

**AN-NAJAH NATIONAL UNIVERSITY**



**FACULTY OF ENGINEERING AND INFORMATION  
TECHNOLOGY**

**Computer Engineering Department**

**Graduation Project 1**

**Logic**



**Accomplished by: Sama Shawa ,Samah Qaradeh  
Supervisor: Dr. Asmaa Afeefi, Dr. Suleiman Abu Kharmeh**

## Acknowledgment

We are truly grateful to our supervisors, **Dr. Asmaa Afeefi** and **Dr. Suleiman Abu Kharmeh**, for their cooperation and support throughout the semester. Their guidance and encouragement made a significant difference in the progress of our project.

We would also like to give special thanks to all the professors in the **Computer Engineering Department** who were always available and willing to help. Their answers to our questions and the support they provided gave us an incredible experience that really helped us improve and complete our work.

YoursSincerely

Sama,Samah.

## Disclaimer

This report was written by sama shawa and samah qaradeh at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

# Contents

|   |          |
|---|----------|
| 1. Abstract.....                                  | 6        |
| 2. Introduction .....                             | 7        |
| 2.1 Background.....                               | 7        |
| 2.2 Problem Statement.....                        | 7        |
| 2.3 Significance.....                             | 7        |
| 2.4 Objectives and Scope.....                     | 8        |
| 3.Constraints.....                                | 9        |
| <b>3.1 Question &amp; Answer Constraints.....</b> | <b>9</b> |
| 3.2 User Constraints.....                         | 9        |
| 3.3 Contest & Leaderboard Constraints.....        | 9        |
| 3.4 Notification System Constraints.....          | 9        |
| 3.5 Performance & Scalability Constraints.....    | 9        |
| 3.6 Security Constraints.....                     | 9        |
| 4.Methodology.....                                | 10       |
| 4.1 Architecture & Technologies Utilized.....     | 10       |
| 4.1.1 Frontend Design.....                        | 10       |
| 4.1.2 Backend Operations.....                     | 10       |
| 4.1.3 Real-Time Chat and Notifications.....       | 11       |
| 4.2 Features & Implementation.....                | 12       |
| 5.Results & Discussion.....                       | 10       |
| 5.1Final Application .....                        | 38       |
| 5.2 Project Outcomes .....                        | 39       |
| 6.Conclusion & Future Work.....                   | 40       |
| 6.1 Conclusion.....                               | 40       |
| 6.2Future Work .....                              | 40       |
| 7.Bibliography.....                               | 41       |

## List of Figures

- 1.login &signup
- 2.Add admin
- 3.Profile page
- 4.Edit profile
- 5.Menu
- 6.Question page & Show Question &Show Answer (user)
- 7.Stage one & Stage one & stage three(admin)
- 8.AddQuestion&Question process
- 9.Latest answer
- 10.Chats page
- 11.Advertisement
- 12.Contest page(admin)
- 13.Add contest
- 14.Question contest
- 15.Contest page(user)
- 16.Favorites page
- 17.Leaderboard page
- 18.Notifications page
- 19.Recommendation Page
- 20.Attempts

# Chapter 1

## Abstract

In our project we will use flutter for frontend and mongo for the backend, also we will use firebase for chatting and notification (extra feature).

We will make a mobile and web application for education, our website will teach people logical thinking, without need to know any programming language, our project will be similar to hacker rank, with out coding, there is only 1 answer (number or multiple choice), there is 2 roles admin and user, admin can add questions to the project, also user can add them, and they will be approved by admin. Questions have many tags like math, probability, logic, graph ..... , each question has a specific score depend on it is difficulty, also it has specific number of attempts, user can leave a comment and rate the question , each user has an account that include his information like name, picture, name of the country, number of question he solved, with it is type. For every question in the project it has specific point, type of question, the solution of question (it will appear after exceeded the number of attempts).

Also we have contests in our project, admin make them with specific number of questions with different difficulties, specific time for each contest, the contest is live and it has multiplier, they compete with each other, finally after the contest there is a scoreboard, and each player will have his own score, this score is saved also in his profile as a chart.

# Chapter 2

## Introduction

### 2.1 Background

Logical thinking and problem-solving are skills that play a vital role in both education and professional life. However, most platforms that help people develop these skills rely on programming knowledge, which can be overwhelming for those who are just starting out or have no coding experience. While platforms like HackerRank are great for programmers, they leave out people who simply want to improve their logical reasoning without diving into coding. Our project aims to fill this gap by creating a platform that focuses on teaching logical thinking in a simple, fun, and engaging way. It's designed to help people strengthen their reasoning skills without needing to know how to code.

### 2.2 Problem Statement

Many people want to improve their logical thinking skills, but most tools available today are heavily focused on coding, which can discourage non-programmers. There's also a lack of platforms that allow users to actively contribute, compete, and learn in a personalized, interactive way. What's missing is a platform where users can practice logical thinking with straightforward challenges, participate in contests.

### 2.3 Significance

This project is important because it gives everyone a chance to improve their logical thinking, no matter their background or experience. By making the platform interactive and user-friendly, we encourage people to learn at their own pace while staying motivated. Users can contribute their own questions, join live contests, and see their progress in real-time through personalized profiles. With features like chatting and notifications, the platform also fosters a sense of community. By combining education and fun, this project can help people worldwide develop critical thinking skills in a way that's accessible and engaging.

## 2.4 Objectives and Scope

### Objectives

- Build a mobile and web application that focuses on teaching logical thinking without requiring coding knowledge.
- Allow both admins and users to add questions, with admin approval for user submissions.
- Include features like scoring, rating, and comments on questions to improve user engagement.
- Provide user profiles that display personal statistics, solved questions, and performance in contests.
- Organize live contests with real-time leaderboards to encourage competition and learning.
- Use Firebase for chatting and notifications as extra features to enhance interaction.

### Scope

Our platform is designed for anyone who wants to improve their logical thinking skills, whether they're students, professionals, or just curious learners. The project will focus on making the experience seamless across mobile and web, using Flutter for the frontend and MongoDB for the backend. With its interactive and competitive elements, the platform will serve as a fun and accessible tool for logical reasoning.

# Chapter 3

## Constraints

### 3.1 Question & Answer Constraints

- Each question must have a **single correct answer** (either a number or multiple-choice).
- Users have a **limited number of attempts** before the solution is revealed.
- Questions must be **tagged properly** (e.g., math, logic, probability, graphs) to allow filtering and categorization.
- Difficulty levels must be assigned correctly to **affect scoring and user rankings**.

### 3.2 User Constraints

- Users can submit questions, but they require **admin approval** before being added to the system.
- Users can comment and rate questions, meaning **spam or inappropriate content filtering is needed**.
- User profiles must store information such as **name, picture, country, and statistics**.

### 3.3 Contest & Leaderboard Constraints

- The contest system must ensure **real-time updates on the scoreboard**.
- User ratings should be stored and displayed in **charts** for historical tracking.
- There should be **no possibility of cheating** (e.g., users should not see the correct answer before exhausting attempts).

### 3.4 Notification System Constraints

- Users should receive **timely notifications** about new questions, contests, and important updates.
- Notifications should be **efficiently handled** to prevent system overload.

### 3.5 Performance & Scalability Constraints

- The system should be **optimized for fast query performance**, especially when fetching and updating scores.
- If many users participate in a contest, the leaderboard should be updated **without significant delays**.
- The application should be designed to handle **high traffic efficiently**.

### 3.6 Security Constraints

- User data, including login credentials, must be **securely stored and encrypted**.
- The system should prevent **unauthorized modifications** to scores or questions.

# Chapter 4

## Methodology

### 4.1 Architecture & Technologies Utilized

To create an engaging and efficient platform, we've carefully chosen modern tools and a robust architecture that works well for both mobile and web users. Here's an overview of how everything comes together.

#### 4.1.1 Frontend Design

The frontend is all about making the platform look good and feel easy to use, whether you're on a smartphone or a computer.

- **Why Flutter?**  
Flutter was our top choice because it allows us to create a single app that works seamlessly on both mobile devices and web browsers. Its widgets make designing smooth animations and responsive layouts a breeze.
- **Key Features:**
  - A consistent and attractive user interface across devices.
  - Easy navigation to contests, questions, and user profiles.
  - Fluid animations to make the user experience enjoyable.

#### 4.1.2 Backend Operations

The backend takes care of all the behind-the-scenes work, like storing data and managing user actions.

- **Why MongoDB and Node.js?**
  - **MongoDB:** Its flexible structure lets us easily store all kinds of data, like user profiles, questions, and ratings, without worrying about rigid schemas.
  - **Node.js:** It handles requests quickly, making sure the platform runs smoothly, even when lots of users are active.
- **Key Features:**
  - Secure login with roles for users and admins.
  - Fast handling of user interactions, like submitting answers or leaving comments.
  - Real-time updates for contests and leaderboards.

### 4.1.3 Real-Time Chat and Notifications

We wanted users to feel connected and informed, so we added a chat and notification system.

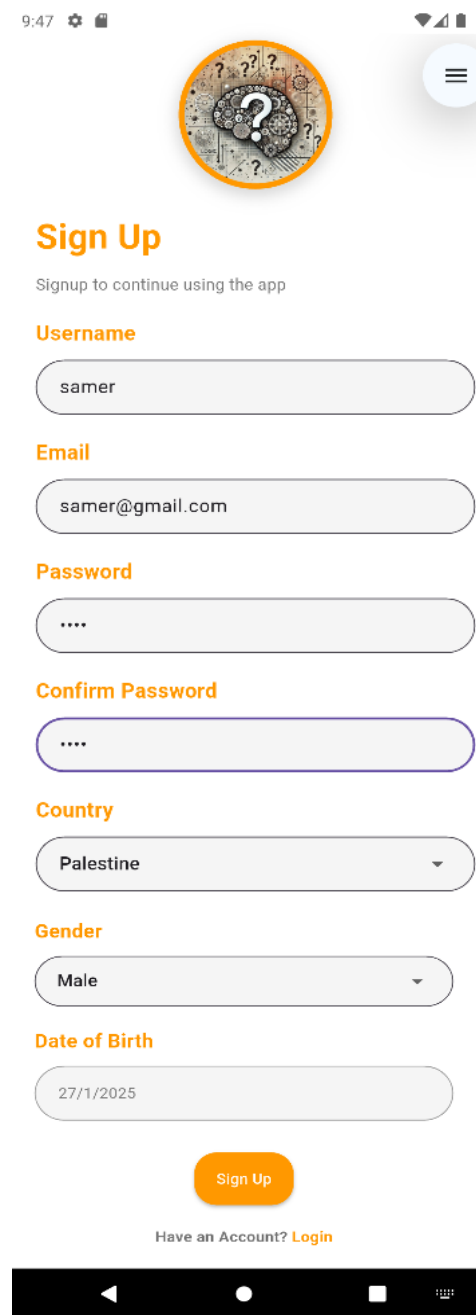
- **Why Firebase?**  
Firebase is perfect for real-time updates and sending notifications. It keeps users in the loop with things like contest reminders or replies to their comments.
- **Key Features:**
  - Instant messaging between users and admins.
  - Push notifications for important updates, like new contests or leaderboard results.

## 4.2 Features & Implementation

The application provides many features and a lot to talk about. This section will focus on the application features and the role of each library in the implementation.

### login &signup

When a user opens the app for the first time, they'll need to sign up by providing their information. Once that's done, they can log in and start using the app. As for the admin, they can skip the sign-up process and log in directly.



9:47

9:47

9:47

### Sign Up

Signup to continue using the app

**Username**

samer

**Email**

samer@gmail.com

**Password**

....

**Confirm Password**

....

**Country**

Palestine

**Gender**

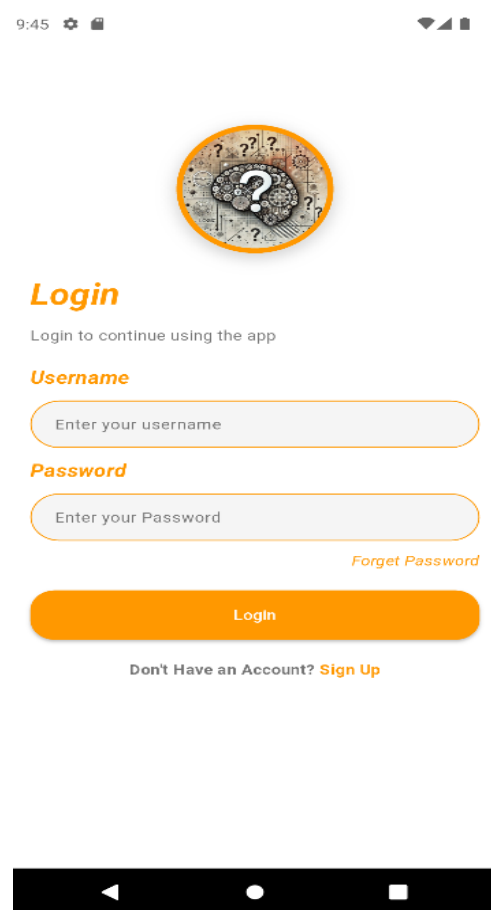
Male

**Date of Birth**

27/1/2025

Sign Up

Have an Account? [Login](#)



9:45

9:45

### Login

Login to continue using the app

**Username**

Enter your username

**Password**

Enter your Password

[Forget Password](#)

Login

Don't Have an Account? [Sign Up](#)

## Add admin

Admins can add new admins and specify their roles—whether they are responsible for **Logical**, **Writing**, or act as the **Main Admin**.

Admins can view all other admins in the system as well as all users. They have access to a complete list of both.

11:41

Admin

**Add Admin**  
Sign up to continue using the app

**Admin Name**  
Enter your admin name  
rami

**Email**  
Enter your email  
rami@gmail.com

**Password**  
Enter your password  
.....

**Confirm Password**  
Confirm your password  
.....

**Country**  
Palestine

**Gender**

Gender  
Male

Admin Type  
For Writing

Birth Date  
Select your birth date  
16/1/2025

Add Admin

11:41

Admin & Users

+ Add Admin

Search... Search

**Admin Table**

| Username | Role         | Email           | Gender |
|----------|--------------|-----------------|--------|
| amal     | logicalAdmin | amal@gmail.com  | F      |
| ola      | writerAdmin  | ola@gmail.com   | F      |
| rami     | writerAdmin  | rami@gmail.com  | M      |
| samaa    | admin        | samaa@gmail.com | F      |
| samar    | admin        | samar@gmail.com | F      |
| sara     | writerAdmin  | sara@gmail.com  | F      |
| soso     | admin        | soso@gmail.com  | F      |
| yara     | admin        | yara@gmail.com  | F      |

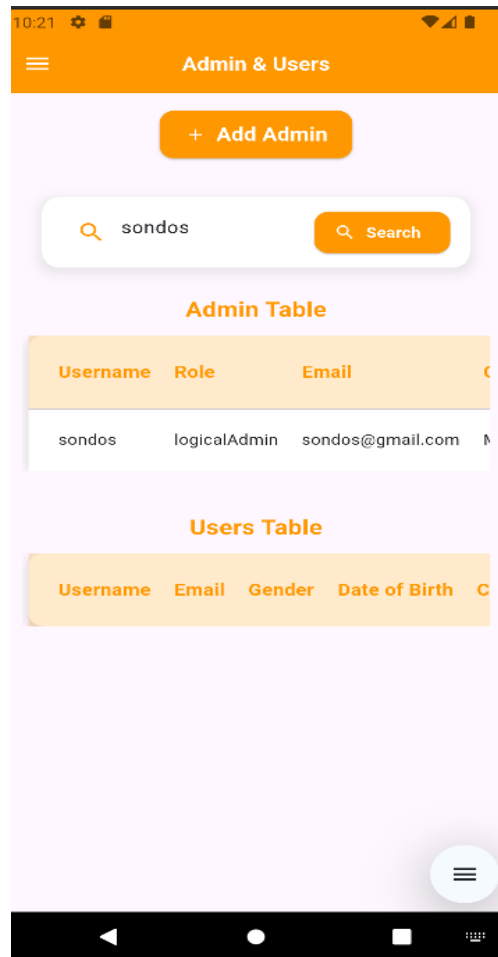
11:42

Admin & Users

**Users Table**

| Username | Email             | Gender | Date |
|----------|-------------------|--------|------|
| ahmad    | ah@123            | Male   | Dece |
| basel    | basel@123         | Male   | Dece |
| basell   | base1l@123        | Male   | Dece |
| lala     | lala@123gmail.com | Male   | Dece |
| sama     | sama@gmail.com    | Female | Dece |
| samaaa   | samaaa@gmail.com  | Female | Janu |
| samah    | samah@gmail.com   | Female | Janu |
| samera   | samera@123        | Male   | Dece |
| saraaa   | saraaaa@gmail.com | Female | Janu |
| yazan    | yazan@gmail.com   | Male   | Dece |

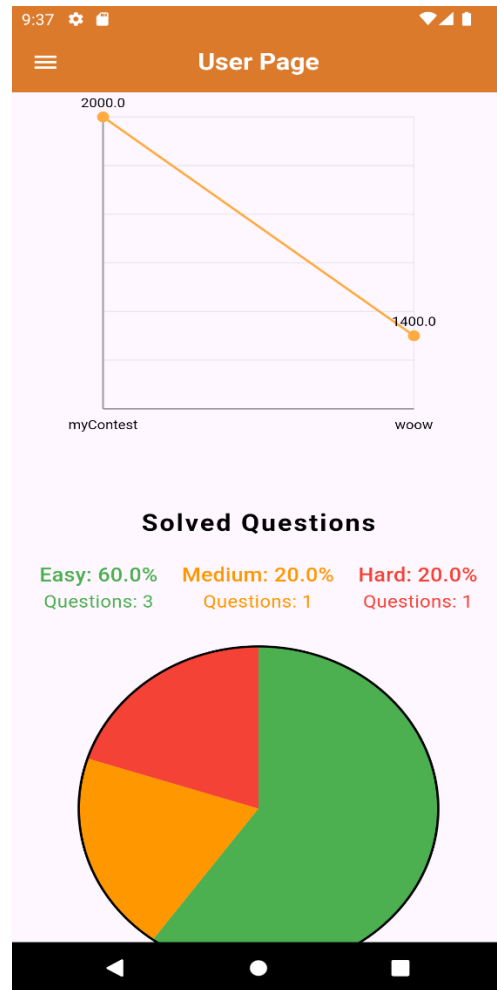
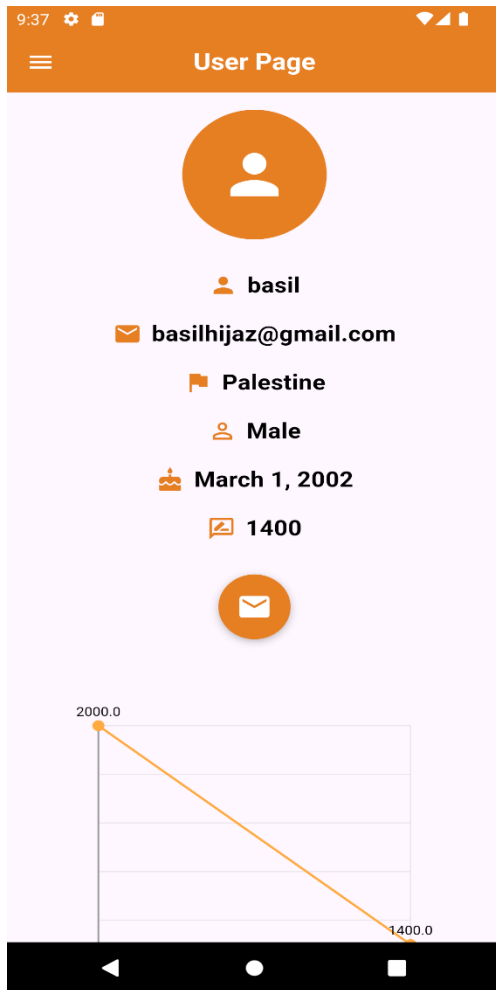
If I want to quickly access any admin or user and view their information, I can search for them by name. Once found, I can click on their name to enter their profile pa



## Profile page

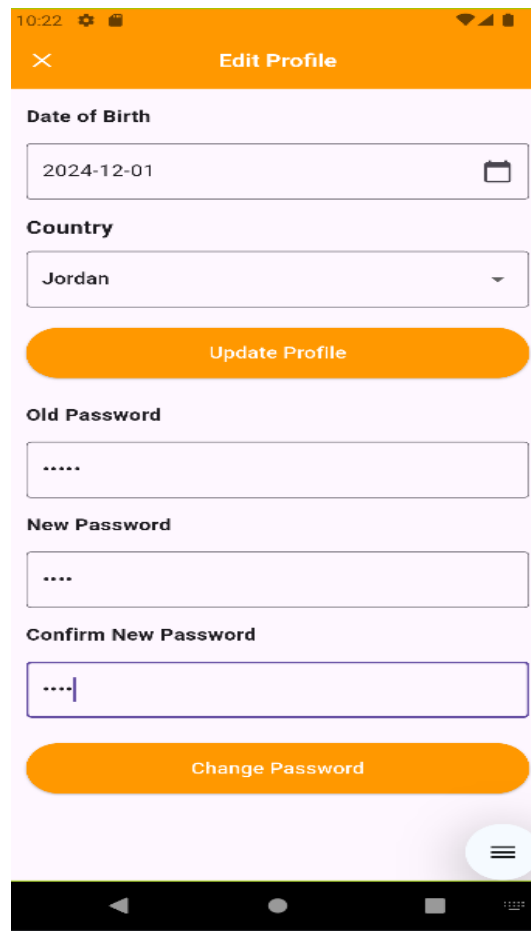
When a user logs in, they're taken to their profile page. This page shows their personal information, their rating based on the points they've earned in competitions, and a pie chart breaking down the types of questions they've solved (easy, medium, and hard). They can also see a list of the questions they've solved, the ones they've created, and the ones they've reviewed.

At the top of the page, there's a menu where the user can easily navigate to different sections like chatting, contests, adding a question, browsing questions, and viewing the latest answers.



## Edit profile

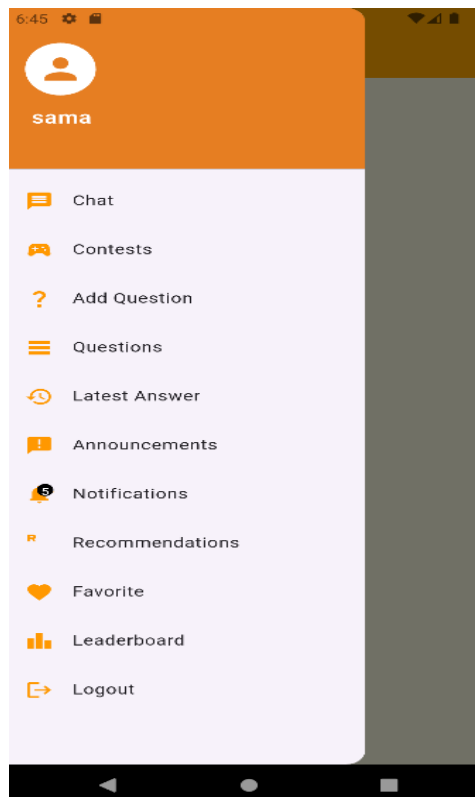
Users can update their information by clicking the "Edit Profile" button on their profile page. This will take them to a page where they can easily make any changes to their details.



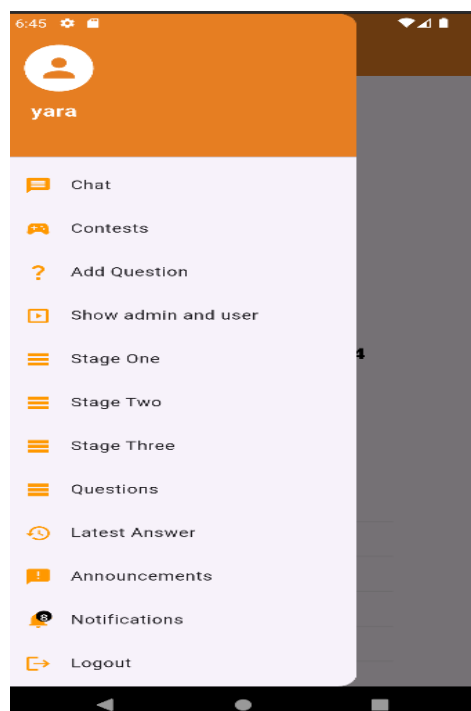
The screenshot shows a mobile application interface for editing a user's profile. The screen has a white background with orange accents. At the top, there is a status bar with the time 10:22 and various icons. Below that is a navigation bar with a close button (X) on the left and the title "Edit Profile" in the center. The main content area contains several form fields: "Date of Birth" with a date picker showing "2024-12-01"; "Country" with a dropdown menu showing "Jordan"; "Old Password" with a text input field containing "....."; "New Password" with a text input field containing "...."; and "Confirm New Password" with a text input field containing "....". Below the "Old Password" field is an orange button labeled "Update Profile". Below the "New Password" and "Confirm New Password" fields is another orange button labeled "Change Password". At the bottom right corner, there is a circular menu icon with three horizontal lines. The bottom of the screen shows the standard Android navigation bar.

## Menu

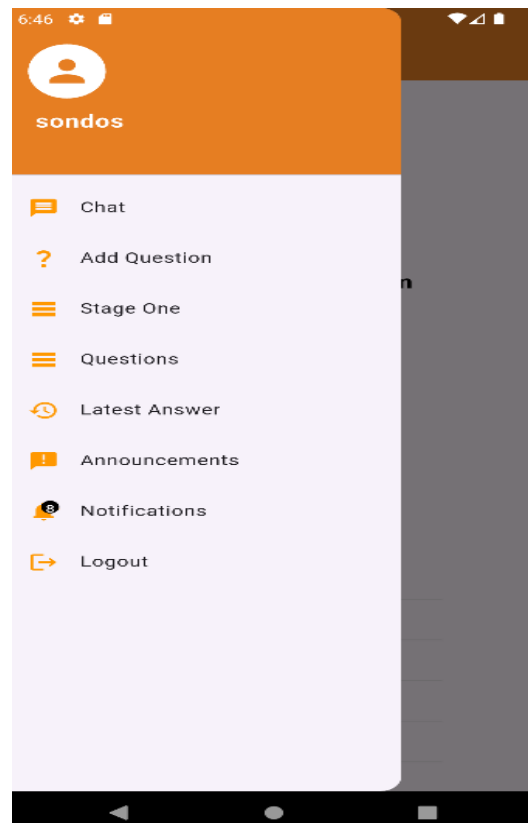
## User



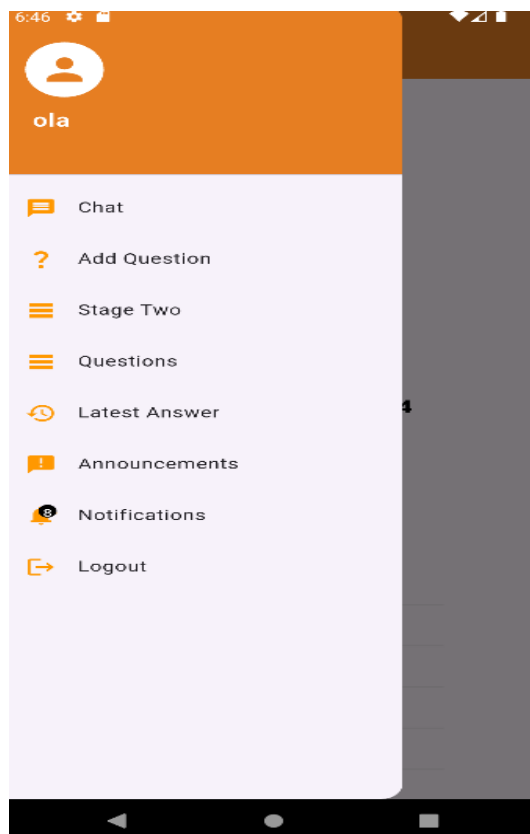
## Admin



## Logical admin

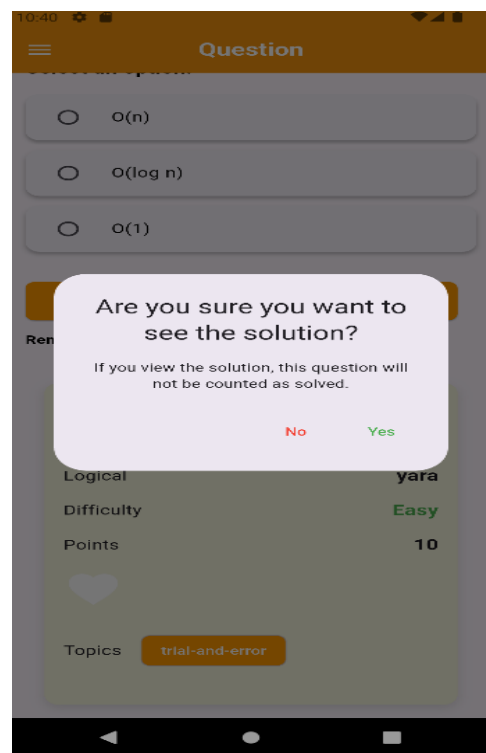
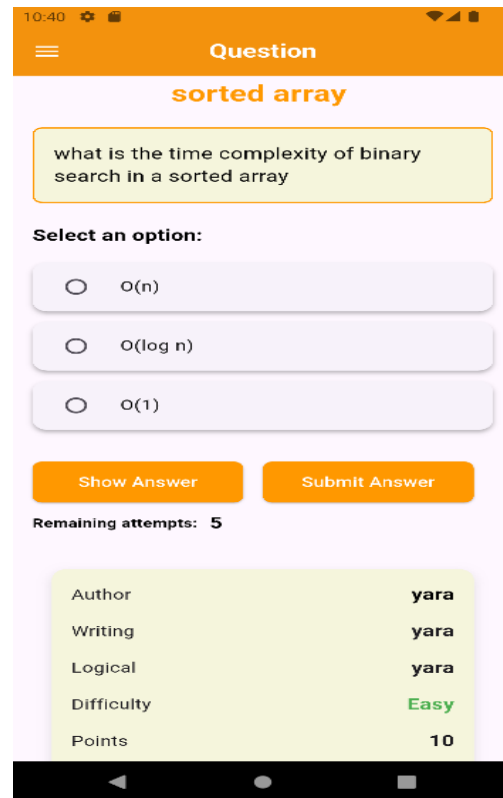
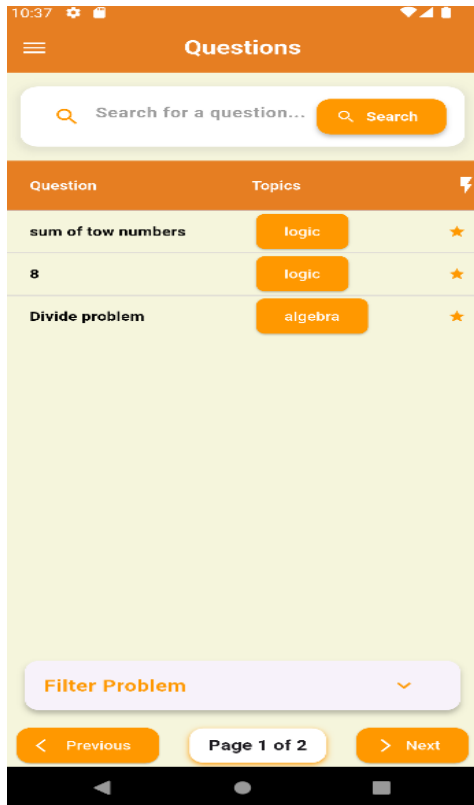


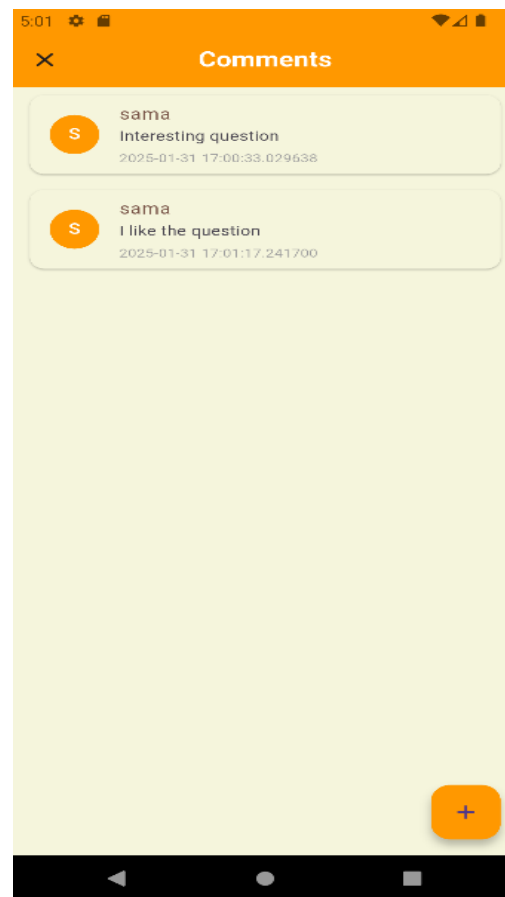
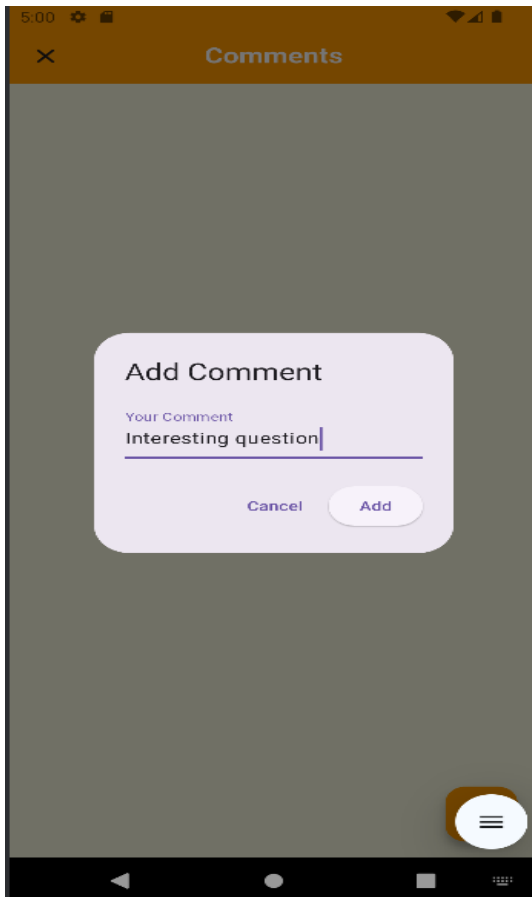
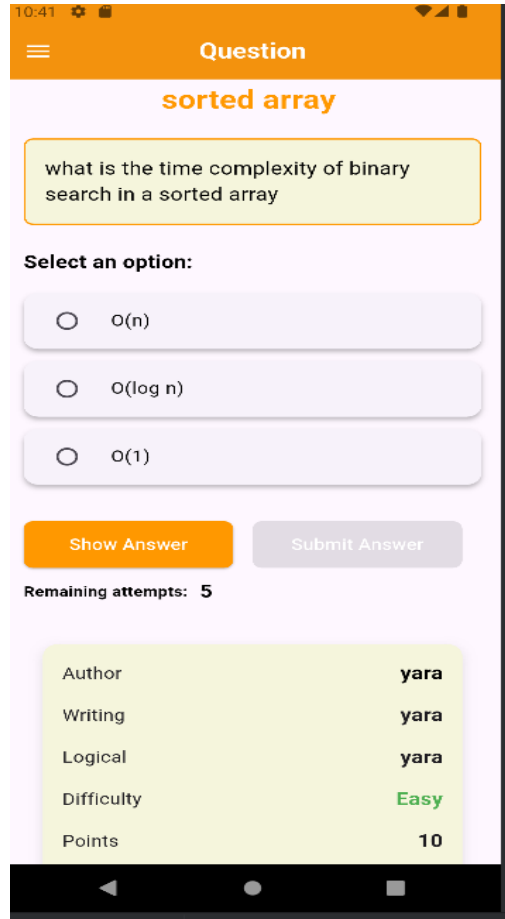
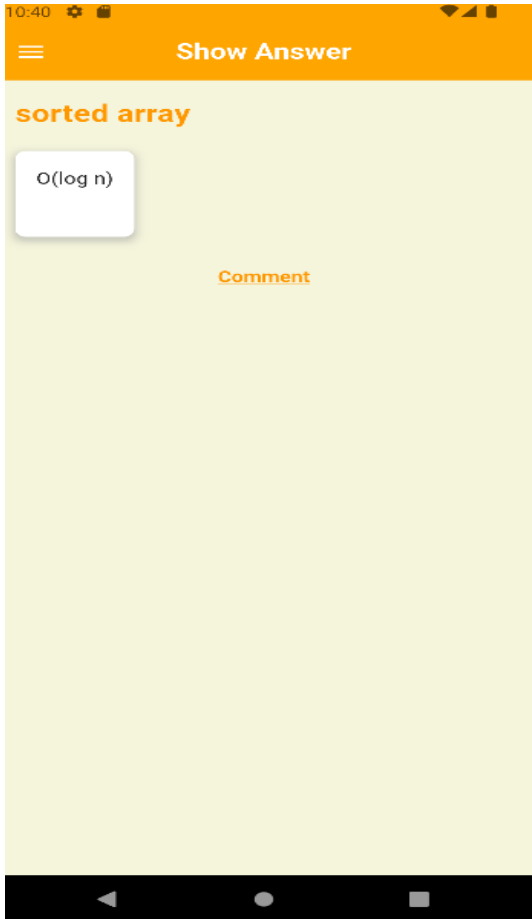
## Writer admin



## Question page & Show Question & Show Answer (user)

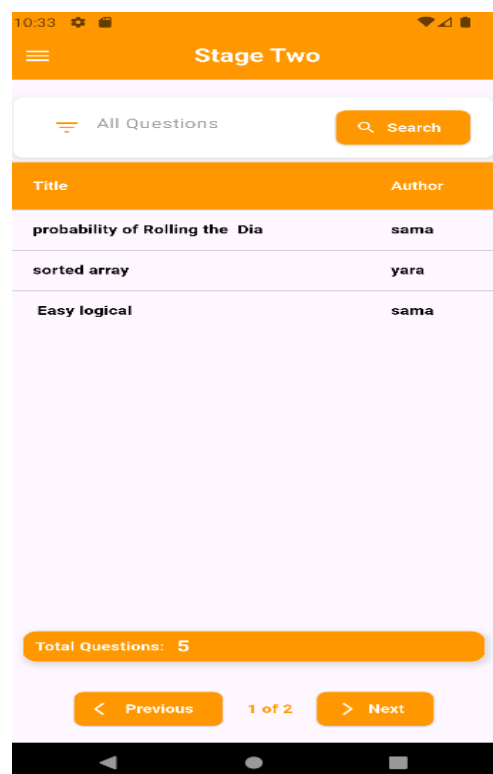
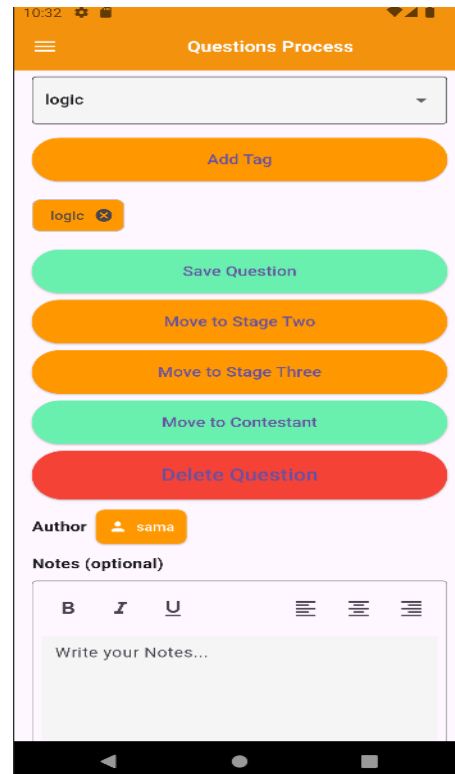
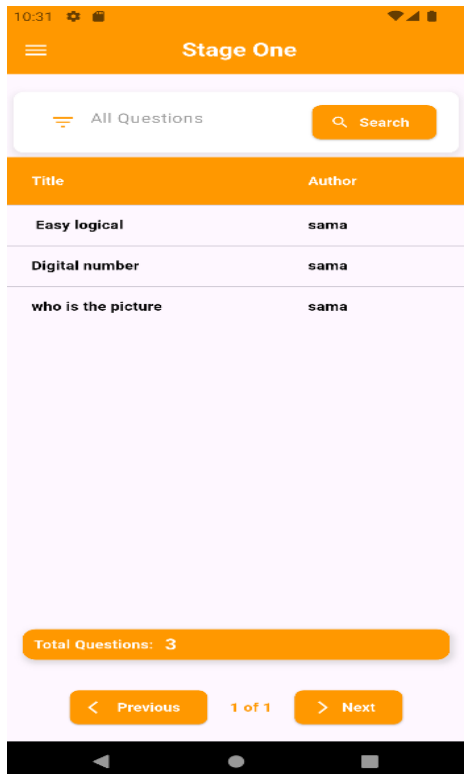
On the **Questions Page**, users can see the title, type, and how many attempts they have left for each question. Clicking on a question takes them to the **Question Page**, where they can try to answer it. Every wrong answer reduces the number of attempts. If they choose to reveal the answer by clicking "**Show Answer**", they won't be able to answer the question anymore or earn points since they already know the solution. a comment can be added to each question

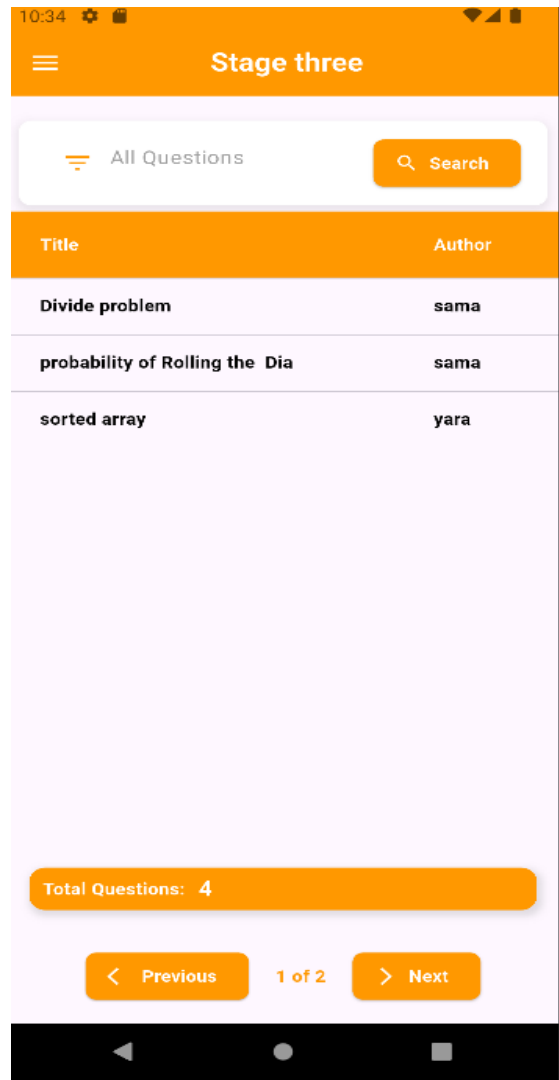
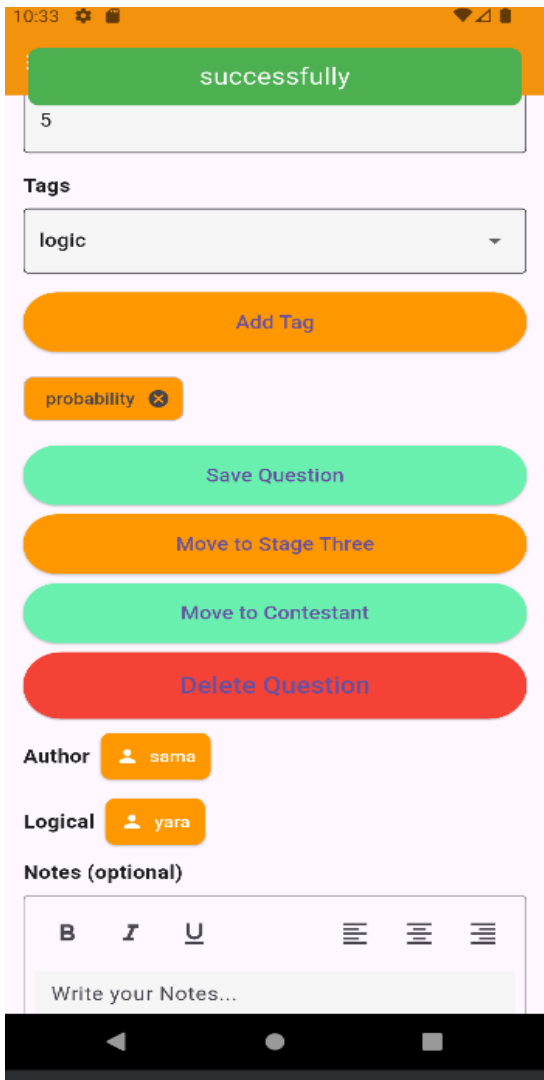




## Stage one & Stage one & stage three(admin)

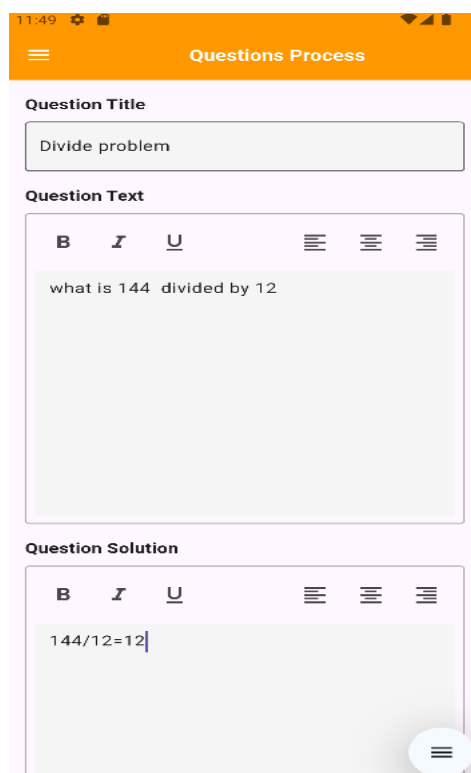
When a user submits a question, it doesn't go straight to the Questions page. Instead, it goes through a review process with three stages. First, the **Logic Admin** checks if the question makes sense logically. Then, the **Writing Admin** reviews it for grammar and clarity. Finally, the **Main Admin** gives the final approval. Once the question passes all these stages, it gets added to the Questions page.



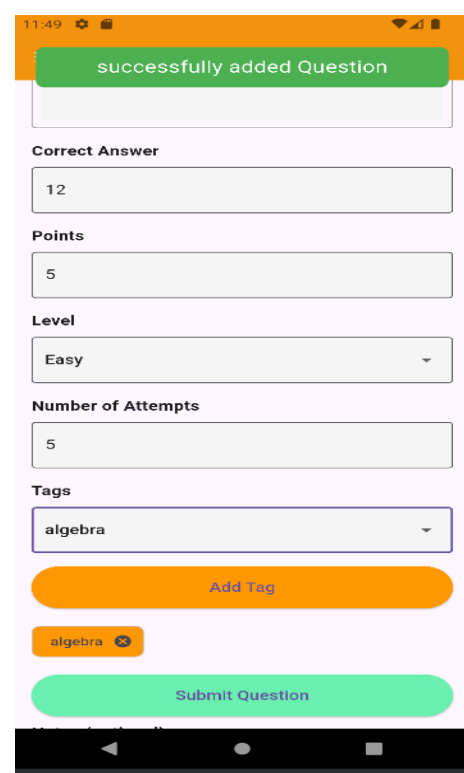


## AddQuestion&Question process

When I, as a user, want to add a question, I go to the page where I can submit it. I first select the type of question, such as multiple choice or fill in the blank, then I fill in the title, question text, and the correct answer (along with answer choices if needed). After that, I click the "Submit" button. Once submitted, the question goes into a "waiting" state, meaning it's waiting for admin approval. The admin reviews the question, and once approved, it is added to the list of questions for everyone to see and answer, Admins can also add questions to the platform. These questions go through the same review process before appearing on the **Questions Page**.



The screenshot shows the 'Questions Process' form. It has an orange header with a menu icon and the title 'Questions Process'. The form is divided into three sections: 'Question Title' with a text input field containing 'Divide problem'; 'Question Text' with a rich text editor containing 'what is 144 divided by 12'; and 'Question Solution' with a rich text editor containing '144/12=12'. Each section has a toolbar with icons for bold, italic, underline, and list creation.



The screenshot shows the continuation of the 'Questions Process' form. It features a green notification banner at the top that says 'successfully added Question'. Below this are several input fields: 'Correct Answer' (12), 'Points' (5), 'Level' (Easy), and 'Number of Attempts' (5). There is a 'Tags' section with a dropdown menu showing 'algebra' and an 'Add Tag' button. A 'Submit Question' button is at the bottom.

11:57 Questions Process

**Question Title**

Enter the question title

**Question Text**

**B I U** [List Icons]

Write your question here...

**Question Solution**

**B I U** [List Icons]

Write your solution here...

11:57 Questions Process

No options added yet.

**+Option**

**Points**

Enter the points

**Level**

Easy

**Number of Attempts**

Enter the number of attempts

**Tags**

logic

**Add Tag**

**Submit Question**

**Notes (optional)**

**B I U** [List Icons]

11:57 Add Question

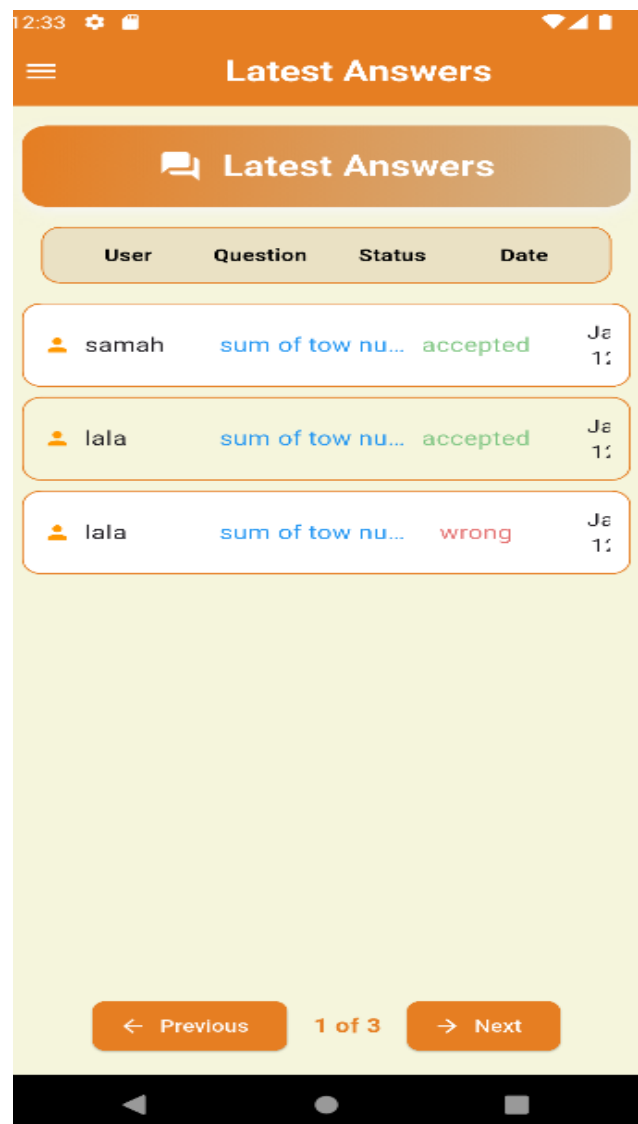
**+ Add Question**

Search for a title **Search**

| Question               | Status  | Date      |
|------------------------|---------|-----------|
| probability of Rolling | waiting | 27-01-202 |
| Divide problem         | waiting | 27-01-202 |
| name                   | waiting | 12-12-202 |

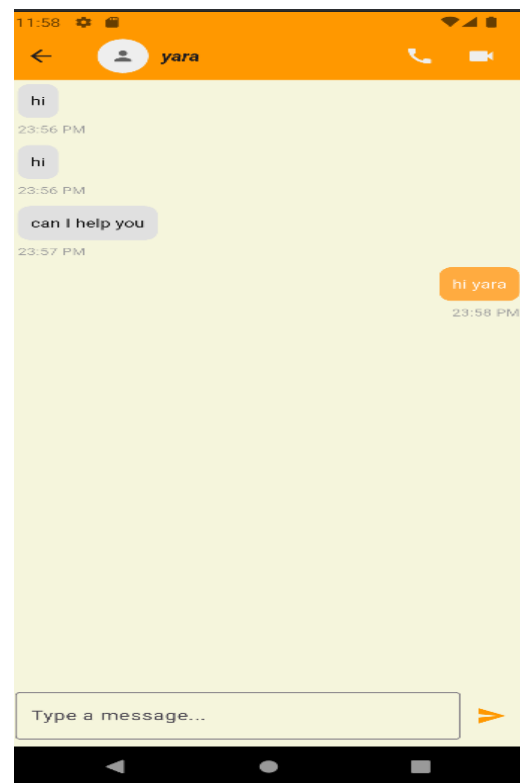
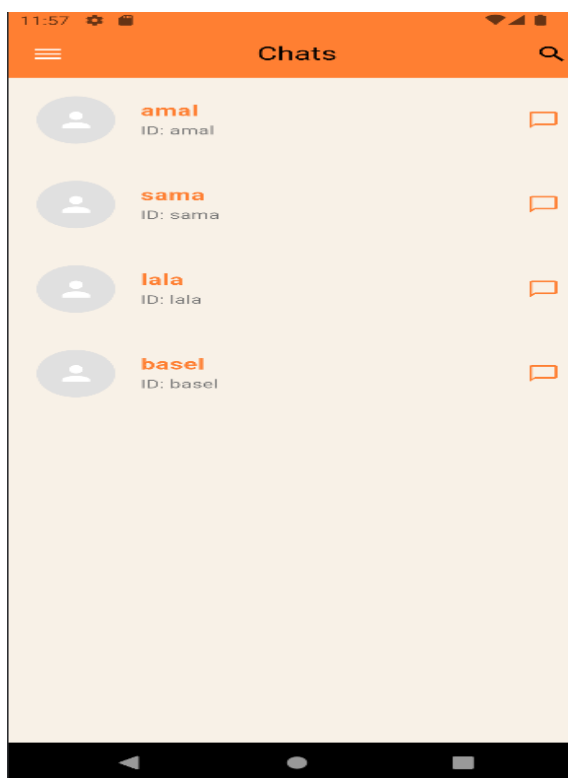
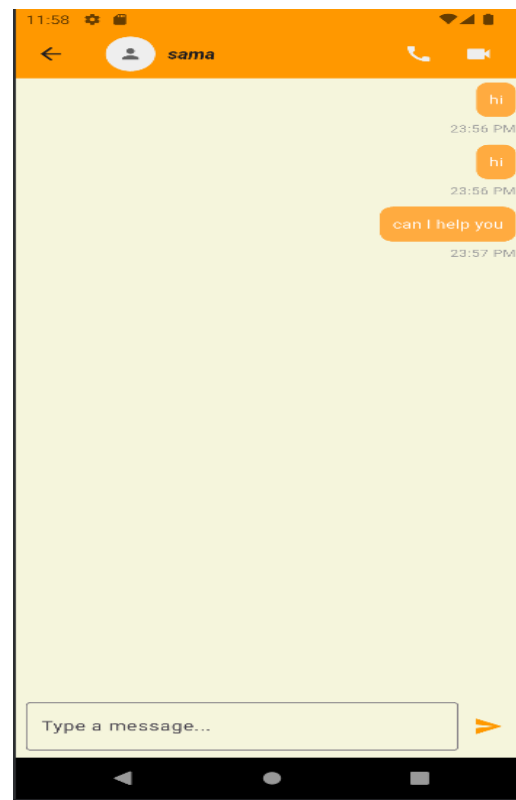
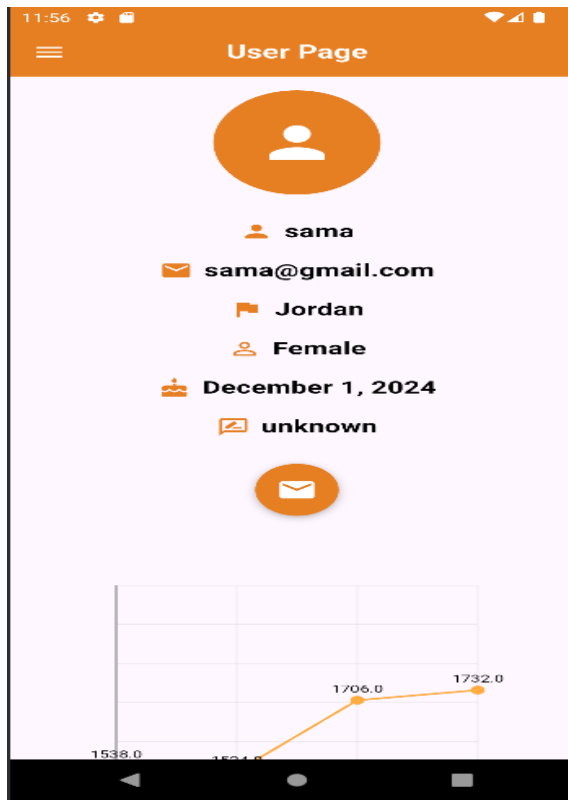
## Latest answer

In the "Latest Answers" page, the names of the most recent users who have answered questions are displayed.



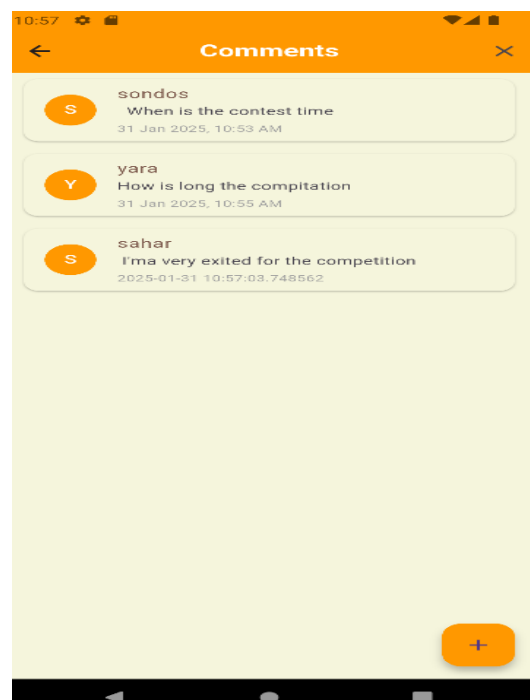
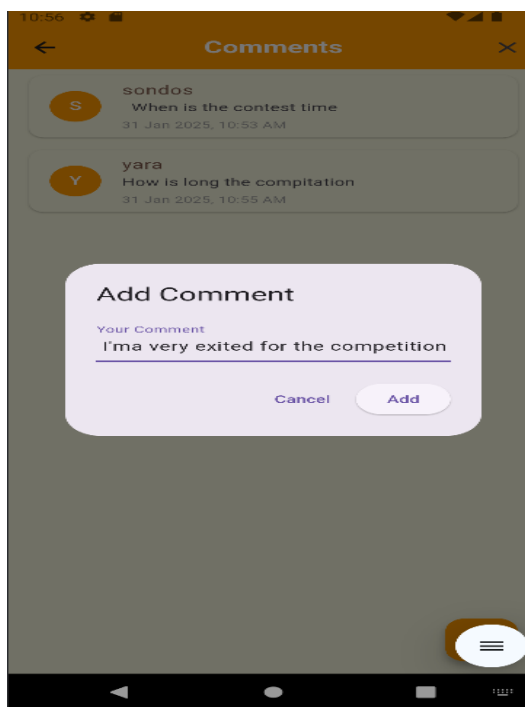
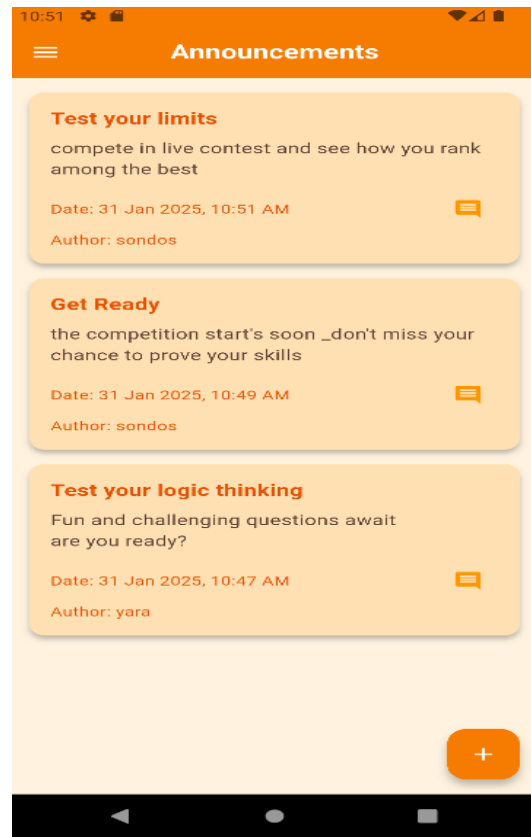
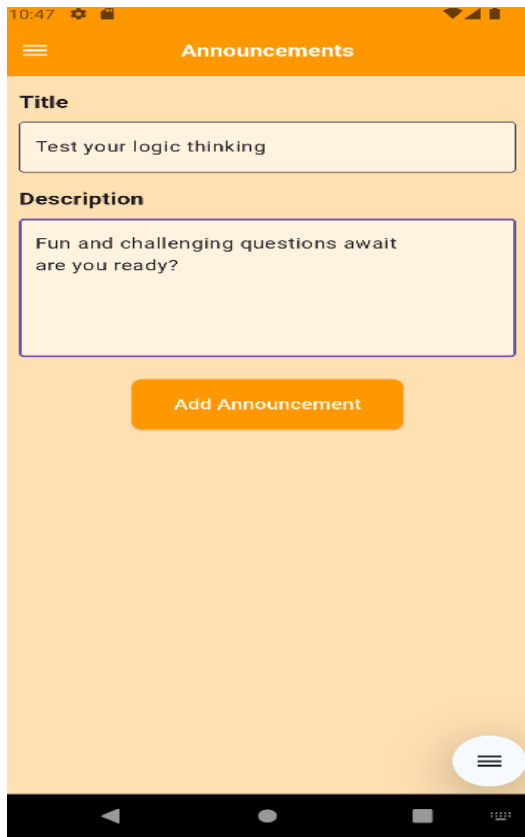
## Chats page

Admins can communicate with any user, and admins can also communicate with each other. Users can message each other as well. This is done by opening the profile of the person they want to contact and clicking on the message icon. A message is automatically sent, starting the conversation. A **chat page** then shows all the people the user has communicated with.



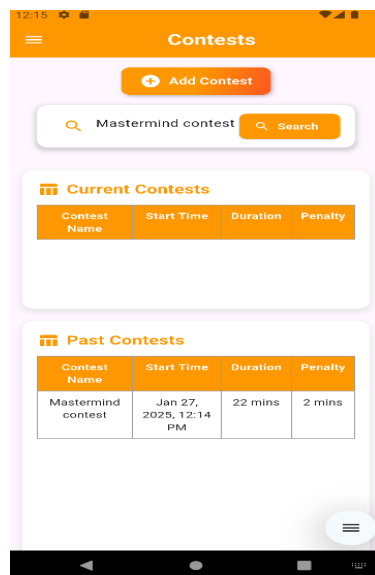
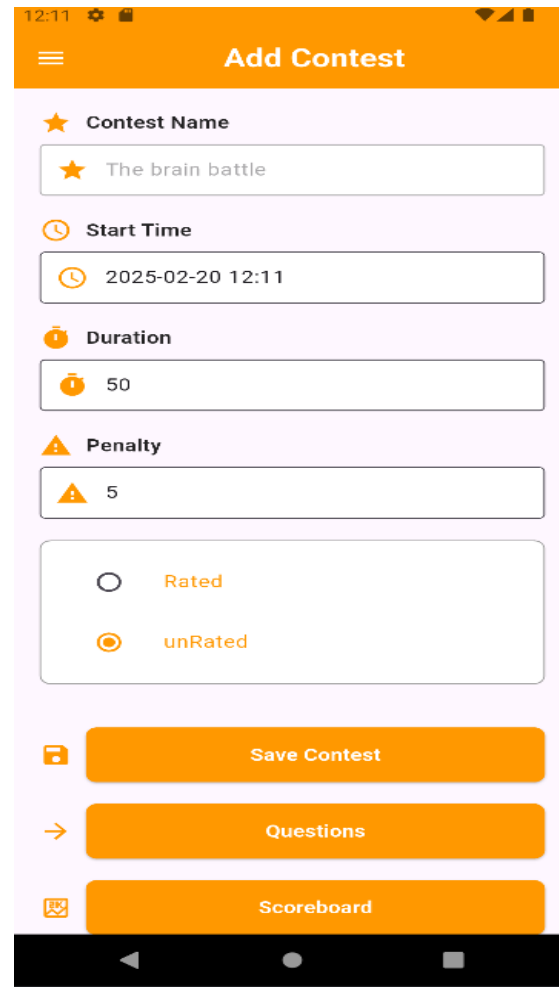
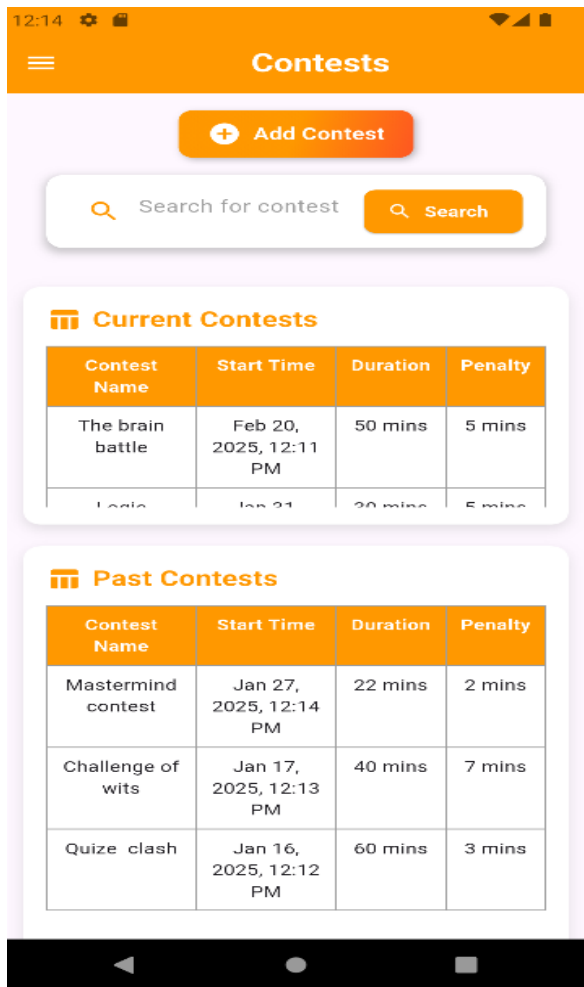
## Advertisement

As an admin, I can add an advertisement by clicking on the ad icon, writing the ad content, and it will appear on the advertisements page. Users can also comment on the ad by clicking on the comment icon specific to the ad, writing their comment, and the comment will appear on the same page for others who have commented on this ad.



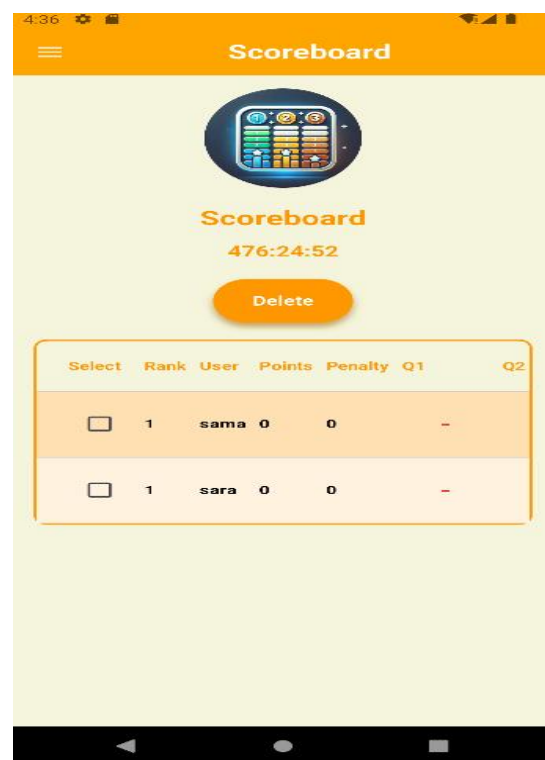
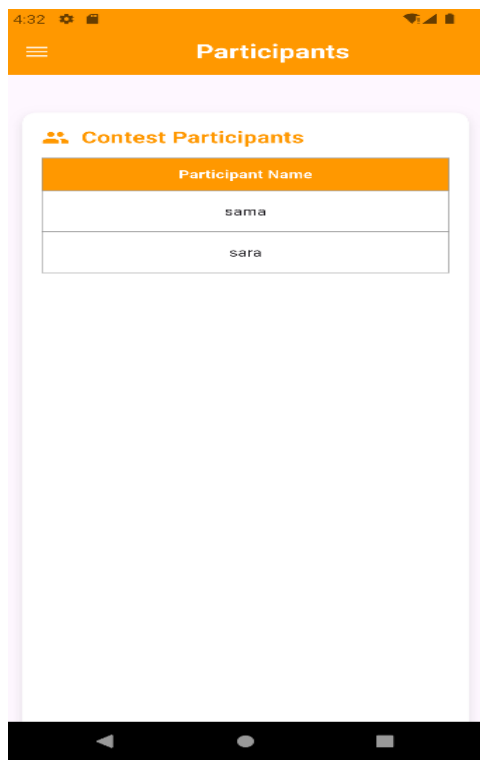
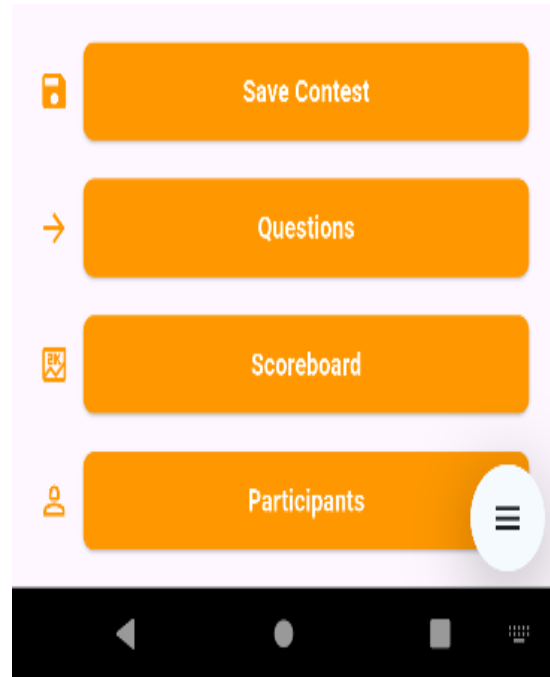
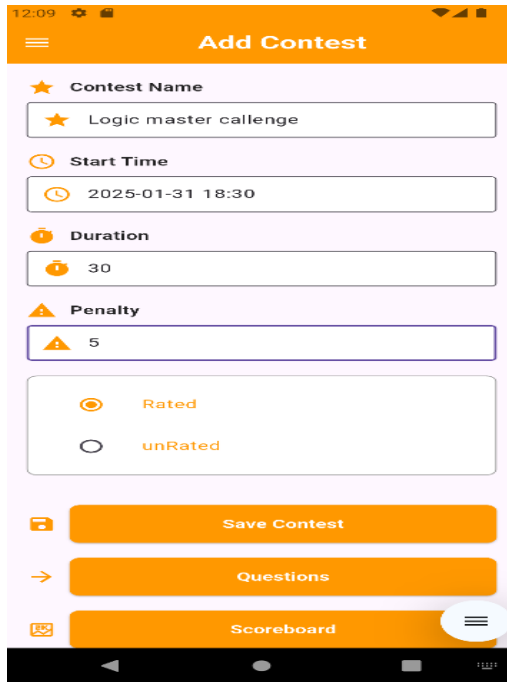
## Contest page(admin)

In the main contest page, I display the current and past contests for the user. This includes the contest name, contest time, penalty, and duration. When clicking on the contest name, it takes me to the contest page where I can edit any of these details, except for the contest name.



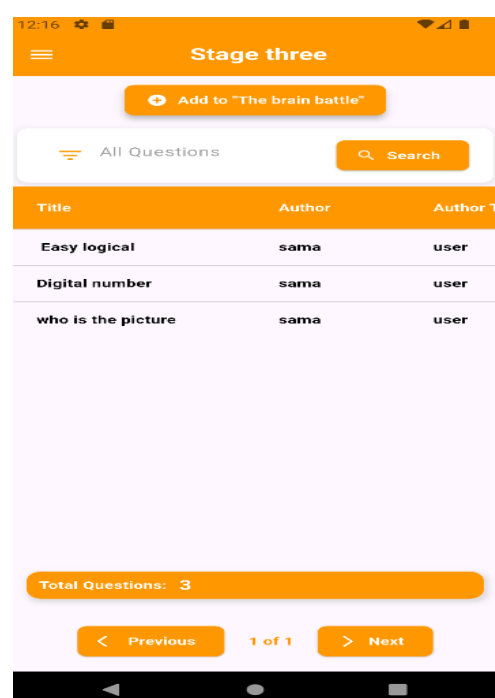
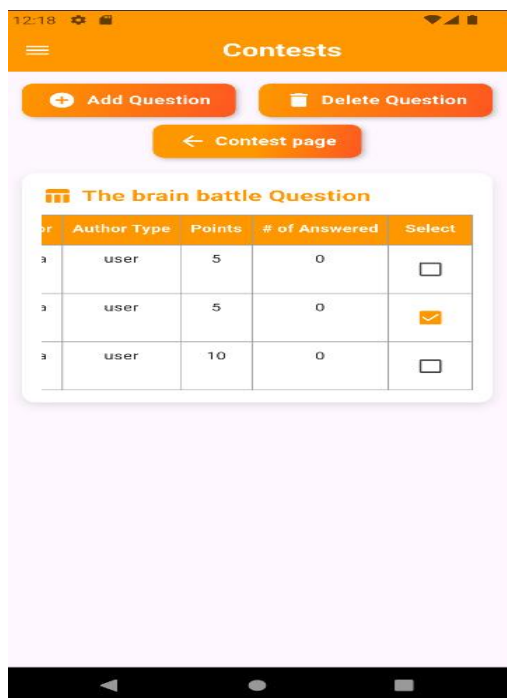
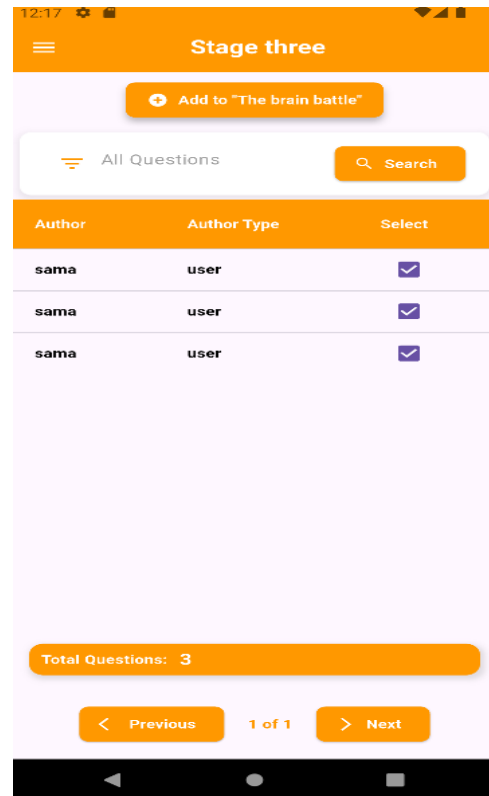
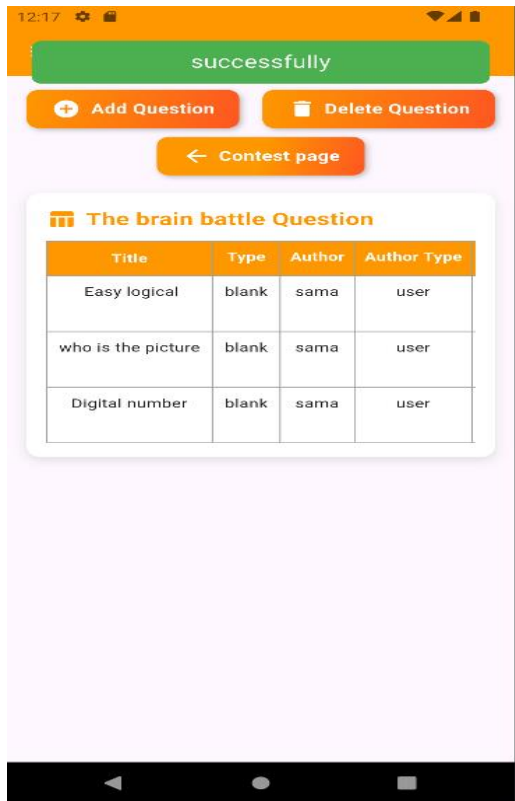
## Add contest

When I click on "Add Contest," it takes me to the add contest page where I can enter the contest name, duration, start time, and penalty. After that, I save the contest. When I click on the question, it takes me to the questions page specific to that contest. Clicking on the participants will take me to the list of participants in that contest. Finally, clicking on "Publish" makes the contest visible to the users. The "Publish" button is only available in past contests.



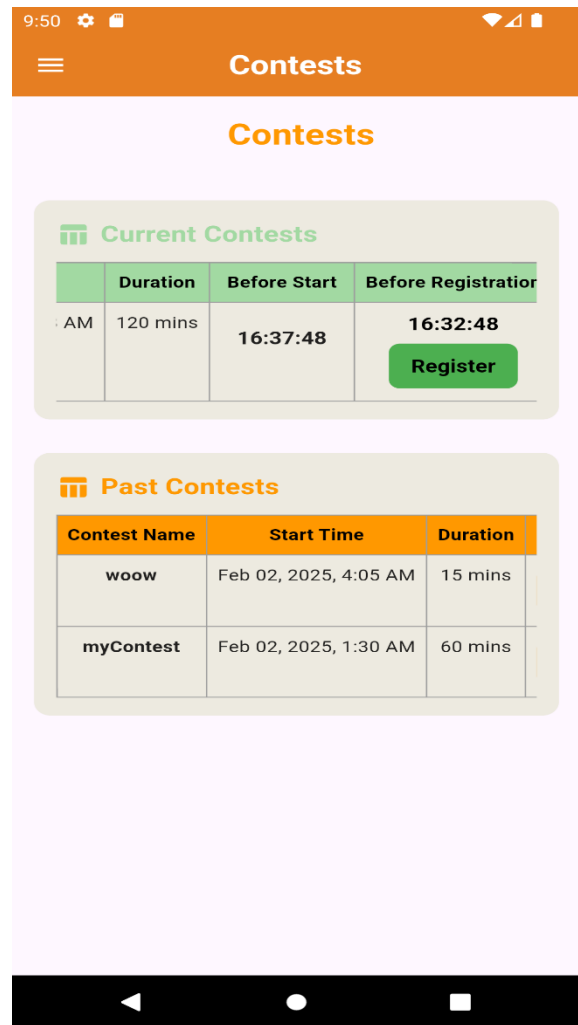
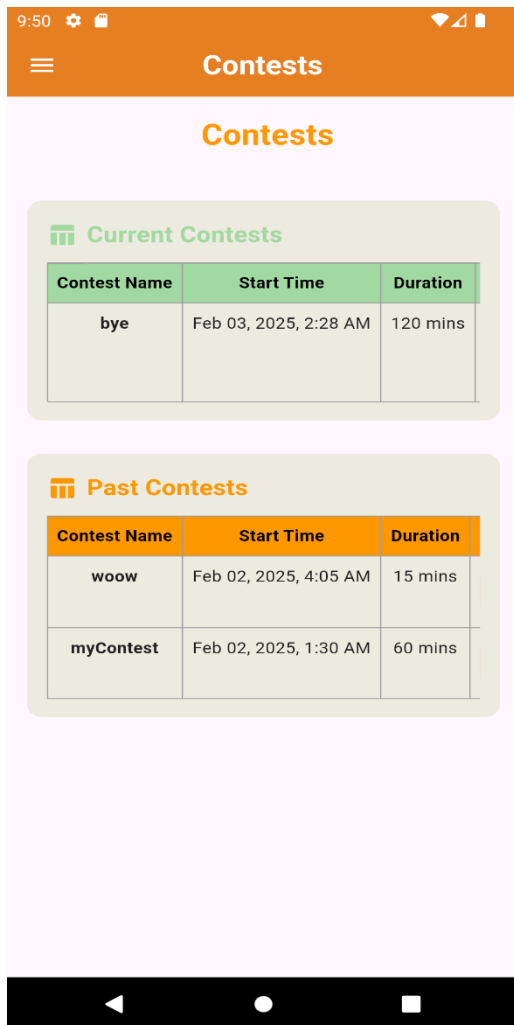
## Question contest

When I click on "Add Contest," it takes me to the add contest page where I can enter the contest name, duration, start time, and penalty. After that, I save the contest. When I click on the question, it takes me to the questions page specific to that contest. Clicking on the participants will take me to the list of participants in that contest. Finally, clicking on "Publish" makes the contest visible to the users.




## Contest page(user)

The user can see both past and current contests; for past contests, they can click "Enter" to access and solve the questions, while for current contests, they must click "Registration" to join. Additionally, from the questions, the user can navigate to the leaderboard for that contest.



9:50 9:50

**Experimental**




**Questions**  
active  
00:00:00

[Go to Scoreboard](#)

| Question         | #of solution |
|------------------|--------------|
| Omar & coins     | 3            |
| Flipping coins   | 4            |
| Number 9         | 2            |
| Distinct numbers | 1            |
| sorted array     | 2            |

9:50 9:50

**Scoreboard**



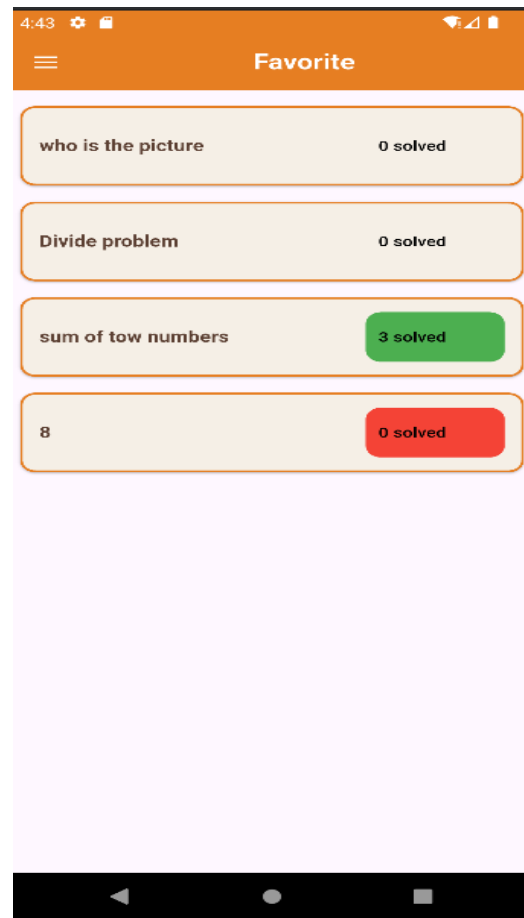
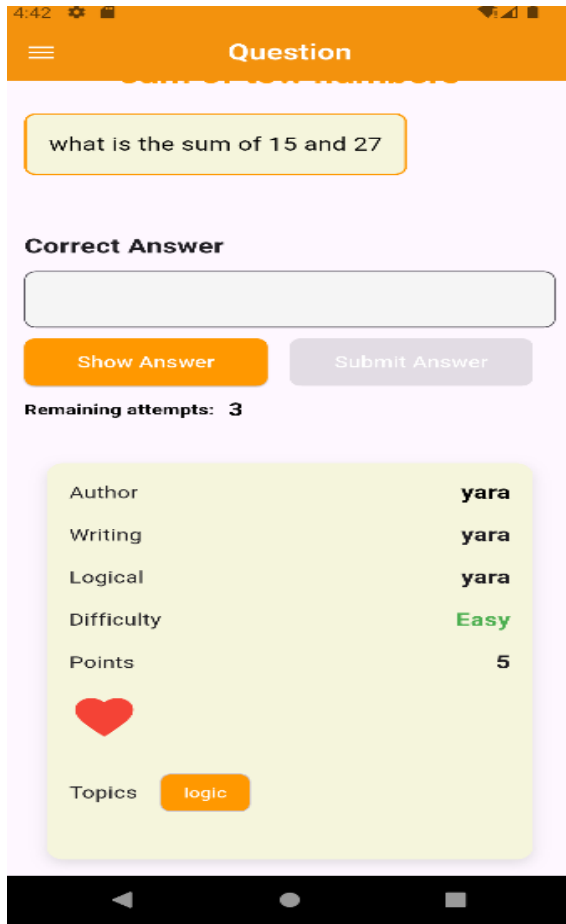
**Scoreboard**  
00:00:00

[Go to Questions](#)

| Rank | User     | Points | Penalty | Q1          | Q2          |
|------|----------|--------|---------|-------------|-------------|
| 1    | ruba     | 5      | 103     | +<br>00:02  | +3<br>00:02 |
| 2    | suhaib   | 3      | 15      | +<br>00:05  | +<br>00:05  |
| 3    | basil    | 1      | 11      | -2<br>00:01 | +1<br>00:01 |
| 4    | mohammad | 1      | 14      | -           | -           |

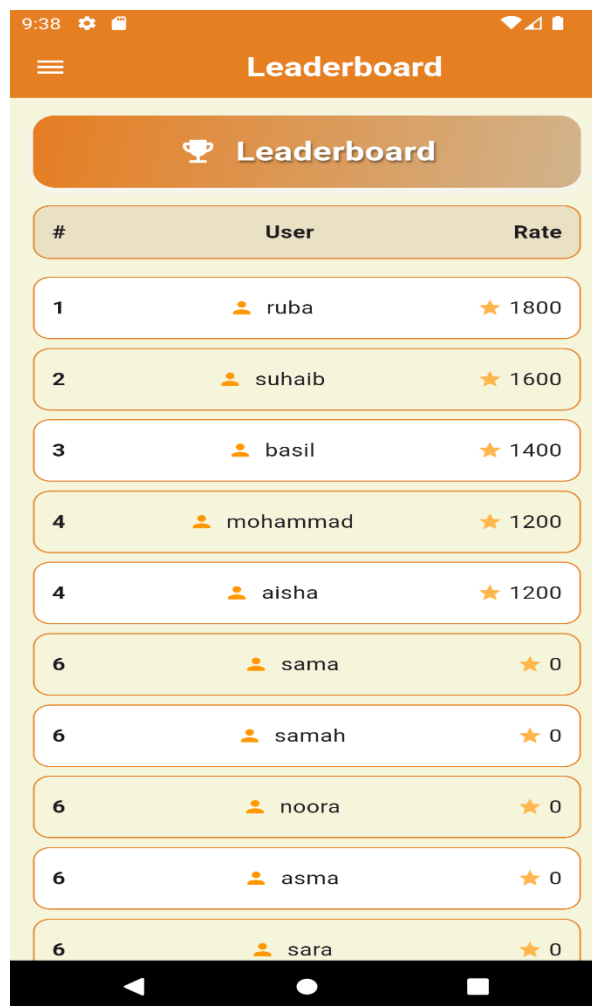
## Favorites page











The **Favorites Page** allows users to save questions by clicking on the white heart, adding them to their favorites. On the Favorites Page, the saved questions are displayed along with the number of times they have been solved.



## Leaderboard page

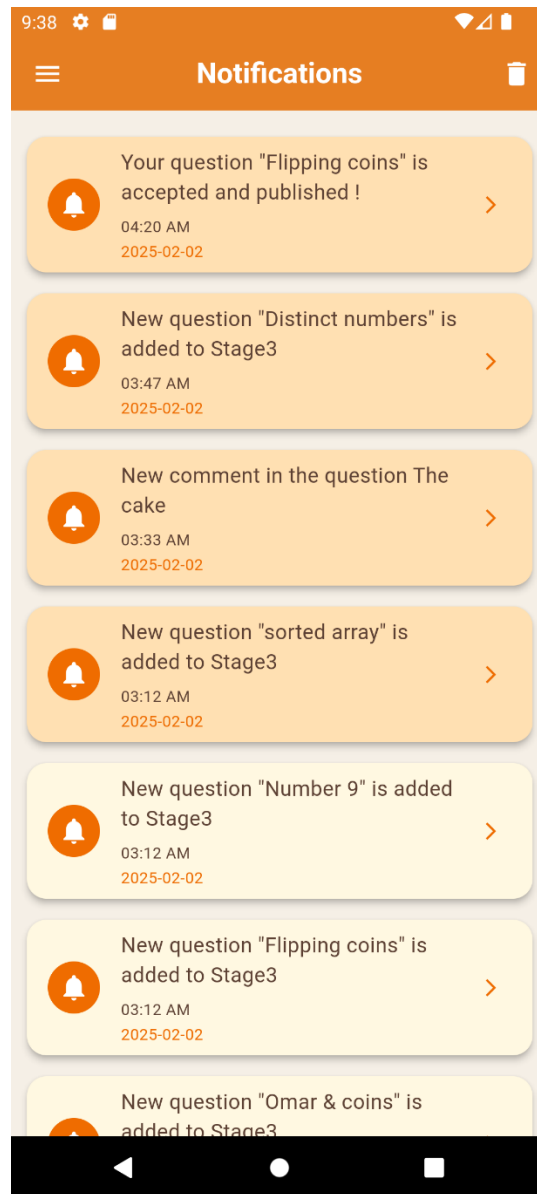
This page displays the usernames along with the rating of each user.



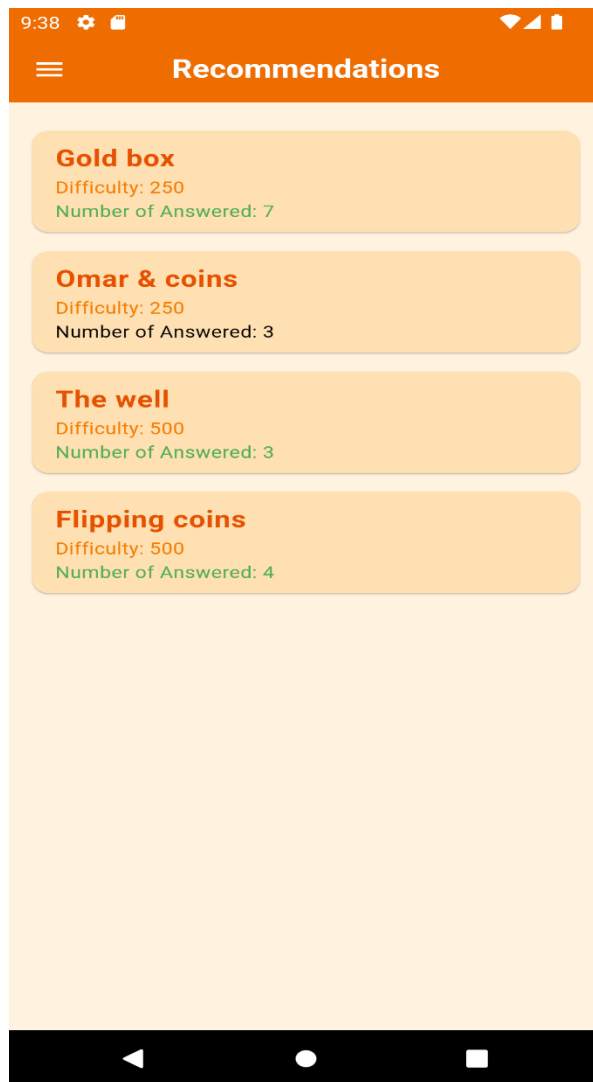
| # | User   | Rate   |
|---|--|--------|
| 1 |  ruba     | ★ 1800 |
| 2 |  suhaib   | ★ 1600 |
| 3 |  basil    | ★ 1400 |
| 4 |  mohammad | ★ 1200 |
| 4 |  aisha    | ★ 1200 |
| 6 |  sama     | ★ 0    |
| 6 |  samah  | ★ 0    |
| 6 |  noora  | ★ 0    |
| 6 |  asma   | ★ 0    |
| 6 |  sara   | ★ 0    |

## Notifications page

The user receives a notification when they submit a question, and the admin either approves or rejects it. Additionally, all users get notifications when the admin creates a contest or posts an announcement. The **Logical Admin** receives notifications for submissions at **Stage 1**, while the **Writer Admin** gets notified for submissions at **Stage 2**. Also, whenever someone comments, the admin receives a notification.

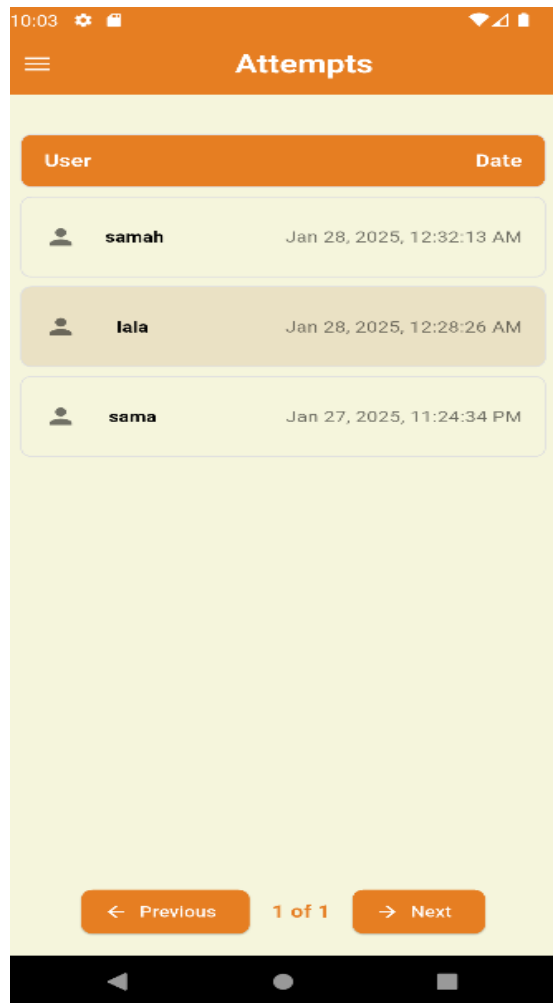


## Recommendation Page



## Attempts

When clicking on the number of attempts, it should navigate to the "Attempts" page, which shows the users who solved this question and the date.



# Chapter 5

## Results & Discussion

### 5.1 Final Application

The final application successfully integrates all the required functionalities, providing an intuitive and efficient user experience. The system allows users to answer logical questions, track their progress, and participate in contests. Additionally, the multi-stage approval process for question submissions ensures the quality and accuracy of content. The messaging system facilitates seamless communication between users and administrators, enhancing engagement within the platform.

Key features of the final application include:

- **Question Management System:** Users can view questions, attempt answers, and receive immediate feedback.
- **Multi-Stage Question Approval:** Ensures that questions are logically and grammatically verified before being published.
- **User and Admin Communication:** Enables interaction through direct messaging within the platform.
- **Contest System:** Users can participate in time-based challenges, with rankings displayed on a live scoreboard.
- **User Profiles & Statistics:** Each user has a profile displaying their achievements, solved questions, and rankings.

The application is designed to be scalable, ensuring smooth performance as the user base grows. Feedback from early testers indicates that the platform meets its intended objectives effectively.

### 5.2 Project Outcomes

The project has achieved its primary goals, successfully implementing a platform that enhances logical thinking through interactive problem-solving. The following outcomes have been observed:

1. **Enhanced User Engagement:** The interactive question-answer system encourages users to think critically and engage with various problem-solving techniques.

2. **Structured Content Management:** The multi-stage review process ensures high-quality questions, maintaining the credibility of the platform.
3. **Seamless Communication:** The integrated messaging system facilitates better interaction between users and administrators.
4. **Accurate Performance Tracking:** The leaderboard and user statistics provide motivation and allow users to track their progress.
5. **Scalability & Future Enhancements:** The modular design of the application allows for future enhancements, such as additional question categories and AI-driven recommendations.

In conclusion, the project successfully meets its objectives and provides a solid foundation for further improvements. Future iterations could include more personalized learning experiences and expanded question types to cater to a wider audience.

# Chapter 6

## Conclusion & Future Work

### 6.1 Conclusion

The project aimed to create an interactive platform that promotes logical thinking through structured problem-solving. By integrating a question management system, a multi-stage review process, and interactive user engagement features, the platform has successfully achieved its objectives. Users benefit from a well-organized learning experience, while administrators can efficiently manage and curate high-quality content. The application's scalability ensures long-term usability and potential for further development.

### 6.2 Future Work

To enhance the platform further, several improvements and new features can be considered:

- **Expanded Question Categories:** Introducing new types of logical and analytical questions to diversify learning opportunities.
- **AI-Based Recommendations:** Implementing machine learning to suggest personalized questions based on user performance.
- **Enhanced Contest Features:** Adding new contest formats, team-based challenges, and real-time collaboration.
- **Gamification Elements:** Introducing badges, achievements, and reward systems to enhance user motivation.

By incorporating these future improvements, the platform can continue to evolve, providing an even richer and more engaging learning environment for users.

# Bibliography

- **Flutter Documentation**

Flutter: Beautiful native apps in record time. Retrieved from <https://flutter.dev>

- **Firestore Documentation**

Firestore: Real-time database and notifications. Retrieved from <https://firebase.google.com>

- **MongoDB Documentation**

MongoDB: The NoSQL Database. Retrieved from <https://www.mongodb.com>

- **Node.js Documentation**

Node.js: JavaScript runtime built on Chrome's V8 engine. Retrieved from <https://nodejs.org>

- **HackerRank Platform Overview**

HackerRank: Coding challenges and competitions. Retrieved from <https://www.hackerrank.com>

- **Educational Gamification Techniques**

Zichermann, G., & Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media.

- **Learning Platforms and Logical Thinking**

Anderson, L. W., & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon.

- **Live Leaderboards in Competitive Systems**

Millington, I. (2019). *Game Physics Engine Development*. CRC Press.

- **User Experience Design for Education Platforms**

Norman, D. A. (2013). *The Design of Everyday Things*. Basic Books.

- **Problem-Solving and Logical Thinking**

Polya, G. (2004). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.

