VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY

THE INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



**DevOverflow**

By

Trương Tấn Phát

Nguyễn Đăng Bảo

Huỳnh Ngọc Hoài Ân

Nguyễn Cần

A report submitted to Dr. Le Duy Tan - Lecturer of the School of

Computer Science and Engineering

in partial fulfillment of the requirements for the assignment of

Mobile Application Development course in Semester 1 (2023 - 2024)

Ho Chi Minh City, Vietnam  
2024

**DevOverflow**

Ho Chi Minh City, Vietnam  
2024

**DevOverflow**

APPROVED BY: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,   
Le Duy Tan, Ph.D

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Trương Tấn Phát, Leader, ITITIU20140

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Nguyễn Đăng Bảo, Member, ITITIU20164

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Huỳnh Ngọc Hoài Ân, Member, ITITIU20155

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
Nguyễn Cần, Member, ITITIU20171

REPORT COMMITTEE

# ACKNOWLEGMENTS

We express our deep gratitude and appreciation for the professional guidance provided by Dr. Le Duy Tan, PhD. Throughout our journey, Dr. Tan's constant encouragement and support have been instrumental in helping us achieve our goal.

We would also like to extend our gratitude to the other members of the project team: Mr. Nguyen Dang Bao, Mr. Huynh Ngoc Hoai An, Mr. Nguyen Can. Their technical support and positive attitude have made these years a genuinely enriching learning experience.

We would like to extend our sincere gratitude to the esteemed faculty of the School of Computer Science at the International University, with particular acknowledgment to Dr. Le Duy Tan, Ph.D, whose steadfast support has been invaluable since the commencement of our academic journey.

Finally, we express our deep appreciation to the distinguished members of our reading and examination committee for their significant contributions, which have greatly enriched this work.

# TABLE OF CONTENTS

[ACKNOWLEGMENTS 2](#_Toc429539951)

[TABLE OF CONTENTS 3](#_Toc627428334)

[LIST OF FIGURES 4](#_Toc344887507)

[ABSTRACT 5](#_Toc45401517)

[CHAPTER 1 6](#_Toc148356401)

[INTRODUCTION 6](#_Toc1125301008)

[1.1. Background 6](#_Toc795060845)

[1.2. Problem Statement 6](#_Toc38256808)

[1.3. Scope and Objectives 7](#_Toc1690157321)

[1.4. Assumption and Solution 8](#_Toc861248206)

[CHAPTER 2 9](#_Toc1897394246)

[LITURATURE REVIEW/RELATED WORK 9](#_Toc1900223648)

[CHAPTER 3 10](#_Toc329652673)

[METHODOLOGY 10](#_Toc2052558598)

[3.1. User requirement analysis 10](#_Toc1141707156)

[3.2. System Design 10](#_Toc1617598377)

[3.3. Database design 11](#_Toc858473962)

[CHAPTER 4 12](#_Toc147909041)

[IMPLEMENT AND RESULTS 12](#_Toc1161731130)

[4.1. Implementation 12](#_Toc29503086)

[4.1.1. Login into the app 12](#_Toc533523759)

[4.1.2. Home Screen 13](#_Toc1549104994)

[4.1.3. Putting up questions 13](#_Toc605210106)

[4.1.4. Viewing questions and answering questions 14](#_Toc370242297)

[4.1.5. Search for specific questions 15](#_Toc150943004)

[4.1.6. View your own questions 16](#_Toc1636073728)

[CHAPTER 5 18](#_Toc42173020)

[DISCUSSION AND EVALUATION 18](#_Toc754905293)

[CHAPTER 6 19](#_Toc928625774)

[CONCLUSION AND FUTURE WORK 19](#_Toc1525248744)

[6.1. Conclusion 19](#_Toc1590405385)

[6.2. Future work 19](#_Toc1338655299)

[REFERENCES 21](#_Toc566574865)

[APPENDIX 22](#_Toc258982011)

# ABSTRACT

As mobile technologies continue to grow, the need for platforms that allow developers to transact while on the go has become dire and pressing. To address this need, a mobile application known as DevoverFlow was developed with the intention of offering a seamless interface that allows developers to engage and interact over technical questions, answers and discussions similar to Stack Overflow. The primary aim of the research is to understand the relevance of DevoverFlow in solving developer problems and facilitating information exchange in particular on mobile devices. Developed using Kotlin and operating on Android Compose, DevoverFlow ensures lightning fast interactions and user-friendly interface. Application integrates the features of voting, real-time submission of questions and answers, and material creation according to the user’s needs, which makes it wise to use. The evaluation incorporates a performance statistics and users feedback by deploying a mixed methods approach to assess the impact of the app on knowledge exchange as well as the extent to which it aids developers in addressing the concerns. In this abstract, a mobile developer-centric collaboration platform named DevoverFlow is described together with its peculiarities and development. The authors describe the purposes of the research and the prospective capability of facilitating community of developers and passion for technology among them is assessed.

# CHAPTER 1

# INTRODUCTION

## Background

The staggering growth of the IT sector has led to the demand for trustworthy and effective resources which assist programmers in the improvement of their skills and in resolving complicated issues. Such platforms cater to problems arising in the industry, and one such tool which has played a pivotal role in assisting developers is Stack Overflow due to its extensive database with user generated content and solutions to multiple programming questions. Even though Stack Overflow has been revolutionary for developers, it is quite unfriendly to developers who change the device they are solving the problem on since it is heavily tailored to desktops.

Another area of opportunity for developers is the transformation of stack overflow’s primary function into something that is more mobile-first friendly in light of the rapid increase in the use of mobile phones among the coding community. To fill this gap, DevoverFlow was developed as a mobile application where developers can create, share and answer technical questions from anywhere and at any time. By taking advantage of the mobile-first approach and advanced technologies such as Kotlin and Android Compose, DevoverFlow aims to enhance the experience of professional developers who are always on the go and are in need to solve problems and gain knowledge seamlessly.

DevoverFlow's underlying problem is that there is a desire for a more convenient software with the capability of vertical communication with the international developer community, rather being constrained to desktop computers. As there is a growing trend, so is the use of DevoverFlow, as a mobile-first approach in today’s developer network is of great importance and the potential benefits of offering mobile users access to the huge resource pool of these communities, such as Stack Overflow, is significant.

## Problem Statement

With the growth of the IT sector, developers are getting more dependent on online platforms such as Stack Overflow for solving programming difficulties, sharing expertise, and collaborating with others. However, the desktop-based nature of Stack Overflow and related platforms limits their usability and accessibility for developers who need solutions on the move. This creates a dent for mobile developers who need a mobile-first and convenient solution to reach out to the developer community and get that important piece of information anywhere, anytime.

The inability to develop a mobile-friendly collaborative platform indeed creates a real problem in productivity for those professionals who are never at their desk. That is hard to offer today as a surrogate for the mainstream desktop-centric platforms: first, current mobile solutions come either ith reduced capability or, secondly, they dont correspond to the needs of the developers themselves.

DevoverFlow intended to solve this problem by introducing a fully-mobile-optimized platform with the complete functions of Stack Overflow, which introduces convenience to users by making every action easy for the mobile users. By having such software enable developers to communicate with the community, to be able to ask questions or give answers on the go, bridge this gap in developing their capabilities of solving and learning, instead of sitting in front of a computer. This is further compounded by the rarity of applications that mix and match various modes of collaboration, in context, with a mobile-first approach—ingredients essential to fostering a vibrant developer ecosystem.

## Scope and Objectives

**Scope:**

This project involves the design and evaluation of DevoverFlow, a mobile application developed to emulate most of the essential features of Stack Overflow for the mobile-first paradigm. The program will allow developers to interact with the wider development community from their cellphones and create avenues to share questions and answers. Included within the scope of work is the following:

1. **Mobile Application Development**: The application is to be developed using Kotlin, together with Android Compose, which shall be fully optimized for mobile, giving full importance to responsive performance, ease of use, and accessibility.

1. **Key Features of the Application**: The following are the key features of the application.

* Question Submission: "The developers can share code-related questions or problems."
* Answer Sharing: "When the question is posted, a member of this community can give his or her response with personal opinion and solutions for the issues."
* Voting: To stress the most helpful submissions, it supports voting-down or voting-up questions and answers.
* Search and Filter: With an advanced filtering feature, users will be able to search for related queries or answers.
* Personalized Content: The application will automatically give suggestions on queries and responses based on interest and behavior of the user.

1. **User Experience**: Simple, easy-to-use with a seamless interface, the application will focus on soliciting continuous interaction and engagement.

1. **Research and Evaluation**: Several ways like performance analysis, usability testing, and user feedback will be applied to evaluate the effectiveness of DevoverFlow.

**Objective:**

The following are the primary objectives of the project:

1. Create a Mobile-Ready Platform: to develop a mobile application that can simulate the functionality of Stack Overflow and, therefore, provide developers with an efficient and accessible means of solving problems in programming while on transit.

1. Enhancing User Accessibility: Ensuring that developers can access the application on their respective mobile devices without any limiting factors, and therefore provide full participation and contribution with ease and frequency.

1. Exchange Information: Create a mobile community of developers who can collaborate in real time by asking questions and sharing answers, thereby facilitating the flow of information across different programming languages, frameworks, and technologies.
2. Effectiveness: Based on collection and usage metrics, with a comparison to current desktop-based solutions, the effectiveness of the Devoverflow platform should be evaluated in enhancing productivity and problem-solving ability in mobile developers.
3. Adapt Desktop Functionality for Mobile Use: Re-enable the key functionality of desktop-based platforms, such as Stack Overflow, for mobile use in a manner that retains essential elements and is correctly optimized for mobile users.

By achieving these objectives, DevoverFlow aims to provide a valuable tool for developers, enabling them to continue learning, sharing, and collaborating in an increasingly mobile-driven world.

## Assumption and Solution

**Assumption:**

1. Mobile in Development Use is Growing: The belief is held that developers make increasing use of mobile devices to browse, learn, and solve problems. The need for this therefore constitutes an increasing need for mobile-first solutions that satisfy their work needs while on the go.
2. Current platforms are designed for desktop usage. For example, Stack Overflow and other developer communities have been designed for desktop usage, while their mobile versions either don't work fully or aren't optimized, which may reduce developers' productivity while working with them on either tablets or smartphones.
3. Developers desire speed and access easiness in solving coding problems. Delays and poor mobile usability may even drive users into a rage, at which they just stop using a platform.
4. A Need for Personalization: The developers benefit in having the content tailored to their past behavior, interests, and areas of expertise; this will provide better user engagement and make the atmosphere more productive for problem-solving by giving tailored suggestions for queries or responses.
5. More Engagement due to Mobile-First Design: Emphasizing mobile-first design, it is believed, would provide users with a more seamless and intuitive experience that will drive user happiness, contributing more often to the platform and driving more use.

**Solution:**

DevoverFlow is a mobile-first application that emulates the main features of Stack Overflow but is specially crafted for smartphones and tablets to address the presumptions and close the detected gap. The following are the basic elements of the solution:

1. Completely Mobile-Optimized Experience: DevoverFlow's UI and UX were crafted with the mobile device in mind, so the developers should feel a smooth, responsive, and easy-to-use experience. Due to characteristics such as simplicity in interaction patterns, fast load times, and ease of navigation, this software works effectively even with the highly constrained resources typical on mobile devices.
2. Real-Time Interaction: The platform will allow for real-time question-and-answer sessions that shall have developers collaborate in a global developer community and obtain immediate feedback.
3. Push Notifications and Updates: Real-time push notifications pop up when their questions are answered, when answers are updated, or when new questions are posted around their interests. This feature shall ensure that the developers will always use the platform and stay updated.
4. Improved User Engagement: DevoverFlow has features for promoting user activity: notification, personalized information, mobile-first design of the UI, which lower barriers to participation, enabling developers to learn, solve problems, and contribute from any location.

DevoverFlow solves the problem of limited access to developer communities on mobile by providing these capabilities in a mobile-optimized way. The app enhances cooperation and makes solving problems more effective by allowing developers to continue working without interruption and eventually makes the global development community more vivid and accessible.

# CHAPTER 2

# LITURATURE REVIEW/RELATED WORK

Within the last couple of years, much interest has taken place in mobile applications regarding developers' collaboration and cognitive enhancement. According to researches of cognitive training games like Lumosity and Peak, mobile apps can effectively enhance one's memory, attention, and problem-solving skills (Jaeggi et al., 2008; Owen et al., 2010). These applications provide adaptable, easily available solutions that have shown the potential of mobile platforms for cognitive growth.

Sites such as Stack Overflow have become integral parts of the processes to share knowledge and troubleshoot problems for many developers. However, their desktop-centric designs significantly hinder mobile access. According to McPherson et al. (2018) and Sillito et al. (2014), these are real, actual problems encountered by developers: slow loading and restricted functionality. Although Stack Exchange and GitHub's mobile versions have tried to bridge these issues, they are still not fully in feature-parity with their respective desktop versions.

Thus, the solution would involve an explicit requirement for a mobile-first platform like DevoverFlow that really represents the complete functionalities of desktop developer platforms and gives a seamless, mobile-optimized experience. DevoverFlow is working on closing this gap by providing developers with a very friendly and intuitive mobile way to question, share, and productively work while on-the-go.

# CHAPTER 3

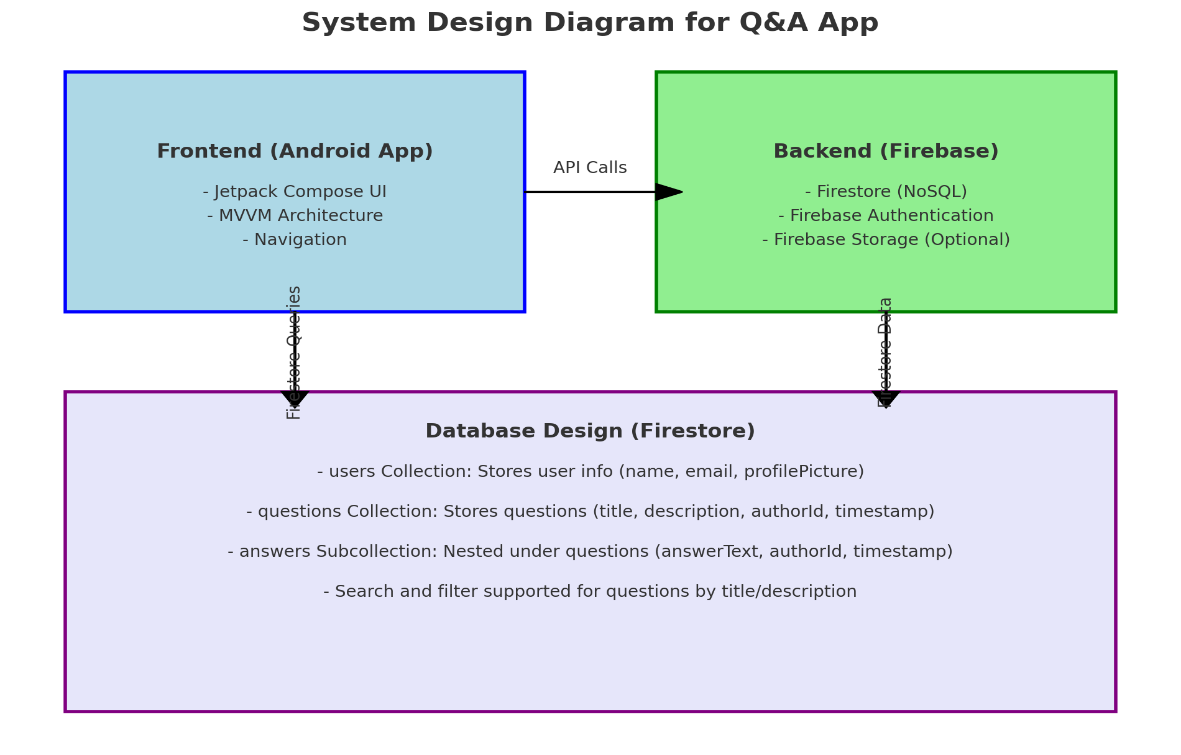
# METHODOLOGY

## User requirement analysis

This application aims to identify and define the features, functionalities, and user expectations for the Q&A app. This analysis will help ensure the app aligns with user needs and provides a seamless experience

## System Design

Based on the extracted structure and the description of your app (a question-and-answer platform like a mini StackOverflow), here's a high-level system design:



**Figure :General interaction flow**

* Frontend: The Android app, built with Jetpack Compose, follows the MVVM architecture and uses Jetpack Navigation.
* Backend: Firebase handles authentication, data storage with Firestore, and optional storage for files or media.
* Database Design: Firestore has collections like users, questions, and a subcollection answers nested under questions. Queries allow filtering and searching.

## Database design

The app uses Firebase Firestore, a NoSQL database, with the following collections and structure:

1. **users**

Collection:  
- **Purpose**: Stores user information.  
- **Document ID**: Firebase Authentication UID.  
- Fields:

* name: User's display name.
* email: User's email address.
* profilePicture: URL of the user's profile picture (optional).

1. **questions** Collection:  
   - **Purpose**: Stores all questions.  
   - **Document ID**: Unique ID generated by Firestore.  
   - **Fields**:

* title: Title of the question.
* description: Detailed description of the question.
* authorId: UID of the user who created the question.
* timestamp: Timestamp of when the question was created.

1. **answers** Subcollection (Nested under **questions**):  
   - **Purpose**: Stores answers for each question.  
   - **Document ID**: Unique ID generated by Firestore.  
   - **Fields**:

* answerText: Text of the answer.
* authorId: UID of the user who created the answer.
* timestamp: Timestamp of when the answer was created.

# CHAPTER 4

# IMPLEMENT AND RESULTS

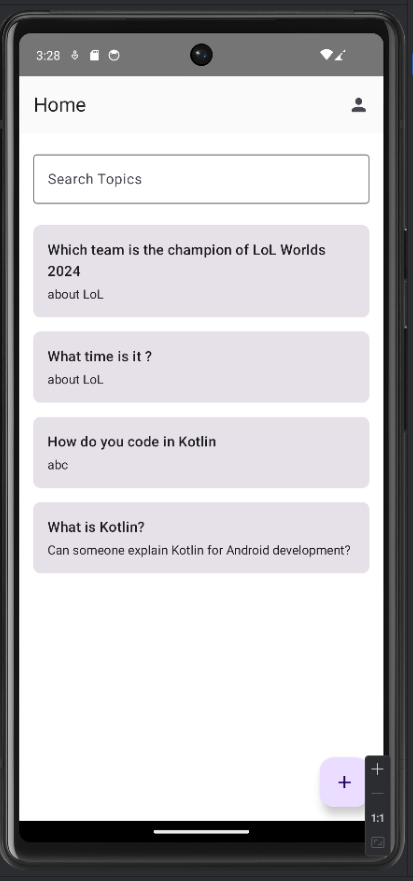
## Implementation

### Login into the app



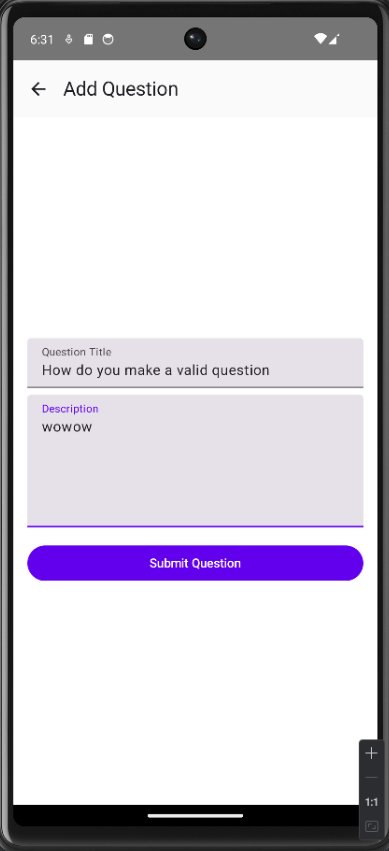
First you have to login with an account to be able to access the application

### Home Screen



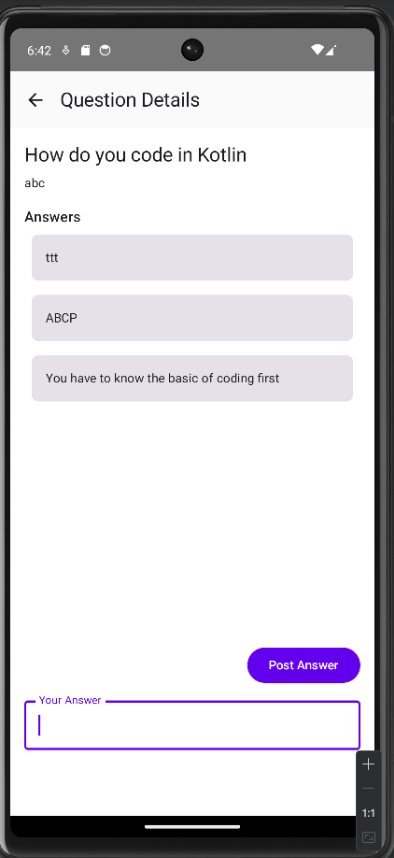
After you’ve successfully login, you are default to be set at the question home screen

### Puttig up questions



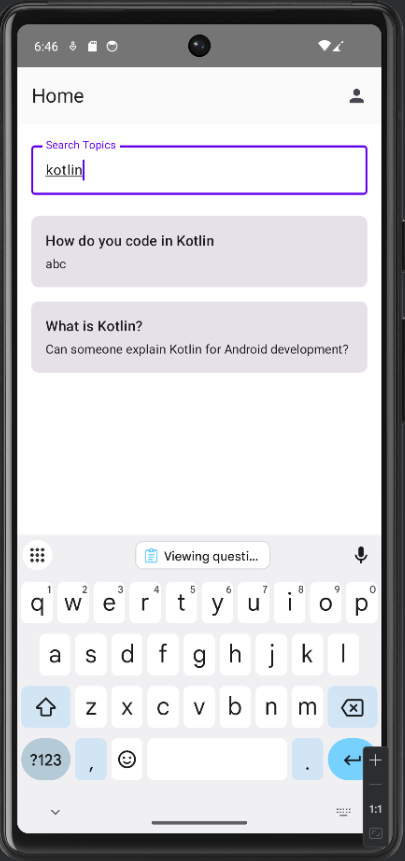
You can put up questions to ask other members about a problem that you are facing

### Viewing questions and answering questions



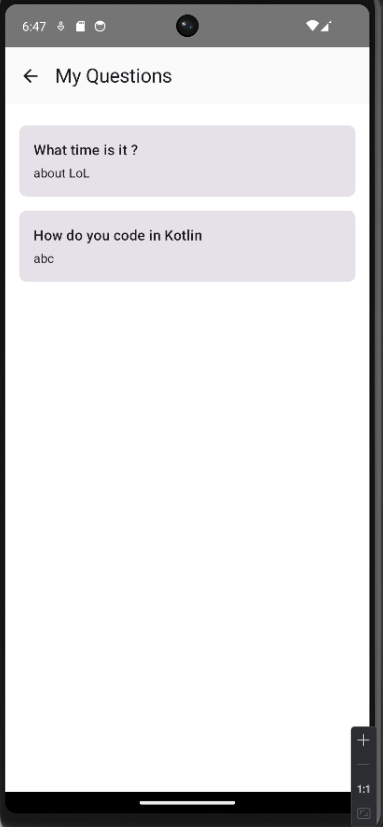
You and other members can give answers to questions that has been posted

### Search for specific questions



You can also search for a specific question

### View your own questions



You can also view your own question that has been posted to see if others had answered or not

# CHAPTER 5

# DISCUSSION AND EVALUATION

This application was developed as a custom implementation of a Question and Answer (Q&A) platform, taking inspiration from popular knowledge-sharing platforms like StackOverflow. The main objective was to create a simple yet functional platform where developers, students, and professionals can ask questions, provide answers, and engage in discussions. Built using Android Jetpack Compose for the user interface, Firebase for authentication and database management, and Kotlin for logic handling, the project showcases a modern development approach with a focus on usability.

Unlike established platforms, this application was built entirely from scratch, without using pre-existing source code or templates. Every feature, including user authentication, question management, search functionality, and data handling, was manually designed and implemented to fit the specific requirements of the project. This provided an opportunity to tailor the system design and database structure to meet the desired objectives. However, some limitations were inevitable due to the scope and timeline of the development. For example, features like collaborative editing, voting on answers, tagging questions, or a reputation system for rewarding users are currently not included. These features, while desirable, were outside the immediate goals of the project.

One of the key challenges during development was optimizing the Firebase Firestore database for scalability. With larger datasets, maintaining fast query responses, especially for search functionality and real-time updates, can become a bottleneck. While the app performs well with small to medium datasets, further optimization would be necessary for larger-scale deployments. Another aspect that could benefit from improvement is the user experience (UX). Although the app’s interface is functional and intuitive, adding features like improved accessibility, enhanced visual feedback, and more polished UI elements could significantly improve user engagement.

Despite these challenges, the core functionalities of the application, including adding questions, viewing details, searching, and managing user-specific content, have been successfully implemented. While it lacks some advanced features of larger platforms, this project effectively demonstrates the fundamental capabilities of a Q&A platform. It provides a solid starting point for future enhancements, such as introducing advanced features, optimizing database queries for larger datasets, and refining the user interface to create a more seamless experience.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## Conclusion

This project successfully delivers a functional Question and Answer (Q&A) platform that provides a streamlined and intuitive experience for users to ask questions, provide answers, and engage in discussions. The application leverages modern tools and technologies such as Android Jetpack Compose, Firebase Firestore, and Firebase Authentication to ensure scalability, security, and performance. The adoption of the MVVM architecture ensures a clean separation of concerns, making the codebase modular and maintainable.

Key functionalities, including user authentication, question posting, answer management, and search capabilities, have been implemented effectively. The application also offers a personalized experience by allowing users to manage and view their questions separately in the "My Questions" section. The search bar enhances usability by enabling users to filter and locate questions efficiently. Overall, the project meets its primary objective of fostering a knowledge-sharing environment in a lightweight and user-friendly manner.

Despite its success, the project is not without limitations. Advanced features such as voting on answers, tagging, or a reputation system are currently missing. Additionally, while the app performs well for small to medium datasets, optimizing database queries for large-scale use remains an area of improvement. The user interface, while functional, can also be further refined to improve visual appeal and accessibility.

## Future work

There are several opportunities to enhance the functionality and user experience of this application in future iterations. These improvements focus on both expanding the app's feature set and refining its performance to better meet user needs.

1. Adding Advanced Features: Future updates could introduce a voting system for questions and answers, allowing users to rank content based on its usefulness. Features like tagging questions would make it easier to organize and search for specific topics, while a reputation system could reward active contributors and encourage greater participation.
2. Improving Search Capabilities: The search functionality can be expanded to include full-text search, enabling more accurate and relevant results. Additional search filters, such as sorting by date, popularity, or tags, would further enhance the user experience.
3. Enabling Real-Time Updates: Real-time synchronization for questions and answers could be implemented using Firebase Firestore’s live data capabilities. This would allow users to see updates immediately, creating a more interactive and dynamic platform.
4. Enhancing the User Interface: A redesigned user interface with a modern look and feel would make the app more visually appealing. Accessibility improvements, such as voice input for search, adjustable text sizes, and better contrast, could make the app more inclusive and user-friendly.
5. Optimizing Performance: To ensure the app performs well as the user base and dataset grow, database queries can be optimized for scalability. Adding caching mechanisms would also help reduce database load and improve overall performance.
6. Expanding Platform Availability: To reach a wider audience, the app could be developed for iOS using cross-platform tools like Kotlin Multiplatform or Flutter. Additionally, creating a web version of the platform would make it accessible to users across all devices.
7. Introducing Community Features: Community-driven enhancements, such as moderation tools to manage inappropriate content and the ability to comment on questions and answers, could foster better discussions and maintain the quality of content on the platform.

By focusing on these areas, the application can grow into a comprehensive and engaging platform that not only meets but exceeds user expectations. These proposed updates build on the strong foundation established in this project and pave the way for future scalability and innovation.

# REFERENCES

# APPENDIX