PROMINENT

A Personalized Mobile Solution For Real-Time Progress Tracking & Monitoring

Guided By: Dr. Mohamad Nor Bin Hassan Hassan

Prepared By:

Name	Yong Chun Hao
Matric No.	S65505
Program	Bachelor of Computer Science (Software Engineering) with Honours
Course	Framework-Based Mobile Application Development

JANUARY 1, 2024

YONG CHUN HAO

yongchunhao2003@gamil.com

Table of Contents

1 Project Background

ABC Technologies, a leading project management solution provider, offers a robust Flutter-based mobile application for efficient project tracking and management. Users, while unable to create accounts independently, are provided access by ABC Technologies to ensure a controlled and secure environment. The app seamlessly integrates with Firebase services, leveraging Cloud Firestore for storing and retrieving project and activity data.

Key features include a dynamic project list that updates in real-time, displaying projects based on user roles. Admin users, identified by email containing "admin," have access to a comprehensive view of all projects, while non-admin users see only projects associated with their account. The application employs the FutureBuilder widget for asynchronous operations, enhancing performance and responsiveness.

PROMinenT serves as a testament to ABC Technologies' commitment to excellence, offering a centralized platform for real-time project oversight. Its user-friendly interface fosters seamless communication, eliminating barriers and facilitating collaboration among team members and stakeholders. Tailored for the evolving work environment, PROMinenT ensures accessibility from any location, meeting the demands of remote work and global collaboration.

Embracing advanced security measures, PROMinenT prioritizes the protection of sensitive project data through robust encryption and secure user authentication. It stands as an embodiment of efficiency, reliability, and security—a transformative force in the realm of project management.

2 Use Case

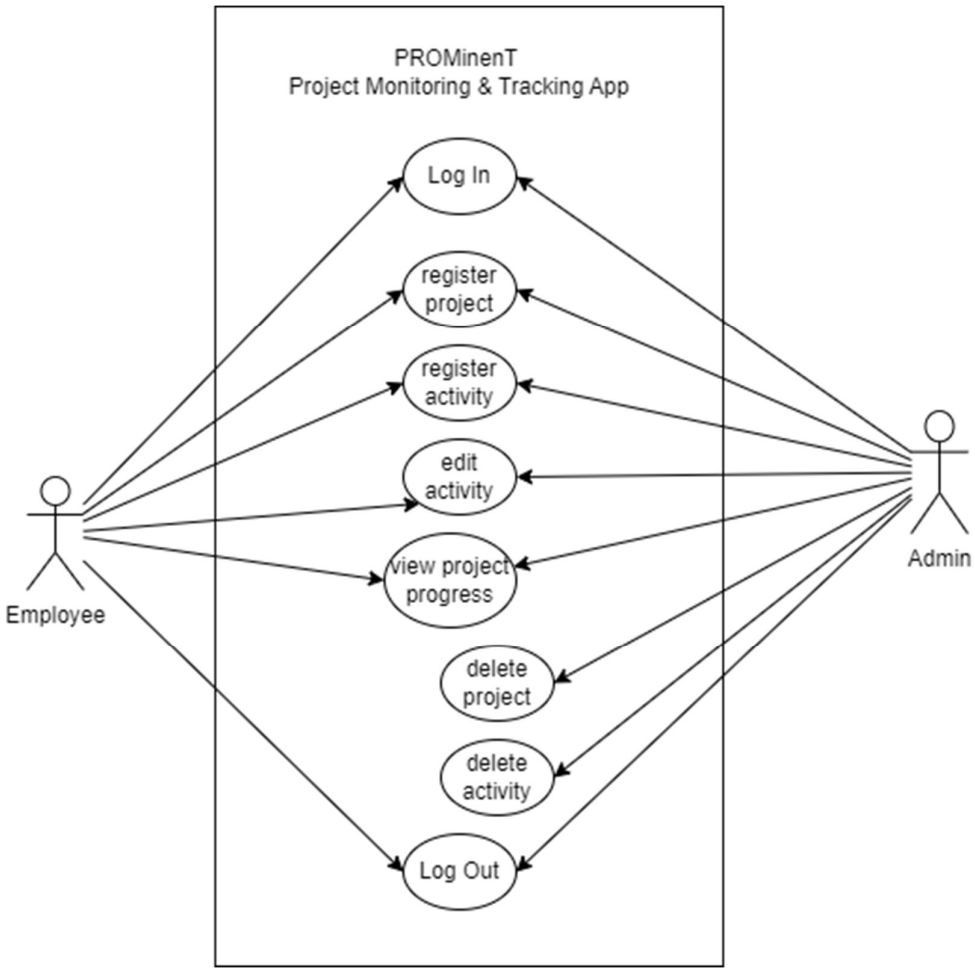
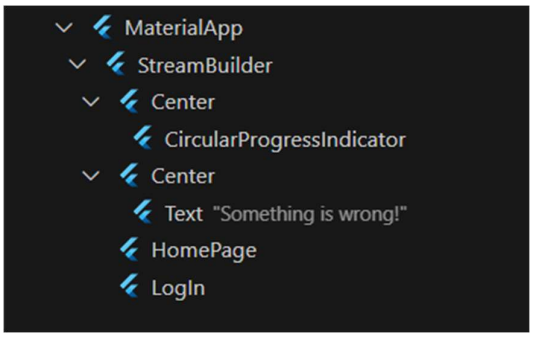


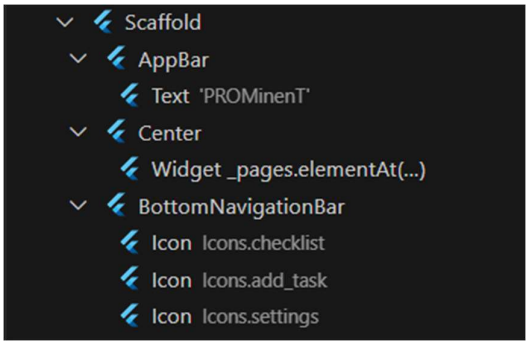
Figure 1: Use Case diagram of PROMinenT. An employee can only register project, register activity, edit activity, and view project progress. Meanwhile an admin can also delete a project or activity. Both user can log in and log out the system.

3 Tree Widget

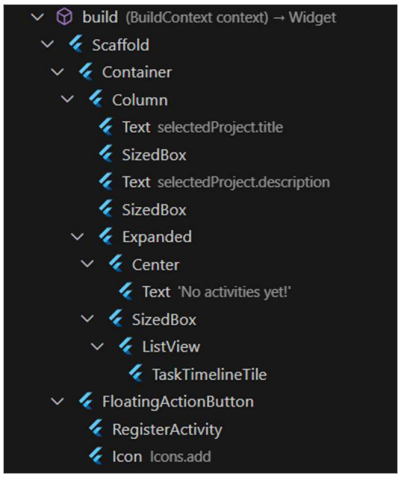
Main



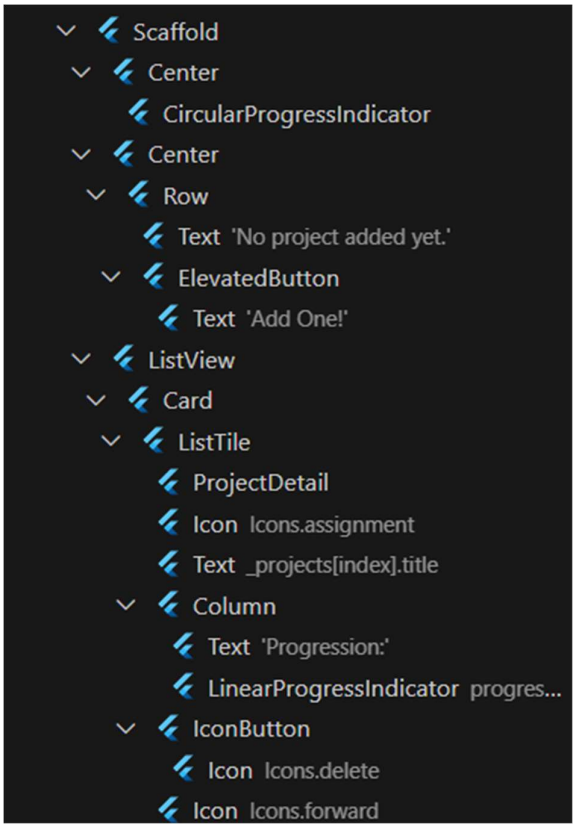
Home page:



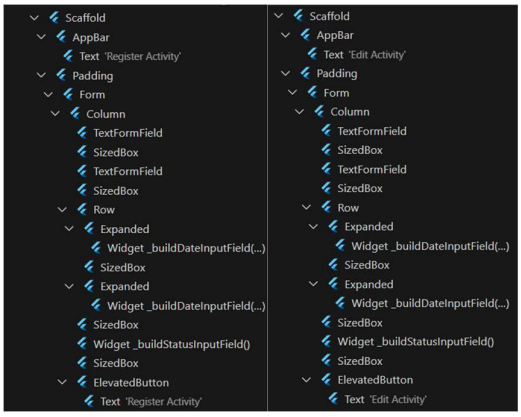
Project Detail page



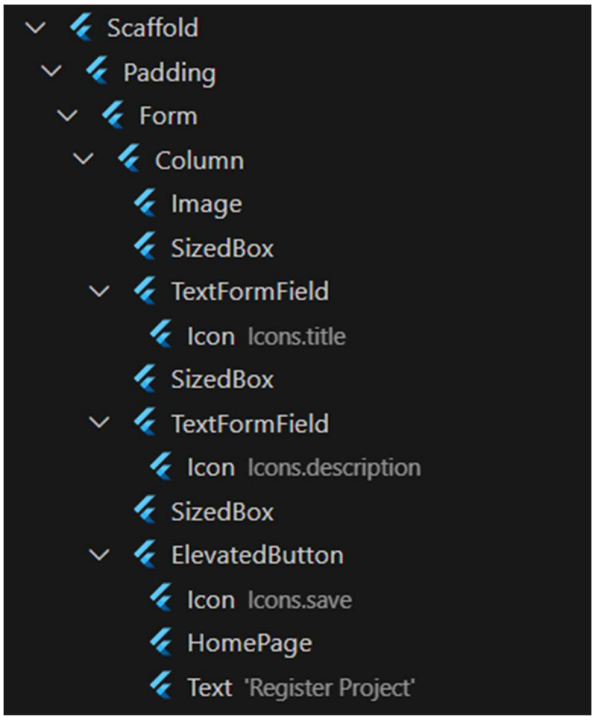
Project list page:



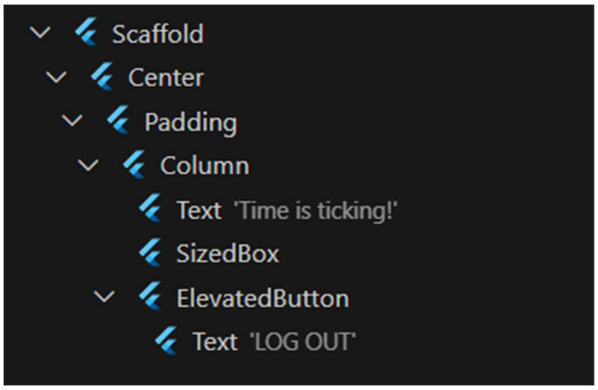
Register Activity & Edit Activity page:



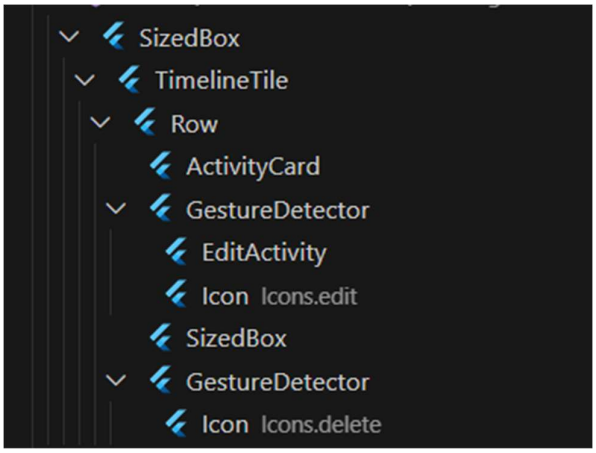
Register Project page:



Setting widget:



TaskTimeLine widget:

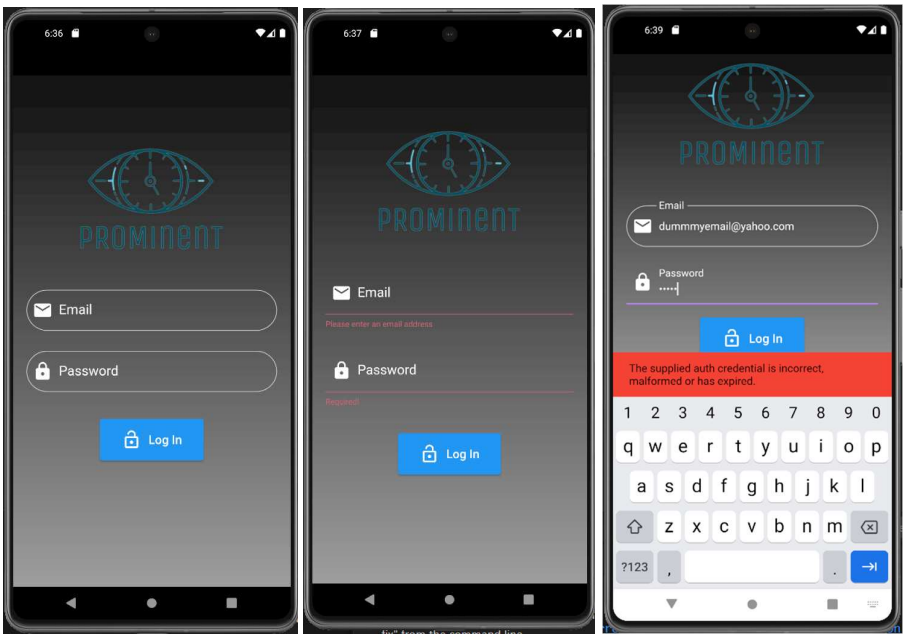


4 Flutter Widget And Features Adopted

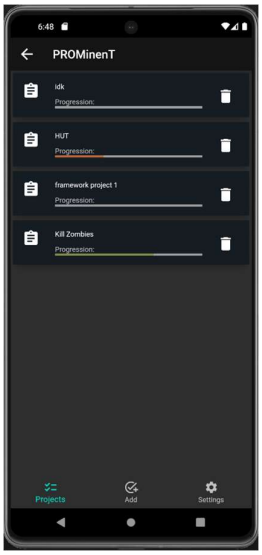
1. **StatefulWidget**: The ProjecList class is a StatefulWidget, allowing dynamic updates and changes in the UI based on user interactions or data changes.
2. **FutureBuilder**: The FutureBuilder widget is used for asynchronous operations, such as fetching project data from Firestore and updating the UI accordingly.
3. **ListView.builder**: This widget efficiently creates a scrollable list of widgets based on the length of the project data, optimizing performance by only building the widgets that are currently in view.
4. **LinearProgressIndicator**: Representing the project progression, it visually displays a linear progress bar.
5. **Card**: The Card widget is used to create visually appealing project list items with elevation, rounded corners, and a background color.
6. **GestureDetector**: Applied to the delete and edit icons, it enables gesture-based interactions, allowing users to tap on the icons to trigger specific actions.
7. **AlertDialog**: This widget displays a pop-up dialog to confirm the deletion of an activity, providing a user-friendly way to handle critical actions.
8. **TimelineTile**: A custom timeline tile to visualize project activities on a timeline, enhancing the user experience with a visual representation.
9. **SizedBox**: Used for specifying dimensions and adding space, enhancing the layout and structure of the timeline tile.
10. **Icons**: Various icons, such as 'delete', 'edit', and status icons, are incorporated to provide visual cues and enhance the user interface.
11. **Column**: Utilized to arrange widgets vertically, specifically for structuring the content of the ActivityCard.
12. **Row**: Organizes widgets horizontally, combining the project information and icons in the timeline tile.
13. **Expanded**: Wrapped around ActivityCard to ensure it takes available space within the Row, allowing flexible and proportional rendering.
14. **Text**: Displays text information, including project titles, descriptions, and activity details, contributing to the textual representation of data.
15. **Navigator**: Facilitates navigation between different screens, enabling users to transition from the project list to the project details or the activity editing screen.
16. **CircularProgressIndicator**: Used as a loading indicator when fetching project data, enhancing the user experience by indicating ongoing background processes.
17. **TextButton**: Employed within the AlertDialog for creating cancel and delete buttons, offering a clear and consistent UI for user interaction.
18. **SnackBar**: Could be used (not explicitly in the provided code) to display brief messages or notifications at the bottom of the screen, providing feedback on actions such as project deletion.
19. **Firestore Integration**: Utilizes the Cloud Firestore database to store and retrieve project and activity data, showcasing integration with Firebase services.

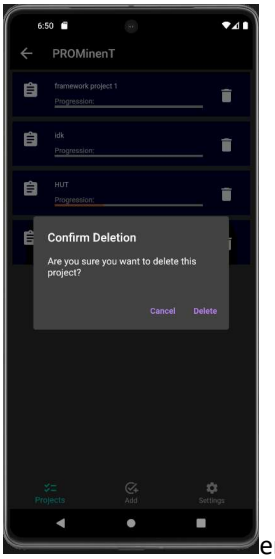
5 UI UX

1. Log In page: prompt the user whenever needed for better usability



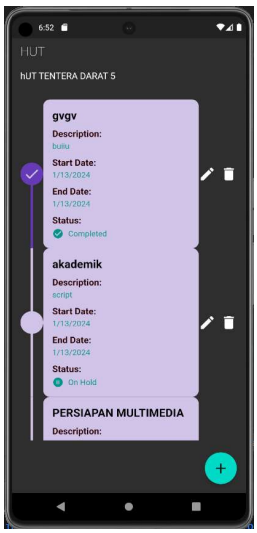
2. Home page: Display a list of projects, only admin can delete a project, and he will be prompted before deleting it. Admin will see the full project list meanwhile employee can only see project under him.

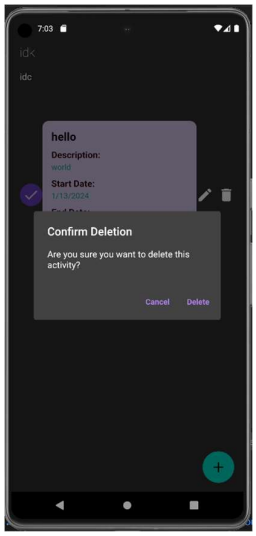
Admin's view: 



employee'view: 

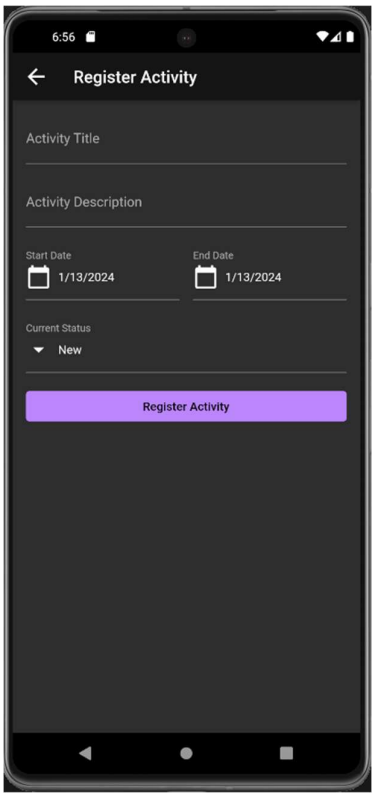
3. Project Detail page: Both employees and administrators can edit, but only an admin can delete

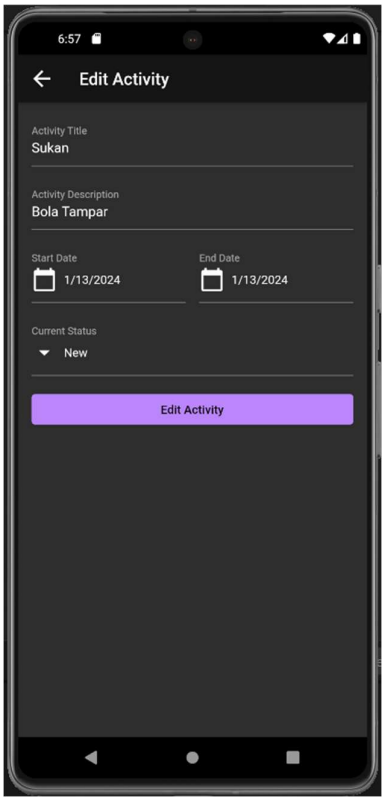
Admin's view: 



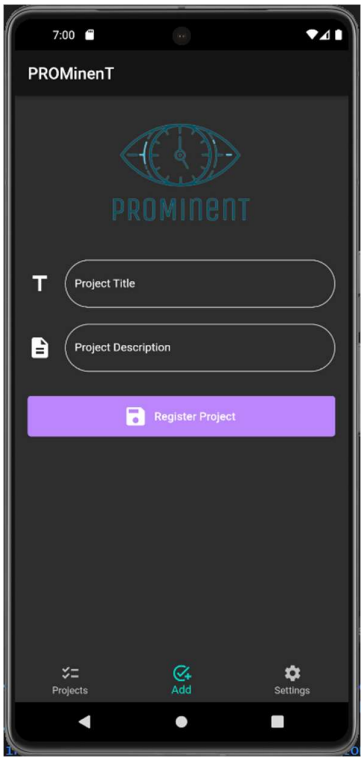
employee' view: 

4. Register Activity Page & Edit Activity Page: TextFormField and prompt to guide user on how to insert/ the information needed for a project or change them

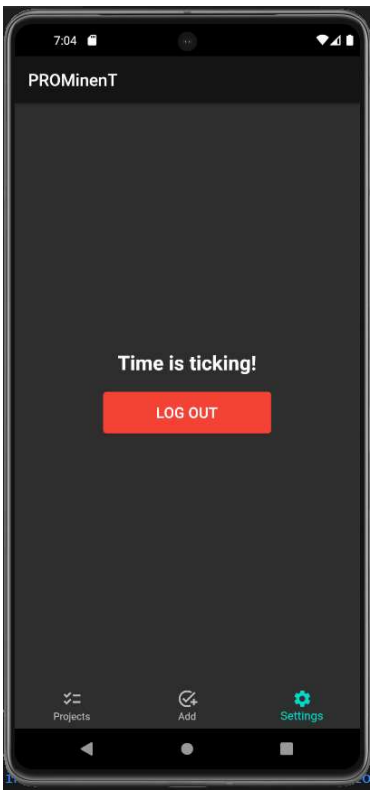


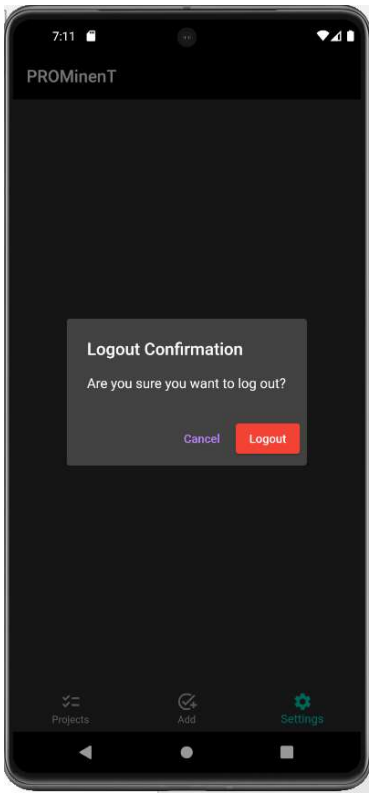


5. Register Project Page: Label text to guide the user



6. Setting page: for log out purpose





6 Conclusion

In conclusion, the developed project management application, named PROMinenT, showcases effective utilization of Flutter and Firebase technologies. The application provides a seamless user experience with features such as project creation, activity tracking, and a dynamic timeline view.

The integration of Firebase's Cloud Firestore enables real-time data synchronization, ensuring that users receive up-to-date information about their projects and activities. The authentication system, powered by Firebase Authentication, enhances the security of user data and allows for personalized experiences.

The UI/UX design of the application emphasizes simplicity and functionality, offering users an intuitive interface to manage their projects and associated activities. The timeline view provides a visual representation of project progress, enhancing project management capabilities.

Throughout the development process, best practices such as asynchronous programming, state management, and code structuring were adhered to, contributing to the application's maintainability and scalability.

While the application currently meets functional requirements, continuous improvement and future enhancements could include additional features, performance optimizations, and a refined user interface to further elevate the user experience.

In summary, PROMinenT stands as a testament to the successful integration of Flutter and Firebase technologies, delivering a robust project management solution for users seeking an efficient and user-friendly platform for organizing and tracking their projects.

7 References

1. Firebase(2014). Add Firebase to your flutter app. (n.d.). Firebase. Retrieved from https://firebase.google.com/docs/flutter/setup?platform=ios
2. Samra Khan.(Jul 19,2023). Getting Started with Firebase Firestore in Flutter: A Comprehensive Guide with CRUD Operations. Retrieve from https://medium.com/@samra.sajjad0001/getting-started-with-firebase-firestore-in-flutter-a-comprehensive-guide-with-crud-operations-ec75f2188355
3. Pinkesh Darji(May 26,2021). How to build a bottom navigation bar in Flutter. Retrieved from https://blog.logrocket.com/how-to-build-a-bottom-navigation-bar-in-flutter/
4. Mitch Koko, (Aug 3,2023). ⏳📱 TIMELINE Widget • Flutter Tutorial 💜. Retrieved from https://www.youtube.com/watch?v=WP0h7utvaUc
5. Jim Jones.(Apr 5, 2018). How to update a single firebase firestore document. Retrieved from https://stackoverflow.com/questions/49682327/how-to-update-a-single-firebase-firestore-document
6. LinearProgressIndicator class. Retrieve from https://api.flutter.dev/flutter/material/LinearProgressIndicator-class.html