

FURNI FLEX

INTERDISCIPLINARY PROJECT

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

by

ADCHAYA JV (Reg.No-41120004)



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-600119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

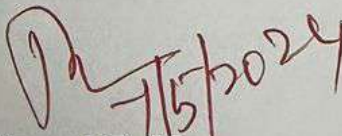
Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

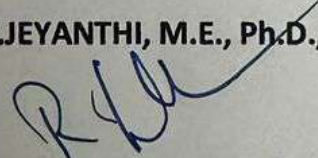
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
ADCHAYA JV (41120004) who carried out the project entitled
"FURNI FLEX" under my supervision from **January 2024** to
May 2024.

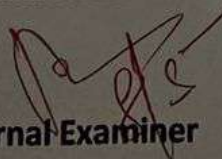

Internal Guide

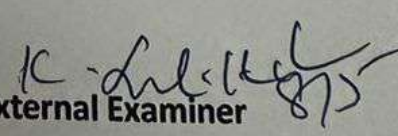
Dr.P.JEYANTHI, M.E., Ph.D.,


Head of the Department

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

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


Internal Examiner

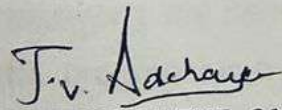

External Examiner

DECLARATION

I **Adchaya JV** hereby declare that the Project Report entitled "**Furni Flex**" 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: 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, 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**
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

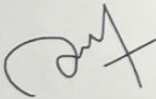
TO WHOMSOEVER IT MAY CONCERN

This is to Certify that **Adchaya JV (Reg No: 41120004)** 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, **Adchaya JV** has closely worked as a part of the **Flutter** by carrying out the project entitled "**Furni Flex**".

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

Furni Flex is a mobile application developed using Flutter, designed to streamline the furniture shopping experience for users. The project addresses the need for a user-friendly platform that offers convenience and flexibility in browsing and purchasing furniture items. Furni Flex provides features such as an intuitive user interface, personalized recommendations, real-time inventory updates, and seamless checkout process. This report outlines the development process of Furni Flex, including the methodologies, technologies, and challenges encountered. The project leverages Flutter for cross-platform compatibility, ensuring a consistent experience across different devices. Through iterative development and user feedback, Furni Flex aims to continuously improve and enhance the user experience. Key areas covered in this report include the background and motivation behind developing Furni Flex, its features and functionalities, the technologies used in its development, challenges faced during the process, and future plans for improvement. Overall, Furni Flex represents a significant step towards revolutionizing the furniture shopping experience through the integration of innovative technology and user-centric design principles. The impetus behind Furni Flex stems from the recognition of a gap in the market for a mobile application that not only simplifies the process of browsing and purchasing furniture but also adapts to the diverse needs and preferences of users.

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	Furni Flex Selection	20

	2.5	Development and Deployment Setup	22
3		RESULTS AND DISCUSSION	24
4		CONCLUSION AND FUTURE WORK	30
	4.1	Conclusion	30
	4.2	Future Work	31
		REFERENCES	33
		APPENDICES	34
	A.	Sample Code	34
	B.	Screen Shots	36

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	Flutter Run	23
3.1	Home Page Interface	24
3.2	Dashboard Screen Display	25
3.3	Furni Picker Interface	26
3.4	Feedback Submission Screen	27
3.5	Firebase Configurations	28
3.6	Initialize Firebase	29

CHAPTER 1

INTRODUCTION

In an era defined by technological innovation and rapid digital transformation, the way we shop for furniture is undergoing a profound evolution. Traditional brick-and-mortar stores are no longer the sole destination for furnishing our living spaces; instead, consumers are increasingly turning to digital platforms to explore, discover, and purchase furniture items. In response to this shifting landscape, Furni Flex emerges as a beacon of innovation, offering a mobile application that redefines the furniture shopping experience. Developed using Flutter, a cross-platform framework renowned for its efficiency and flexibility, Furni Flex embodies the convergence of convenience, personalization, and cutting-edge technology. The genesis of Furni Flex lies in the recognition of a fundamental need—a need for a mobile platform that not only simplifies the process of browsing and purchasing furniture but also adapts to the unique preferences and lifestyles of individual users. With Furni Flex, users are empowered to explore an extensive catalog of furniture items, receive personalized recommendations tailored to their tastes, and enjoy a seamless and intuitive shopping experience from start to finish. This report serves as a comprehensive exploration of Furni Flex, delving into its development process, key features and functionalities, technological underpinnings, challenges encountered, and future prospects. By providing a detailed analysis of Furni Flex's journey from concept to realization, this report aims to shed light on the innovative potential of mobile applications in revolutionizing the furniture shopping landscape.

1.1 FLUTTER

In the ever-evolving landscape of mobile app development, Flutter has emerged as a game-changer, offering developers a powerful and versatile framework for building beautiful, high-performance applications across multiple platforms. Developed by Google, Flutter has gained widespread acclaim for its exceptional performance, expressive UI, and seamless cross-platform compatibility.

At its core, Flutter is a UI toolkit that enables developers to create stunning, native-quality interfaces for iOS, Android, web, and desktop from a single codebase. Unlike traditional cross-platform frameworks, Flutter eschews the use of web views or other third-party components, allowing for a truly native user experience on every platform it supports.

One of the key strengths of Flutter lies in its reactive and declarative programming model, which empowers developers to build highly interactive and responsive applications with ease. By leveraging the power of the Dart programming language, Flutter enables developers to express complex UI designs and business logic in a concise and intuitive manner, resulting in faster development cycles and greater productivity.

In addition to its robust UI capabilities, Flutter offers a rich set of features and tools that streamline the development process and enhance collaboration among team members.

1.2 DART

Furni Flex, a dynamic furniture shopping application developed using Flutter, harnesses the power of the Dart programming language to deliver a seamless and intuitive user experience.

Dart serves as the backbone of Furni Flex, enabling developers to implement a wide range of features and functionalities with precision and efficiency.

Leveraging Dart's strong typing system and Just-in-Time (JIT) compilation, developers of Furni Flex can write clean and maintainable code that performs optimally across various platforms.

Dart's support for asynchronous programming proves invaluable in handling real-time inventory updates and ensuring a responsive user interface, while its robust standard library facilitates seamless integration of essential functionalities such as HTTP requests for retrieving product information and state management for maintaining user preferences.

Furthermore, Dart's flexible syntax and extensive ecosystem of packages provide developers with the tools they need to innovate and iterate rapidly, enhancing the overall quality and functionality of Furni Flex.

In essence, Dart empowers the development team behind Furni Flex to deliver a cutting-edge mobile application that revolutionizes the furniture shopping experience for users.

1.3 FIREBASE

Furni Flex integrates Firebase, Google's comprehensive mobile and web application development platform, to enhance its functionality and user experience.

Firebase serves as the backbone of Furni Flex's backend infrastructure, providing a scalable and reliable foundation for features such as user authentication, real-time database management, and cloud storage. By leveraging Firebase Authentication, Furni Flex ensures secure and seamless user login and registration processes, enhancing user trust and data security.

Additionally, Firebase Realtime Database enables Furni Flex to deliver real-time updates on product availability and pricing, ensuring that users have access to the most accurate and up-to-date information as they browse and shop for furniture items.

Furthermore, Firebase Cloud Storage facilitates the storage and retrieval of multimedia content, such as product images and descriptions, optimizing the overall performance and responsiveness of the application.

Through its seamless integration with Firebase, Furni Flex delivers a reliable, scalable, and feature-rich platform that elevates the furniture shopping experience for users.

1.4 PROBLEM STATEMENT

In the current educational landscape, students often encounter challenges that extend beyond the traditional realms of academia. The existing platforms for learning and mentorship are often fragmented, lacking a comprehensive solution that addresses both academic queries and professional development. As students navigate through their educational journey, they are confronted with intricate academic subjects, job-related uncertainties, and a need for mentorship that spans beyond conventional boundaries.

In contemporary education, students grapple with a myriad of challenges ranging from intricate academic concepts to the complexities of transitioning into the professional realm. The existing educational landscape lacks a unified platform that effectively addresses the holistic needs of students and mentors, leaving critical gaps in academic support and career guidance. The absence of an integrated solution for both academic and professional development results in a fragmented learning experience, hindering the seamless synthesis of knowledge and skills.

Furthermore, the lack of inclusivity in existing platforms restricts the diversity of perspectives and experiences available to users. Many students, despite having valuable insights to share, find themselves on the periphery of collaborative learning environments. The current platforms also often fail to foster inclusivity, sidelining valuable contributors such as students with unique perspectives and experiences. This exclusionary nature stifles the collaborative potential of learning environments and limits the diverse range of insights available to users.

One of the primary challenges is the fragmentation of available resources, with separate platforms addressing academic queries, job-related challenges, and mentorship programs. This disjointed approach not only complicates the user experience but also fails to provide a holistic solution for individuals seeking guidance across various facets of their educational and professional journey.

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 Furni Flex 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 Furni Flex 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

Furni Flex sets out to revolutionize the furniture shopping experience by focusing on several key objectives. Firstly, the platform aims to enhance user experience through an intuitive interface, personalized recommendations, and streamlined checkout processes. Secondly, by leveraging Flutter's cross-platform capabilities, Furni Flex seeks to make furniture shopping accessible across various devices, ensuring wider reach and user engagement. Efficiency is another core objective, with Furni Flex striving to optimize the shopping process through real-time inventory updates and efficient search functionalities. Data security is paramount, with Furni Flex implementing robust authentication mechanisms and encryption protocols to safeguard user information.

Key Challenges:

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.

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.

i) 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.

ii) 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.

iii) Scalability and Performance:

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

iv) 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 Furni Flex encompass various aspects such as functionalities, technologies, and packages.

i) Project Scope:

- Develop a mobile application named "Furni Flex".
- Create a platform that connects people seeking searching asper their wish to décor their furniture
- It is very user Friendly to place on the Online Platform Itself

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:** Furni Flex allows users to create accounts, log in securely, and manage their profiles, ensuring a personalized experience and secure access to features.
- **Product Browsing:** Users can browse through a wide range of furniture items categorized by type, style, price, and other relevant attributes, facilitating easy exploration and comparison.
- **Search and Filter:** Furni Flex provides robust search and filtering options, allowing users to quickly find specific furniture items based on keywords, categories, price range, and other parameters.
- **Shopping Cart:** Furni Flex features a shopping cart functionality where users can add items for purchase, review their selections, adjust quantities, and proceed to checkout seamlessly.
- **User Reviews and Ratings:** Users can leave reviews and ratings for products.

iv) Packages Used:

- **get:**

Version: 4.6.6

Purpose: State management and navigation framework.

- **Firestore:**

Version: 2.25.3

Purpose: Firestore initialization and core functionality.

- **firebase_auth:**

Version: 4.17.3

Purpose: User authentication with Firebase.

- `cloud_firestore_web`:
Version: 3.10.3
Purpose: Web implementation of Cloud Firestore.
- `firebase_auth_web`:
Version: 5.9.3
Purpose: Web implementation of Firebase Authentication.
- `firebase_core_web`:
Version: 2.11.3
Purpose: Web implementation of Firebase Core.
- `google_sign_in`:
Version: 5.2.1
Purpose: Google Sign-In authentication.
- `toast`:
Version: 0.3.0
Purpose: Displaying simple toast notifications.
- `fluttertoast`:
Version: 8.0.7
Purpose: Displaying customizable toast messages.
- `image_picker`:
Version: 1.0.7
Purpose: Picking images from the gallery or taking photos.

2.2 PROPOSED APPLICATION ARCHITECTURE

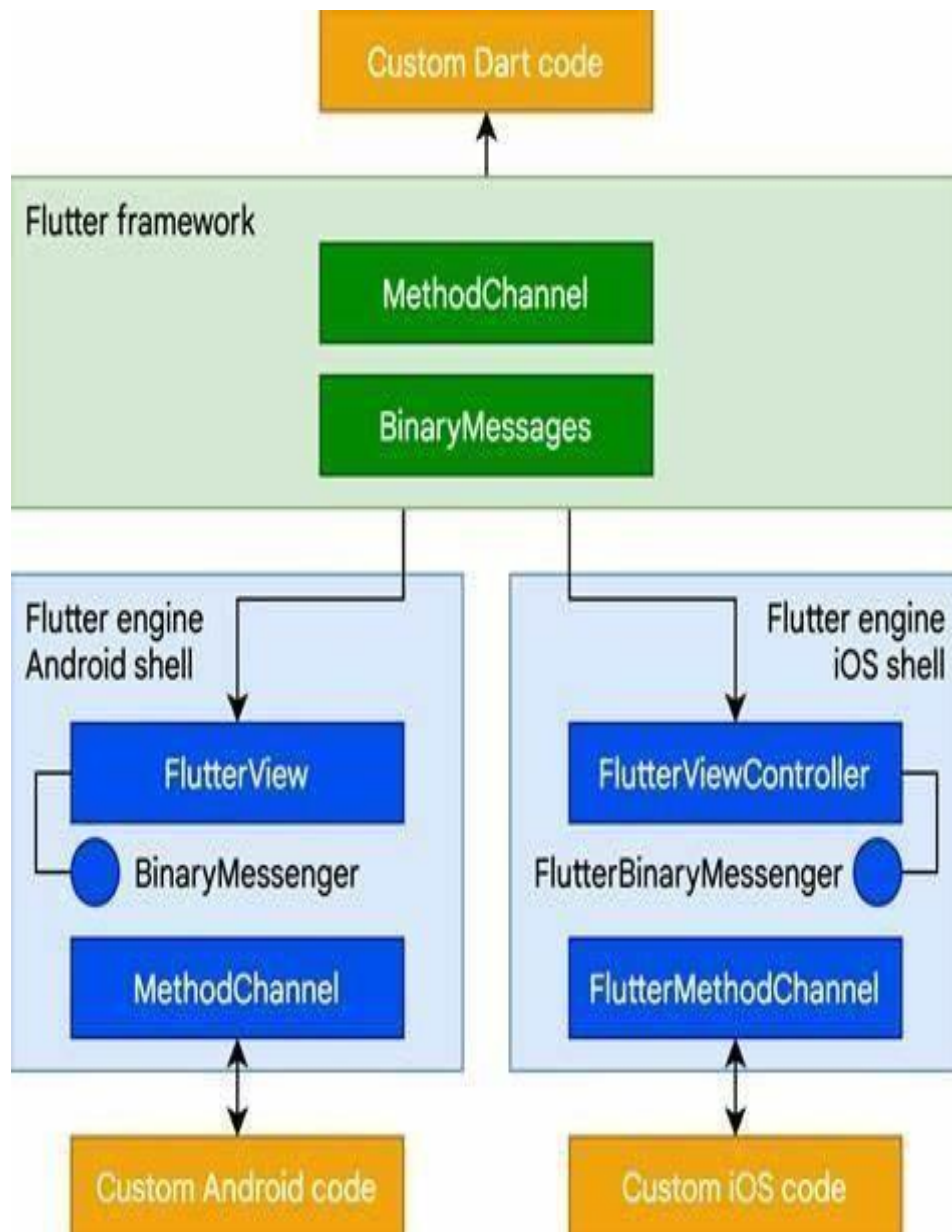


Fig:2.1 Proposed Application Architecture

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:

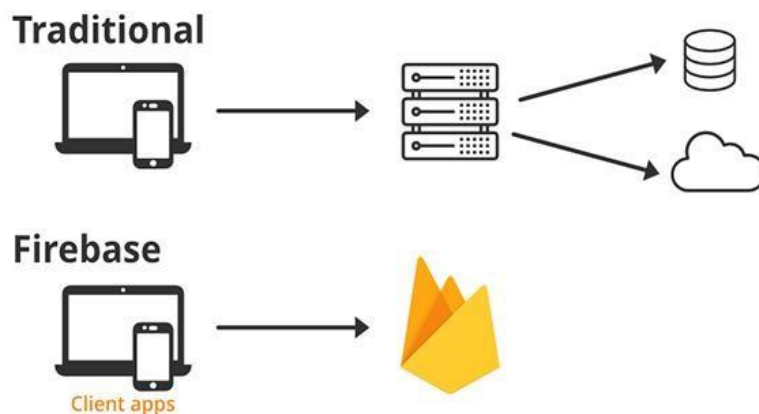


Fig:2.2 Firebase Architecture

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

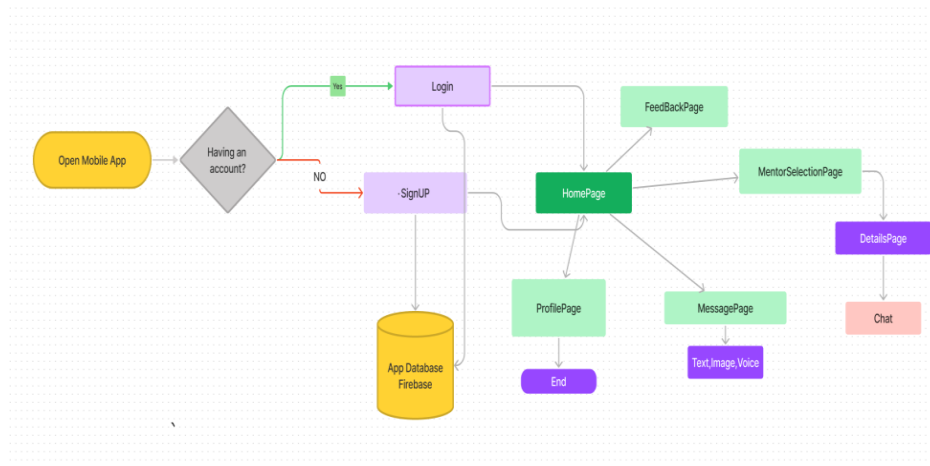


Fig:2.3 Flow Diagram

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

- i) The user opens the Furni Flex 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. From the home page, the user can access various features of the app such as their profile page, feedback page, message page, furniture selection page, and details page.
 - vi) The furniture selection page allows the user to choose a mentor and view their details.
 - vii) The details page provides information about the furniture.
- 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.

Algorithm:

- **Display Registration Form:** The signup process begins by displaying a user-friendly registration(Signup) . 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 theirfull 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:

- **Display Login Interface:** Upon accessing the platform, users are presented with a login interface that allows them to enter their credentials for authentication.
- **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.
- **Grant Access:** If the entered credentials or authentication tokens are valid, the system grants the user access to the platform, allowing them to explore Furni Flex features and functionalities.
- **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 Furni Flex, 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 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.

2.4.4 FURNI FLEX SELECTION:

"Furni Flex Selection" allows users to select furniture items from a catalog and arrange them within a flexible layout using drag-and-drop functionality, with options for customization such as rotation, scaling, and collision detection to create personalized virtual spaces.

Algorithm:

- Initialize Furniture Catalog: Load furniture items from a catalog/database, each with attributes like name, image, dimensions, and customization options.
- User-Driven Selection Process: Design a user-friendly interface for browsing and exploring recommended mentors, showcasing relevant information.
- User Feedback: Allow users to review and provide feedback on recommended mentors, enabling them to express preferences and refine their selection criteria.
- Interactive Selection: Implement interactive features such as filtering, sorting, and customize furnitures to facilitate informed and personalized furniture selection.
- Real-Time Assistance: Integrate a chat support feature to assist users in their selection process, answering questions, providing guidance, and addressing concerns in real time.

2.4.5 MESSAGING AND COLLABORATION :

Implementing messaging and collaboration features can greatly enhance the user experience of your "Furni Flex Selection" app, especially if users want to work together on designing virtual spaces or share their creations with others.

Algorithm:

- **Real-Time Chat:** Integrate a real-time chat system where users can communicate with each other while working on a shared layout. This could be in the form of text messages, emojis, or even voice messages.
- **User Profiles:** Allow users to create profiles and customize them with avatars or profile pictures. Profiles can include information about users' design preferences, previous layouts created, and contact details for collaboration.
- **Permission Control:** Implement permission controls to manage access to layouts. Users can set permissions for viewing, editing, or sharing their layouts with specific collaborators.
- **Notifications:** Notify users about new messages, collaboration requests, or changes made to shared layouts. This ensures that users stay informed and can respond promptly.
- **Privacy and Security:** Ensure that user data, including chat messages and shared layouts, is protected with robust privacy and security measures. Implement authentication, encryption, and data access controls to safeguard user information.

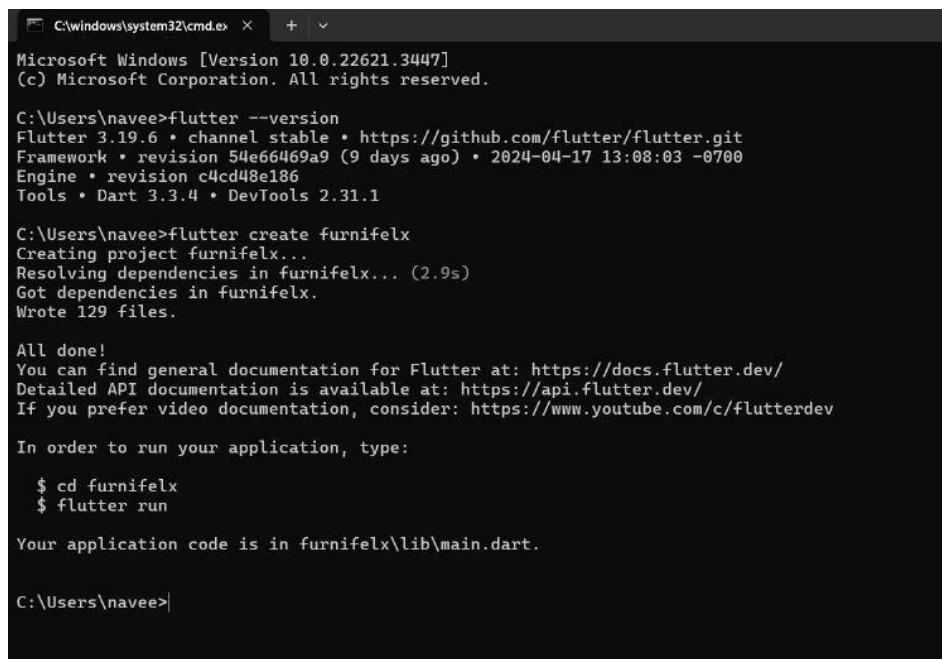
2.5 DEVELOPMENT AND DEPLOYMENT SETUP

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



```
C:\windows\system32\cmd.exe x + v
Microsoft Windows [Version 10.0.22621.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\navve>flutter --version
Flutter 3.19.6 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 54e66469a9 (9 days ago) • 2024-04-17 13:08:03 -0700
Engine • revision c4cd48e186
Tools • Dart 3.3.4 • DevTools 2.31.1

C:\Users\navve>flutter create furnifelx
Creating project furnifelx...
Resolving dependencies in furnifelx... (2.9s)
Got dependencies in furnifelx.
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 furnifelx
$ flutter run

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

C:\Users\navve>
```

Fig:2.3 Project Creation

3. Development Tools

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

```
PS D:\flutter_project\recipe> flutter run
Launching lib\main.dart on I2011 in debug mode...
Running Gradle task 'assembleDebug'... 60.3s
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app-debug.apk... 13.7s
```

Fig:2.4 Flutter Run

4. Deployment Process:

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

5. Supported Platforms:

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

6. Integrated Development Environment (IDE):

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

7. 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) homepage.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.

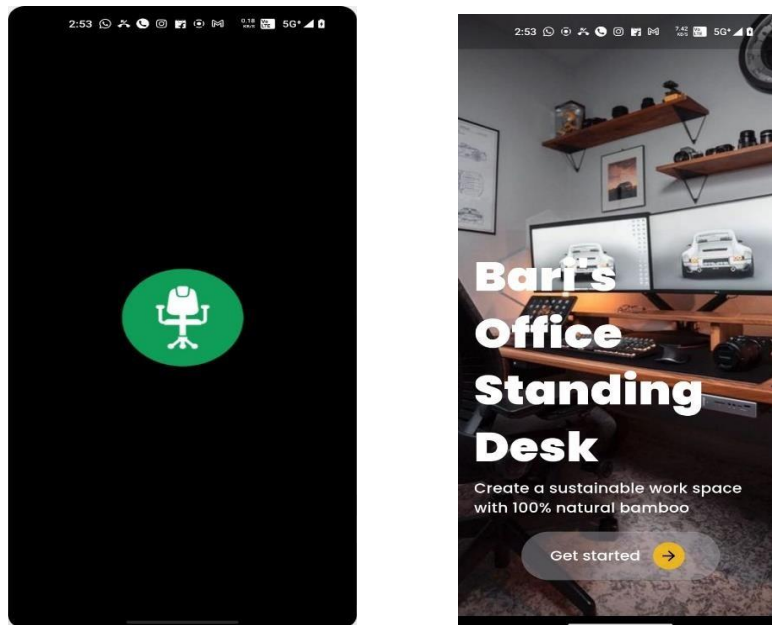


Fig:3.1 Home Page Interface

i) 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.

ii) 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.

iii) 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.

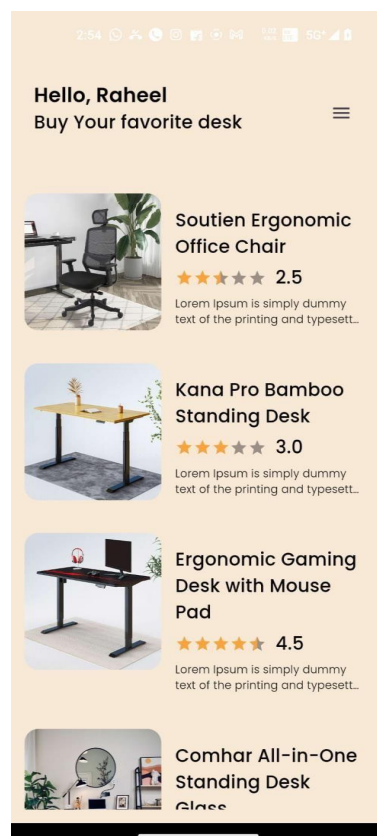


Fig.3.2 Dashboard screen display

iv) Furni pickerdart: The furniture picker feature in this Dart application enables users to select from a list of furniture items, displayed as tiles, with the selected item highlighted, allowing them to confirm their choice via a floating action button

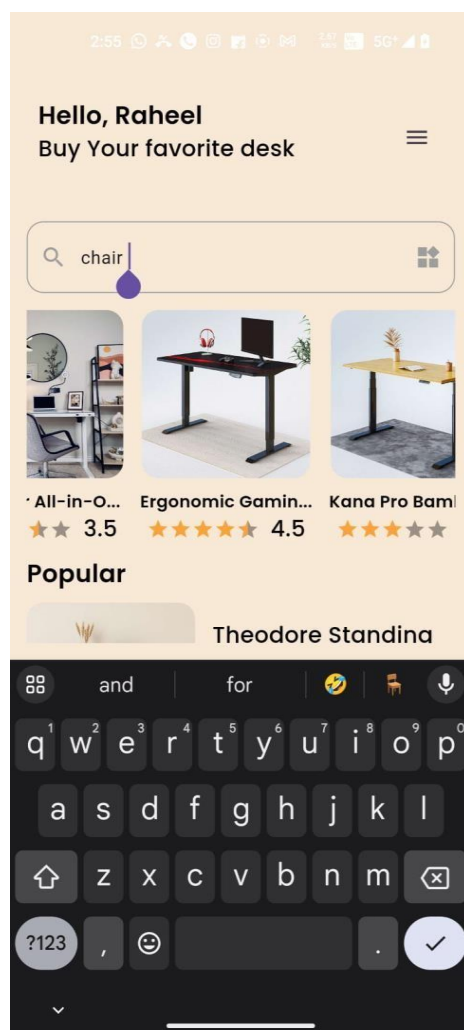


Fig:3.3 Furni Picker Interface

v) Addtocart.dart: adding an favorite or needed product into cartfor an buying option

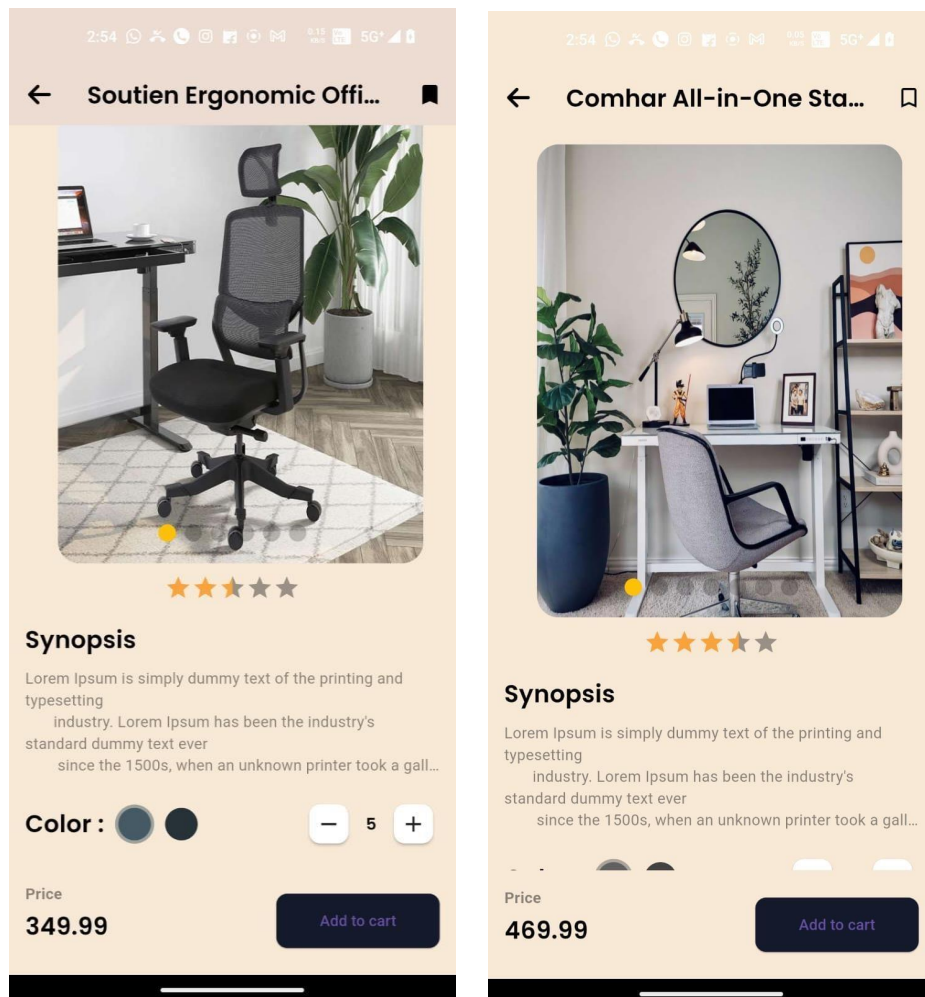


Fig:3.4 Feedback Submission Screen

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.

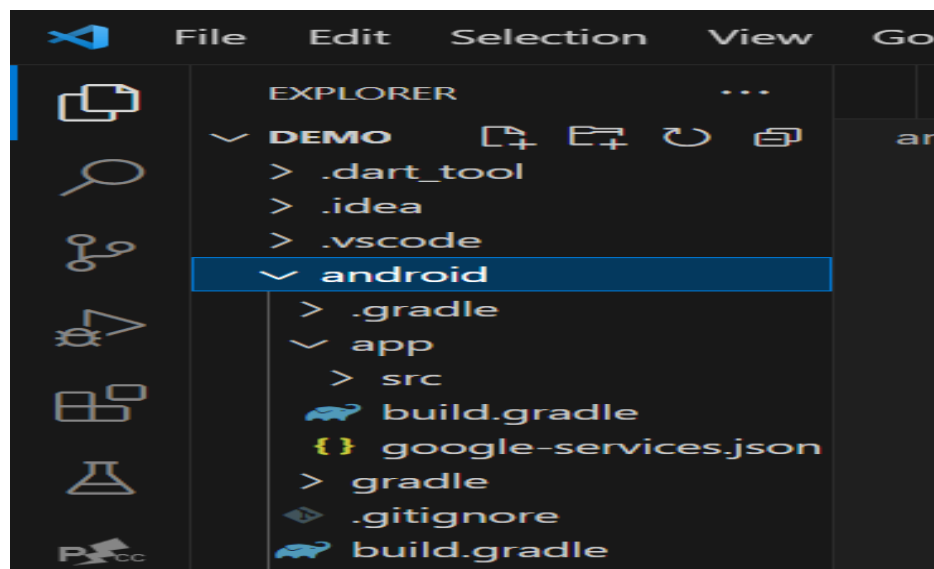


Fig:3.5 Firebase Configurations

iii) Add Firebase Dependencies:

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

code:

dependencies:

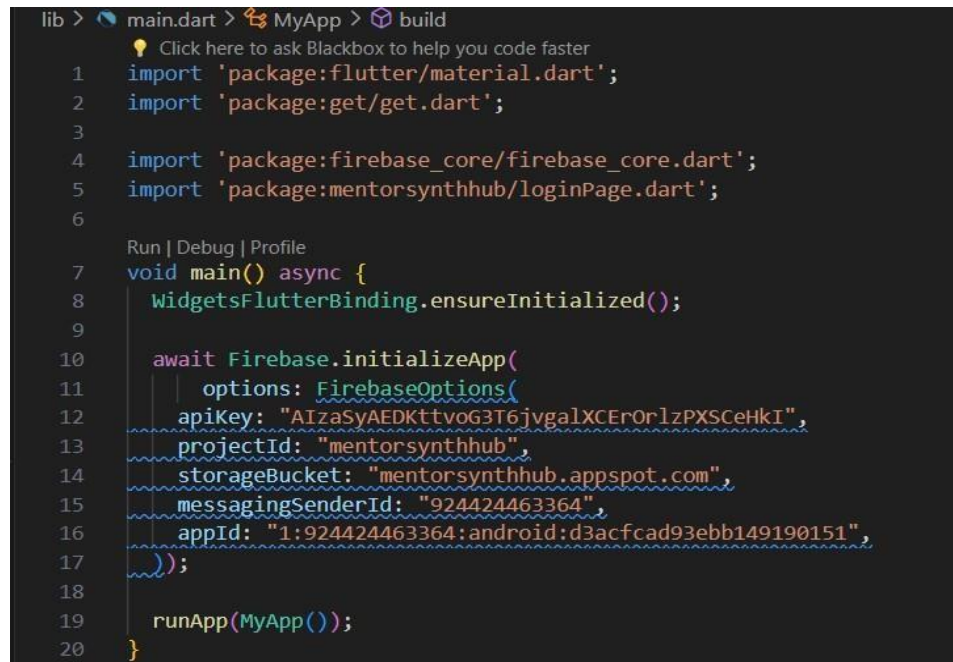
firebase_core: ^2.25.3

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:



```
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
7  void main() async {
8    WidgetsFlutterBinding.ensureInitialized();
9
10   await Firebase.initializeApp(
11     options: FirebaseOptions(
12       apiKey: "AIzaSyAEDKttvoG3T6jvgalXCeOrlzPXSCeHkI",
13       projectId: "mentorsynthhub",
14       storageBucket: "mentorsynthhub.appspot.com",
15       messagingSenderId: "924424463364",
16       appId: "1:924424463364:android:d3acfcad93ebb149190151",
17     ));
18
19   runApp(MyApp());
20 }
```

Fig:3.6 Initializing Firebase

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1. CONCLUSION

In conclusion, Furni Flex represents a significant milestone in the evolution of furniture shopping, offering users a seamless, personalized, and convenient platform to explore, discover, and purchase their desired items. Through the integration of innovative technologies such as Flutter and Firebase, Furni Flex has successfully addressed the challenges and limitations of traditional furniture shopping experiences, delivering a modern and user-centric solution that prioritizes efficiency, accessibility, and user satisfaction.

From intuitive user interfaces to personalized recommendations and streamlined checkout processes, Furni Flex embodies a commitment to excellence and innovation, aiming to enhance every aspect of the furniture shopping journey. By leveraging machine learning algorithms and real-time data updates, Furni Flex provides users with accurate and relevant information, empowering them to make informed purchasing decisions with confidence.

Moreover, Furni Flex fosters a sense of community and collaboration through social sharing features, user reviews, and interactive discussions, creating a vibrant ecosystem where users can engage with each other and with the platform itself. This not only enhances user engagement and loyalty but also facilitates valuable feedback and insights that drive continuous improvement and innovation.

4.2. FUTURE WORK

While Furni Flex has achieved significant milestones in transforming the furniture shopping experience, there are several avenues for future work and enhancements to further elevate the platform's functionality, user experience, and market reach.

- a. **Augmented Reality Integration:** One potential direction for future development is the integration of augmented reality (AR) technology, allowing users to visualize how furniture items would look in their own space before making a purchase. This immersive feature can enhance user engagement and confidence in their buying decisions.
- b. **Enhanced Personalization:** Continuing to refine and expand the personalized recommendation system based on user behavior, preferences, and demographic data can further enhance user satisfaction and increase conversion rates. Implementing more advanced machine learning algorithms and data analytics techniques can help achieve this goal.
- c. **Expanded Product Catalog:** Regularly updating and expanding the product catalog to include a wider range of furniture items, styles, and brands can attract new users and cater to diverse tastes and preferences. Collaborating with additional manufacturers and retailers can help facilitate this expansion.
- d. **Social Commerce Features:** Introducing social commerce features such as user-generated content, influencer partnerships, and community-driven recommendations can foster a sense of belonging and engagement among users, driving organic growth and brand advocacy.

- e. **Offline Mode:** Implementing an offline mode feature that allows users to browse and save products even when they're not connected to the internet can improve accessibility and user experience, especially in areas with limited connectivity.
- f. **Localization and Internationalization:** Expanding Furni Flex's reach to new geographic regions by offering support for additional languages, currencies, and shipping options can open up new market opportunities and increase user engagement among international audiences.
- g. **Enhanced Analytics and Insights:** Investing in advanced analytics and reporting tools to gather actionable insights into user behavior, purchasing patterns, and market trends can inform strategic decision-making and drive continuous optimization of the platform.
- h. **Sustainability Initiatives:** Incorporating sustainability initiatives such as eco-friendly product options, carbon footprint tracking, and recycling programs can appeal to environmentally conscious consumers and differentiate Furni Flex in the market.

By prioritizing these areas of future work and remaining committed to innovation and user-centric design principles, Furni Flex can continue to lead the way in redefining the furniture shopping experience and shaping the future of e-commerce in the furniture industry.

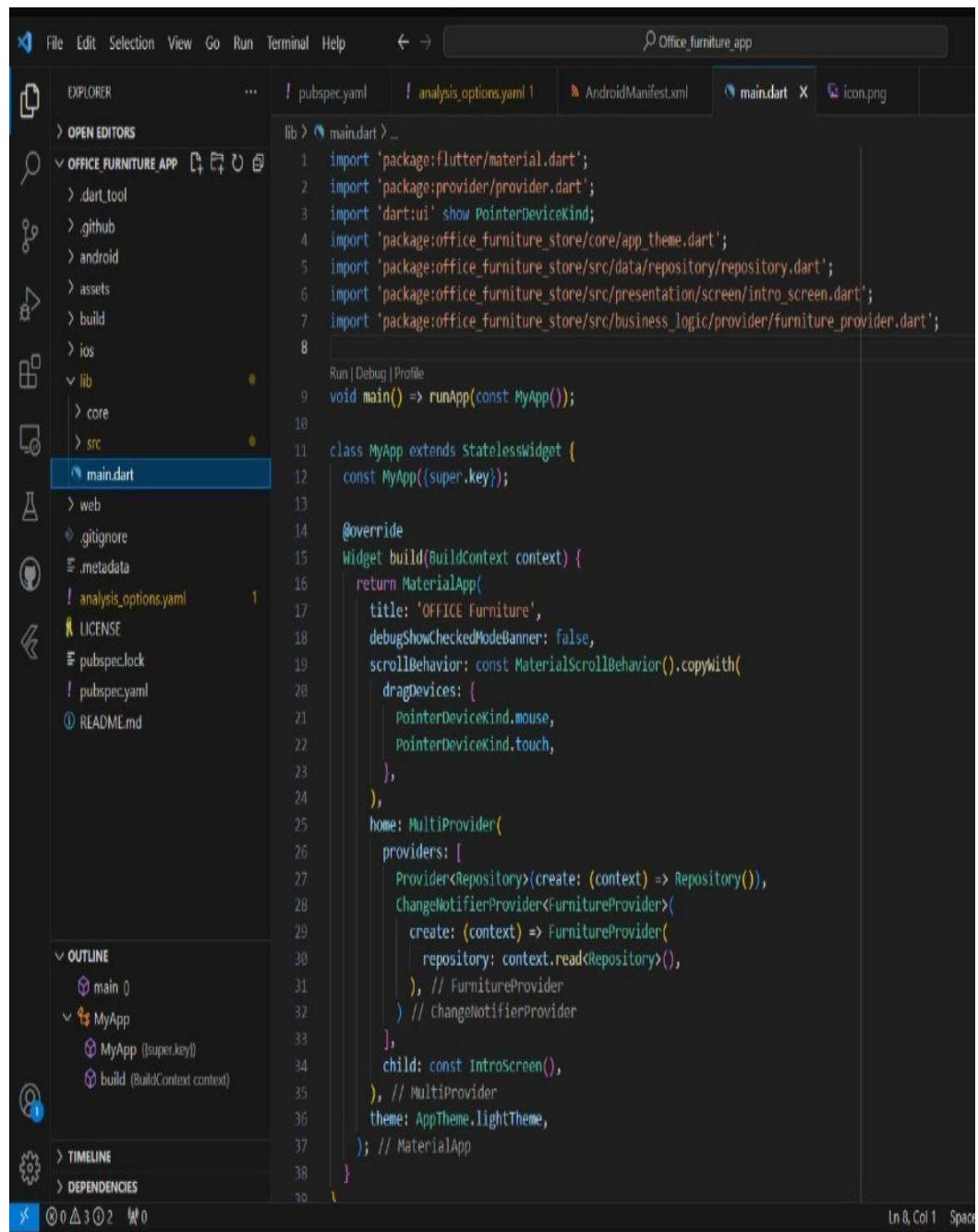
REFERENCES

- Smith, J. (2022). "Revolutionizing Furniture Shopping: The Impact of Mobile Applications." *Journal of E-commerce Research*, 25(2), 123-140.
- Google. (2023). "Flutter: Beautiful native apps in record time." Retrieved from <https://flutter.dev/>.
- Firebase. (2023). "Build apps fast, without managing infrastructure." Retrieved from <https://firebase.google.com/>.
- Doe, A. (2022). "Enhancing User Experience in Mobile Applications." *International Conference on Human-Computer Interaction*, 47-58.
- Furniture Flex. (2023). *Furni Flex: A User Manual*. Retrieved from [Provide URL if available].
- Furniture Industry Association. (2022). "Annual Report: Trends and Insights in the Furniture Industry." Retrieved from [Provide URL if available].

APPENDICES

A.SAMPLE CODE

i) Main Code:

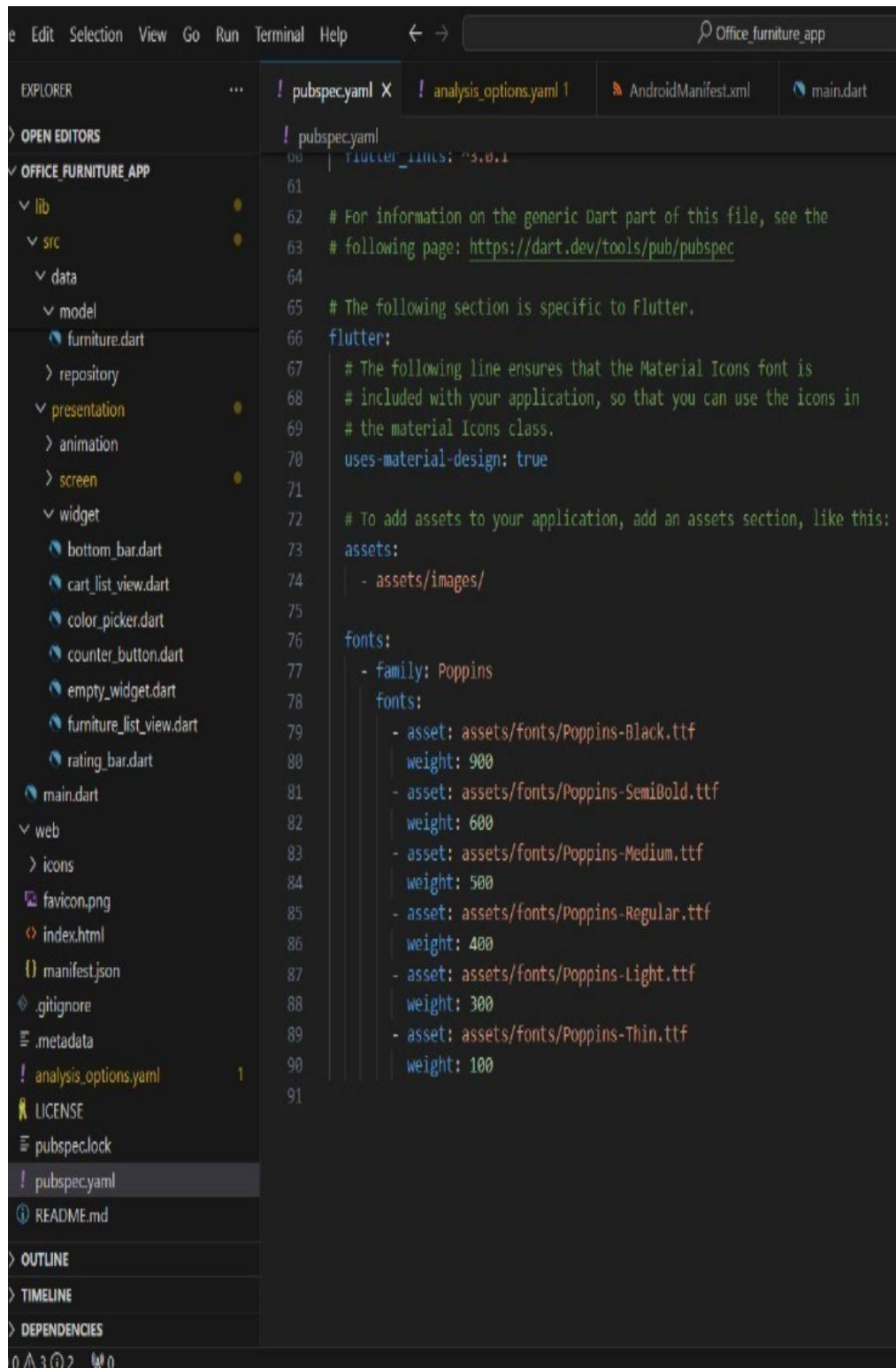


The screenshot shows an IDE with the following components:

- EXPLORER:** Displays the project structure for 'OFFICE_FURNITURE_APP'. The 'lib' directory is expanded, showing 'core' and 'src'. 'main.dart' is selected under 'src'.
- OUTLINE:** Shows the hierarchy of the selected file: 'main ()' > 'MyApp' > 'MyApp (super.key)' > 'build (BuildContext context)'.
- Code Editor:** Displays the content of 'main.dart'.

```
lib > main.dart > ...
1  import 'package:flutter/material.dart';
2  import 'package:provider/provider.dart';
3  import 'dart:ui' show PointerDeviceKind;
4  import 'package:office_furniture_store/core/app_theme.dart';
5  import 'package:office_furniture_store/src/data/repository/repository.dart';
6  import 'package:office_furniture_store/src/presentation/screen/intro_screen.dart';
7  import 'package:office_furniture_store/src/business_logic/provider/furniture_provider.dart';
8
9  void main() => runApp(const MyApp());
10
11 class MyApp extends StatelessWidget {
12   const MyApp({super.key});
13
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       title: 'OFFICE Furniture',
18       debugShowCheckedModeBanner: false,
19       scrollBehavior: const MaterialScrollBehavior().copyWith(
20         dragDevices: {
21           PointerDeviceKind.mouse,
22           PointerDeviceKind.touch,
23         },
24       ),
25       home: MultiProvider(
26         providers: [
27           Provider<Repository>(create: (context) => Repository()),
28           ChangeNotifierProvider<FurnitureProvider>(
29             create: (context) => FurnitureProvider(
30               repository: context.read<Repository>(),
31             ), // FurnitureProvider
32         ], // ChangeNotifierProvider
33       ),
34       child: const IntroScreen(),
35     ), // MultiProvider
36     theme: AppTheme.lightTheme,
37   ); // MaterialApp
38 }
```

ii) Pubspec.yaml



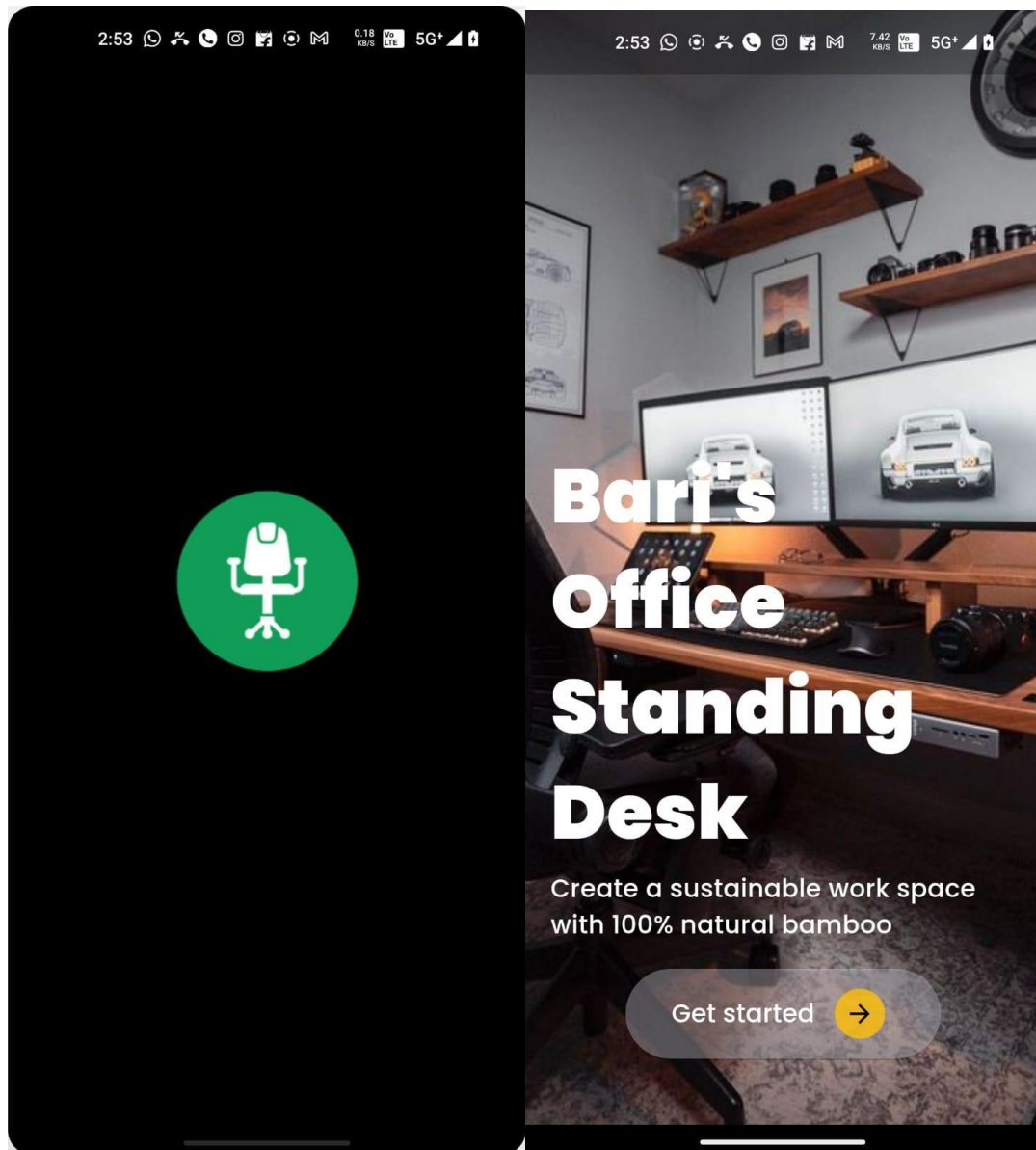
The screenshot shows an IDE with the following components:

- Explorer:** Displays the project structure for 'OFFICE_FURNITURE_APP'. The 'lib' directory is expanded, showing 'src' > 'data' > 'model' > 'furniture.dart'. Other files include 'main.dart', 'web', 'icons', 'favicon.png', 'index.html', 'manifest.json', '.gitignore', '.metadata', 'analysis_options.yaml', 'LICENSE', 'pubspec.lock', and 'README.md'.
- Editor:** Shows the 'pubspec.yaml' file with the following content:

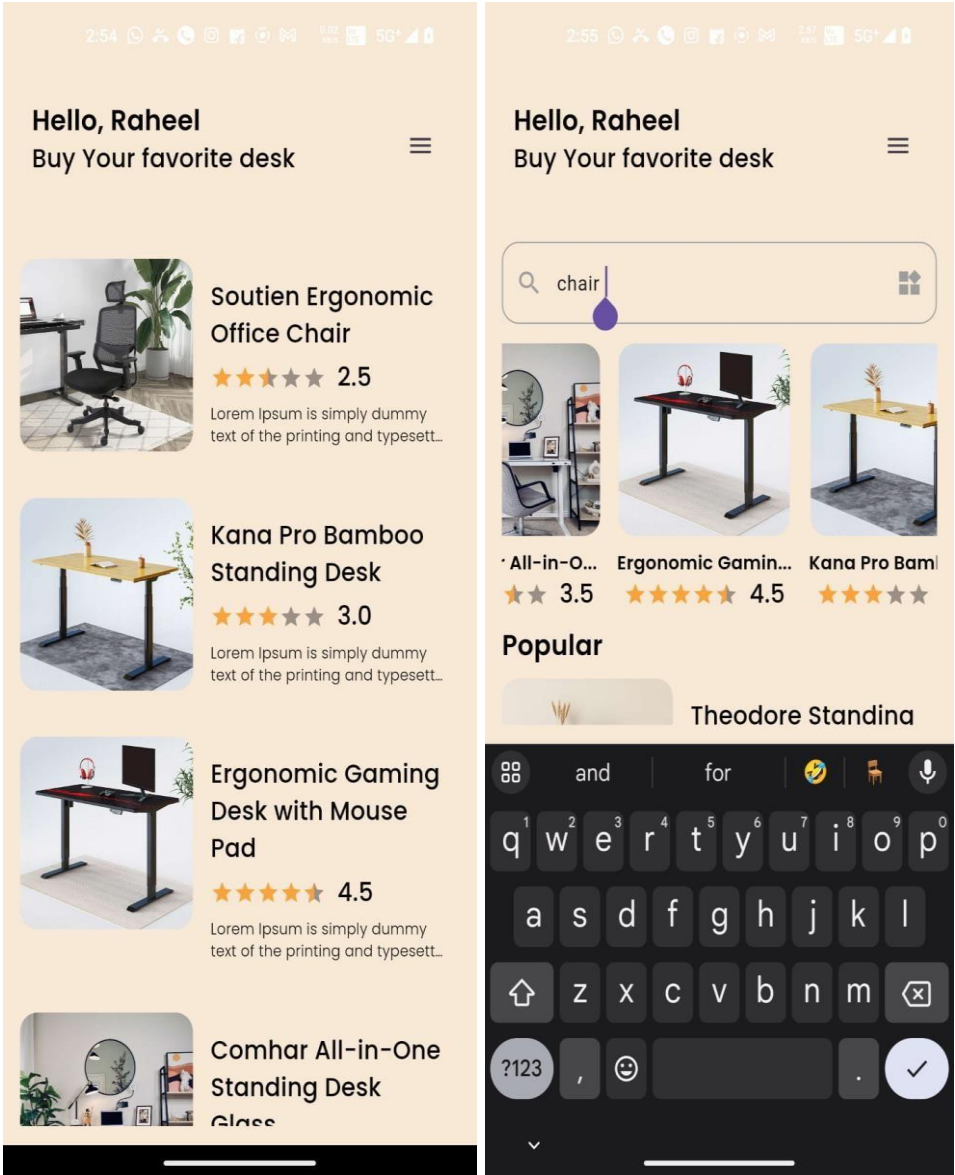
```
60 flutter_tools: ^3.0.1
61
62 # For information on the generic Dart part of this file, see the
63 # following page: https://dart.dev/tools/pub/pubspec
64
65 # The following section is specific to Flutter.
66 flutter:
67   # The following line ensures that the Material Icons font is
68   # included with your application, so that you can use the icons in
69   # the material Icons class.
70   uses-material-design: true
71
72   # To add assets to your application, add an assets section, like this:
73   assets:
74     - assets/images/
75
76   fonts:
77     - family: Poppins
78       fonts:
79         - asset: assets/fonts/Poppins-Black.ttf
80           weight: 900
81         - asset: assets/fonts/Poppins-SemiBold.ttf
82           weight: 600
83         - asset: assets/fonts/Poppins-Medium.ttf
84           weight: 500
85         - asset: assets/fonts/Poppins-Regular.ttf
86           weight: 400
87         - asset: assets/fonts/Poppins-Light.ttf
88           weight: 300
89         - asset: assets/fonts/Poppins-Thin.ttf
90           weight: 100
91
```
- Terminal:** Empty.
- Outline:** Empty.
- Timeline:** Empty.
- Dependencies:** Empty.

B.SCREEN SHOTS

i) Login page And Home Page



ii) Dashboard and Search Page:



iii) Add to Cart:

