,

          child: **CustomPaint**(

            painter: **ClockPainter**(),

          ),

        ),

      ),

    );

  }

}

class **ClockPainter** extends **CustomPainter** {

  var dateTime = **DateTime**.**now**();

  @override

  void **paint**(**Canvas** canvas, **Size** size) {

    var centerX = size.width / 2;

    var centerY = size.height / 2;

    var center = **Offset**(centerX, centerY);

    var radius = **min**(centerX, centerY);

    var fillBrush = **Paint**()..color = **CustomColors**.clockBG;

    var outlineBrush = **Paint**()

      ..color = **CustomColors**.clockOutline

      ..style = **PaintingStyle**.stroke

      ..strokeWidth = size.width / 20;

    var centerDotBrush = **Paint**()..color = **CustomColors**.clockOutline;

    var secHandBrush = **Paint**()

      ..color = **CustomColors**.secHandColor

      ..style = **PaintingStyle**.stroke

      ..strokeCap = **StrokeCap**.round

      ..strokeWidth = size.width / 60;

    var minHandBrush = **Paint**()

      ..shader = **RadialGradient**(colors: [

**CustomColors**.minHandStatColor,

**CustomColors**.minHandEndColor

      ]).**createShader**(**Rect**.**fromCircle**(center: center, radius: radius))

      ..style = **PaintingStyle**.stroke

      ..strokeCap = **StrokeCap**.round

      ..strokeWidth = size.width / 30;

    var hourHandBrush = **Paint**()

      ..shader = **RadialGradient**(colors: [

**CustomColors**.hourHandStatColor,

**CustomColors**.hourHandEndColor

      ]).**createShader**(**Rect**.**fromCircle**(center: center, radius: radius))

      ..style = **PaintingStyle**.stroke

      ..strokeCap = **StrokeCap**.round

      ..strokeWidth = size.width / 24;

    var dashBrush = **Paint**()

      ..color = **CustomColors**.clockOutline

      ..style = **PaintingStyle**.stroke

      ..strokeWidth = 1;

    canvas.**drawCircle**(center, radius \* 0.75, fillBrush);

    canvas.**drawCircle**(center, radius \* 0.75, outlineBrush);

    var hourHandX = centerX +

        radius \*

            0.4 \*

**cos**((dateTime.hour \* 30 + dateTime.minute \* 0.5) \* pi / 180);

    var hourHandY = centerY +

        radius \*

            0.4 \*

**sin**((dateTime.hour \* 30 + dateTime.minute \* 0.5) \* pi / 180);

    canvas.**drawLine**(center, **Offset**(hourHandX, hourHandY), hourHandBrush);

    var minHandX = centerX + radius \* 0.6 \* **cos**(dateTime.minute \* 6 \* pi / 180);

    var minHandY = centerY + radius \* 0.6 \* **sin**(dateTime.minute \* 6 \* pi / 180);

    canvas.**drawLine**(center, **Offset**(minHandX, minHandY), minHandBrush);

    var secHandX = centerX + radius \* 0.6 \* **cos**(dateTime.second \* 6 \* pi / 180);

    var secHandY = centerY + radius \* 0.6 \* **sin**(dateTime.second \* 6 \* pi / 180);

    canvas.**drawLine**(center, **Offset**(secHandX, secHandY), secHandBrush);

    canvas.**drawCircle**(center, radius \* 0.12, centerDotBrush);

    var outerRadius = radius;

    var innerRadius = radius \* 0.9;

    for (var i = 0; i < 360; i += 12) {

      var x1 = centerX + outerRadius \* **cos**(i \* pi / 180);

      var y1 = centerY + outerRadius \* **sin**(i \* pi / 180);

      var x2 = centerX + innerRadius \* **cos**(i \* pi / 180);

      var y2 = centerY + innerRadius \* **sin**(i \* pi / 180);

      canvas.**drawLine**(**Offset**(x1, y1), **Offset**(x2, y2), dashBrush);

    }

  }

  @override

**bool** **shouldRepaint**(**CustomPainter** oldDelegate) {

    return true;

  }

}

**10.15 AlarmHelper.dart**

import 'package:kuva/models/alarm\_info.dart';

import 'package:sqflite/sqflite.dart';

import 'package:sqflite/sqlite\_api.dart';

final **String** tableAlarm = 'alarm';

final **String** columnId = 'id';

final **String** columnTitle = 'title';

final **String** columnDateTime = 'alarmDateTime';

final **String** columnPending = 'isPending';

final **String** columnColorIndex = 'gradientColorIndex';

class **AlarmHelper** {

  static **Database** \_database;

  static **AlarmHelper** \_alarmHelper;

**AlarmHelper**.**\_createInstance**();

  factory **AlarmHelper**() {

    if (\_alarmHelper == null) {

      \_alarmHelper = **AlarmHelper**.**\_createInstance**();

    }

    return \_alarmHelper;

  }

**Future**<**Database**> get database async {

    if (\_database == null) {

      \_database = await **initializeDatabase**();

    }

    return \_database;

  }

**Future**<**Database**> **initializeDatabase**() async {

    var dir = await **getDatabasesPath**();

    var path = dir + "alarm.db";

    var database = await **openDatabase**(

      path,

      version: 1,

      onCreate: (db, version) {

        db.**execute**('''

          create table $tableAlarm (

          $columnId integer primary key autoincrement,

          $columnTitle text not null,

          $columnDateTime text not null,

          $columnPending integer,

          $columnColorIndex integer)

        ''');

      },

    );

    return database;

  }

  void **insertAlarm**(**AlarmInfo** alarmInfo) async {

    var db = await this.database;

    var result = await db.**insert**(tableAlarm, alarmInfo.**toMap**());

**print**('result : $result');

  }

**Future**<**List**<**AlarmInfo**>> **getAlarms**() async {

**List**<**AlarmInfo**> \_alarms = [];

    var db = await this.database;

    var result = await db.**query**(tableAlarm);

    result.**forEach**((element) {

      var alarmInfo = **AlarmInfo**.**fromMap**(element);

      \_alarms.**add**(alarmInfo);

    });

    return \_alarms;

  }

**Future**<**int**> **delete**(**int** id) async {

    var db = await this.database;

    return await db.**delete**(tableAlarm, where: '$columnId = ?', whereArgs: [id]);

  }

}

**10.16 Data.dart**

import 'package:kuva/constants/theme\_data.dart';

import 'package:kuva/enums.dart';

import 'models/alarm\_info.dart';

import 'models/menu\_info.dart';

**List**<**MenuInfo**> menuItems = [

**MenuInfo**(**MenuType**.clock,

      title: 'Clock', imageSource: 'assets/clock\_icon.png'),

**MenuInfo**(**MenuType**.alarm,

      title: 'Alarm', imageSource: 'assets/alarm\_icon.png'),

**MenuInfo**(**MenuType**.timer,

      title: 'Timer', imageSource: 'assets/timer\_icon.png'),

**MenuInfo**(**MenuType**.stopwatch,

      title: 'Stopwatch', imageSource: 'assets/stopwatch\_icon.png'),

];

**List**<**AlarmInfo**> alarms = [

**AlarmInfo**(

      alarmDateTime: **DateTime**.**now**().**add**(**Duration**(hours: 1)),

      title: 'Office',

      gradientColorIndex: 0),

**AlarmInfo**(

      alarmDateTime: **DateTime**.**now**().**add**(**Duration**(hours: 2)),

      title: 'Sport',

      gradientColorIndex: 1),

];

**10.17 Enums.dart**

enum **MenuType** { clock, alarm, timer, stopwatch }

**10.17 Task.dart**

import 'package:flutter/material.dart';

class **Task** {

  final **String** name;

  final **String** category;

  final **String** time;

  final **Color** color;

  final **bool** completed;

**Task**({this.name, this.category, this.time, this.color, this.completed});

}

**10.18 TaskRow.dart**

import 'task.dart';

import 'package:flutter/material.dart';

class **TaskRow** extends **StatefulWidget** {

  final **List** task;

  final **Animation**<**double**> animation;

  const **TaskRow**({**Key** key, this.task, this.animation}) : super(key: key);

  @override

**\_TaskRowState** **createState**() => **\_TaskRowState**();

}

class **\_TaskRowState** extends **State**<**TaskRow**> {

  final **double** dotSize = 12.0;

  var count=1;

  @override

**Widget** **build**(**BuildContext** context) {

**print**('Task');

**print**(widget.task);

    return new **FadeTransition**(

      opacity: widget.animation,

      child: new **SizeTransition**(

        sizeFactor: widget.animation,

        child: new **Padding**(

          padding: const **EdgeInsets**.**symmetric**(vertical: 16.0),

          child: new **Row**(

            children: <**Widget**>[

              new **Padding**(

                padding:

                    new **EdgeInsets**.**symmetric**(horizontal: 32.0 - dotSize / 2),

                child: new **Container**(

                  height: dotSize,

                  width: dotSize,

                  decoration: new **BoxDecoration**(

                      shape: **BoxShape**.circle),

                ),

              ),

              new **Expanded**(

                child: new **Column**(

                  crossAxisAlignment: **CrossAxisAlignment**.start,

                  children: <**Widget**>[

**ListView**.**builder**(

                      shrinkWrap: true,

                       itemBuilder: (ctx, index) =>

**Text**(

                        count==1?widget.task[index]:count=count+1,

                        style: new **TextStyle**(fontSize: 18.0),

                      ),

                      itemCount: widget.task.length

                    ),],

                ),

              ),],

          ),

        ),

      ),

    );

  }

}

**10.19 Alarm\_info.dart**

class **AlarmInfo** {

**int** id;

**String** title;

**DateTime** alarmDateTime;

**bool** isPending;

**int** gradientColorIndex;

**AlarmInfo**(

      {this.id,

      this.title,

      this.alarmDateTime,

      this.isPending,

      this.gradientColorIndex});

  factory **AlarmInfo**.**fromMap**(**Map**<**String**, **dynamic**> json) => **AlarmInfo**(

        id: json["id"],

        title: json["title"],

        alarmDateTime: **DateTime**.**parse**(json["alarmDateTime"]),

        isPending: json["isPending"],

        gradientColorIndex: json["gradientColorIndex"],

      );

**Map**<**String**, **dynamic**> **toMap**() => {

        "id": id,

        "title": title,

        "alarmDateTime": alarmDateTime.**toIso8601String**(),

        "isPending": isPending,

        "gradientColorIndex": gradientColorIndex,

      };

}

**10.20 ThemeData.dart**

import 'package:flutter/material.dart';

class **CustomColors** {

  static **Color** primaryTextColor = **Colors**.white;

  static **Color** dividerColor = **Colors**.white54;

  static **Color** pageBackgroundColor = **Color**(0xFF2D2F41);

  static **Color** menuBackgroundColor = **Color**(0xFF242634);

  static **Color** clockBG = **Color**(0xFF444974);

  static **Color** clockOutline = **Color**(0xFFEAECFF);

  static **Color** secHandColor = **Colors**.orange[300];

  static **Color** minHandStatColor = **Color**(0xFF748EF6);

  static **Color** minHandEndColor = **Color**(0xFF77DDFF);

  static **Color** hourHandStatColor = **Color**(0xFFC279FB);

  static **Color** hourHandEndColor = **Color**(0xFFEA74AB);

}

class **GradientColors** {

  final **List**<**Color**> colors;

**GradientColors**(this.colors);

  static **List**<**Color**> sky = [**Color**(0xFF6448FE), **Color**(0xFF5FC6FF)];

  static **List**<**Color**> sunset = [**Color**(0xFFFE6197), **Color**(0xFFFFB463)];

  static **List**<**Color**> sea = [**Color**(0xFF61A3FE), **Color**(0xFF63FFD5)];

  static **List**<**Color**> mango = [**Color**(0xFFFFA738), **Color**(0xFFFFE130)];

  static **List**<**Color**> fire = [**Color**(0xFFFF5DCD), **Color**(0xFFFF8484)];

}

class **GradientTemplate** {

  static **List**<**GradientColors**> gradientTemplate = [

**GradientColors**(**GradientColors**.sky),

**GradientColors**(**GradientColors**.sunset),

**GradientColors**(**GradientColors**.sea),

**GradientColors**(**GradientColors**.mango),

**GradientColors**(**GradientColors**.fire),

  ];

}

**10.21 Affirmations.dart**

import 'package:flutter/foundation.dart';

import 'package:flutter/src/widgets/framework.dart';

class **Affirmations**

{

  final **String** id;

  final **String** title;

  final **List**<**String**> categories;

  final **String** imageUrl;

  final **List**<**String**> audioName;

  final **List**<**String**> affirmations;

**bool** akValue;

**Affirmations**({@required this.id,@required this.title,@required this.categories,@required this.imageUrl,

  @required this.affirmations,@required this.audioName,@required this.akValue });

}

**10.22 Categories.dart**

import 'package:flutter/material.dart';

class **Category**

{

  final **String** id;

  final **String** title;

  final **Color** color;

  final **Icon** iconname;

const **Category**({

  this.id,

    this.title,

    this.color=**Colors**.orange,

    this.iconname});

}

**10.23 Categories.dart**

import 'package:flutter/material.dart';

import './models/categories.dart';

import './models/affirmations.dart';

 var **DUMMY\_CATEGORIES** = [

**Category**(

    id: 'c1',

    title: 'Health',

    color: **Colors**.purple,

    iconname:**Icon**(**Icons**.healing),

  ),

**Category**(

    id: 'c2',

    title: 'Wealth',

    color: **Colors**.red,

    iconname:**Icon**(**Icons**.money\_off),

  ),

**Category**(

    id: 'c3',

    title: 'Sports',

    color: **Colors**.orange,

     iconname:**Icon**(**Icons**.play\_arrow),

  ),

**Category**(

    id: 'c4',

    title: 'Family',

    color: **Colors**.amber,

    iconname:**Icon**(**Icons**.people),

  ),

**Category**(

    id: 'c5',

    title: 'Love',

    color: **Colors**.blue,

    iconname:**Icon**(**Icons**.favorite),

  ),

**Category**(

    id: 'c6',

    title: 'Relation',

    color: **Colors**.green,

    iconname:**Icon**(**Icons**.people),

  ),

**Category**(

    id: 'c7',

    title: 'Studies',

    color: **Colors**.lightBlue,

    iconname:**Icon**(**Icons**.book),

  ),

**Category**(

    id: 'c8',

    title: 'Exams',

    color: **Colors**.lightGreen,

    iconname:**Icon**(**Icons**.book),

  ),

**Category**(

    id: 'c9',

    title: 'Business',

    color: **Colors**.pink,

    iconname:**Icon**(**Icons**.money\_off),

  ),

**Category**(

    id: 'c10',

    title: 'Motivate',

    color: **Colors**.teal,

    iconname:**Icon**(**Icons**.book),

  ),

];

 var **DUMMY\_Affirmations** =  [

**Affirmations**(

    id: 'm1',

    categories: [

      'c1',

    ],

    title: 'Health',

    imageUrl:

        'https://www.healthcareitnews.com/sites/hitn/files/Global%20healthcare\_2.jpg',

    affirmations: [

      'Health 1',

      'Health 2',

      'Health 3',

      'Health 4',

      'Health 5',

      'Health 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

    akValue: false,

  ),

**Affirmations**(

    id: 'm2',

    categories: [

      'c2',

    ],

    title: 'Wealth',

    imageUrl:

        'https://images.pexels.com/photos/47047/gold-ingots-golden-treasure-47047.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500',

    affirmations: [

       'I am a magnet of success',

        'I am wealthy and prosperous',

      'I am smart',

      'I have the skills',

     'I keep the money I make',

      'I deserve everthing I want'

    ],

    audioName: ['assets/audio3.mp3','assets/audio4.mp3','assets/audio3.mp3','assets/audio4.mp3','assets/audio5.mp3','assets/audio6.mp3'],

   akValue: false,

  ),

**Affirmations**(

    id: 'm3',

    categories: [

      'c3',

    ],

    title: 'Sports',

    imageUrl:

        'https://biew.ac.in/wp-content/uploads/2020/02/biew-sports.jpg',

    affirmations: [

      'Sports 1',

      'Sports 2',

      'Sports 3',

      'Sports 4',

      'Sports 5',

      'Sports 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

akValue: false,

  ),

**Affirmations**(

    id: 'm4',

    categories: [

      'c4',

    ],

    title: 'Family',

    imageUrl:

        'https://i.pinimg.com/originals/1c/af/b4/1cafb40f8a056b0c0bb3ca61e38a98df.jpg',

    affirmations: [

      'Family 1',

      'Family 2',

      'Family 3',

      'Family 4',

      'Family 5',

      'Family 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

    akValue: false,

  ),

**Affirmations**(

    id: 'm5',

    categories: [

          'c5',

    ],

    title: 'Love',

    imageUrl:

        'https://images.firstpost.com/wp-content/uploads/2020/01/love-3061483\_1280.jpg',

    affirmations: [

      'Love 1',

      'Love 2',

      'Love 3',

      'Love 4',

      'Love 5',

      'Love 6'

    ],

   audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

   akValue: false,

  ),

**Affirmations**(

    id: 'm6',

    categories: [

      'c6',

    ],

    title: 'Relationship',

    imageUrl:

        'https://pyxis.nymag.com/v1/imgs/750/5a5/fd3e1a46d6bc8cca856682335b413059f5-23-relationship-anarchy.rsquare.w1200.jpg',

    affirmations: [

      'Relationship 1',

      'Relationship 2',

      'Relationship 3',

      'Relationship 4',

      'Relationship 5',

      'Relationship 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

 akValue: false,

  ),

**Affirmations**(

    id: 'm7',

    categories: [

      'c7',

    ],

    title: 'Studies',

    imageUrl:

        'https://static.toiimg.com/thumb/msid-69524865,width-1070,height-580,imgsize-1114562,resizemode-75,overlay-toi\_sw,pt-32,y\_pad-40/photo.jpg',

    affirmations: [

      'Studies 1',

      'Studies 2',

      'Studies 3',

      'Studies 4',

      'Studies 5',

      'Studies 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

    akValue: false,

  ),

**Affirmations**(

    id: 'm8',

    categories: [

      'c8',

    ],

    title: 'exams',

    imageUrl:

        'https://www.stoodnt.com/blog/wp-content/uploads/2020/01/exam-1575366183.jpg',

    affirmations: [

      'exams 1',

      'exams 2',

      'exams 3',

      'exams 4',

      'exams 5',

      'exams 6',

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

 akValue: false,

  ),

**Affirmations**(

    id: 'm9',

    categories: [

      'c9',

    ],

    title: 'Business',

    imageUrl:

        'https://thumbor.forbes.com/thumbor/960x0/https%3A%2F%2Fblogs-images.forbes.com%2Falejandrocremades%2Ffiles%2F2018%2F07%2Fdesk-3139127\_1920-1200x773.jpg',

    affirmations: [

      'Business 1',

      'Business 2',

      'Business 3',

      'Business 4',

      'Business 5',

      'Business 6'

    ],

    audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

    akValue: false,

  ),

**Affirmations**(

    id: 'm10',

    categories: [

      'c10',

    ],

    title: 'Motivation',

    imageUrl:

        'https://1.bp.blogspot.com/-n0cHyU75eeY/WaPqvSdfSUI/AAAAAAAAbxw/4yF0RGMFceo70b3J9\_72ehhYuQp-n4MyACLcBGAs/s1600/Self%2Bmotivation%2Bis%2Bkey.jpg',

    affirmations: [

      'Motivation 1',

      'Motivation 2',

      'Motivation 3',

      'Motivation 4',

      'Motivation 5',

      'Motivation 6'

    ],

   audioName: ['assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3','assets/audio.mp3'],

    akValue: false,

  ),

];

# 11.Limitations

• This App is subscription based (7 days trial period)

• The user can not have the touch/feel experience.

• Cash back features are not so well defined.

• Affirmations are not analysed and segregated according to good, bad or satisfactory.

• Requires Third party authentication for payments (Payment Gateways).

Graphical user interface, diagram

Description automatically generated with medium confidence

# 12. Applications

• Patients suffering from mental illness

• Frontline Workers

• Athletes

• Students

• Working professionals

• People who are confined to their homes

# Future Scope

• Security and privacy issues can be handled.

• Expert consultancy through app.

• Coupon codes on the subscriptions.

• Online Wallet/Varied Advanced Payment Options.

# Conclusion

India is a developing country which is advancing towards Digitization due to which the traditional market is migrating towards E-commerce. Having to drive long distances and take significant time out of a busy schedule to seek therapy can be a burden on many people in need of help. The main objective of making this project is to provide an end user with relatively quick and easy access to treatment. It allows them to plan everything according to their requirements sitting at home, in comfort, without travelling to various locations and choosing affirmations according to their will and time.

It is concluded that this app provides them all the requirements and essential necessities which are used to provide a user with affordable, accessible, convenient and effective therapy to stay high on morale and be able to carry out his tasks productively!

# References

1. Discussions carried on with Deep Upadhyay and Devdutta Puntambekar – Senior Developers at Sattvarise Technologies.
2. Novel named ‘The Power of Positive Thinking’ by Norman Vincent Peale (Father of Positive Thinking)
3. Mental Health Study conducted by The Economic Times of India.
4. Udemy Course by Maximilan Schwarzmuller on ‘Flutter & Dart – The Complete Guide’.
5. https://flutter.dev/docs for official Flutter Documentation.
6. https://dart.dev/guides for knowledge on how to write effective programs in Dart Language.
7. https://pub.dev/ for getting more knowledge on Flutter packages.
8. https://github.com/razorpay/razorpay-flutter for getting more information on Integration of Razorpay Payment Gateway with Flutter Modules.
9. https://positivepsychology.com/daily-affirmations/ and https://www.mindtools.com/pages/article/affirmations.htm on knowing the science behind self-affirmations and how to use them effectively.
10. https://stackoverflow.com/ on getting solved various queries that were raised while developing the App.