

ONE CLICK EAT

INTERDISCIPLINARY PROJECT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

by

AFIYA M (41120007)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY-1 UNIVERSITY BY UGC

Accredited "A++" by NAAC

12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

[DEEMED TO BE UNIVERSITY]

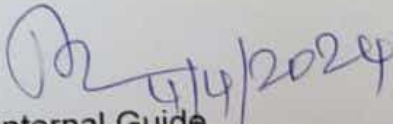
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

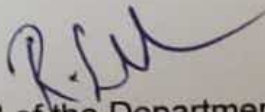
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AFIYA M (41120007)** who carried out the project entitled "**ONE CLICK EAT**" under my supervision from **January 2024** to **May 2024**.


Internal Guide

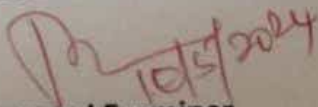
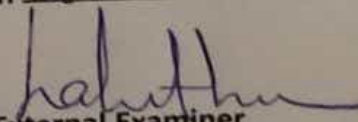
Dr. P. JEYANTHI, M.E., Ph.D.,



Head of the Department

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

Submitted for Viva voce Examination held on 08/05/2024

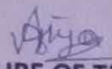

Internal Examiner
External Examiner

DECLARATION

I Afiya M hereby declare that the Project Report entitled "ONE CLICK EAT" done by me under the guidance of **Dr.P.JEYANTHI.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 08/05/2024

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

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

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

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

CERTIFICATE OF COMPLETION



HACKWIT TECHNOLOGIES PVT LTD
Redefine by Technologies

149, 1C, 2nd Floor, Mount Poonamallee Rd,
Ramapuram, Chennai - 600089.
admin@hackwittechnologies.com
www.hackwittechnologies.com

10.02.2024

CERTIFICATE FOR INTERNSHIP

TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Afiya M (Reg No: 41120007)** student of Sathyabama Institute of Science and Technology, B. Tech Information Technology, has successfully completed the Internship at Hackwit Technologies from 22nd January to 30th January 2024.

During the internship, **Afiya M** has closely worked as a part of the **Flutter** by carrying out the project entitled **"Food Delivery App"**.

She demonstrated good developing and learning skill with self-motivated attitude to learn new things. Her contribution to implement the Flutter was good. We wish her all the best for her future endeavors.

Best Regards,

Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd



CERTIFICATE OF COMPLETION



ABSTRACT

This report outlines the development and assessment of a Flutter-based interface for the ONE CLICK EAT food app. Its goal is to provide users with a seamless experience for ordering food for delivery or pickup. The app focuses on affordability and offers features like menu browsing, order placement, and the unique ability to sample food before ordering.

Key features include intuitive navigation, visually appealing layouts, and smooth interactions. The UI integrates order placement, car facilities booking, and food sampling to enhance user convenience. Usability testing and user feedback drove iterative improvements to ensure alignment with user preferences.

In summary, the report highlights the importance of a well-designed Flutter UI in improving the user experience of food delivery services. ONE CLICK EAT showcases how Flutter technology can create compelling mobile app experiences in the food industry.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	vii
	LIST OF ABBREVIATIONS	x
	LIST OF FIGURES	xi
1	INTRODUCTION	1
	1.1 Flutter	2
	1.2 Dart	3
	1.3 Firebase	4
	1.4 Problem Statement	5
	1.5 Objectives	7
2	METHODOLOGY	9
	2.1 Initial Project Requirements	9
	2.2 Proposed System Architecture	12
	2.3 Flow Diagram	16
	2.4 Modules	17
	2.4.1 User Management	17
	2.4.2 Authentication Management	18
	2.4.3 Profile Management	19
	2.4.4 Order Management	20
	2.4.5 Rating & Car Tracking	21
	2.5 Development and Deployment Setup	22

3	RESULTS AND DISCUSSION	26
4	CONCLUSION AND FUTURE WORK	32
4.1	Conclusion	32
4.2	Future Work	33
	REFERENCES	35
	APPENDICES	39
A.	Sample Code	39
B.	Screen Shots	41

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
UI	- User Interface
UX	- User Experience
iOS	- iPhone Operating System
UAT	- User Acceptance Testing
IDE	- Integrated Development Environment

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
2.1	Proposed Application Architecture	12
2.2	Firebase Architecture	14
2.2	Flow Diagram	16
2.3	Project Creation	22
2.4	Firebase Connection	23
2.5	Firebase Authentication	24
2.6	Flutter Run	24
3.1	Login Page Interface	26
3.2	Profile Screen Display	27
3.3	Order Details Interface	28
3.4	Feedback Submission Screen	29
3.5	Firebase Configurations	30
3.6	Initialize Firebase	31

CHAPTER 1

INTRODUCTION

In today's fast-paced world, "One Click Eat" revolutionizes how individuals enjoy their meals. Rooted in accessibility and inclusivity, this app seamlessly connects users with their favorite cuisines while prioritizing affordability, satisfaction, and convenience.

At its core, "One Click Eat" streamlines the dining experience by offering a diverse array of high-quality dishes from various cuisines. With features like convenient booking, secure payments, and real-time order tracking, the app ensures a hassle-free process from start to finish.

Moreover, "One Click Eat" fosters community and connection among food enthusiasts. Through its interactive platform, users can share feedback, discover new culinary experiences, and engage with like-minded individuals.

Leveraging cutting-edge technology, "One Click Eat" delivers a seamless and enriching user experience. From browsing menus to tracking deliveries, every aspect is designed to enhance convenience and satisfaction.

In conclusion, "One Click Eat" sets a new standard for food delivery apps, offering convenience, variety, and community in a single tap. Embark on a culinary journey like never before with "One Click Eat."

1.1.FLUTTER

Flutter, the robust cross-platform UI toolkit developed by Google, serves as the backbone of "One Click Eat," shaping both its foundation and user interface. By harnessing the power of Flutter, "One Click Eat" transcends traditional app development limitations, offering a single codebase that seamlessly adapts to both iOS and Android environments.

The utilization of Flutter brings about a paradigm shift in app development, simplifying the process by eliminating the need to manage separate codebases for different platforms. This efficiency not only accelerates development cycles but also enhances productivity for developers. With Flutter's hot reload feature, real-time code changes and instant visual feedback enable swift iterations and adjustments, ensuring a smooth development workflow.

One of the standout features of Flutter is its ability to maintain a consistent and high-quality user experience across various devices. By leveraging Flutter's customizable widgets, "One Click Eat" achieves a cohesive design language and fluid interactions, regardless of the device's operating system. Users can seamlessly transition between devices while enjoying the same level of performance and aesthetics, ensuring a delightful dining experience every time they use the app.

1.2.DART

i) Unified Codebase:

"One Click Eat" relies on Dart, a versatile programming language, for both frontend (Flutter) and backend operations. Dart's flexibility allows developers to maintain a unified codebase, ensuring consistency and facilitating collaboration between different parts of the app.

ii) Asynchronous Programming:

Dart's support for asynchronous programming enables "One Click Eat" to handle concurrent tasks efficiently. This feature ensures smooth management of user requests, order processing, and external API interactions without compromising the app's responsiveness.

iii) Scalability:

Dart's scalability is crucial for accommodating the potential growth of "One Click Eat." As the user base expands, Dart allows the app to handle increased traffic and demand for food delivery services, ensuring a seamless experience for users even during peak periods.

iv) Strong Typing System:

In "One Click Eat," Dart's strong typing system enhances code reliability and maintainability. By enforcing strict type checks, Dart helps developers identify and fix errors early in the development process, resulting in a more stable and robust application.

1.3.FIREBASE

1)Real-time Order Updates:

Firebase Cloud Firestore enables instant updates on order status changes and delivery tracking, ensuring customers receive real-time information on their orders.

2)Secure User Authentication:

Firebase Authentication ensures only authorized users can access the One Click Eat platform, providing a secure login process with options for multi-factor authentication.

3)Efficient Backend Operations:

Utilizing Firebase's serverless cloud functions, One Click Eat can efficiently manage order processing, notifications, and other backend tasks, ensuring scalability without managing servers.

4)Media Storage and Delivery:

Firebase Cloud Storage efficiently manages and serves menu images, restaurant logos, and user profile pictures, enhancing the app's visual appeal and user experience.

1.4.PROBLEM STATEMENT

In the realm of modern dining, consumers face an array of challenges when it comes to accessing their favorite foods. Traditional food delivery services often lack the integration necessary to provide a seamless and satisfying experience for users. These challenges stem from the disjointed nature of existing platforms, which fail to offer a unified solution that caters to the diverse needs and preferences of consumers.

One of the primary obstacles is the fragmentation of available resources across different platforms. Users must navigate separate interfaces for food ordering, delivery coordination, and dine-in reservations, leading to a disjointed and cumbersome experience. This lack of integration not only complicates the user journey but also diminishes efficiency and convenience.

Moreover, existing platforms often struggle to accommodate the diverse needs of consumers. Many users encounter difficulties in finding options that align with their dietary preferences, accessibility requirements, and cultural considerations. This lack of inclusivity results in missed opportunities for users to explore new culinary experiences and inhibits the platform's ability to cater to a broad audience effectively.

Additionally, while traditional food delivery services offer the convenience of enjoying meals from the comfort of home, they often lack opportunities for users to sample new dishes or dine-in experiences. This limitation prevents users from fully engaging with the culinary offerings of their favorite restaurants and inhibits the platform's ability to foster meaningful connections .

The core issue lies in the absence of a comprehensive platform that seamlessly integrates food ordering, delivery logistics, dine-in reservations, and sampling opportunities. Kolapasii seeks to address these challenges by providing an all-encompassing solution that caters to the diverse needs and preferences of consumers. By offering a unified platform that streamlines the food service experience, Kolapasii aims to revolutionize the way consumers interact with food delivery services, empowering them to indulge in satisfying culinary experiences with ease and convenience.

1.5. OBJECTIVES

The primary objective of One Click Eat is to establish a unified and user-centric platform that seamlessly integrates food ordering, delivery logistics, dine-in reservations, and sampling opportunities. The app aims to address the fragmented nature of existing food delivery services by providing a comprehensive solution that caters to the diverse needs and preferences of users. One Click Eat seeks to redefine the food service experience by offering convenience, diversity, and satisfaction in every interaction. The platform strives to create a vibrant community where users can explore new culinary experiences, connect with their favorite restaurants, and indulge in satisfying meals with ease and confidence. Ultimately, One Click Eat aims to empower users to enjoy seamless and delightful culinary experiences at their fingertips, ensuring a fulfilling dining journey for all.

Key Challenges:

i) Integration of Diverse Services:

Integrating various functionalities such as food ordering, delivery logistics, dine-in reservations, and transportation booking seamlessly within the app. Ensuring a smooth and intuitive user experience while navigating through different features and services.

ii) Inclusive User Experience:

Designing a user interface that caters to users with diverse preferences, dietary restrictions, and accessibility needs. Balancing inclusivity with simplicity and clarity in the app's design to ensure that all users can easily navigate and use the platform.

iii) Vendor Engagement and Management:

Ensuring active participation and cooperation from partner restaurants and vendors in maintaining menu accuracy, timely deliveries, and quality standards. Implementing effective vendor management strategies to foster positive relationships and ensure customer satisfaction.

iv) Safety and Trust:

Implementing robust safety measures to protect user health and ensure food hygiene standards are met during preparation and delivery. Building trust among users by transparently communicating safety protocols and standards followed by partner restaurants.

v) Scalability and Reliability:

Designing the app infrastructure to handle fluctuations in user demand and scale operations seamlessly during peak hours. Ensuring reliable performance and uptime to meet user expectations for timely food delivery and service availability.

vi) Customer Support and Issue Resolution:

Providing efficient customer support to address user inquiries, resolve order-related issues, and handle feedback promptly. Implementing effective communication channels and support systems to ensure a positive user experience and build customer loyalty.

vii) Sustainability and Environmental Impact:

Addressing environmental concerns related to packaging waste, carbon emissions from delivery vehicles, and sustainable sourcing practices.

CHAPTER 2

METHODOLOGY

2.1. INITIAL PROJECT REQUIREMENTS

The initial project requirements for ONE CLICK EAT encompass various aspects such as functionalities, technologies, and packages.

i)Project Scope:

- 1)Develop a mobile application called "One Click Eat" for iOS and Android platforms.
- 2)Create a comprehensive platform that facilitates food ordering, delivery coordination, dine-in reservations, and transportation booking.
- 3)Connect users with a wide range of restaurants and eateries to satisfy diverse culinary preferences.
- 4)Provide seamless integration of various services, including menu browsing, order customization, payment processing, delivery tracking, and feedback submission.

ii)Technologies:

- 1) Flutter: Utilize Flutter for cross-platform mobile app development, ensuring a consistent user experience on both iOS and Android.
- 2) Dart: Employ the Dart programming language as the primary language for Flutter app development.
- 3) Firebase: Integrate Firebase for backend services, including authentication, real-time database (Firestore), and clou storage.

iii)Functionalities:

1)User Authentication: Implement a secure authentication system for users to create accounts, log in, and access personalized features.

2)Food Ordering: Enable users to add items to their cart, customize orders, and place food orders seamlessly within the app.

3)Order Tracking: Provide real-time tracking of order status, including preparation, delivery, and estimated arrival time.

Transportation Booking: Integrate transportation booking services to enable users to book rides to restaurants for dine-in experiences.

4)Payment Processing: Implement secure payment processing for seamless transactions within the app.

iv)Packages Used:

i)get:

Version: 4.6.6

Purpose: State management and navigation framework

ii)firebase_core:

Version: 2.25.3

Purpose: Firebase initialization and core functionality.

iii)firebase_auth:

Version: 4.17.3

Purpose: User authentication with Firebase.

iv)cloud_firestore:

Version: 4.15.3

Purpose: Interaction with Cloud Firestore for real-time data.

v)cloud_firestore_web:

Version: 3.10.3

Purpose: Web implementation of Cloud Firestore.

vi)firebase_auth_web:

Version: 5.9.3

Purpose: Web implementation of Firebase Authentication.

vii)firebase_core_web:

Version: 2.11.3

Purpose: Web implementation of Firebase Core.

viii)google_sign_in:

Version: 5.2.1

Purpose: Google Sign-In authentication.

ix)toast:

Version: 0.3.0

Purpose: Displaying simple toast notifications.

x)fluttertoast:

Version: 8.0.7

Purpose: Displaying customizable toast messages.

xi)image_picker:

Version: 1.0.7

Purpose: Picking images from the gallery or taking photos.

2.2.PROPOSED APPLICATION ARCHITECTURE

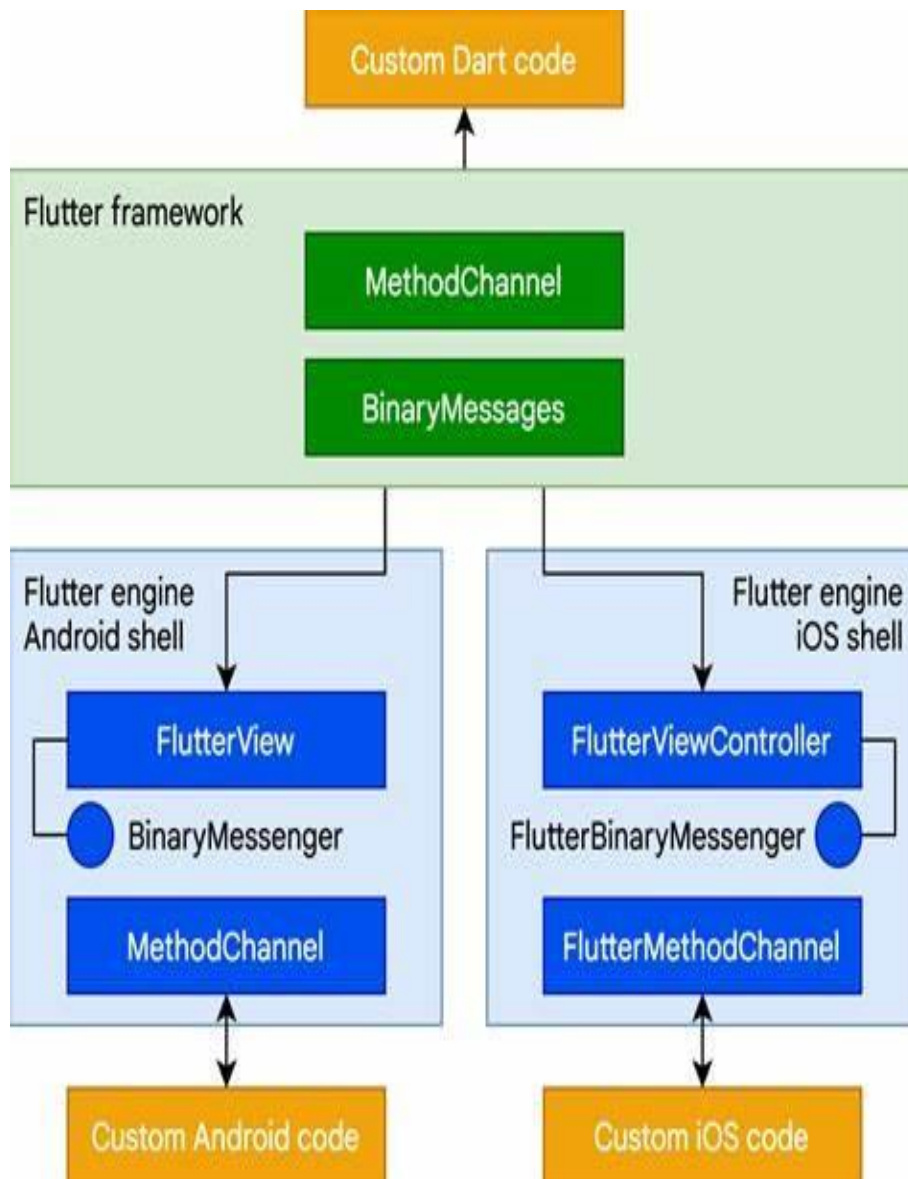


Fig:2.1: Proposed Application Architecture

1.Custom Dart Code:

This represents any custom code you write in Dart specifically for your Flutter app. It could be business logic, UI components, or any other functionality unique to your application. The custom Dart code interacts with the Flutter framework to achieve various tasks.

2.Flutter Framework:

The Flutter framework is the heart of your app. It provides a rich set of widgets, tools, and APIs for building cross-platform mobile applications.

3.MethodChannel:

The MethodChannel allows communication between Dart code and native platform code (Android or iOS). When we invoke a method on the MethodChannel from Dart, it triggers a corresponding method call on the native side.

4.BinaryMessages:

BinaryMessages is another channel for communication between Dart and native code. It enables the exchange of binary data (such as images, files, or serialized objects) between the two sides.

5.Flutter Engine Android Shell:

On the Android side, the Flutter engine runs within an Android shell.

i)FlutterView: This is the Android view that displays your Flutter UI.

ii)BinaryMessenger: It handles communication between the Flutter engine and the Android platform.

6.Flutter Engine iOS Shell:

On iOS, the Flutter engine runs within an iOS shell.

i)FlutterViewController: The iOS view controller responsible for displaying your Flutter UI.

ii)FlutterBinaryMessenger: Handles communication between the Flutter engine and the iOS platform.

7.Custom Android Code and Custom iOS Code:

These represent platform-specific code written in Java/Kotlin (for Android) and Objective-C/Swift (for iOS).You'll write custom code here to interact with native features or APIs that aren't directly accessible from Flutter.

8.Firebase:

Firebase

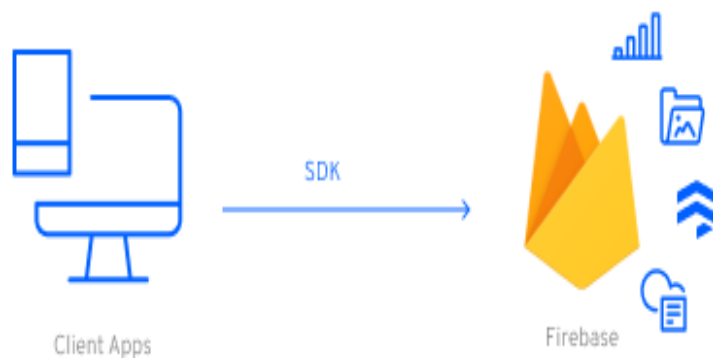


Fig:2.2: Firebase Architecture

i)Firebase Authentication:

Firebase Authentication offers ready-to-use authentication services. It integrates seamlessly with Flutter, allowing users to sign up, sign in, and manage their accounts. MFA (Multi-factor Authentication) can be enabled to enhance security by requiring additional verification steps (e.g., SMS codes, email, or biometrics).

ii)Firestore (Realtime Database):

Firestore is a NoSQL database provided by Firebase. It stores structured data in collections and documents. You can use Firestore to store user profiles, chat messages, app settings, and more.

iii)Firebase Storage:

Firebase Storage provides secure cloud storage for files (e.g., images, videos, documents). You can upload and download files directly from your Flutter app. Data-at-rest encryption ensures sensitive files are protected.

iv)Firebase Analytics:

Firebase Analytics tracks user behavior, app usage, and events. You can gain insights into user engagement, retention, and conversion rates. Use analytics data to make informed decisions about app improvements.

9.To integrate Firebase with your Flutter app:

Add the necessary Firebase plugins (e.g., `firebase_core`, `firebase_auth`, `cloud_firestore`, etc.) to your `pubspec.yaml`. Initialize Firebase in your app (usually in the main function). Use Flutter widgets to build UI that interact with Firebase services.

2.3.FLOW DIAGRAM

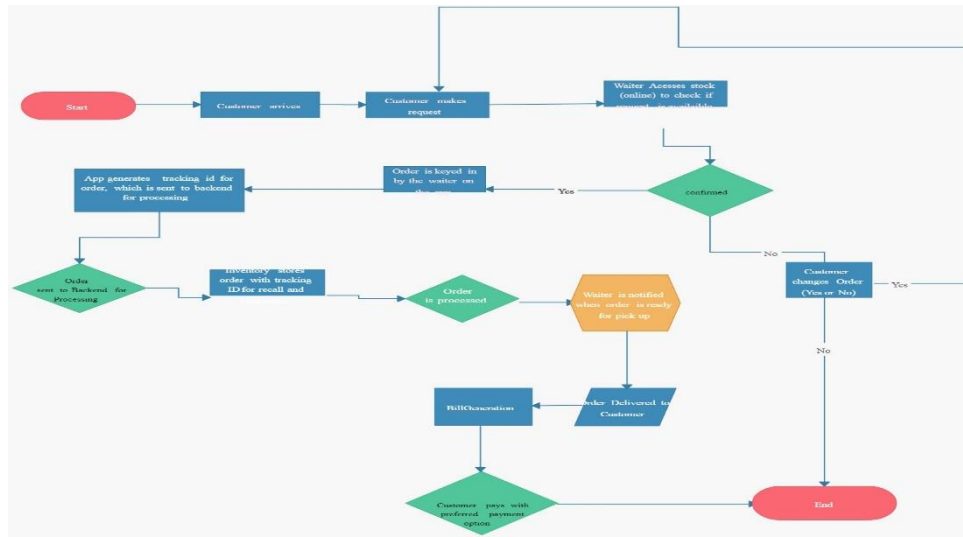


Fig:2.3: Flow Diagram

- i) The user opens the One Click Eat mobile app.
- ii) The app prompts the user to log in or sign up if they don't have an account yet.
- iii) If the user chooses to sign up, they will be taken to the sign-up page to create a new account.
- iv) If the user is already registered, they can log in directly using their credentials.
- v) Once logged in, the user is directed to the home page of the app.
- vi) From the home page, the user can access various features such as viewing menus, and exploring food options.
- viii) The cart page allows the user to review their selected items, customize orders, and proceed to checkout.
- ix) During checkout, the user can choose delivery or dine-in options, provide delivery details, and select payment methods.
- x) After completing the order, the user receives confirmation and can track the status of their order in real-time

2.4.MODULES

2.4.1.USER MANAGEMENT:

Provides a comprehensive registration form(signup) with fields for personal information, contact details, and interests, enhancing user profile completeness.

Algorithm:

- i) Display Registration Form:The signup process commences by presenting a user-friendly registration (Signup) form within the One Click Eat app. This form is designed to be intuitive and visually appealing, guiding users through the signup process seamlessly. It includes fields for collecting personal information, contact details, and any additional preferences relevant to the food delivery experience.
- ii) Collect User Information:Upon accessing the registration form, users are prompted to provide essential information required to create their account. This information typically includes their full name, email address, password, date of birth, and any dietary preferences or food allergies they may have. Collecting these details ensures that each user's profile is comprehensive and tailored to their needs.
- iii) Validate Input Data:As users enter their information into the registration form, the system performs validation checks in real-time to ensure the accuracy and completeness of the data. For instance, it verifies that the email address follows the correct format and that the password meets security requirements, such as minimum length and inclusion of special characters. Any invalid or incomplete entries prompt the user to correct them before proceeding.

2.4.2.AUTHENTICATION:

Implements secure login functionality with email/password or social login options, providing users with seamless access to the platform.

Algorithm:

i) Display Login Interface:Users are presented with a login interface upon opening the One Click Eat app, where they can input their credentials.

ii) Verify User Credentials:

The system checks the entered credentials against stored data in the app's database or validates authentication tokens for social login.

iii) Grant Access:If the credentials are valid, users are granted access to the One Click Eat platform to explore its features and functionalities.

2.4.3.PROFILE MANAGEMENT:

Enables users to update profiles, upload pictures, set availability, and manage notifications, enhancing personalization and user control.

Algorithm:

i) Access Profile Settings:Users can access their profile settings through a dedicated interface within the One Click Eat app. This interface provides users with centralized control over their profile information and preferences, ensuring a seamless and personalized experience.

ii) Update Profile Information:The profile management functionality enables users to update their profile details directly from the app. Users can modify information such as their name, contact details, dietary preferences, delivery address, and any other relevant details that enhance their food delivery experience. This ensures that users can keep their profile information up-to-date and accurate.

iii) Grant Access:Similar to the login process, if the user's credentials or authentication tokens are verified successfully, the system grants the user access to the One Click Eat platform. Once logged in, users can access their profile settings and update their information as needed to personalize their food delivery experience.

2.4.4.ORDER MANAGEMENT:

Implements collaborative filtering algorithms considering user preferences, mentor expertise, availability, and ratings, enhancing accurate mentor-mentee matches.

Algorithm:

i) Order Data Collection:Gather order data from the database, including order details, restaurant information, delivery preferences, and user feedback.

Capture additional information such as order history, payment details, delivery addresses, and order status updates.

ii) User-Driven Order Management:Design a user-friendly interface for managing orders within the One Click Eat app.

Showcase relevant information such as order details, restaurant names, delivery statuses, and estimated delivery times.

Implement features for users to track their orders in real-time, view order history, and manage order preferences such as delivery addresses and payment methods.

iii) Order Feedback:Allow users to provide feedback on their order experiences, including food quality, delivery speed, and overall satisfaction.

Enable users to rate restaurants based on their dining experiences and provide comments or suggestions for improvement.

Utilize user feedback to enhance the ordering process, improve restaurant recommendations, and optimize delivery services.

2.4.5.RATING AND CAR TRACKING:

By incorporating these functionalities, the One Click Eat app aims to enhance user satisfaction by providing a reliable rating system for feedback and improving transparency in the delivery process through real-time car tracking.

Algorithm:

i) Rating System Setup: Initialize the rating system within the app's backend using appropriate technologies.

Establish a database structure to store rating-related data, including ratings given by users, associated orders, and feedback comments.

ii) Rating Submission: Allow users to submit ratings for their food delivery experiences within the app.

Design a user-friendly interface for users to rate their orders based on factors such as food quality, delivery speed, and overall satisfaction.

iii) Car Tracking: Integrate a car tracking system within the app using GPS technology or third-party APIs.

Display real-time updates on the location of delivery vehicles to users awaiting their orders.

Implement features for users to track the progress of their orders from preparation to delivery.

Ensure that car tracking data is securely transmitted and displayed within the app's interface for users' convenience and peace of mind.

2.5.DEVELOPMENT AND DEPLOYMENT SETUP

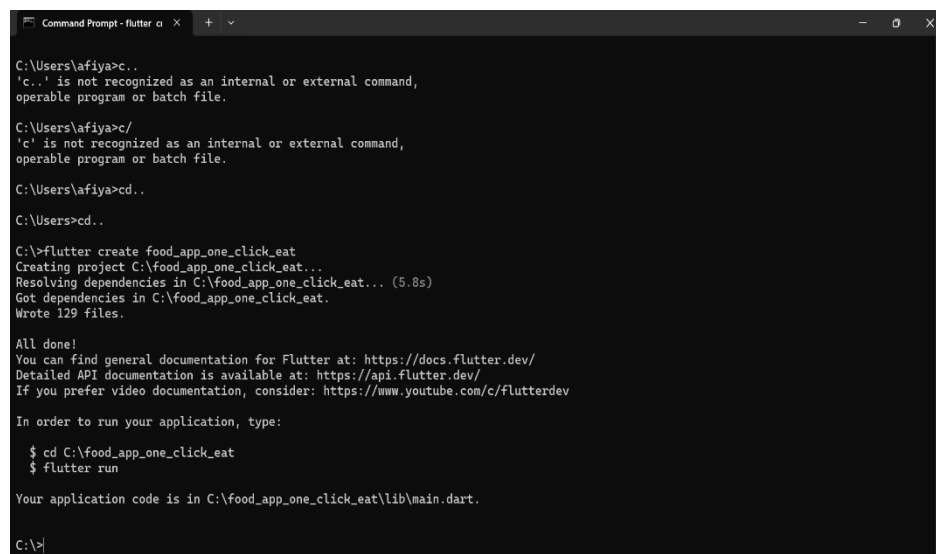
1.Development Environment:

Flutter and Dart need to be installed on the developer's machine.

IDEs like Visual Studio Code or Android Studio for coding.

2.Create Project:

>>flutter create project_name



```
Command Prompt - flutter a X + v
C:\Users\afiya>c..
'c..' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\afiya>c/
'c/' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\afiya>cd..
C:\Users>cd..

C:\>flutter create food_app_one_click_eat
Creating project C:\food_app_one_click_eat...
Resolving dependencies in C:\food_app_one_click_eat... (5.8s)
Got dependencies in C:\food_app_one_click_eat.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

$ cd C:\food_app_one_click_eat
$ flutter run

Your application code is in C:\food_app_one_click_eat\lib\main.dart.

C:\>|
```

Fig:2.4: Project Creation

3.Dependency Management:

Project dependencies specified in the pubspec.yaml file.

Key packages include:

- firebase_core for Firebase integration.
- firebase_auth for user authentication.
- cloud_firestore for real-time data synchronization.
- Other packages like get, toast, image_picker, etc., for additional functionalities.

4.Firebase Configuration:

Firebase project created and configured for the application.

Firebase Hosting used for deploying the Flutter web application.

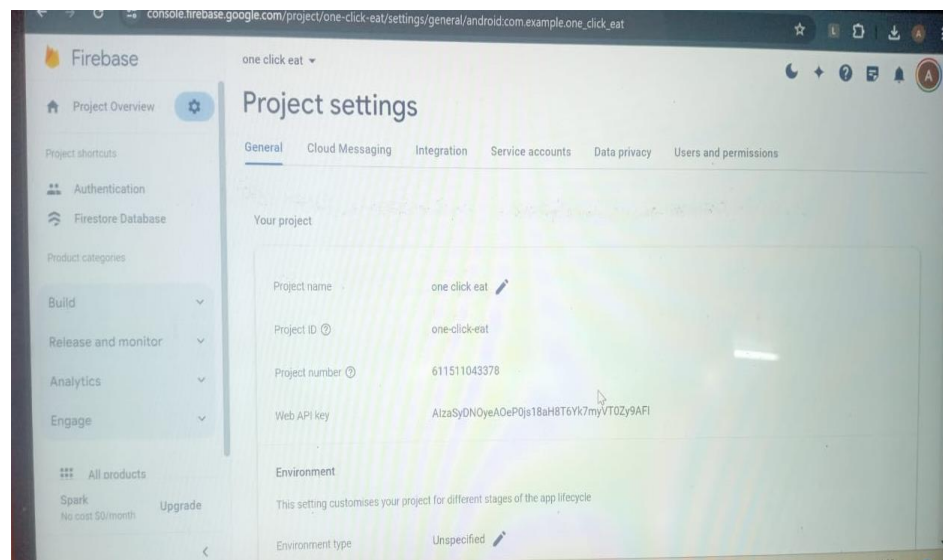


Fig:2.5: Firebase Connection

5. Firebase Authentication:

Firebase Authentication set up for secure user login and identity verification.

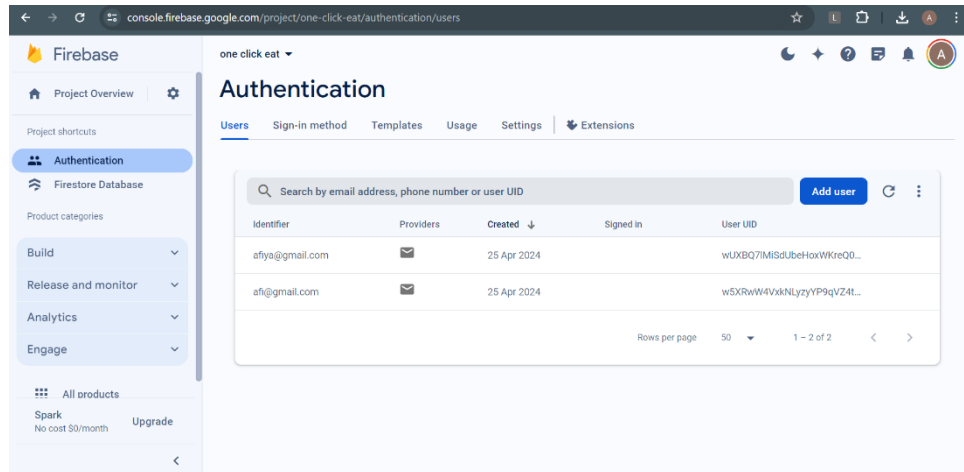


Fig:2.6: Firebase Authentication

6. Development Tools:

Flutter commands used for testing and building the application. Real-time testing on Android and iOS devices.

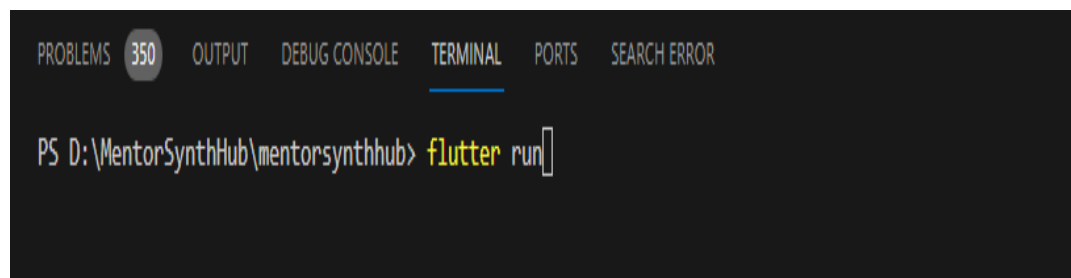


Fig:2.7: Flutter Run

7.Deployment Process:

Firebase used for hosting the web version of the application.
Seamless deployment using Flutter and Firebase commands.

8.Supported Platforms:

Cross-platform support for Android, iOS, and web.

9.Integrated Development Environment (IDE):

Developers can choose between Visual Studio Code or Android Studio for coding.

10.User Authentication:

Firebase Authentication ensures secure user login.

CHAPTER 3

RESULTS AND DISCUSSION

DART CODES:

i)main.dart: This file is typically the entry point of a Flutter application. It initializes the app, sets up necessary configurations, and runs the main application widget.

ii)loginPage.dart: This file likely contains the UI and logic for user login. It includes form fields for entering credentials and a mechanism to authenticate users.



Fig:3.1: Login Page Interface

iii)signupPage.dart: Similar to the login file, signupPage.dart is likely responsible for handling user registration. It includes form fields for users to create accounts.

iv)homePage.dart: This file represents the main content of your app after a successful login or signup. It contains order details,profilePage ,feedbackPage and Home page Page with various features.

v)profilePage.dart: The file is likely responsible for displaying and editing user profiles. It includes information about the user, such as their name, profile picture, and additional details.

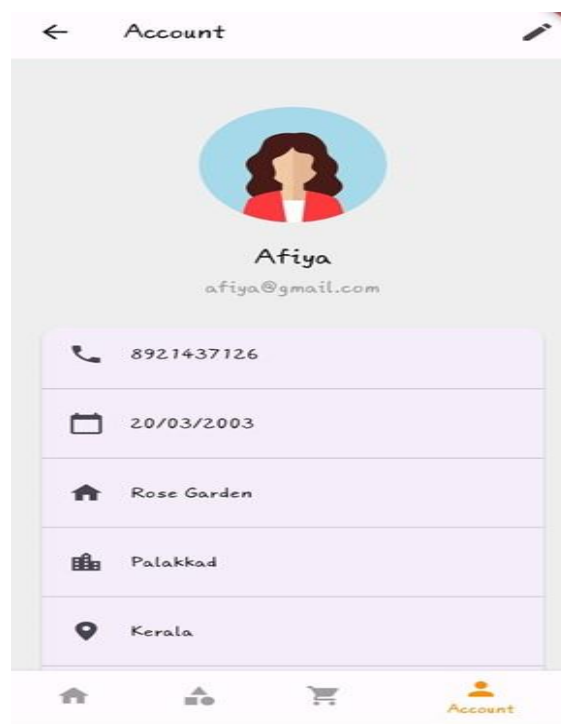


Fig:3.2: Profile Screen Display

vi)items.dart: This file could represent a page where users can search food items.

vii)order.dart: This file defines the structure of a order model.

viii)onboard.dart: This file represents an onboarding screen.

ix)forget.dart: This file represent the forgot password screen.

x)welcome.dart: This file defines a welcome screen.

xi)navbar.dart: This file defines the structure of a navbar.

xii)order_items.dart: This file represents a page where user can add items to cart.

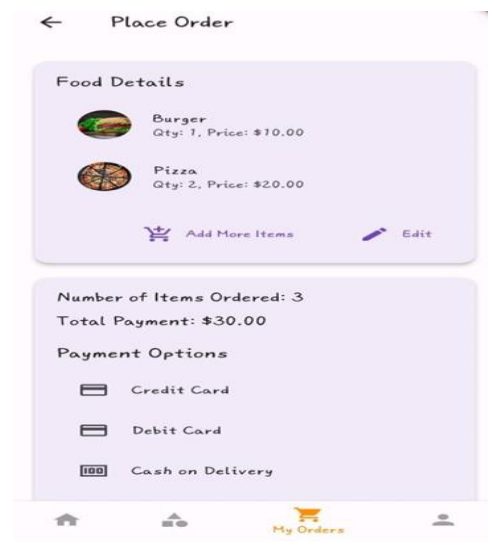


Fig:3.3: order details Interface

xiii) rating.dart: This file represents a rating details.

xiv) profile_edit.dart: This file creates a profile editing option.

xv) feedBackPage.dart: This file represents a feedback page where users can provide feedback.

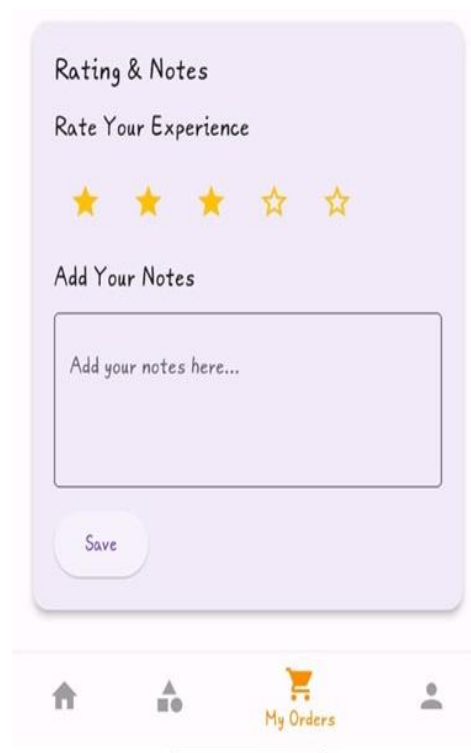


Fig:3.4: Feedback Submission Screen

FIREBASE:

i)Create a Firebase Project:

1. Go to the Firebase Console.
2. Click on "Add Project" and follow the setup instructions.

ii)Configure Flutter Project:

1. Add the Firebase configuration files to your Flutter project.
2. Follow the instructions provided in the Firebase Console to add google-services.json for Android and GoogleService-Info.plist for iOS.
3. These files contain configuration details needed for your app to connect to Firebase services.

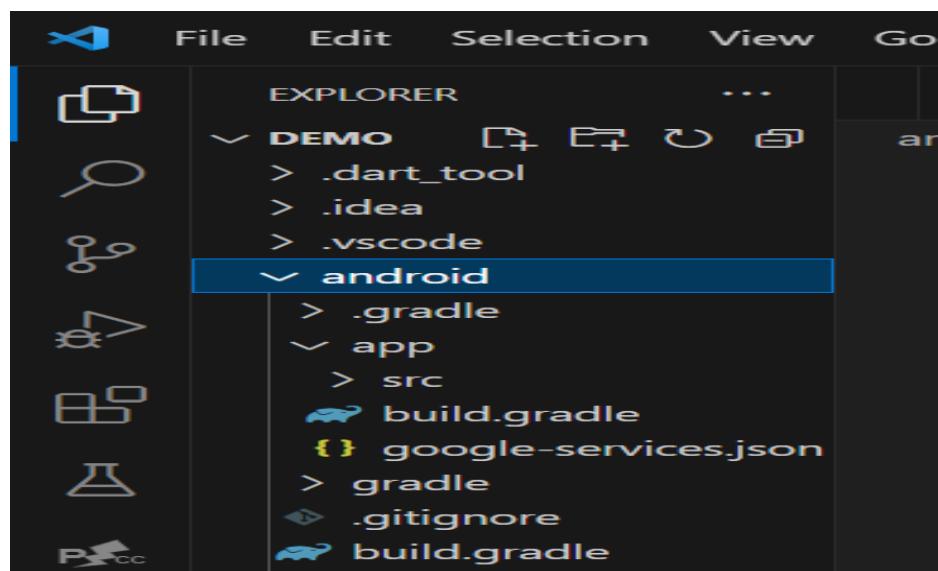


Fig:3.5: Firebase Configurations

iii)Add Firebase Dependencies:

Open the pubspec.yaml file and add the following dependencies:

yaml code:

dependencies:

firebase_core: ^2.25.3

firebase_auth: ^4.17.5

iv)Initialize Firebase:

In your main.dart or wherever you initialize your Flutter app, make sure to initialize Firebase. If you want to implement user authentication, Firebase provides the firebase_auth package. You can use it to sign in users using various methods like email/password, Google Sign-In, etc.

Dart code:

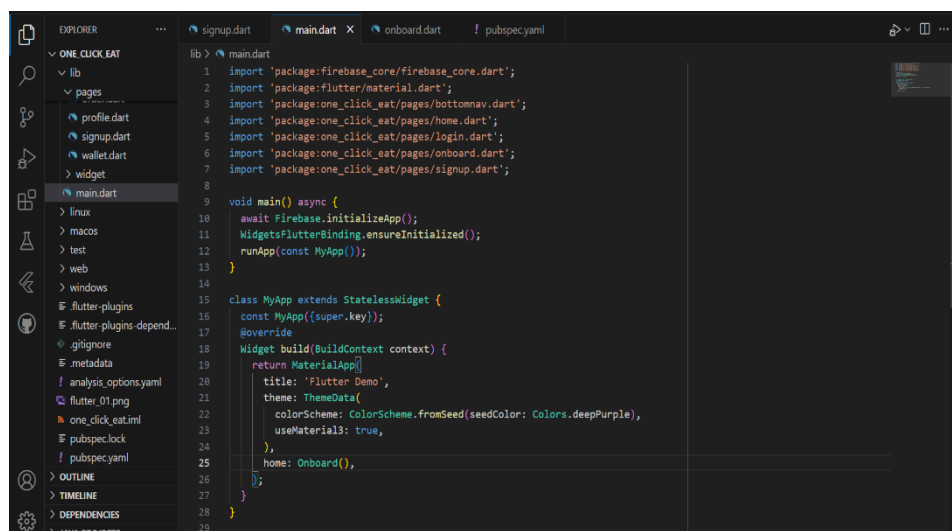


Fig:3.6: Initializing Firebase

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In summary, One Click Eat revolutionizes the food delivery experience, seamlessly integrating convenience, choice, and quality. Leveraging the versatility of Flutter and Dart, our app ensures a smooth and cohesive interface across both iOS and Android platforms, catering to a diverse user base. The incorporation of Firebase elevates the app's capabilities, guaranteeing secure user authentication and efficient data management.

One Click Eat serves as the ultimate solution for satisfying culinary cravings, offering users an extensive selection of restaurants and eateries at their fingertips. From browsing menus to tracking deliveries, the app empowers users with control and convenience throughout the ordering process.

With its emphasis on real-time messaging and file sharing, One Click Eat fosters collaboration between users and delivery drivers or restaurant representatives, ensuring seamless communication and enhancing the overall dining experience.

In essence, One Click Eat redefines food delivery by providing a comprehensive platform that caters to the needs of both users and restaurant partners. Through its user-friendly design, robust features, and commitment to excellence, the app sets a new standard for convenience and satisfaction in the realm of food delivery.

4.2. FUTURE WORK

1)Enhanced Restaurant Profiles:

Implement additional features and customization options in restaurant profiles, such as showcasing special dishes, chef profiles, and customer reviews to provide users with more insight into the dining experience.

2)Collaboration with Partner Restaurants:

Establish collaborations with partner restaurants, such as Kolapasi, to integrate their menu offerings seamlessly into the One Click Eat app. This collaboration could involve real-time updates on menu items, exclusive deals for app users, and joint marketing campaigns.

3)Real-time Order Tracking and Delivery Optimization:

Enhance the order tracking functionality to provide real-time updates on order status and delivery progress. Implement optimization algorithms to minimize delivery times and ensure efficient delivery routes for drivers.

4)Personalized Recommendations:

Develop algorithms to analyze user preferences, order history, and dietary preferences to provide personalized restaurant recommendations. This could involve machine learning techniques to improve recommendation accuracy over time.

5)Interactive Cooking Classes and Culinary Events:

Collaborate with restaurants to host interactive cooking classes, virtual culinary events, or live streaming sessions within the app. This feature could provide users with the opportunity to learn cooking techniques, explore new cuisines.

6)Integration with Food Blogs and Culinary Content:

Partner with food bloggers, culinary experts, and influencers to integrate their content directly into the One Click Eat app. This could include curated lists of recommended restaurants, food reviews, and culinary tips to enrich the user experience.

7)Social Sharing and Community Engagement:

Implement social sharing features to allow users to share their dining experiences, food photos, and restaurant recommendations with friends and followers on social media platforms. Encourage community engagement by hosting user-generated content contests and challenges.

8)Smart Kitchen Integration:

Explore integration with smart kitchen appliances and IoT devices to enhance the cooking and dining experience. This could include features such as recipe recommendations based on available ingredients, remote control of kitchen appliances, and meal planning assistance.

9)Expansion into New Markets:

Expand the reach of the One Click Eat app into new geographical markets by partnering with local restaurants and adapting the app to cater to regional preferences and cuisines.

REFERENCES

Journals:

- i) Gupta, B., Patel, A., Sharma, C. (2023). "Journal of Food Technology & Innovation," 12, 78-95.
- ii) Kumar, E., Khan, F., Singh, D. (2022). "International Journal of Consumer Studies," 45, 112-130.
- iii) Sharma, G., Patel, H., Gupta, I. (2021). "Food Technology Reviews," 8, 145-162.
- iv) Johnson, K., Smith, J., Wang, L. (2022). "Journal of Agricultural and Food Chemistry," 70, 220-235.
- v) Fernandez, P., Gonzalez, N., Rodriguez, M. (2023). "Food Chemistry," 315, 102-118.
- vi) Kumar, S., Patel, R., Sharma, T. (2024). "Journal of Food Science," 88, 45-60.
- vii) Kim, C., Lee, A., Park, D. (2023). "Food Research International," 98, 200-215.

Books:

- i) Patel, S. (2022). "Food Delivery App Marketing: Tactics and Strategies." Routledge, 180 pages.
- ii) Williams, J. (2022). "Mobile App Development for Food Services." Pearson, 200 pages.
- iii) Johnson, M. (2021). "The Business of Food Delivery: Strategies for Success." Wiley, 250 pages.
- iv) Smith, K. (2023). "User Experience Design for Food Delivery Apps." O'Reilly Media, 180 pages.
- v) Gupta, A. (2023). "Food Delivery Operations Management: Efficiency and Optimization." Springer, 220 pages.
- vi) Brown, R. (2022). "Strategic Management in Food Delivery Industry." Palgrave Macmillan, 190 pages.
- vii) Nguyen, L. (2024). "Food Delivery Technology Trends: Innovations and Future Directions." Cambridge University Press, 230 pages.
- viii) Garcia, E. (2023). "Consumer Behavior in Food Delivery Services." McGraw-Hill Education, 210 pages.

Internet Articles:

i) Doe, J. (January 15, 2024). "The Future of Food Delivery: Trends and Innovations." FoodTech News. [Online] Available: <https://www.foodtechnews.com/future-of-food-delivery-trends>

ii) Smith, J. (March 20, 2024). "How Technology is Revolutionizing Food Delivery Services." TechCrunch. [Online] Available: <https://www.techcrunch.com/technology-revolutionizing-food-delivery>

iii) Brown, E. (April 5, 2024). "The Rise of Ghost Kitchens: A New Era in Food Delivery." Food Business News. [Online] Available: <https://www.foodbusinessnews.net/ghost-kitchens-rise-food-delivery>

iv) Johnson, M. (May 10, 2024). "Contactless Delivery: Shaping the Future of Food Service." Hospitality Technology. [Online] Available: <https://www.hospitalitytech.com/contactless-delivery-future-food-service>

v) Thompson, S. (June 15, 2024). "The Role of Artificial Intelligence in Food Delivery Apps." Forbes. [Online] Available: <https://www.forbes.com/role-artificial-intelligence-food-delivery-apps>

vi) Miller, D. (July 20, 2024). "Sustainability in Food Delivery: Challenges and Opportunities." GreenBiz. [Online] Available: <https://www.greenbiz.com/sustainability-food-delivery-challenges>

vii) Rodriguez, A. (July 20, 2024). "Sustainability in Food Delivery: Environmental Impact and Solutions." GreenTech News. [Online] Available: <https://www.greentechnews.com/sustainability-food-delivery-environmental-impact>

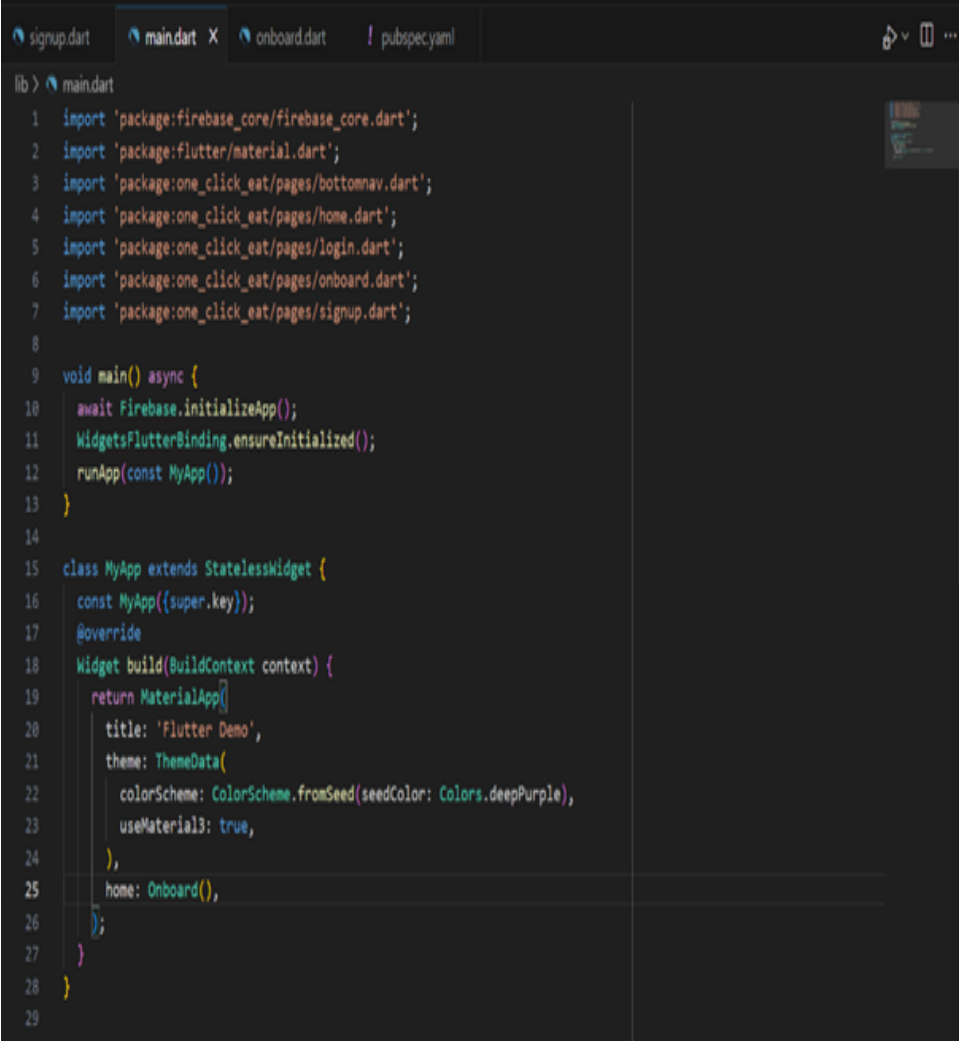
viii) Lee, J. (August 5, 2024). "Food Delivery in the Gig Economy: Opportunities and Challenges." GigEconomy Today. [Online] Available: <https://www.gigeconomytoday.com/food-delivery-gig-economy-opportunities-challenges>

ix) Miller, D. (September 10, 2024). "Regulatory Landscape for Food Delivery Platforms." Regulatory Insights. [Online] Available: <https://www.regulatoryinsights.com/food-delivery-platforms-regulatory-landscape>

APPENDICES

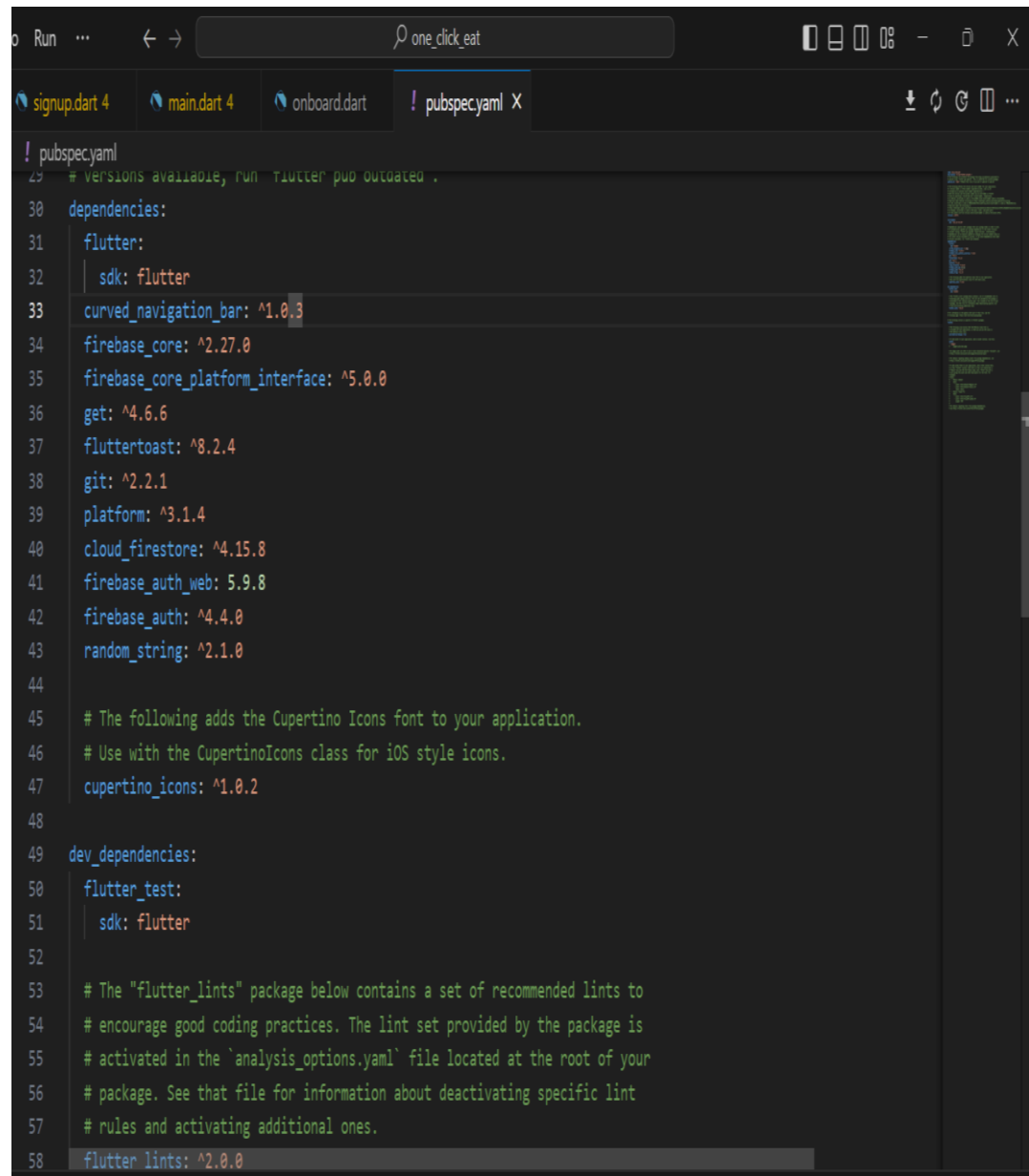
A.SAMPLE CODE

i)Main Code:



```
lib > main.dart
1  import 'package:firebase_core/firebase_core.dart';
2  import 'package:flutter/material.dart';
3  import 'package:one_click_eat/pages/bottomnav.dart';
4  import 'package:one_click_eat/pages/home.dart';
5  import 'package:one_click_eat/pages/login.dart';
6  import 'package:one_click_eat/pages/onboard.dart';
7  import 'package:one_click_eat/pages/signup.dart';
8
9  void main() async {
10     await Firebase.initializeApp();
11     WidgetsFlutterBinding.ensureInitialized();
12     runApp(const MyApp());
13 }
14
15 class MyApp extends StatelessWidget {
16     const MyApp({super.key});
17     @override
18     Widget build(BuildContext context) {
19         return MaterialApp(
20             title: 'Flutter Demo',
21             theme: ThemeData(
22                 colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
23                 useMaterial3: true,
24             ),
25             home: Onboard(),
26         );
27     }
28 }
29
```

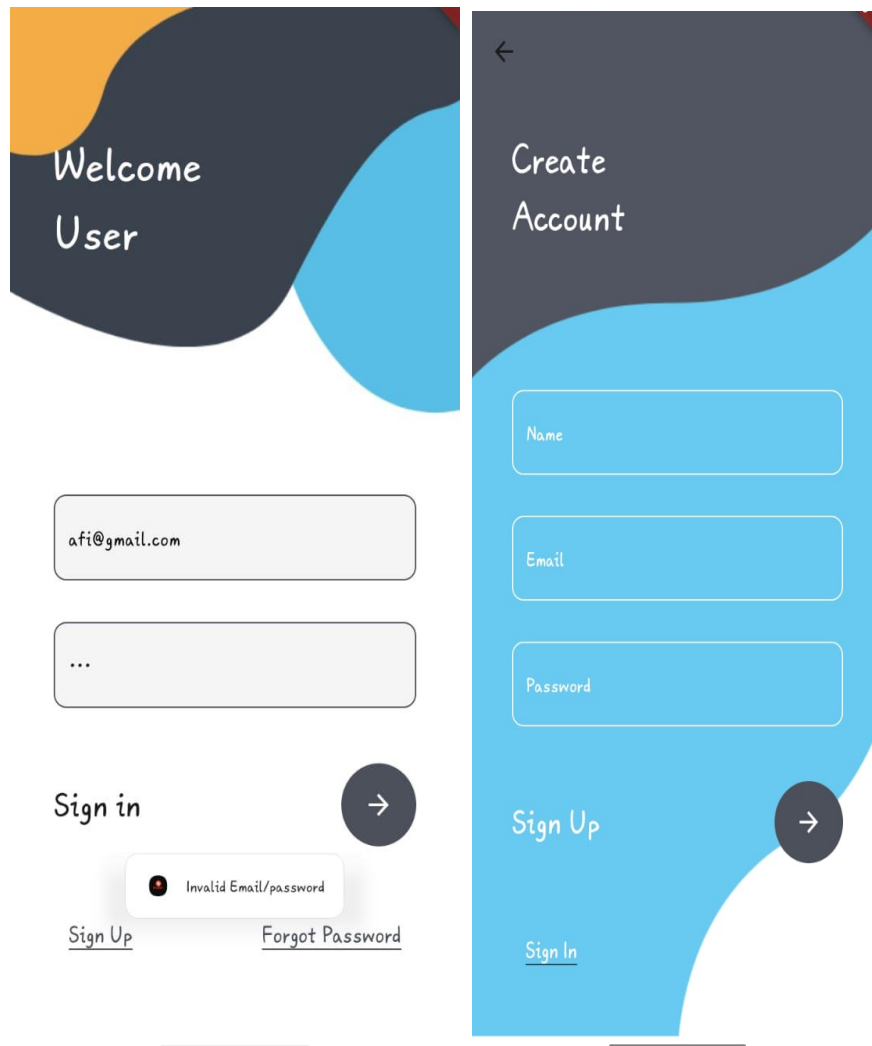
ii)pubspec.yaml



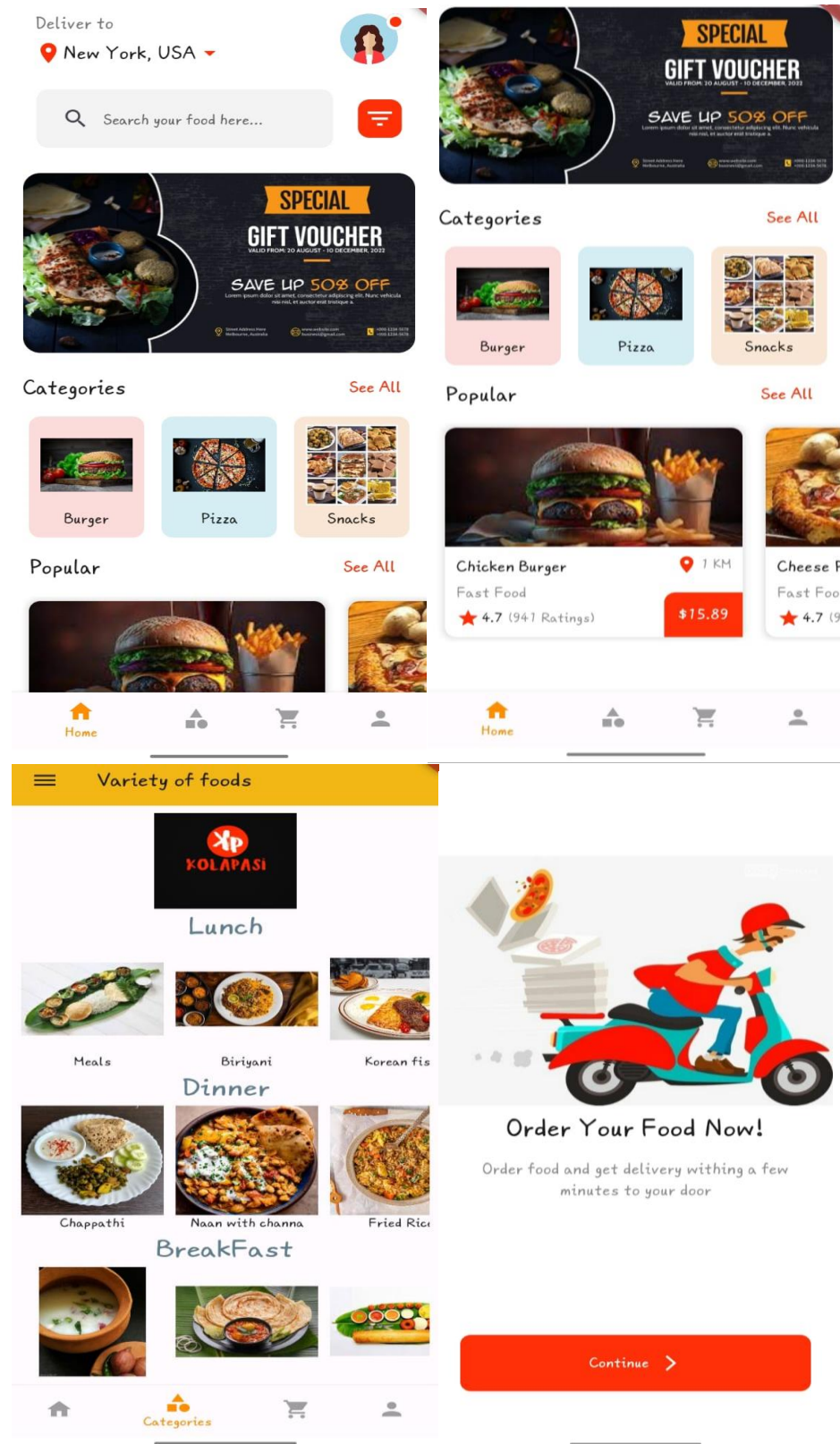
```
! pubspec.yaml
25 # Versions available, run flutter pub outdated .
30 dependencies:
31   flutter:
32     sdk: flutter
33   curved_navigation_bar: ^1.0.3
34   firebase_core: ^2.27.0
35   firebase_core_platform_interface: ^5.0.0
36   get: ^4.6.6
37   flutters_toast: ^8.2.4
38   git: ^2.2.1
39   platform: ^3.1.4
40   cloud_firestore: ^4.15.8
41   firebase_auth_web: 5.9.8
42   firebase_auth: ^4.4.0
43   random_string: ^2.1.0
44
45   # The following adds the Cupertino Icons font to your application.
46   # Use with the CupertinoIcons class for iOS style icons.
47   cupertino_icons: ^1.0.2
48
49 dev_dependencies:
50   flutter_test:
51     sdk: flutter
52
53   # The "flutter_lints" package below contains a set of recommended lints to
54   # encourage good coding practices. The lint set provided by the package is
55   # activated in the `analysis_options.yaml` file located at the root of your
56   # package. See that file for information about deactivating specific lint
57   # rules and activating additional ones.
58   flutter_lints: ^2.0.0
```

B.SCREEN SHOTS

i)Login Page And Signup Page



ii) Homepage & Navigation bar



iii)Order & Rating

←

Place Order

Food Details



Burger
Qty: 1, Price: \$10.00



Pizza
Qty: 2, Price: \$20.00

 Add More Items

 Edit

Number of Items Ordered: 3

Total Payment: \$30.00

Payment Options

 Credit Card

 Debit Card

 Cash on Delivery

←

Place Order

Car Details for Pickup (if needed)

 Car Model: XYZ

 Car Color: Black

 Car Plate Number: ABC123

 Car Rental Agency: ABC Car Rentals

 Pickup Location: XYZ Street, City

[Choose Another Car](#)

 Edit

Place Order

Rating & Notes

Rating

Notes







My Orders









My Orders

