

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

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

By

Name: JESTIN JOSE

(Reg .no. 39120047)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

AUGUST 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTEMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Jestin Jose(39120047)** who carried out the project entitled “**Cooking Recipe App**” under my supervision from October 2021 to November 2021.

INTERNAL GUIDE:

Name: Dr.Senduru. Srinivasulu M.E., Ph.D.,

HEAD OF DEPARTMENT

Dr.R.Subhashini M.E., Ph.D.,

Submitted for Viva Voice Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **JESTIN JOSE** hereby declare that the Project Report entitled '**Cooking Recipe App**' done by me under the guidance of **Dr.Senduru. Srinivasulu M.E., Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE:CHENNAI



SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. R.SUBHASHINI M.E., Ph.D.**, Heads of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

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

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

TRAINING CERTIFICATE



ABSTRACT

Our generation relies mostly on phones to get through the day. Due to this, phones have become more of a personal assistant than a means to just communicate. Keeping this thing in mind i wanted to use the concept of a phone as a personal assistant for helping the users to cook like a master even when they don't know a thing about cooking. The system is an Flutter Application- CookingRecipe, which provides the users with simplified recipes with real-time search which are easy to understand and easier to make and can be accessed even when the user is offline with the assistance of something which they carry with them all the time and hence can access it at anyplace and anytime. The system has a structured UI unlike most of the cooking apps that are available. It provides users recipes along with their preparation time, number of servings, and ingredients that will be required which can be added to save them later. Users can even find the things for the ingredients as per their requirement. The proposed system is a proper source for any type of cook-a beginner or a professional, to find the recipe of the food that they want to make. I created various pages to split our code to make sure it looks neat. Working with this app was a huge breakthrough in my career and I had learned a lot of things. This report takes us through all the details of the Food Recipe app which I had developed.

TABLE OF CONTENTS:

Chapter No.	Title.
1.	INTRODUCTION 1.1. INTRODUCTION TO FLUTTER 1.2. HISTORY OF FLUTTER 1.3. INTRODUCTION TO DART
2.	AIM AND SCOPE 2.1. AIM 2.2. SCOPE 2.3. HOW THE APP WORKS 2.4. FIREBASE
3.	SYSTEM SPECIFICATION 3.1. SOFTWARE AND HARDWARE REQUIREMENTS 3.2 WORK ENVIRONMENT
4.	SYSTEM STUDY AND ANALYSIS 4.1 ANALYSIS 4.2 PROBLEM 4.3 SOLUTION 4.4 RESULT

CHAPTER 1

1.INTRODUCTION

1.1. Introduction to Flutter:

Flutter is an open-source UI Software Development Kit created by Google. It is used to develop cross platform applications for Android, IOS, Linux, MAC, Windows, Google Fuchsia and the web from a single codebase. First Pitched in 2015, Flutter was released in May 2017.

Flutter apps are written in the Dart language and make use of many of the language's more advanced features. On Windows, macOS, and Linux. Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just in Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with ahead-of-time (AOT) Compilation

Flutter's engine, written primarily in C++. provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDK's such as those provided by Android and IOS.

The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and *Cupertino* widgets implement Apple's iOS Human Interface guidelines.

1.2. History of Flutter:

The first version of Flutter was called 'Sky' and ran on Android Operating System. It was unveiled at the 2015 Dart developer summit with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai in September 2018.

On May 6, 2020, the Dart software development kit (SDK) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

1.3 INTRODUCTION TO DART

Dart was unveiled at the GOTO conference in Aarhus, Denmark, October 10–12, 2012. The project was founded by Lars Bak and Kasper Lund. Dart 1.0 was released on November 14, 2013

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications.

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference.

Dart 2.6 introduced a new extension, dart2native. The feature extends native compilation to

the Linux, macOS, and Windows desktop platforms. Earlier developers were able to create new tools only using Android or iOS devices. Moreover, with this extension it becomes possible to compose a Dart program into self-contained executables. Thus, according to the company representatives, it's not obligatory now to have Dart SDK installed, the self-contained executables can now start running in a few seconds. The new extension is also integrated with Flutter toolkit, thus making it possible to use the compiler on small services (backend supporting for example)

Use of Dart :

There are four ways to run Dart code:

Web

To run in mainstream web browsers, Dart relies on a source-to-source compiler to JavaScript. According to the project site, Dart was "designed to be easy to write development tools for, well-suited to modern app development, and capable of high-performance implementations. When running Dart code in a web browser the code is precompiled into JavaScript using the dart2js compiler. Compiled as JavaScript, Dart code is compatible with all major browsers with no need for browsers to adopt Dart. Through optimizing the compiled JavaScript output to avoid expensive checks and operations, code written in Dart can, in some cases, run faster than equivalent code hand-written using JavaScript idioms.

Stand-alone

The Dart software development kit (SDK) ships with a stand-alone Dart VM, allowing Dart code to run in a command-line interface environment. As the language tools included in the Dart SDK are written mostly in Dart, the stand-alone Dart VM is a critical part of the SDK. These tools include the dart2js compiler and a package manager called pub. Dart ships with a complete standard library allowing users to write fully working system apps, such as custom web servers

Ahead-of-time compiled

Dart code can be AOT-compiled into machine code (native instruction sets). Apps built with Flutter, a mobile app SDK built with Dart, are deployed to app stores as AOT-compiled Dart code.

Native

Dart 2.6 with dart2native compiler to compile to self-contained, native executables code. Before Dart 2.6, this feature only exposed this capability on iOS and Android mobile devices via Flutter.

CHAPTER 2

AIM AND SCOPE

- **AIM**

Food is important for physical and mental well-being. It is more than just a primary factor for survival for some, for others, it is a major factor which can change a grumpy mood into a frivolous one. After a long day of work, or a hectic schedule, people want to come home, eat food, and relax. Surely takeouts are easy and delicious, but home-cooked meals have their own flavor and satisfaction. Finding a proper source of learning to cook for a beginner is difficult. That's why I have come up with the idea of a cooking recipe app which provides users with step-by-step recipes to make the task of preparing a meal less hectic.

- **Scope**

Greater Market Reach: With cross-platform applications, you can cover the broad user base available on the Apple App Store and Google Play Store. You can develop an application that can run on both iPhone and Android, giving you the advantage of having access to a vast pool of potential users.

Reduce App Development Cost: By leveraging a single codebase, developers can develop an application for both native platforms without having to engage two dedicated development teams.

Codes are Cheaper to Maintain: Since cross-platform applications are developed with a single code base, therefore, businesses from across the world having knowledge of working with cross-platform frameworks. By making changes and updates in one codebase, you can easily maintain the app. The final changes will be automatically reflected in all the applications without having to devote extra efforts and time on maintaining the codes.

App Testing is Easier: Instead of running different applications on different platforms, developers can discover errors by merely running a single app on multiple platforms. They won't need to devote long hours on testing each mobile app.

Availability of Better Plugins: Integration of third-party plugins increase the functionality of the application and speed up the process of mobile app development. And all the leading cross-platform app development frameworks like Flutter, Cordova, React Native, Ionic are open-source technologies that provide a great source of plugins. Developers can choose from the various widgets and can meet the demand of enterprises by customizing the app leveraging useful plugin libraries.

- **HOW THE APP WORKS**

The app houses a large collection of recipes integrated within the app itself, which the user can access anytime. The recipes are available completely offline and for free. The user can easily navigate around the app and find any relevant recipes. Moreover, the users can also make use of the search bar (smart search filter) to quickly find the desired recipes. With an integrated shopping list feature, it makes it easier for the users to make a note of any ingredients that they plan to cook. Along with the Like feature, the app also has an integrated Book mark feature, which helps users to find the ingredient

- **Firestore**

I used Firestore for backend. Firestore is a Google-backed application development software that enables developers to develop iOS, Android, and Web apps. Firestore provides tools for tracking analytics, reporting, and fixing app crashes, creating marketing and product experiment.

Services offered by Firestore

- **Analytics** – Google Analytics for Firestore offers free, unlimited reporting on as many as 500 separate events. Analytics presents data about user behavior in iOS and Android apps, enabling better decision-making about improving performance and app marketing.
- **Authentication** – Firestore Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- **Cloud messaging** – Firestore Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android, and the web at no cost.

One of the main reason I chose Firestore for my application is that it provides detailed documentation and cross-platform SDKs to help you build and ship apps on Android, iOS, the web, C++, and Unity and Firestore is used by various other applications such as INSTACART, CRED, ACCENTURE etc.

CHAPTER 3

SYSTEM SPECIFICATION AND METHODS

Software and Hardware Requirements:

- 8 Gb Ram or more
- 1280 x 800 minimum screen resolution
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
- Flutter, Java and Dart required
- A virtual device must be installed to run the application(Optional)

Work Environment

Font End Developer: The front-end developers generally work at the client side designing the User Interface of the application and add the requirements.

Back-End Developer: The back-end developer is a person who is responsible for the back-end development that interacts with the server. They oversee Firebase applications.



CHAPTER 4

SYSTEM STUDY AND ANALYSIS

Analysis

Problems and Solution

There are more problems in developing an app. I had faced a lot of errors while designing the UI also while developing the backend but those problems could be rectified easily

Problems

Today Do It Yourself concept has become very popular in the world. Urban Indian consumers are wildly drawn to the idea of cooking meals from the comfort of their homes. Surely there are many good recipe apps which provide thousands of recipes but a good application lacks a small feature that prevents it from making it a great application. During the process even a single semicolon had made me to find what was the error for almost 2 hours.

On the contrary, there are many challenges faced by mobile-app developers. One such thing is making the app load faster. During the initial phase, the app could face a lot of bugs or could load slowly or could crash. These kinds of things would irritate the client and the app could fail. One of the toughest challenges faced by app designers is making sure the app works perfectly fine and doesn't crash. Sometimes, the app may work on one mobile and wouldn't work in another one. App developers should make sure these kinds of things doesn't happen, sometimes these issues will develop because of the structure of the mobile or due to a software. Time spent detecting the source of a problem means a delay in tackling the problem itself.

Solution:

One of the main advantages while developing an app is you can easily find solution. I learned a

lot of new things while developing this application. Developing an app was challenging as well as exciting. There are lot of websites and mentors who guided me throughout this course.

Automated website monitoring solutions often allow the developers to view their clients' websites in real time and to set alerts to notify them when potential issues begin to arise. Not only does this allow the developers to identify a problem before the customer does, in many cases it also gives them the chance to fix it before the issue has any impact on the client's business.

Results

Learning Outcome:

- **Flutter Concepts**
- **Object Oriented Programming**
- **Data Structures**
- **Firebase**
- **Securing App with Authentication**

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Conclusion:

As a Conclusion, I can say the internship was a great experience. Thanks to this project, I acquired deeper knowledge concerning my technical skills, but I also personally benefited. Currently, Flutter is used as a major application while developing mobile apps and it has become a popular language worldwide. If we surf the internet, we can find number of apps developed using Flutter. I learn to live in a different environment from the one I am used to. Indeed, I grew more independent in work and in everyday life. I realized that I could do more things than I thought like learning new things by myself.

Cooking Recipe – is a completely free and easy to use mobile application. It provides users with tons of recipes to browse through and with ease of quick search options. It also has book mark and like feature integrated within the app itself to add to the user's convenience.

There are huge opportunities available for the students who want to work in this field. Many private and public organizations hire Flutter developers for designing mobile apps. With the rapid advent of online industry, the demand of Flutter developers is increasing, and this has created a huge job opportunity for the aspirants in the upcoming days. Also, an experienced person in this field can also work as a freelancer; there are many online companies which provide online projects to the individuals.

Future Scopes:

- Can work as a Flutter developer
- Can work as a Software Engineer
- Can work as a freelance mobile app's developer

REFERENCES

- [1] www.Google.com
- [2] youtube – Santos Enoque
- [3] youtube – Marcus Ng
- [4] youtube – Coading with Imran
- [5] <https://console.firebase.google.com>
- [6] <https://developers.googleblog.com/2019/05/Flutter-io19.html>
- [7] “Modeling of Cooking Recipes” Xiaoxi Dong McGill University, Montreal Qc Hxx xxx, Canada
- [8] “Foodie: An Android Application Rahul Dandekar, Sanket Gadkari, Mayuri Sadudia, Department of Information Technology Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering, Mumbai”

APPENDIX

Main.dart :

```
import 'package:flutter/material.dart';
```

```
/*import 'package:flutter_recipe_app/Screens/HomePage.dart';*/
```

```
import 'HomePage.dart';
```

```
void main() {
```

```
  runApp(MyApp());
```

```
}
```

```
class MyApp extends StatelessWidget {
```

```
  // This widget is the root of your application.
```

```
  @override
```

```

Widget build(BuildContext context) {

  return MaterialApp(

    debugShowCheckedModeBanner: false,

    title: 'Flutter Demo',

    theme: ThemeData(

      primarySwatch: Colors.blue,

    ),

    home: MyHomePage(),

  );

}

}

```

New Recipe.dart :

```

import 'package:flutter/material.dart';

import 'package:flutter_icons/flutter_icons.dart';

/*import 'package:flutter_recipee_app/Screens/RecipeDetails.dart';

import 'package:flutter_recipee_app/model/RecipeMode.dart';*/


import 'RecipeDetail.dart';

import 'RecipeModel.dart';


class NewRecipe extends StatelessWidget {

```

```

@override

Widget build(BuildContext context) {

  return Scaffold(

    body: SingleChildScrollView(

      child: Column(

        children: [

              SizedBox(

                  height: 20,

                ),

                ListView.builder(

                    physics: ScrollPhysics(),

                    shrinkWrap: true,

                    itemCount: RecipeModel.demoRecipe.length,

                    itemBuilder: (BuildContext context, int index) {

                        return Padding(

                            padding: const EdgeInsets.symmetric(

                                horizontal: 12,

                                vertical: 12,

                              ),

                            child: InkWell(

                                onTap: () => Navigator.push(

                                    context,

```

```
MaterialPageRoute(
    builder: (context) => RecipeDetails(
        recipeModel: RecipeModel.demoRecipe[index],
    ),
),
)),
child: RecipeCard(
    recipeModel: RecipeModel.demoRecipe[index],
),
),
),
);
},
),
),
),
);
}
```

```

@required this.recipeModel,

});

@override

RecipeCardState createState() => RecipeCardState();

}

```

```

class RecipeCardState extends State<RecipeCard> {

bool loved = false;

bool saved = false;

@override

Widget build(BuildContext context) {

return Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Stack(

children: [

Align(

alignment: Alignment.topCenter,

child: ClipRRect(

borderRadius: BorderRadius.circular(24),

child: Hero(

```

```

tag: widget.recipeModel.imgPath,
child: Image(
  height: 320,
  width: 320,
  fit: BoxFit.cover,
  image: AssetImage(widget.recipeModel.imgPath),
),
),
),
),
Positioned(
  top: 20,
  right: 40,
  child: InkWell(
    onTap: () {
      setState(() {
        saved = !saved;
      });
    },
    child: Icon(
      saved
        ? FlutterIcons.bookmark_check_mco

```

```

: FlutterIcons.bookmark_outline_mco,
color: Colors.white,
size: 38,
),
),
),
],
),
PreferredSize(
height: 20,
),
Padding(
padding: const EdgeInsets.symmetric(horizontal: 24.0),
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
children: [
Flexible(
flex: 2,
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(

```

```

_____ widget.recipeModel.title,
_____ style: Theme.of(context).textTheme.subtitle1,
_____) ,
_____ SizedBox(
_____ height: 8,
_____) ,
_____ Text(
_____ widget.recipeModel.writer,
_____ style: Theme.of(context).textTheme.caption,
_____) ,
_____ ,
_____ ,
_____) ,
_____) ,
_____ // Spacer(),
_____ Flexible(
_____ flex: 1,
_____ child: Row(
_____ mainAxisAlignment: MainAxisAlignment.end,
_____ children: [
_____ SizedBox(
_____ width: 20,
_____) ,

```

```

Icon(
  FlutterIcons.timer_mco,
),
  SizedBox(
    width: 4,
  ),
  Text(
    widget.recipeModel.cookingTime.toString() + "\",
  ),
  Spacer(),
  InkWell(
    onTap: () {
      setState() {
        loved = !loved;
      };
    },
    child: Icon(
      FlutterIcons.heart_circle_mco,
      color: loved ? Colors.red : Colors.black,
    ),
  ),
),

```



```
}
```

```
class _MyHomePageState extends State<MyHomePage> {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Scaffold(
```

```
      bottomNavigationBar: Container(
```

```
        // color: Colors.grey[300],
```

```
        height: 100,
```

```
        child: Row(
```

```
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
```

```
          children: [
```

```
            Icon(
```

```
              FlutterIcons.home_outline_mco,
```

```
              color: Colors.blue,
```

```
            ),
```

```
            Icon(
```

```
              FlutterIcons.account_group_outline_mco,
```

```
            ),
```

```
            Icon(
```

```
              FlutterIcons.heart_outlined_ent,
```

```
            ),
```

```

Icon(
  FlutterIcons.account_outline_mco,
),
],
),
),
body: SafeArea(
  bottom: false,
  child: DefaultTabController(
    length: 3,
    initialIndex: 0,
    child: Column(
      children: [
        SizedBox(
          height: 40,
        ),
        TabBar(
          isScrollable: true,
          indicatorColor: Colors.red,
          tabs: [
            Tab(
              text: "New Recipes".toUpperCase(),

```

```

    ),
    Tab(
      text: "Favourites".toUpperCase(),
    ),
    Tab(
      text: "Categories".toUpperCase(),
    ),
  ],
  labelColor: Colors.black,
  indicator: DotIndicator(
    color: Colors.black,
    distanceFromCenter: 16,
    radius: 3,
    paintingStyle: PaintingStyle.fill,
  ),
  unselectedLabelColor: Colors.black.withOpacity(0.3),
  labelStyle: TextStyle(
    fontSize: 12,
    fontWeight: FontWeight.w600,
  ),
  labelPadding: EdgeInsets.symmetric(
    horizontal: 24,

```

```

    ),
  ),
  Expanded(
    child: TabBarView(
      children: <Widget>[
        NewRecipe(),
        Container(
          child: Center(
            child: Text(
              'Favourite Section',
            ),
          ),
        ),
        Container(
          child: Center(
            child: Text(
              'Cetegories',
            ),
          ),
        ),
      ],
    ),
  ),

```

```

        )
      ],
    ),
  ),
);
}
}

```

RECIPE DETAIL :

```

import 'package:flutter/cupertino.dart';

import 'package:flutter/material.dart';

import 'package:flutter_icons/flutter_icons.dart';

/*import 'package:flutter_recipe_app/model/RecipeMode.dart';*/

import 'package:sliding_up_panel/sliding_up_panel.dart';

import 'package:tab_indicator_styler/tab_indicator_styler.dart';

import 'RecipeModel.dart';

class RecipeDetails extends StatelessWidget {

```

```
final RecipeModel recipeModel;
```

```
RecipeDetails({
```

```
    @required this.recipeModel,
```

```
});
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
    Size size = MediaQuery.of(context).size;
```

```
    final _textTheme = Theme.of(context).textTheme;
```

```
    return Scaffold(
```

```
        body: SlidingUpPanel(
```

```
            parallaxEnabled: true,
```

```
            borderRadius: BorderRadius.only(
```

```
                topLeft: Radius.circular(24),
```

```
                topRight: Radius.circular(24),
```

```
            ),
```

```
            padding: EdgeInsets.symmetric(
```

```
                horizontal: 12,
```

```
            ),
```

```
            minHeight: (size.height / 2),
```

```
            maxHeight: size.height / 1.2,
```

```
            panel: Padding(
```

```

padding: const EdgeInsets.all(16.0),

child: Column(

  crossAxisAlignment: CrossAxisAlignment.start,

  children: [

    Center(

      child: Container(

        height: 5,

        width: 40,

        decoration: BoxDecoration(

          color: Colors.black.withOpacity(0.3),

          borderRadius: BorderRadius.circular(12),

        ),

      ),

    ),

    ),

    SizedBox(

      height: 30,

    ),

    Text(

      recipeModel.title,

      style: _textTheme.headline6,

    ),

    SizedBox(

```

```
        height: 10,

    ),

    Text(

        recipeModel.writer,

        style: _textTheme.caption,

    ),

    SizedBox(

        height: 10,

    ),

    Row(

        children: [

            Icon(

                FlutterIcons.heart_circle_mco,

                color: Colors.red,

            ),

            SizedBox(

                width: 5,

            ),

            Text(

                "198",

                // style: _textTheme.caption,

            ),
```

```
    SizedBox(  
      width: 10,  
    ),  
    Icon(  
      FlutterIcons.timer_mco,  
    ),  
    SizedBox(  
      width: 4,  
    ),  
    Text(  
      recipeModel.cookingTime.toString() + '\",  
    ),  
    SizedBox(  
      width: 20,  
    ),  
    Container(  
      width: 2,  
      height: 30,  
      color: Colors.black,  
    ),  
    SizedBox(  
      width: 10,
```

```

    ),
    Text(
      recipeModel.servings.toString() + ' Servings',
    ),
  ],
),
SizedBox(
  height: 10,
),
Divider(
  color: Colors.black.withOpacity(0.3),
),
Expanded(
  child: DefaultTabController(
    length: 3,
    initialIndex: 0,
    child: Column(
      children: [
        TabBar(
          isScrollable: true,
          indicatorColor: Colors.red,
          tabs: [

```

```

    Tab(
      text: "Ingredients".toUpperCase(),
    ),
    Tab(
      text: "Preparation".toUpperCase(),
    ),
    Tab(
      text: "Reviews".toUpperCase(),
    ),
  ],
  labelColor: Colors.black,
  indicator: DotIndicator(
    color: Colors.black,
    distanceFromCenter: 16,
    radius: 3,
    paintingStyle: PaintingStyle.fill,
  ),
  unselectedLabelColor: Colors.black.withOpacity(0.3),
  labelStyle: TextStyle(
    fontSize: 12,
    fontWeight: FontWeight.w600,
  ),

```

```

        labelPadding: EdgeInsets.symmetric(
            horizontal: 32,
        ),
    ),
    Divider(
        color: Colors.black.withOpacity(0.3),
    ),
    Expanded(
        child: TabBarView(
            children: [
                Ingredients(recipeModel: recipeModel),
                Container(
                    child: Text("Preparation Tab"),
                ),
                Container(
                    child: Text("Reviews Tab"),
                ),
            ],
        ),
    ],
),

```

```

    ),
    ),
  ],
),
),
body: SingleChildScrollView(
  child: Stack(
    children: [
      Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: <Widget>[
          Hero(
            tag: recipeModel.imgPath,
            child: ClipRRect(
              child: Image(
                width: double.infinity,
                height: (size.height / 2) + 50,
                fit: BoxFit.cover,
                image: AssetImage(recipeModel.imgPath),
              ),
            ),
          ),
        ],
      ),
    ],
  ),
),

```

```

],
),
Positioned(
  top: 40,
  right: 20,
  child: Icon(
    FlutterIcons.bookmark_outline_mco,
    color: Colors.white,
    size: 38,
  ),
),
Positioned(
  top: 40,
  left: 20,
  child: InkWell(
    onTap: () => Navigator.pop(context),
    child: Icon(
      CupertinoIcons.back,
      color: Colors.white,
      size: 38,
    ),
  ),
),

```

```

        ),
    ],
),
),
),
);
}
}

```

```

class Ingredients extends StatelessWidget {

  const Ingredients({

    Key key,

    @required this.recipeModel,

  }) : super(key: key);

  final RecipeModel recipeModel;

  @override

  Widget build(BuildContext context) {

    return SingleChildScrollView(

      child: Padding(

        padding: const EdgeInsets.only(bottom: 12.0),

```

```

child: Column(

  children: [

    ListView.separated(

      shrinkWrap: true,

      physics: ScrollPhysics(),

      itemCount: recipeModel.ingredients.length,

      itemBuilder: (BuildContext context, int index) {

        return Padding(

          padding: const EdgeInsets.symmetric(

            vertical: 2.0,

          ),

          child: Text('● ' + recipeModel.ingredients[index]),

        );

      },

      separatorBuilder: (BuildContext context, int index) {

        return Divider(color: Colors.black.withOpacity(0.3));

      },

    ),

  ],

),

);

```

```
}
```

```
}
```

RECIPE MODAL :

```
import 'package:flutter/material.dart';
```

```
class RecipeModel {  
  String title, writer, description;  
  int cookingTime;  
  int servings;  
  List<String> ingredients = [];  
  String imgPath;  
  RecipeModel({  
    @required this.title,  
    @required this.writer,  
    @required this.description,  
    @required this.cookingTime,  
    @required this.servings,  
    @required this.imgPath,  
    @required this.ingredients,  
  });  
  static List<RecipeModel> demoRecipe = [  
    RecipeModel(  
      title: 'Spinach Scrambled Eggs',  
      writer: "JESTIN JOSE",  
      description:  
        'A touch of mustard, salty bacon, melty cheese, and a handful of greens seriously  
        upgrades scrambled eggs, without too much effort!',  
      cookingTime: 10,  
      servings: 4,  
      imgPath: 'assets/images/scrambled eggs.jpg',  
      ingredients: [  
        '8 large eggs',  
        '1 tsp. Dijon mustard',  
        'salt and pepper',  
        '1 tbsp. olive oil or unsalted butter',  
        '2 slices thick-cut bacon, cooked and broken into pieces',  
        '2 spinach, torn',  
        '2 cheese, shredded',  
      ],  
    ),  
    RecipeModel(  

```

```

    title: 'Chicken Fried Rice ',
    writer: "ANISH GEORGE",
    description:
        'Sneak some spinach into your morning meal for a boost of nutrients to start your
day off right.',
    cookingTime: 10,
    servings: 4,
    imagePath: 'assets/images/chicken.png.jpg',
    ingredients: [
        '1 egg',
        '1 tablespoon water',
        '1 tablespoon butter',
        '1 tablespoon vegetable oil',
        '1 onion, chopped',
        '2 cups cooked white rice, cold',
        '2 tablespoons soy sauce',
        '1 teaspoon ground black pepper',
        '1 cup cooked, chopped chicken meat',
    ],
),
RecipeModel(
    title: 'Idly Dosa Vada ',
    writer: "JESTIN JOSE",
    description:
        'A hearty breakfast that easily feeds a family of four, all on one sheet Vada? Yes,
please.',
    cookingTime: 10,
    servings: 4,
    imagePath: 'assets/images/dosa.png.jpg',
    ingredients: [
        'Dosa Batter',
        'Gram flour',
        'Peper',
        'Salt',
        'oil',
        'Masala ingrediants',
    ],
),
RecipeModel(
    title: 'Chicken Biriyaani',
    writer: "JESTIN JOSE",
    description:
        'Just wait til you break this one out at the Lunch table: sweet tomatoes, runny
yolks, and plenty of spices brings the aroma.',
    cookingTime: 10,
    servings: 4,
    imagePath: 'assets/images/biriyaani.png.jpg',

```

```

ingredients: [
  '4tablespoons vegetable oil, 4 small potatoes, peeled and halved',
  ' 2 large onions, finely chopped',
  '2 cloves garlic, minced',
  '1 tablespoon minced fresh ginger root',
  '½ teaspoon chili powder',
  '½ teaspoon ground black pepper',
  '½ teaspoon ground turmeric',
  '1 teaspoon ground cumin',
  '1 teaspoon salt',
  '2 medium tomatoes peeled and chopped',
  '2 tablespoons plain yogurt',
  '2 tablespoons chopped fresh mint leaves',
  '½ teaspoon ground cardamom',
  '1 (2 inch) piece cinnamon stick',
  '3 pounds boneless, skinless chicken pieces cut into chunks',
  '2 ½ tablespoons vegetable oil',
  '1 large onion, diced',
  '1 pinch powdered saffron',
  '5 pods cardamom',
  '3 whole cloves',
  '1 (1 inch) piece cinnamon stick',
  '½ teaspoon ground ginger',
  '1 pound basmati rice',
  '4 cups chicken stock',
  '1 ½ teaspoons salt',
],
),
];
}

```

PUBSPEC.YAML :

Aname: flutter_recipe_app
description: A new Flutter project.

The following line prevents the package from being accidentally published to
pub.dev using `pub publish`. This is preferred for private packages.
publish_to: "none" # Remove this line if you wish to publish to pub.dev

The following defines the version and build number for your application.
A version number is three numbers separated by dots, like 1.2.43
followed by an optional build number separated by a +.
Both the version and the builder number may be overridden in flutter
build by specifying --build-name and --build-number, respectively.

```
# In Android, build-name is used as versionName while build-number used as
versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number used
as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
version: 1.0.0+1
```

```
environment:
  sdk: ">=2.7.0 <3.0.0"
```

```
dependencies:
  flutter:
    sdk: flutter
  flutter_icons: ^1.1.0
  sliding_up_panel: ^1.0.2
  tab_indicator_styler: ^1.0.0
```

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
```

```
dev_dependencies:
  flutter_test:
    sdk: flutter
```

```
# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec
```

```
# The following section is specific to Flutter.
```

```
flutter:
  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true
```

```
# To add assets to your application, add an assets section, like this:
```

```
assets:
  - assets/images/
# - images/a_dot_ham.jpeg
# An image asset can refer to one or more resolution-specific "variants", see
# https://flutter.dev/assets-and-images/#resolution-aware.
# For details regarding adding assets from package dependencies, see
```

```

# https://flutter.dev/assets-and-images/#from-packages
# To add custom fonts to your application, add a fonts section here,
# in this "flutter" section. Each entry in this list should have a
# "family" key with the font family name, and a "fonts" key with a
# list giving the asset and other descriptors for the font. For
# example:
# fonts:
#   - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-Regular.ttf
#       - asset: fonts/Schyler-Italic.ttf
#         style: italic

```

OUTPUT :

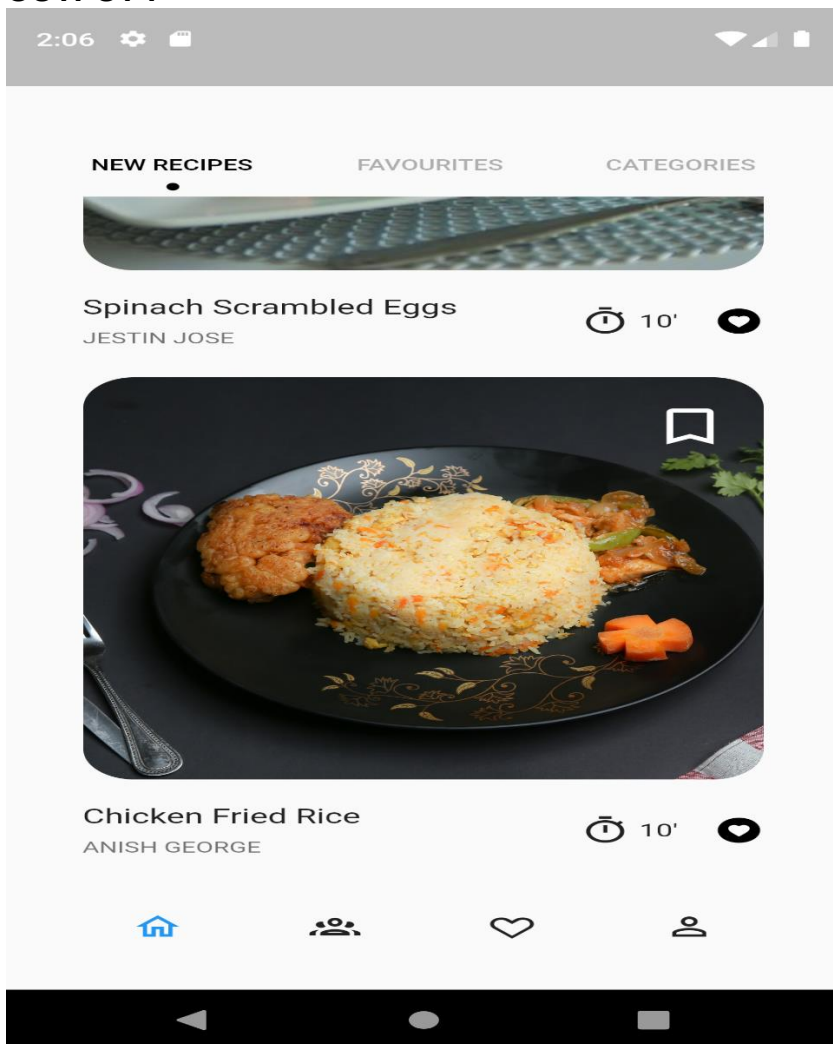


Fig 2.1 Recipe page

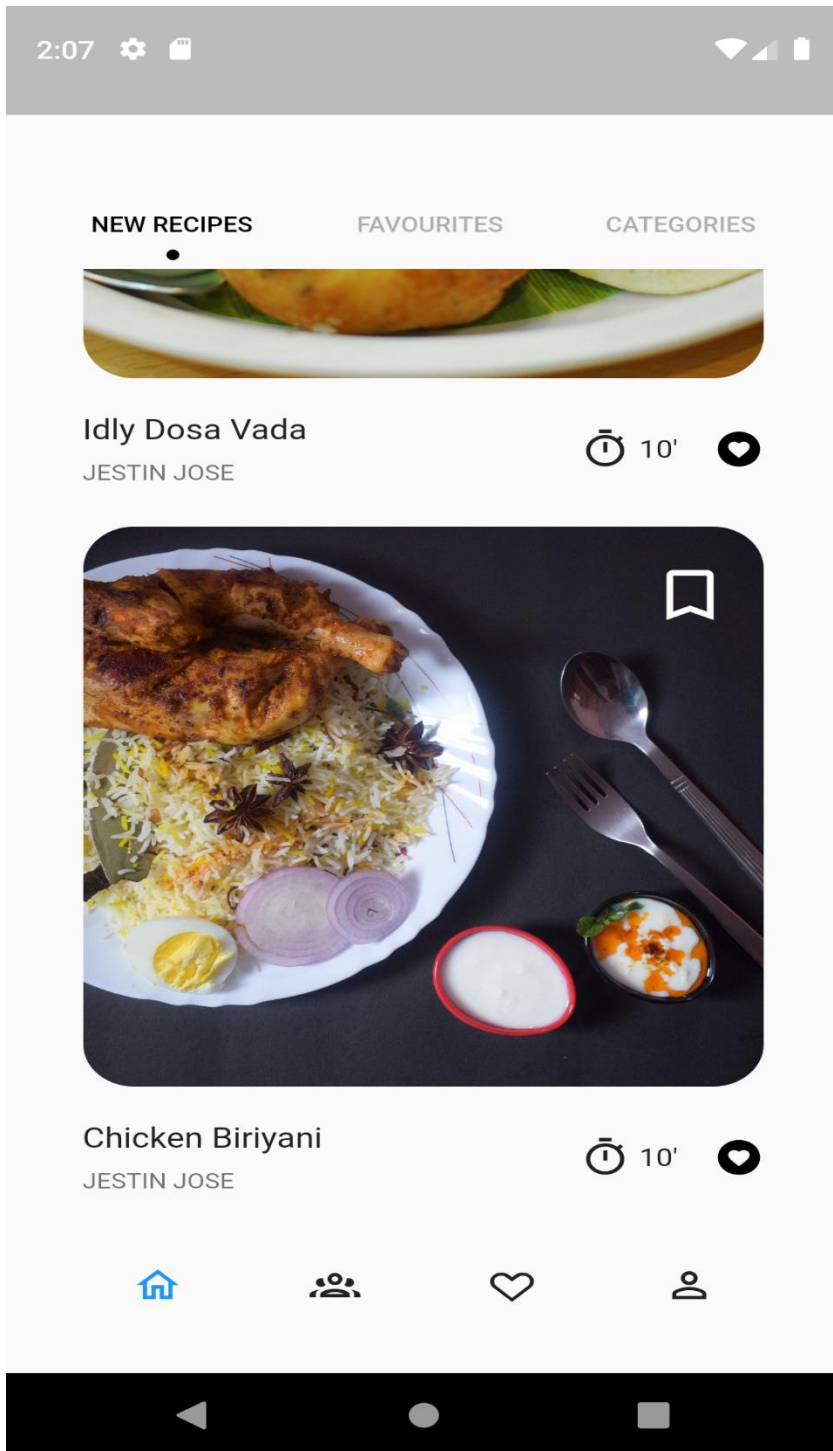
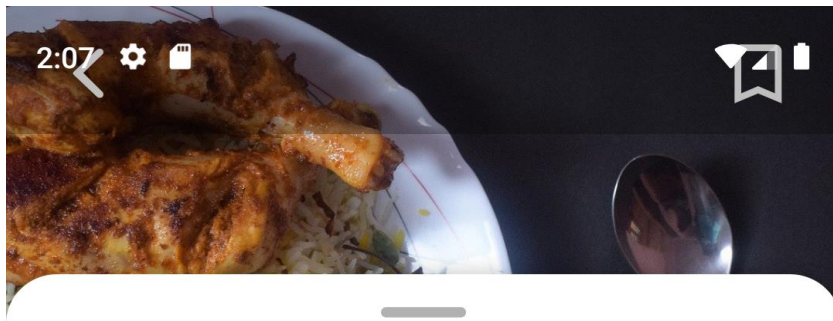


Fig 2.2 Recipe page



Chicken Biryani

JESTIN JOSE



198



10'



4 Servings

INGREDIENTS

PREPARATION

REV



● 4tablespoons vegetable oil, 4 small potatoes, peeled and halved

● 2 large onions, finely chopped

● 2 cloves garlic, minced

● 1 tablespoon minced fresh ginger root

● ½ teaspoon chili powder

● ½ teaspoon ground black pepper

● ½ teaspoon ground turmeric

● 1 teaspoon ground cumin

● 1 teaspoon salt

Fig 2.3 Ingrident page