

Neta4U On-Road Car Repair Servicing

INTERDISCIPLINARY PROJECT

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

By

MAJID FASI AHMED (Reg.No-41120107)



**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 **MAJID FASI AHMED (41120107)** who carried out the project entitled "**Neta4U On-Road Car Repair Servicing**" under my supervision from January 2024 to May 2024.

Internal Guide

Ms.P.Shamili, M.Tech.,

Head of the Department

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

Submitted for Viva voce Examination held on 08/05/24

Internal Examiner
External Examiner

DECLARATION

I **Majid Fasi Ahmed** hereby declare that the Project Report entitled "Neta4U On-Road Car Repair Servicing" done by me under the guidance of **Ms.P.Shamili M.Tech.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 08/05/24

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 **Ms.P.Shamili M.Tech.**, 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

INTERNSHIP OFFER LETTER

January 8, 2024, 2023

Majid Fasi Ahmed
Sathyabama Institute of Science & Technology,
Chennai, Tamil Nadu.

Dear **Mr. Majid Fasi Ahmed,**

We are delighted to share that your Internship at Hackwit Technologies Pvt Ltd., has been Extended. Your performance was outstanding in the projects, so we have considered to extend your Internship.

Your internship will continue for another 6 Months. So your Internship will come to an end on July 2024. During the internship with the company, You will be paid a stipend of Rs. 5000 (Five Thousand Only) monthly.

You will not be eligible for any benefits that regular employees receive. Please let us know Your acceptance on this offer.

If you have any queries related to this offer letter, please be free to contact us at admin@hackwittechnologies.com.

Sincerely,

Ravikumar S
Managing Director
For Hackwit Technologies Pvt Ltd,



ABSTRACT

In the ever-evolving landscape of automotive services, Neta4U emerges as a trailblazer, redefining the paradigm of car care through its innovative integration of WhatsApp communication. This paper delves into the transformative potential of Neta4U, offering a comprehensive analysis of its multifaceted approach to on-road car repair servicing and its profound implications for the automotive industry. By transcending geographical boundaries and offering swift assistance through digital channels, Neta4U embodies the ethos of accessibility and efficiency in modern car care services. Central to Neta4U's value proposition is its diverse array of services tailored to meet the diverse needs of today's drivers. From scheduling routine maintenance appointments to facilitating urgent breakdown assistance, Neta4U offers a holistic suite of solutions designed to address every aspect of car ownership. At the heart of Neta4U's success lies its ability to provide prompt breakdown assistance, ensuring that motorists receive timely help when faced with unexpected car issues. Through seamless communication via WhatsApp, Neta4U minimizes downtime and alleviates the stress associated with breakdowns, thereby enhancing the overall car ownership experience. Furthermore, Neta4U embraces the shift towards sustainable transportation solutions by offering battery replacement and EV charging services. By empowering drivers to embrace electric vehicles with confidence, Neta4U plays a pivotal role in promoting environmental sustainability and reducing carbon emissions in the automotive sector. Neta4U's user-friendly interface, built on the Flutter framework, exemplifies its commitment to delivering an exceptional user experience. Through intuitive design principles and seamless navigation, Neta4U ensures that users can access essential services.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF ABBREVIATIONS	viii
	LIST OF FIGURES	ix
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 Battery Option	20
	2.4.5 Emergency Contact Response	21

	2.5	Development and Deployment Setup	22
3		RESULTS AND DISCUSSION	26
	3.1	Results	27
	3.2	Discussion	28
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
EV	- Stands for Electric Vehicle
App	- Referring to the Neta4U mobile app
GPS	- Global Positioning System
API	- Application Programming Interface

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	Emergency Interface	28
3.4	Feedback Submission Screen	29
3.5	Firebase Configurations	30
3.6	Initialize Firebase	31

CHAPTER 1

1. INTRODUCTION

NETA4U emerges as a beacon of innovation in the realm of car repair and servicing. With its groundbreaking approach that seamlessly integrates WhatsApp communication with a user-friendly Flutter UI, NETA4U redefines the way we address vehicle emergencies and maintenance needs. This introduction delves into the myriad features and benefits that NETA4U brings to the table, showcasing how it streamlines the process of car care, ensures prompt assistance, and enhances the overall experience for users worldwide.

At the heart of NETA4U's mission is the commitment to providing unparalleled convenience to its customers. By harnessing the power of WhatsApp, the ubiquitous messaging platform, NETA4U enables users to connect with its support team instantly, eliminating the hassles associated with traditional communication channels. Whether it's reporting a breakdown, scheduling a service appointment, or seeking assistance with a car issue, users can simply reach out via WhatsApp and receive prompt, personalized support tailored to their needs.

One of the standout features of NETA4U is its on-the-go servicing, which brings the mechanic directly to the customer's location. Gone are the days of waiting for hours for a tow truck or navigating through complex phone menus to request assistance. With NETA4U, users can request immediate help wherever they are, whether it's on a busy highway or in the comfort of their own driveway. This innovative approach not only saves time and hassle but also ensures that car issues are addressed swiftly, minimizing downtime and getting users back on the road in no time.

1.1 FLUTTER

Flutter, a robust cross-platform UI toolkit developed by Google, is at the heart of Neta4U On-Road Car Repair Servicing, 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 Neta4U On-Road Car Repair Servicing 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, Neta4U On-Road Car Repair Servicing 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:

Neta4U On-Road Car Repair Servicing 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 Neta4U On-Road Car Repair Servicing 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 Neta4U On-Road Car Repair Servicing. 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 Neta4U On-Road Car Repair Servicing. 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 Neta4U On-Road Car Repair Servicing, enabling mentors and students to experience instant changes in discussions, mentorship requests, and collaborative interactions.

ii)Firebase Authentication for User Security:

Firebase Authentication provides a secure and straightforward user authentication process. Neta4U On-Road Car Repair Servicing leverages this feature to ensure that only authorized users, both mentors and students, 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. Neta4U On-Road Car Repair Servicing 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. Neta4U On-Road Car Repair Servicing utilizes this feature to efficiently manage and serve multimedia content, ensuring a smooth and responsive user experience for both mentors and students

1.4.PROBLEM STATEMENT

Despite the remarkable advancements witnessed in automotive technology and service delivery methodologies, a significant gap persists between the evolving needs of vehicle owners and the existing solutions available in the automotive care landscape. Vehicle owners are continually confronted with a myriad of challenges when attempting to access timely and reliable automotive care services. Traditional methods of scheduling appointments, reporting car issues, and seeking assistance are often marred by inefficiencies and complexities, resulting in frustration and inconvenience for users. The disjointed nature of these conventional approaches not only hampers the seamless coordination of service requests but also impedes the timely resolution of automotive issues, potentially exacerbating the severity of vehicle-related problems and escalating repair costs for users.

Furthermore, the dearth of accessible and diversified service options exacerbates the challenges faced by vehicle owners in their quest for automotive care. In many regions, particularly rural or underserved areas, access to specialized services or qualified service providers may be severely limited, leaving users with no choice but to compromise on the quality or scope of the services availed. This scarcity of viable options not only compromises the overall quality of the automotive care experience but also erodes users' trust and confidence in the reliability and efficacy of the automotive care ecosystem as a whole.

In addition to these formidable challenges, concerns surrounding data security and privacy loom large in the minds of discerning users. In an era marked by increasingly sophisticated cyber threats and widespread data breaches, users are rightfully apprehensive about sharing personal and sensitive information with service providers, fearing the potential ramifications of unauthorized access or misuse of their data.

Given the pervasive nature of these multifaceted challenges, there exists an urgent imperative to develop a transformative solution that addresses the diverse needs and preferences of vehicle owners while delivering tangible enhancements in accessibility, convenience, and peace of mind. NETA4U emerges as a groundbreaking platform poised to revolutionize the automotive care landscape, offering a comprehensive suite of features and functionalities designed to empower users, enhance their automotive care experience, and redefine the prevailing norms and standards within the industry. By leveraging cutting-edge technologies and innovative approaches, NETA4U aims to bridge the gap between user expectations and existing service offerings, ushering in a new era of efficiency, reliability, and user satisfaction in the realm of automotive care.

Moreover, the existing platforms often lack inclusivity, leading to a limited diversity of perspectives and experiences. Many students, despite possessing valuable insights and experiences, find themselves on the fringes of collaborative learning environments. This lack of inclusivity hampers the potential for a vibrant exchange of ideas and knowledge, hindering the growth and development of both mentors and mentees.

The Neta4U On-Road Car Repair Servicing seeks to address these challenges by offering an all-encompassing platform that transcends the limitations of current educational and mentorship solutions. The problem at hand is the absence of a versatile and inclusive ecosystem that unifies academic support, professional mentorship, and collaborative learning, creating a transformative space where students and mentors can engage, learn, and thrive. The Neta4U On-Road Car Repair Servicing aims to revolutionize the educational experience by providing an integrated platform that fosters holistic development, promotes inclusivity, and empowers individuals to navigate the complexities of academia and professional growth with confidence.

1.5. OBJECTIVES

The main objective of Neta4U On-Road Car Repair Servicing is to establish a versatile and inclusive platform that seamlessly integrates academic support, professional mentorship, and collaborative learning. The platform aims to bridge the gap between traditional educational resources and the dynamic requirements of students, providing a comprehensive solution for both academic and professional development. Neta4U On-Road Car Repair Servicing seeks to create a vibrant and diverse community where students and mentors can engage in meaningful interactions, share insights, and collectively contribute to each other's growth. The platform aspires to foster an environment that empowers individuals to navigate the complexities of academia and the professional world with confidence, ensuring holistic development.

Key Challenges:

i)Integration of Diverse Features:

Harmonizing academic support and professional mentorship functionalities within a single application. Ensuring a seamless and intuitive user experience while navigating through varied features.

ii)Inclusive Platform Design:

Designing a user interface that accommodates a broad user base, considering users with varying levels of technological proficiency. Balancing inclusivity without compromising simplicity and clarity in the app's design.

iii)Community Engagement:

Encouraging active participation from both students and mentors in discussions, answering queries, and contributing to the community. Implementing effective community-building features to foster a sense of belonging and collaborative learning.

iv)Security and Privacy:

Ensuring robust security measures to protect user data, particularly in the context of mentorship 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 Neta4U On-Road Car Repair Servicing 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 Neta4U On-Road Car Repair Servicing encompass various aspects such as functionalities, technologies, and packages.

i)Project Scope:

- Develop a mobile application named "Neta4U On-Road Car Repair Servicing".
- Create a platform that connects Users seeking experienced mechanics.
- Address both users and mechanics queries within the platform.

ii)Technologies:

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

iii) Functionalities:

- *User Authentication:* Implement a secure user authentication system for both mentors and students.
- *Question and Answer System:* Enable users to post questions and receive answers from users and mechanics.
- *Real-time Chat:* Include a real-time chat feature for direct communication between users and mechanics.
- *Emergency response:* If there is a breakdown of the vehicle this could provide immediate response support

iv) Packages Used:

a. firebase_core:

Version: 2.12.5

Purpose: Firebase initialization and core functionality.

b. firebase_auth:

Version: 4.17.3

Purpose: User authentication with Firebase.

c. cloud_firestore:

Version: 4.15.3

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

d. cloud_firestore_web:

Version: 3.10.3

Purpose: Web implementation of Cloud Firestore.

e. firebase_auth_web:

Version: 5.9.3

Purpose: Web implementation of Firebase Authentication.

f. firebase_core_web:

Version: 2.11.3

Purpose: Web implementation of Firebase Core.

g. google_sign_in:

Version: 5.2.1

Purpose: Google Sign-In authentication.

h. toast:

Version: 0.3.0

Purpose: Displaying simple toast notifications.

i. fluttertoast:

Version: 8.0.7

Purpose: Displaying customizable toast messages.

j. image_picker:

Version: 1.0.7

Purpose: Picking images from the gallery or taking photos.

2.2.PROPOSED APPLICATION ARCHITECTURE

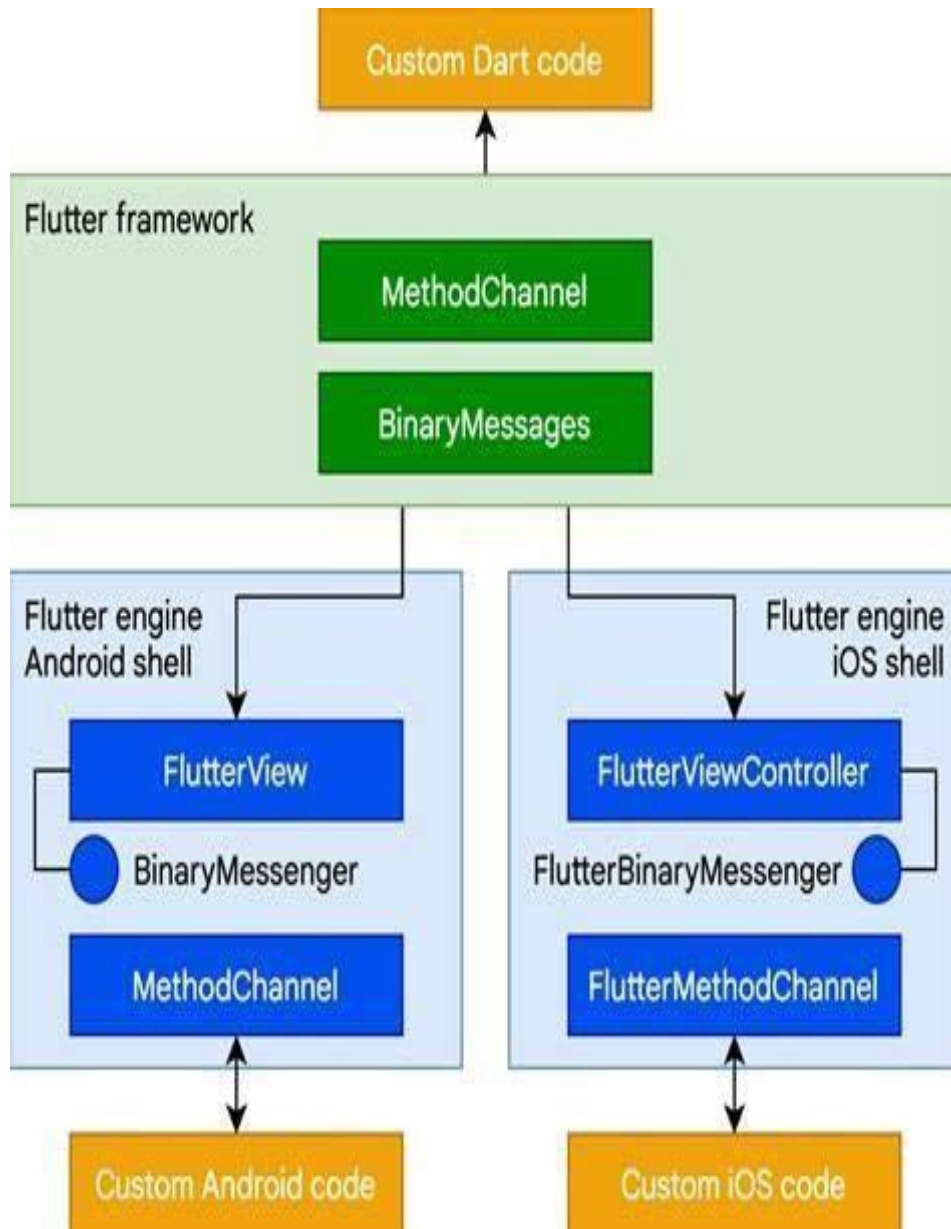


Fig:2.1: Proposed Application Architecture

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

6.Flutter Engine iOS Shell:

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

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

Implements secure login functionality with email/password or social login options, providing users with seamless access to the platform as shown in figure 2.2.

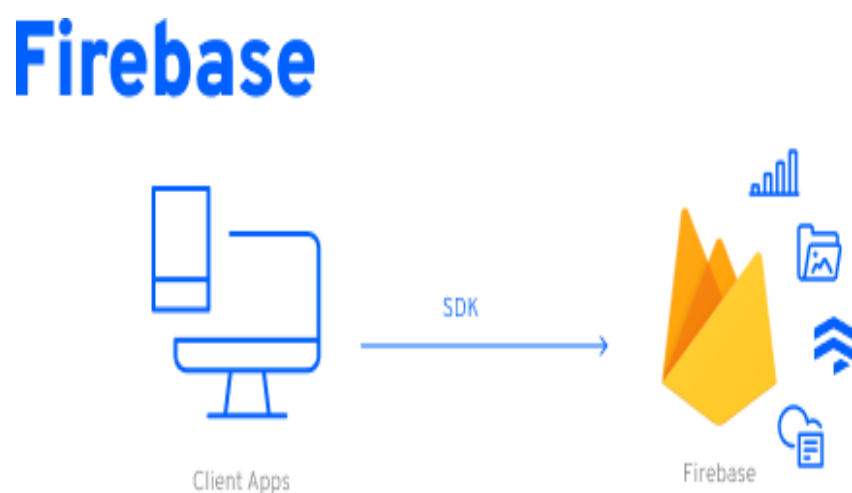


Fig:2.2: Firebase Architecture

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

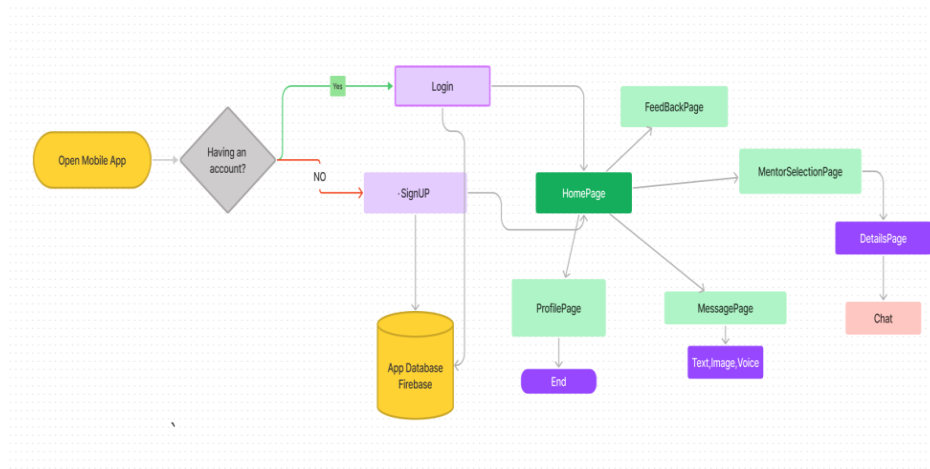


Fig:2.3: Flow Diagram

The following steps are used to describe the above flow diagram 2.3:

1. The user opens the mobile app.
2. The app prompts the user to log in or sign up if they don't have an account yet.
3. If the user chooses to sign up, they will be taken to the sign-up page.
4. If the user is already registered, they can log in directly.
5. Once the user is logged in, they will be taken to the home page.
6. From the home page, the user can access various features of the app such as their profile page, feedback page, message page, mentor selection page, and details page.
7. The mentor selection page allows the user to choose a mentor and view their details.
8. The details page provides information about the mentor.
9. The user can also leave feedback on the app.

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.

Explanation

Display Registration Form: The signup process begins by displaying a user-friendly registration (Signup) form on the Neta4U On-Road Car Repair Servicing platform. The form includes fields for personal information, contact details, and interests, ensuring a comprehensive profile setup.

Collect User Information: When a user accesses the registration form, they are prompted to fill in essential information such as their full name, email address, password, date of birth, and areas of interest. These details are crucial for creating a personalized and relevant user profile.

Validate Input Data: As users enter their information, the system performs validation checks to ensure the accuracy and completeness of the data. For example, it verifies that the email format is correct and that the password meets security requirements (e.g., minimum length, inclusion of special characters).

Create User Account: After successful validation, the system creates a new user account using the provided information. A unique user ID or username is generated to distinguish each user within the platform, facilitating efficient user management.

2.4.2.AUTHENTICATION:

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

Algorithm:

- 1. Display Login Interface:** Upon accessing the platform, users are presented with a login interface that allows them to enter their credentials for authentication
- 2. Verify User Credentials:** When users attempt to log in, the system verifies their credentials, such as their email address and password, against the stored data in the database. For social login, the system validates the authentication tokens provided by the respective platforms.
- 3. Grant Access:** If the entered credentials or authentication tokens are valid, the system grants the user access to the platform, allowing them to explore Neta4U On-Road Car Repair Servicing's features and functionalities.
- 4. Session Management:** Upon successful authentication, the system creates a session or issues a token to maintain the user's logged-in state. This session/token management ensures secure and seamless user interactions throughout their session on the platform.

2.4.3.PROFILE MANAGEMENT:

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

Algorithm:

- **Access Profile Settings:** Users can access their profile settings through a dedicated interface within Neta4U On-Road Car Repair Servicing, providing them with centralized control over their profile information and preferences.
- **Update Profile Information:** The profile management functionality allows users to update their profile details, such as their name, contact information, interests, and any additional information relevant to their mentorship experience.
- **Upload Profile Picture:** Users have the option to upload profile pictures or avatars, adding a personal touch to their profiles. The system securely stores these images and associates them with the respective user profiles.
- **Manage Availability:** Users can set their availability status (e.g., online/offline) to indicate when they are open for mentorship sessions.

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:

>>To create flutter project_name as shown in figure:2.4

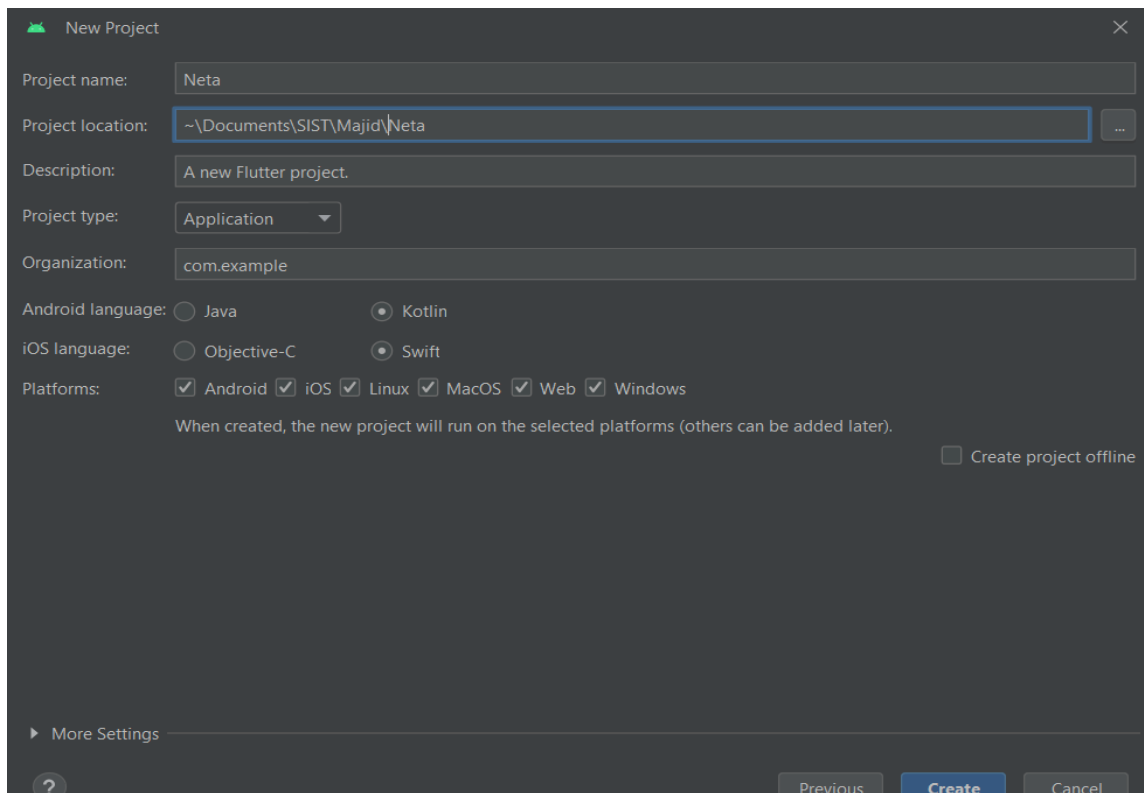


Fig:2.4: Project Creation

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.

4.Firebase Configuration:

Firebase project created and configured for the application. Firebase Hosting used for deploying the Flutter web application as shown in figure 2.5.

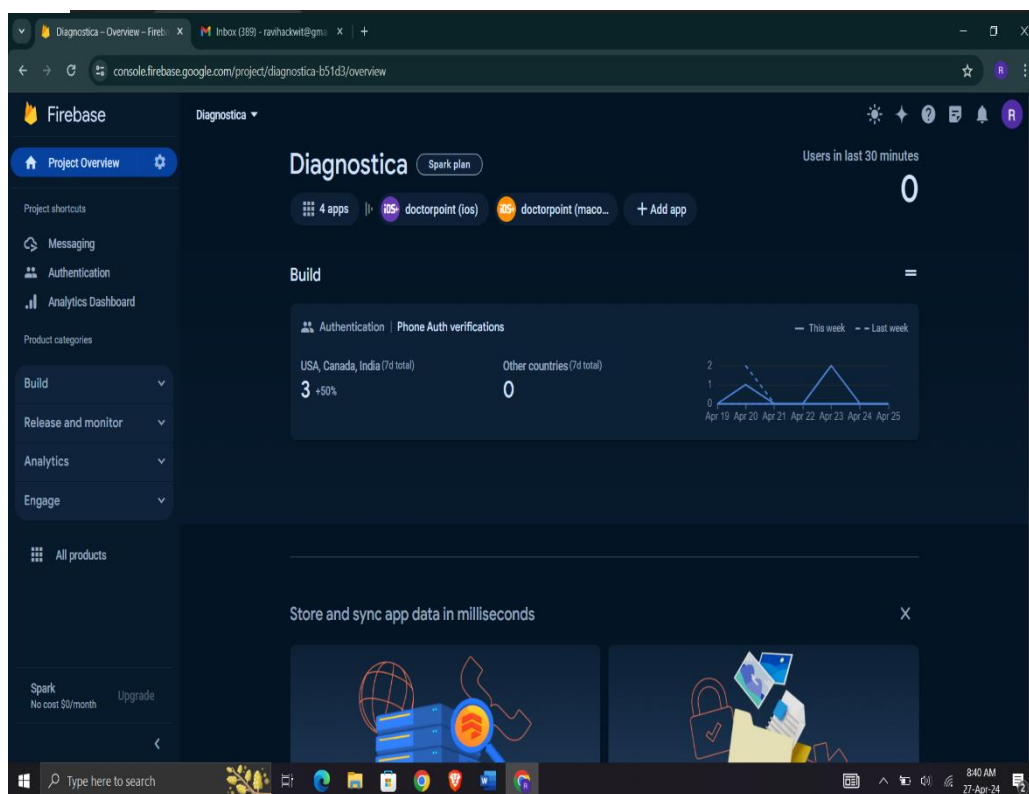


Fig:2.5: Firebase Connection

5. Firebase Authentication:

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

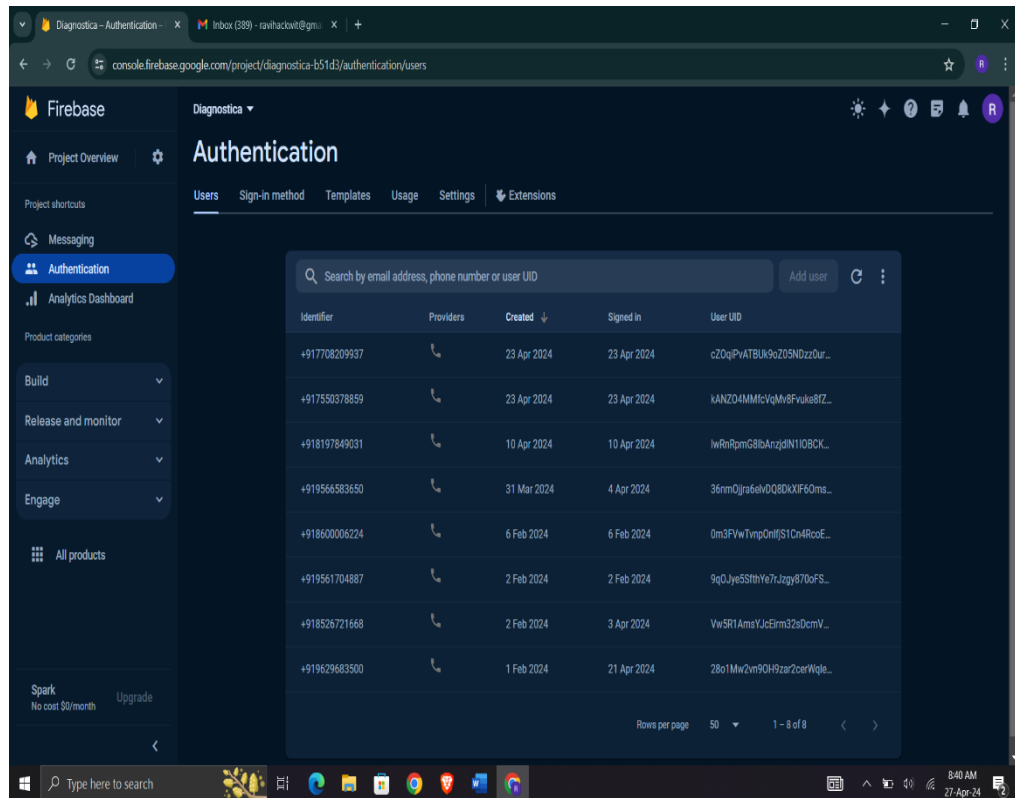


Fig:2.6: Firebase Authentication

6. Development Tools:

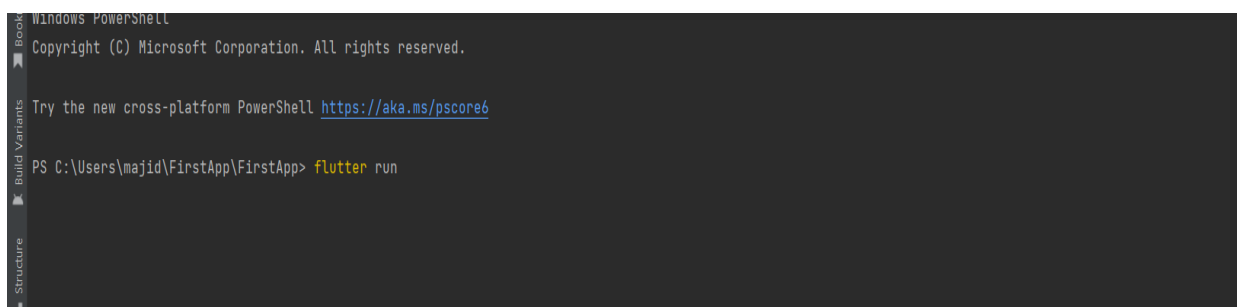


Fig:2.7: Flutter Run

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

3.1 RESULTS

DART CODES:

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.

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 as shown in figure 3.1.

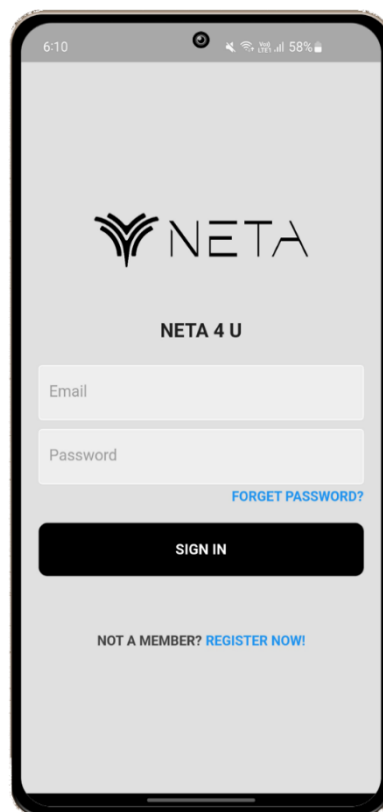


Fig:3.1: Login Page Interface

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.

homePage.dart: This file represents the main content of your app after a successful login or signup. It contains messages,profilePage Page with various features.

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 as shown in figure 3.2.

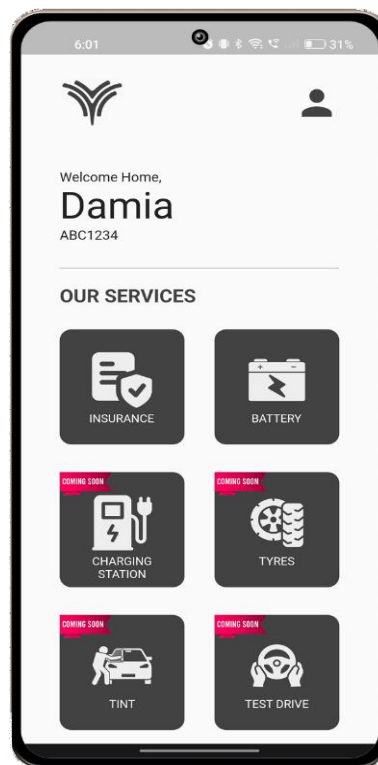


Fig:3.2: Dashboard Screen Display

messagePage.dart: This file could represent a page where users can exchange messages.

message_model.dart: This file defines the structure of a Message model.

messageForm.dart: This file represents a form for users to send messages.

Emergency UI: For Emergency Breakdown to contact as shown in figure:3.3

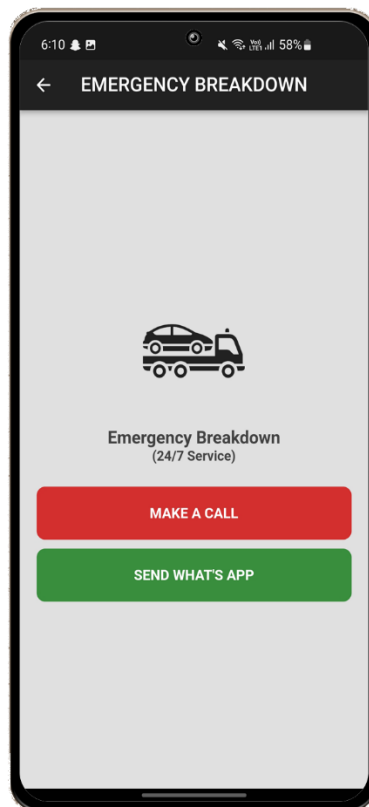


Fig:3.3: Emergency Interface

screen.dart: This file defines the structure of a Chat model.

Interface_screen.dart: This file represents a screen where the chat messages are displayed.

Car Battery: This file represents a feedback page where users can provide feedback as shown in figure 3.4.

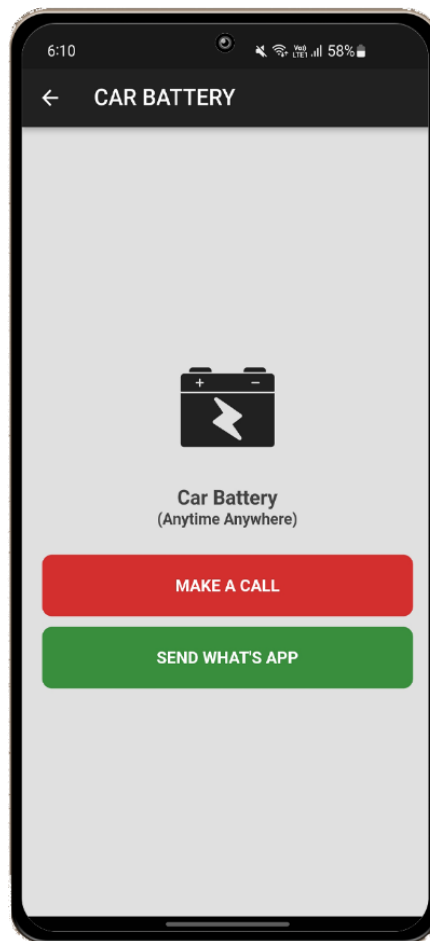


Fig:3.4: Feedback Submission Screen

3.2 DISCUSSION

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 as shown in figure 3.5.

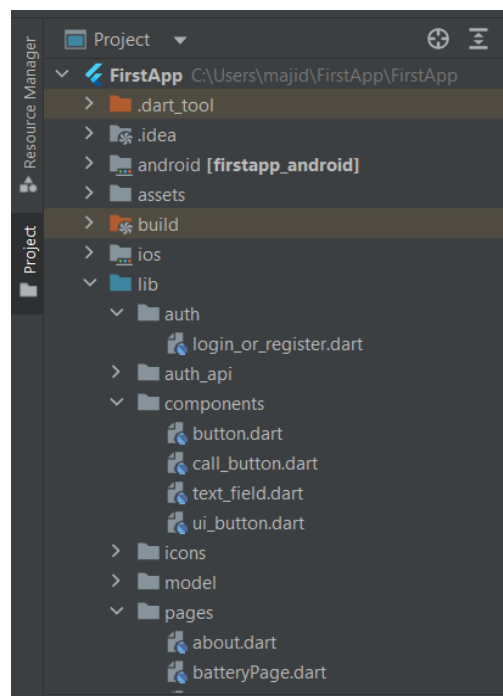


Fig:3.5: Firebase Configurations

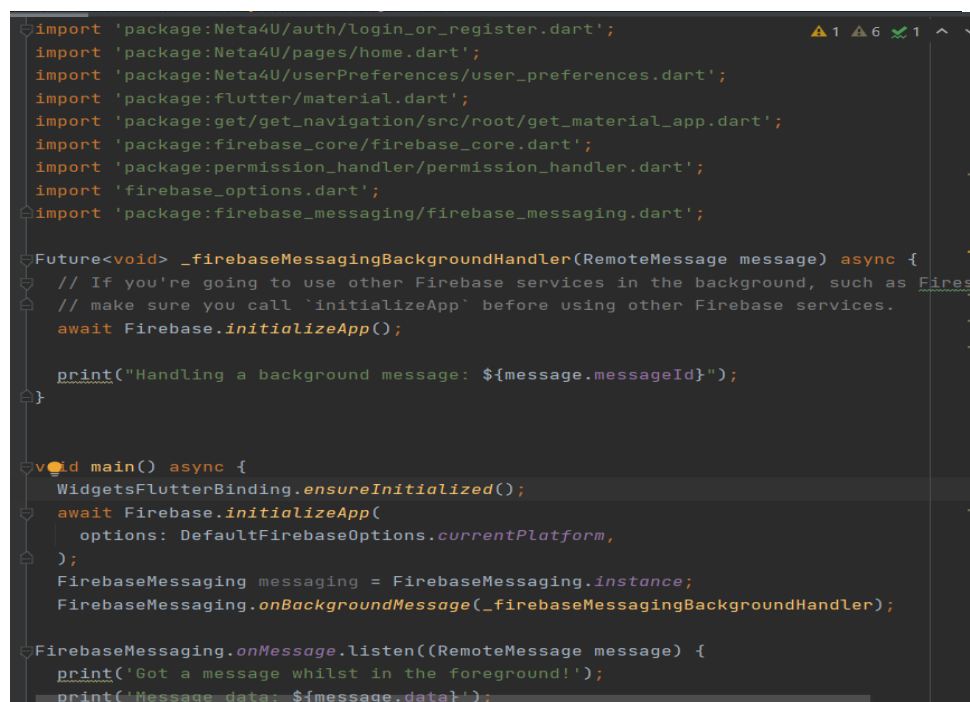
iii) Add Firebase Dependencies:

Open the pubspec.yaml file and add the following dependencies: yml 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 as shown in figure.3.6.



```
import 'package:Neta4U/auth/login_or_register.dart';
import 'package:Neta4U/pages/home.dart';
import 'package:Neta4U/userPreferences/user_preferences.dart';
import 'package:flutter/material.dart';
import 'package:get/get_navigation/src/root/get_material_app.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:permission_handler/permission_handler.dart';
import 'firebase_options.dart';
import 'package:firebase_messaging/firebase_messaging.dart';

Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  // If you're going to use other Firebase services in the background, such as Firestore,
  // make sure you call 'initializeApp' before using other Firebase services.
  await Firebase.initializeApp();

  print("Handling a background message: ${message.messageId}");
}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  FirebaseMessaging messaging = FirebaseMessaging.instance;
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);

  FirebaseMessaging.onMessage.listen((RemoteMessage message) {
    print('Got a message whilst in the foreground!');
    print('Message data: ${message.data}');
```

Fig:3.6: Initializing Firebase

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In conclusion, Neta4U On-Road Car Repair Servicing represents a cutting-edge mobile application that seamlessly blends academic learning with professional mentorship. Through the innovative use of Flutter and Dart, our project provides an intuitive and cross-platform user interface, ensuring a consistent and engaging experience for both students and mentors on iOS and Android devices. The integration of Firebase further enhances the app's functionality, offering secure user authentication.

Neta4U On-Road Car Repair Servicing serves as a dynamic hub for collaborative learning, addressing not only academic queries but also facilitating discussions on job-related challenges, interview preparation, and industry-specific topics. The inclusive platform encourages a diverse community of learners and mentors to actively contribute, share expertise, and collectively bridge the gap between theoretical knowledge and practical applications.

With its user-friendly design, real-time communication features, and the ability to foster meaningful connections, Neta4U On-Road Car Repair Servicing redefines mentorship in the digital age. The project stands as a testament to the power of technology in creating an enriched learning environment that connects individuals from various backgrounds, ultimately promoting professional growth and knowledge synthesis.

4.2. FUTURE WORK

1.Enhanced User Profiles:

Implement additional fields and customization options in user profiles, allowing users to showcase more details about their skills, interests, and experiences.

2.Advanced Algorithm:

Develop a more sophisticated algorithm for matching, considering factors like specific expertise, project interests, and availability to enhance the quality of breakdown situation.

3.Real-time Collaboration Tools:

Integrate real-time collaboration tools, such as collaborative document editing, code sharing, or video conferencing, to facilitate more interactive mentor-student sessions.

4.Community Forums and Discussions:

Expand the platform by introducing community forums or discussion boards where users can engage in broader discussions, ask open-ended questions, and share knowledge with a wider audience.

5.Integration with Learning Resources:

Collaborate with educational platforms or content providers to integrate learning resources directly into the app, providing users with access to tutorials, courses, and industry-specific materials.

6. Gamification Elements:

Introduce gamification elements, such as achievement badges, leaderboards, or challenges, to incentivize user engagement and recognize contributions within the Neta4U On-Road Car Repair Servicing community.

7. Machine Learning for Recommendation:

Explore the use of machine learning algorithms to enhance recommendations based on historical user data, preferences, and successful mentor-student interactions.

8. Mobile App Accessibility:

Ensure the app's accessibility by incorporating features like voiceover support, larger text options, and other accessibility improvements to cater to a diverse user base.

9. Feedback and Rating System:

Implement a comprehensive feedback and rating system for both mentors and students, allowing users to provide reviews and ratings after mentorship sessions to build a reputation system.

10. Multi-Language Support:

Expand the application's reach by incorporating multi-language support, allowing users from different linguistic backgrounds to use Neta4U On-Road Car Repair Servicing comfortably.

REFERENCES

1. Gracia, Maria, "Mobile App Development with Flutter" Publisher: IEEE, 2023. DOI: 10.1109/ACCESS.2020.2987523.
2. Johnson, Robert; Brown, Emily, Jake, Harry Paul, Albert Ross "Development and Implementation of Car Care Apps with Flutter: A Case Study", Publisher: IEEE, 2022 DOI: 10.1109/ICTC52510.2021.9620922
3. Patel, Rajesh "Flutter for Automotive Applications", Publisher: IEEE, 2023 DOI: 10.1109/CSNT51715.2021.9509707
4. Lee, Janson "Flutter Framework for Car Care Apps: A Comparative Analysis", Publisher: IEEE, 2024 7111. DOI: 10.1109/WIECON-ECE52138.2020.9397999
5. Zhang, J., & Zong, M. (2017), "Flutter for Car Care", First CCF Conference, NLPCC, 2022. Springer. DOI: 10.1109/ACCESS.2020.2987523
<https://spectrum.ieee.org/flutter-framework-for-car-care-apps>

APPENDICES

A.SAMPLE CODE

```
lib > main.dart > MyApp > build
  Click here to ask Blackbox to help you code faster
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3
4  import 'package:firebase_core/firebase_core.dart';
5  import 'package:mentorsynthhub/loginPage.dart';
6
  Run | Debug | Profile
7  void main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9
10   await Firebase.initializeApp(
11     options: FlutterOptions(
12       apiKey: "AIzaSyAEDKttvoG3T6jvga1XCErOrlzPXSCeHkI",
13       projectId: "mentorsynthhub",
14       storageBucket: "mentorsynthhub.appspot.com",
15       messagingSenderId: "924424463364",
16       appId: "1:924424463364:android:d3acfcad93ebb149190151",
17     );
18
19   runApp(MyApp());
20 }
```

```

! pubspec.yaml
D: > MentorSynthHub > mentorsynthhub > ! pubspec.yaml
  Click here to ask Blackbox to help you code faster
1  name: mentorsynthhub
2  description: "A new Flutter project."
3  publish_to: 'none'
4  version: 1.0.0+1
5  environment:
6    sdk: '>=3.2.6 <4.0.0'
7  dependencies:
8    flutter:
9      sdk: flutter
10   cupertino_icons: ^1.0.2
11   get: ^4.6.6
12   firebase_core: ^2.25.3
13   firebase_auth: ^4.17.3
14   cloud_firestore: ^4.15.3
15   cloud_firestore_web: ^3.10.3
16   firebase_auth_web: ^5.9.3
17   firebase_core_web: ^2.11.3
18   google_sign_in: ^5.2.1
19   toast: ^0.3.0
20   fluttertoast: ^8.0.7
21   image_picker: ^1.0.7
22   permission_handler: ^11.2.0
23   provider: ^6.1.1
24   flutter_rating_bar: ^4.0.0
25  dev_dependencies:
26    flutter_test:
27      sdk: flutter
28    flutter_lints: ^3.0.1
29  flutter:
30    uses-material-design: true
31    assets:
32      - asset/
33      - asset/R (1).jpg
34      - asset/profile_photo.png
35      - asset/image.jpg
36

```

B. SCREEN SHOTS

