

Cookipidia

A Project Work

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

WITH SPECIALIZATION

IN

MOBILE COMPUTING

Submitted by:

Eish Jindal(19BCS4251)

Utkarsh Chauhan (19BCS4270)

Shubham Mahajan (19BCS4275)

Under the Supervision of:

Mr. Gaurav Soni



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

APEX INSTITUE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,

PUNJAB

May, 2022

DECLARATION

I, ‘Utkarsh Chauhan’, student of ‘**Bachelor of Engineering in Computer Science and Engineering with specialization in Mobile Computing**’, session: **2019-2023**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Cookipidia**’ is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Utkarsh Chauhan

19BCS4270

Date: 03/05/2022

Place: Chandigarh University

ABSTRACT

Cookipidia is an application that helps us to provide recipes of dishes. It provides shopping for required veggie. In this application there is a chat feature so that viewers can directly connect to chefs and get their doubts solved. It has intelligent dish recommendation based on timing of the day. It also has a creator's panel where a person can post blog/photo of self-made things.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Mr. Gaurav Soni** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents and my department for their kind cooperation and encouragement which help me in completion of this project.

THANKS AGAIN TO ALL WHO HELPED

List of Figures

Figure Title	Page No.
Existing System Fig. -1	2.
Existing System Fig. -2	2.
Existing System Fig. -3	3.
Existing System Fig. -4	3.
Existing System Fig. -5	4.
Proposed System Fig. -6	4.
Proposed System Fig. -7	4.
Proposed System Fig. -8	4.
Proposed System Fig. -9	5.
Proposed System Fig. -10	5.
Proposed System Fig. -11	5.
Proposed System Fig. -12	5.
Methodology Fig. -13	17
Methodology Fig. -14	17
Methodology Fig. -15	18
Methodology Fig. -16	18
Methodology Fig. -17	19
Methodology Fig. -18	19
Methodology Fig. -19	20
Methodology Fig. -20	20
Methodology Fig. -21	21
Methodology Fig. -22	21
Methodology Fig. -23	22
Methodology Fig. -24	22
Methodology Fig. -25	23
Methodology Fig. -26	23
Methodology Fig. -27	24
Methodology Fig. -28	24
Methodology Fig. -29	25
Methodology Fig. -30	25
Methodology Fig. -31	26
Methodology Fig. -32	26
Methodology Fig. -33	27

Methodology Fig. -34	27
Methodology Fig. -35	28
Methodology Fig. -36	28
Methodology Fig. -37	29
Methodology Fig. -38	29
Methodology Fig. -39	30
Methodology Fig. -40	30
Methodology Fig. -41	31
Methodology Fig. -42	31
Methodology Fig. -43	32
Methodology Fig. -44	32
Methodology Fig. -45	33
Methodology Fig. -46	33
Methodology Fig. -47	34
Methodology Fig. -48	34
Methodology Fig. -49	35
Methodology Fig. -50	35
Methodology Fig. -51	36
Methodology Fig. -52	36
Methodology Fig. -53	37
Methodology Fig. -54	37
Methodology Fig. -55	38
Methodology Fig. -56	38
Methodology Fig. -57	39
Methodology Fig. -58	39
Methodology Fig. -59	40
Methodology Fig. -60	40
Methodology Fig. -61	41
Methodology Fig. -62	41
Methodology Fig. -63	42
Methodology Fig. -64	42
Methodology Fig. -65	43
Methodology Fig. -66	43
Methodology Fig. -67	44
Methodology Fig. -68	44
Methodology Fig. -69	45
Methodology Fig. -70	45

Methodology Fig. -71	46
Methodology Fig. -72	46
Methodology Fig. -73	47
Methodology Fig. -74	47
Methodology Fig. -75	48
Methodology Fig. -76	48
Methodology Fig. -77	49
Methodology Fig. -78	49
Methodology Fig. -79	50
Methodology Fig. -80	50
Methodology Fig. -81	51
Methodology Fig. -82	51
Methodology Fig. -83	52
Methodology Fig. -84	52
Methodology Fig. -85	53
Methodology Fig. -86	53
Methodology Fig. -87	54
Methodology Fig. -88	54
Methodology Fig. -89	55
Methodology Fig. -90	55
Methodology Fig. -91	56
Methodology Fig. -92	56
Methodology Fig. -93	57
Methodology Fig. -94	57
Methodology Fig. -95	58
Methodology Fig. -96	58
Methodology Fig. -97	59
Methodology Fig. -98	59
Methodology Fig. -99	60
Methodology Fig. -100	60
Methodology Fig. -101	61
Methodology Fig. -102	61
Methodology Fig. -103	62
Methodology Fig. -104	62
Methodology Fig. -105	63
Methodology Fig. -106	63
Methodology Fig. -107	64

Methodology Fig. -108	64
Methodology Fig. -109	65
Methodology Fig. -110	65
Methodology Fig. -111	66
Methodology Fig. -112	66
Methodology Fig. -113	67
Methodology Fig. -114	67
Methodology Fig. -115	68
Methodology Fig. -116	68
Methodology Fig. -117	69
Methodology Fig. -118	69
Methodology Fig. -119	70
Methodology Fig. -120	70
Methodology Fig. -121	71
Methodology Fig. -122	71
Methodology Fig. -123	72
Methodology Fig. -124	72
Methodology Fig. -125	73
Methodology Fig. -126	73
Methodology Fig. -127	74
Methodology Fig. -128	74
Methodology Fig. -129	75
Methodology Fig. -130	75
Methodology Fig. -131	76
Methodology Fig. -132	76
Methodology Fig. -133	77
Methodology Fig. -134	77
Methodology Fig. -135	78
Methodology Fig. -136	78
Methodology Fig. -137	79
Methodology Fig. -138	79
Methodology Fig. -139	80
Methodology Fig. -140	80
Methodology Fig. -141	81
Methodology Fig. -142	81
Methodology Fig. -143	82

Methodology Fig. -144	82
Methodology Fig. -145	83
Methodology Fig. -146	83
Methodology Fig. -147	84
Methodology Fig. -148	84
Methodology Fig. -149	85
Methodology Fig. -150	85
Methodology Fig. -151	86
Methodology Fig. -152	86
Methodology Fig. -153	87
Methodology Fig. -154	87
Methodology Fig. -155	88
Methodology Fig. -156	88
Methodology Fig. -157	89

Timeline

- 1. 27th February 2022:** Getting started.
- 2. 1st March 2022:** Basic Widgets modelling.
- 3. 3rd March 2022:** Designing Splash Screen and certain pages.
- 4. 5th March 2022:** Applying Scroll View.
- 5. 8th March 2022:** Making Interactive Widgets for users.
- 6. 10th March 2022:** Routing and Navigating the app with Navigator 2.0.
- 7. 12th March 2022:** Applying Deep links and web URL's.
- 8. 14th March 2022:** Using Shared Preferences to Store data.
- 9. 18th March 2022:** Serialization with JSON.
- 10. 20th March 2022:** Providing a network.
- 11. 24th March 2022:** Using Chopper Library in app.
- 12. 28th March 2022:** Building a better state management architecture.
- 13. 1st April 2022:** Saving data with SQLite.
- 14. 3rd April 2022:** Adding platform specific app assets.
- 15. 5th April 2022:** Adding Firestore functionality.
- 16. 7th April 2022:** Adding chat functionality in the app.
- 17. 11th April 2022:** Debugging DAO.
- 18. 15th April 2022:** Connecting the Backend.
- 19. 19th April 2022:** Implementing Messaging Services.
- 20. 23rd April 2022:** Extending the expiry date of I/O operations in firebase.
- 21. 27th April 2022:** Making the UI clean in common room database.
- 22. 1st May 2022:** Integrating all three parts.
- 23. 5th May 2022:** Making infinite scrolling pagination in common room.
- 24. 7th May 2022:** Final Debugging + Cleaning the UI.

Table of Content

Title Page	i
Declaration	ii
Abstract	iii
Acknowledgement	iv
List of Figures	v-ix
Timeline	x
1. INTRODUCTION	1
1.1 Problem Definition	1
1.2 Project Overview/Specifications	1
1.3 Software Specification	1
2. LITERATURE SURVEY	2-13
2.1 Existing System	2-5
2.2 Proposed System	6-13
3. PROBLEM FORMULATION	14
4. RESEARCH OBJECTIVES	15
5. METHODOLOGY	16-89
5.1. Codes for Developing the App	17-89
6. RESULTS AND CONCLUSION	90
7. REFERENCES	91-92

1. INTRODUCTION

1.1. Problem Definition: This app focuses on the problems of the new generation that is cooking and eating healthy food. Most of working-class people are going the easy and unhealthy way that is fast food, because, they don't know how to make good home cooked food.

1.2. Project Overview/Specification:

- This app is cross platform app which helps people to learn cooking.
- This app contains written as well as video tutorials of recipes.
- This app provides chatting interface for clearing doubts and getting additional guidance.
- This app is ads free and clutter free which makes it easy to use.

1.3. Software Specification:

1) Windows / Mac OS

- 2 GB RAM minimum, 4 GB RAM recommended.
- 400 MB hard disk space plus at least 1 GB for Android SDK, emulator system images, and caches.
- Java Development Kit (JDK) 8.
- Java Runtime Environment (JRE) 8.
- Flutter SDK and DART SDK. (Latest Version)
- Optional for accelerated emulator: Intel processor with support for Intel VTx, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality.

2) Linux

- 2 GB RAM minimum, 4 GB RAM recommended.
- 400 MB hard disk space plus at least 1 GB for Android SDK, emulator system images, and caches.
- Oracle Java Development Kit (JDK) 8.
- GNU C Library (glibc) 2.11 or later

2. LITERATURE REVIEW

2.1. Existing System: In Existing Systems Cookipidia is Cluttered and feature less UI which is less user intuitive:

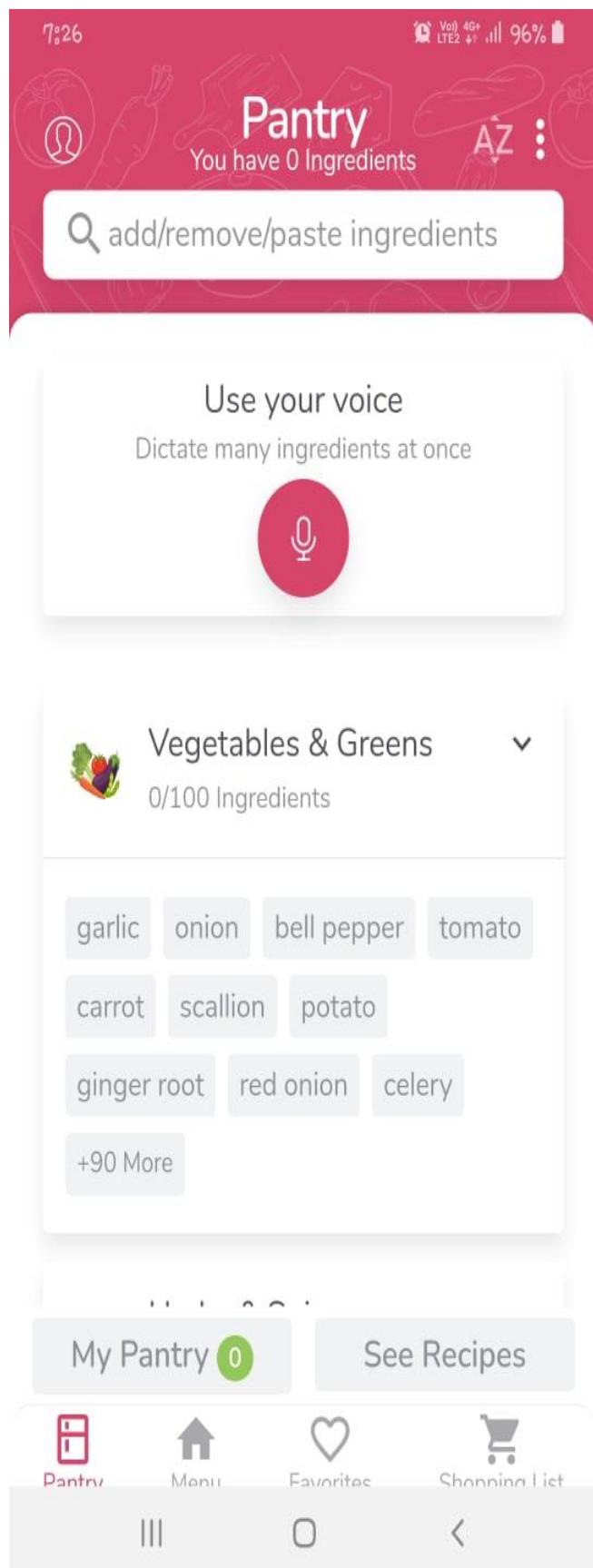


Fig 1.



Your shopping list is
empty

In this Image the App shows that
it will contain Shopping List.

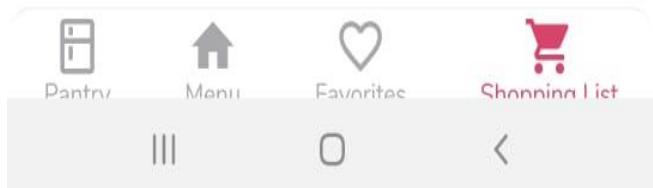
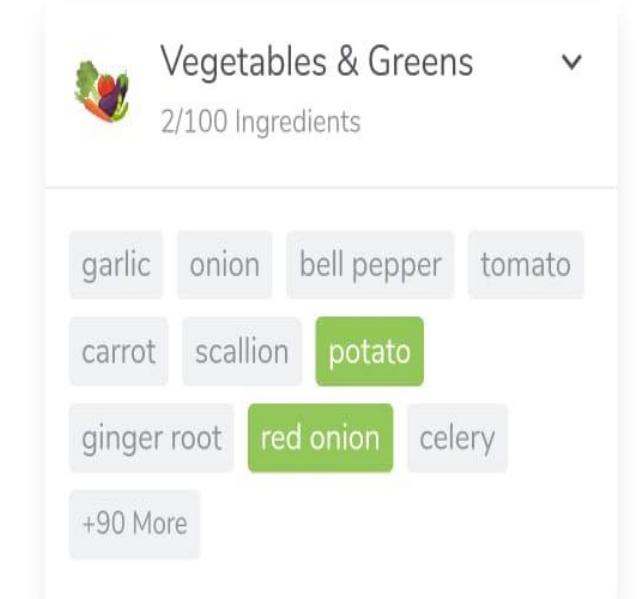
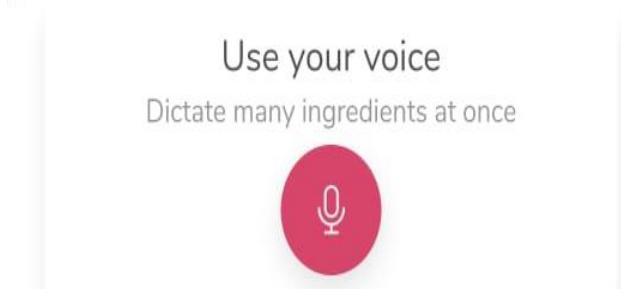
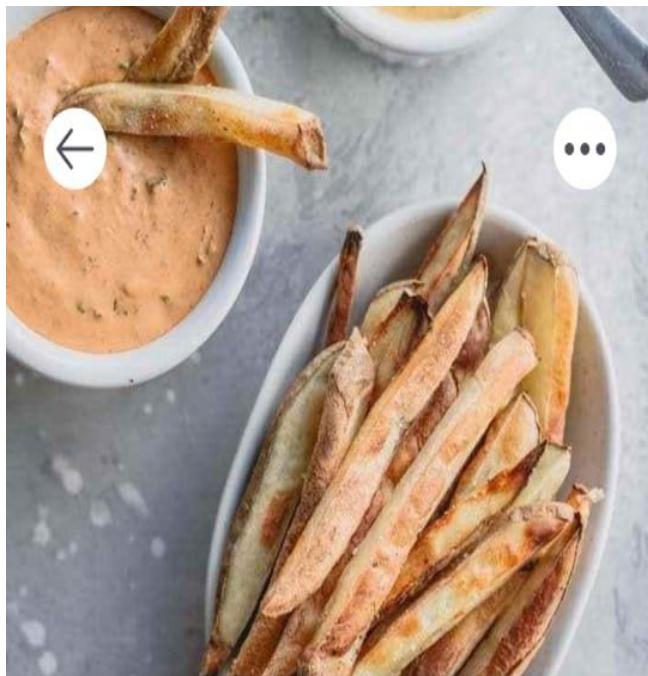


Fig 2.



In this Image the app shows that it will contain Pantry which shows that ingredients for recipe.

Fig 3.



Oil-Free Baked Fries



You have all the ingredients



50 mins

Tags

american

sides

healthy

easy

Ingredients

2 russet potatoes



salt and pepper



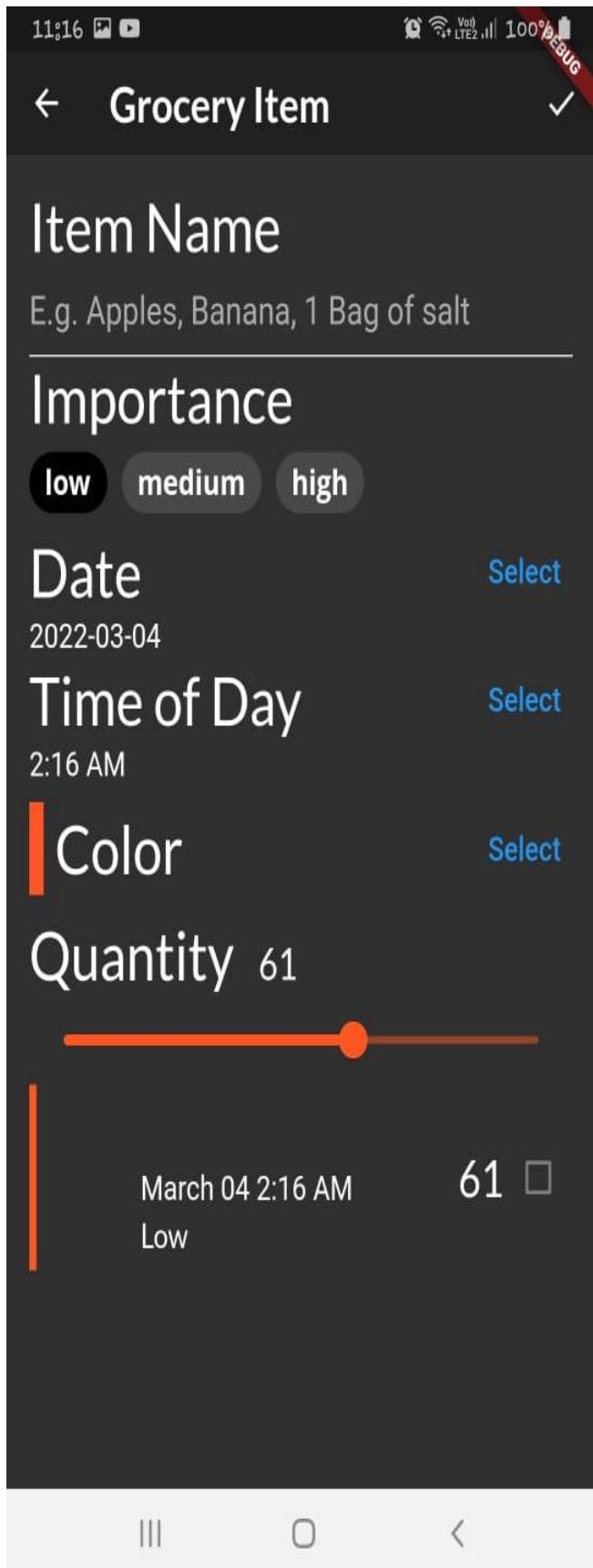
[View Full Recipe](#)

runningonrealfood.com



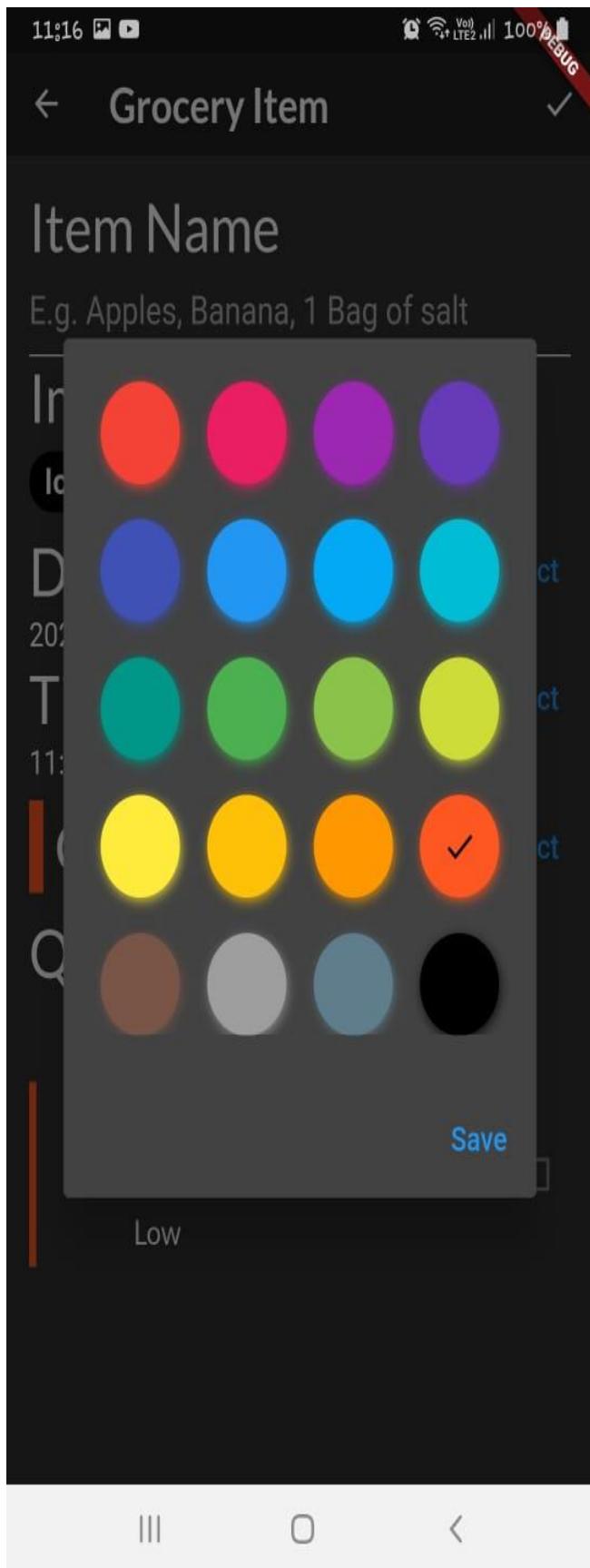
Fig 4.

2.2. Proposed System: With the proposed system we were able to bring functional and mature designs as follows:



In this Image the app shows the Grocery Item Page.

Fig 5.



In this Image the app shows the customization option.

Fig 6.

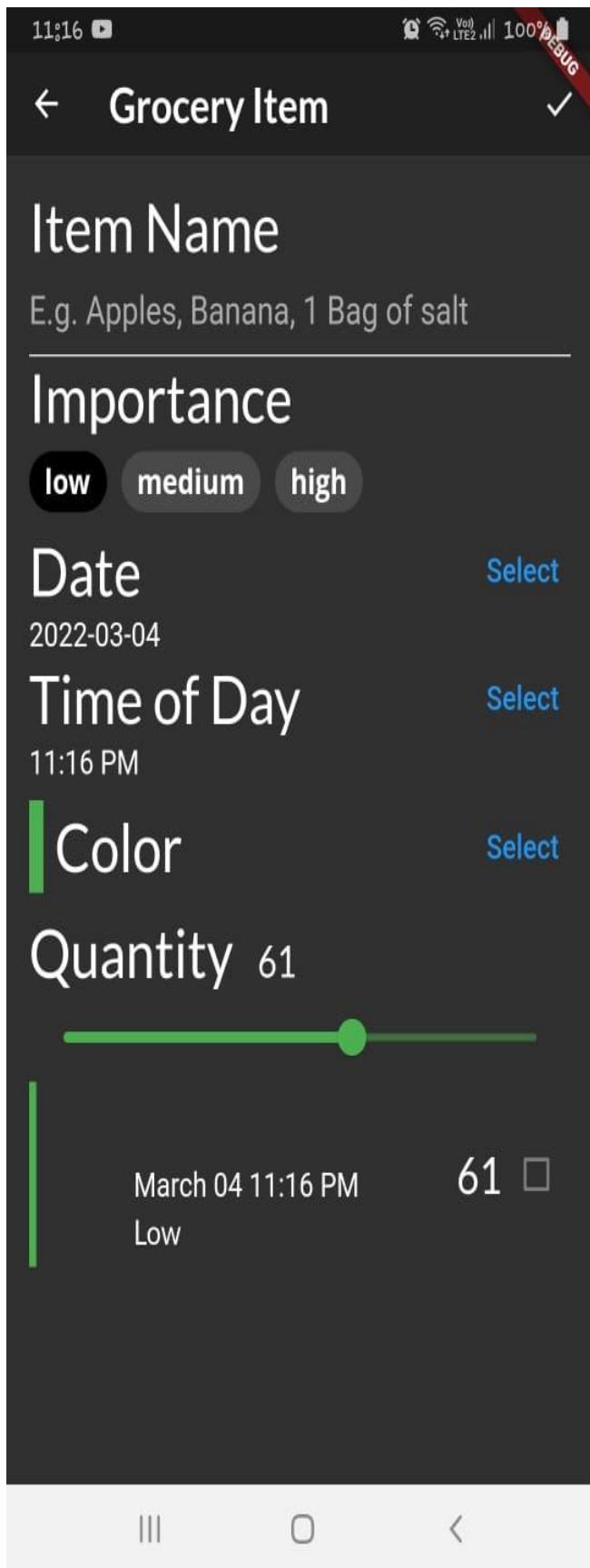


Fig 7.

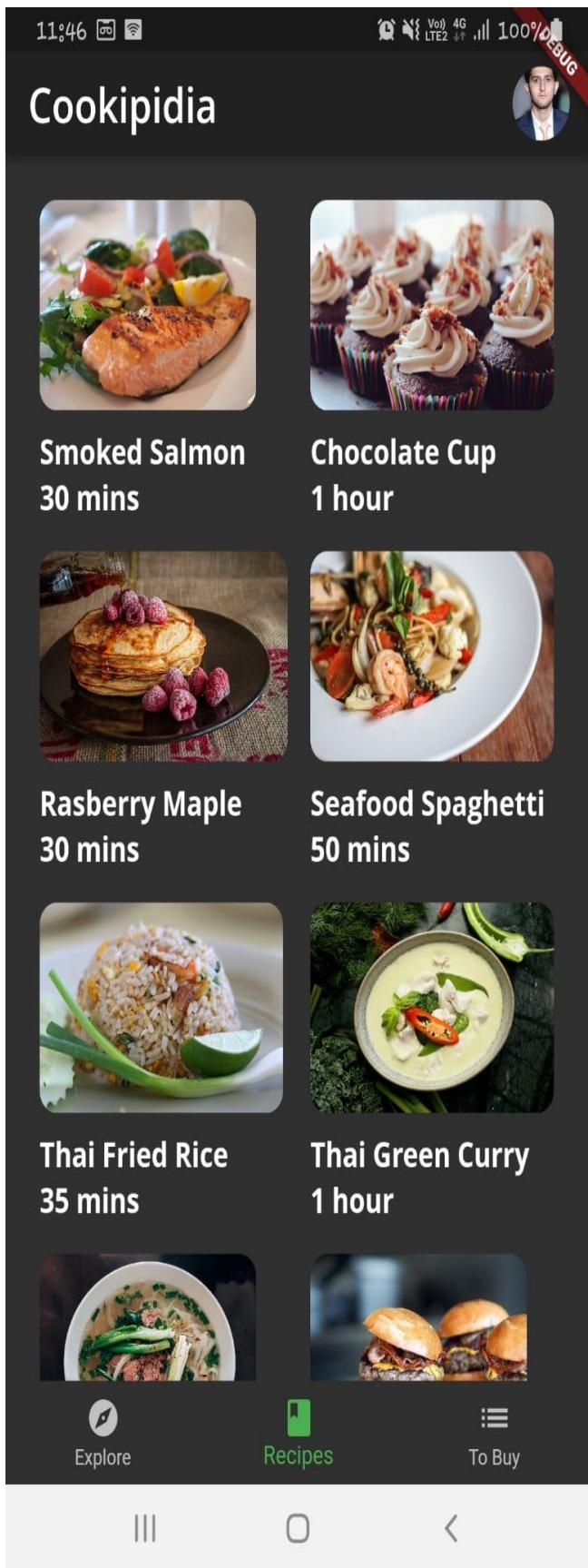
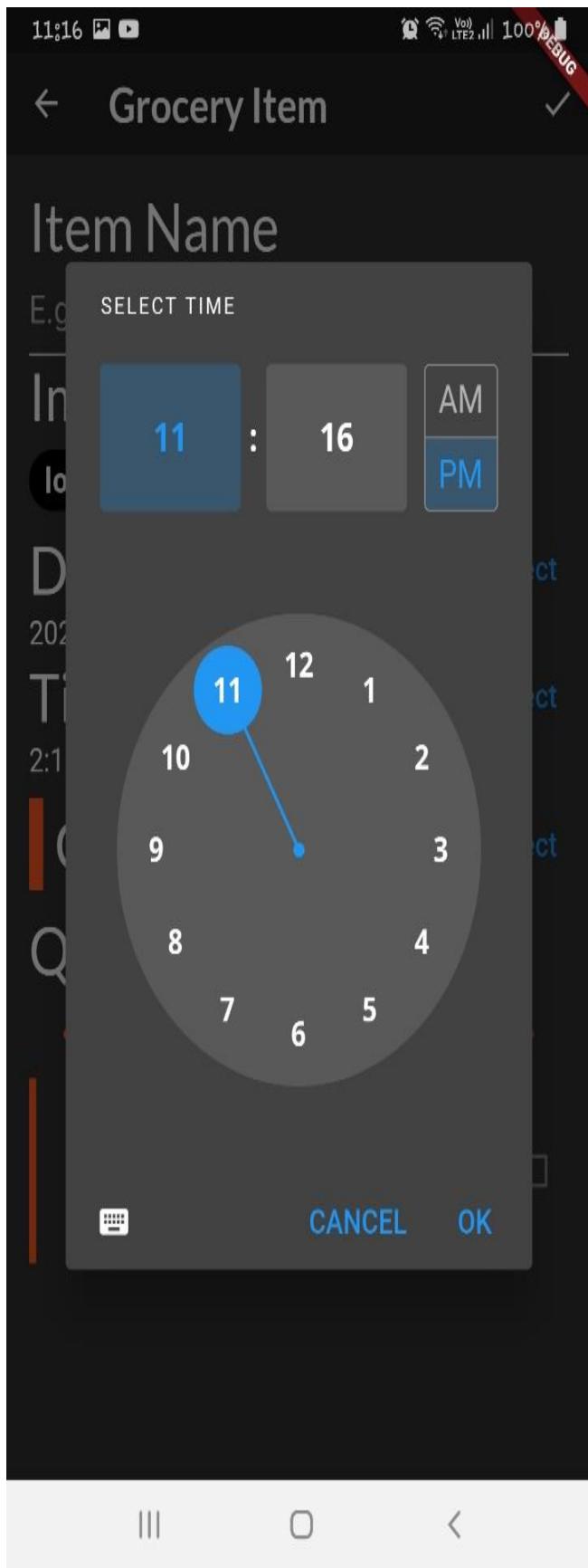


Fig 8.

In this Image the app shows the Recipe Page.



In this Image the app shows the Reminder of the grocery items.

Fig 9.



In this Image the app shows the Explorer Page in which you can chat as well.

Fig 10.

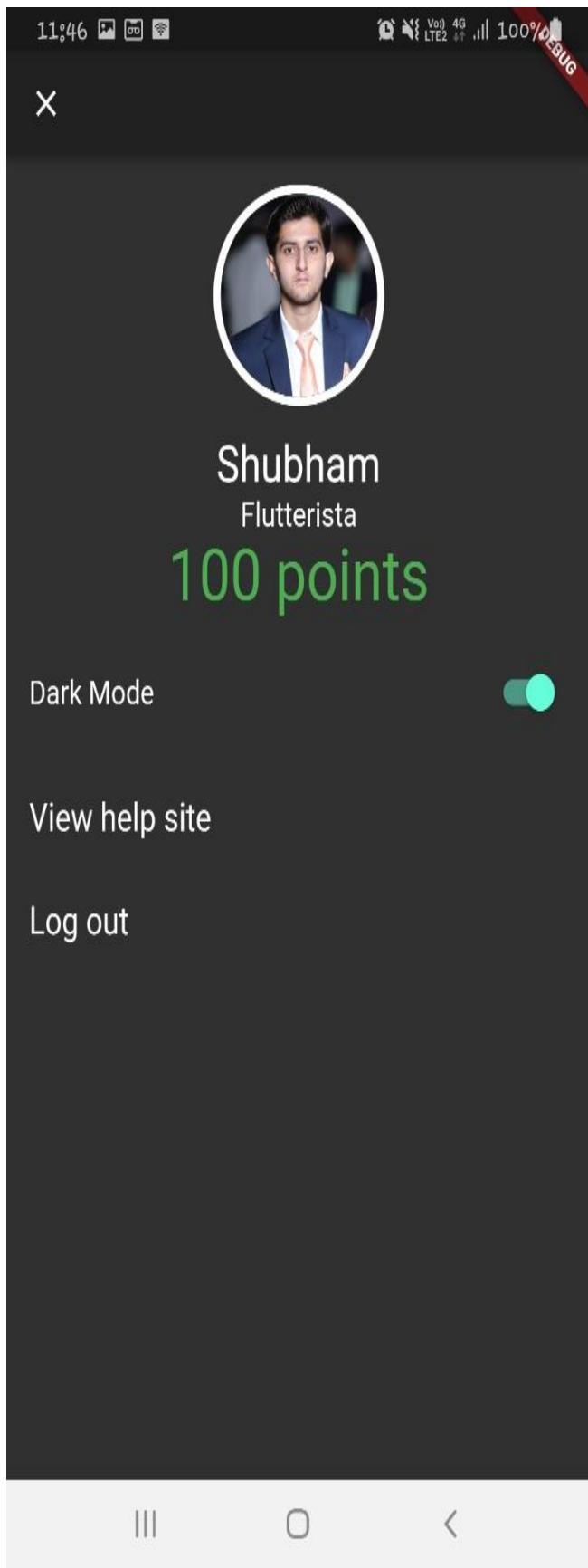


Fig 11.

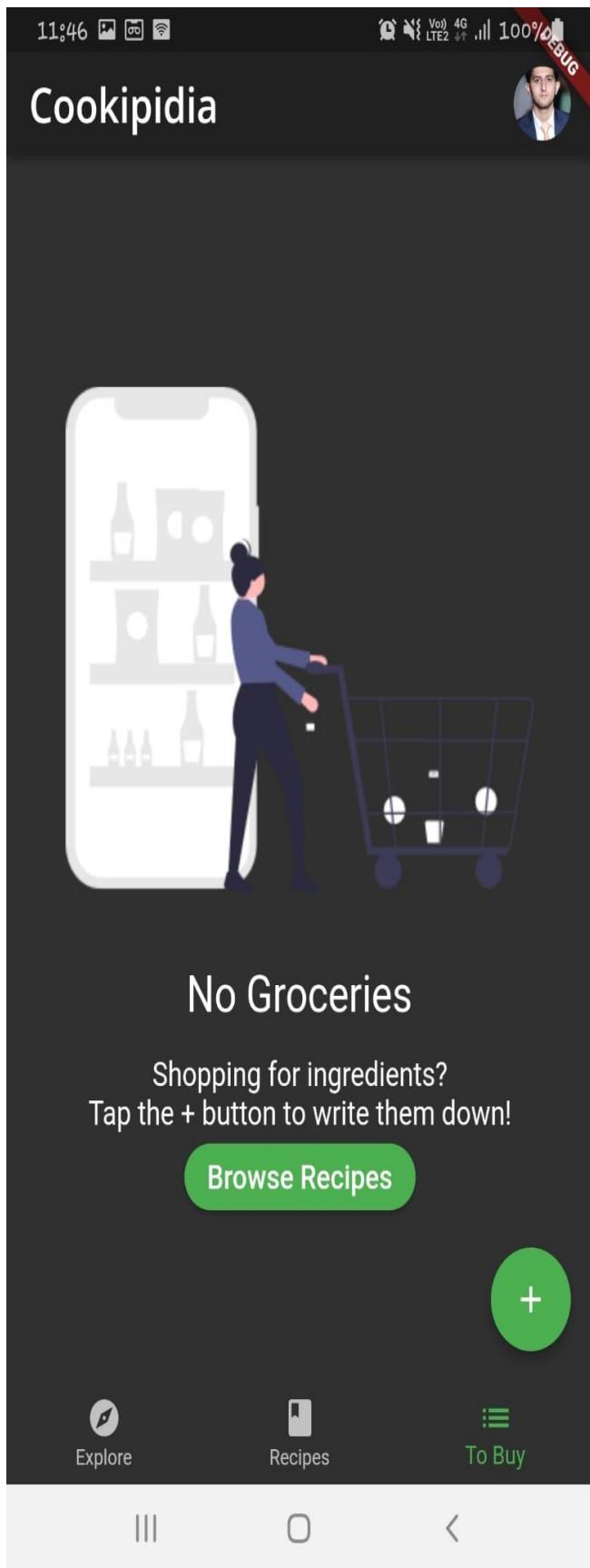


Fig 12.

3. PROBLEM FORMULATION

During software development, the main problem with existing app is:

- 1.** No existing system gives interactive learning on cooking.
- 2.** Most app available are just cook books or video Library that too un maintained.
- 3.** Current apps have adware.
- 4.** There is no app which has collaborative chat feature.
- 5.** No app contains video + text recipes.

The Problems were tackled by:

- 1.** App made will be totally free.
- 2.** No subscription fees or Ad-ware in the app.
- 3.** Unlimited Members can be added to the chat room.
- 4.** No limit on Number of queries per user.
- 5.** Easy to use clutter free UI.

4. RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of an approach for cross platform app for learning cooking through written and videos method assisted with chat interface. The proposed aim will be achieved by dividing the work into following objectives:

- 1.** The main aim of Cookipidia is to give people who are not very aware how to cook properly a guidance to tackle their everyday problem.
- 2.** Since the pandemic people prefer to stay in home and eat something tasty and nutritious.
- 3.** Current system was tested.
- 4.** Flaws of current system were detected.
- 5.** Wishlist of requested features was made.
- 6.** New app with ironed flaws and new features was created and tested.

5. METHODOLOGY

The following methodology will be followed to achieve the objectives defined for proposed research work:

- 1.** Client-Server Architecture Pattern is used for designing the App.
- 2.** New features will be implemented to enhance user experience.
- 3.** Various parameters will be identified to evaluate the proposed system.
- 4.** Comparison of new implemented approach with existing approaches will be done.
- 5.** Feedback will be taken from research.
- 6.** New ideas will be implemented to eliminate cons.
- 7.** New features will be implemented to enhance user experience.
- 8.** It focuses on providing users with basic idea of cooking ingredients.
- 9.** App using following tools was created:
 - Flutter toolkit
 - Base language dart
 - Cloud database: Firebase
 - VS Code
 - Android Studio
 - Few Flutter widgets

5.1. Codes For Developing the App

```
1 main.dart
2
3
4 import 'package:cookipidia/screens/splash_screen.dart';
5 import 'package:flutter/material.dart';
6 import 'package:flutter/scheduler.dart';
7 import 'cookipidia_theme.dart';
8 import 'screens/home.dart';
9 import 'package:provider/provider.dart';
10 import 'models/models.dart';
11 import 'navigation/app_router.dart';
12
13 void main() {
14   runApp(const Cookipidia());
15 }
16
17 class Cookipidia extends StatefulWidget {
18   const Cookipidia({Key? key}) : super(key: key);
19
20   @override
21   State<Cookipidia> createState() => _CookipidiaState();
22 }
23
24 class _CookipidiaState extends State<Cookipidia> {
25   final _groceryManager = GroceryManager();
26   final _profileManager = ProfileManager();
27   final _appStateManager = AppStateManager();
28
29   late AppRouter _appRouter;
30
31   @override
32   void initState() {
33     _appRouter = AppRouter(
34       appBarManager: _appStateManager,
35       groceryManager: _groceryManager,
36       profileManager: _profileManager,
37     );
38     super.initState();
39   }
40
41   @override
42   Widget build(BuildContext context) {
43     return MultiProvider(
44       providers: [
45         ChangeNotifierProvider(
46           create: (context) => _groceryManager,
47         ),
48         ChangeNotifierProvider(
49           create: (context) => _profileManager,
50         ),
51         ChangeNotifierProvider(
52       ],
53     );
54   }
55 }
```

Fig 13.

Activities Text Editor

May 3 11:00

*Untitled Document 1

Save

```
88 import 'package:google_fonts/google_fonts.dart';
89
90 class CookipediaTheme {
91   // 1
92   static TextTheme lightTextTheme = TextTheme(
93     bodyText1: GoogleFonts.openSans(
94       fontSize: 14.0,
95       fontWeight: FontWeight.w700,
96       color: Colors.black,
97     ),
98     headline1: GoogleFonts.openSans(
99       fontSize: 32.0,
100      fontWeight: FontWeight.bold,
101      color: Colors.black,
102    ),
103     headline2: GoogleFonts.openSans(
104       fontSize: 21.0,
105       fontWeight: FontWeight.w700,
106       color: Colors.black,
107     ),
108     headline3: GoogleFonts.openSans(
109       fontSize: 16.0,
110       fontWeight: FontWeight.w600,
111       color: Colors.black,
112     ),
113     headline4: GoogleFonts.openSans(
114       fontSize: 20.0,
115       fontWeight: FontWeight.w600,
116       color: Colors.black,
117     ),
118   );
119 }
120
121 // 2
122 static TextTheme darkTextTheme = TextTheme(
123   bodyText1: GoogleFonts.openSans(
124     fontSize: 14.0,
125     fontWeight: FontWeight.w700,
126     color: Colors.white,
127   ),
128   headline1: GoogleFonts.openSans(
129     fontSize: 32.0,
130     fontWeight: FontWeight.bold,
131     color: Colors.white,
132   ),
133   headline2: GoogleFonts.openSans(
134     fontSize: 21.0,
135     fontWeight: FontWeight.w700,
136     color: Colors.white,
137   ),
138   headline3: GoogleFonts.openSans(
139     fontSize: 16.0,
```

Tab Width: 8 ▾ Ln 19, Col 1 ▾ INS

Fig 14.

Activities Text Editor

May 3 11:01 *Untitled Document 1 Save

```
342     }],
343   ],
344 },
345 );
346 }
347 }
348 }
349 }
350 card1.dart
351 import 'package:cookipidia/cookipidia_theme.dart';
352 import 'package:cookipidia/main.dart';
353 import 'package:flutter/material.dart';
354 import 'package:cookipidia/cookipidia_theme.dart';
355
356 import '../models/explore_recipe.dart';
357
358 class Card1 extends StatelessWidget {
359   final ExploreRecipe recipe;
360
361   const Card1({
362     Key? key,
363     required this.recipe,
364   }) : super(key: key);
365
366   // 2
367   @override
368   Widget build(BuildContext context) {
369   // 3
370     return Center(
371       child: Container(
372         child: Stack(
373           children: [
374             Text(
375               recipe.subtitle,
376               style: CookipidiaTheme.darkTextTheme.bodyText1,
377             ),
378             Positioned(
379               child: Text(
380                 recipe.title,
381                 style: CookipidiaTheme.darkTextTheme.headline2,
382               ),
383               top: 20,
384             ),
385             Positioned(
386               child: Text(
387                 recipe.message,
388                 style: CookipidiaTheme.darkTextTheme.bodyText1,
389               ),
390               bottom: 30,
391               right: 0,
392             )
393           ],
394         ),
395       ),
396     );
397   }
398 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 15.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
421 card2.dart
422 import 'package:flutter/material.dart';
423 import '../models/explore_recipe.dart';
424 import 'author_card.dart';
425 import '../cookipidia_theme.dart';
426
427 class Card2 extends StatelessWidget {
428   final ExploreRecipe recipe;
429
430   const Card2({
431     Key? key,
432     required this.recipe,
433   }) : super(key: key);
434   @override
435   Widget build(BuildContext context) {
436     return Center(
437       child: Container(
438         constraints: const BoxConstraints.expand(
439           width: 350,
440           height: 450,
441         ),
442         decoration: BoxDecoration(
443           image: DecorationImage(
444             image: AssetImage(recipe.backgroundImage),
445             fit: BoxFit.cover,
446           ),
447           borderRadius: const BorderRadius.all(Radius.circular(10.0)),
448         ),
449         child: Column(
450           children: [
451             AuthorCard(
452               authorName: recipe.authorName,
453               title: recipe.role,
454               imageProvider: AssetImage(recipe.profileImage),
455             ),
456             Expanded(
457               child: Stack(
458                 children: [
459                   Positioned(
460                     bottom: 16,
461                     right: 16,
462                     child: Text(
463                       recipe.title,
464                       style: CookipidiaTheme.lightTextTheme.headline1,
465                     ),
466                   ),
467                   Positioned(
468                     bottom: 78,
469                     left: 16,
470                     child: RotatedBox(
471
```

Tab Width: 8 ▾ Ln 19, Col 1 ▾ INS

Fig 16.

A screenshot of a Linux desktop environment showing a text editor window titled "Untitled Document 1". The window displays Dart code for a "Card3" widget. The code includes imports for flutter/material.dart, cookipedia_theme.dart, and models/explore_recipe.dart. It defines a "Card3" class that extends StatelessWidget and contains logic for creating a list of "TagChips" from the recipe's tags. The code uses various Flutter and Dart libraries like BoxDecoration, BorderRadius, and Container. The text editor interface includes tabs for "Dart", "Tab Width: 8", "Ln 19, Col 1", and "INS". The bottom of the screen shows a dock with icons for various applications.

```
473     ),
474     style: CookipediaTheme.lightTextTheme.headline1,
475   ),
476   ],
477   ],
478   ],
479   ],
480   ],
481   ],
482   ],
483   );
484 );
485 );
486 }
487
488 card3.dart
489 import 'package:flutter/material.dart';
490 import '../cookipedia_theme.dart';
491 import '../models/explore_recipe.dart';
492
493 class Card3 extends StatelessWidget {
494   final ExploreRecipe recipe;
495
496   const Card3({
497     Key? key,
498     required this.recipe,
499   }) : super(key: key);
500
501   List<Widget> createTagChips() {
502     final chips = <Widget>[];
503     recipe.tags.take(6).forEach((element) {
504       final chip = Chip(
505         label: Text(
506           element,
507           style: CookipediaTheme.darkTextTheme.bodyText1,
508         ),
509         backgroundColor: Colors.black.withOpacity(0.7),
510       );
511       chips.add(chip);
512     });
513
514     return chips;
515   }
516   @override
517   Widget build(BuildContext context) {
518     return Center(
519       child: Container(
520         constraints: const BoxConstraints.expand(
521           width: 350,
522           height: 450,
523         ),
524         decoration: DecorationImage(
525           image: AssetImage(recipe.backgroundImage),
526           fit: BoxFit.cover,
527         ),
528         borderRadius: const BorderRadius.all(
529           Radius.circular(10.0),
530         ),
531       ),
532       child: Stack(
533         children: [
534           Container(
535             decoration: BoxDecoration(
536               color: Colors.black.withOpacity(0.6),
537               borderRadius: const BorderRadius.all(Radius.circular(10.0)),
538             ),
539           ),
540           Container(
541             padding: const EdgeInsets.all(16),
542             child: Column(
543               crossAxisAlignment: CrossAxisAlignment.start,
544               children: [
545                 const Icon(
546                   Icons.book,
547                   color: Colors.white,
548                   size: 40,
549                 ),
550                 const SizedBox(height: 8),
551                 Text(recipe.title,
552                   style: CookipediaTheme.darkTextTheme.headline2),
553                 const SizedBox(height: 30),
554               ],
555             ),
556           ),
557           Center(
558             child: Wrap(
559               alignment: WrapAlignment.start,
560               spacing: 12,
561               runSpacing: 12,
562               children: createTagChips(),
563             ),
564           ),
565         ],
566       ),
567     );
568   );
569 }
570
571 circle_image.dart
572 import 'package:flutter/material.dart';
573 class CircleImage extends StatelessWidget {
574   const CircleImage({
```

Fig 17.

A screenshot of a Linux desktop environment showing a text editor window titled "Untitled Document 1". The window displays Dart code for a "CircleImage" widget. The code includes imports for flutter/material.dart. It defines a "CircleImage" class that extends StatelessWidget and contains logic for creating a "Wrap" of "Icon"s. The code uses various Flutter and Dart libraries like BoxDecoration, BorderRadius, and Container. The text editor interface includes tabs for "Dart", "Tab Width: 8", "Ln 19, Col 1", and "INS". The bottom of the screen shows a dock with icons for various applications.

```
525   ),
526   style: CookipediaTheme.lightTextTheme.headline1,
527   ),
528   ],
529   ],
530   ],
531   ],
532   ],
533   ],
534   ],
535   ],
536   ],
537   ],
538   ],
539   ],
540   ],
541   ],
542   ],
543   ],
544   ],
545   ],
546   ],
547   ],
548   ],
549   ],
550   ],
551   ],
552   ],
553   ],
554   ],
555   ],
556   ],
557   ],
558   ],
559   ],
560   ],
561   ],
562   ],
563   ],
564   ],
565   ],
566   ],
567   ],
568   ],
569   ],
570   ],
571   ],
572   ],
573   ],
574   ],
575   ],
576   ],
577   ],
578   ],
579   ],
580   ],
581   ],
582   ],
583   ],
584   ],
585   ],
586   ],
587   ],
588   ],
589   ],
590   ],
591   ],
592   ],
593   ],
594   ],
595   ],
596   ],
597   ],
598   ],
599   ],
600   ],
601   ],
602   ],
603   ],
604   ],
605   ],
606   ],
607   ],
608   ],
609   ],
610   ],
611   ],
612   ],
613   ],
614   ],
615   ],
616   ],
617   ],
618   ],
619   ],
620   ],
621   ],
622   ],
623   ],
624   ],
625   ],
626   ],
627   ],
628   ],
629   ],
630   ],
631   ],
632   ],
633   ],
634   ],
635   ],
636   ],
637   ],
638   ],
639   ],
640   ],
641   ],
642   ],
643   ],
644   ],
645   ],
646   ],
647   ],
648   ],
649   ],
650   ],
651   ],
652   ],
653   ],
654   ],
655   ],
656   ],
657   ],
658   ],
659   ],
660   ],
661   ],
662   ],
663   ],
664   ],
665   ],
666   ],
667   ],
668   ],
669   ],
670   ],
671   ],
672   ],
673   ],
674   ],
675   ],
676   ],
677   ],
678   ],
679   ],
680   ],
681   ],
682   ],
683   ],
684   ],
685   ],
686   ],
687   ],
688   ],
689   ],
690   ],
691   ],
692   ],
693   ],
694   ],
695   ],
696   ],
697   ],
698   ],
699   ],
700   ],
701   ],
702   ],
703   ],
704   ],
705   ],
706   ],
707   ],
708   ],
709   ],
710   ],
711   ],
712   ],
713   ],
714   ],
715   ],
716   ],
717   ],
718   ],
719   ],
720   ],
721   ],
722   ],
723   ],
724   ],
725   ],
726   ],
727   ],
728   ],
729   ],
730   ],
731   ],
732   ],
733   ],
734   ],
735   ],
736   ],
737   ],
738   ],
739   ],
740   ],
741   ],
742   ],
743   ],
744   ],
745   ],
746   ],
747   ],
748   ],
749   ],
750   ],
751   ],
752   ],
753   ],
754   ],
755   ],
756   ],
757   ],
758   ],
759   ],
760   ],
761   ],
762   ],
763   ],
764   ],
765   ],
766   ],
767   ],
768   ],
769   ],
770   ],
771   ],
772   ],
773   ],
774   ],
775   ],
776   ],
777   ],
778   ],
779   ],
780   ],
781   ],
782   ],
783   ],
784   ],
785   ],
786   ],
787   ],
788   ],
789   ],
790   ],
791   ],
792   ],
793   ],
794   ],
795   ],
796   ],
797   ],
798   ],
799   ],
800   ],
801   ],
802   ],
803   ],
804   ],
805   ],
806   ],
807   ],
808   ],
809   ],
810   ],
811   ],
812   ],
813   ],
814   ],
815   ],
816   ],
817   ],
818   ],
819   ],
820   ],
821   ],
822   ],
823   ],
824   ],
825   ],
826   ],
827   ],
828   ],
829   ],
830   ],
831   ],
832   ],
833   ],
834   ],
835   ],
836   ],
837   ],
838   ],
839   ],
840   ],
841   ],
842   ],
843   ],
844   ],
845   ],
846   ],
847   ],
848   ],
849   ],
850   ],
851   ],
852   ],
853   ],
854   ],
855   ],
856   ],
857   ],
858   ],
859   ],
860   ],
861   ],
862   ],
863   ],
864   ],
865   ],
866   ],
867   ],
868   ],
869   ],
870   ],
871   ],
872   ],
873   ],
874   ],
875   ],
876   ],
877   ],
878   ],
879   ],
880   ],
881   ],
882   ],
883   ],
884   ],
885   ],
886   ],
887   ],
888   ],
889   ],
890   ],
891   ],
892   ],
893   ],
894   ],
895   ],
896   ],
897   ],
898   ],
899   ],
900   ],
901   ],
902   ],
903   ],
904   ],
905   ],
906   ],
907   ],
908   ],
909   ],
910   ],
911   ],
912   ],
913   ],
914   ],
915   ],
916   ],
917   ],
918   ],
919   ],
920   ],
921   ],
922   ],
923   ],
924   ],
925   ],
926   ],
927   ],
928   ],
929   ],
930   ],
931   ],
932   ],
933   ],
934   ],
935   ],
936   ],
937   ],
938   ],
939   ],
940   ],
941   ],
942   ],
943   ],
944   ],
945   ],
946   ],
947   ],
948   ],
949   ],
950   ],
951   ],
952   ],
953   ],
954   ],
955   ],
956   ],
957   ],
958   ],
959   ],
960   ],
961   ],
962   ],
963   ],
964   ],
965   ],
966   ],
967   ],
968   ],
969   ],
970   ],
971   ],
972   ],
973   ],
974   ],
975   ],
976   ],
977   ],
978   ],
979   ],
980   ],
981   ],
982   ],
983   ],
984   ],
985   ],
986   ],
987   ],
988   ],
989   ],
990   ],
991   ],
992   ],
993   ],
994   ],
995   ],
996   ],
997   ],
998   ],
999   ],
1000   ],
1001   ],
1002   ],
1003   ],
1004   ],
1005   ],
1006   ],
1007   ],
1008   ],
1009   ],
1010   ],
1011   ],
1012   ],
1013   ],
1014   ],
1015   ],
1016   ],
1017   ],
1018   ],
1019   ],
1020   ],
1021   ],
1022   ],
1023   ],
1024   ],
1025   ],
1026   ],
1027   ],
1028   ],
1029   ],
1030   ],
1031   ],
1032   ],
1033   ],
1034   ],
1035   ],
1036   ],
1037   ],
1038   ],
1039   ],
1040   ],
1041   ],
1042   ],
1043   ],
1044   ],
1045   ],
1046   ],
1047   ],
1048   ],
1049   ],
1050   ],
1051   ],
1052   ],
1053   ],
1054   ],
1055   ],
1056   ],
1057   ],
1058   ],
1059   ],
1060   ],
1061   ],
1062   ],
1063   ],
1064   ],
1065   ],
1066   ],
1067   ],
1068   ],
1069   ],
1070   ],
1071   ],
1072   ],
1073   ],
1074   ],
1075   ],
1076   ],
1077   ],
1078   ],
1079   ],
1080   ],
1081   ],
1082   ],
1083   ],
1084   ],
1085   ],
1086   ],
1087   ],
1088   ],
1089   ],
1090   ],
1091   ],
1092   ],
1093   ],
1094   ],
1095   ],
1096   ],
1097   ],
1098   ],
1099   ],
1100   ],
1101   ],
1102   ],
1103   ],
1104   ],
1105   ],
1106   ],
1107   ],
1108   ],
1109   ],
1110   ],
1111   ],
1112   ],
1113   ],
1114   ],
1115   ],
1116   ],
1117   ],
1118   ],
1119   ],
1120   ],
1121   ],
1122   ],
1123   ],
1124   ],
1125   ],
1126   ],
1127   ],
1128   ],
1129   ],
1130   ],
1131   ],
1132   ],
1133   ],
1134   ],
1135   ],
1136   ],
1137   ],
1138   ],
1139   ],
1140   ],
1141   ],
1142   ],
1143   ],
1144   ],
1145   ],
1146   ],
1147   ],
1148   ],
1149   ],
1150   ],
1151   ],
1152   ],
1153   ],
1154   ],
1155   ],
1156   ],
1157   ],
1158   ],
1159   ],
1160   ],
1161   ],
1162   ],
1163   ],
1164   ],
1165   ],
1166   ],
1167   ],
1168   ],
1169   ],
1170   ],
1171   ],
1172   ],
1173   ],
1174   ],
1175   ],
1176   ],
1177   ],
1178   ],
1179   ],
1180   ],
1181   ],
1182   ],
1183   ],
1184   ],
1185   ],
1186   ],
1187   ],
1188   ],
1189   ],
1190   ],
1191   ],
1192   ],
1193   ],
1194   ],
1195   ],
1196   ],
1197   ],
1198   ],
1199   ],
1200   ],
1201   ],
1202   ],
1203   ],
1204   ],
1205   ],
1206   ],
1207   ],
1208   ],
1209   ],
1210   ],
1211   ],
1212   ],
1213   ],
1214   ],
1215   ],
1216   ],
1217   ],
1218   ],
1219   ],
1220   ],
1221   ],
1222   ],
1223   ],
1224   ],
1225   ],
1226   ],
1227   ],
1228   ],
1229   ],
1230   ],
1231   ],
1232   ],
1233   ],
1234   ],
1235   ],
1236   ],
1237   ],
1238   ],
1239   ],
1240   ],
1241   ],
1242   ],
1243   ],
1244   ],
1245   ],
1246   ],
1247   ],
1248   ],
1249   ],
1250   ],
1251   ],
1252   ],
1253   ],
1254   ],
1255   ],
1256   ],
1257   ],
1258   ],
1259   ],
1260   ],
1261   ],
1262   ],
1263   ],
1264   ],
1265   ],
1266   ],
1267   ],
1268   ],
1269   ],
1270   ],
1271   ],
1272   ],
1273   ],
1274   ],
1275   ],
1276   ],
1277   ],
1278   ],
1279   ],
1280   ],
1281   ],
1282   ],
1283   ],
1284   ],
1285   ],
1286   ],
1287   ],
1288   ],
1289   ],
1290   ],
1291   ],
1292   ],
1293   ],
1294   ],
1295   ],
1296   ],
1297   ],
1298   ],
1299   ],
1300   ],
1301   ],
1302   ],
1303   ],
1304   ],
1305   ],
1306   ],
1307   ],
1308   ],
1309   ],
1310   ],
1311   ],
1312   ],
1313   ],
1314   ],
1315   ],
1316   ],
1317   ],
1318   ],
1319   ],
1320   ],
1321   ],
1322   ],
1323   ],
1324   ],
1325   ],
1326   ],
1327   ],
1328   ],
1329   ],
1330   ],
1331   ],
1332   ],
1333   ],
1334   ],
1335   ],
1336   ],
1337   ],
1338   ],
1339   ],
1340   ],
1341   ],
1342   ],
1343   ],
1344   ],
1345   ],
1346   ],
1347   ],
1348   ],
1349   ],
1350   ],
1351   ],
1352   ],
1353   ],
1354   ],
1355   ],
1356   ],
1357   ],
1358   ],
1359   ],
1360   ],
1361   ],
1362   ],
1363   ],
1364   ],
1365   ],
1366   ],
1367   ],
1368   ],
1369   ],
1370   ],
1371   ],
1372   ],
1373   ],
1374   ],
1375   ],
1376   ],
1377   ],
1378   ],
1379   ],
1380   ],
1381   ],
1382   ],
1383   ],
1384   ],
1385   ],
1386   ],
1387   ],
1388   ],
1389   ],
1390   ],
1391   ],
1392   ],
1393   ],
1394   ],
1395   ],
1396   ],
1397   ],
1398   ],
1399   ],
1400   ],
1401   ],
1402   ],
1403   ],
1404   ],
1405   ],
1406   ],
1407   ],
1408   ],
1409   ],
1410   ],
1411   ],
1412   ],
1413   ],
1414   ],
1415   ],
1416   ],
1417   ],
1418   ],
1419   ],
1420   ],
1421   ],
1422   ],
1423   ],
1424   ],
1425   ],
1426   ],
1427   ],
1428   ],
1429   ],
1430   ],
1431   ],
1432   ],
1433   ],
1434   ],
1435   ],
1436   ],
1437   ],
1438   ],
1439   ],
1440   ],
1441   ],
1442   ],
1443   ],
1444   ],
1445   ],
1446   ],
1447   ],
1448   ],
1449   ],
1450   ],
1451   ],
1452   ],
1453   ],
1454   ],
1455   ],
1456   ],
1457   ],
1458   ],
1459   ],
1460   ],
1461   ],
1462   ],
1463   ],
1464   ],
1465   ],
1466   ],
1467   ],
1468   ],
1469   ],
1470   ],
1471   ],
1472   ],
1473   ],
1474   ],
1475   ],
1476   ],
1477   ],
1478   ],
1479   ],
1480   ],
1481   ],
1482   ],
1483   ],
1484   ],
1485   ],
1486   ],
1487   ],
1488   ],
1489   ],
1490   ],
1491   ],
1492   ],
1493   ],
1494   ],
1495   ],
1496   ],
1497   ],
1498   ],
1499   ],
1500   ],
1501   ],
1502   ],
1503   ],
1504   ],
1505   ],
1506   ],
1507   ],
1508   ],
1509   ],
1510   ],
1511   ],
1512   ],
1513   ],
1514   ],
1515   ],
1516   ],
1517   ],
1518   ],
1519   ],
1520   ],
1521   ],
1522   ],
1523   ],
1524   ],
1525   ],
1526   ],
1527   ],
1528   ],
1529   ],
1530   ],
1531   ],
1532   ],
1533   ],
1534   ],
1535   ],
1536   ],
1537   ],
1538   ],
1539   ],
1540   ],
1541   ],
1542   ],
1543   ],
1544   ],
1545   ],
1546   ],
1547   ],
1548   ],
1549   ],
1550   ],
1551   ],
1552   ],
1553   ],
1554   ],
1555   ],
1556   ],
1557   ],
1558   ],
1559   ],
1560   ],
1561   ],
1562   ],
1563   ],
1564   ],
1565   ],
1566   ],
1567   ],
1568   ],
1569   ],
1570   ],
1571   ],
1572   ],
1573   ],
1574   ],
1575   ],
1576   ],
1577   ],
1578   ],
1579   ],
1580   ],
1581   ],
1582   ],
1583   ],
1584   ],
1585   ],
1586   ],
1587   ],
1588   ],
1589   ],
1590   ],
1591   ],
1592   ],
1593   ],
1594   ],
1595   ],
1596   ],
1597   ],
1598   ],
1599   ],
1599   ],
1600   ],
1601   ],
1602   ],
1603   ],
1604   ],
1605   ],
1606   ],
1607   ],
1608   ],
1609   ],
1610   ],
1611   ],
1612   ],
1613   ],
1614   ],
1615   ],
1616   ],
1617   ],
1618   ],
1619   ],
1620   ],
1621   ],
1622   ],
1623   ],
1624   ],
1625   ],
1626   ],
1627   ],
1628   ],
1629   ],
1630   ],
1631   ],
1632   ],
1633   ],
1634   ],
1635   ],
1636   ],
1637   ],
1638   ],
1639   ],
1640   ],
1641   ],
1642   ],
1643   ],
1644   ],
1645   ],
1646   ],
1647   ],
1648   ],
1649   ],
1650   ],
1651   ],
1652   ],
1653   ],
1654   ],
1655   ],
1656   ],
1657   ],
1658   ],
1659   ],
1660   ],
1661   ],
1662   ],
1663   ],
1664   ],
1665   ],
1666   ],
1667   ],
1668   ],
1669   ],
1670   ],
1671   ],
1672   ],
1673   ],
1674   ],
1675   ],
1676   ],
1677   ],
1678   ],
1679   ],
1680   ],
1681   ],
1682   ],
1683   ],
1684   ],
1685   ],
1686   ],
1687   ],
1688   ],
1689   ],
1690   ],
1691   ],
1692   ],
1693   ],
1694   ],
1695   ],
1696   ],
1697   ],
1698   ],
1699   ],
1700   ],
1701   ],
1702   ],
1703   ],
1704   ],
1705   ],
1706   ],
1707   ],
1708   ],
1709   ],
1710   ],
1711   ],
1712   ],
1713   ],
1714   ],
1715   ],
1716   ],
1717   ],
1718   ],
1719   ],
1720   ],
1721   ],
1722   ],
1723   ],
1724   ],
1725   ],
1726   ],
1727   ],
1728   ],
1729   ],
1730   ],
1731   ],
1732   ],
1733   ],
1734   ],
1735   ],
1736   ],
1737   ],
1738   ],
1739   ],
1740   ],
1741   ],
1742   ],
1743   ],
1744   ],
1745   ],
1746   ],
1747   ],
1748   ],
1749   ],
1750   ],
1751   ],
1752   ],
1753   ],
1754   ],
1755   ],
1756   ],
1757   ],
1758   ],
1759   ],
1760   ],
1761   ],
1762   ],
1763   ],
1764   ],
1765   ],
1766   ],
1767   ],
1768   ],
1769   ],
1770   ],
1771   ],
1772   ],
1773   ],
1774   ],
1775   ],
1776   ],
1777   ],
1778   ],
1779   ],
1780   ],
1781   ],
1782   ],
1783   ],
1784   ],
1785   ],
1786   ],
1787   ],
1788   ],
1789   ],
1790   ],
1791   ],
1792   ],
1793   ],
1794   ],
1795   ],
1796   ],
1797   ],
1798   ],
1799   ],
1800   ],
1801   ],
1802   ],
1803   ],
1804   ],
1805   ],
1806   ],
1807   ],
1808   ],
1809   ],
1810   ],
1811   ],
1812   ],
1813   ],
1814   ],
1815   ],
1816   ],
1817   ],
1818   ],
1819   ],
1820   ],
1821   ],
1822   ],
1823   ],
1824   ],
1825   ],
1826   ],
1827   ],
1828   ],
1829   ],
1830   ],
1831   ],
1832   ],
1833   ],
1834   ],
1835   ],
1836   ],
1837   ],
1838   ],
1839   ],
1840   ],
1841   ],
1842   ],
1843   ],
1844   ],
1845   ],
1846   ],
1847   ],
1848   ],
1849   ],
1850   ],
1851   ],
1852   ],
1853   ],
1854   ],
1855   ],
1856   ],
1857   ],
1858   ],
1859   ],
1860   ],
1861   ],
1862   ],
1863   ],
1864   ],
1865   ],
1866   ],
1867   ],
1868   ],
1869   ],
1870   ],
1871   ],
1872   ],
1873   ],
1874   ],
1875   ],
1876   ],
1877   ],
1878   ],
1879   ],
1880   ],
1881   ],
1882   ],
1883   ],
1884   ],
1885   ],
1886   ],
1887   ],
1888   ],
1889   ],
1890   ],
1891   ],
1892   ],
1893   ],
1894   ],
1895   ],
1896   ],
1897   ],
1898   ],
1899   ],
1900   ],
1901   ],
1902   ],
1903   ],
1904   ],
1905   ],
1906   ],
1907   ],
1908   ],
1909   ],
1910   ],
1911   ],
1912   ],
1913   ],
1914   ],
1915   ],
1916   ],
1917   ],
1918   ],
1919   ],
1920   ],
1921   ],
1922   ],
1923   ],
1924   ],
1925   ],
1926   ],
1927   ],
1928   ],
1929   ],
1930   ],
1931   ],
1932   ],
1933   ],
1934   ],
1935   ],
1936   ],
1937   ],
1938   ],
1939   ],
1940   ],
1941   ],
1942   ],
1943   ],
1944   ],
1945   ],
1946   ],
1947   ],
1948   ],
1949   ],
1950   ],
1951   ],
1952   ],
1953   ],
1954   ],
1955   ],
1956   ],
1957   ],
1958   ],
1959   ],
1960   ],
1961   ],
1962   ],
1963   ],
1964   ],
1965   ],
1966   ],
1967   ],
1968   ],
1969   ],
1970   ],
1971   ],
1972   ],
1973   ],
1974   ],
1975   ],
1976   ],
1977   ],
1978   ],
1979   ],
1980   ],
1981   ],
1982   ],
1983   ],
1984   ],
1985   ],
1986   ],
1987   ],
1988   ],
1989   ],
1990   ],
1991   ],
1992   ],
1993   ],
1994   ],
1995   ],
1996   ],
1997   ],
1998   ],
1999   ],
1999   ],
2000   ],
2001   ],
2002   ],
2003   ],
2004   ],
2005   ],
2006   ],
2007   ],
2008   ],
2009   ],
2010   ],
2011   ],
2012   ],
2013   ],
2014   ],
2015   ],
2016   ],
2017   ],
2018   ],
2019   ],
2020   ],
2021   ],
2022   ],
2023   ],
2024   ],
2025   ],
2026   ],
2027   ],
2028   ],
2029   ],
2030   ],
2031   ],
2032   ],
2033   ],
2034   ],
2035   ],
2036   ],
2037   ],
2038   ],
2039   ],
2040   ],
2041   ],
2042   ],
2043   ],
2044   ],
2045   ],
2046   ],
2047   ],
2048   ],
2049   ],
2050   ],
2051   ],
2052   ],
2053   ],
2054   ],
2055   ],
2056   ],
2057   ],
2058   ],
2059   ],
2060   ],
2061   ],
2062   ],
2063   ],
2064   ],
2065   ],
2066   ],
2067   ],
2068   ],
2069   ],
2070   ],
2071   ],
2072   ],
2073   ],
2074   ],
2075   ],
2076   ],
2077   ],
2078   ],
2079   ],
2080   ],
2081   ],
2082   ],
2083   ],
2084   ],
2085   ],
2086   ],
2087   ],
2088   ],
2089   ],
2089   ],
2090   ],
2091   ],
2092   ],
2093   ],
2094   ],
2095   ],
2096   ],
2097   ],
2098   ],
2099   ],
2099   ],
2100   ],
2101   ],
2102   ],
2103   ],
2104   ],
2105   ],
2106   ],
2107   ],
2108   ],
2109   ],
2110   ],
2111   ],
2112   ],
2113   ],
2114   ],
2115   ],
2116   ],
2117   ],
2118   ],
2119   ],
2120   ],
2121   ],
2122   ],
2123   ],
2124   ],
2125   ],
2126   ],
2127   ],
2128   ],
2129   ],
2129   ],
2130   ],
2131   ],
2132   ],
2133   ],
2134   ],
2135   ],
2136   ],
2137   ],
2138   ],
2139   ],
2139   ],
2140   ],
2141   ],
2142   ],
2143   ],
2144   ],
2145   ],
2146   ],
2147   ],
2148   ],
2149   ],
2149   ],
2150   ],
2151   ],
2152   ],
2153   ],
2154   ],
2155   ],
2156   ],
2157   ],
2158   ],
2159   ],
2159   ],
2160   ],
2161   ],
2162   ],
2163   ],
2164   ],
2165   ],
2166   ],
2167   ],
2168   ],
2169   ],
2169   ],
2170   ],
2171   ],
2172   ],
2173   ],
2174   ],
2175   ],
2176   ],
2177   ],
2178   ],
2179   ],
2179   ],
2180   ],
2181   ],
2182   ],
2183   ],
2184   ],
2185   ],
2186   ],
2187   ],
2188   ],
2189   ],
2189   ],
2190   ],
2191   ],
2192   ],
2193   ],
2194   ],
2195   ],
2196   ],
2197   ],
2198   ],
2199   ],
2199   ],
2200   ],
2201   ],
2202   ],
2203   ],
2204   ],
2205   ],
2206   ],
2207   ],
2208   ],
2209   ],
2210   ],
2211   ],
2212   ],
2213   ],
2214   ],
2215   ],
2216   ],
2217   ],
2218   ],
2219   ],
2220   ],
2221   ],
2222   ],
2223   ],
2224   ],
2225   ],
2226   ],
2227   ],
2228   ],
2229   ],
2229   ],
2230   ],
2231   ],
2232   ],
2233   ],
2234   ],
2235   ],
2236   ],
2237   ],
2238   ],
2239   ],
2239   ],
2240   ],
2241   ],
2242   ],
2243   ],
2244   ],
2245   ],
2246
```

A screenshot of a Flutter IDE interface. The main area displays the Dart code for the `FriendPostListView` class. The code includes imports for various components and models, and defines a `StatelessWidget` that builds a `Column` with a `Text` header and a `ListView.separated` for friend posts. The code uses `FriendPostTile` as the item builder. The bottom of the screen shows a toolbar with icons for file operations, navigation, and other development tools.

```
578     child: CircleAvatar(
579       imageRadius = 20,
580     )) : super(key: key);
581   final double imageRadius;
582   final ImageProvider? imageProvider;
583 
584   @override
585   Widget build(BuildContext context) {
586     // ...
587     return CircleAvatar(
588       backgroundColor: Colors.white,
589       radius: imageRadius,
590     );
591     child: CircleAvatar(
592       radius: imageRadius - 5,
593       backgroundImage: imageProvider,
594     ),
595   );
596 }
597 }
598 import 'components.dart';
599 export 'author_card.dart';
600 export 'card1.dart';
601 export 'card2.dart';
602 export 'card3.dart';
603 export 'circle_image.dart';
604 export 'today_recipe_list_view.dart';
605 export 'friend_post_tile.dart';
606 export 'friend_post_list_view.dart';
607 export 'recipe_thumbnail.dart';
608 export 'recipes_grid_view.dart';
609 export 'today_recipe_list_view.dart';
610
611 friend_post_list_view.dart
612 import 'package:flutter/material.dart';
613 import '../models/models.dart';
614 import 'components.dart';
615
616 class FriendPostListView extends StatelessWidget {
617   // ...
618   final List<Post> friendPosts;
619   const FriendPostListView({
620     Key? key,
621     required this.friendPosts,
622   }) : super(key: key);
623 
624   @override
625   Widget build(BuildContext context) {
626     // ...
627     padding: const EdgeInsets.only(
628       left: 16,
629       right: 16,
630       top: 0,
631     ),
632     child: Column(
633       crossAxisAlignment: CrossAxisAlignment.start,
634       children: [
635         // ...
636         Text('Social Chefs ! ', style: Theme.of(context).textTheme.headline1),
637         // ...
638         const SizedBox(height: 16),
639         // TODO: Add PostListTile here
640         ListView.separated(
641           // ...
642           primary: false,
643           // ...
644           physics: const NeverScrollableScrollPhysics(),
645           // ...
646           shrinkWrap: true,
647           scrollDirection: Axis.vertical,
648           itemCount: friendPosts.length,
649           itemBuilder: (context, index) {
650             // ...
651             final post = friendPosts[index];
652             return FriendPostTitle(post: post);
653           },
654           // ...
655           separatorBuilder: (context, index) {
656             // ...
657             return const SizedBox(height: 16);
658           },
659           // ...
660         ),
661       ],
662     ),
663     const SizedBox(height: 16),
664   ],
665 ),
666 );
667 );
668 ]
669 }
670 friend_post_tile.dart
671 import 'package:flutter/material.dart';
672 import '../components/components.dart';
673 import '../models/models.dart';
674
675 class FriendPostTitle extends StatelessWidget {
676   final Post post;
677   const FriendPostTitle({
678     Key? key,
679   }) : super(key: key);
```

Fig 19.

A screenshot of a Flutter IDE interface. The main area displays the Dart code for the `FriendPostTitle` class. The code defines a `StatelessWidget` that takes a `Post` object and returns a `CircleAvatar` with a `radius` of 20 and a `backgroundImage` from the `post`. The bottom of the screen shows a toolbar with icons for file operations, navigation, and other development tools.

```
578     child: CircleAvatar(
579       imageRadius = 20,
580     )) : super(key: key);
581   final double imageRadius;
582   final ImageProvider? imageProvider;
583 
584   @override
585   Widget build(BuildContext context) {
586     // ...
587     return CircleAvatar(
588       backgroundColor: Colors.white,
589       radius: imageRadius,
590     );
591     child: CircleAvatar(
592       radius: imageRadius - 5,
593       backgroundImage: imageProvider,
594     ),
595   );
596 }
597 }
598 import 'components.dart';
599 export 'author_card.dart';
600 export 'card1.dart';
601 export 'card2.dart';
602 export 'card3.dart';
603 export 'circle_image.dart';
604 export 'today_recipe_list_view.dart';
605 export 'friend_post_tile.dart';
606 export 'friend_post_list_view.dart';
607 export 'recipe_thumbnail.dart';
608 export 'recipes_grid_view.dart';
609 export 'today_recipe_list_view.dart';
610
611 friend_post_list_view.dart
612 import 'package:flutter/material.dart';
613 import '../models/models.dart';
614 import 'components.dart';
615
616 class FriendPostTitle extends StatelessWidget {
617   final Post post;
618   const FriendPostTitle({
619     Key? key,
620   }) : super(key: key);
```

Fig 20.

A screenshot of a Linux desktop environment showing a code editor window. The window title is "Text Editor" and the document title is "*Untitled Document 1". The code is written in Dart and defines a StatelessWidget named GroceryTile. It includes imports for flutter/material.dart, google_fonts/google_fonts.dart, intl/intl.dart, and cookipedia/models/grocery_item.dart. The class contains a constructor taking a key and a required item parameter, and an onComplete callback. The build method returns a Row with two children: a CircleImage and a Column with a Text for the comment and a Text for the timestamp.

```
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open ▾
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open ▾
735
736 // 4
737 GroceryTile({
738   Key? key,
739   required this.item,
740   this.onComplete,
741 }: textDecoration =
742   item.isIncomplete ? TextDecoration.lineThrough : TextDecoration.none,
743   super(key: key);
744
745 @override
746 Widget build(BuildContext context) {
747 // TODO: Change this Widget
748   return SizedBox(
749     height: 106.0,
750
751     child: Row(
752       mainAxisAlignment: MainAxisAlignment.spaceBetween,
753       children: [
754         // 2
755         Container(
756           width: 5.0,
757           color: item.color,
758         ),
759       ],
760     );
761   );
762 }
763
764 Column(
765   mainAxisAlignment: MainAxisAlignment.center,
766   crossAxisAlignment: CrossAxisAlignment.start,
767   children: [
768     Text(
769       item.name,
770       style: GoogleFonts.lato(
771         decoration: textDecoration,
772         fontSize: 21.0,
773         fontWeight: FontWeight.bold),
774     ),
775     const SizedBox(height: 4.0),
776     buildDate(),
777     const SizedBox(height: 4.0),
778     buildImportance(),
779   ],
780 ),
781 Row(
782   children: [
783     Text(
784 // 7
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2
```

A screenshot of a Linux desktop environment. In the foreground, a terminal window titled "Text Editor" displays a large amount of Dart code. The code includes methods for building importance levels ("Widget buildImportance") and checkboxes ("Widget buildCheckbox"). It uses Google Fonts Lato and various font weights (bold, medium, low, high) and sizes (21.0). The code editor interface shows line numbers from 761 to 837. The background shows a dock with icons for various applications like a file manager, browser, and system tools. The top bar has standard window controls and a date/time indicator.

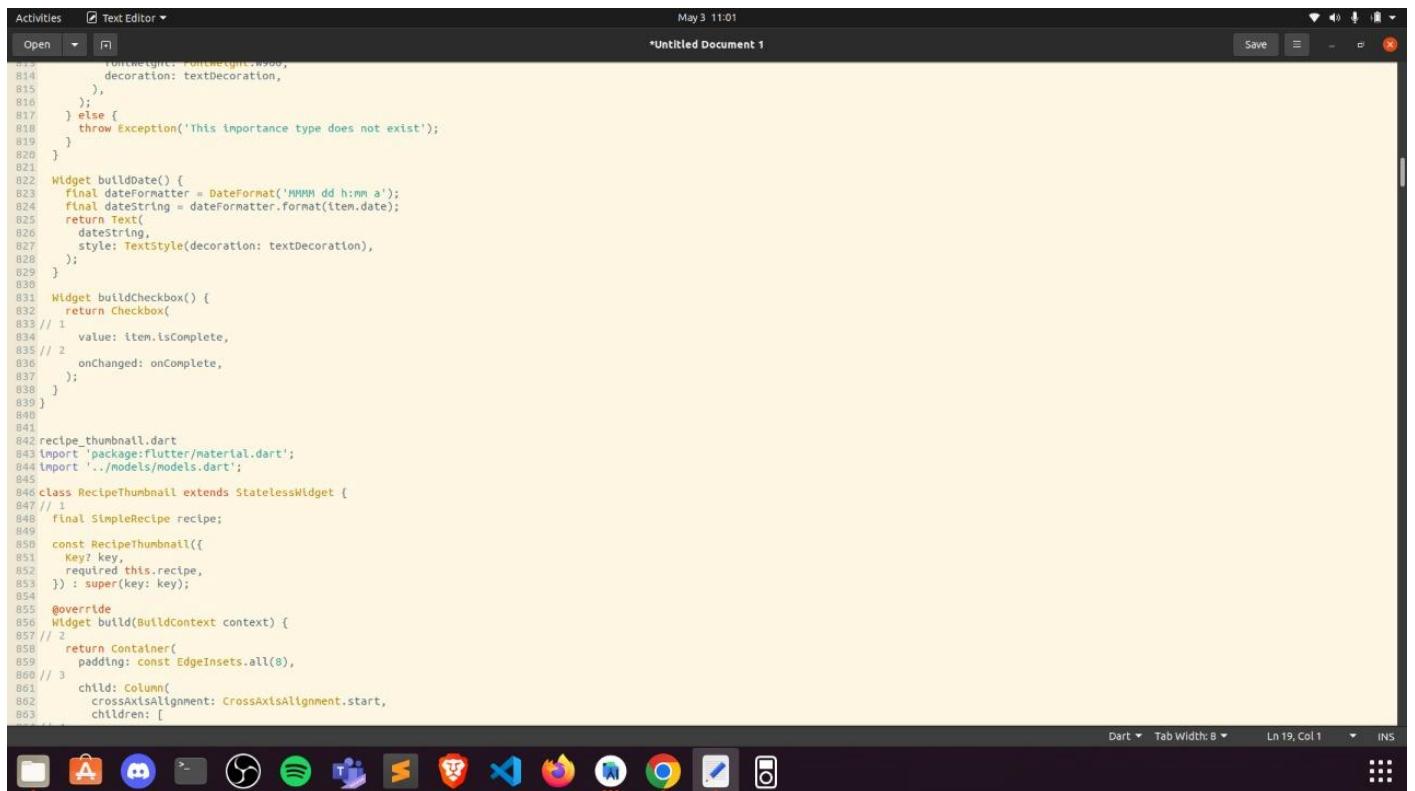
```
May 3 11:01
*Untitled Document 1
Save
Dart Tab Width: 8 Ln 19, Col 1 INS
Activities Text Editor Open □ May 3 11:01 *Untitled Document 1 Save Dart Tab Width: 8 Ln 19, Col 1 INS
761
762 // 4
763     Column(
764       mainAxisAlignment: MainAxisAlignment.center,
765       crossAxisAlignment: CrossAxisAlignment.start,
766       children: [
767 // 5
768         Text(
769           item.name,
770           style: GoogleFonts.lato(
771             decoration: textDecoration,
772             fontSize: 21.0,
773             fontWeight: FontWeight.bold),
774         ),
775         const SizedBox(height: 4.0),
776         buildDate(),
777         const SizedBox(height: 4.0),
778         buildImportance(),
779       ],
780     ),
781     Row(
782       children: [
783 // 7
784         Text(
785           item.quantity.toString(),
786           style: GoogleFonts.lato(
787             decoration: textDecoration,
788             fontSize: 21.0,
789           ),
790         ),
791       ],
792 // 8
793       buildCheckbox(),
794     ],
795   ],
796 },
797 ),
798 );
799
800 Widget buildImportance() {
801   if (item.importance == Importance.low) {
802     return Text('Low', style: GoogleFonts.lato(decoration: textDecoration));
803   } else if (item.importance == Importance.medium) {
804     return Text('Medium',
805       style: GoogleFonts.lato(
806         fontWeight: FontWeight.w800, decoration: textDecoration));
807   } else if (item.importance == Importance.high) {
808     return Text('High',
809       style: GoogleFonts.lato(
810         color: Colors.red,
811         fontWeight: FontWeight.w900,
812         decoration: textDecoration,
813       ),
814     );
815   } else {
816     throw Exception('This importance type does not exist');
817   }
818 }
819
820
821 Widget buildDate() {
822   final dateFormatter = DateFormat('MMMM dd h:mm a');
823   final dateString = dateFormatter.format(item.date);
824   return Text(
825     dateString,
826     style: TextStyle(decoration: textDecoration),
827   );
828 }
829
830 Widget buildCheckbox() {
831   return Checkbox(
832 // 1
833     value: item.isComplete,
834 // 2
835     onChanged: onComplete,
836   );
837 }
```

Fig 23.

A screenshot of a Linux desktop environment, similar to Fig 23. A terminal window titled "Text Editor" displays a large amount of Dart code. This version of the code is more complex, involving nested columns and rows to build a UI structure. It includes methods for building importance levels and checkboxes, using Google Fonts Lato and specific font weights and sizes. The code editor interface shows line numbers from 761 to 837. The background dock and top bar are identical to Fig 23.

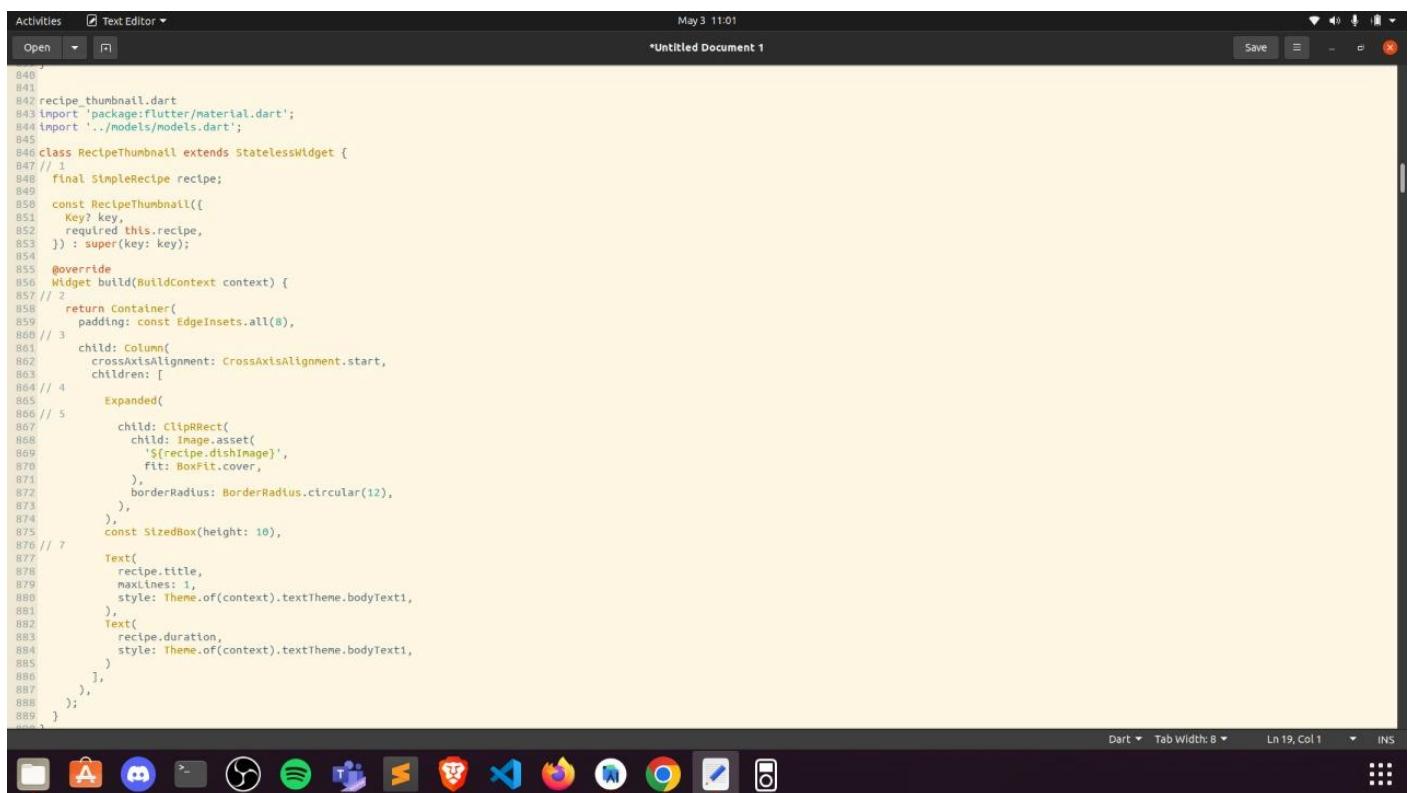
```
May 3 11:01
*Untitled Document 1
Save
Dart Tab Width: 8 Ln 19, Col 1 INS
Activities Text Editor Open □ May 3 11:01 *Untitled Document 1 Save Dart Tab Width: 8 Ln 19, Col 1 INS
761
762 // 4
763     Column(
764       mainAxisAlignment: MainAxisAlignment.center,
765       crossAxisAlignment: CrossAxisAlignment.start,
766       children: [
767 // 5
768         Text(
769           item.name,
770           style: GoogleFonts.lato(
771             decoration: textDecoration,
772             fontSize: 21.0,
773             fontWeight: FontWeight.bold),
774         ),
775         const SizedBox(height: 4.0),
776         buildDate(),
777         const SizedBox(height: 4.0),
778         buildImportance(),
779       ],
780     ),
781     Row(
782       children: [
783 // 7
784         Text(
785           item.quantity.toString(),
786           style: GoogleFonts.lato(
787             decoration: textDecoration,
788             fontSize: 21.0,
789           ),
790         ),
791       ],
792 // 8
793       buildCheckbox(),
794     ],
795   ],
796 },
797 ),
798 );
799
800 Widget buildImportance() {
801   if (item.importance == Importance.low) {
802     return Text('Low', style: GoogleFonts.lato(decoration: textDecoration));
803   } else if (item.importance == Importance.medium) {
804     return Text('Medium',
805       style: GoogleFonts.lato(
806         fontWeight: FontWeight.w800, decoration: textDecoration));
807   } else if (item.importance == Importance.high) {
808     return Text('High',
809       style: GoogleFonts.lato(
810         color: Colors.red,
811         fontWeight: FontWeight.w900,
812         decoration: textDecoration,
813       ),
814     );
815   } else {
816     throw Exception('This importance type does not exist');
817   }
818 }
819
820
821 Widget buildDate() {
822   final dateFormatter = DateFormat('MMMM dd h:mm a');
823   final dateString = dateFormatter.format(item.date);
824   return Text(
825     dateString,
826     style: TextStyle(decoration: textDecoration),
827   );
828 }
829
830 Widget buildCheckbox() {
831   return Checkbox(
832 // 1
833     value: item.isComplete,
834 // 2
835     onChanged: onComplete,
836   );
837 }
```

Fig 24.



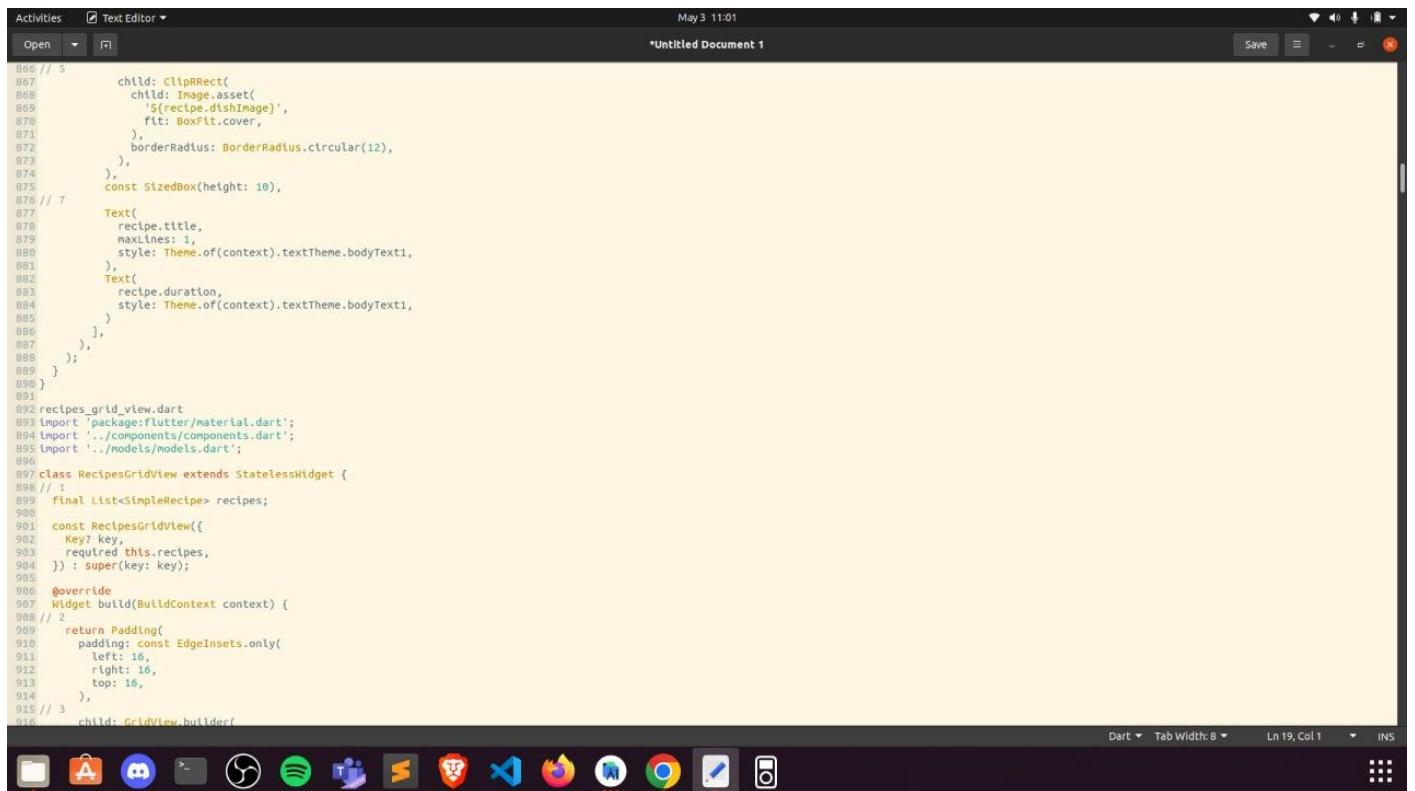
```
822     ),
823     decoration: textDecoration,
824   );
825 }
826 } else {
827   throw Exception('This importance type does not exist');
828 }
829 }
830
831 Widget buildDate() {
832   final dateFormatter = DateFormat('MMMM dd h:mm a');
833   final dateString = dateFormatter.format(item.date);
834   return Text(
835     dateString,
836     style: TextStyle(decoration: textDecoration),
837   );
838 }
839
840 Widget buildCheckbox() {
841   return Checkbox(
842     value: item.isComplete,
843     onChanged: onComplete,
844   );
845 }
846
847 recipe_thumbnail.dart
848 import 'package:flutter/material.dart';
849 import '../models/models.dart';
850
851 class RecipeThumbnail extends StatelessWidget {
852   // 1
853   final SimpleRecipe recipe;
854
855   const RecipeThumbnail({
856     Key? key,
857     required this.recipe,
858   }) : super(key: key);
859
860   @override
861   Widget build(BuildContext context) {
862     // 2
863     return Container(
864       padding: const EdgeInsets.all(8),
865     // 3
866     child: Column(
867       crossAxisAlignment: CrossAxisAlignment.start,
868       children: [
869         // 4
870         Expanded(
871           // 5
872             child: ClipRRect(
873               // 6
874               child: Image.asset(
875                 '${recipe.dishImage}',
876                 fit: BoxFit.cover,
877               ),
878               // 7
879               borderRadius: BorderRadius.circular(12),
880             ),
881         ),
882         const SizedBox(height: 10),
883         Text(
884           recipe.title,
885           maxLines: 1,
886           style: Theme.of(context).textTheme.bodyText1,
887         ),
888         Text(
889           recipe.duration,
890           style: Theme.of(context).textTheme.bodyText1,
891         ),
892       ],
893     );
894   }
895 }
```

Fig 25.



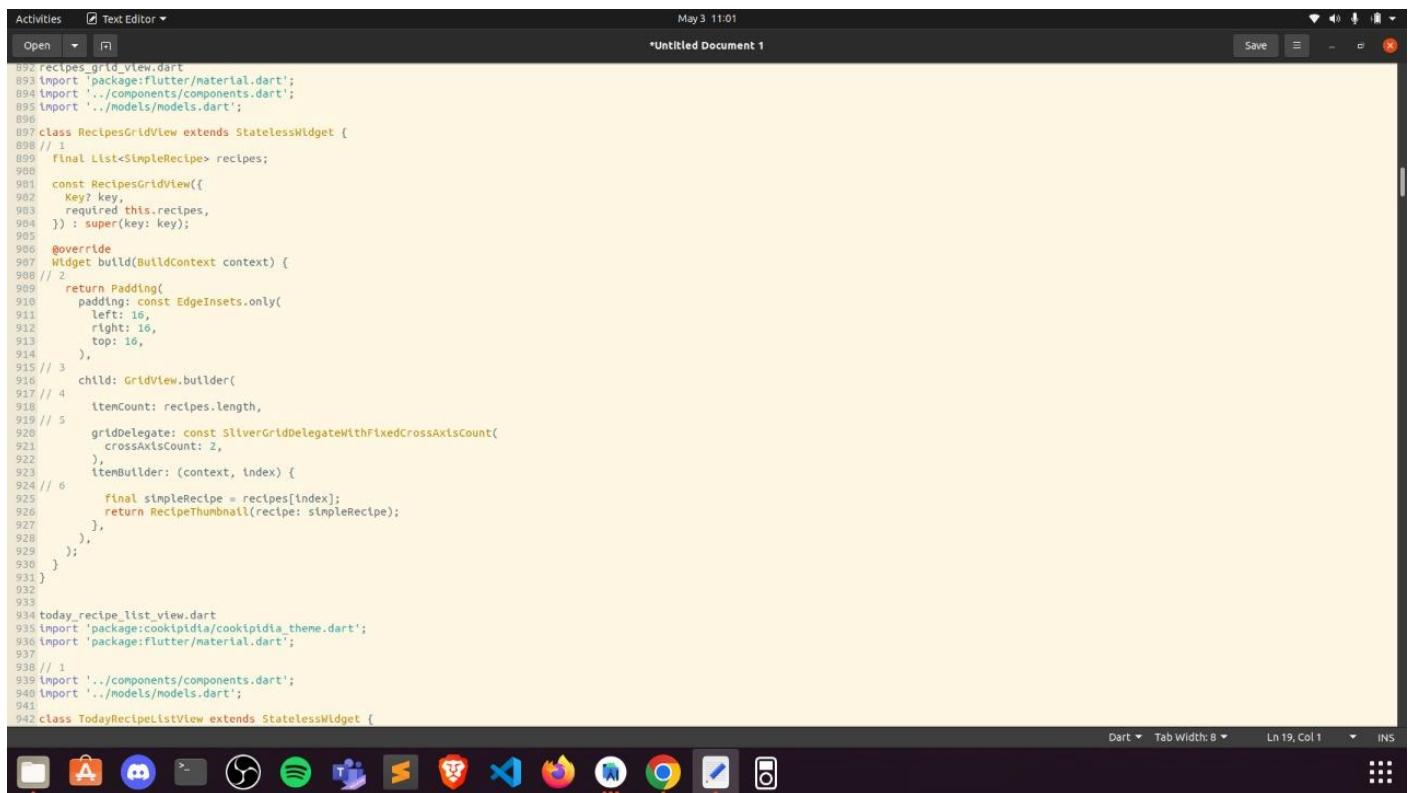
```
840
841
842 recipe_thumbnail.dart
843 import 'package:flutter/material.dart';
844 import '../models/models.dart';
845
846 class RecipeThumbnail extends StatelessWidget {
847   // 1
848   final SimpleRecipe recipe;
849
850   const RecipeThumbnail({
851     Key? key,
852     required this.recipe,
853   }) : super(key: key);
854
855   @override
856   Widget build(BuildContext context) {
857     // 2
858     return Container(
859       padding: const EdgeInsets.all(8),
860     // 3
861     child: Column(
862       crossAxisAlignment: CrossAxisAlignment.start,
863       children: [
864         // 4
865         Expanded(
866           // 5
867             child: ClipRRect(
868               // 6
869               child: Image.asset(
870                 '${recipe.dishImage}',
871                 fit: BoxFit.cover,
872               ),
873               // 7
874               borderRadius: BorderRadius.circular(12),
875             ),
876         ),
877         const SizedBox(height: 10),
878         Text(
879           recipe.title,
880           maxLines: 1,
881           style: Theme.of(context).textTheme.bodyText1,
882         ),
883         Text(
884           recipe.duration,
885           style: Theme.of(context).textTheme.bodyText1,
886         ),
887       ],
888     );
889   }
890 }
```

Fig 26.



```
Activities Text Editor
Open Untitled Document 1 Save
May 3 11:01
866 // 5
867     child: ClipRect(
868         child: Image.asset(
869             '${recipe.dishImage}',
870             fit: BoxFit.cover,
871         ),
872         borderRadius: BorderRadius.circular(12),
873     ),
874 ),
875 const SizedBox(height: 10),
876 // 7
877     Text(
878         recipe.title,
879         maxLines: 1,
880         style: Theme.of(context).textTheme.bodyText1,
881     ),
882     Text(
883         recipe.duration,
884         style: Theme.of(context).textTheme.bodyText1,
885     ),
886 ],
887 ),
888 );
889 )
890 }
891
892 recipes_grid_view.dart
893 import 'package:flutter/material.dart';
894 import '../components/components.dart';
895 import '../models/models.dart';
896
897 class RecipesGridView extends StatelessWidget {
898 // 1
899     final List<SimpleRecipe> recipes;
900
901     const RecipesGridView({
902         Key? key,
903         required this.recipes,
904     }) : super(key: key);
905
906     @override
907     Widget build(BuildContext context) {
908 // 2
909         return Padding(
910             padding: const EdgeInsets.only(
911                 left: 16,
912                 right: 16,
913                 top: 16,
914             ),
915 // 3
916             child: GridView.builder(
917             // 4
918             itemCount: recipes.length,
919             // 5
920             gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
921                 crossAxisCount: 2,
922             ),
923             itemBuilder: (context, index) {
924 // 6
925                 final simpleRecipe = recipes[index];
926                 return RecipeThumbnail(recipe: simpleRecipe);
927             },
928         );
929     );
930 }
931
932
933 today_recipe_list_view.dart
934 import 'package:cookipedia/cookipedia_theme.dart';
935 import 'package:flutter/material.dart';
936
937
938 // 1
939 import '../components/components.dart';
940 import '../models/models.dart';
941
942 class TodayRecipeListView extends StatelessWidget {
```

Fig 27.



```
Activities Text Editor
Open Untitled Document 1 Save
May 3 11:01
898 // 1
899     final List<SimpleRecipe> recipes;
900
901     const TodayRecipeListView({
902         Key? key,
903         required this.recipes,
904     }) : super(key: key);
905
906     @override
907     Widget build(BuildContext context) {
908 // 2
909         return Padding(
910             padding: const EdgeInsets.only(
911                 left: 16,
912                 right: 16,
913                 top: 16,
914             ),
915 // 3
916             child: ListView.builder(
917             // 4
918             itemCount: recipes.length,
919             // 5
920             gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
921                 crossAxisCount: 2,
922             ),
923             itemBuilder: (context, index) {
924 // 6
925                 final simpleRecipe = recipes[index];
926                 return RecipeThumbnail(recipe: simpleRecipe);
927             },
928         );
929     );
930 }
931
932
933 today_recipe_list_view.dart
934 import 'package:cookipedia/cookipedia_theme.dart';
935 import 'package:flutter/material.dart';
936
937
938 // 1
939 import '../components/components.dart';
940 import '../models/models.dart';
941
942 class TodayRecipeListView extends StatelessWidget {
```

Fig 28.

A screenshot of a Linux desktop environment. The main window is a code editor titled "Untitled Document 1" showing Dart code for a "TodayRecipeListView" widget. The code includes imports for "recipes.dart", "gridDelegateWithFixedCrossAxisCount", "itemBuilder", "simpleRecipe", "RecipeThumbnail", "List<ExploreRecipe> recipes", and "TodayRecipeListView". It defines a padding of 16 on all sides and a column with a headline2-style text "Recipes of the Day". Below this is a ListView with a scroll direction of horizontal, item count of "recipes.length", and an itemBuilder that returns a "buildCard(recipe)". A separator builder is also defined. The bottom of the screen shows a dock with various icons for applications like a file manager, terminal, browser, and others.

```
919 // 5
920     gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
921         crossAxisCount: 2,
922     ),
923     itemBuilder: (context, index) {
924 // 6
925         final simpleRecipe = recipes[index];
926         return RecipeThumbnail(recipe: simpleRecipe);
927     },
928 );
929 );
930 ]
931 ]
932 ]
933 ]
934 today_recipe_list_view.dart
935 import 'package:cookipidia/cookipidia_theme.dart';
936 import 'package:flutter/material.dart';
937
938 // 1
939 import '../components/components.dart';
940 import '../models/models.dart';
941
942 class TodayRecipeListView extends StatelessWidget {
943 // 2
944     final List<ExploreRecipe> recipes;
945
946     const TodayRecipeListView({
947         Key? key,
948         required this.recipes,
949     }) : super(key: key);
950
951     @override
952     Widget build(BuildContext context) {
953 // 3
954         return Padding(
955             padding: const EdgeInsets.only(
956                 left: 16,
957                 right: 16,
958                 top: 16,
959             ),
960 // 4
961         child: Column(
962             mainAxisAlignment: CrossAxisAlignment.start,
963             children: [
964 // 5
965             Text(
966                 'Recipes of the Day',
967                 style: Theme.of(context).textTheme.headline2,
968             ),
969         ],
970         ),
971         ),
972         padding: const EdgeInsets.only(
973             left: 16,
974             right: 16,
975             top: 16,
976         ),
977         ),
978         child: Column(
979             mainAxisAlignment: CrossAxisAlignment.start,
980             children: [
981                 Text(
982                     'Recipes of the Day',
983                     style: Theme.of(context).textTheme.headline2,
984                 ),
985                 const SizedBox(height: 16),
986                 Container(
987                     height: 400,
988                     // 1
989                     color: Colors.transparent,
990                 ),
991                 // 2
992                 child: ListView.separated(
993                     scrollDirection: Axis.horizontal,
994                     itemCount: recipes.length,
995                     itemBuilder: (context, index) {
996                         final recipe = recipes[index];
997                         return buildCard(recipe);
998                     },
999                     // 3
1000                     separatorBuilder: (context, index) {
1001                         return const SizedBox(width: 16);
1002                     },
1003                 ),
1004             ],
1005         ),
1006     );
1007 }
1008
1009 
```

Fig 29.

A screenshot of a Linux desktop environment, similar to Fig 29, showing a code editor window for the same Dart code. The code is identical to the one in Fig 29, with minor differences in line numbers and some comments removed. The bottom dock shows the same set of application icons.

```
945
946     const TodayRecipeListView({
947         Key? key,
948         required this.recipes,
949     }) : super(key: key);
950
951     @override
952     Widget build(BuildContext context) {
953 // 3
954         return Padding(
955             padding: const EdgeInsets.only(
956                 left: 16,
957                 right: 16,
958                 top: 16,
959             ),
960 // 4
961         child: Column(
962             mainAxisAlignment: CrossAxisAlignment.start,
963             children: [
964 // 5
965             Text(
966                 'Recipes of the Day',
967                 style: Theme.of(context).textTheme.headline2,
968             ),
969         ],
970         ),
971         ),
972         padding: const EdgeInsets.only(
973             left: 16,
974             right: 16,
975             top: 16,
976         ),
977         ),
978         child: Column(
979             mainAxisAlignment: CrossAxisAlignment.start,
980             children: [
981                 Text(
982                     'Recipes of the Day',
983                     style: Theme.of(context).textTheme.headline2,
984                 ),
985                 const SizedBox(height: 16),
986                 Container(
987                     height: 400,
988                     // 1
989                     color: Colors.transparent,
990                 ),
991                 // 2
992                 child: ListView.separated(
993                     scrollDirection: Axis.horizontal,
994                     itemCount: recipes.length,
995                     itemBuilder: (context, index) {
996                         final recipe = recipes[index];
997                         return buildCard(recipe);
998                     },
999                     // 3
1000                     separatorBuilder: (context, index) {
1001                         return const SizedBox(width: 16);
1002                     },
1003                 ),
1004             ],
1005         ),
1006     );
1007 }
1008
1009 
```

Fig 30.

A screenshot of a code editor window titled "Text Editor". The code is written in Dart. It includes a Container widget with a height of 400 pixels, a child ListView.separated, and a separatorBuilder. The ListView has a scrollDirection of Axis.horizontal and an itemCount of recipes.length. The itemBuilder is a function that returns a buildCard for each recipe. The buildCard function checks the cardType of the recipe and returns a Card1, Card2, or Card3 accordingly. There is also a catch-all case that throws an exception. Below this is a class AppCache that imports shared_preferences. It contains methods for invalidate, cacheUser, completeOnboarding, isUserLoggedIn, and didCompleteOnboarding. The invalidate method uses SharedPreferences to set user and onboarding booleans to false. The cacheUser method sets the user boolean to true. The completeOnboarding method sets the onboarding boolean to true. The isUserLoggedIn method checks if the user boolean is true. The didCompleteOnboarding method checks if the onboarding boolean is true. The code is annotated with line numbers from 971 to 1010.

```
971 // 7
972 Container(
973   height: 400,
974   // 1
975   color: Colors.transparent,
976 // 2
977   child: ListView.separated(
978 // 3
979     scrollDirection: Axis.horizontal,
980     itemCount: recipes.length,
981     itemBuilder: (context, index) {
982 // 4
983       final recipe = recipes[index];
984       return buildCard(recipe);
985     },
986   ),
987   separatorBuilder: (context, index) {
988 // 5
989     return const SizedBox(width: 16);
990   },
991 },
992 ),
993 ],
994 ),
995 ],
996 );
997 );
998 }
999 Widget buildCard(ExploreRecipe recipe) {
1000   if (recipe.cardType == RecipeCardType.card1) {
1001     return Card1(recipe: recipe);
1002   } else if (recipe.cardType == RecipeCardType.card2) {
1003     return Card2(recipe: recipe);
1004   } else if (recipe.cardType == RecipeCardType.card3) {
1005     return Card3(recipe: recipe);
1006   } else {
1007     throw Exception('This card doesn\'t exist yet');
1008   }
1009 }
1010 ]
1011
1012
1013 app_cache.dart
1014 import 'package:shared_preferences/shared_preferences.dart';
1015
1016 class AppCache {
1017   static const kUser = 'user';
1018   static const kOnboarding = 'onboarding';
1019
1020 Future<void> invalidate() async {
1021   final prefs = await SharedPreferences.getInstance();
1022 }
```

Fig 31.

A screenshot of a code editor window titled "Text Editor". The code is written in Dart. It includes a class AppCache with methods for invalidate, cacheUser, completeOnboarding, isUserLoggedIn, and didCompleteOnboarding. The invalidate method uses SharedPreferences to set user and onboarding booleans to false. The cacheUser method sets the user boolean to true. The completeOnboarding method sets the onboarding boolean to true. The isUserLoggedIn method checks if the user boolean is true. The didCompleteOnboarding method checks if the onboarding boolean is true. The code is annotated with line numbers from 971 to 1045.

```
971 }
972 Widget buildCard(ExploreRecipe recipe) {
973   if (recipe.cardType == RecipeCardType.card1) {
974     return Card1(recipe: recipe);
975   } else if (recipe.cardType == RecipeCardType.card2) {
976     return Card2(recipe: recipe);
977   } else if (recipe.cardType == RecipeCardType.card3) {
978     return Card3(recipe: recipe);
979   } else {
980     throw Exception('This card doesn\'t exist yet');
981   }
982 }
983 ]
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013 app_cache.dart
1014 import 'package:shared_preferences/shared_preferences.dart';
1015
1016 class AppCache {
1017   static const kUser = 'user';
1018   static const kOnboarding = 'onboarding';
1019
1020 Future<void> invalidate() async {
1021   final prefs = await SharedPreferences.getInstance();
1022   await prefs.setBool(kUser, false);
1023   await prefs.setBool(kOnboarding, false);
1024 }
1025
1026 Future<void> cacheUser() async {
1027   final prefs = await SharedPreferences.getInstance();
1028   await prefs.setBool(kUser, true);
1029 }
1030
1031 Future<void> completeOnboarding() async {
1032   final prefs = await SharedPreferences.getInstance();
1033   await prefs.setBool(kOnboarding, true);
1034 }
1035
1036 Future<bool> isUserLoggedIn() async {
1037   final prefs = await SharedPreferences.getInstance();
1038   return prefs.getBool(kUser) ?? false;
1039 }
1040
1041 Future<bool> didCompleteOnboarding() async {
1042   final prefs = await SharedPreferences.getInstance();
1043   return prefs.getBool(kOnboarding) ?? false;
1044 }
1045 }
```

Fig 32.

A screenshot of a Linux desktop environment. At the top, there's a dark header bar with the title "Activities Text Editor" and a date/time stamp "May 3 11:01". Below the header is an "Untitled Document 1" tab. On the right side of the header, there are standard window control buttons (minimize, maximize, close) and some status indicators like "Save" and "INS". The main area is a code editor with a light beige background. The code is written in Dart and defines a class `AppStateManager` with methods for initializing the app, logging in, completing onboarding, and navigating tabs. It also includes a `CookipidiaTab` class and a `cookipidia_pages.dart` file. The bottom of the screen shows a dock with various application icons, including a terminal, file manager, browser, and productivity tools. A status bar at the very bottom displays "Dart Tab Width: 8 Ln 19, Col 1 INS".

```
1050 import 'package:flutter/material.dart';
1051
1052 import 'app_cache.dart';
1053
1054 class CookipidiaTab {
1055   static const int explore = 0;
1056   static const int recipes = 1;
1057   static const int toBuy = 2;
1058 }
1059
1060 class AppStateManager extends ChangeNotifier {
1061   bool _initialized = false;
1062   bool _loggedin = false;
1063   bool _onboardingComplete = false;
1064   int _selectedTab = CookipidiaTab.explore;
1065   final _appCache = AppCache();
1066
1067   bool get isInitialized => _initialized;
1068   bool get isloggedin => _loggedin;
1069   bool get isOnboardingComplete => _onboardingComplete;
1070   int get selectedTab => _selectedTab;
1071
1072   void initializeApp() async {
1073     _loggedin = await _appCache.lsUserLoggedIn();
1074     _onboardingComplete = await _appCache.didCompleteOnboarding();
1075
1076     Timer(
1077       const Duration(milliseconds: 2000),
1078       () {
1079         _initialized = true;
1080         notifyListeners();
1081       },
1082     );
1083   }
1084 }
1085
1086 void login(String username, String password) async {
1087   _loggedin = true;
1088   await _appCache.cacheUser();
1089   notifyListeners();
1090 }
1091
1092 void completeOnboarding() async {
1093   _onboardingComplete = true;
1094   await _appCache.completeOnboarding();
1095   notifyListeners();
1096 }
1097
1098 void gotoTab(index) {
1099   _selectedTab = index;
1100   notifyListeners();
1101 }
```

Fig 33.

A screenshot of a Linux desktop environment, similar to Fig 33 but with more content visible. The code editor window shows a longer Dart file. The `initializeApp` method now includes a timer to set `_initialized` to true after a 2-second delay. The `login` and `completeOnboarding` methods remain the same. The `gotoTab` method is updated to accept an index parameter. A new method `goToRecipes` is added, which sets the selected tab to the `recipes` value. The `logout` method is updated to clear the selected tab and invalidate the cache. The `cookipidia_pages.dart` file is also partially visible at the bottom.

```
1070
1071   Timer(
1072     const Duration(milliseconds: 2000),
1073     () {
1074       _initialized = true;
1075       notifyListeners();
1076     },
1077   );
1078
1079
1080 void login(String username, String password) async {
1081   _loggedin = true;
1082   await _appCache.cacheUser();
1083   notifyListeners();
1084 }
1085
1086 void completeOnboarding() async {
1087   _onboardingComplete = true;
1088   await _appCache.completeOnboarding();
1089   notifyListeners();
1090 }
1091
1092 void gotoTab(index) {
1093   _selectedTab = index;
1094   notifyListeners();
1095 }
1096
1097 void goToRecipes() {
1098   _selectedTab = CookipidiaTab.recipes;
1099   notifyListeners();
1100 }
1101
1102 void logout() async {
1103   _initialized = false;
1104   _selectedTab = 0;
1105   await _appCache.invalidate();
1106
1107   initializeApp();
1108   notifyListeners();
1109 }
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119 cookipidia_pages.dart
1120 class CookipidiaPages {
1121   static String splashPath = '/splash';
1122   static String loginPath = '/login';
1123   static String onboardPath = '/onboarding';
1124   static String home = '/';
1125   static String groceryItemDetails = '/item';
1126   static String profilePath = '/profile';
1127 }
```

Fig 34.

```
Activities Text Editor
Open Untitled Document 1
May 3 11:01
1103 void goToRecipes() {
1104     _selectedTab = CookipediaTab.recipes;
1105     notifyListeners();
1106 }
1107
1108 void logout() async {
1109     _initialized = false;
1110     _selectedTab = 0;
1111     await _appCache.invalidate();
1112     initializeApp();
1113     notifyListeners();
1114 }
1115 }
1116
1117
1118 cookipedia_pages.dart
1119 class CookipediaPages {
1120     static String splashPath = '/splash';
1121     static String loginPath = '/login';
1122     static String onboardPath = '/onboarding';
1123     static String home = '/';
1124     static String groceryItemDetails = '/item';
1125     static String profilePath = '/profile';
1126     static String raywenderlich = '/raywenderlich';
1127     static String accountPath = '/account';
1128 }
1129
1130
1131
1132 explore_data.dart
1133 import 'models.dart';
1134
1135 class ExploreData {
1136     final List<ExploreRecipe> todayRecipes;
1137     final List<Post> friendPosts;
1138 }
1139
1140 ExploreData(
1141     this.todayRecipes,
1142     this.friendPosts,
1143 );
1144
1145
1146 explore_recipe.dart
1147 part 'ingredient.dart';
1148 part 'instruction.dart';
1149
1150
1151 class RecipeCardType {
1152     static const card1 = 'card1';
1153     static const card2 = 'card2';
1154 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 35.

```
Activities Text Editor
Open Untitled Document 1
May 3 11:01
Save
May 3 11:01
*Untitled Document 1
1129 ]
1130
1131
1132 explore_data.dart
1133 import 'models.dart';
1134
1135 class ExploreData {
1136     final List<ExploreRecipe> todayRecipes;
1137     final List<Post> friendPosts;
1138 }
1139
1140 ExploreData(
1141     this.todayRecipes,
1142     this.friendPosts,
1143 );
1144
1145
1146 explore_recipe.dart
1147 part 'ingredient.dart';
1148 part 'instruction.dart';
1149
1150
1151 class RecipeCardType {
1152     static const card1 = 'card1';
1153     static const card2 = 'card2';
1154     static const card3 = 'card3';
1155 }
1156
1157 class ExploreRecipe {
1158     String id;
1159     String cardType;
1160     String title;
1161     String subtitle;
1162     String backgroundImage;
1163     String backgroundImageSource;
1164     String message;
1165     String authorName;
1166     String authorRole;
1167     String profileImage;
1168     int durationInMinutes;
1169     String dietType;
1170     int calories;
1171     List<String> tags;
1172     String description;
1173     String source;
1174     List<Ingredients> ingredients;
1175     List<Instruction> instructions;
1176
1177     ExploreRecipe({
1178         required this.id,
1179         required this.cardType,
1180     });
1181 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 36.

A screenshot of a code editor window titled "Text Editor". The code is written in Dart and defines a class named `ExploreRecipe`. The class has several fields: `id`, `cardType`, `title`, `subtitle`, `backgroundImage`, `backgroundImageSource`, `message`, `authorName`, `role`, `profileImage`, `durationInMinutes`, `dietType`, `calories`, `tags`, `description`, `source`, `Ingredients`, and `Instructions`. The constructor `ExploreRecipe()` initializes these fields with empty strings or zero values. A factory method `fromJson` takes a map of strings and converts it into an instance of `ExploreRecipe`, populating the fields from the JSON data.

```
Activities Text Editor
Open Untitled Document 1 Save
May 3 11:01
1156
1157 class ExploreRecipe {
1158   String id;
1159   String cardType;
1160   String title;
1161   String subtitle;
1162   String backgroundImage;
1163   String backgroundImageSource;
1164   String message;
1165   String authorName;
1166   String role;
1167   String profileImage;
1168   int durationInMinutes;
1169   String dietType;
1170   int calories;
1171   List<String> tags;
1172   String description;
1173   String source;
1174   List<Ingredients> ingredients;
1175   List<Instruction> instructions;
1176
1177   ExploreRecipe({
1178     required this.id,
1179     required this.cardType,
1180     required this.title,
1181     required this.subtitle,
1182     this.backgroundImage = '',
1183     this.backgroundImageSource = '',
1184     this.message = '',
1185     this.authorName = '',
1186     this.role = '',
1187     this.profileImage = '',
1188     this.durationInMinutes = 0,
1189     this.dietType = '',
1190     this.calories = 0,
1191     this.tags = const [],
1192     this.description = '',
1193     this.source = '',
1194     this.ingredients = const [],
1195     this.instructions = const [],
1196   });
1197
1198   factory ExploreRecipe.fromJson(Map<String, dynamic> json) {
1199     final ingredients = <Ingredients>[];
1200     final instructions = <Instruction>[];
1201
1202     if (json['ingredients'] != null) {
1203       json['ingredients'].forEach((v) {
1204         ingredients.add(Ingredients.fromJson(v));
1205       });
1206
1207     if (json['instructions'] != null) {
1208       json['instructions'].forEach((v) {
1209         instructions.add(Instruction.fromJson(v));
1210       });
1211     }
1212   }
1213
1214   return ExploreRecipe(
1215     id: json['id'] ?? '',
1216     cardType: json['cardType'] ?? '',
1217     title: json['title'] ?? '',
1218     subtitle: json['subtitle'] ?? '',
1219     backgroundImage: json['backgroundImage'] ?? '',
1220     backgroundImageSource: json['backgroundImageSource'] ?? '',
1221     message: json['message'] ?? '',
1222     authorName: json['authorName'] ?? '',
1223     role: json['role'] ?? '',
1224     profileImage: json['profileImage'] ?? '',
1225     durationInMinutes: json['durationInMinutes'] ?? 0,
1226     dietType: json['dietType'] ?? '',
1227     calories: json['calories'] ?? 0,
1228     tags: json['tags'].cast<String>() ?? [],
1229     description: json['description'] ?? '',
1230     source: json['source'] ?? '',
1231     ingredients: ingredients,
1232   );
1233 }
```

Fig 37.

A screenshot of a code editor window titled "Text Editor". The code is identical to Fig 37, defining a `ExploreRecipe` class with fields: `id`, `cardType`, `title`, `subtitle`, `backgroundImage`, `backgroundImageSource`, `message`, `authorName`, `role`, `profileImage`, `durationInMinutes`, `dietType`, `calories`, `tags`, `description`, `source`, `Ingredients`, and `Instructions`. The constructor `ExploreRecipe()` initializes these fields with empty strings or zero values. A factory method `fromJson` takes a map of strings and converts it into an instance of `ExploreRecipe`, populating the fields from the JSON data.

```
Activities Text Editor
Open Untitled Document 1 Save
May 3 11:01
1156
1157 class ExploreRecipe {
1158   String id;
1159   String cardType;
1160   String title;
1161   String subtitle;
1162   String backgroundImage;
1163   String backgroundImageSource;
1164   String message;
1165   String authorName;
1166   String role;
1167   String profileImage;
1168   int durationInMinutes;
1169   String dietType;
1170   int calories;
1171   List<String> tags;
1172   String description;
1173   String source;
1174   List<Ingredients> ingredients;
1175   List<Instruction> instructions;
1176
1177   ExploreRecipe({
1178     required this.id,
1179     required this.cardType,
1180     required this.title,
1181     required this.subtitle,
1182     this.backgroundImage = '',
1183     this.backgroundImageSource = '',
1184     this.message = '',
1185     this.authorName = '',
1186     this.role = '',
1187     this.profileImage = '',
1188     this.durationInMinutes = 0,
1189     this.dietType = '',
1190     this.calories = 0,
1191     this.tags = const [],
1192     this.description = '',
1193     this.source = '',
1194     this.ingredients = const [],
1195     this.instructions = const [],
1196   });
1197
1198   factory ExploreRecipe.fromJson(Map<String, dynamic> json) {
1199     final ingredients = <Ingredients>[];
1200     final instructions = <Instruction>[];
1201
1202     if (json['ingredients'] != null) {
1203       json['ingredients'].forEach((v) {
1204         ingredients.add(Ingredients.fromJson(v));
1205       });
1206
1207     if (json['instructions'] != null) {
1208       json['instructions'].forEach((v) {
1209         instructions.add(Instruction.fromJson(v));
1210       });
1211     }
1212   }
1213
1214   return ExploreRecipe(
1215     id: json['id'] ?? '',
1216     cardType: json['cardType'] ?? '',
1217     title: json['title'] ?? '',
1218     subtitle: json['subtitle'] ?? '',
1219     backgroundImage: json['backgroundImage'] ?? '',
1220     backgroundImageSource: json['backgroundImageSource'] ?? '',
1221     message: json['message'] ?? '',
1222     authorName: json['authorName'] ?? '',
1223     role: json['role'] ?? '',
1224     profileImage: json['profileImage'] ?? '',
1225     durationInMinutes: json['durationInMinutes'] ?? 0,
1226     dietType: json['dietType'] ?? '',
1227     calories: json['calories'] ?? 0,
1228     tags: json['tags'].cast<String>() ?? [],
1229     description: json['description'] ?? '',
1230     source: json['source'] ?? '',
1231     ingredients: ingredients,
1232   );
1233 }
```

Fig 38.

A screenshot of a Linux desktop environment. The window title is "Text Editor". The status bar at the top shows "May 3 11:01" and "Untitled Document 1". The status bar at the bottom shows "Dart Tab Width: 8 Ln 19, Col 1 INS". The desktop dock at the bottom includes icons for a file manager, terminal, browser, and other applications.

```
1208     if (json['instructions'] != null) {
1209         json['instructions'].forEach((v) {
1210             instructions.add(Instruction.fromJson(v));
1211         });
1212     }
1213
1214     return ExploreRecipe(
1215         id: json['id'] ?? '',
1216         cardType: json['cardtype'] ?? '',
1217         title: json['title'] ?? '',
1218         subtitle: json['subtitle'] ?? '',
1219         backgroundImage: json['backgroundImage'] ?? '',
1220         backgroundImageSource: json['backgroundImageSource'] ?? '',
1221         message: json['message'] ?? '',
1222         authorName: json['authorName'] ?? '',
1223         role: json['role'] ?? '',
1224         profileImage: json['profileImage'] ?? '',
1225         durationInMinutes: json['durationInMinutes'] ?? 0,
1226         dietType: json['dietype'] ?? '',
1227         calories: json['calories'] ?? 0,
1228         tags: json['tags'].cast<String>() ?? [],
1229         description: json['description'] ?? '',
1230         source: json['source'] ?? '',
1231         ingredients: ingredients,
1232         instructions: instructions,
1233     );
1234 }
1235 }
1236
1237 grocery_item.dart
1238 import 'package:flutter/painting.dart';
1239
1240 // 1
1241 enum Importance {
1242     low,
1243     medium,
1244     high,
1245 }
1246
1247 class GroceryItem {
1248 // 2
1249     final String id;
1250
1251 // 3
1252     final String name;
1253     final Importance importance;
1254     final Color color;
1255     final int quantity;
1256     final DateTime date;
1257     final bool isComplete;
1258 }
```

Fig 39.

A screenshot of a Linux desktop environment. The window title is "Text Editor". The status bar at the top shows "May 3 11:01" and "Untitled Document 1". The status bar at the bottom shows "Dart Tab Width: 8 Ln 19, Col 1 INS". The desktop dock at the bottom includes icons for a file manager, terminal, browser, and other applications.

```
1234 }
1235 ]
1236
1237 grocery_item.dart
1238 import 'package:flutter/painting.dart';
1239
1240 // 1
1241 enum Importance {
1242     low,
1243     medium,
1244     high,
1245 }
1246
1247 class GroceryItem {
1248 // 2
1249     final String id;
1250
1251 // 3
1252     final String name;
1253     final Importance importance;
1254     final Color color;
1255     final int quantity;
1256     final DateTime date;
1257     final bool isComplete;
1258
1259     GroceryItem({
1260         required this.id,
1261         required this.name,
1262         required this.importance,
1263         required this.color,
1264         required this.quantity,
1265         required this.date,
1266         this.isComplete = false,
1267     });
1268
1269     GroceryItem copyWith({
1270         String? id,
1271         String? name,
1272         Importance? importance,
1273         Color? color,
1274         int? quantity,
1275         DateTime? date,
1276         bool? isComplete,
1277     }) {
1278         return GroceryItem(
1279             id: id ?? this.id,
1280             name: name ?? this.name,
1281             importance: importance ?? this.importance,
1282             color: color ?? this.color,
1283             quantity: quantity ?? this.quantity,
1284         );
1285     }
1286 }
```

Fig 40.

Activities Text Editor

May 3 11:01 *Untitled Document 1 Save

```
1260 GroceryItem({  
1261   required this.id,  
1262   required this.name,  
1263   required this.importance,  
1264   required this.color,  
1265   required this.quantity,  
1266   required this.date,  
1267   this.isComplete = false,  
1268 });  
1269  
1270 GroceryItem copyWith({  
1271   String? id,  
1272   String? name,  
1273   Importance? importance,  
1274   Color? color,  
1275   int? quantity,  
1276   DateTime? date,  
1277   bool? isComplete,  
1278 }) {  
1279   return GroceryItem(  
1280     id: id ?? this.id,  
1281     name: name ?? this.name,  
1282     importance: importance ?? this.importance,  
1283     color: color ?? this.color,  
1284     quantity: quantity ?? this.quantity,  
1285     date: date ?? this.date,  
1286     isComplete: isComplete ?? this.isComplete,  
1287   );  
1288 }  
1289  
1290 grocery_manager.dart  
1291  
1292 import 'package:flutter/material.dart';  
1293  
1294 import 'grocery_item.dart';  
1295  
1296 class GroceryManager extends ChangeNotifier {  
1297   final _groceryItems = <GroceryItem>[];  
1298   int _selectedIndex = -1;  
1299   bool _createNewItem = false;  
1300  
1301 List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);  
1302 int get selectedIndex => _selectedIndex;  
1303 GroceryItem? get selectedGroceryitem =>  
1304   _selectedIndex != -1 ? _groceryItems[_selectedIndex] : null;  
1305 bool get isCreatingNewItem => _createNewItem;  
1306  
1307 void createNewItem() {  
1308   _createNewItem = true;  
1309   notifyListeners();  
1310 }
```

Save Tab Width: 8 Ln 19, Col 1 INS

Fig 41.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
1286     });
1287   );
1288 }
1289 }
1290
1291 grocery_manager.dart
1292 import 'package:flutter/material.dart';
1293
1294 import 'grocery_item.dart';
1295
1296 class GroceryManager extends ChangeNotifier {
1297   final _groceryItems = <GroceryItem>[];
1298   int _selectedIndex = -1;
1299   bool _createNewItem = false;
1300
1301   List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);
1302   int get selectedIndex => _selectedIndex;
1303   GroceryItem? get selectedGroceryItem =>
1304     _selectedIndex != -1 ? _groceryItems[_selectedIndex] : null;
1305   bool get isCreatingNewItem => _createNewItem;
1306
1307   void createNewItem() {
1308     _createNewItem = true;
1309     notifyListeners();
1310   }
1311
1312   void deleteItem(int index) {
1313     _groceryItems.removeAt(index);
1314     notifyListeners();
1315   }
1316
1317   void groceryItemTapped(int index) {
1318     _selectedIndex = index;
1319     _createNewItem = false;
1320     notifyListeners();
1321   }
1322
1323   void addNewItem(GroceryItem item) {
1324     _groceryItems.add(item);
1325     _createNewItem = false;
1326     notifyListeners();
1327   }
1328
1329   void updateItem(GroceryItem item, int index) {
1330     _groceryItems[index] = item;
1331     _selectedIndex = -1;
1332     _createNewItem = false;
1333     notifyListeners();
1334   }
1335 }
1336 void onCompleteItem(int index, bool channel, f
```

Fig 42.

```
1313 void deleteItem(int index) {
1314     _groceryItems.removeAt(index);
1315     notifyListeners();
1316 }
1317
1318 void groceryItemTapped(int index) {
1319     _selectedIndex = index;
1320     _createNewItem = false;
1321     notifyListeners();
1322 }
1323
1324 void addItem(GroceryItem item) {
1325     _groceryItems.add(item);
1326     _createNewItem = false;
1327     notifyListeners();
1328 }
1329
1330 void updateItem(GroceryItem item, int index) {
1331     _groceryItems[index] = item;
1332     _selectedIndex = -1;
1333     _createNewItem = false;
1334     notifyListeners();
1335 }
1336
1337 void completeItem(int index, bool change) {
1338     final item = _groceryItems[index];
1339     _groceryItems[index] = item.copyWith(isComplete: change);
1340     notifyListeners();
1341 }
1342
1343
1344 ingredient.dart
1345 part of 'explore_recipe.dart';
1346
1347 class Ingredients {
1348     String imageUrl;
1349     String title;
1350     String source;
1351
1352     Ingredients({
1353         required this.imageUrl,
1354         required this.title,
1355         required this.source,
1356     });
1357
1358     factory Ingredients.fromJson(Map<String, dynamic> json) {
1359         return Ingredients(
1360             imageUrl: json['imageUrl'] ?? '',
1361             title: json['title'] ?? '',
1362             source: json['source'] ?? '',
1363         );
1364     }
1365 }
1366
1367 instruction.dart
1368 part of 'explore_recipe.dart';
1369
1370 class Instruction {
1371     String imageUrl;
1372     String description;
1373     int durationInMinutes;
1374
1375     Instruction({
1376         required this.imageUrl,
1377         required this.description,
1378         required this.durationInMinutes,
1379     });
1380
1381     factory Instruction.fromJson(Map<String, dynamic> json) {
1382         return Instruction(
1383             imageUrl: json['imageUrl'] ?? '',
1384             description: json['description'] ?? '',
1385             durationInMinutes: json['durationInMinutes'] ?? '',
1386         );
1387     }
1388 }
```

Fig 43.

```
1359     _groceryItems[index] = item.copyWith(isComplete: change);
1360     notifyListeners();
1361 }
1362 }
1363
1364 ingredient.dart
1365 part of 'explore_recipe.dart';
1366
1367 class Ingredients {
1368     String imageUrl;
1369     String title;
1370     String source;
1371
1372     Ingredients({
1373         required this.imageUrl,
1374         required this.title,
1375         required this.source,
1376     });
1377
1378     factory Ingredients.fromJson(Map<String, dynamic> json) {
1379         return Ingredients(
1380             imageUrl: json['imageUrl'] ?? '',
1381             title: json['title'] ?? '',
1382             source: json['source'] ?? '',
1383         );
1384     }
1385 }
1386
1387 instruction.dart
1388 part of 'explore_recipe.dart';
1389
1390 class Instruction {
1391     String imageUrl;
1392     String description;
1393     int durationInMinutes;
1394
1395     Instruction({
1396         required this.imageUrl,
1397         required this.description,
1398         required this.durationInMinutes,
1399     });
1400
1401     factory Instruction.fromJson(Map<String, dynamic> json) {
1402         return Instruction(
1403             imageUrl: json['imageUrl'] ?? '',
1404             description: json['description'] ?? '',
1405             durationInMinutes: json['durationInMinutes'] ?? '',
1406         );
1407     }
1408 }
```

Fig 44.

Activities Text Editor

May 3 11:01 *Untitled Document 1 Save

```
342     }],
343   ],
344 },
345 );
346 }
347 }
348 }
349 }
350 card1.dart
351 import 'package:cookipidia/cookipidia_theme.dart';
352 import 'package:cookipidia/main.dart';
353 import 'package:flutter/material.dart';
354 import 'package:cookipidia/cookipidia_theme.dart';
355
356 import '../models/explore_recipe.dart';
357
358 class Card1 extends StatelessWidget {
359   final ExploreRecipe recipe;
360
361   const Card1({
362     Key? key,
363     required this.recipe,
364   }) : super(key: key);
365
366   // 2
367   @override
368   Widget build(BuildContext context) {
369   // 3
370     return Center(
371       child: Container(
372         child: Stack(
373           children: [
374             Text(
375               recipe.subtitle,
376               style: CookipidiaTheme.darkTextTheme.bodyText1,
377             ),
378             Positioned(
379               child: Text(
380                 recipe.title,
381                 style: CookipidiaTheme.darkTextTheme.headline2,
382               ),
383               top: 20,
384             ),
385             Positioned(
386               child: Text(
387                 recipe.message,
388                 style: CookipidiaTheme.darkTextTheme.bodyText1,
389               ),
390               bottom: 30,
391               right: 0,
392             )
393           ],
394         ),
395       ),
396     );
397   }
398 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 45.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
421 card2.dart
422 import 'package:flutter/material.dart';
423 import '../models/explore_recipe.dart';
424 import 'author_card.dart';
425 import '../cookipidia_theme.dart';
426
427 class Card2 extends StatelessWidget {
428   final ExploreRecipe recipe;
429
430   const Card2({
431     Key? key,
432     required this.recipe,
433   }) : super(key: key);
434   @override
435   Widget build(BuildContext context) {
436     return Center(
437       child: Container(
438         constraints: const BoxConstraints.expand(
439           width: 350,
440           height: 450,
441         ),
442         decoration: BoxDecoration(
443           image: DecorationImage(
444             image: AssetImage(recipe.backgroundImage),
445             fit: BoxFit.cover,
446           ),
447           borderRadius: const BorderRadius.all(Radius.circular(10.0)),
448         ),
449         child: Column(
450           children: [
451             AuthorCard(
452               authorName: recipe.authorName,
453               title: recipe.role,
454               imageProvider: AssetImage(recipe.profileImage),
455             ),
456             Expanded(
457               child: Stack(
458                 children: [
459                   Positioned(
460                     bottom: 16,
461                     right: 16,
462                     child: Text(
463                       recipe.title,
464                       style: CookipidiaTheme.lightTextTheme.headline1,
465                     ),
466                   ),
467                   Positioned(
468                     bottom: 78,
469                     left: 16,
470                     child: RotatedBox(
471
```

Tab Width: 8 ▾ Ln 19, Col 1 ▾ INS

Fig 46.

A screenshot of a Linux desktop environment showing a text editor window titled "Untitled Document 1". The window displays Dart code for a "Card3" widget. The code includes imports for flutter/material.dart, cookipedia_theme.dart, and models/explore_recipe.dart. It defines a "Card3" class that extends StatelessWidget and contains logic for creating tag chips from recipe.tags. The code is annotated with line numbers from 473 to 513. The desktop bar at the bottom shows various application icons.

```
473     ),
474     style: CookipediaTheme.lightTextTheme.headline1,
475   ),
476   ],
477   ],
478   ],
479   ],
480   ],
481   ],
482   ],
483   );
484 }
485 }
486 }
487
488 card3.dart
489 import 'package:flutter/material.dart';
490 import '../cookipedia_theme.dart';
491 import '../models/explore_recipe.dart';
492
493 class Card3 extends StatelessWidget {
494   final ExploreRecipe recipe;
495
496   const Card3({
497     Key? key,
498     required this.recipe,
499   }) : super(key: key);
500
501   List<Widget> createTagChips() {
502     final chips = <Widget>[];
503     recipe.tags.take(6).forEach((element) {
504       final chip = Chip(
505         label: Text(
506           element,
507           style: CookipediaTheme.darkTextTheme.bodyText1,
508         ),
509         backgroundColor: Colors.black.withOpacity(0.7),
510       );
511       chips.add(chip);
512     });
513
514     return chips;
515   }
516   @override
517   Widget build(BuildContext context) {
518     return Center(
519       child: Container(
520         constraints: const BoxConstraints.expand(
521           width: 350,
522           height: 450,
523         ),
524       ),
525       decoration: BoxDecoration(
526         image: AssetImage(recipe.backgroundImage),
527         fit: BoxFit.cover,
528       ),
529       borderRadius: const BorderRadius.all(
530         Radius.circular(10.0),
531       ),
532     );
533     child: Stack(
534       children: [
535         Container(
536           decoration: BoxDecoration(
537             color: Colors.black.withOpacity(0.6),
538             borderRadius: const BorderRadius.all(Radius.circular(10.0)),
539           ),
540         ),
541         Container(
542           padding: const EdgeInsets.all(16),
543           child: Column(
544             crossAxisAlignment: CrossAxisAlignment.start,
545             children: [
546               const Icon(
547                 Icons.book,
548                 color: Colors.white,
549                 size: 40,
550               ),
551               const SizedBox(height: 8),
552               Text(recipe.title,
553                 style: CookipediaTheme.darkTextTheme.headline2),
554               const SizedBox(height: 30),
555             ],
556           ),
557         ),
558         Center(
559           child: Wrap(
560             alignment: WrapAlignment.start,
561             spacing: 12,
562             runSpacing: 12,
563             children: createTagChips(),
564           ),
565         ),
566       ),
567     );
568   );
569 }
570
571 circle_image.dart
572 import 'package:flutter/material.dart';
573 class CircleImage extends StatelessWidget {
574   const CircleImage({
```

Fig 47.

A screenshot of a Linux desktop environment showing a text editor window titled "Untitled Document 1". The window displays Dart code for a "CircleImage" widget. The code includes imports for flutter/material.dart. It defines a "CircleImage" class that extends StatelessWidget and contains logic for creating a wrap of children with specific alignment and spacing. The code is annotated with line numbers from 525 to 578. The desktop bar at the bottom shows various application icons.

```
525   ),
526   style: CookipediaTheme.lightTextTheme.headline1,
527   ),
528   ],
529   ],
530   ],
531   ],
532   ],
533   ],
534   ],
535   ],
536   ],
537   ],
538   ],
539   ],
540   ],
541   ],
542   ],
543   ],
544   ],
545   ],
546   ],
547   ],
548   ],
549   ],
550   ],
551   ],
552   ],
553   ],
554   ],
555   ],
556   ],
557   ],
558   ],
559   ],
560   ],
561   ],
562   ],
563   ],
564   ],
565   ],
566   ],
567   ],
568   ],
569   ],
570
571 circle_image.dart
572 import 'package:flutter/material.dart';
573 class CircleImage extends StatelessWidget {
574   const CircleImage({
```

Fig 48.

A screenshot of a Flutter IDE interface. The main area displays the following Dart code:

```
576     child: CircleAvatar(
577       imageRadius = 20,
578     )) : super(key: key);
579   }
580 
581   final double imageRadius;
582   final ImageProvider? imageProvider;
583 
584   @override
585   Widget build(BuildContext context) {
586     // ...
587     return CircleAvatar(
588       backgroundColor: Colors.white,
589       radius: imageRadius,
590     );
591     child: CircleAvatar(
592       radius: imageRadius - 5,
593       backgroundImage: imageProvider,
594     ),
595   );
596 }
597 }
598 
599 components.dart
600 export 'author_card.dart';
601 export 'card1.dart';
602 export 'card2.dart';
603 export 'card3.dart';
604 export 'circle_image.dart';
605 export 'today_recipe_list_view.dart';
606 export 'friend_post_tile.dart';
607 export 'friend_post_list_view.dart';
608 export 'recipe_thumbnail.dart';
609 export 'recipes_grid_view.dart';
610 export 'today_recipe_list_view.dart';
611 
612 friend_post_list_view.dart
613 import 'package:flutter/material.dart';
614 import '../models/models.dart';
615 import 'components.dart';
616 
617 class FriendPostListView extends StatelessWidget {
618   // ...
619   final List<Post> friendPosts;
620 
621   const FriendPostListView({
622     Key? key,
623     required this.friendPosts,
624   }) : super(key: key);
625 
626   @override
627   Widget build(BuildContext context) {
628     // ...
629     padding: const EdgeInsets.only(
630       left: 16,
631       right: 16,
632       top: 0,
633     ),
634     ),
635   );
636     child: Column(
637       crossAxisAlignment: CrossAxisAlignment.start,
638       children: [
639       // ...
640       Text('Social Chefs ! ', style: Theme.of(context).textTheme.headline1),
641       // ...
642       const SizedBox(height: 16),
643       // TODO: Add PostListTile here
644       ListView.separated(
645         // ...
646         primary: false,
647         // ...
648         physics: const NeverScrollableScrollPhysics(),
649       );
650       shrinkWrap: true,
651       scrollDirection: Axis.vertical,
652       itemCount: friendPosts.length,
653       itemBuilder: (context, index) {
654         // ...
655         final post = friendPosts[index];
656         return FriendPostTitle(post: post);
657       },
658       separatorBuilder: (context, index) {
659       // ...
660       return const SizedBox(height: 16);
661     },
662   ),
663   const SizedBox(height: 16),
664   ],
665   ),
666   );
667 );
668 ]
669 }
670 
671 friend_post_tile.dart
672 import 'package:flutter/material.dart';
673 import '../components/components.dart';
674 import '../models/models.dart';
675 
676 class FriendPostTitle extends StatelessWidget {
677   final Post post;
678 
679   const FriendPostTitle({
680     Key? key,
681   }) : super(key: key);
682 
683   @override
684   Widget build(BuildContext context) {
685     // ...
686     return Container(
687       // ...
688       padding: const EdgeInsets.all(16),
689       // ...
690       child: Column(
691         // ...
692         mainAxisAlignment: MainAxisAlignment.end,
693         children: [
694           // ...
695           Text(post.title),
696           // ...
697         ],
698       ),
699     );
700   }
701 }
```

The status bar at the bottom shows various application icons and the text "Dart Tab Width: 8 Ln 19, Col 1 INS".

Fig 49.

A screenshot of a Flutter IDE interface. The main area displays the following Dart code:

```
630     padding: const EdgeInsets.only(
631       left: 16,
632       right: 16,
633       top: 0,
634     ),
635   );
636   child: Column(
637     crossAxisAlignment: CrossAxisAlignment.start,
638     children: [
639     // ...
640     Text('Social Chefs ! ', style: Theme.of(context).textTheme.headline1),
641     // ...
642     const SizedBox(height: 16),
643     // TODO: Add PostListTile here
644     ListView.separated(
645       // ...
646       primary: false,
647       // ...
648       physics: const NeverScrollableScrollPhysics(),
649     );
650     shrinkWrap: true,
651     scrollDirection: Axis.vertical,
652     itemCount: friendPosts.length,
653     itemBuilder: (context, index) {
654       // ...
655       final post = friendPosts[index];
656       return FriendPostTitle(post: post);
657     },
658     separatorBuilder: (context, index) {
659     // ...
660     return const SizedBox(height: 16);
661   },
662   ),
663   const SizedBox(height: 16),
664   ],
665   ),
666   );
667 );
668 ]
669 }
670 
671 friend_post_tile.dart
672 import 'package:flutter/material.dart';
673 import '../components/components.dart';
674 import '../models/models.dart';
675 
676 class FriendPostTitle extends StatelessWidget {
677   final Post post;
678 
679   const FriendPostTitle({
680     Key? key,
681   }) : super(key: key);
682 
683   @override
684   Widget build(BuildContext context) {
685     // ...
686     return Container(
687       // ...
688       padding: const EdgeInsets.all(16),
689       // ...
690       child: Column(
691         // ...
692         mainAxisAlignment: MainAxisAlignment.end,
693         children: [
694           // ...
695           Text(post.title),
696           // ...
697         ],
698       ),
699     );
700   }
701 }
```

The status bar at the bottom shows various application icons and the text "Dart Tab Width: 8 Ln 19, Col 1 INS".

Fig 50.

A screenshot of a Linux desktop environment showing a code editor window. The window title is "Text Editor" and the document title is "*Untitled Document 1". The code is written in Dart and defines a StatelessWidget named GroceryTile. It includes imports for flutter/material.dart, google_fonts/google_fonts.dart, intl/intl.dart, and cookipedia/models/grocery_item.dart. The class contains a constructor taking a key and an item, and an onComplete callback. The build method returns a Row with a CircleImage and a Column containing a Text for the comment and a Text for the timestamp.

```
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
735
736 // 4
737 GroceryTile({
738   Key? key,
739   required this.item,
740   this.onComplete,
741 }: textDecoration =
742   item.isIncomplete ? TextDecoration.lineThrough : TextDecoration.none,
743   super(key: key);
744
745 @override
746 Widget build(BuildContext context) {
747 // TODO: Change this Widget
748   return SizedBox(
749     height: 106.0,
750     child: Row(
751       mainAxisAlignment: MainAxisAlignment.spaceBetween,
752       children: [
753         // 2
754         Container(
755           width: 5.0,
756           color: item.color,
757         ),
758       ],
759     ),
760     //const SizedBox(width: 16.0),
761   );
762 // 4
763   Column(
764     mainAxisAlignment: MainAxisAlignment.center,
765     crossAxisAlignment: CrossAxisAlignment.start,
766     children: [
767       // 5
768       Text(
769         item.name,
770         style: GoogleFonts.lato(
771           decoration: textDecoration,
772           fontSize: 21.0,
773           fontWeight: FontWeight.bold),
774       ),
775       const SizedBox(height: 4.0),
776       buildDate(),
777       const SizedBox(height: 4.0),
778       buildImportance(),
779     ],
780   ),
781   Row(
782     children: [
783       Text(
784 // 7
785
```

Fig 51.

A screenshot of a Linux desktop environment showing a code editor window. The window title is "Text Editor" and the document title is "*Untitled Document 1". The code is written in Dart and defines a StatelessWidget named GroceryTile. It includes imports for flutter/material.dart, google_fonts/google_fonts.dart, intl/intl.dart, and cookipedia/models/grocery_item.dart. The class contains a constructor taking a key and an item, and an onComplete callback. The build method returns a Row with a Container and a Column containing a Text for the name, a Text for the date, and a Text for the importance.

```
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
735
736 // 4
737 GroceryTile({
738   Key? key,
739   required this.item,
740   this.onComplete,
741 }: textDecoration =
742   item.isIncomplete ? TextDecoration.lineThrough : TextDecoration.none,
743   super(key: key);
744
745 @override
746 Widget build(BuildContext context) {
747 // TODO: Change this Widget
748   return SizedBox(
749     height: 106.0,
750     child: Row(
751       mainAxisAlignment: MainAxisAlignment.spaceBetween,
752       children: [
753         // 2
754         Container(
755           width: 5.0,
756           color: item.color,
757         ),
758       ],
759     ),
760     //const SizedBox(width: 16.0),
761   );
762 // 4
763   Column(
764     mainAxisAlignment: MainAxisAlignment.center,
765     crossAxisAlignment: CrossAxisAlignment.start,
766     children: [
767       // 5
768       Text(
769         item.name,
770         style: GoogleFonts.lato(
771           decoration: textDecoration,
772           fontSize: 21.0,
773           fontWeight: FontWeight.bold),
774       ),
775       const SizedBox(height: 4.0),
776       buildDate(),
777       const SizedBox(height: 4.0),
778       buildImportance(),
779     ],
780   ),
781   Row(
782     children: [
783       Text(
784 // 7
785
```

Fig 52.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
786     style: GoogleFonts.lato(
787       decoration: textDecoration,
788       fontSize: 21.0,
789     ),
790   ),
791 },
792 // B
793 buildCheckbox(),
794 ],
795 ],
796 ],
797 ),
798 );
799 }
800
801 Widget buildImportance() {
802   if (item.importance == Importance.low) {
803     return Text('Low', style: GoogleFonts.lato(decoration: textDecoration));
804   } else if (item.importance == Importance.medium) {
805     return Text('Medium',
806       style: GoogleFonts.lato(
807         fontWeight: FontWeight.w800, decoration: textDecoration));
808   } else if (item.importance == Importance.high) {
809     return Text(
810       'High',
811       style: GoogleFonts.lato(
812         color: Colors.red,
813         fontWeight: FontWeight.w900,
814         decoration: textDecoration,
815       ),
816     );
817   } else {
818     throw Exception('This importance type does not exist');
819   }
820 }
821
822 Widget buildDate() {
823   final dateFormatter = DateFormat('EEEE dd h:mm a');
824   final dateString = dateFormatter.format(item.date);
825   return Text(
826     dateString,
827     style: TextStyle(decoration: textDecoration),
828   );
829 }
830
831 Widget buildCheckbox() {
832   return Checkbox(
833     // 1
834     value: item.isComplete,
835     // 2
836     onChanged: onComplete,
837   );
838 }
```

Fig 53.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
761
762 // 4
763
764 Column(
765   mainAxisAlignment: MainAxisAlignment.center,
766   crossAxisAlignment: CrossAxisAlignment.start,
767   children: [
768     Text(
769       item.name,
770       style: GoogleFonts.lato(
771         decoration: textDecoration,
772         fontSize: 21.0,
773         fontWeight: FontWeight.bold),
774     ),
775     const SizedBox(height: 4.0),
776     buildDate(),
777     const SizedBox(height: 4.0),
778     buildImportance(),
779   ],
780 ),
781
782 Row(
783   children: [
784 // 7
785   Text(
786     item.quantity.toString(),
787     style: GoogleFonts.lato(
788       decoration: textDecoration,
789       fontSize: 21.0,
790     ),
791   ),
792 // 8
793   buildCheckbox(),
794   ],
795   ],
796   ],
797   ),
798 );
799 }
800
801 Widget buildImportance() {
802   if (item.importance == Importance.low) {
803     return Text('Low', style: GoogleFonts.lato(decoration: textDecoration));
804   } else if (item.importance == Importance.medium) {
805     return Text('Medium',
806       style: GoogleFonts.lato(
807         fontWeight: FontWeight.w800, decoration: textDecoration));
808   } else if (item.importance == Importance.high) {
809     return Text(
810       'High',
811       style: GoogleFonts.lato(
```

Fig 54.

A screenshot of a Flutter IDE interface. The top bar shows 'Activities' and 'Text Editor'. The status bar indicates 'May 3 11:01' and 'Untitled Document 1'. The main area contains Dart code for the 'RecipeThumbnail' widget. The code imports 'package:flutter/material.dart' and 'models/models.dart'. It defines a StatelessWidget 'RecipeThumbnail' with a 'key' parameter. Inside the build method, it creates a Container with padding and a Column child. The Column has a crossAxisAlignment of 'start' and a single child, which is an Expanded widget containing a ClipRRect with a BoxFit.cover fit and a circular border radius. Below this is a SizedBox with height 10. Then there are two Text widgets: one for the recipe title (maxLines: 1) and one for the duration. The code uses 'Theme.of(context).textTheme.bodyText1' for both. The bottom of the screen shows a dock with various application icons.

```
812     fontWeight: FontWeight.w900,
813     decoration: textDecoration,
814   ),
815 );
816 } else {
817   throw Exception('This importance type does not exist');
818 }
819 }
820 }
821
822 Widget buildDate() {
823   final dateFormatter = DateFormat('MMMM dd h:mm a');
824   final dateString = dateFormatter.format(item.date);
825   return Text(
826     dateString,
827     style: TextStyle(decoration: textDecoration),
828   );
829 }
830
831 Widget buildCheckbox() {
832   return Checkbox(
833 // 1
834     value: item.isComplete,
835 // 2
836     onChanged: onComplete,
837   );
838 }
839 ]
840
841
842 recipe_thumbnail.dart
843 import 'package:flutter/material.dart';
844 import '../models/models.dart';
845
846 class RecipeThumbnail extends StatelessWidget {
847 // 1
848   final SimpleRecipe recipe;
849
850   const RecipeThumbnail({
851     Key? key,
852     required this.recipe,
853   }) : super(key: key);
854
855 @override
856 Widget build(BuildContext context) {
857 // 2
858   return Container(
859     padding: const EdgeInsets.all(8),
860 // 3
861     child: Column(
862       crossAxisAlignment: CrossAxisAlignment.start,
863       children: [
864
865 // 4
866       Expanded(
867         child: ClipRRect(
868           child: Image.asset(
869             '${recipe.dishImage}',
870             fit: BoxFit.cover,
871           ),
872           borderRadius: BorderRadius.circular(12),
873         ),
874       ),
875       const SizedBox(height: 10),
876 // 7
877       Text(
878         recipe.title,
879         maxLines: 1,
880         style: Theme.of(context).textTheme.bodyText1,
881       ),
882       Text(
883         recipe.duration,
884         style: Theme.of(context).textTheme.bodyText1,
885       ),
886     ],
887   ),
888 }
889 }
```

Fig 55.

A screenshot of a Flutter IDE interface, identical to Fig 55 but with more code visible. The top bar shows 'Activities' and 'Text Editor'. The status bar indicates 'May 3 11:01' and 'Untitled Document 1'. The main area contains Dart code for the 'RecipeThumbnail' widget. The code imports 'package:flutter/material.dart' and 'models/models.dart'. It defines a StatelessWidget 'RecipeThumbnail' with a 'key' parameter. Inside the build method, it creates a Container with padding and a Column child. The Column has a crossAxisAlignment of 'start' and a single child, which is an Expanded widget containing a ClipRRect with a BoxFit.cover fit and a circular border radius. Below this is a SizedBox with height 10. Then there are two Text widgets: one for the recipe title (maxLines: 1) and one for the duration. The code uses 'Theme.of(context).textTheme.bodyText1' for both. The bottom of the screen shows a dock with various application icons.

```
840
841
842 recipe_thumbnail.dart
843 import 'package:flutter/material.dart';
844 import '../models/models.dart';
845
846 class RecipeThumbnail extends StatelessWidget {
847 // 1
848   final SimpleRecipe recipe;
849
850   const RecipeThumbnail({
851     Key? key,
852     required this.recipe,
853   }) : super(key: key);
854
855 @override
856 Widget build(BuildContext context) {
857 // 2
858   return Container(
859     padding: const EdgeInsets.all(8),
860 // 3
861     child: Column(
862       crossAxisAlignment: CrossAxisAlignment.start,
863       children: [
864
865 // 4
866       Expanded(
867         child: ClipRRect(
868           child: Image.asset(
869             '${recipe.dishImage}',
870             fit: BoxFit.cover,
871           ),
872           borderRadius: BorderRadius.circular(12),
873         ),
874       ),
875       const SizedBox(height: 10),
876 // 7
877       Text(
878         recipe.title,
879         maxLines: 1,
880         style: Theme.of(context).textTheme.bodyText1,
881       ),
882       Text(
883         recipe.duration,
884         style: Theme.of(context).textTheme.bodyText1,
885       ),
886     ],
887   );
888 }
```

Fig 56.

```
866 // 5
867     child: ClipRect(
868         child: Image.asset(
869             '${recipe.dishImage}',
870             fit: BoxFit.cover,
871         ),
872         borderRadius: BorderRadius.circular(12),
873     ),
874 ),
875 const SizedBox(height: 10),
876 // 7
877 Text(
878     recipe.title,
879     maxLines: 1,
880     style: Theme.of(context).textTheme.bodyText1,
881 ),
882 Text(
883     recipe.duration,
884     style: Theme.of(context).textTheme.bodyText1,
885 )
886 ],
887 ),
888 );
889 }
890 }
891
892 recipes_grid_view.dart
893 import 'package:flutter/material.dart';
894 import '../components/components.dart';
895 import '../models/models.dart';
896
897 class RecipesGridView extends StatelessWidget {
898 // 1
899 final List<SimpleRecipe> recipes;
900
901 const RecipesGridView({
902     Key? key,
903     required this.recipes,
904 }) : super(key: key);
905
906 @override
907 Widget build(BuildContext context) {
908 // 2
909     return Padding(
910         padding: const EdgeInsets.only(
911             left: 16,
912             right: 16,
913             top: 16,
914         ),
915 // 3
916         child: GridView.builder(

```

Fig 57.

```
892 recipes_grid_view.dart
893 import 'package:flutter/material.dart';
894 import '../components/components.dart';
895 import '../models/models.dart';
896
897 class RecipesGridView extends StatelessWidget {
898 // 1
899 final List<SimpleRecipe> recipes;
900
901 const RecipesGridView({
902     Key? key,
903     required this.recipes,
904 }) : super(key: key);
905
906 @override
907 Widget build(BuildContext context) {
908 // 2
909     return Padding(
910         padding: const EdgeInsets.only(
911             left: 16,
912             right: 16,
913             top: 16,
914         ),
915 // 3
916         child: GridView.builder(
917 // 4
918             itemCount: recipes.length,
919 // 5
920             gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
921                 crossAxisCount: 2,
922             ),
923             itemBuilder: (context, index) {
924 // 6
925                 final simpleRecipe = recipes[index];
926                 return RecipeThumbnail(recipe: simpleRecipe);
927             },
928         );
929     );
930 }
931
932
933 today_recipe_list_view.dart
934 import 'package:cookipedia/cockipedia_theme.dart';
935 import 'package:flutter/material.dart';
936
937
938 // 1
939 import '../components/components.dart';
940 import '../models/models.dart';
941
942 class TodayRecipeListView extends StatelessWidget {

```

Fig 58.

A screenshot of a code editor window titled "Text Editor". The file is named "Untitled Document 1" and was last modified on May 3 at 11:01. The code is written in Dart and defines a class named "TodayRecipeListView" which extends StatelessWidget. The class has a constructor that takes a key and a required list of recipes. It overrides the build method to return a Padding widget with a const EdgeInsets.only(left: 16, right: 16, top: 16). Inside this padding, there is a Column widget with crossAxisAlignment: CrossAxisAlignment.start. The first child of the column is a Text widget with the text "Recipes of the Day" and a style of Theme.of(context).textTheme.headline2. The second child is a ListView.separated widget. This ListView has a scrollDirection of Axis.horizontal, an itemCount of recipes.length, and an itemBuilder that takes a context and index, returning a buildCard(recipe) for each item. A separatorBuilder is also provided for each item.

```
Activities Text Editor
Open *Untitled Document 1
May 3 11:01
910   itemCount: recipes.length,
911 // 5
912   gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
913     crossAxisCount: 2,
914   ),
915   itemBuilder: (context, index) {
916 // 6
917     final simpleRecipe = recipes[index];
918     return RecipeThumbnail(recipe: simpleRecipe);
919   },
920 },
921 ),
922 itemBuilder: (context, index) {
923   final simpleRecipe = recipes[index];
924   return RecipeThumbnail(recipe: simpleRecipe);
925 },
926 ),
927 ),
928 itemBuilder: (context, index) {
929   final simpleRecipe = recipes[index];
930   return RecipeThumbnail(recipe: simpleRecipe);
931 },
932 ),
933 itemBuilder: (context, index) {
934   final simpleRecipe = recipes[index];
935   return RecipeThumbnail(recipe: simpleRecipe);
936 },
937 ),
938 // 1
939 import '../components/components.dart';
940 import '../models/models.dart';
941 
```

Fig 59.

A screenshot of a code editor window titled "Text Editor". The file is named "Untitled Document 1" and was last modified on May 3 at 11:01. The code is identical to Fig 59, defining a "TodayRecipeListView" class that extends StatelessWidget. It has a constructor for a key and a required list of recipes. The build method returns a Padding with EdgeInsets.only(left: 16, right: 16, top: 16), followed by a Column with CrossAxisAlignment.start. The first child is a Text "Recipes of the Day" in headline2 style. The second child is a ListView.separated with horizontal scroll direction, itemCount equal to the length of the recipes list, and an itemBuilder that builds cards for each recipe. A separatorBuilder is also present.

```
Activities Text Editor
Open *Untitled Document 1
May 3 11:01
945   const TodayRecipeListView({
946     Key? key,
947     required this.recipes,
948   }) : super(key: key);
949 
```

Fig 60.

A screenshot of a code editor window titled "Untitled Document 1". The code is written in Dart. It includes a UI component definition with nested containers, scroll directions, and item counts, followed by a method to build cards based on recipe types. Below this is a class named AppCache that handles shared preferences. The code uses await and async keywords throughout. The status bar at the bottom shows "Dart Tab Width: B Ln 19, Col 1 INS".

```
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
971 // 7
972 Container(
973   height: 400,
974   // ...
975   color: Colors.transparent,
976 // 2
977   child: ListView.separated(
978 // 3
979     scrollDirection: Axis.horizontal,
980 // 4
981     itemCount: recipes.length,
982 // 5
983     itemBuilder: (context, index) {
984 // 6
985       final recipe = recipes[index];
986       return buildCard(recipe);
987     },
988 // 7
989     separatorBuilder: (context, index) {
990 // 8
991       return const SizedBox(width: 16);
992     },
993   ),
994 ],
995 ],
996 ),
997 );
998 }
999 Widget buildCard(ExploreRecipe recipe) {
1000   if (recipe.cardType == RecipeCardType.card1) {
1001     return Card1(recipe: recipe);
1002   } else if (recipe.cardType == RecipeCardType.card2) {
1003     return Card2(recipe: recipe);
1004   } else if (recipe.cardType == RecipeCardType.card3) {
1005     return Card3(recipe: recipe);
1006   } else {
1007     throw Exception('This card doesn\'t exist yet');
1008   }
1009 }
1010 ]
1011
1012
1013 app_cache.dart
1014 import 'package:shared_preferences/shared_preferences.dart';
1015
1016 class AppCache {
1017   static const kUser = 'user';
1018   static const kOnboarding = 'onboarding';
1019
1020 Future<void> invalidate() async {
1021   final prefs = await SharedPreferences.getInstance();
1022
1023   });
1024
1025   Widget buildCard(ExploreRecipe recipe) {
1026     if (recipe.cardType == RecipeCardType.card1) {
1027       return Card1(recipe: recipe);
1028     } else if (recipe.cardType == RecipeCardType.card2) {
1029       return Card2(recipe: recipe);
1030     } else if (recipe.cardType == RecipeCardType.card3) {
1031       return Card3(recipe: recipe);
1032     } else {
1033       throw Exception('This card doesn\'t exist yet');
1034     }
1035   }
1036
1037   Future<void> cacheUser() async {
1038     final prefs = await SharedPreferences.getInstance();
1039     await prefs.setBool(kUser, true);
1040
1041   Future<bool> completeOnboarding() async {
1042     final prefs = await SharedPreferences.getInstance();
1043     return prefs.getBool(kOnboarding) ?? false;
1044   }
1045 }
```

Fig 61.

A screenshot of a code editor window titled "Untitled Document 1". The code is identical to Fig 61, showing the same Dart code for a UI component and a shared preferences class. The status bar at the bottom shows "Dart Tab Width: B Ln 19, Col 1 INS".

```
May 3 11:01
*Untitled Document 1
Save
Activities Text Editor
Open
997   );
998 }
999 Widget buildCard(ExploreRecipe recipe) {
1000   if (recipe.cardType == RecipeCardType.card1) {
1001     return Card1(recipe: recipe);
1002   } else if (recipe.cardType == RecipeCardType.card2) {
1003     return Card2(recipe: recipe);
1004   } else if (recipe.cardType == RecipeCardType.card3) {
1005     return Card3(recipe: recipe);
1006   } else {
1007     throw Exception('This card doesn\'t exist yet');
1008   }
1009 }
1010 ]
1011
1012
1013 app_cache.dart
1014 import 'package:shared_preferences/shared_preferences.dart';
1015
1016 class AppCache {
1017   static const kUser = 'user';
1018   static const kOnboarding = 'onboarding';
1019
1020 Future<void> invalidate() async {
1021   final prefs = await SharedPreferences.getInstance();
1022   await prefs.setBool(kUser, false);
1023   await prefs.setBool(kOnboarding, false);
1024 }
1025
1026 Future<void> cacheUser() async {
1027   final prefs = await SharedPreferences.getInstance();
1028   await prefs.setBool(kUser, true);
1029 }
1030
1031 Future<void> completeOnboarding() async {
1032   final prefs = await SharedPreferences.getInstance();
1033   await prefs.setBool(kOnboarding, true);
1034 }
1035
1036 Future<bool> isUserLoggedIn() async {
1037   final prefs = await SharedPreferences.getInstance();
1038   return prefs.getBool(kUser) ?? false;
1039 }
1040
1041 Future<bool> didCompleteOnboarding() async {
1042   final prefs = await SharedPreferences.getInstance();
1043   return prefs.getBool(kOnboarding) ?? false;
1044 }
1045 }
```

Fig 62.

A screenshot of a Linux desktop environment showing a code editor window. The window title is "Text Editor" and the document title is "*Untitled Document 1". The date and time at the top right are May 3 11:01. The code editor displays Dart code for a Flutter application. The code includes imports for flutter/material.dart and app_cache.dart, defines a CookipidiaTab class with static constants for explore, recipes, and toBuy counts, and an AppStateManager class that extends ChangeNotifier. It handles initialization, logging in, completing onboarding, and navigating tabs. A Timer is used to set _initialized to true after 2000 milliseconds. Methods like login, completeOnboarding, and goToTab are asynchronous. The code editor interface includes tabs for Dart, Tab Width: 8, Line 19, Column 1, and Insert mode (INS). The bottom of the screen shows a dock with various application icons.

```
1050 import 'package:flutter/material.dart';
1051 import 'app_cache.dart';
1052
1053 class CookipidiaTab {
1054   static const int explore = 0;
1055   static const int recipes = 1;
1056   static const int toBuy = 2;
1057 }
1058
1059 }
1060
1061 class AppStateManager extends ChangeNotifier {
1062   bool _initialized = false;
1063   bool _loggedin = false;
1064   bool _onboardingComplete = false;
1065   int _selectedTab = CookipidiaTab.explore;
1066   final _appCache = AppCache();
1067
1068   bool get isInitialized => _initialized;
1069   bool get isLoggedIn => _loggedin;
1070   bool get isOnboardingComplete => _onboardingComplete;
1071   int get selectedTab => _selectedTab;
1072
1073   void initializeApp() async {
1074     _loggedin = await _appCache.isUserLoggedIn();
1075     _onboardingComplete = await _appCache.didCompleteOnboarding();
1076
1077     Timer(
1078       const Duration(milliseconds: 2000),
1079       () {
1080         _initialized = true;
1081         notifyListeners();
1082       },
1083     );
1084   }
1085
1086   void login(String username, String password) async {
1087     _loggedin = true;
1088     await _appCache.cacheUser();
1089     notifyListeners();
1090   }
1091
1092   void completeOnboarding() async {
1093     _onboardingComplete = true;
1094     await _appCache.completeOnboarding();
1095     notifyListeners();
1096   }
1097
1098   void goToTab(index) {
1099     _selectedTab = index;
1100     notifyListeners();
1101   }
1102
1103   void goToRecipes() {
1104     _selectedTab = CookipidiaTab.recipes;
1105     notifyListeners();
1106   }
1107
1108   void logout() async {
1109     _initialized = false;
1110     _selectedTab = 0;
1111     await _appCache.invalidate();
1112
1113     initializeApp();
1114     notifyListeners();
1115   }
1116 }
1117
1118 cookipidia_pages.dart
1119 class CookipidiaPages {
1120   static String splashPath = '/splash';
1121   static String loginPath = '/login';
1122   static String onboardingPath = '/onboarding';
1123   static String home = '/';
1124   static String groceryItemDetails = '/item';
1125   static String profilePath = '/profile';
1126 }
```

Fig 63.

A screenshot of a Linux desktop environment showing a code editor window. The window title is "Text Editor" and the document title is "*Untitled Document 1". The date and time at the top right are May 3 11:01. The code editor displays Dart code for a Flutter application. The code includes imports for flutter/material.dart and app_cache.dart, defines a CookipidiaTab class with static constants for splash, login, onboarding, home, item, and profile paths, and an AppStateManager class that extends ChangeNotifier. It handles initialization, logging in, navigating pages, and invalidating the cache. A Timer is used to set _initialized to true after 2000 milliseconds. Methods like goToPage, goToProfile, and logout are asynchronous. The code editor interface includes tabs for Dart, Tab Width: 8, Line 19, Column 1, and Insert mode (INS). The bottom of the screen shows a dock with various application icons.

```
1070
1071   Timer(
1072     const Duration(milliseconds: 2000),
1073     () {
1074       _initialized = true;
1075       notifyListeners();
1076     },
1077   );
1078
1079   void login(String username, String password) async {
1080     _loggedin = true;
1081     await _appCache.cacheUser();
1082     notifyListeners();
1083   }
1084
1085   void completeOnboarding() async {
1086     _onboardingComplete = true;
1087     await _appCache.completeOnboarding();
1088     notifyListeners();
1089   }
1090
1091   void goToPage(String path) {
1092     _selectedTab = CookipidiaTab[path];
1093     notifyListeners();
1094   }
1095
1096   void goToProfile() {
1097     _selectedTab = CookipidiaTab.profile;
1098     notifyListeners();
1099   }
1100
1101   void logout() async {
1102     _initialized = false;
1103     _selectedTab = 0;
1104     await _appCache.invalidate();
1105
1106     initializeApp();
1107     notifyListeners();
1108   }
1109
1110 }
1111
1112 cookipidia_pages.dart
1113 class CookipidiaPages {
1114   static String splashPath = '/splash';
1115   static String loginPath = '/login';
1116   static String onboardingPath = '/onboarding';
1117   static String home = '/';
1118   static String groceryItemDetails = '/item';
1119   static String profilePath = '/profile';
1120 }
```

Fig 64.

```
Activities Text Editor
Open Untitled Document 1
May 3 11:01
1103 void goToRecipes() {
1104     _selectedTab = CookipediaTab.recipes;
1105     notifyListeners();
1106 }
1107
1108 void logout() async {
1109     _initialized = false;
1110     _selectedTab = 0;
1111     await _appCache.invalidate();
1112     initializeApp();
1113     notifyListeners();
1114 }
1115 }
1116
1117
1118 cookipedia_pages.dart
1119 class CookipediaPages {
1120     static String splashPath = '/splash';
1121     static String loginPath = '/login';
1122     static String onboardPath = '/onboarding';
1123     static String home = '/';
1124     static String groceryItemDetails = '/item';
1125     static String profilePath = '/profile';
1126     static String raywenderlich = '/raywenderlich';
1127     static String accountPath = '/account';
1128 }
1129
1130
1131
1132 explore_data.dart
1133 import 'models.dart';
1134
1135 class ExploreData {
1136     final List<ExploreRecipe> todayRecipes;
1137     final List<Post> friendPosts;
1138 }
1139
1140 ExploreData(
1141     this.todayRecipes,
1142     this.friendPosts,
1143 );
1144
1145
1146 explore_recipe.dart
1147 part 'ingredient.dart';
1148 part 'instruction.dart';
1149
1150
1151 class RecipeCardType {
1152     static const card1 = 'card1';
1153     static const card2 = 'card2';
1154 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 65.

```
Activities Text Editor
Open Untitled Document 1
May 3 11:01
Save
May 3 11:01
*Untitled Document 1
1129 ]
1130
1131
1132 explore_data.dart
1133 import 'models.dart';
1134
1135 class ExploreData {
1136     final List<ExploreRecipe> todayRecipes;
1137     final List<Post> friendPosts;
1138 }
1139
1140 ExploreData(
1141     this.todayRecipes,
1142     this.friendPosts,
1143 );
1144
1145
1146 explore_recipe.dart
1147 part 'ingredient.dart';
1148 part 'instruction.dart';
1149
1150
1151 class RecipeCardType {
1152     static const card1 = 'card1';
1153     static const card2 = 'card2';
1154     static const card3 = 'card3';
1155 }
1156
1157 class ExploreRecipe {
1158     String id;
1159     String cardType;
1160     String title;
1161     String subtitle;
1162     String backgroundImage;
1163     String backgroundImageSource;
1164     String message;
1165     String authorName;
1166     String authorRole;
1167     String profileImage;
1168     int durationInMinutes;
1169     String dietType;
1170     int calories;
1171     List<String> tags;
1172     String description;
1173     String source;
1174     List<Ingredients> ingredients;
1175     List<Instruction> instructions;
1176
1177     ExploreRecipe({
1178         required this.id,
1179         required this.cardType,
1180     });
1181 }
```

Dart Tab Width: 8 Ln 19, Col 1 INS

Fig 66.

Activities Text Editor

May 3 11:01

*Untitled Document 1

Save  

```
1159 }
1160
1161 class ExploreRecipe {
1162   String id;
1163   String cardType;
1164   String title;
1165   String subtitle;
1166   String backgroundImage;
1167   String backgroundImageSource;
1168   String message;
1169   String authorName;
1170   String role;
1171   String profileImage;
1172   int durationInMinutes;
1173   String dietType;
1174   int calories;
1175   List<String> tags;
1176   String description;
1177   String source;
1178   List<Ingredients> ingredients;
1179   List<Instruction> instructions;
1180
1181 ExploreRecipe({
1182   required this.id,
1183   required this.cardType,
1184   required this.title,
1185   this.subtitle = '',
1186   this.backgroundImage = '',
1187   this.backgroundImageSource = '',
1188   this.message = '',
1189   this.authorName = '',
1190   this.role = '',
1191   this.profileImage = '',
1192   this.durationInMinutes = 0,
1193   this.dietType = '',
1194   this.calories = 0,
1195   this.tags = const [],
1196   this.description = '',
1197   this.source = '',
1198   this.ingredients = const [],
1199   this.instructions = const [],
1200 });
1201
1202 factory ExploreRecipe.fromJson(Map<String, dynamic> json) {
1203   final ingredients = <Ingredients>[];
1204   final instructions = <Instruction>[];
1205
1206   if (json['ingredients'] != null) {
1207     json['ingredients'].forEach((v) {
1208       ingredients.add(Ingredients.fromJson(v));
1209     });
1210   }
1211 }
```

Save  Tab Width: 8  Ln 19, Col 1  INS 

Fig 67.

Activities Text Editor May 3 11:01 *Untitled Document 1 Save

```
1182 this.cardType = '',
1183 this.backgroundImage = '',
1184 this.message = '',
1185 this.authorName = '',
1186 this.role = '',
1187 this.profileImage = '',
1188 this.durationInMinutes = 0,
1189 this.dietType = '',
1190 this.calories = 0,
1191 this.tags = const [],
1192 this.description = '',
1193 this.source = '',
1194 this.ingredients = const [],
1195 this.instructions = const [],
1196 );
1197
1198 factory ExploreRecipe.fromJson(Map<String, dynamic> json) {
1199   final ingredients = <Ingredient>[];
1200   final instructions = <Instruction>[];
1201
1202   if (json['ingredients'] != null) {
1203     json['ingredients'].forEach((v) {
1204       ingredients.add(Ingredient.fromJson(v));
1205     });
1206   }
1207
1208   if (json['instructions'] != null) {
1209     json['instructions'].forEach((v) {
1210       instructions.add(Instruction.fromJson(v));
1211     });
1212   }
1213
1214   return ExploreRecipe(
1215     id: json['id'] ?? '',
1216     cardType: json['cardType'] ?? '',
1217     title: json['title'] ?? '',
1218     subtitle: json['subtitle'] ?? '',
1219     backgroundImage: json['backgroundImage'] ?? '',
1220     backgroundImageSource: json['backgroundImageSource'] ?? '',
1221     message: json['message'] ?? '',
1222     authorName: json['authorName'] ?? '',
1223     role: json['role'] ?? '',
1224     profileImage: json['profileImage'] ?? '',
1225     durationInMinutes: json['durationInMinutes'] ?? 0,
1226     dietType: json['dietType'] ?? '',
1227     calories: json['calories'] ?? 0,
1228     tags: json['tags'].cast<String>() ?? [],
1229     description: json['description'] ?? '',
1230     source: json['source'] ?? '',
1231     ingredients: ingredients,
1232   );
1233 }
```

Save Tab Width: 8 Ln 19, Col 1 INS

Fig 68.

A screenshot of a code editor window titled "Text Editor". The code is written in Dart and defines a class named `ExploreRecipe`. The code includes imports for `dart:convert` and `package:flutter/painting.dart`, and defines an enum `Importance` with values `low`, `medium`, and `high`. The `ExploreRecipe` class has properties like `id`, `cardType`, `title`, `subtitle`, `backgroundImage`, `backgroundImageSource`, `message`, `authorName`, `role`, `profileImage`, `durationInMinutes`, `dtimeType`, `calories`, `tags`, `description`, `source`, `ingredients`, and `instructions`. It also includes a constructor and a copyWith method.

```
1288     if (json['instructions'] != null) {
1289         json['instructions'].forEach((v) {
1290             instructions.add(Instruction.fromJson(v));
1291         });
1292     }
1293 
1294     return ExploreRecipe(
1295         id: json['id'] ?? '',
1296         cardType: json['cardtype'] ?? '',
1297         title: json['title'] ?? '',
1298         subtitle: json['subtitle'] ?? '',
1299         backgroundImage: json['backgroundImage'] ?? '',
1300         backgroundImageSource: json['backgroundImageSource'] ?? '',
1301         message: json['message'] ?? '',
1302         authorName: json['authorname'] ?? '',
1303         role: json['role'] ?? '',
1304         profileImage: json['profileImage'] ?? '',
1305         durationInMinutes: json['durationinminutes'] ?? 0,
1306         dtimeType: json['dttimetype'] ?? '',
1307         calories: json['calories'] ?? 0,
1308         tags: json['tags'].cast<String>() ?? [],
1309         description: json['description'] ?? '',
1310         source: json['source'] ?? '',
1311         ingredients: ingredients,
1312         instructions: instructions,
1313     );
1314 }
1315 }
1316 
1317 grocery_item.dart
1318 import 'package:flutter/painting.dart';
1319 
1320 // 1
1321 enum Importance {
1322     low,
1323     medium,
1324     high,
1325 }
1326 
1327 class GroceryItem {
1328     // 2
1329     final String id;
1330 
1331     // 3
1332     final String name;
1333     final Importance importance;
1334     final Color color;
1335     final int quantity;
1336     final DateTime date;
1337     final bool isComplete;
1338 }
```

Fig 69.

A screenshot of a code editor window titled "Text Editor". The code is written in Dart and defines a class named `GroceryItem`. It includes imports for `dart:convert` and `package:flutter/painting.dart`, and defines an enum `Importance` with values `low`, `medium`, and `high`. The `GroceryItem` class has properties `id`, `name`, `importance`, `color`, `quantity`, `date`, and `isComplete`. It includes a constructor and a `copyWith` method that creates a new `GroceryItem` object with updated values.

```
1234 }
1235 }
1236 
1237 grocery_item.dart
1238 import 'package:flutter/painting.dart';
1239 
1240 // 1
1241 enum Importance {
1242     low,
1243     medium,
1244     high,
1245 }
1246 
1247 class GroceryItem {
1248     // 2
1249     final String id;
1250 
1251     // 3
1252     final String name;
1253     final Importance importance;
1254     final Color color;
1255     final int quantity;
1256     final DateTime date;
1257     final bool isComplete;
1258 
1259     GroceryItem({
1260         required this.id,
1261         required this.name,
1262         required this.importance,
1263         required this.color,
1264         required this.quantity,
1265         required this.date,
1266         this.isComplete = false,
1267     });
1268 
1269     GroceryItem copyWith({
1270         String? id,
1271         String? name,
1272         Importance? importance,
1273         Color? color,
1274         int? quantity,
1275         DateTime? date,
1276         bool? isComplete,
1277     }) {
1278         return GroceryItem(
1279             id: id ?? this.id,
1280             name: name ?? this.name,
1281             importance: importance ?? this.importance,
1282             color: color ?? this.color,
1283             quantity: quantity ?? this.quantity,
1284         );
1285     }
1286 }
```

Fig 70.

```
Activities Text Editor *Untitled Document 1 May 3 11:01 Save
Open ⌂ ⌂ ⌂ *Untitled Document 1
1260     GroceryItem<{
1261         required this.id,
1262         required this.name,
1263         required this.importance,
1264         required this.color,
1265         required this.quantity,
1266         required this.date,
1267         this.isComplete = false,
1268     }>;
1269
1270     GroceryItem copyWith({
1271         String? id,
1272         String? name,
1273         Importance? importance,
1274         Color? color,
1275         int? quantity,
1276         DateTime? date,
1277         bool? isComplete,
1278     }) {
1279         return GroceryItem(
1280             id: id ?? this.id,
1281             name: name ?? this.name,
1282             importance: importance ?? this.importance,
1283             color: color ?? this.color,
1284             quantity: quantity ?? this.quantity,
1285             date: date ?? this.date,
1286             isComplete: isComplete ?? this.isComplete,
1287         );
1288     }
1289 }
1290
1291 grocery_manager.dart
1292 import 'package:flutter/material.dart';
1293
1294 import 'grocery_item.dart';
1295
1296 class GroceryManager extends ChangeNotifier {
1297     final _groceryItems = <GroceryItem>[];
1298     int _selectedIndex = -1;
1299     bool _createNewItem = false;
1300
1301     List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);
1302     int get selectedIndex => _selectedIndex;
1303     GroceryItem? get selectedGroceryItem =>
1304         _selectedIndex != -1 ? _groceryItems[_selectedIndex] : null;
1305     bool get isCreatingNewItem => _createNewItem;
1306
1307     void createNewItem() {
1308         _createNewItem = true;
1309         notifyListeners();
1310     }
1311
1312     void deleteItem(int index) {
1313         _groceryItems.removeAt(index);
1314         notifyListeners();
1315     }
1316
1317     void groceryItemTapped(int index) {
1318         _selectedIndex = index;
1319         _createNewItem = false;
1320         notifyListeners();
1321     }
1322
1323     void addNewItem(GroceryItem item) {
1324         _groceryItems.add(item);
1325         _createNewItem = false;
1326         notifyListeners();
1327     }
1328
1329     void updateItem(GroceryItem item, int index) {
1330         _groceryItems[index] = item;
1331         _selectedIndex = -1;
1332         _createNewItem = false;
1333         notifyListeners();
1334     }
1335
1336     void completeItem(int index, bool change) {
1337 }
```

Fig 71.

```
Activities Text Editor *Untitled Document 1 May 3 11:01 Save
Open ⌂ ⌂ ⌂ *Untitled Document 1
1260     isComplete: isComplete ?? this.isComplete,
1261 }
1262 }
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291 grocery_manager.dart
1292 import 'package:flutter/material.dart';
1293
1294 import 'grocery_item.dart';
1295
1296 class GroceryManager extends ChangeNotifier {
1297     final _groceryItems = <GroceryItem>[];
1298     int _selectedIndex = -1;
1299     bool _createNewItem = false;
1300
1301     List<GroceryItem> get groceryItems => List.unmodifiable(_groceryItems);
1302     int get selectedIndex => _selectedIndex;
1303     GroceryItem? get selectedGroceryItem =>
1304         _selectedIndex != -1 ? _groceryItems[_selectedIndex] : null;
1305     bool get isCreatingNewItem => _createNewItem;
1306
1307     void createNewItem() {
1308         _createNewItem = true;
1309         notifyListeners();
1310     }
1311
1312     void deleteItem(int index) {
1313         _groceryItems.removeAt(index);
1314         notifyListeners();
1315     }
1316
1317     void groceryItemTapped(int index) {
1318         _selectedIndex = index;
1319         _createNewItem = false;
1320         notifyListeners();
1321     }
1322
1323     void addNewItem(GroceryItem item) {
1324         _groceryItems.add(item);
1325         _createNewItem = false;
1326         notifyListeners();
1327     }
1328
1329     void updateItem(GroceryItem item, int index) {
1330         _groceryItems[index] = item;
1331         _selectedIndex = -1;
1332         _createNewItem = false;
1333         notifyListeners();
1334     }
1335
1336     void completeItem(int index, bool change) {
1337 }
```

Fig 72.

A screenshot of a Linux desktop environment. In the foreground, a code editor window titled "Untitled Document 1" is open, showing Dart code for a grocery item manager. The code includes methods for deleting items, tapping items, adding items, updating items, and completing items. It also defines a class for Ingredients and a factory method to create them from JSON. The code editor interface includes tabs for "Activities" and "Text Editor", a toolbar with icons for Open, Save, and Print, and a status bar at the bottom.

```
1313 void deleteItem(int index) {
1314   _groceryItems.removeAt(index);
1315   notifyListeners();
1316 }
1317
1318 void groceryItemTapped(int index) {
1319   _selectedIndex = index;
1320   _createNewItem = false;
1321   notifyListeners();
1322 }
1323
1324 void addItem(GroceryItem item) {
1325   _groceryItems.add(item);
1326   _createNewItem = false;
1327   notifyListeners();
1328 }
1329
1330 void updateItem(GroceryItem item, int index) {
1331   _groceryItems[index] = item;
1332   _selectedIndex = -1;
1333   _createNewItem = false;
1334   notifyListeners();
1335 }
1336
1337 void completeItem(int index, bool change) {
1338   final item = _groceryItems[index];
1339   _groceryItems[index] = item.copyWith(isComplete: change);
1340   notifyListeners();
1341 }
1342
1343
1344 ingredient.dart
1345 part of 'explore_recipe.dart';
1346
1347 class Ingredients {
1348   String imageUrl;
1349   String title;
1350   String source;
1351
1352   Ingredients({
1353     required this.imageUrl,
1354     required this.title,
1355     required this.source,
1356   });
1357
1358 factory Ingredients.fromJson(Map<String, dynamic> json) {
1359   return Ingredients(
1360     imageUrl: json['imageUrl'] ?? '',
1361     title: json['title'] ?? '',
1362     source: json['source'] ?? '',
1363   );
1364 }
```

Fig 73.

A screenshot of a Linux desktop environment. In the foreground, a code editor window titled "Untitled Document 1" is open, showing Dart code for a recipe application. The code includes a class for Ingredients and a factory method to create them from JSON. It also includes a class for Instruction and a factory method to create it from JSON. The code editor interface includes tabs for "Activities" and "Text Editor", a toolbar with icons for Open, Save, and Print, and a status bar at the bottom.

```
1339   _groceryItems[index] = item.copyWith(isComplete: change);
1340   notifyListeners();
1341 }
1342
1343
1344 ingredient.dart
1345 part of 'explore_recipe.dart';
1346
1347 class Ingredients {
1348   String imageUrl;
1349   String title;
1350   String source;
1351
1352   Ingredients({
1353     required this.imageUrl,
1354     required this.title,
1355     required this.source,
1356   });
1357
1358 factory Ingredients.fromJson(Map<String, dynamic> json) {
1359   return Ingredients(
1360     imageUrl: json['imageUrl'] ?? '',
1361     title: json['title'] ?? '',
1362     source: json['source'] ?? '',
1363   );
1364 }
1365
1366 }
1367
1368 instruction.dart
1369 part of 'explore_recipe.dart';
1370
1371 class Instruction {
1372   String imageUrl;
1373   String description;
1374   int durationInMinutes;
1375
1376   Instruction({
1377     required this.imageUrl,
1378     required this.description,
1379     required this.durationInMinutes,
1380   });
1381
1382 factory Instruction.fromJson(Map<String, dynamic> json) {
1383   return Instruction(
1384     imageUrl: json['imageUrl'] ?? '',
1385     description: json['description'] ?? '',
1386     durationInMinutes: json['durationInMinutes'] ?? '',
1387   );
1388 }
```

Fig 74.

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Activities, Android Studio, May 3 11:08.
- File Menu:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, Git, Window, Help.
- Project Tree:** cookipidia (AndroidStudioProjects/cookipidia) -> lib/api/mock_cookipidia_service.dart
- Code Editor:** Content pane displays the file `mock_cookipidia_service.dart`. The code defines a `MockCookipidiaService` class with methods for `getExploreData` and `_getTodayRecipes`.
- Toolbars and Status Bar:** Includes Git, TODO, Problems, Terminal, Dart Analysis, Logcat, Profiler, and Layout Inspector. The status bar shows Failed to start monitoring 32000bcaba01950d (50 minutes ago), 18:57, LF, UTF-8, 2 spaces, and master.
- Systray:** Shows battery level, signal strength, and volume controls.

Fig 75.

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Activities, Android Studio, May 3 11:08, cookipidia - card1.dart [cookipidia]
- File Menu:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, Git, Window, Help
- Toolbar:** Project, Build, Run, Tools, Git, Layout Inspector, Device Manager, Filter Inspector, Emulator, Device File Explorer.
- Project Structure:** cookipidia (Android Studio Projects), build, lib, api, mock_cookipidia_service.dart, components (author_card.dart, card1.dart, card2.dart, card3.dart, circle_image.dart, components.dart, friend_post_list_view.dart, friend_tile.dart, grocery_tile.dart, recipe_thumbnail.dart, recipes_grid_view.dart, today_recipe_list_view.dart), models (app_cache.dart, app_state_manager.dart, cookipidia_pages.dart, explore_data.dart, explore_recipe.dart, grocery_item.dart, grocery_manager.dart, ingredient.dart, instruction.dart, models.dart, post.dart, profile_manager.dart, simple_recipe.dart, user_manager.dart).
- Code Editor:** Card1.dart (selected). The code defines a StatelessWidget 'Card1' that displays a recipe card with title, subtitle, and message. It uses dark mode styling from 'CookipidiaTheme'. The code includes imports for package:cookipidia/cookipidia_theme.dart and ../models/explore_recipe.dart.
- Bottom Status:** Git, TODO, Problems, Terminal, Dart Analysis, Logcat, App Inspection, Failed to start monitoring 32000bcaba01950d (50 minutes ago), Event Log, Layout Inspector, 5528 LF, UTF-8, 2 spaces, master.

Fig 76.

The screenshot shows the Android Studio interface with the project 'cookipidia' open. The code editor displays the file 'card2.dart' under the 'lib/components' directory. The code defines a StatelessWidget named 'Card2' that extends 'ExploreRecipe'. It includes imports for 'package:flutter/material.dart', 'models/explore_recipe.dart', 'author_card.dart', and 'cookipidia_theme.dart'. The widget's build method creates a Centered Container with a height of 450 pixels, containing a Column with a BoxDecoration and an AuthorCard child. The AuthorCard contains authorName, title, and imageProvider properties. Below the AuthorCard is an Expanded Stack with children for 'instruction.dart' and 'models.dart'. The bottom status bar shows the date as May 3 11:08 and the time as 21:35.

Fig 77.

The screenshot shows the Android Studio interface with the project 'cookipidia' open. The code editor displays the file 'card3.dart' under the 'lib/components' directory. The code defines a StatelessWidget named 'Card3' that extends 'ExploreRecipe'. It includes imports for 'package:flutter/material.dart', 'screens.dart', 'splash_screen.dart', 'webview_screen.dart', 'cookipidia_theme.dart', 'mock_cookipidia_service.dart', 'card1.dart', 'card2.dart', and 'card3.dart'. The widget's build method creates a Container with padding and a Column. Inside the Column, there is an Icon (book), a Text title, and a Wrap with spacing and runSpacing. The bottom status bar shows the date as May 3 11:08 and the time as 22:9.

Fig 78.

The screenshot shows the Android Studio interface with the project 'cookipidia' open. The code editor displays the file 'grocery_list_screen.dart'. The code defines a StatelessWidget named 'GroceryListScreen' that extends 'StatelessWidget'. It imports several packages and classes. The build method creates a padding of 16.0, a ListView.separated with itemCount equal to 'groceryItems.length', and an itemBuilder that returns a Dismissible widget. This Dismissible widget has a key, a direction of DismissDirection.endToStart, a background color of red, and a child icon of Icons.delete_forever. The onDismissed callback removes the item from the manager's groceryItems list and shows a snack bar.

```
import 'package:flutter/material.dart';
import '../components/grocery_tile.dart';
import '../models/models.dart';

class GroceryListScreen extends StatelessWidget {
  final GroceryManager manager;

  const GroceryListScreen({
    Key? key,
    required this.manager,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final groceryItems = manager.groceryItems;
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: ListView.separated(
        itemCount: groceryItems.length,
        itemBuilder: (context, index) {
          final item = groceryItems[index];
          return Dismissible(
            key: Key(item.id),
            direction: DismissDirection.endToStart,
            background: Container(
              color: Colors.red,
              alignment: Alignment.centerRight,
              child: const Icon(
                Icons.delete_forever,
                color: Colors.white,
                size: 50.0,
              ),
            ), // Icon
            onDismissed: (direction) {
              manager.deleteItem(index);
              ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                  content: Text('Item removed'),
                ),
              );
            },
          );
        },
      );
    }
}
```

Fig 79.

The screenshot shows the Android Studio interface with the project 'cookipidia' open. The code editor displays the file 'onboarding_screen.dart'. The code defines a StatefulWidget named 'OnboardingScreen' that extends 'StatefulWidget'. It imports various packages and classes. The build method creates a MaterialPage with a name of 'CookipediaPages.onboardingPath' and a key of ValueKey('CookipediaPages.onboardingPath'). The child of the MaterialPage is an OnboardingScreen widget. The OnboardingScreen constructor takes a key. The _OnboardingScreenState createState method returns an _OnboardingScreenState object. The build method of _OnboardingScreenState creates a Scaffold with an AppBar containing a leading icon of Icons.chevron_left and a title of 'Getting Started'. The body of the scaffold contains a CenteredText with the text 'Welcome to Cookipedia!'. The bottom of the screen shows a navigation bar with icons for Home, Explore, Recipe, and Profile.

```
import 'package:cookipedia/models/cookipedia_pages.dart';
import 'package:flutter/material.dart';
import 'package/smooth_page_indicator/smooth_page_indicator.dart';
import 'package/provider/provider.dart';
import './models/models.dart';

class OnboardingScreen extends StatefulWidget {
  static MaterialPageRoute page() {
    return MaterialPageRoute(
      name: CookipediaPages.onboardingPath,
      key: ValueKey(CookipediaPages.onboardingPath),
      child: const OnboardingScreen(),
    );
  }

  const OnboardingScreen({Key? key}) : super(key: key);

  @override
  _OnboardingScreenState createState() => _OnboardingScreenState();
}

class _OnboardingScreenState extends State<OnboardingScreen> {
  final controller = PageController();
  final Color rwColor = const Color.fromRGBO(64, 143, 77, 1);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0.0,
        title: const Text('Getting Started'),
        leading: GestureDetector(
          child: const Icon(
            Icons.chevron_left,
            size: 35,
          ),
        ),
      ),
      body: Center(
        child: CenteredText(
          text: 'Welcome to Cookipedia!',
        ),
      ),
    );
  }
}
```

Fig 80.

The screenshot shows the Android Studio interface with the code editor open to the file `recipes_screen.dart`. The code defines a `RecipesScreen` widget that extends `StatelessWidget`. It uses `FutureBuilder` to fetch recipes from a service and display them in a grid view. The code includes imports for `package:flutter/material.dart`, `models.dart`, `api/mock_cookipidia_service.dart`, and `components/components.dart`. The `build` method returns a `FutureBuilder` with a `future` of `exploreService.getRecipes()` and a `builder` that handles the connection state. If the connection is done, it returns a `RecipesGridView` with the fetched data. Otherwise, it shows a `const Center` with a `CircularProgressIndicator`.

```
import 'package:flutter/material.dart';
import '../models/models.dart';
import '../api/mock_cookipidia_service.dart';
import '../components/components.dart';

class RecipesScreen extends StatelessWidget {
    final exploreService = MockCookipidiaService();

    RecipesScreen({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        // 2
        return FutureBuilder(
            // 3
            future: exploreService.getRecipes(),
            builder: (context, AsyncSnapshot<List<SimpleRecipe>> snapshot) {
                // 4
                if (snapshot.connectionState == ConnectionState.done) {
                    return RecipesGridView(recipes: snapshot.data ?? []);
                } else {
                    return const Center(
                        child: CircularProgressIndicator(),
                    );
                }
            },
        ); // FutureBuilder
    }
}
```

Fig 81.

The screenshot shows the Android Studio interface with the code editor open to the file `cookipidia_theme.dart`. The code defines a `CookipidiaTheme` class with two static `TextTheme` variables: `lightTextTheme` and `darkTextTheme`. Both themes use `openSans` from `google_fonts`. The `lightTextTheme` has a body text size of 14.0, headline 1 size of 32.0, headline 2 size of 21.0, headline 3 size of 16.0, and headline 6 size of 20.0. The `darkTextTheme` has a body text size of 14.0, font weight of `w700`, and headline 6 size of 20.0.

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

class CookipidiaTheme {
    static TextTheme lightTextTheme = TextTheme(
        bodyText1: GoogleFonts.openSans(
            fontSize: 14.0,
            fontWeight: FontWeight.w700,
            color: Colors.black,
        ),
        headline1: GoogleFonts.openSans(
            fontSize: 32.0,
            fontWeight: FontWeight.bold,
            color: Colors.black,
        ),
        headline2: GoogleFonts.openSans(
            fontSize: 21.0,
            fontWeight: FontWeight.w700,
            color: Colors.black,
        ),
        headline3: GoogleFonts.openSans(
            fontSize: 16.0,
            fontWeight: FontWeight.w600,
            color: Colors.black,
        ),
        headline6: GoogleFonts.openSans(
            fontSize: 20.0,
            fontWeight: FontWeight.w600,
            color: Colors.black,
        ),
    );
    static TextTheme darkTextTheme = TextTheme(
        bodyText1: GoogleFonts.openSans(
            fontSize: 14.0,
            fontWeight: FontWeight.w700
        ),
    );
}
```

Fig 82.

```
import 'package:flutter/material.dart';
import 'package:cookipidia/screens/screens.dart';
import 'package:cookipidia/models/models.dart';
import 'package:cookipidia/navigation/app_router.dart';
import 'package:cookipidia/screens/screens.dart';

void main() {
  runApp(const Cookipidia());
}

class Cookipidia extends StatelessWidget {
  const Cookipidia({Key? key}) : super(key: key);

  @override
  State<Cookipidia> createState() => _CookipidiaState();
}

class _CookipidiaState extends State<Cookipidia> {
  final _groceryManager = GroceryManager();
  final _profileManager = ProfileManager();
  final _appStateManager = AppStateManager();

  late AppRouter _appRouter;

  @override
  void initState() {
    _appRouter = AppRouter(
      appStateManager: _appStateManager,
      groceryManager: _groceryManager,
      profileManager: _profileManager,
    );
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        ChangeNotifierProvider(
          create: (context) => _groceryManager,
        ), // ChangeNotifierProvider
      ],
      child: const CookipidiaTheme(),
    );
  }
}
```

Fig 83.

```
import 'package:cloud_firestore/cloud_firestore.dart';

class Message {
  final String text;
  final DateTime date;
  final String? email;
  DocumentReference? reference;

  Message({
    required this.text,
    required this.date,
    this.email,
    this.reference,
  });

  factory Message.fromJson(Map<dynamic, dynamic> json) => Message(
    text: json['text'] as String,
    date: DateTime.parse(json['date'] as String),
    email: json['email'] as String?,
    Map<String, dynamic> toJson() => <String, dynamic>{
      'date': date.toString(),
      'text': text,
      'email': email,
    },
  );

  factory Message.fromSnapshot(DocumentSnapshot snapshot) {
    final message = Message.fromJson(snapshot.data() as Map<String, dynamic>);
    message.reference = snapshot.reference;
    return message;
  }
}
```

Fig 84.

The screenshot shows the Android Studio interface with the project 'raychat' open. The code editor displays the file 'message_dao.dart'. The code defines a class 'MessageDao' with methods for saving messages to a Firestore collection and getting a stream of messages.

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'message.dart';

class MessageDao {
    final CollectionReference collection =
        FirebaseFirestore.instance.collection('messages');

    void saveMessage(Message message) {
        collection.add(message.toJson());
    }

    Stream<QuerySnapshot> getMessageStream() {
        return collection.snapshots();
    }
}
```

Fig 85.

The screenshot shows the Android Studio interface with the project 'raychat' open. The code editor displays the file 'login.dart'. The code defines a stateful widget 'Login' with a login form and state management logic.

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter/widgets.dart';
import 'package:provider/provider.dart';
import '../data/user_dao.dart';

class Login extends StatefulWidget {
    const Login({Key? key}) : super(key: key);
    @override
    _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
    // 1
    final _emailController = TextEditingController();
    // 2
    final _passwordController = TextEditingController();
    // 3
    final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
    @override
    void dispose() {
        // 4
        _emailController.dispose();
        _passwordController.dispose();
        super.dispose();
    }
    @override
    Widget build(BuildContext context) {
        // 1
        final userDao = Provider.of<UserDao>(context, listen: false);
        return Scaffold(
            // 2
            appBar: AppBar(
                title: const Text('Cookipedia Chat'),
            ), // AppBar
            body: Padding(

```

Fig 86.

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Activities, Android Studio, May 3 11:09, raychat - message_list.dart [raychat].
- File Menu:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, Git, Window, Help.
- Project Tree:** Project, raychat (AndroidStudioProjects/raychat), lib, ui, message_list.dart.
- Code Editor:** Displays message_list.dart with Dart code for fetching messages from a database and displaying them in a ListView.
- Bottom Status Bar:** Failed to start monitoring 32006bcaba01950d (50 minutes ago), Event Log, Layout Inspector, 27:73, LF, UTF-8, 2 spaces, master.

Fig 87.

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Activities, Android Studio, May 3 11:09, raychat - message_widget.dart [raychat]
- Menu Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, Git, Window, Help
- Toolbar:** SM A750F (mobile), main.dart, Samsung SM-A750F, Git: ✓, Layout Inspector
- Project Structure:** raychat (AndroidStudioProjects/raychat), lib, ui, message_widget.dart
- Code Editor:** The file message_widget.dart is open. The code defines a StatelessWidget named MessageWidget that takes a message, date, email, and key. It uses a Column with a Container and a Padding widget.
- Side Panels:** Flutter Outline, Flutter Performance, Device Manager, Flutter Inspector, Emulator, Device File Explorer
- Bottom Navigation:** Git, TODO, Problems, Terminal, Dart Analysis, Logcat, Profiler, App Inspection, Layout Inspector
- Status Bar:** Failed to start monitoring 32000bcaba01959d (50 minutes ago), 01:15, LF, UTF-8, 2 spaces, master

Fig 88.

The screenshot shows the Android Studio interface with the project 'raychat' open. The main window displays the code for the main.dart file. The code defines a StatelessWidget named App that builds a MultiProvider widget. This provider includes ChangeNotifierProvider for UserDao and MessageDao, and a Provider for MaterialApp. The MaterialApp has a theme defined and a Consumer for UserDao. The build method checks if UserDao is logged in; if true, it returns a MessageList; otherwise, it returns a Login screen. The code editor shows several imports at the top, including flutter, firebase, and various DAO and model files.

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:provider/provider.dart';
import 'package:material_app/material_app.dart';
import 'package:consumer/consumer.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const App());
}

class App extends StatelessWidget {
  const App({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        ChangeNotifierProvider<UserDao>(
          lazy: false,
          create: () => UserDao(),
        ), // ChangeNotifierProvider
        Provider<MessageDao>(
          lazy: false,
          create: () => MessageDao(),
        ), // Provider
      ],
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'CookipediaChat',
        theme: ThemeData(primaryColor: const Color(0xFF30814A)),
        // ...
        home: Consumer<UserDao>(
          builder: (context, userDao, child) {
            if (userDao.isLoggedIn()) {
              return const MessageList();
            } else {
              return const Login();
            }
          },
        ),
      ),
    );
  }
}
```

Fig 89.

The screenshot shows the Visual Studio Code interface with the pubspec.yaml file open. The file lists dependencies: moor_flutter (~4.1.0), flutter_test, and flutter. It also includes sections for dev_dependencies (flutter_lints: ^1.0.0, build_runner: ^2.1.8, json_serializable: ^6.1.5, chopper_generator: ^4.0.5) and moor_generator (~4.6.0+1). The file concludes with generic Dart documentation and a specific section for Flutter regarding the Material Icons font. The code editor shows syntax highlighting for YAML and Dart code.

```
moor_flutter: ^4.1.0
flutter_test:
  sdk: flutter

# The "flutter_lints" package below contains a set of recommended lints to
# encourage good coding practices. The lint set provided by the package is
# activated in the "analysis_options.yaml" file located at the root of your
# package. See that file for information about deactivating specific lint
# rules and activating additional ones.
flutter_lints: ^1.0.0
build_runner: ^2.1.8
json_serializable: ^6.1.5
chopper_generator: ^4.0.5

moor_generator: ^4.6.0+1

# 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

  assets:
    - assets/recipes1.json
    - assets/recipes2.json
```

Fig 90.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'colors.dart' file is selected.
- Editor:** Displays the 'colors.dart' file content:

```
1 import 'dart:ui';
2 const Color green = Color(0xff158442);
3 const Color lightGrey = Color(0xfffff6367);
4 const Color shim = Color(0x7f000000);
```
- Bottom Bar:** PROBLEMS (38), OUTPUT, DEBUG CONSOLE, TERMINAL. The terminal shows the command: `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$`
- Taskbar:** Shows various application icons including Finder, App Store, Mail, Spotify, and browser icons.

Fig 91.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'main_screen.dart' file is selected.
- Editor:** Displays the 'main_screen.dart' file content:

```
1 import 'package:flutter/services.dart';
2 import 'package:flutter_svg/flutter_svg.dart';
3 import 'colors.dart';
4
5 import 'myrecipes/my_recipes_list.dart';
6 import 'recipes/recipe_list.dart';
7 import 'shopping/shopping_list.dart';
8 import 'package:flutter/material.dart';
9 import 'package:shared_preferences/shared_preferences.dart';
10
11 class MainScreen extends StatefulWidget {
12   const MainScreen({Key? key}) : super(key: key);
13
14   @override
15   _MainScreenState createState() => _MainScreenState();
16 }
17
18 class _MainScreenState extends State<MainScreen> {
19   int _selectedIndex = 0;
20   List<Widget> pageList = [];
21
22   static const String prefSelectedIndexKey = 'selectedIndex';
23
24   @override
25   void initState() {
26     super.initState();
27     pageList.add(const RecipeList());
28     pageList.add(const MyRecipesList());
29     pageList.add(const ShoppingList());
30     getCurrentIndex();
31   }
32 }
```
- Bottom Bar:** PROBLEMS (38), OUTPUT, DEBUG CONSOLE, TERMINAL. The terminal shows the command: `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$`
- Taskbar:** Shows various application icons including Finder, App Store, Mail, Spotify, and browser icons.

Fig 92.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for COOKIPIDIA_TEST_1, including files like ingredient.dart, models.dart, recipe.dart, moor, sqlite, memory_repository.dart, repository.dart, mock_service, network, ui, myrecipes, recipes, shopping, widgets, colors.dart, main_screen.dart, and recipe_card.dart.
- Editor:** The main.dart and recipe_card.dart files are open. recipe_card.dart contains Dart code for a recipe card widget.
- Terminal:** The terminal shows the command: `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$`
- Bottom Bar:** Includes icons for various tools and a docked application bar.

Fig 93.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for COOKIPIDIA_TEST_1, including files like ingredient.dart, models.dart, recipe.dart, moor, sqlite, memory_repository.dart, repository.dart, mock_service, network, ui, myrecipes, recipes, shopping, and shopping_list.dart.
- Editor:** The main.dart and shopping_list.dart files are open. shopping_list.dart contains Dart code for a ShoppingList StatefulWidget.
- Terminal:** The terminal shows the command: `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$`
- Bottom Bar:** Includes icons for various tools and a docked application bar.

Fig 94.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'shopping' folder is expanded, showing files like ingredient.dart, models.dart, recipe.dart, moor, memory_repository.dart, repository.dart, ui, recipes, and shopping. The 'shopping_list.dart' file is selected.
- Code Editor:** The 'shopping_list.dart' file is open, displaying Dart code for a shopping list screen. It uses StreamBuilder to listen to a repository's watchAllIngredients stream and build a ListView of ingredients with checkboxes.
- Terminal:** The terminal shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$'.
- Bottom Bar:** Includes icons for file operations, a search bar, and various developer tools like Dart DevTools and Flutter.

Fig 95.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'mock_service' folder is selected.
- Code Editor:** The 'mock_service.dart' file is open, displaying Dart code for a MockService class that implements ServiceInterface. It uses Random to generate recipes and loads them from assets.
- Terminal:** The terminal shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$'.
- Bottom Bar:** Includes icons for file operations, a search bar, and various developer tools like Dart DevTools and Flutter.

Fig 96.

Activities Visual Studio Code ▾ May 3 23:52

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIPIDIA_TEST_1

- ingredient.dart
- models.dart
- recipe.dart
- moor
- moor_db.dart
- moor_db.g.dart
- moor_repository.dart
- sqlite
- memory_repository.dart
- repository.dart
- mock_service
- mock_service.dart
- network
- model_converter.dart
- model_response.dart
- recipe_model.dart
- recipe_model.g.dart
- recipe_service.chopper.dart
- recipe_service.dart
- service_interface.dart
- ui
- myrecipes
- recipes
- shopping
- shopping_list.dart
- widgets
- colors.dart
- main_screen.dart

main.dart model_converter.dart

```
lib > network > model_converter.dart > ...
1 import 'dart:convert';
2 import 'package:chopper/chopper.dart';
3
4 import 'model_response.dart';
5 import 'recipe_model.dart';
6
7 class ModelConverter implements Converter {
8   @override
9   Request convertRequest(Request request) {
10     final req = applyHeader(
11       request,
12       contentTypeKey,
13       jsonHeaders,
14       override: false,
15     );
16
17     return encodeJson(req);
18   }
19
20   Request encodeJson(Request request) {
21     final contentType = request.headers[contentTypeKey];
22     if (contentType != null && contentType.contains(jsonHeaders)) {
23       return request.copyWith(body: json.encode(request.body));
24     }
25     return request;
26   }
27
28   Response<BodyType> decodeJson<BodyType, InnerType>(Response response) {
29     final contentType = response.headers[contentTypeKey];
30     var body = response.body;
31     if (contentType != null && contentType.contains(jsonHeaders)) {
32       body = utf8.decode(response.bodyBytes);
33     }
34     return response;
35   }
36
37 }
```

PROBLEMS 38 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$

Ln 55, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

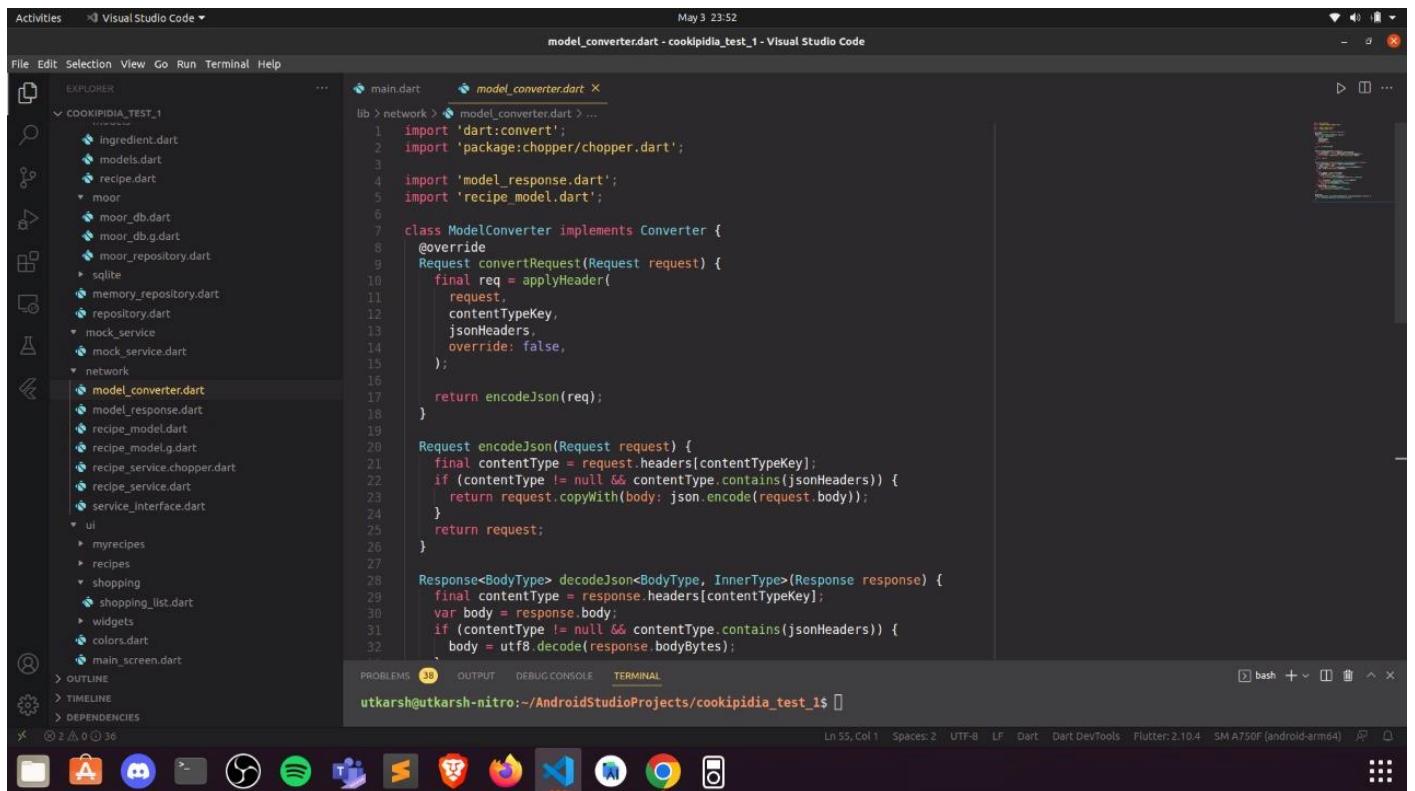


Fig 97.

Activities Visual Studio Code ▾ May 3 23:52

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIPIDIA_TEST_1

- ingredient.dart
- models.dart
- recipe.dart
- moor
- moor_db.dart
- moor_db.g.dart
- moor_repository.dart
- sqlite
- memory_repository.dart
- repository.dart
- mock_service
- mock_service.dart
- network
- model_converter.dart
- model_response.dart
- recipe_model.dart
- recipe_model.g.dart
- recipe_service.chopper.dart
- recipe_service.dart
- service_interface.dart
- ui
- myrecipes
- recipes
- shopping
- shopping_list.dart
- widgets
- colors.dart
- main_screen.dart

main.dart model_response.dart

```
lib > network > model_response.dart > ...
1 // 1
2 abstract class Result<T> {
3 }
4 // 2
5 class Success<T> extends Result<T> {
6   final T value;
7   Success(this.value);
8 }
9 // 3
10 class Error<T> extends Result<T> {
11   final Exception exception;
12
13   Error(this.exception);
14 }
```

PROBLEMS 38 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

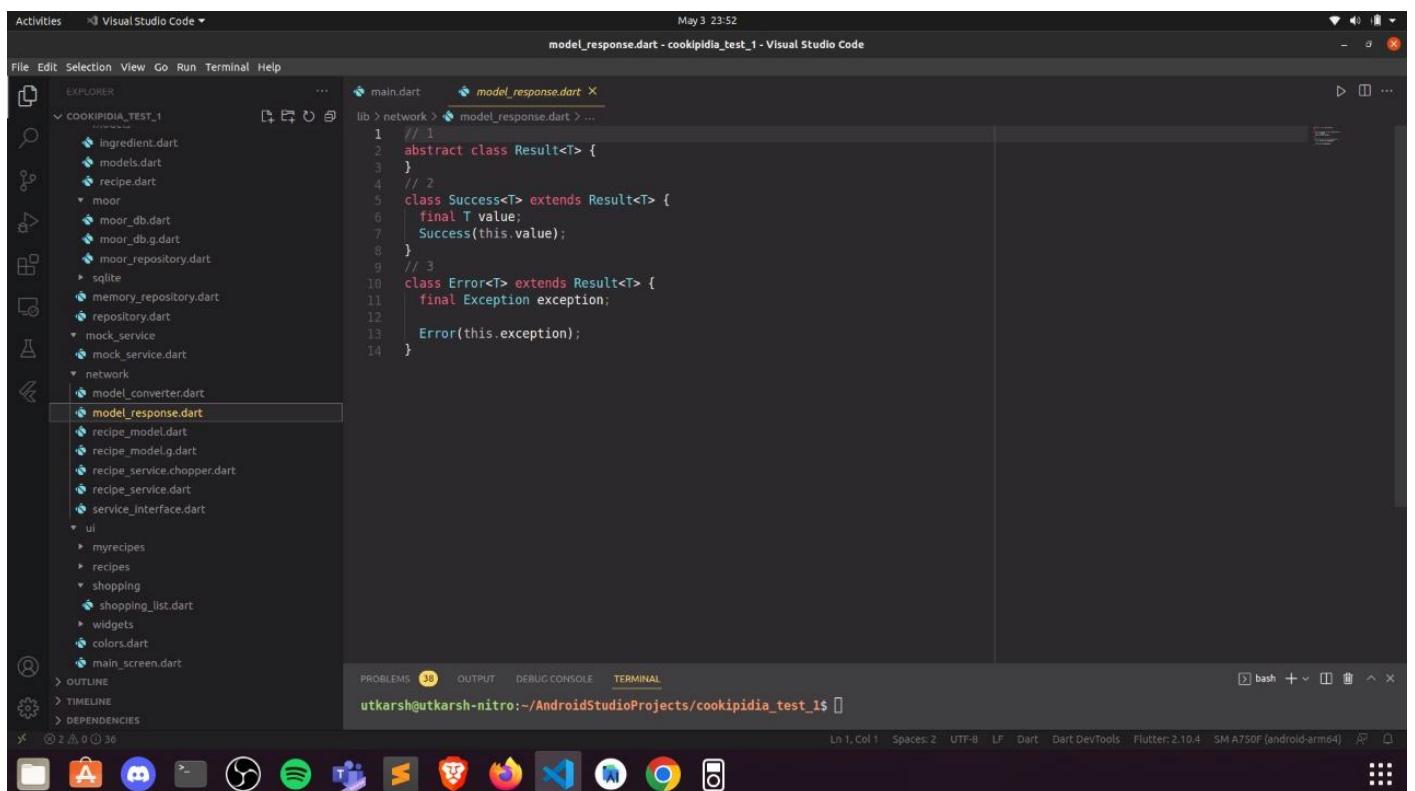


Fig 98.

Activities Visual Studio Code ▾ May 3 23:52

File Edit Selection View Go Run Terminal Help

EXPLORER lib > network recipe_model.dart - cookipidia_test_1 - Visual Studio Code

main.dart recipe_model.dart

```
1 import 'package:json_annotation/json_annotation.dart';
2 part 'package:cookipidia_test_1/data/models/models.dart';
3
4 part 'recipe_model.g.dart';
5
6 @JsonSerializable()
7 class APIRecipeQuery {
8   factory APIRecipeQuery.fromJson(Map<String, dynamic> json) =>
9     _$APIRecipeQueryFromJson(json);
10
11 Map<String, dynamic> toJson() => _$APIRecipeQueryToJson(this);
12 @JsonKey(name: 'q')
13 String query;
14 int from;
15 int to;
16 bool more;
17 int count;
18 List<APIHits> hits;
19
20 APIRecipeQuery({
21   required this.query,
22   required this.from,
23   required this.to,
24   required this.more,
25   required this.count,
26   required this.hits,
27 });
28
29 @JsonSerializable()
30 class APIHits {
31   APIRecipe recipe;
32 }
```

PROBLEMS 38 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

File Explorer, Editor, Problems, Output, Debug Console, Terminal, Bottom Taskbar.

Fig 99.

Activities Visual Studio Code ▾ May 3 23:52

File Edit Selection View Go Run Terminal Help

EXPLORER lib > network recipe_service.chopper.dart - cookipidia_test_1 - Visual Studio Code

main.dart recipe_service.chopper.dart

```
1 // GENERATED CODE - DO NOT MODIFY BY HAND
2
3 part of 'recipe_service.dart';
4
5 // ****
6 // ChopperGenerator
7 // ****
8
9 // ignore for file: always_put_control_body_on_new_line, always_specify_types, prefer_const_declarations, unnecessary_braces
10 class $RecipeService extends RecipeService {
11   _$RecipeService([ChopperClient? client]) {
12     if (client == null) return;
13     this.client = client;
14   }
15
16   @override
17   final definitionType = RecipeService;
18
19   @override
20   Future<Response<Result<APIRecipeQuery>>> queryRecipes(
21     String query, int from, int to) {
22     final $url = 'search';
23     final $params = <String, dynamic>{'q': query, 'from': from, 'to': to};
24     final $request = Request('GET', $url, client.baseUrl, parameters: $params);
25     return client.send<Result<APIRecipeQuery>>($request);
26   }
27 }
28
```

PROBLEMS 38 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

File Explorer, Editor, Problems, Output, Debug Console, Terminal, Bottom Taskbar.

Fig 100.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'lib' folder contains 'network', 'model_converter.dart', 'model_response.dart', 'recipe_model.dart', 'recipe_model.g.dart', 'recipe_service.chopper.dart', 'recipe_service.dart', and 'service_Interface.dart'. The 'ui' folder contains 'myrecipes', 'recipes', and 'shopping'.
- Editor:** The 'recipe_service.dart' file is open, showing Dart code for creating a RecipeService instance using ChopperClient and HttpLoggingInterceptor.
- Terminal:** The terminal shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$'.
- Bottom Taskbar:** Includes icons for various applications like Spotify, VS Code, and a browser.

Fig 101.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for 'COOKIPIDIA_TEST_1'. The 'lib' folder contains 'network', 'model_converter.dart', 'model_response.dart', 'recipe_model.dart', 'recipe_model.g.dart', 'recipe_service.chopper.dart', 'recipe_service.dart', and 'service_Interface.dart'. The 'ui' folder contains 'myrecipes', 'recipes', and 'shopping'.
- Editor:** The 'service_Interface.dart' file is open, showing Dart code for an abstract ServiceInterface class with a queryRecipes method.
- Terminal:** The terminal shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1\$'.
- Bottom Taskbar:** Includes icons for various applications like Spotify, VS Code, and a browser.

Fig 102.

```
lib > ui > myrecipes > my_recipes_list.dart > _MyRecipesListState > _buildRecipeList
35 // 4
36 stream: repository.watchAllRecipes(),
37 // 5
38 builder: (context, AsyncSnapshot<List<Recipe>> snapshot) {
39 // 6
40 if (snapshot.connectionState == ConnectionState.active) {
41 final recipes = snapshot.data ?? [];
42 return ListView.builder(
43 itemCount: recipes.length,
44 itemBuilder: (BuildContext context, int index) {
45 final recipe = recipes[index];
46 return SizedBox(
47 height: 100,
48 child: Slidable(
49 actionPane: const SlidableDrawerActionPane(),
50 actionExtentRatio: 0.25,
51 child: Card(
52 elevation: 1.0,
53 shape: RoundedRectangleBorder(
54 borderRadius: BorderRadius.circular(10.0),
55 ), // RoundedRectangleBorder
56 color: Colors.white,
57 child: Align(
58 alignment: Alignment.center,
59 child: Padding(
60 padding: const EdgeInsets.all(8.0),
61 child: ListTile(
62 leading: CachedNetworkImage(
63 imageUrl: recipe.image ?? '',
64 height: 120,
65 width: 60,
66 fit: BoxFit.cover,
67 
```

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cokipidia_test_1\$

Fig 103.

```
flutter:
  sdk: flutter

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
google_fonts: ^2.3.1
provider: ^6.0.2
flutter_colorpicker: ^1.0.3
intl: ^0.17.0
uuid: ^3.0.6
smooth_page_indicator: ^1.0.0+2
webview_flutter: ^3.0.1
url_launcher: ^6.0.20
shared_preferences: ^2.0.13
image_picker: ^0.8.5
path_provider: ^2.0.9
firebase_core: ^1.16.0
cloud_firestore: ^3.1.14

dev_dependencies:
  flutter_test:
    sdk: flutter

# The "flutter_lints" package below contains a set of recommended lints to
# encourage good coding practices. The lint set provided by the package is
# activated in the analysis_options.yaml file located at the root of your
# package. See that file for information about deactivating specific lint
# rules and activating additional ones.
flutter_lints: ^1.0.0

# 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.

```

Fig 104.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Editor:** The main editor window displays the `build.gradle` file for a Flutter project named "COOKIPIDIA". The code is as follows:

```
1 android > app > # build.gradle
2     def localProperties = new Properties()
3     def localPropertiesfile = rootProject.file('local.properties')
4     if (localPropertiesfile.exists()) {
5         localPropertiesfile.withReader('UTF-8') { reader ->
6             localProperties.load(reader)
7         }
8     }
9
10    def flutterRoot = localProperties.getProperty('flutter.sdk')
11    if (flutterRoot == null) {
12        throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")
13    }
14
15    def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
16    if (flutterVersionCode == null) {
17        flutterVersionCode = '1'
18    }
19
20    def flutterVersionName = localProperties.getProperty('flutter.versionName')
21    if (flutterVersionName == null) {
22        flutterVersionName = '1.0'
23    }
24
25    apply plugin: 'com.android.application'
26    apply plugin: 'kotlin-android'
27    apply from: '$flutterRoot/packages/flutter_tools/gradle/flutter.gradle'
28    apply plugin: 'com.google.gms.google-services'
29
30    android {
31        compileSdkVersion flutter.compileSdkVersion
32
33        compileOptions {
34            sourceCompatibility JavaVersion.VERSION_1_8
35            targetCompatibility JavaVersion.VERSION_1_8
36        }
37    }
38
39
```

Bottom Status Bar: Ln 27, Col 47 Spaces: 4 UTF-8 LF Groovy Dart DevTools SM A750F (android-arm64)

Fig 105.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Editor:** The main editor window displays the `cookipidia_android.iml` file for a Flutter project named "COOKIPIDIA". The code is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <module type="JAVA_MODULE" version="4">
3     <component name="FacetManager">
4         <facet type="android" name="Android">
5             <configuration>
6                 <option name="ALLOW_USER_CONFIGURATION" value="false" />
7                 <option name="GEN_FOLDER_RELATIVE_PATH_APT" value="/gen" />
8                 <option name="GEN_FOLDER_RELATIVE_PATH_AIDL" value="/gen" />
9                 <option name="MANIFEST_FILE_RELATIVE_PATH" value="/app/src/main/AndroidManifest.xml" />
10                <option name="RES_FOLDER_RELATIVE_PATH" value="/app/src/main/res" />
11                <option name="ASSETS_FOLDER_RELATIVE_PATH" value="/app/src/main/assets" />
12                <option name="LIBS_FOLDER_RELATIVE_PATH" value="/app/src/main/libs" />
13                <option name="PROGUARD_LOGS_FOLDER_RELATIVE_PATH" value="/app/src/main/proguard_logs" />
14            </configuration>
15        </facet>
16    </component>
17    <component name="NewModuleRootManager" inherit-compiler-output="true">
18        <exclude-output />
19        <content url="file://$MODULE_DIR$">
20            <sourceFolder url="file://$MODULE_DIR$/app/src/main/java" isTestSource="false" />
21            <sourceFolder url="file://$MODULE_DIR$/app/src/main/kotlin" isTestSource="false" />
22            <sourceFolder url="file://$MODULE_DIR$/gen" isTestSource="false" generated="true" />
23        </content>
24        <orderEntry type="jdk" jdkName="Android API 29 Platform" jdkType="Android SDK" />
25        <orderEntry type="sourceFolder" forTests="false" />
26        <orderEntry type="library" name="Flutter for Android" level="project" />
27        <orderEntry type="library" name="KotlinJavaRuntime" level="project" />
28    </component>
29</module>
```

Bottom Status Bar: Ln 1, Col 1 Spaces: 2 UTF-8 LF XML Dart DevTools SM A750F (android-arm64)

Fig 106.

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

COOKIPIDIA

- ↳ .dart_tool
- ↳ .idea
- ↳ android

 - ↳ .gradle
 - ↳ app

 - ↳ src

 - ↳ build.gradle
 - ↳ google-services.json
 - ↳ gradle
 - ↳ .gitignore
 - ↳ build.gradle

- ↳ cookipidia_android.iml
- ↳ gradle.properties
- ↳ gradlew
- ↳ gradlew.bat
- ↳ local.properties
- ↳ settings.gradle

login_screen.dart message.dart M pubspec.yaml M settings.gradle

```
1 include ':app'
2
3 def localPropertiesFile = new File(rootProject.projectDir, "local.properties")
4 def properties = new Properties()
5
6 assert localPropertiesFile.exists()
7 localPropertiesFile.withReader("UTF-8") { reader -> properties.load(reader) }
8
9 def flutterSdkPath = properties.getProperty("flutter.sdk")
10 assert flutterSdkPath != null, "flutter.sdk not set in local.properties"
11 apply from: "$flutterSdkPath/packages/flutter_tools/gradle/app_plugin_loader.gradle"
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Groovy Dart DevTools SM A750F (android-arm64)

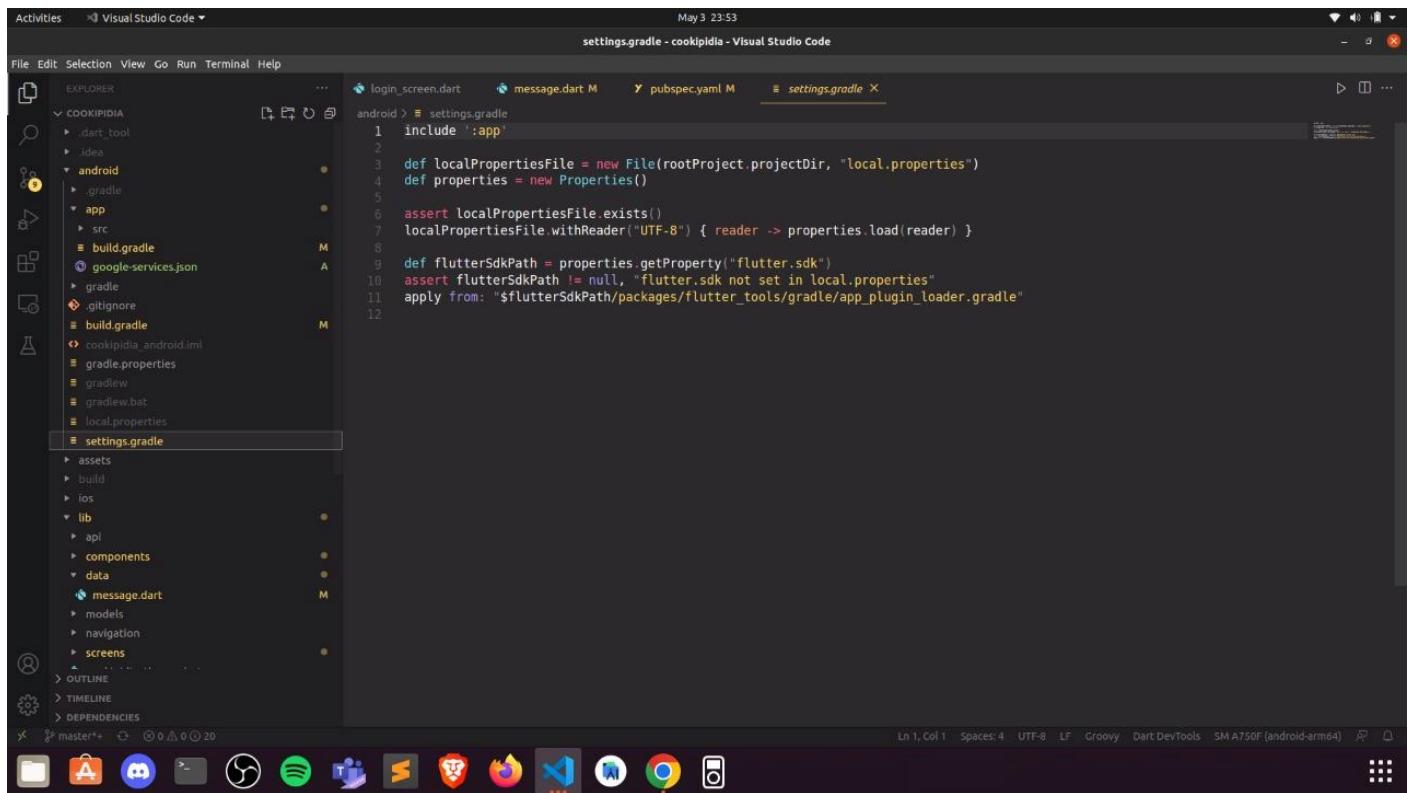


Fig 107.

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

COOKIPIDIA

- ↳ .dart_tool
- ↳ .idea
- ↳ android

 - ↳ .gradle
 - ↳ app

 - ↳ src

 - ↳ build.gradle
 - ↳ google-services.json
 - ↳ gradle
 - ↳ .gitignore
 - ↳ build.gradle

- ↳ cookipidia_android.iml
- ↳ gradle.properties
- ↳ gradlew
- ↳ gradlew.bat
- ↳ local.properties
- ↳ settings.gradle

login_screen.dart message.dart M pubspec.yaml M gradlew.bat

```
19 @rem Find JAVA_HOME
20 if defined JAVA_HOME goto findJavaFromJavaHome
21
22 set JAVA_EXE=java.exe
23 %JAVA_EXE% -version >NUL 2>&1
24 if "%ERRORLEVEL%" == "0" goto init
25
26 echo.
27 echo ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH.
28 echo.
29 echo Please set the JAVA_HOME variable in your environment to match the
30 echo location of your Java installation.
31
32 goto fail
33
34 :findJavaFromJavaHome
35 set JAVA_HOME=%JAVA_HOME:"=%
36 set JAVA_EXE=%JAVA_HOME%/bin/java.exe
37
38 if exist "%JAVA_EXE%" goto init
39
40 echo.
41 echo ERROR: JAVA_HOME is set to an invalid directory: %JAVA_HOME%
42 echo.
43 echo Please set the JAVA_HOME variable in your environment to match the
44 echo location of your Java installation.
45
46 goto fail
47
48 :init
49 @rem Get command-line arguments, handling Windows variants
50
51 if not "%OS%" == "Windows_NT" goto win9xME_args
52 if "%@eval[2+2]" == "4" goto 4NT_args
53
54 :win9xME_args
55 @rem Slurp the command line arguments.
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Batch Dart DevTools SM A750F (android-arm64)

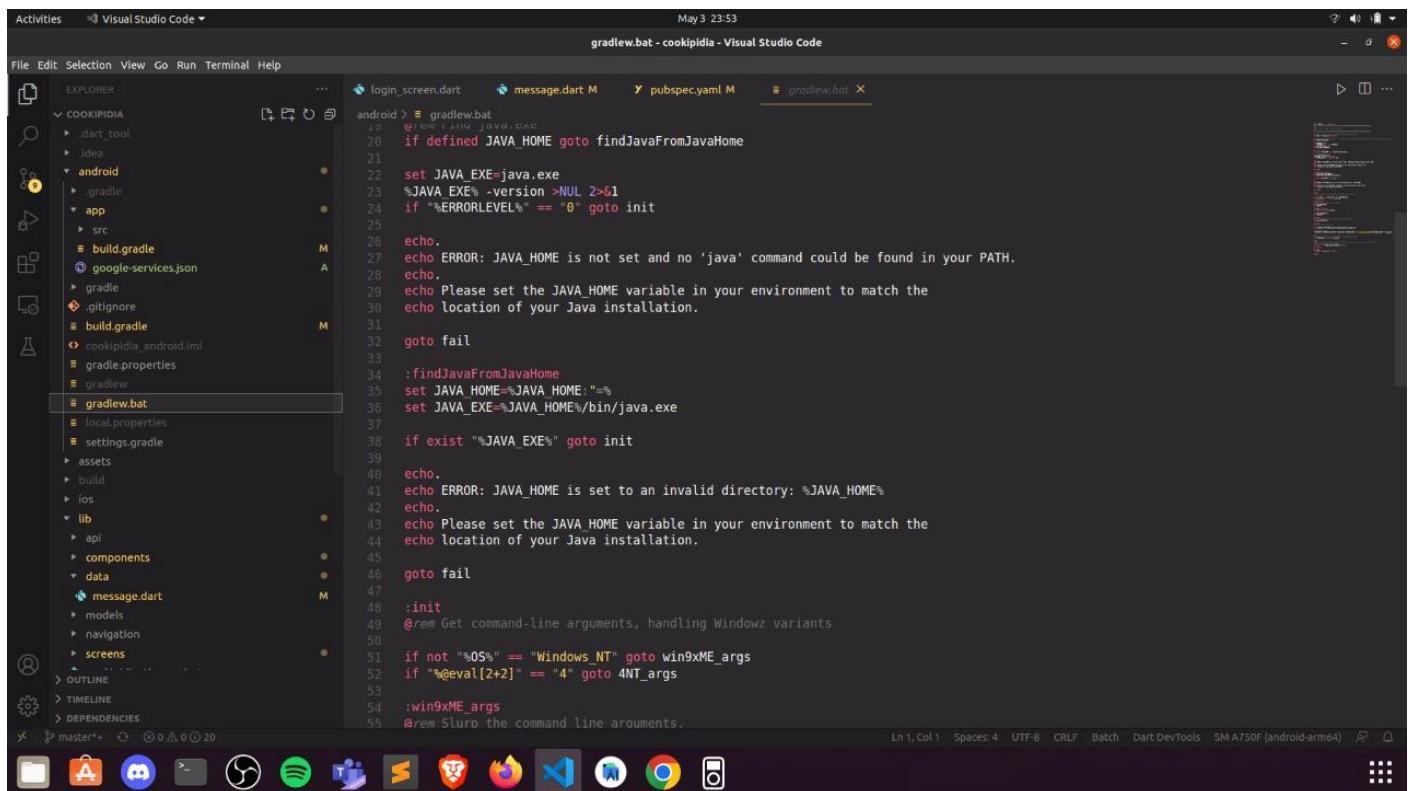


Fig 108.

Activities Visual Studio Code ▾ May 3 23:53

File Edit Selection View Go Run Terminal Help

EXPLORER

COOKIPIDIA

- gradlew.bat
- local.properties
- settings.gradle
- assets
- build
- ios
- lib
- api
- mock_cookipidia_service.dart
- components
- data
- message.dart
- models
- navigation
- screens
- cookipidia_theme.dart
- generated_plugin_registrant.dart
- main.dart
- test
- web
- flutter_plugins
- .Flutter-plugins-dependencies
- .gitignore
- .metadata
- .packages
- analysis_options.yaml
- cookipidia.iml
- pubspec.lock

mock_cookipidia_service.dart

```
lib > api > mock_cookipidia_service.dart > ...  
1 import 'dart:convert';  
2  
3 import 'package:flutter/services.dart';  
4  
5 import '../models/models.dart';  
6  
7 // Mock recipe service that grabs sample json data to mock recipe request/response  
8 class MockCookipidiaService {  
9   // Batch request that gets both today recipes and friend's feed  
10  Future<ExploreData> getExploreData() async {  
11    final todayRecipes = await _getTodayRecipes();  
12    final friendPosts = await _getFriendFeed();  
13  
14    return ExploreData(todayRecipes, friendPosts);  
15  }  
16  
17  // Get sample explore recipes json to display in ui  
18  Future<List<ExploreRecipe>> _getTodayRecipes() async {  
19    // Simulate api request wait time  
20    await Future.delayed(const Duration(milliseconds: 1000));  
21    // Load json from file system  
22    final jsonString =  
23      await _loadAsset('assets/sample_data/sample_explore_recipes.json');  
24    // Decode to json  
25    final Map<String, dynamic> json = jsonDecode(jsonString);  
26  
27    // Go through each recipe and convert json to ExploreRecipe object.  
28    if (json['recipes'] != null) {  
29      final recipes = <ExploreRecipe>[];  
30      json['recipes'].forEach((v) {  
31        recipes.add(ExploreRecipe.fromJson(v));  
32      });  
33      return recipes;  
34    } else {  
35      return [];  
36    }  
37  }  
38}
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 109.

Activities Visual Studio Code ▾ May 3 23:53

File Edit Selection View Go Run Terminal Help

EXPLORER

COOKIPIDIA

- gradlew.bat
- local.properties
- settings.gradle
- assets
- build
- ios
- lib
- api
- mock_cookipidia_service.dart
- components
- author_card.dart
- card1.dart
- card2.dart
- card3.dart
- circle_image.dart
- components.dart
- friend_post_list_view.dart
- friend_post_tile.dart
- grocery_tile.dart
- recipe_thumbnail.dart
- recipes_grid_view.dart
- today_recipe_list_view.dart
- data
- message.dart
- models
- navigation
- screens
- cookipidia_theme.dart

author_card.dart

```
lib > components > author_card.dart > AuthorCard > AuthorCard  
1 import 'package:flutter/material.dart';  
2 import '../cookipidia_theme.dart';  
3 import 'circle_image.dart';  
4  
5 class AuthorCard extends StatefulWidget {  
6   final String authorName;  
7   final String title;  
8   final ImageProvider? imageProvider;  
9  
10  const AuthorCard({  
11    Key? key,  
12    required this.authorName,  
13    required this.title,  
14    this.imageProvider,  
15  }) : super(key: key);  
16  
17  @override  
18  State<AuthorCard> createState() => _AuthorCardState();  
19  
20  class _AuthorCardState extends State<AuthorCard> {  
21    bool _isFavorited = false;  
22    @override  
23    Widget build(BuildContext context) {  
24      return Container(  
25        padding: const EdgeInsets.all(16),  
26        child: Row(  
27          mainAxisAlignment: MainAxisAlignment.spaceBetween,  
28          children: [  
29            // 1  
30            Row(  
31              children: [  
32                CircleImage(  
33                  imageProvider: widget.imageProvider,  
34                  imageRadius: 25,  
35                ), // CircleImage
```

Ln 13, Col 25 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 10.

The screenshot shows a Visual Studio Code interface with the following details:

- Activity Bar:** Activities, Visual Studio Code, May 3 23:53.
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer Sidebar:** COOKIPIDIA folder structure:
 - gradlew.bat
 - local.properties
 - settings.gradle
 - assets
 - build
 - ios
 - lib
 - api
 - mock_cookipidia_service.dart
 - components
 - author_card.dart
 - card1.dart M
 - card2.dart
 - card3.dart
 - circle_image.dart
 - components.dart
 - friend_post_list_view.dart
 - Friend_post_tile.dart
 - grocery_tile.dart
 - recipe_thumbnail.dart
 - recipes_grid_view.dart
 - today_recipe_list_view.dart
 - data
 - message.dart M
 - models
 - navigation
 - screens
 - cookipidia_theme.dart
- Code Editor:** card1.dart - cookipidia - Visual Studio Code. The code defines a StatelessWidget named Card1 that displays a recipe card with a subtitle, title, and headline2 text.

```
import 'package:cookipidia/cookipidia_theme.dart';
import 'package:cookipidia/main.dart';
import 'package:flutter/material.dart';
import 'package:cookipidia/cookipidia_theme.dart';

import '../models/explore_recipe.dart';

class Card1 extends StatelessWidget {
    final ExploreRecipe recipe;

    const Card1({
        Key? key,
        required this.recipe,
    }) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Center(
            child: Container(
                child: Stack(
                    children: [
                        Text(
                            recipe.subtitle,
                            style: CookipidiaTheme.darkTextTheme.bodyText1,
                        ),
                        Positioned(
                            child: Text(
                                recipe.title,
                                style: CookipidiaTheme.darkTextTheme.headline2,
                            ),
                            top: 20,
                        ),
                        Positioned(
                            child: Text(

```
- Bottom Bar:** master+, OUTLINE, TIMELINE, DEPENDENCIES, GitHub icon, SASS icon, Docker icon, VS Code icon, Flutter icon, Dart DevTools icon, Flutter 2.10.4 icon, SM A750F (android-arm64) icon.

Fig 111.

Activities Visual Studio Code - May 3, 23:53

File Edit Selection View Go Run Terminal Help

EXPLORER

COOKIPIDIA

- gradlew.bat
- local.properties
- settings.gradle
- assets
- build
- ios
- lib
 - api
 - mock_cookipidia_service.dart
 - components
 - author_card.dart
 - card1.dart M
 - card2.dart
 - card3.dart
 - circle_image.dart
 - components.dart
 - Friend_post_list_view.dart
 - friend_post_tile.dart
 - grocery_tile.dart
 - recipe_thumbnail.dart
 - recipes_grid_view.dart
 - today_recipe_list_view.dart
- data
- message.dart M
- models
- navigation
- screens
- cookipidia_theme.dart

OUTLINE

TIMELINE

DEPENDENCIES

master* 0 0 0 0 20

login_screen.dart message.dart M pubspec.yaml M card1.dart M

```
lib > components > card1.dart > Card1
  style: CookipidiaTheme.darkTextTheme.bodyText1,
    ), // Text
    bottom: 10,
    right: 0,
  ) // Positioned
),
),
),
padding: const EdgeInsets.all(16),
constraints: const BoxConstraints.expand(
width: 350,
height: 450,
),
// BoxConstraints.expand
decoration: BoxDecoration(
image: DecorationImage(
image: AssetImage(recipe.backgroundImage),
fit: BoxFit.cover,
),
// DecorationImage
borderRadius: const BorderRadius.all(Radius.circular(10.0)),
),
// BoxDecoration
),
// Container
); // Center
}
```

Ln 9, Col 19 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 112.

Activities Visual Studio Code May 3 23:53 card2.dart - cookipidia - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIDIA lib > components > card2.dart > ...

```
30   AuthorCard( 31     authorName: recipe.authorName, 32     title: recipe.role, 33     imageProvider: AssetImage(recipe.profileImage), 34   ), // AuthorCard 35   Expanded( 36     child: Stack( 37       children: [ 38         Positioned( 39           bottom: 16, 40           right: 16, 41           child: Text( 42             recipe.title, 43             style: CookipidiaTheme.lightTextTheme.headline1, 44           ), // Text 45         Positioned( 46           bottom: 70, 47           left: 16, 48           child: RotatedBox( 49             quarterTurns: 3, 50             child: Text( 51               recipe.subtitle, 52               style: CookipidiaTheme.lightTextTheme.headline1, 53             ), // Text 54           ), // RotatedBox 55         ), // Positioned 56       ], 57     ), // Stack 58   ), // Expanded 59   ], // Column 60   ), // Container 61   ); // Center 62 } // Center 63 ); // Center 64 } // Center 65 }
```

master+ 0 0 0 0 20

login_screen.dart message.dart M pubspec.yaml M card2.dart

Line 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 113.

Activities Visual Studio Code May 3 23:53 card3.dart - cookipidia - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIDIA lib > components > card3.dart > ...

```
27   } @override 28   Widget build(BuildContext context) { 29     return Center( 30       child: Container( 31         constraints: const BoxConstraints.expand( 32           width: 350, 33           height: 450, 34         ), // BoxConstraints.expand 35         decoration: BoxDecoration( 36           image: DecorationImage( 37             image: AssetImage(recipe.backgroundImage), 38             fit: BoxFit.cover, 39           ), // DecorationImage 40           borderRadius: const BorderRadius.all( 41             Radius.circular(10.0), 42           ), // BorderRadius.all 43         ), // BoxDecoration 44         child: Stack( 45           children: [ 46             Container( 47               decoration: BoxDecoration( 48                 color: Colors.black.withOpacity(0.6), 49                 borderRadius: const BorderRadius.all(Radius.circular(10.0)), 50               ), // BoxDecoration 51             Container( 52               padding: const EdgeInsets.all(16), 53               child: Column( 54                 crossAxisAlignment: CrossAxisAlignment.start, 55                 children: [ 56                   const Icon( 57                     Icons.book, 58                     color: Colors.white, 59                     size: 40, 60                   ), // Icon 61                   const SizedBox(height: 8) 62                 ], // children 63               ), // Column 64             ), // Container 65           ], // children 66         ), // Stack 67       ), // Center 68     ); // Center 69   } // build 70 }
```

master+ 0 0 0 0 20

login_screen.dart message.dart M pubspec.yaml M card3.dart

Line 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 114.

A screenshot of Visual Studio Code showing the code for `circle_image.dart`. The code defines a `CircleImage` widget that extends `StatelessWidget`. It imports `package:flutter/material.dart` and uses `CircleAvatar` to create a circular image with a white background and a radius of 20. The code includes several comments and a `build` method that returns a `CircleAvatar` with a `backgroundImage` and a `radius` of `imageRadius`.

```
import 'package:flutter/material.dart';
class CircleImage extends StatelessWidget {
  const CircleImage({
    Key? key,
    this.imageProvider,
    this.imageRadius = 20,
  }) : super(key: key);
  final double imageRadius;
  final ImageProvider? imageProvider;
  @override
  Widget build(BuildContext context) {
    // 3
    return CircleAvatar(
      backgroundColor: Colors.white,
      radius: imageRadius,
      // 4
      child: CircleAvatar(
        radius: imageRadius - 5,
        backgroundImage: imageProvider,
      ), // CircleAvatar
    ); // CircleAvatar
  }
}
```

Fig 115.

A screenshot of Visual Studio Code showing the code for `friend_post_list_view.dart`. The code defines a `FriendPostListView` widget that extends `StatelessWidget`. It imports `package:flutter/material.dart` and uses `Padding`, `Column`, and `ListView.separated` to create a list of posts. The code includes several comments and a `build` method that returns a `Padding` with a `padding` of `const EdgeInsets.only(left: 16, right: 16, top: 0)`, a `child` of a `Column` with a `crossAxisAlignment` of `CrossAxisAlignment.start`, and a `children` list containing a `Text` and a `ListView.separated` with a `primary` value of `false`.

```
const FriendPostListView({
  Key? key,
  required this.friendPosts,
}) : super(key: key);
@Override
Widget build(BuildContext context) {
  // 2
  return Padding(
    padding: const EdgeInsets.only(
      left: 16,
      right: 16,
      top: 0,
    ), // EdgeInsets.only
    // 3
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        // 4
        Text('Social Chefs ! ', style: Theme.of(context).textTheme.headline1),
        // 5
        const SizedBox(height: 16),
        // TODO: Add PostlistView here
        ListView.separated(
          // 2
          primary: false,
          // 3
          physics: const NeverScrollableScrollPhysics(),
          shrinkWrap: true,
          scrollDirection: Axis.vertical,
          itemCount: friendPosts.length,
          itemBuilder: (context, index) {
            // 4
            final post = friendPosts[index];
            return FriendPostTile(post: post);
          }
        ),
      ],
    ),
  );
}
```

Fig 116.

Activities Visual Studio Code May 3 23:53 friend_post_tile.dart - cookipidia - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIPIDIA build ios lib api components mock_cookipidia_service.dart card1.dart card2.dart card3.dart circle_image.dart components.dart Friend_post_list_view.dart friend_post_tile.dart grocery_tile.dart recipe_thumbnail.dart recipes_grid_view.dart today_recipe_list_view.dart data message.dart models navigation screens cookipidia_theme.dart generated_plugin_registrant.dart main.dart test web

friend_post_tile.dart

```
lib > components > friend_post_tile.dart > ...
1 import 'package:flutter/material.dart';
2 import './components/components.dart';
3 import '../models/models.dart';
4
5 class FriendPostTile extends StatelessWidget {
6   final Post post;
7
8   const FriendPostTile({
9     Key? key,
10    required this.post,
11  }) : super(key: key);
12
13   @override
14   Widget build(BuildContext context) {
15     // 1
16     return Row(
17       mainAxisAlignment: MainAxisAlignment.start,
18       mainAxisSize: MainAxisSize.start,
19       children: [
20         // 2
21         CircleImage(
22           imageProvider: AssetImage(post.profileImageUrl),
23           imageRadius: 20,
24         ), // CircleImage
25         // 3
26         const SizedBox(width: 16),
27         // 4
28         Expanded(
29           // 5
30           child: Column(
31             mainAxisAlignment: MainAxisAlignment.start,
32             children: [
33               // 6
34               Text(post.comment),
35               // 7
36               Text(
37                 post.name,
38                 style: GoogleFonts.lato(
39                   color: Colors.black,
40                   fontWeight: FontWeight.w400,
41                   fontSize: 14,
42                 ),
43               ),
44             ],
45           ),
46         ),
47       ],
48     );
49   }
50 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 117.

Activities Visual Studio Code May 3 23:53 grocery_tile.dart - cookipidia - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER COOKIPIDIA build ios lib api components mock_cookipidia_service.dart card1.dart card2.dart card3.dart circle_image.dart components.dart Friend_post_list_view.dart friend_post_tile.dart grocery_tile.dart recipe_thumbnail.dart recipes_grid_view.dart today_recipe_list_view.dart data message.dart models navigation screens cookipidia_theme.dart generated_plugin_registrant.dart main.dart test web

grocery_tile.dart

```
lib > components > grocery_tile.dart > ...
14   final TextDecoration textDecoration;
15
16   // 4
17   GroceryTile({
18     Key? key,
19     required this.item,
20     this.onComplete,
21   }) : textDecoration =
22     item.isComplete ? TextDecoration.lineThrough : TextDecoration.none,
23   super(key: key);
24
25   @override
26   Widget build(BuildContext context) {
27     // TODO: Change this Widget
28     return SizedBox(
29       height: 100.0,
30
31       child: Row(
32         mainAxisAlignment: MainAxisAlignment.spaceBetween,
33         children: [
34           // 2
35           Container(
36             width: 5.0,
37             color: item.color,
38           ), // Container
39         ],
40         //const SizedBox(width: 16.0),
41       ),
42     );
43   }
44
45   Column(
46     mainAxisAlignment: MainAxisAlignment.center,
47     crossAxisAlignment: CrossAxisAlignment.start,
48     children: [
49       Text(
50         item.name,
51         style: GoogleFonts.lato(
52           color: Colors.black,
53           fontWeight: FontWeight.w400,
54           fontSize: 14,
55         ),
56       ),
57     ],
58   );
59 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64)

Fig 118.

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

class RecipeThumbnail extends StatelessWidget {
  final SimpleRecipe recipe;
  const RecipeThumbnail({
    Key? key,
    required this.recipe,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.all(8),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Expanded(
            child: ClipRRect(
              child: Image.asset(
                '$(recipe.dishImage)',
                fit: BoxFit.cover,
              ), // Image.asset
              borderRadius: BorderRadius.circular(12),
            ), // ClipRRect
          ), // Expanded
          const SizedBox(height: 10),
          Text(
            recipe.title,
          ),
        ],
      ),
    );
  }
}
```

Fig 119.

```
{
  "q": "chicken",
  "from": 0,
  "to": 10,
  "more": true,
  "count": 135812,
  "hits": [
    {
      "recipe": {
        "uri": "http://www.edamam.com/ontologies/edamam.owl#recipe_b79327d05b8e5b838ad6cf9576b30b6",
        "label": "Chicken Vesuvio",
        "image": "https://www.edamam.com/web-img/e42f9119813e890af34c259785ae1cfb.jpg",
        "source": "Serious Eats",
        "url": "http://www.seriouseats.com/recipes/2011/12/chicken-vesuvio-recipe.html",
        "shareAs": "http://www.edamam.com/recipe/chicken-vesuvio-b79327d05b8e5b838ad6cf9576b30b6/chicken",
        "yield": 4.0,
        "dietLabels": [
          "Low-Carb"
        ],
        "healthLabels": [
          "Peanut-Free",
          "Tree-Nut-Free"
        ],
        "cautions": [
          "Sulfites"
        ],
        "ingredientLines": [
          "1/2 cup olive oil",
          "5 cloves garlic, peeled",
          "2 large russet potatoes, peeled and cut into chunks",
          "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
          "3/4 cup white wine"
        ],
        "ingredientText": "1/2 cup olive oil\n5 cloves garlic, peeled\n2 large russet potatoes, peeled and cut into chunks\n1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)\n3/4 cup white wine"
      }
    }
  ]
}
```

Fig 120.

```
May 3 23:51
recipes1.json - cookipidia_test_1 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER
COOKIPIDIA_TEST_1
  .dart_tool
  .idea
  android
  assets
    images
    recipes1.json
    recipes2.json
  build
  ios
  lib
    data
      models
        Ingredient.dart
        models.dart
        recipe.dart
    moor
      moor_db.dart
      moor_db.g.dart
      moor_repository.dart
    sqlite
    memory_repository.dart
    repository.dart
  mock_service
  network
  ui
  generated_plugin_registrant.dart
  main.dart
  test
OUTLINE
TIMELINE
DEPENDENCIES
PROBLEMS 38
OUTPUT DEBUG CONSOLE TERMINAL
utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$ bash + x
Ln 18, Col 21 Spaces: 2 UTF-8 LF JSON Dart DevTools SM A750F (android-arm64) P D
main.dart  recipes1.json
assets > recipes1.json > hits > 0 > 0 recipe > dietLabels > 0
  "2 large russet potatoes, peeled and cut into chunks",
  "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
  "3/4 cup white wine",
  "3/4 cup chicken stock",
  "3 tablespoons chopped parsley",
  "1 tablespoon dried oregano",
  "Salt and pepper",
  "1 cup frozen peas, thawed"
],
"ingredients": [
  {
    "text": "1/2 cup olive oil",
    "weight": 108.0
  },
  {
    "text": "5 cloves garlic, peeled",
    "weight": 15.0
  },
  {
    "text": "2 large russet potatoes, peeled and cut into chunks",
    "weight": 738.0
  },
  {
    "text": "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
    "weight": 1587.573295000001
  },
  {
    "text": "3/4 cup white wine",
    "weight": 176.3999999999998
  },
  {
    "text": "3/4 cup chicken stock",
    "weight": 108.0
  }
]
```

Fig 121.

```
May 3 23:51
recipes2.json - cookipidia_test_1 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER
COOKIPIDIA_TEST_1
  lib > main.dart > MyApp > build
    Logger.root.level = Level.ALL;
    Logger.root.onRecord.listen((rec) {
      print('${rec.level.name}: ${rec.time}: ${rec.message}');
    });
  }
  class MyApp extends StatelessWidget {
    final Repository repository;
    const MyApp({
      Key key,
      required this.repository,
    }) : super(key: key);

    // This widget is the root of your application.
    @override
    Widget build(BuildContext context) {
      return MultiProvider(
        providers: [
          // 1
          Provider<Repository>(
            lazy: false,
            create: () => repository,
            dispose: (_, Repository repository) => repository.close(),
          ), // Provider
          // 2
          Provider<ServiceInterface>(
            create: () => RecipeService.create(),
            lazy: false,
          ), // Provider
        ],
      );
    }
  }
}

PROBLEMS 38
OUTPUT DEBUG CONSOLE TERMINAL
utkarsh@utkarsh-nitro:~/AndroidStudioProjects/cookipidia_test_1$ bash + x
Ln 41, Col 5 Spaces: 2 UTF-8 LF Dart DevTools Flutter: 2.10.4 SM A750F (android-arm64) P D
main.dart  recipes2.json
```

Fig 122.

```
main.dart      pubspec.yaml
28 # versions available, run `flutter pub outdated`.
29 dependencies:
30   flutter:
31     sdk: flutter
32
33   # The following adds the Cupertino Icons font to your application.
34   # Use with the CupertinoIcons class for iOS style icons.
35   cupertino_icons: ^1.0.2
36   cached_network_image: ^3.1.0
37   flutter_slidable: ^0.6.0
38   flutter_svg: ^0.22.0
39   shared_preferences: ^2.0.13
40   json_annotation: ^4.4.0
41   chopper: ^4.0.5
42   logging: ^1.0.2
43   provider: ^6.0.2
44   equatable: ^2.0.3
45
46   sqflite: ^2.0.2
47   path_provider: ^2.0.9
48   synchronized: ^3.0.0+2
49   sqflite: ^2.2.0
50
51   moor_flutter: ^4.1.0
52
53   dev_dependencies:
54     flutter_test:
55       sdk: flutter
56
57   # The "flutter_lints" package below contains a set of recommended lints to
58   # encourage good coding practices. The Lint set provided by the package is
59   # activated in the "analysis_options.yaml" file located at the root of your
60   # project.
61
62
63
64
65
66
67
68
69
69
```

Fig 123.

```
manifest.json
{
  "web": {
    "icons": [
      {
        "src": "icons/Icon-maskable-512.png",
        "sizes": "512x512",
        "type": "image/png",
        "purpose": "maskable"
      }
    ]
  }
}
```

Fig 124.

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

RAYCHAT

- ui
 - login.dart
 - message_list.dart
 - message_widget.dart
 - generated_plugin_registrant.dart
 - main.dart
- test
- web
 - Icons
 - Icon-192.png
 - Icon-512.png
 - Icon-maskable-192.png
 - Icon-maskable-512.png
 - favicon.png
 - index.html
 - manifest.json
- windows
 - flutter
 - ephemeral/.plugin_symlinks

CMakeLists.txt

```

windows > flutter > CMakeLists.txt
1 cmake_minimum_required(VERSION 3.14)
2
3 set(EPHEMERAL_DIR "${CMAKE_CURRENT_SOURCE_DIR}/ephemeral")
4
5 # Configuration provided via flutter tool.
6 include(${EPHEMERAL_DIR}/generated_config.cmake)
7
8 # TODO: Move the rest of this into files in ephemeral. See
9 # https://github.com/flutter/flutter/issues/57146.
10 set(WRAPPER_ROOT "${EPHEMERAL_DIR}/cpp_client_wrapper")
11
12 # === Flutter Library ===
13 set(FLUTTER_LIBRARY "${EPHEMERAL_DIR}/flutter_windows.dll")
14
15 # Published to parent scope for install step.
16 set(FLUTTER_LIBRARY ${FLUTTER_LIBRARY} PARENT_SCOPE)
17 set(FLUTTER_ICU_DATA_FILE "${EPHEMERAL_DIR}/icudtl.dat" PARENT_SCOPE)
18 set(PROJECT_BUILD_DIR "${PROJECT_DIR}/build/" PARENT_SCOPE)
19 set(AOT_LIBRARY "${PROJECT_DIR}/build/windows/app.so" PARENT_SCOPE)
20
21 list(APPEND FLUTTER_LIBRARY_HEADERS
22     "flutter_export.h"
23     "flutter_windows.h"
24     "flutter_messenger.h"
25     "flutter_plugin_registrar.h"
26     "flutter_texture_registrar.h"
27 )
28 list(TRANSFORM FLUTTER_LIBRARY_HEADERS PREPEND "${EPHEMERAL_DIR}/")
29 add_library(flutter INTERFACE)
30 target_include_directories(flutter INTERFACE
31     "${EPHEMERAL_DIR}"
32 )

```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF CMake Dart DevTools SM A750F (android-arm64) bash

Fig 125.

Activities Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

RAYCHAT

- ui
 - login.dart
 - message_list.dart
 - message_widget.dart
 - generated_plugin_registrant.dart
 - main.dart
- test
- web
 - Icons
 - Icon-192.png
 - Icon-512.png
 - Icon-maskable-192.png
 - Icon-maskable-512.png
 - favicon.png
 - index.html
 - manifest.json
- windows
 - flutter
 - ephemeral/.plugin_symlinks

generated_plugin_registrant.cc

```

windows > flutter > generated_plugin_registrant.cc ...
1 /**
2 // Generated file. Do not edit.
3 /**
4 /**
5 // clang-format off
6
7 #include "generated_plugin_registrant.h"
8
9 void RegisterPlugins/flutter::PluginRegistry* registry) {
10 }
11
12

```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 4 UTF-8 LF C++ Dart DevTools SM A750F (android-arm64) Linux bash

Fig 126.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT".
- Editor:** The current file is "generated_plugin_registrant.h". The code content is as follows:

```
// Generated file. Do not edit.
#ifndef GENERATED_PLUGIN_REGISTRANT_
#define GENERATED_PLUGIN_REGISTRANT_
#include <flutter/plugin_registry.h>
// Registers Flutter plugins.
void RegisterPlugins(flutter::PluginRegistry* registry);
#endif // GENERATED_PLUGIN_REGISTRANT_
```

- Terminal:** bash
- Bottom Bar:** PROBLEMS (14), OUTPUT, DEBUG CONSOLE, TERMINAL, utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$
- Bottom Status Bar:** Ln 1, Col 1, Spaces: 4, UTF-8, LF, C++, Dart DevTools, SM A750F (android-arm64), Linux

Fig 127.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "runner".
- Editor:** The current file is "CMakeLists.txt". The code content is as follows:

```
cmake_minimum_required(VERSION 3.14)
project(runner LANGUAGES CXX)

add_executable(${BINARY_NAME} WIN32
    "flutter_window.cpp"
    "main.cpp"
    "utils.cpp"
    "win32_window.cpp"
    "${FLUTTER_MANAGED_DIR}/generated_plugin_registrant.cc"
    "Runner.rc"
    "runner.exe.manifest"
)
apply_standard_settings(${BINARY_NAME})
target_compile_definitions(${BINARY_NAME} PRIVATE "NOMINMAX")
target_link_libraries(${BINARY_NAME} PRIVATE flutter flutter_wrapper_app)
target_include_directories(${BINARY_NAME} PRIVATE "${CMAKE_SOURCE_DIR}")
add_dependencies(${BINARY_NAME} flutter assemble)
```

- Terminal:** bash
- Bottom Bar:** PROBLEMS (13), OUTPUT, DEBUG CONSOLE, TERMINAL, utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$
- Bottom Status Bar:** Ln 1, Col 1, Spaces: 2, UTF-8, LF, CMake, Dart DevTools, SM A750F (android-arm64)

Fig 128.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Bar:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT".
- Editor:** The current file is "flutter_window.cpp". The code is as follows:

```
windows > runner > flutter_window.cpp
1 #include "flutter_window.h"
2
3 #include <optional>
4
5 #include "flutter/generated_plugin_registrant.h"
6
7 FlutterWindow::FlutterWindow(const flutter::DartProject& project)
8     : project_(project) {}
9
10 FlutterWindow::~FlutterWindow() {}
11
12 bool FlutterWindow::OnCreate() {
13     if (!Win32Window::OnCreate()) {
14         return false;
15     }
16
17     RECT frame = GetClientArea();
18
19     // The size here must match the window dimensions to avoid unnecessary surface
20     // creation / destruction in the startup path.
21     flutter_controller_ = std::make_unique<flutter::FlutterViewController>(
22         frame.right - frame.left, frame.bottom - frame.top, project_);
23     // Ensure that basic setup of the controller was successful.
24     if (!flutter_controller_->engine() || !flutter_controller_->view()) {
25         return false;
26     }
27     RegisterPlugins(flutter_controller_->engine());
28     SetChildContent(flutter_controller_->view()->GetNativeWindow());
29     return true;
30 }
31
32 void FlutterWindow::OnDestroy() {
33
34 }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 }
```

- Terminal:** Shows the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom:** A row of icons for various applications like Spotify, Google Chrome, and others.

Fig 129.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Bar:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT".
- Editor:** The current file is "flutter_window.cpp". The code is as follows, with a syntax error at line 32:

```
windows > runner > flutter_window.cpp >...
1 #include "flutter_window.h";
2
3 #include <optional>;
4
5 #include "flutter/generated_plugin_registrant.h";
6
7 FlutterWindow::FlutterWindow(const flutter::DartProject& project)
8     : project_(project) {}
9
10 FlutterWindow::~FlutterWindow() {}
11
12 bool FlutterWindow::OnCreate() {
13     if (!Win32Window::OnCreate()) {
14         return false;
15     }
16
17     RECT frame = GetClientArea();
18
19     // The size here must match the window dimensions to avoid unnecessary surface
20     // creation / destruction in the startup path.
21     flutter_controller_ = std::make_unique<flutter::FlutterViewController>(
22         frame.right - frame.left, frame.bottom - frame.top, project_);
23     // Ensure that basic setup of the controller was successful.
24     if (!flutter_controller_->engine() || !flutter_controller_->view()) {
25         return false;
26     }
27     RegisterPlugins(flutter_controller_->engine());
28     SetChildContent(flutter_controller_->view()->GetNativeWindow());
29     return true;
30 }
31
32 void FlutterWindow::OnDestroy() {
33
34 }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 }
```

- Terminal:** Shows the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom:** A row of icons for various applications like Spotify, Google Chrome, and others.

Fig 130.

A screenshot of the Visual Studio Code interface. The title bar says "Activities" and "Visual Studio Code". The status bar at the top right shows "May 4 16:26" and "flutter_window.cpp - raychat - Visual Studio Code". The main area shows the code for `flutter_window.cpp`. The code handles window messages, specifically `WM_FONTCCHANGE`, by calling `flutter_controller_>engine()>ReloadSystemFonts()`. The code editor has syntax highlighting for C++ and Dart. The Explorer sidebar on the left shows the project structure for "RAYCHAT" with files like `Icon-192.png`, `Icon-512.png`, `index.html`, `manifest.json`, `flutter`, `runner`, and `flutter_window.cpp`. The bottom taskbar shows various icons for other applications.

Fig 131.

A screenshot of the Visual Studio Code interface, similar to Fig 131. The title bar says "Activities" and "Visual Studio Code". The status bar at the top right shows "May 4 16:26" and "flutter_window.h - raychat - Visual Studio Code". The main area shows the code for `flutter_window.h`. It defines a class `FlutterWindow` that inherits from `Win32Window`. The class has a protected constructor that takes a `DartProject` parameter and calls `FlutterWindow(const flutter::DartProject& project)`. It also has a private member `flutter::FlutterViewController flutter_controller_`. The code editor has syntax highlighting for C++ and Dart. The Explorer sidebar on the left shows the project structure for "RAYCHAT" with files like `Icon-192.png`, `Icon-512.png`, `index.html`, `manifest.json`, `flutter`, `runner`, and `flutter_window.h`. The bottom taskbar shows various icons for other applications.

Fig 132.

The screenshot shows the Visual Studio Code interface with the file `flutter_window.h` open in the editor. The code defines a class `FlutterWindow` with methods for creating a window, handling messages, and running a Dart project. The code is heavily annotated with comments explaining its purpose.

```
windows > runner > C:\flutter\flutter_window.h > ...
14 // Creates a new FlutterWindow hosting a Flutter view running [project].
15 explicit FlutterWindow(const flutter::DartProject& project);
16 virtual ~FlutterWindow();
17
18 protected:
19 // Win32Window:
20 bool OnCreate() override;
21 void OnDestroy() override;
22 LRESULT MessageHandler(HWND window, UINT const message, WPARAM const wparam,
23 const LPARAM const lparam) noexcept override;
24
25 private:
26 // The project to run.
27 flutter::DartProject project_;
28
29 // The Flutter instance hosted by this window.
30 std::unique_ptr<flutter::FlutterViewController> flutter_controller_;
31 }
32
33 #endif // RUNNER_FLUTTER_WINDOW_H_
34
```

Fig 133.

The screenshot shows the Visual Studio Code interface with the file `main.cpp` open in the editor. The code implements the `wWinMain` function, which initializes COM, creates a `FlutterWindow`, and runs a Dart project named "data". It also handles command-line arguments and attaches to a debugger if present.

```
windows > runner > C:\main.cpp > ...
1 #include <flutter/dart_project.h>
2 #include <flutter/flutter_view_controller.h>
3 #include <windows.h>
4
5 #include "flutter_window.h"
6 #include "utils.h"
7
8 int APIENTRY wWinMain(_In_ HINSTANCE instance, _In_opt_ HINSTANCE prev,
9                      _In_ wchar_t *command_line, _In_ int show_command) {
10    // Attach to console when present (e.g., 'flutter run') or create a
11    // new console when running with a debugger.
12    if (::AttachConsole(ATTACH_PARENT_PROCESS) && ::IsDebuggerPresent()) {
13        CreateAndAttachConsole();
14    }
15
16    // Initialize COM, so that it is available for use in the library and/or
17    // plugins.
18    ::CoInitializeEx(nullptr, COINIT_APARTMENTTHREADED);
19
20    flutter::DartProject project(L"data");
21
22    std::vector<std::string> command_line_arguments =
23        GetCommandLineArguments();
24
25    project.set_dart_entrypoint_arguments(std::move(command_line_arguments));
26
27    FlutterWindow window(project);
28    Win32Window::Point origin(10, 10);
29    Win32Window::Size size(1280, 720);
30    if (!window.CreateAndShow(L"raychat", origin, size)) {
31        return EXIT_FAILURE;
32    }
33}
```

Fig 134.

```
14     }
15
16 // Initialize COM, so that it is available for use in the library and/or
17 // plugins.
18 ::CoInitializeEx(nullptr, COINIT_APARTMENTTHREADED);
19
20 flutter::DartProject project(L"data");
21
22 std::vector<std::string> command_line_arguments =
23     GetCommandLineArguments();
24
25 project.set_dart_entrypoint_arguments(std::move(command_line_arguments));
26
27 FlutterWindow window(project);
28 Win32Window::Point origin(10, 10);
29 Win32Window::Size size(1280, 720);
30 if (!window.CreateAndShow(L"raychat", origin, size)) {
31     return EXIT_FAILURE;
32 }
33 window.SetQuitOnClose(true);
34
35 ::MSG msg;
36 while (::GetMessage(&msg, nullptr, 0, 0)) {
37     ::TranslateMessage(&msg);
38     ::DispatchMessage(&msg);
39 }
40 ::CoUninitialize();
41 return EXIT_SUCCESS;
42
43 }
```

Fig 135.

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code - May 4 16:26 - resource.h - raychat - Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT". The "resource.h" file is selected in the Explorer sidebar.
- Editor:** Displays the content of "resource.h". The line "#define IDI_APP_ICON" is highlighted with a red dotted underline, indicating a potential issue or warning.
- Bottom Status Bar:** Shows the current file path as "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom Icons:** Includes icons for GitHub, Git, Gitignore, Packages, Analysis, Pubspec.lock, Pubspec.yaml, and Raychat.yaml.

Fig 136.

Activities Visual Studio Code ▾ May 4 16:26

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT

```
windows > runner > Runner.rc
09 #define FLUTTER_BUILD_NAME
10 #define VERSION_AS_STRING #FLUTTER_BUILD_NAME
11 #else
12 #define VERSION_AS_STRING "1.0.0"
13 #endif
14
15 VS_VERSION_INFO VERSIONINFO
16 FILEVERSION VERSION AS NUMBER
17 PRODUCTVERSION VERSION AS NUMBER
18 FILEFLAGSMASK VS_FFI_FILEFLAGSMASK
19 #ifdef _DEBUG
20 FILEFLAGS VS_FF_DEBUG
21 #else
22 FILEFLAGS 0x0L
23 #endif
24 FILEOS VOS_WINDOWS32
25 FILETYPE VFT_APP
26 FILESUBTYPE 0x0L
27 BEGIN
28     BLOCK "StringFileInfo"
29     BEGIN
30         BLOCK "040904e4"
31         BEGIN
32             VALUE "CompanyName", "com.example" "\0"
33             VALUE "FileDescription", "raychat" "\0"
34             VALUE "FileVersion", VERSION_AS_STRING "\0"
35             VALUE "InternalName", "raychat" "\0"
36             VALUE "LegalCopyright", "Copyright (C) 2022 com.example. All rights reserved." "\0"
37             VALUE "OriginalFilename", "raychat.exe" "\0"
38             VALUE "ProductName", "raychat" "\0"
39             VALUE "ProductVersion", VERSION_AS_STRING "\0"
40         END
41     END
42 END
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text Dart DevTools SM A750F (android-arm64)

Fig 137.

Activities Visual Studio Code ▾ May 4 16:26

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT

```
windows > runner > runner.exe.manifest
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
3     <application xmlns="urn:schemas-microsoft-com:asm.v3">
4         <windowsSettings>
5             <dpiAwareness xmlns="http://schemas.microsoft.com/SMI/2016/WindowsSettings">PerMonitorV2</dpiAwareness>
6         </windowsSettings>
7     </application>
8     <compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
9         <!-- Windows 10 -->
10        <supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}" />
11        <!-- Windows 8.1 -->
12        <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}" />
13        <!-- Windows 8 -->
14        <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}" />
15        <!-- Windows 7 -->
16        <supportedOS Id="{35138b9a-5d96-4fb0-8e2d-a2440225f93a}" />
17    </compatibility>
18 </application>
19 </assembly>
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF XML Dart DevTools SM A750F (android-arm64)

Fig 138.

```

May 4 16:26
utils.h - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER windows > runner > utils.h ...
utils.h
1 #ifndef RUNNER_UTILS_H_
2 #define RUNNER_UTILS_H_
3
4 #include <string>
5 #include <vector>
6
7 // Creates a console for the process, and redirects stdout and stderr to
8 // it for both the runner and the Flutter library.
9 void CreateAndAttachConsole();
10
11 // Takes a null-terminated wchar_t* encoded in UTF-16 and returns a std::string
12 // encoded in UTF-8. Returns an empty std::string on failure.
13 std::string Utf8FromUtf16(const wchar_t* utf16_string);
14
15 // Gets the command line arguments passed in as a std::vector<std::string>,
16 // encoded in UTF-8. Returns an empty std::vector<std::string> on failure.
17 std::vector<std::string> GetCommandLineArguments();
18
19#endif // RUNNER_UTILS_H_
20

```

The Marketplace has extensions that can help with '.manifest' files

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 4 UTF-8 LF C++ Dart DevTools SM A750F (android-arm64) Linux

Fig 139.

```

May 4 16:26
utils.cpp - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER windows > runner > utils.cpp ...
utils.cpp
1 #include "utils.h"
2
3 #include <flutter_windows.h>
4 #include <io.h>
5 #include <stdio.h>
6 #include <windows.h>
7
8 #include <iostream>
9
10 void CreateAndAttachConsole() {
11     if (::AllocConsole()) {
12         FILE *unused;
13         if (freopen_s(&unused, "CONOUT$", "w", stdout)) {
14             _dup2(_fileno(stdout), 1);
15         }
16         if (freopen_s(&unused, "CONOUT$", "w", stderr)) {
17             _dup2(_fileno(stderr), 2);
18         }
19         std::ios::sync_with_stdio();
20         FlutterDesktopResyncOutputStreams();
21     }
22 }
23
24 std::vector<std::string> GetCommandLineArguments() {
25     // Convert the UTF-16 command line arguments to UTF-8 for the Engine to use.
26     int argc;
27     wchar_t** argv = ::CommandLineToArgvW(::GetCommandLineW(), &argc);
28     if (argv == nullptr) {
29         return std::vector<std::string>();
30     }
31     std::vector<std::string> command_line_arguments;
32 }
```

The Marketplace has extensions that can help with '.manifest' files

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF C++ Dart DevTools SM A750F (android-arm64) Linux

Fig 140.

```
windows > runner > C:\utils.cpp > ...
27     wchar_t** argv = ::COMMANDLINE::Argv();
28     if (argv == nullptr) {
29         return std::vector<std::string>();
30     }
31
32     std::vector<std::string> command_line_arguments;
33
34     // Skip the first argument as it's the binary name.
35     for (int i = 1; i < argc; i++) {
36         command_line_arguments.push_back(Utf8FromUtf16(argv[i]));
37     }
38
39     ::LocalFree(argv);
40
41     return command_line_arguments;
42 }
43
44 std::string Utf8FromUtf16(const wchar_t* utf16_string) {
45     if (utf16_string == nullptr) {
46         return std::string();
47     }
48     int target_length = ::WideCharToMultiByte(
49         CP_UTF8, WC_ERR_INVALID_CHARS, utf16_string,
50         -1, nullptr, 0, nullptr, nullptr);
51     if (target_length == 0) {
52         return std::string();
53     }
54     std::string utf8_string;
55     utf8_string.resize(target_length);
56     int converted_length = ::WideCharToMultiByte(
57         CP_UTF8, WC_ERR_INVALID_CHARS, utf16_string,
58         -1, utf8_string.data(), target_length, nullptr, nullptr);
59     if (converted_length == 0) {
60         return std::string();
61     }
62     return utf8_string;
63 }
64 }
```

The screenshot shows the Visual Studio Code interface with the code editor open to `utils.cpp` in the `runner` directory of a project named `RAYCHAT`. The code implements a function to parse command-line arguments from a Windows command-line interface. It uses `WideCharToMultiByte` to convert wide-character strings to UTF-8 strings. A tooltip in the bottom right corner suggests Marketplace extensions for `.manifest` files.

Fig 141.

```
windows > runner > C:\utils.cpp > ...
54     std::string utf8_string;
55     utf8_string.resize(target_length);
56     int converted_length = ::WideCharToMultiByte(
57         CP_UTF8, WC_ERR_INVALID_CHARS, utf16_string,
58         -1, utf8_string.data(), target_length, nullptr, nullptr);
59     if (converted_length == 0) {
60         return std::string();
61     }
62     return utf8_string;
63 }
64 }
```

This screenshot is nearly identical to Fig 141, showing the same code in `utils.cpp` for a Windows runner. The line numbers have changed, likely due to a code refactoring or a different commit. The code remains the same: reading command-line arguments and converting them from UTF-16 to UTF-8 using `WideCharToMultiByte`.

Fig 142.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Explorer:** Shows the project structure for "RAYCHAT". The "win32_window.h" file is selected.
- Code Editor:** Displays the content of the "win32_window.h" file. The code defines a class `Win32Window` with methods for creating windows based on position and size.
- Terminal:** Shows the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom Bar:** Includes icons for various applications like Spotify, Google Chrome, and Microsoft Edge.

Fig 143.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Explorer:** Shows the project structure for "RAYCHAT". The "win32_window.h" file is selected.
- Code Editor:** Displays the content of the "win32_window.h" file. The code includes a new method `CreateAndShow` which takes a title, origin, and size, and returns a boolean indicating success.
- Terminal:** Shows the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom Bar:** Includes icons for various applications like Spotify, Google Chrome, and Microsoft Edge.

Fig 144.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT". The "runner" folder is expanded, showing files like generated_plugin_registrant.cc, generated_plugin_registrant.h, generated_plugins.cmake, flutter_window.cpp, flutter_window.h, main.cpp, resource.h, runner.exe.manifest, Runner.rc, utils.cpp, utils.h, win32_window.cpp, and win32_window.h.
- Code Editor:** The file "win32_window.h" is open. The code defines a class "Win32Window" with various methods and a static callback function "WndProc".
- Terminal:** The terminal tab is active, showing the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom Bar:** Shows various application icons and the status bar indicating "Ln 33, Col 73" and "UTF-8".

Fig 145.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT". The "runner" folder is expanded, showing files like generated_plugin_registrant.cc, generated_plugin_registrant.h, generated_plugins.cmake, flutter_window.cpp, flutter_window.h, main.cpp, resource.h, runner.exe.manifest, Runner.rc, utils.cpp, utils.h, win32_window.cpp, and win32_window.h.
- Code Editor:** The file "win32_window.h" is open. The code has been modified to include additional members and methods, including a "GetThisFromHandle" static member and a "quit_on_close_" boolean variable.
- Terminal:** The terminal tab is active, showing the command "utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$".
- Bottom Bar:** Shows various application icons and the status bar indicating "Ln 33, Col 73" and "UTF-8".

Fig 146.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Explorer:** Shows the project structure under "RAYCHAT". The "runner" folder is expanded, showing files like app_icon.ico, CMakeLists.txt, flutter_window.cpp, flutter_window.h, main.cpp, resource.h, runner.exe.manifest, Runner.rc, utils.cpp, utils.h, win32_window.cpp, and win32_window.h.
- Code Editor:** The "win32_window.h" file is open. The code defines a class `win32Window` with methods for handling messages and retrieving class instances. It also includes definitions for `window_handle_` and `child_content_`.
- Terminal:** The terminal tab is active, showing the command `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat$`.
- Bottom Bar:** Shows various application icons and the status bar indicating "Ln 33, Col 73" and "UTF-8".

Fig 147.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Explorer:** Shows the project structure under "RAYCHAT". The "runner" folder is expanded, showing files like app_icon.ico, CMakeLists.txt, flutter_window.cpp, flutter_window.h, main.cpp, resource.h, runner.exe.manifest, Runner.rc, utils.cpp, utils.h, win32_window.cpp, and win32_window.h.
- Code Editor:** The "CMakeLists.txt" file is open. The code defines a CMake project named "raychat" with binary name "raychat". It sets up build options and handles multiconfiguration using `if(IS_MULTI_CONFIG)` blocks.
- Terminal:** The terminal tab is active, showing the command `utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat$`.
- Bottom Bar:** Shows various application icons and the status bar indicating "Ln 33, Col 1" and "UTF-8".

Fig 148.

Activities Visual Studio Code May 4 16:26 CMakeLists.txt - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT CMakeLists.txt

```

windows > CMakeLists.txt
14   CACHE STRING "" FORCE)
15 else()
16   if(NOT CMAKE_BUILD_TYPE AND NOT CMAKE_CONFIGURATION_TYPES)
17     set(CMAKE_BUILD_TYPE "Debug" CACHE
18       STRING "Flutter build mode" FORCE)
19     set_property(CACHE CMAKE_BUILD_TYPE PROPERTY STRINGS
20       "Debug" "Profile" "Release")
21   endif()
22 endif()

23 set(CMAKE_EXE_LINKER_FLAGS_PROFILE "${CMAKE_EXE_LINKER_FLAGS_RELEASE}")
24 set(CMAKE_SHARED_LINKER_FLAGS_PROFILE "${CMAKE_SHARED_LINKER_FLAGS_RELEASE}")
25 set(CMAKE_C_FLAGS_PROFILE "${CMAKE_C_FLAGS_RELEASE}")
26 set(CMAKE_CXX_FLAGS_PROFILE "${CMAKE_CXX_FLAGS_RELEASE}")

27 # Use Unicode for all projects.
28 add_definitions(-DUNICODE -D UNICODE)

29 # Compilation settings that should be applied to most targets.
30 function(APPLY_STANDARD_SETTINGS TARGET)
31   target_compile_features(${TARGET} PUBLIC cxx_std_17)
32   target_compile_options(${TARGET} PRIVATE /W4 /WX /wd"4100")
33   target_compile_options(${TARGET} PRIVATE /EHsc)
34   target_compile_definitions(${TARGET} PRIVATE "_HAS_EXCEPTIONS=0")
35   target_compile_definitions(${TARGET} PRIVATE "$<<CONFIG:Debug>:_DEBUG")
36 endfunction()

37 set(FLUTTER_MANAGED_DIR "${CMAKE_CURRENT_SOURCE_DIR}/flutter")
38
39 # Flutter library and tool build rules.
40 add_subdirectory(${FLUTTER_MANAGED_DIR})

41 # Application build
42 add_subdirectory("runner")

43 # Generated plugin build rules, which manage building the plugins and adding
44 # them to the application.
45 include/flutter/generated_plugins.cmake

46 # === Installation ===
47 # Support files are copied into place next to the executable, so that it can
48 # run in place. This is done instead of making a separate bundle (as on Linux)
49 # so that building and running from within Visual Studio will work.
50 set(BUILD_BUNDLE_DIR "$<TARGET_FILE_DIR:${BINARY_NAME}>")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF CMake Dart DevTools SM A750F (android-arm64) bash

Fig 149.

Activities Visual Studio Code May 4 16:26 CMakeLists.txt - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT CMakeLists.txt

```

windows > CMakeLists.txt
14   CACHE STRING "" FORCE)
15 else()
16   if(NOT CMAKE_BUILD_TYPE AND NOT CMAKE_CONFIGURATION_TYPES)
17     set(CMAKE_BUILD_TYPE "Debug" CACHE
18       STRING "Flutter build mode" FORCE)
19     set_property(CACHE CMAKE_BUILD_TYPE PROPERTY STRINGS
20       "Debug" "Profile" "Release")
21   endif()
22 endif()

23 set(CMAKE_EXE_LINKER_FLAGS_PROFILE "${CMAKE_EXE_LINKER_FLAGS_RELEASE}")
24 set(CMAKE_SHARED_LINKER_FLAGS_PROFILE "${CMAKE_SHARED_LINKER_FLAGS_RELEASE}")
25 set(CMAKE_C_FLAGS_PROFILE "${CMAKE_C_FLAGS_RELEASE}")
26 set(CMAKE_CXX_FLAGS_PROFILE "${CMAKE_CXX_FLAGS_RELEASE}")

27 # Use Unicode for all projects.
28 add_definitions(-DUNICODE -D UNICODE)

29 # Compilation settings that should be applied to most targets.
30 function(APPLY_STANDARD_SETTINGS TARGET)
31   target_compile_features(${TARGET} PUBLIC cxx_std_17)
32   target_compile_options(${TARGET} PRIVATE /W4 /WX /wd"4100")
33   target_compile_options(${TARGET} PRIVATE /EHsc)
34   target_compile_definitions(${TARGET} PRIVATE "_HAS_EXCEPTIONS=0")
35   target_compile_definitions(${TARGET} PRIVATE "$<<CONFIG:Debug>:_DEBUG")
36 endfunction()

37 set(FLUTTER_MANAGED_DIR "${CMAKE_CURRENT_SOURCE_DIR}/flutter")
38
39 # Flutter library and tool build rules.
40 add_subdirectory(${FLUTTER_MANAGED_DIR})

41 # Application build
42 add_subdirectory("runner")

43 # Generated plugin build rules, which manage building the plugins and adding
44 # them to the application.
45 include/flutter/generated_plugins.cmake

46 # === Installation ===
47 # Support files are copied into place next to the executable, so that it can
48 # run in place. This is done instead of making a separate bundle (as on Linux)
49 # so that building and running from within Visual Studio will work.
50 set(BUILD_BUNDLE_DIR "$<TARGET_FILE_DIR:${BINARY_NAME}>")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF CMake Dart DevTools SM A750F (android-arm64) bash

Fig 150.

Activities Visual Studio Code May 4 16:26 CMakeLists.txt - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT windows > CMakeLists.txt

```
41 set(FLUTTER_MANAGED_DIR "${CMAKE_CURRENT_SOURCE_DIR}/flutter")
42
43 # Flutter library and tool build rules.
44 add_subdirectory(${FLUTTER_MANAGED_DIR})
45
46 # Application build
47 add_subdirectory("runner")
48
49 # Generated plugin build rules, which manage building the plugins and adding
50 # them to the application.
51 include/flutter/generated_plugins.cmake)
52
53
54 # === Installation ===
55 # Support files are copied into place next to the executable, so that it can
56 # run in place. This is done instead of making a separate bundle (as on Linux)
57 # so that building and running from within Visual Studio will work.
58 set(BUILD_BUNDLE_DIR "$<TARGET_FILE_DIR:$<BINARY_NAME>")
59 # Make the "install" step default, as it's required to run.
60 set(CMAKE_VS_INCLUDE_INSTALL_TO_DEFAULT_BUILD 1)
61 if(CMAKE_INSTALL_PREFIX_INITIALIZED_TO_DEFAULT)
62 set(CMAKE_INSTALL_PREFIX "${BUILD_BUNDLE_DIR}" CACHE PATH "..." FORCE)
63 endif()
64
65 set(INSTALL_BUNDLE_DATA_DIR "${CMAKE_INSTALL_PREFIX}/data")
66 set(INSTALL_BUNDLE_LIB_DIR "${CMAKE_INSTALL_PREFIX}")
67
68 install(TARGETS ${BINARY_NAME} RUNTIME DESTINATION "${CMAKE_INSTALL_PREFIX}"
69 COMPONENT Runtime)
70
71 install(FILES "${FLUTTER_ICU_DATA_FILE}" DESTINATION "${INSTALL_BUNDLE_DATA_DIR}"
72 COMPONENT Runtime)
73
74 install(FILES "${FLUTTER_LIBRARY}" DESTINATION "${INSTALL_BUNDLE_LIB_DIR}"
75 COMPONENT Runtime)
76
77 if(PLUGIN_BUNDLED_LIBRARIES)
78 install(FILES ${PLUGIN_BUNDLED_LIBRARIES}
79 DESTINATION "${INSTALL_BUNDLE_LIB_DIR}"
80 COMPONENT Runtime)
81 endif()
82
83 # Fully re-copy the assets directory on each build to avoid having stale files
84 # from a previous install.
85 set(FLUTTER_ASSET_DIR_NAME "flutter_assets")
86 install(CODE "
87 file(REMOVE_RECURSE \\"${INSTALL_BUNDLE_DATA_DIR}/${FLUTTER_ASSET_DIR_NAME}\\" )
88 # COMPONENT Runtime)
89 install(DIRECTORY "${PROJECT_BUILD_DIR}/${FLUTTER_ASSET_DIR_NAME}"
90 DESTINATION "${INSTALL_BUNDLE_DATA_DIR}" COMPONENT Runtime)
91
92 # Install the AOT library on non-Debug builds only.
93 install(FILES "${AOT_LIBRARY}" DESTINATION "${INSTALL_BUNDLE_DATA_DIR}"
94 CONFIGURATIONS Profile;Release
95 COMPONENT Runtime)
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF CMake Dart DevTools SM A750F (android-arm64) bash

Fig 151.

Activities Visual Studio Code May 4 16:26 CMakeLists.txt - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER RAYCHAT windows > CMakeLists.txt

```
67
68 install(TARGETS ${BINARY_NAME} RUNTIME DESTINATION "${CMAKE_INSTALL_PREFIX}"
69 COMPONENT Runtime)
70
71 install(FILES "${FLUTTER_ICU_DATA_FILE}" DESTINATION "${INSTALL_BUNDLE_DATA_DIR}"
72 COMPONENT Runtime)
73
74 install(FILES "${FLUTTER_LIBRARY}" DESTINATION "${INSTALL_BUNDLE_LIB_DIR}"
75 COMPONENT Runtime)
76
77 if(PLUGIN_BUNDLED_LIBRARIES)
78 install(FILES ${PLUGIN_BUNDLED_LIBRARIES}
79 DESTINATION "${INSTALL_BUNDLE_LIB_DIR}"
80 COMPONENT Runtime)
81 endif()
82
83 # Fully re-copy the assets directory on each build to avoid having stale files
84 # from a previous install.
85 set(FLUTTER_ASSET_DIR_NAME "flutter_assets")
86 install(CODE "
87 file(REMOVE_RECURSE \\"${INSTALL_BUNDLE_DATA_DIR}/${FLUTTER_ASSET_DIR_NAME}\\" )
88 # COMPONENT Runtime)
89 install(DIRECTORY "${PROJECT_BUILD_DIR}/${FLUTTER_ASSET_DIR_NAME}"
90 DESTINATION "${INSTALL_BUNDLE_DATA_DIR}" COMPONENT Runtime)
91
92 # Install the AOT library on non-Debug builds only.
93 install(FILES "${AOT_LIBRARY}" DESTINATION "${INSTALL_BUNDLE_DATA_DIR}"
94 CONFIGURATIONS Profile;Release
95 COMPONENT Runtime)
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL

utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$

Ln 1, Col 1 Spaces: 2 UTF-8 LF CMake Dart DevTools SM A750F (android-arm64) bash

Fig 152.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT". It includes subfolders like .dart_tool, .idea, .vscode, android, app, gradle, .gitignore, build.gradle, gradle.properties, gradlew, gradlew.bat, local.properties, and raychat_android.iml.
- Editor:** The "settings.gradle" file is open. The code is as follows:

```
1 include ':app'
2
3 def localPropertiesFile = new File(rootProject.projectDir, "local.properties")
4 def properties = new Properties()
5
6 assert localPropertiesFile.exists()
7 localPropertiesFile.withReader("UTF-8") { reader -> properties.load(reader) }
8
9 def flutterSdkPath = properties.getProperty("flutter.sdk")
10 assert flutterSdkPath != null, "flutter.sdk not set in local.properties"
11 apply from: "$flutterSdkPath/packages/flutter_tools/gradle/app_plugin_loader.gradle"
```

- Bottom Status Bar:** utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$
- Bottom Taskbar:** Shows icons for various applications including a terminal, browser, and file manager.

Fig 153.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT". It includes subfolders like ui, test, web (with Icons, manifest.json, windows), .flutter-plugins, .flutter-plugins-dependencies, .gitignore, .metadata, .packages, analysis_options.yaml, pubspec.lock, pubspec.yaml, raychat.iml, and README.md.
- Editor:** The "index.html" file is open. The code is as follows:

```
<!DOCTYPE html>
<html>
<head>
<!--
If you are serving your web app in a path other than the root, change the
href value below to reflect the base path you are serving from.

The path provided below has to start and end with a slash "/" in order for
it to work correctly.

For more details:
* https://developer.mozilla.org/en-US/docs/Web/HTML/Element/base

This is a placeholder for base href that will be replaced by the value of
the '--base-href' argument provided to 'flutter build'.
-->
<base href="$FLUTTER_BASE_HREF">

<meta charset="UTF-8">
<meta content="IE=Edge" http-equiv="X-UA-Compatible">
<meta name="description" content="A new Flutter project.">

<!-- iOS meta tags & icons -->
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="apple-mobile-web-app-title" content="raychat">
<link rel="apple-touch-icon" href="icons/icon-192.png">

<!-- Favicon -->
<link rel="icon" type="image/png" href="favicon.png"/>
```

- Bottom Status Bar:** utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$
- Bottom Taskbar:** Shows icons for various applications including a terminal, browser, and file manager.

Fig 154.

```
May 4 16:26
Index.html - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER
RAYCHAT
  ui
    login.dart
    message_list.dart
    message_widget.dart
    generated_plugin_registrant.dart
    main.dart
  test
  web
    icons
      Icon-192.png
      Icon-512.png
      Icon-maskable-192.png
      Icon-maskable-512.png
    favicon.png
  index.html
  manifest.json
  windows
  .flutter-plugins
  .flutter-plugins-dependencies
  .gitignore
  .metadata
  .packages
  analysis_options.yaml
  pubspec.lock
  pubspec.yaml
  raychat.iml
  README.md

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL
utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat$
```

The screenshot shows the Visual Studio Code interface with the 'index.html' file open in the editor. The code handles service worker registration and loading Dart scripts. It includes logic for service workers, state changes, and updates. The terminal at the bottom shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$'.

Fig 155.

```
May 4 16:26
Index.html - raychat - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER
RAYCHAT
  ui
    login.dart
    message_list.dart
    message_widget.dart
    generated_plugin_registrant.dart
    main.dart
  test
  web
    icons
      Icon-192.png
      Icon-512.png
      Icon-maskable-192.png
      Icon-maskable-512.png
    favicon.png
  index.html
  manifest.json
  windows
  .flutter-plugins
  .flutter-plugins-dependencies
  .gitignore
  .metadata
  .packages
  analysis_options.yaml
  pubspec.lock
  pubspec.yaml
  raychat.iml
  README.md

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL
utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat$
```

The screenshot shows the Visual Studio Code interface with the 'index.html' file open in the editor. The code handles service worker registration and loading Dart scripts. It includes logic for service workers, state changes, and updates. The terminal at the bottom shows the command 'utkarsh@utkarsh-nitro:~/AndroidStudioProjects/raychat\$'.

Fig 156.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Activities > Visual Studio Code ▾ May 4 16:26 manifest.json - raychat - Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Explorer:** Shows the project structure for "RAYCHAT".
 - ui**: login.dart, message_list.dart, message_widget.dart, generated_plugin_registrant.dart, main.dart.
 - test**
 - web**: icons (Icon-192.png, Icon-512.png, Icon-maskable-192.png, Icon-maskable-512.png), favicon.png, index.html.
 - manifest.json** (selected)
 - windows**
 - Flutter-plugins**
 - flutter-plugins-dependencies**
 - .gitignore**
 - .metadata**
 - .packages**
 - analysis_options.yaml**
 - pubspec.lock**
 - pubspec.yaml**
 - raychat.lint**
 - README.md**
- Terminal:** bash + x
- Bottom Bar:** PROBLEMS (13) OUTPUT DEBUG CONSOLE TERMINAL
- Bottom Icons:** Finder, App Store, Mail, Home, Spotify, Microsoft Teams, Slack, Firefox, Chrome, Vivaldi, Camera.

Fig 157.

6. RESULT AND CONCLUSION

The app focuses on providing users with the basic idea of cooking ingredients. While also providing various users to communicate with each other and discuss about recipes. It also provides a list to maintain grocery items for the user. The app frontend is implemented using various widgets of flutter and for the backend it uses Firestore & Firebase.

As one can observe that the app is very much ready for daily usage incorporates and families. few features that can be added to make it more productive are:

- 1.** Maps Integration to allow user to see which shop is nearby.
- 2.** Audio call/ video call between people who are sharing a dish.
- 3.** Export the dish final image as PNG/JPEG file.
- 4.** Setting a teacher mode so that a professional chef can guide us properly how to cook.
- 5.** Column for Invention of different kinds of recipes.

7. REFERENCES

1. <https://flutter.dev/>
2. <https://dart.dev/>
3. <https://firebase.google.com/>
4. <https://www.cookinglight.com/cooking-101/12-cooking-skills-every-young-adult-should-learn>
5. <https://retrohousewifegoesgreen.com/basic-cooking-skills/>
6. <https://www.reportlinker.com/insight/americans-cooking-habits.html>
7. Asynchronous Programming google: <https://dart.dev/codelabs/async-await>
8. State Management google: <https://docs.flutter.dev/development/data-and-backend/state-mgmt/intro>
9. Networking google: <https://docs.flutter.dev/development/data-and-backend/networking>
10. J son and Serialization: <https://docs.flutter.dev/development/data-and-backend/json>
11. Firebase google: <https://docs.flutter.dev/development/data-and-backend/firebase>
12. Interactivity: <https://docs.flutter.dev/development/ui/interactive>
13. Assets and Images google Flutter: <https://docs.flutter.dev/development/ui/assets-and-images>
14. Flutter Focus System flutter: <https://docs.flutter.dev/development/ui/advanced/focus>
15. Using Actions and Shortcuts flutter:
https://docs.flutter.dev/development/ui/advanced/actions_and_shortcuts
16. Adding a splash screen to your mobile app flutter:
<https://docs.flutter.dev/development/ui/advanced/splash-screen>
17. Google fonts pub: https://pub.dev/packages/google_fonts/changelog
18. Flutter: Login App using REST API and SQF Lite_ Kashif Minhaj:
<https://medium.com/@kashifmin/flutter-login-app-using-rest-api-and-sqlite-b4815aed2149>
19. Create Your First Flutter App and Learn Flutter Framework Mahmud Ahsan:
<https://thinkdiff.net/getting-started-with-flutter-create-your-first-flutter-app-f6dea473f57d>
20. Firebase use cases google: <https://firebase.google.com/use-cases>
21. Firebase Auth google: <https://firebase.google.com/docs/auth/flutter/start>
22. Structuring database firebase: <https://firebase.google.com/docs/database/flutter/structure-data>
23. Read and Write Data firebase: <https://firebase.google.com/docs/database/flutter/read-and-write>
24. <https://firebase.google.com/docs/database/flutter/lists-of-data>
25. Enabling offline capabilities: <https://firebase.google.com/docs/database/flutter/offline-capabilities>

- 26.** Managing User Firebase: <https://firebase.google.com/docs/auth/flutter/manage-users>
- 27.** Cloud Firestore google: <https://firebase.flutter.dev/docs/firestore/overview/>
- 28.** Authentication google: <https://firebase.flutter.dev/docs/auth/overview/>
- 29.** Connection of App with Firebase google: <https://firebase.flutter.dev/docs/auth/start/>
- 30.** Manage Users in Firebase google: <https://firebase.flutter.dev/docs/auth/manage-users/>
- 31.** Authenticate with Firebase using Password-Based Accounts on Flutter google:
<https://firebase.flutter.dev/docs/auth/password-auth/>
- 32.** Social Authentication google: <https://firebase.flutter.dev/docs/auth/social/>
- 33.** Authenticate with Firebase Using a Custom Authentication System google:
<https://firebase.flutter.dev/docs/auth/custom-auth/>
- 34.** Cloud Firestore google: <https://firebase.flutter.dev/docs/firestore/overview/>
- 35.** Firebase Cloud Messaging google: <https://firebase.flutter.dev/docs/messaging/overview/>
- 36.** FCM via APNs Integration google: <https://firebase.flutter.dev/docs/messaging/apple-integration/>
- 37.** Permission google: <https://firebase.flutter.dev/docs/messaging/permissions/>
- 38.** Notification google: <https://firebase.flutter.dev/docs/messaging/notifications/>