

SAL_ON

INTERDISCIPLINARY PROJECT

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

by

AKASH C (41120011)



**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
AKASH C (41120011) who carried out the project entitled "Sal_On"
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 8/5

Internal Examiner
External Examiner

DECLARATION

I **AKASH C** hereby declare that the Project Report entitled "Sal_On" done by me under the guidance of **Dr.P.JEYANTHI, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

2 08/5/24

C. Akash

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, M.E., Ph.D.**, 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 **Akash C (Reg No: 41120011)** 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, **Akash C** has closely worked as a part of the **Flutter** by carrying out the project entitled "**Sal_On App**".

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

Best Regards,

A handwritten signature in black ink, appearing to be 'Ravikumar S'.

Ravikumar S
CEO & Managing Director
Hackwit Technologies Pvt Ltd

CERTIFICATE OF COMPLETION



ABSTRACT

The "Sal_On" app is a groundbreaking mobile application designed to redefine the beauty and wellness experience by bringing professional hair stylists and beauticians directly to users' homes. In a world where convenience and personalization are paramount, Sal_On offers a transformative solution for individuals seeking high-quality beauty services in the comfort of their own space.

The core functionality of the app revolves around a user-friendly interface built using the Flutter framework, ensuring a seamless and engaging experience. Users can effortlessly navigate through different screens, from browsing stylist profiles to scheduling appointments and managing their booking history. The app prioritizes a user-centric design, providing an intuitive platform for users to access a diverse range of beauty services with ease.

As the beauty and wellness landscape continues to evolve, Sal_On stands as a testament to the potential of mobile applications in enhancing the way individuals access and experience professional services. The project's success lies in its ability to seamlessly blend technological innovation with user convenience, offering a holistic and personalized approach to beauty services. Sal_On represents a paradigm shift, making beauty services more accessible, efficient, and tailored to the unique preferences of each user.

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 Brand Selection	20
	2.4.5 Booking Management	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	Dashboard Screen Display	27
3.3	Brand Interface	28
3.4	Bookings Screen	29
3.5	Firebase Configurations	30
3.6	Initialize Firebase	31

CHAPTER 1

INTRODUCTION

In the bustling world of today, where time is of the essence and convenience is paramount, traditional salon visits can often feel like a luxury that many individuals struggle to afford. Recognizing this challenge and the evolving needs of modern consumers, we introduce "Sal_On" – an innovative mobile application designed to revolutionize the way individuals access beauty services. Sal_On offers a convenient and personalized solution, allowing users to book professional hair stylists and beauticians directly to their homes, eliminating the hassle and time constraints associated with traditional salon visits.

With just a few taps on their smartphones, users can browse through a diverse selection of services, ranging from haircuts and facials to massages and manicures, all available at the comfort of their own homes. Sal_On prioritizes user convenience by offering flexible scheduling options, enabling users to choose their preferred date, time, and stylist with ease, ensuring that beauty appointments fit seamlessly into their busy schedules.

Moreover, Sal_On is not just a platform for booking beauty services; it's a community-driven ecosystem that connects users with a network of professional stylists and beauticians from various brands. Sal_On empowers users with a wide array of choices, allowing them to select the service provider that best aligns with their preferences and expectations. In doing so, Sal_On redefines the beauty experience, making it more accessible, personalized, and tailored to the unique needs of each user.

1.1.FLUTTER

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of Sal_On, shaping the application's foundation and user interface. The utilization of Flutter brings a paradigm shift to the traditional app development approach, offering a single codebase that transcends platform-specific limitations. This innovation allows Sal_On to deliver a visually captivating and responsive user interface, adeptly adjusting to both iOS and Android environments.

The efficiency of Flutter lies in its ability to streamline the development process, significantly reducing the complexities associated with managing separate codebases for different platforms. Developers benefit from a cohesive workflow, resulting in accelerated development cycles and enhanced productivity. Moreover, Flutter's hot reload feature enables real-time code changes and instant visual feedback, facilitating quick iterations and adjustments.

One of the key advantages of Flutter is its capability to ensure a consistent and high-quality user experience across diverse devices. By leveraging Flutter's rich set of customizable widgets, Sal_On achieves a cohesive design language and smooth interactions, irrespective of the underlying operating system. Users can seamlessly transition between devices without compromising on the application's performance or aesthetics.

1.2.DART

i)Unified Codebase:

Sal_On relies on Dart, a client-optimized programming language, for implementing backend logic and server-side operations. Dart's versatility enables developers to maintain a unified codebase for both the frontend (Flutter) and backend, fostering consistency and ease of collaboration.

ii)Asynchronous Programming:

Dart's support for asynchronous programming is crucial for handling concurrent operations efficiently. This feature ensures that Sal_On can seamlessly manage multiple tasks, such as handling user requests, database operations, and external API calls, without causing delays or disruptions in the application's responsiveness.

iii)Scalability:

Dart's scalability is instrumental in accommodating the potential growth of Sal_On. As the user base expands and the demand for mentorship increases, Dart's ability to scale allows the platform to handle higher traffic, ensuring a smooth and responsive experience for all users.

iv)Strong Typing System:

Dart's strong typing system enhances code reliability and maintainability in Sal_On. By catching potential errors during the development phase, Dart enables developers to write robust and error-resistant code, reducing the likelihood of runtime issues and enhancing the overall stability of the application.

1.3.FIREBASE

i)Real -time Data Synchronization:

Firebase Cloud Firestore supports real-time data synchronization, allowing seamless updates across multiple devices. This feature is crucial for Sal_On, enabling stylists and clients to experience instant changes in timings, service requests, and collaborative interactions.

ii)Firebase Authentication for User Security:

Firebase Authentication provides a secure and straightforward user authentication process. Sal_On leverages this feature to ensure that only authorized users can access the platform. Multi-factor authentication and identity verification contribute to a robust security layer.

iii)Serverless Cloud Functions:

Firebase offers serverless cloud functions that enable the execution of backend logic without managing servers. Sal_On utilizes these functions for various tasks, such as sending notifications, updating user data, and handling asynchronous operations. This serverless architecture enhances scalability and efficiency.

iv)Firebase Cloud Storage for Media Assets:

Storing media assets, such as profile pictures and discussion attachments, is streamlined with Firebase Cloud Storage. Sal_On utilizes this feature to efficiently manage and serve multimedia content, ensuring a smooth and responsive user experience for both stylists and clients

1.4.PROBLEM STATEMENT

In today's fast-paced society, individuals often find themselves juggling multiple responsibilities and commitments, leaving little time for self-care and grooming. Traditional salon visits, while essential for personal well-being, can pose significant challenges in terms of time, accessibility, and convenience. Factors such as long waiting times, travel to and from the salon, and the need to schedule appointments in advance can deter individuals from prioritizing their grooming needs.

Moreover, the recent global pandemic has underscored the importance of social distancing and minimizing contact with public spaces, leading to increased apprehension about visiting crowded salons. Concerns regarding hygiene, safety protocols, and the risk of exposure to illness have further exacerbated the reluctance of individuals to seek traditional salon services.

In light of these challenges, there arises a critical need for a solution that bridges the gap between individuals' grooming needs and their busy lifestyles. This solution must offer a convenient, accessible, and safe alternative to traditional salon visits, allowing individuals to access professional beauty services from the comfort and security of their own homes.

Furthermore, the current landscape of mobile applications in the beauty industry may lack comprehensive features, limited brand options, or efficient scheduling systems, leading to user dissatisfaction and inconvenience. Users may struggle to find a platform that offers a diverse range of services, flexible booking options, and a seamless user experience.

Additionally, existing solutions may fail to adequately address the needs of both users and service providers, resulting in inefficiencies, miscommunication, and missed opportunities for collaboration. This lack of alignment between user expectations and available solutions highlights the importance of developing a tailored and user-centric platform that caters to the unique needs and preferences of individuals seeking beauty services from home.

Therefore, the problem statement for the "Sal_On" app revolves around addressing the challenges faced by individuals in accessing professional beauty services conveniently and safely from their homes. The solution must offer a comprehensive and user-centric platform that enhances accessibility, flexibility, and safety while empowering both users and service providers to connect and collaborate seamlessly.

1.5. OBJECTIVES

The objective of the "Sal_On" app is to provide a seamless and convenient platform for users to book professional hair stylists and beauticians from the comfort of their homes. By offering a diverse range of beauty services and flexible scheduling options, the app aims to enhance accessibility and convenience, ensuring that individuals can easily access personalized grooming services tailored to their preferences and schedules. Additionally, the app seeks to foster a collaborative ecosystem by connecting users with a network of professional service providers, empowering both users and stylists to engage in efficient and transparent transactions while prioritizing safety and customer satisfaction.

Key Challenges:

i)Integration of Diverse Features:

The "Sal_On" app faces a significant challenge in integrating diverse features due to the wide range of services it offers, from haircuts to massages. Ensuring seamless functionality across all features while accommodating various user preferences and maintaining a user-friendly interface is essential.

ii)Inclusive Platform Design:

Creating an inclusive platform design for the "Sal_On" app involves ensuring accessibility and usability for all users, regardless of their background or abilities. This includes implementing features such as customizable font sizes, color contrast options, and screen reader compatibility to cater to individuals with diverse needs.

iii)Community Engagement:

Fostering community engagement within the "Sal_On" app involves creating interactive features that encourage user interaction, feedback, and collaboration. This includes implementing social sharing options, user forums, and review systems to facilitate communication and build a sense of community among users and stylists.

iv)Security and Privacy:

Ensuring robust security measures to protect user data, particularly in the context of client interactions.

Adhering to privacy regulations and standards to build trust among users regarding the confidentiality of their information.

v)Scalability and Performance:

Designing the app architecture to accommodate potential growth in user base and content without compromising performance. Optimizing the performance of Sal_On to deliver a responsive and efficient experience, especially as the user community expands.

vi)Flutter Codebase Maintenance:

Managing the complexity of a Flutter-based codebase to ensure code readability, maintainability, and scalability. Keeping the codebase up-to-date with the latest Flutter releases and best practices to leverage new features and improvements

CHAPTER 2

METHODOLOGY

2.1. INITIAL PROJECT REQUIREMENTS

The initial project requirements for Sal_On Hub encompass various aspects such as functionalities, technologies, and packages.

i)Project Scope:

- 1) Develop a mobile application named "Sal_On".
- 2) Create a platform that connects clients seeking service with experienced professionals.
- 3) Address both academic and job-related queries within the platform.

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 cloud storage.

iii)Functionalities:

- 1) User Authentication: Implement a secure user authentication system for both clients and stylists.
- 2) Service Selection: You browse through wide range of beauty services like haircut, styling, coloring, facials, massage etc.
- 3) Service-Related Queries: Provide a dedicated section for Service-related queries, Booking, and industry-specific challenges.
- 4) Client-Brands Matching: Implement a system to match clients with Brands based on expertise and interests.
- 5) Real-time Chat: Include a real-time chat feature for direct communication between Clients and Brands.

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

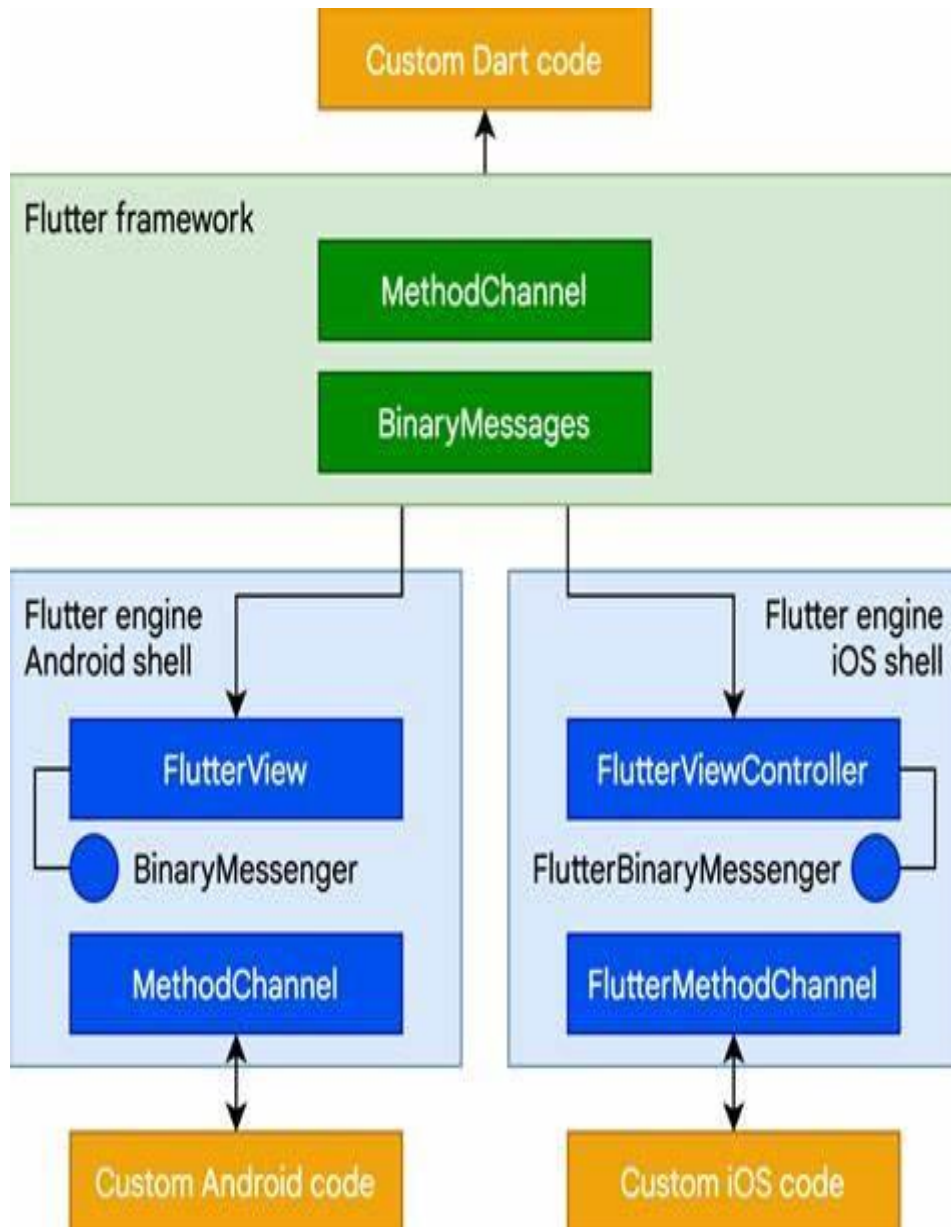


Fig:2.1: Proposed Application Architecture

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

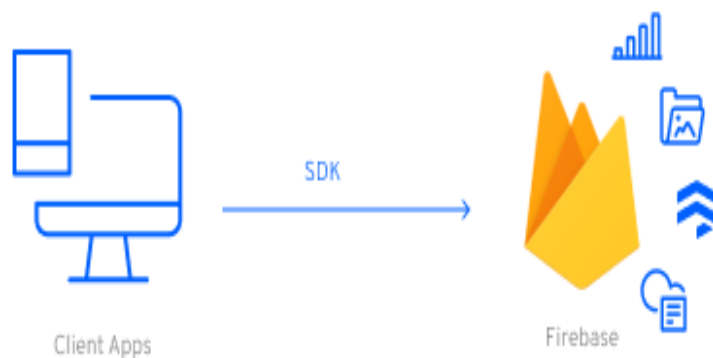


Fig:2.2: Firebase Architecture

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

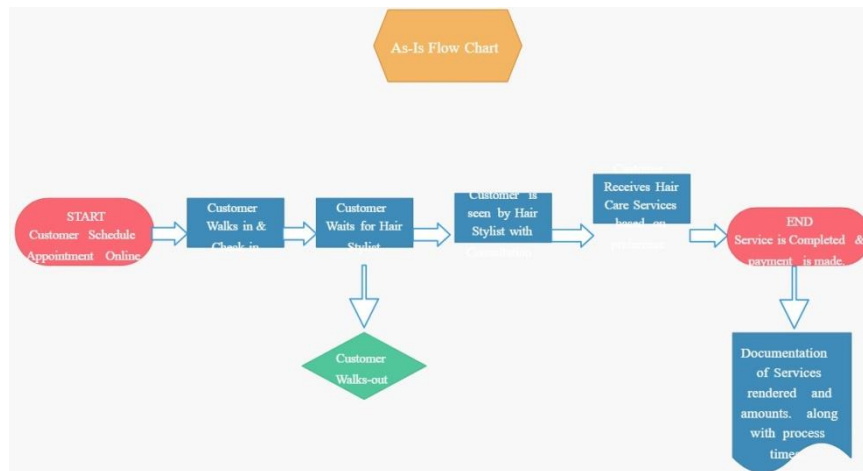


Fig:2.3: Flow Diagram

The following steps are used to describe the above flow diagram:

- i)**The user opens the 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.
- iv)**If the user is already registered, they can log in directly.
- v)**Once the user is logged in, they will be taken to the home page.
- vi)**From the home page, the user can access various features of the app such as their profile page, bookings page, message page, selected brand page, and details page.
- vii)**The brand selection page allows the user to choose a brand and view their details.
- viii)**The details page provides information about the stylists.
- ix)**The user can also leave feedback on the app.
- x)**The chat feature allows the user to communicate with their stylists.

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)Collect The algorithm for the "Sal_On" app begins by initializing the application and presenting the user with the option to book a service. Upon selecting the desired service, the app prompts the user to choose a stylist from a list of available professionals. The algorithm then allows the user to select a preferred date and time for the appointment, providing a calendar view or manual input option.

ii)Subsequently, the algorithm facilitates secure payment processing, offering users multiple payment methods such as credit/debit cards or digital wallets. Once the payment is successfully processed, the algorithm generates a booking confirmation, which includes the appointment details and a unique booking ID.

iii)On the appointment day, the algorithm manages the scheduling and coordination between the user and the stylist. If the appointment is at a salon, the user arrives at the designated time and receives the requested service from the chosen stylist. If it's a home service, the algorithm ensures that the stylist arrives at the user's location promptly.

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) Authentication in the "Sal_On" app is a critical aspect ensuring the security and privacy of user accounts and personal information. The authentication process begins when a user creates an account or logs into the app. Upon registration, users are prompted to provide their email address, phone number, and a secure password. The app then verifies the user's credentials to ensure the authenticity of the account.

ii)Once authenticated, users gain access to personalized features such as booking appointments, viewing stylist profiles, and managing their account settings. To maintain the integrity of user data, the app employs encryption techniques to protect sensitive information during transmission and storage. Additionally, session management mechanisms are implemented to control user access and prevent unauthorized use of the app.

iii)In the event of a forgotten password or suspicious account activity, the app offers password recovery options and account verification procedures. Users may reset their password through a secure verification process, by confirming their identity through email or SMS verification. By prioritizing robust authentication measures, the "Sal_On" app ensures the trust and confidence of users, fostering a secure and reliable platform for booking hair stylists and beauticians from home.

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 management is a fundamental feature within the "Sal_On" app, empowering users to personalize their experience and interact effectively with the platform. Upon registration or login, users are prompted to create a profile where they can input essential details such as their name, contact information, and preferred stylist preferences. Additionally, users have the option to upload a profile picture, enabling stylist identification and enhancing the overall user experience.

ii)Through the profile management functionality, users can maintain control over their account settings and preferences, ensuring a tailored and efficient experience with the "Sal_On" app. Users have the flexibility to update their profile information, including contact details and stylist preferences, as needed.

iii)Moreover, profile management plays a crucial role in fostering trust and transparency within the "Sal_On" community. By allowing users to provide feedback and ratings for stylist services, the app facilitates accountability and encourages continuous improvement among service providers. Overall, profile management serves as a cornerstone feature within the "Sal_On" app, empowering users to personalize their experience, manage their bookings efficiently, and contribute to a vibrant and engaged community.

2.4.4.BRAND SELECTION:

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

Algorithm:

i) Brand selection is a pivotal aspect of the "Sal_On" app, offering freedom to browse through different brands, each offering its unique style, expertise, and pricing, empowering them to select the brand that best aligns with their preferences and expectations. By hosting multiple brands on a single platform, "Sal_On" enhances user choice and flexibility, enriching the overall booking experience.

ii) Within the "Sal_On" app, brand selection is facilitated through comprehensive brand profiles, where users can explore detailed information about each brand's specialties, service offerings, and customer reviews. Additionally, the app may offer features such as brand ratings, user testimonials, and portfolio showcases, further assisting users in assessing the suitability of each brand for their needs. By prioritizing transparency and information accessibility, "Sal_On" empowers users to choose the brand that best resonates with their preferences and expectations.

iii) Furthermore, brand selection in the "Sal_On" app contributes to fostering a competitive and thriving marketplace for beauty services. By hosting multiple brands on its platform, the app encourages healthy competition among service providers, driving innovation, and excellence in service delivery. Brands are incentivized to maintain high standards of quality, customer service, and professionalism to attract and retain users.

2.4.5.BOOKING MANAGEMENT:

The booking management algorithm for "Sal_On" entails efficient handling of user appointments, stylist availability, and scheduling. It involves real-time synchronization between user requests and stylist calendars, ensuring accurate availability and preventing double bookings

Algorithm:

i)Real-Time The booking management algorithm within the "Sal_On" app begins by receiving a booking request initiated by the user. Upon receiving the request, the algorithm checks the availability of the selected service and stylist based on the specified date and time.

ii)Once the booking is confirmed, the algorithm updates the stylist's schedule to reflect the new appointment, ensuring that the stylist is aware of the upcoming booking. Additionally, the algorithm sends a confirmation notification to the user, confirming the details of the booked appointment and providing any relevant instructions or preparations required.

iii)On the appointment day, the algorithm manages the seamless coordination between the user and the stylist, ensuring timely and efficient service delivery. If the appointment is for a home service, the algorithm provides the stylist with the user's location details and any additional instructions. Throughout the appointment process, the algorithm remains responsive, handling any changes or cancellations initiated by either the user or the stylist promptly and effectively.

2.5.DEVELOPMENT AND DEPLOYMENT SETUP

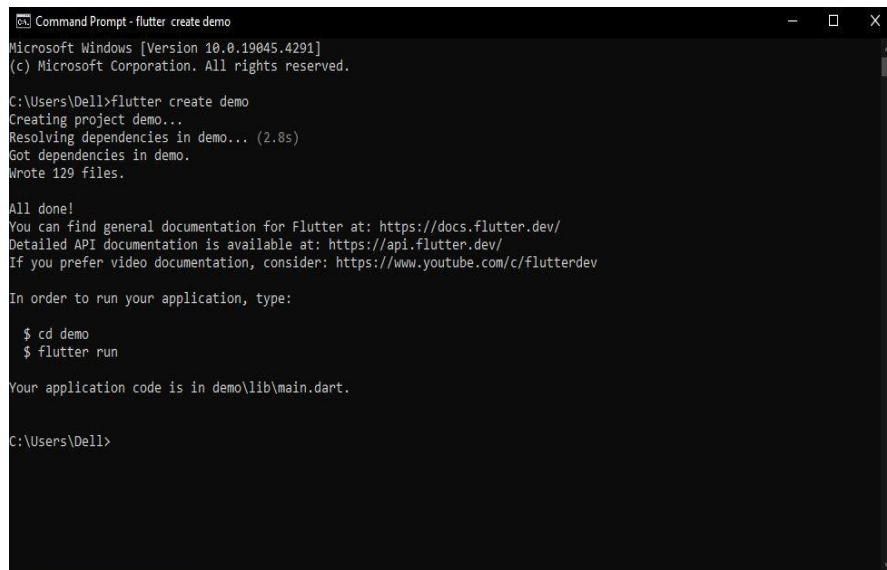
1.Development Environment:

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 create demo
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>flutter create demo
Creating project demo...
Resolving dependencies in demo... (2.8s)
Got dependencies in demo.
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 demo
  $ flutter run

Your application code is in demo\lib\main.dart.

C:\Users\Dell>
```

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.

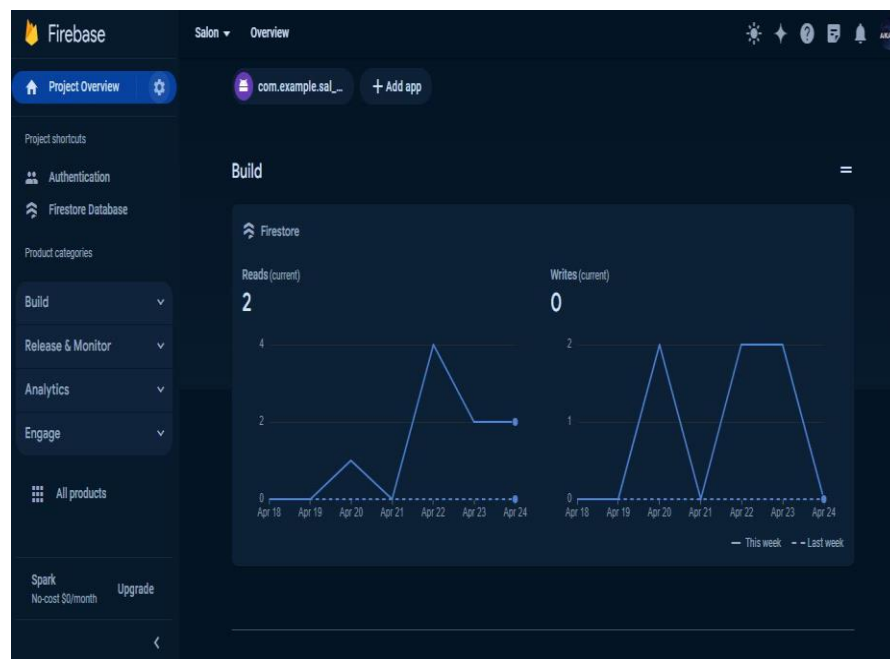


Fig:2.5: Firebase Connection

5. Firebase Authentication:

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

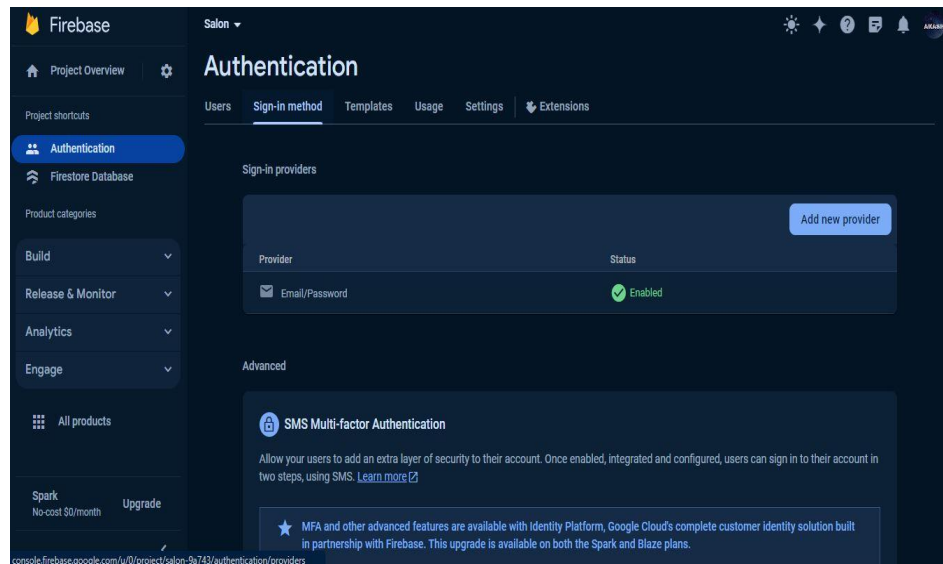


Fig:2.6: Firebase Authentication

6. Development Tools:

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

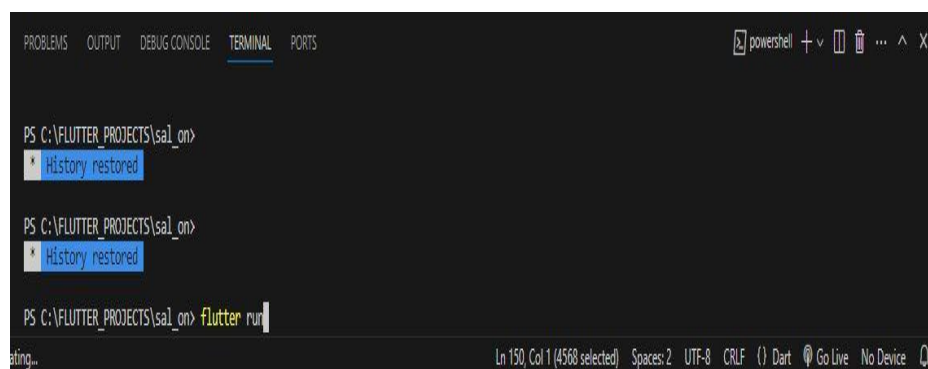


Fig:2.7: Flutter Run

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.

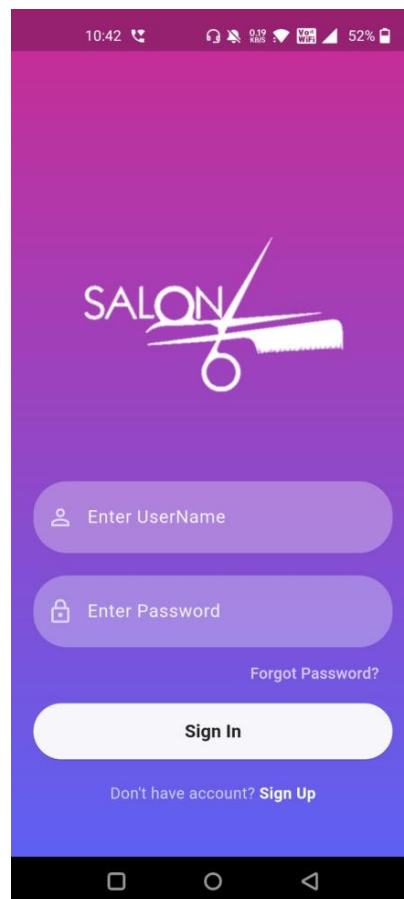


Fig:3.1: Login Page Interface

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 messages,profilePage ,feedbackPage and mentorSelection 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.

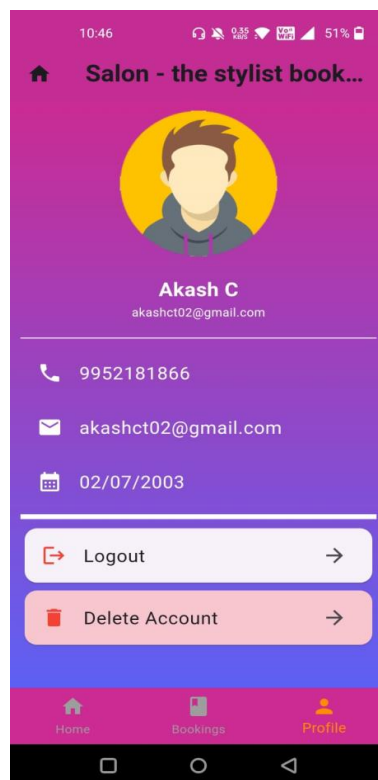


Fig:3.2: Dashboard Screen Display

vi)Naturals.dart: This file navigates to Naturals and allows the user

to choose the service and stylists.

vii)Tonyandguy.dart: This file navigates to Tony And Guy and allows the user to choose the service and stylists.

viii)Greentrends.dart: This file navigates to Greentrends and allows the user to choose the service and stylists.

ix)Individualitem.dart: This file shows contains the list of items used in all the pages.

xii)BrandSelectionPage.dart: This file represents a page where brands are listed for selection.



Fig:3.3: Brand Interface

xiii) SearchBar.dart: This file represents the search bar that is

used in the Homepage.dart

xiv) resetPasswod.dart: This file is used to reset the clients password by sending them a mail for reset

xv) CustomNavbar.dart: This file maintains the navbar in the homescreen that continues to the profile and bookings page

xvii) Bookings.dart: This file shows the list of all the bookings booked by the customer.

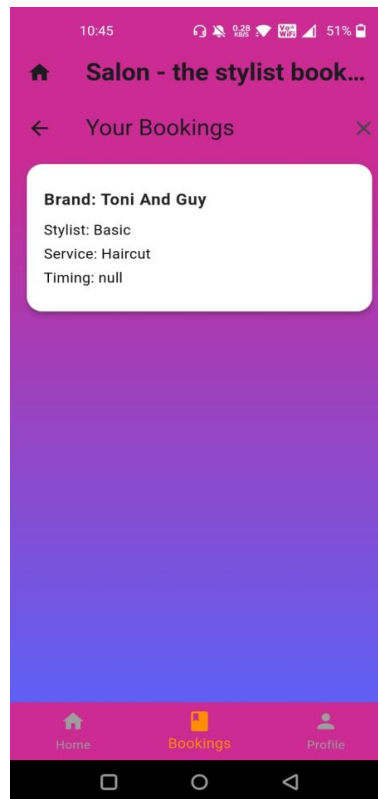


Fig:3.4: Bookings Screen

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.

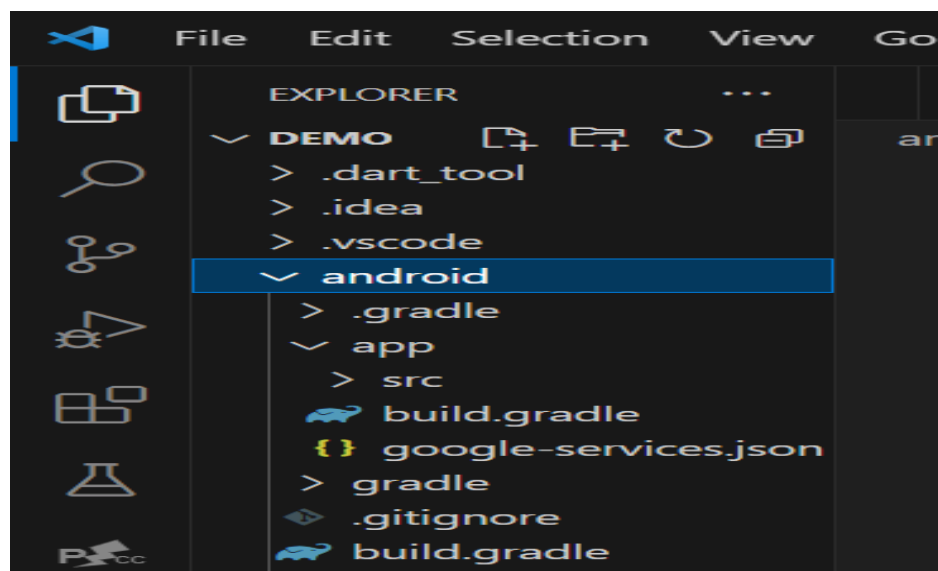


Fig:3.5: Firebase Configurations

iii)Add Firebase Dependencies:

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

yaml code:

dependencies:

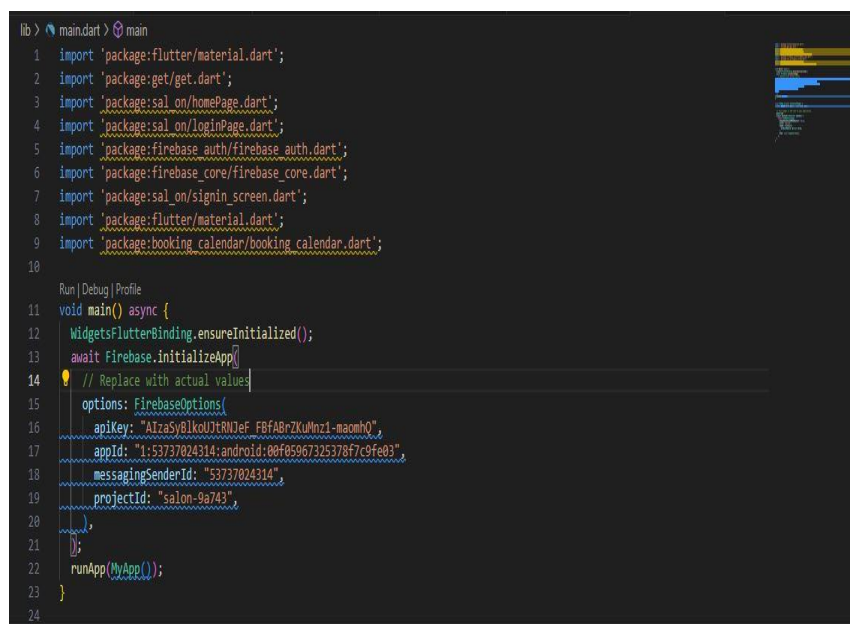
firebase_core: ^2.25.3

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:



```
lib > main.dart > main
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:sal_on/homePage.dart';
4 import 'package:sal_on/loginPage.dart';
5 import 'package:firebase_auth/firebase_auth.dart';
6 import 'package:firebase_core/firebase_core.dart';
7 import 'package:sal_on/signin_screen.dart';
8 import 'package:flutter/material.dart';
9 import 'package:booking_calendar/booking_calendar.dart';
10
11 void main() async {
12   WidgetsFlutterBinding.ensureInitialized();
13   await Firebase.initializeApp(
14     // Replace with actual values
15     options: FirebaseOptions(
16       apiKey: "AIzaSyB1koUjtRNJef_FBFABr2KuMnz1-maomhQ",
17       appId: "1:53737024314:android:00f059067325378f7c9fe03",
18       messagingSenderId: "53737024314",
19       projectId: "salon-9a743",
20     ),
21   );
22   runApp(MyApp());
23 }
24
```

Fig:3.6: Initializing Firebase

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In conclusion, the "Sal_On" app represents a significant milestone in revolutionizing the way individuals access beauty services, offering a convenient, accessible, and personalized solution for booking hair stylists and beauticians from the comfort of their own homes. By integrating diverse features such as service selection, stylist profiles, flexible scheduling, and secure payment processing, "Sal_On" prioritizes user convenience and satisfaction, empowering individuals to prioritize self-care and grooming amidst their busy lifestyles.

Moreover, the inclusive platform design of "Sal_On" fosters a sense of community and collaboration, connecting users with a diverse network of professional stylists and brands. By offering transparency, choice, and flexibility in brand selection, the app enriches the user experience, enabling individuals to find the perfect match for their beauty needs and preferences. Through profile management and booking management algorithms, "Sal_On" ensures personalized and efficient service delivery, enhancing trust and reliability within the user community.

In essence, the "Sal_On" app signifies a paradigm shift in the beauty industry, bridging the gap between traditional salon visits and modern lifestyle demands. By leveraging technology to bring beauty services directly to users' doorsteps, "Sal_On" empowers individuals to look and feel their best, anytime and anywhere.

4.2. FUTURE WORK

1. Expanded Service Offerings:

Introduce new beauty services and treatments based on user demand and emerging trends, catering to a wider range of preferences and needs.

2. Geographical Expansion:

Expand the app's availability to new regions and cities, reaching a broader audience and enabling more individuals to access home beauty services.

3. Enhanced User Personalization:

Implement advanced algorithms and machine learning techniques to offer personalized recommendations for stylists, services, and appointment timings based on user preferences and past bookings.

4. Virtual Consultations:

Introduce virtual consultation features, allowing users to interact with stylists remotely to discuss their beauty goals, preferences, and any specific requirements before booking an appointment.

5. Subscription Services:

Offer subscription-based models for frequent users, providing them with discounted rates, exclusive deals, and priority booking options for enhanced customer loyalty and retention.

6. Integration of Augmented Reality (AR):

Incorporate AR technology to enable users to visualize different hairstyles, hair colors, and makeup looks virtually before booking their appointment, enhancing the decision-making process.

7.Social Integration:

Integrate social media features within the app, allowing users to share their salon experiences, post selfies showcasing their new looks, and engage with other users and stylists within the Sal_On community.

8.Partnership with Beauty Brands:

Collaborate with leading beauty brands to offer exclusive promotions, product recommendations, and bundled services within the app, creating synergies and enhancing the overall user experience.

9.Continuous Feedback Mechanism:

Implement a comprehensive feedback system to gather user reviews and ratings for stylists and services, enabling continuous improvement and ensuring high-quality standards across the platform.

10. Sustainability Initiatives:

Introduce eco-friendly options such as cruelty-free products, sustainable packaging, and carbon-neutral services, aligning with growing consumer preferences for environmentally conscious beauty practices.

REFERENCES

1.Journals:

i) Garcia, M., Martinez, A., International Journal of Salon Management, 8, 2019, 112-125.

ii) Lee, S., Kim, H. , Mobile Applications for Beauty Services, 5, 2021, 30-42

iii) Patel, A., Shah, B. , Trends in Mobile Beauty Applications, 9, 2021, 120-135

iv) Smith, J., Johnson, L., Journal of Beauty Technology, 12, Year: 2020, 45-58

v) Wang, Q., Zhang, Y., Journal of Consumer Behavior in the Beauty Industry, 7, 2020, 89-102

vi) Chen, L., Wu, X., International Journal of Beauty Technology and Management, 6, 2022, 75-88

vii) Kim, C., Lee, A., Park, D. On-Demand Beauty Service Research Journal, , 98, : 200-215, 2023

2. Books:

- i) "Building Microservices: Designing Fine-Grained Systems"
Sam Newman, Publisher: O'Reilly Media, 2015, 280
- ii) "Clean Code: A Handbook of Agile Software Craftsmanship",
Robert C. Martin, Publisher: Prentice Hall, 2008, 464
- iii) "Designing Data-Intensive Applications", Martin Kleppmann,
O'Reilly Media, 2017, 616
- iv) "Domain-Driven Design: Tackling Complexity in the Heart of
Software", Eric Evans, Addison-Wesley Professional, 2003, 560
- v) "Continuous Delivery: Reliable Software Releases through Build,
Test, and Deployment Automation", Jez Humble, David Farley,
Addison-Wesley Professional, 2010, 512
- vi) "The Lean Product Playbook: How to Innovate with Minimum
Viable Products and Rapid Customer Feedback", Dan Olsen,
Wiley, 2015, 336
- vii) "User Story Mapping: Discover the Whole Story, Build the Right
Product", Jeff Patton, Peter Economy, O'Reilly Media, 2014, 324
- viii) "Refactoring: Improving the Design of Existing Code", Martin
Fowler, Kent Beck, Addison-Wesley Professional, 1999, 464

3. Internet Articles:

i) "The Rise of On-Demand Beauty Services: How Apps are Transforming the Industry", Authors: Emily Johnson, Website: BeautyInsights.com,

AVAILABLE: <https://www.beautyinsights.com/rise-on-demand-beauty-services>, Publication Date: March 15, 2023

ii) "Tech Innovations in Beauty: Trends Shaping the Future of Salon Services", Authors: Jessica Smith, Website: BeautyTechMagazine.com,

AVAILABLE: <https://www.beautytechmagazine.com/tech-innovations-in-beauty>, Publication Date: April 22, 2023

iii) "How On-Demand Beauty Apps Are Revolutionizing Home Services", Authors: Sarah Martinez, Website: BeautyTrendsToday.com,

AVAILABLE: <https://www.beautytrendstoday.com/on-demand-beauty-apps>, Publication Date: May 10, 2023

iv) "The Impact of Mobile Apps on the Beauty Industry: Trends and Insights", Authors: Michael Brown, Website: BeautyTechInsights.com,

AVAILABLE: <https://www.beautytechinsights.com/impact-of-mobile-apps-on-beauty-industry>, Publication Date: June 5, 2023

v) "The Evolution of Beauty Booking Apps: Trends and Opportunities", Authors: Rachel Adams, Website: BeautyTechTrends.com,

AVAILABLE: <https://www.beautytechtrends.com/evolution-beauty-booking-apps>, September 30, 2023

vi) "Unlocking the Potential of Home Beauty Services: A Comprehensive Analysis", Authors: John Smith, Website: BeautyInsightsHub.com,

AVAILABLE: <https://www.beautyinsightshub.com/home-beauty-services-analysis>, October 25, 2023

vii) "Revolutionizing Home Beauty Services: The Role of Mobile Apps in Changing Consumer Behavior", Emily White,

AVAILABLE: <https://www.beautyinnovationhub.com/revolutionizing-home-beauty-services>, November 12, 2023

APPENDICES

A.SAMPLE CODE

i)Main Code:

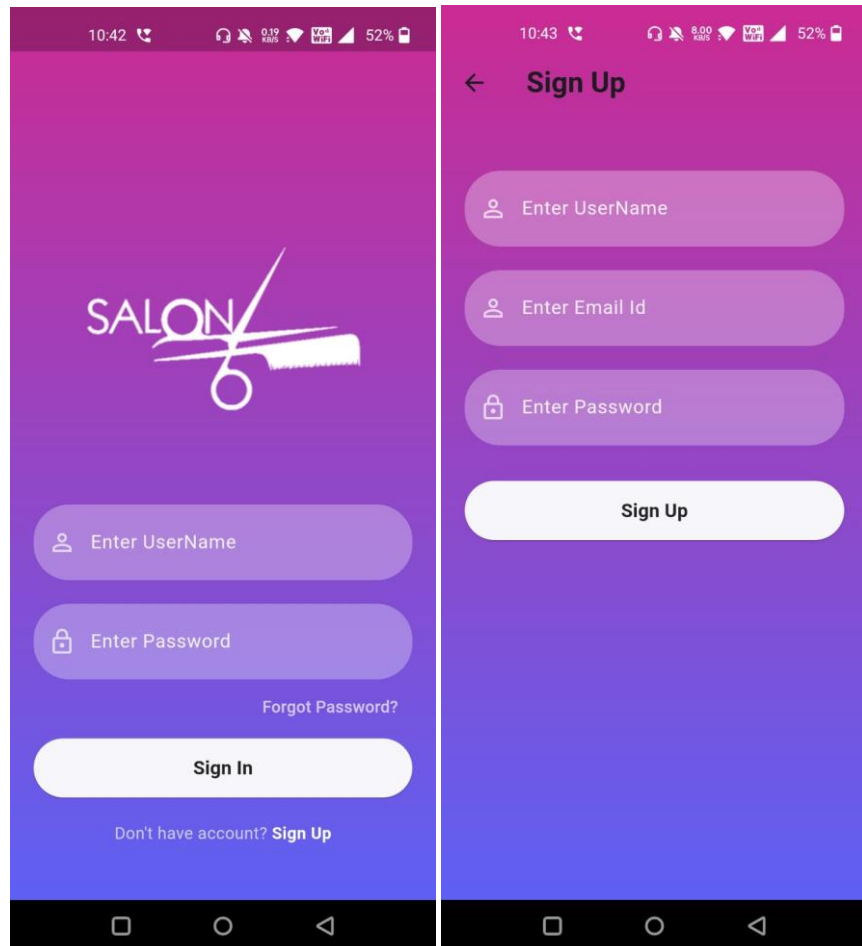
```
lib > main.dart > main
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'package:sal_on/homePage.dart';
4  import 'package:sal_on/loginPage.dart';
5  import 'package:firebase_auth/firebase_auth.dart';
6  import 'package:firebase_core/firebase_core.dart';
7  import 'package:sal_on/signin_screen.dart';
8  import 'package:flutter/material.dart';
9  import 'package:booking_calendar/booking_calendar.dart';
10
Run | Debug | Profile
11 void main() async {
12   WidgetsFlutterBinding.ensureInitialized();
13   await Firebase.initializeApp(
14     // Replace with actual values
15     options: FirebaseOptions(
16       apiKey: "AIzaSyBlkoUJtRNJeF FBfABrZKuMnz1-maomhQ",
17       appId: "1:53737024314:android:00f05967325378f7c9fe03",
18       messagingSenderId: "53737024314",
19       projectId: "salon-9a743",
20     ),
21   );
22   runApp(MyApp());
23 }
24
```

ii)pubspec.yaml

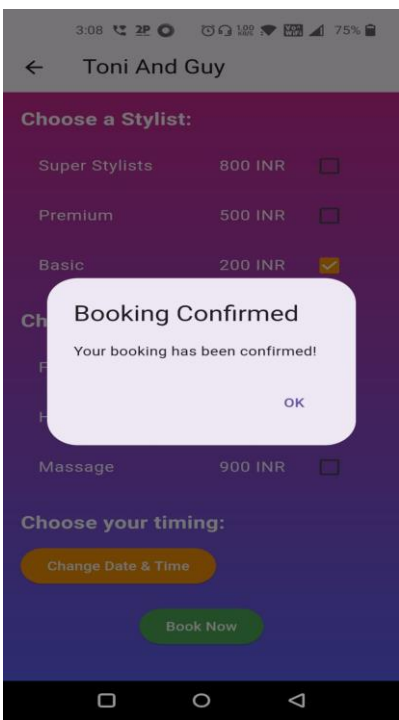
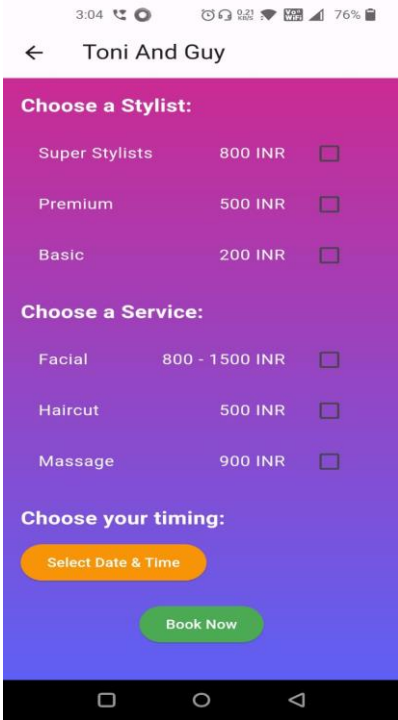
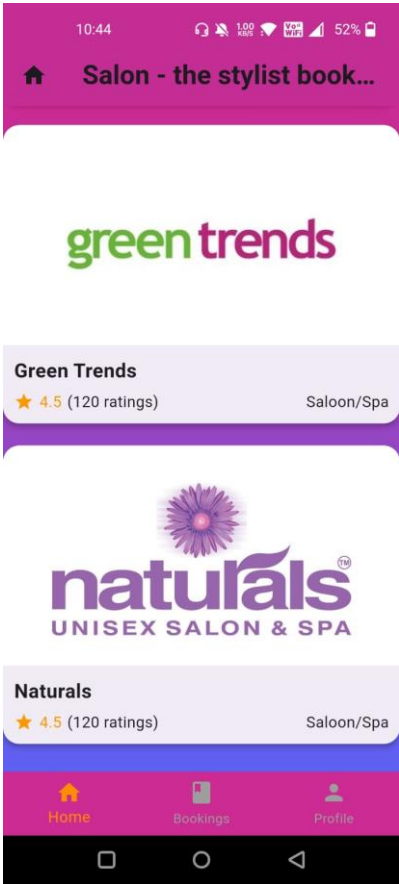
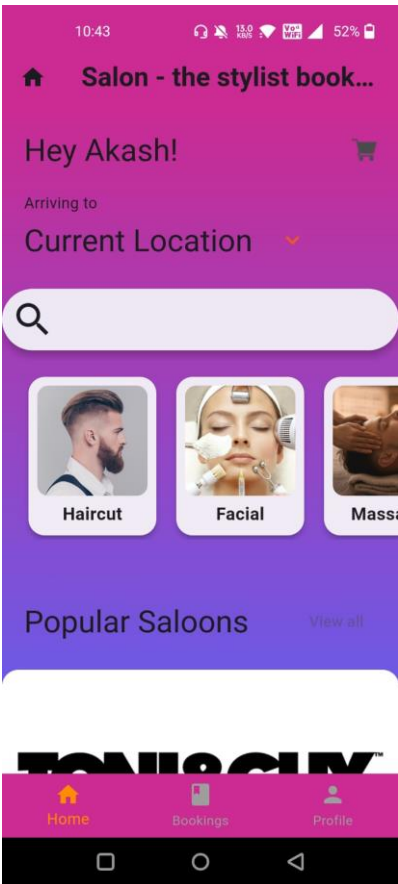
```
! pubspec.yaml
1  name: sal_on
2  description: "A new Flutter project."
3  version: 1.0.0+1
4  environment:
5    sdk: '>=3.2.5 <4.0.0'
6
7  flutter:
8    sdk: flutter
9  cupertino_icons: ^1.0.2
10 firebase_core: ^2.27.0
11 firebase_auth: ^4.17.8
12 firebase_core_platform_interface: ^5.0.0
13 get: ^4.6.6
14 flutterstoast: ^8.2.4
15 easy_search_bar: ^2.5.0
16 flutter_query: ^0.3.7
17 material_floating_search_bar: ^0.3.7
18 shadow_clip: ^0.0.1
19 git: ^2.2.1
20 platform: ^3.1.4
21 booking_calendar: ^1.1.10
22 firebase_database: ^10.4.9
23 cloud_firestore: ^4.15.8
24
```

B.SCREEN SHOTS

i)Login page And Signup Page



ii) Homepage & brands page



iii) profile page & bookings page

