CSE-2216 - Application Development Lab Project Report

Project Name: KodeShell

Team Name: 3Ds

Submitted By

Md. Mahmudul Hasan (Roll - 10)

Md. Istahak Islam (Roll - 16)

Md. Rifat Khan (Roll - 38)

27th Batch

Department of Computer Science & Engineering

University of Dhaka

Submitted To

Md. Fahim Arefin

Md. Tanvir Alam

Md. Mahmudur Rahman

Submitted On

6 November 2023

Table of Contents

Introduction

1.1 About the App

Competitive programming has evolved into a highly demanding arena for software developers, offering an avenue to challenge and hone their problem-solving skills. With a multitude of competitive programming platforms available and an ever-increasing number of contests, managing one's performance and keeping abreast of upcoming events has become a daunting task. In response to these challenges, we embarked on a mission to create a comprehensive solution that streamlines the competitive programming experience. The result of our endeavor is the "KodeShell" mobile application.

"KodeShell" is designed to address the multifaceted challenges faced by competitive programmers, from novices to seasoned experts. This mobile application aims to revolutionize the way programmers engage with competitive programming, enhancing their journey with convenience and efficiency.

Our application offers a unified dashboard to track one's progress across various competitive programming sites. It eliminates the need to navigate through multiple websites and platforms, simplifying the process of monitoring current state and staying informed about upcoming contests.

Moreover, "KodeShell" recognizes the importance of community in the world of competitive programming. It provides an environment where users can search and stalk other programmers across different platforms, facilitating knowledge sharing, mutual growth, and healthy competition.

In addition to simplifying contest management, "KodeShell" goes the extra mile by providing a one-on-one messaging feature. This means you can have direct, private conversations with other programmers, making it easier to discuss problems, share solutions, or simply connect with like-minded individuals.

1.2 Motivation

As a competitive programmer, sometimes you feel overwhelmed by the amount of competitive programming sites available out there. So to keep track of your current state in all of the sites is nearly impossible. Besides who gives regular contests, they sometimes fail to keep themselves up to date with the contests that are approaching in the near future hence missing them sometimes or maybe attending them without major preparation. Also in this arena, we like to compare ourselves to other programmers who are doing better than us but to stalk their

accounts we need to go to every site and search and compare them. Also there's very little opportunity out there to seek help from others. We felt that there is a lack of a strong competitive programming community where we all can share our success, doubts, solutions with others and grow together as a community. From these difficulties, we felt that if we can get them in a nutshell just under a simple mobile app then that would make the life of every competitive programmer very smooth and pleasing. Hence, we decided to make "KodeShell" to solve all the mentioned problems and many more.

1.3 Features

Doubt and Discussion Forum:
"KodeShell" offers a dedicated space where users can post their coding doubts, questions, or engage in discussions on various programming topics. This feature fosters a vibrant community where programmers can seek help, share their knowledge, and collaborate with others.

User Comments and Assistance:
In response to posted doubts and discussions, fellow users can provide comments and assistance, helping those in need to find solutions, gain insights, and learn from experienced programmers. This collaborative environment enhances learning and problem-solving.

User Search and Profile View:
"KodeShell" enables users to search for other members who have accounts on the app. Upon finding a user, you can view their profile, including their competitive programming achievements, contests participated in, and more. This feature encourages networking and connection among users.

Event Updates:
Stay informed about major competitive programming events like ICPC, IUPC, and NCPC contests. "KodeShell" provides timely updates, ensuring that you don't miss out on these significant opportunities.

Upcoming Contest Listings:
Discover and access a comprehensive list of upcoming contests across various competitive programming sites. Users can access contest details, go to the contest site through provided links, set reminders in their calendars, or enable notifications within the app. This feature simplifies contest management and helps users plan their participation efficiently.

Profile Stalking:
Perhaps the most intriguing feature, "KodeShell" allows you to explore and "stalk" the profiles of competitive programmers on popular platforms such as Codeforces,

AtCoder, and LeetCode, even if they aren't using the "KodeShell" app. This provides an easy way to track and compare your progress with top programmers.

Personal Profile Integration:
Users can integrate their Codeforces, AtCoder, and LeetCode profiles within the app. This feature enables them to track their improvement, statistics, and achievements across these platforms, all in one place.

One-to-One Messaging:
"KodeShell" facilitates direct, private communication between users. The one-to-one messaging system is an excellent tool for solving doubts, seeking assistance, and sharing insights. It offers a seamless way to connect with other programmers and build a supportive community.

1.4 Tools, Technologies & Frameworks used

In the course of developing this Android project, We utilized a combination of tools, technologies, and frameworks to achieve the desired functionality and seamless integration with external coding platforms and databases to provide a smooth user experience.

Android Studio: We employed Android Studio as the primary integrated development environment (IDE) for Android application development. It provided a comprehensive set of tools for designing, coding, testing, and debugging.

Android SDK: The Android Software Development Kit (SDK) was essential for accessing Android-specific libraries and resources, allowing me to build Android-compatible applications.

Retrofit: We employed the Retrofit library for making HTTP requests to the APIs of Codeforces, AtCoder, and LeetCode. Retrofit simplifies network operations and data parsing.

OkHttp: OkHttp, a popular HTTP client for Android, was used alongside Retrofit to efficiently handle network operations and requests, ensuring reliable data retrieval.

JSON Parsing: For handling the JSON responses from the APIs, We implemented JSON parsing using libraries like Gson to extract relevant data for use in the application.

Version Control: We employed version control systems like Git to collaborate with team members, manage code changes, and ensure codebase integrity.

Figma: Figma is a cloud-based design tool that allows for collaboration in real-time. It's excellent for designing Android app interfaces and prototypes.

Android Studio (Layout Editor): Android Studio's Layout Editor allows for visual UI design if someone is comfortable with code, simplifying the process of creating the app's user interface.

Firebase Authentication: Firebase Authentication simplifies user sign-in and identity management, providing a secure and scalable solution for your app's authentication needs.

Firebase Realtime Database: Firebase Realtime Database enables seamless real-time data synchronization, fostering dynamic and interactive experiences in your app with its NoSQL cloud database.

1.5 Individual Responsibility

1.5.1 Md. Mahmudul Hasan

UI/UX Design:
The creation of visually appealing and user-friendly layouts, navigation flows, and interactive elements to provide a seamless and engaging user experience.

Frontend Development:
The translation of design concepts into functional code to bring the user interface to life, ensuring it behaves as intended.

App Flow and Navigation:
The definition of the overall flow and navigation of the app, considering user journeys and interconnections between different features.

Integration of Frontend and Backend:
The critical task of merging the frontend with the backend to enable effective communication between the user interface, server, databases, and APIs.

Backend Development:
Involvement in backend development to set up the app for easy adaptation to different databases and API integration, facilitating data collection and management.

1.5.2 Md. Istahak Islam

In this Android project, my primary contribution was to establish connections with popular coding platforms such as Codeforces, AtCoder, and LeetCode using their respective APIs. I implemented functionality to fetch data, including recent problems, submission of user, user profiles, and contest information, to provide users with the relevant content they needed for coding challenges and competitions. This allowed our app to seamlessly integrate with these platforms.

1.5.3 Md. Rifat Khan

Firebase Integration:

a. Implemented Firebase Realtime Database to store and sync app data in real-time.
b. Established a secure and efficient connection between the Android app and the Firebase backend.

User Authentication:

c. Integrated Firebase Authentication to enhance security and control access to the app's features.
d. Implemented user sign-up, login, and logout functionalities.

Real-time Chatting Feature:

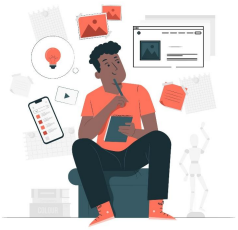
- Implemented a real-time chatting feature using Firebase Realtime Database to enable users to exchange messages seamlessly.
- Designed and developed the chat interface, allowing users to send and receive messages in real-time.

Posting and Sharing:

- Developed a posting feature, allowing users to create and share content within the app.
- Enabled commenting functionality on posts, fostering user engagement and interaction.

Design & Implementation

2.1 UI



Login

Email ID

Password

Forgot Password?

Login

OR

Login with Google

New to KodeShell? Register

Sign up

First Name

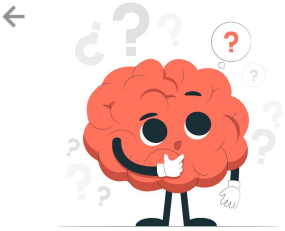
Last Name

Email ID

Phone Number

Continue

Have a Google Account? Login



Type Password

Password

Confirm Password

Register

Forgot Password?

Please enter the email address associated with your account

Email ID

Submit

KodeShell

Mohammed Mahmudul

4 days ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

+2 0

Istahak Islam

4 days ago

You are given an array of integers a1,a2,...,an.

A pair of integers (i,j), such that 1≤i<j≤n, is called good, if there does not exist an integer k (1≤k≤n) such that ai is divisible by ak and aj is divisible by ak at the same time.

Comments

Mohammed Mahmudul 4 days ago

what's your thought process?

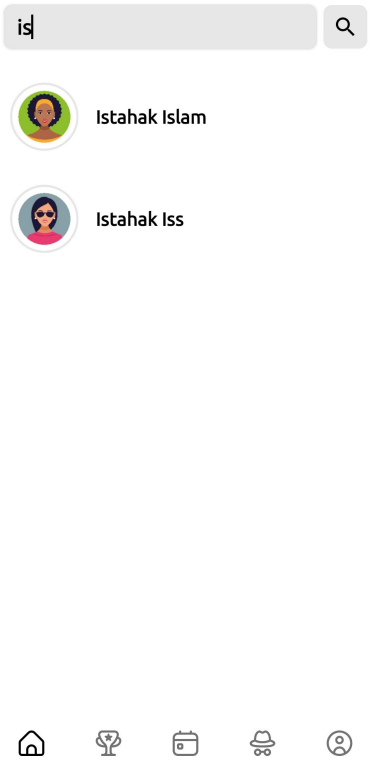
Iffatul Hossain 4 days ago

can share the source of the problem, please?

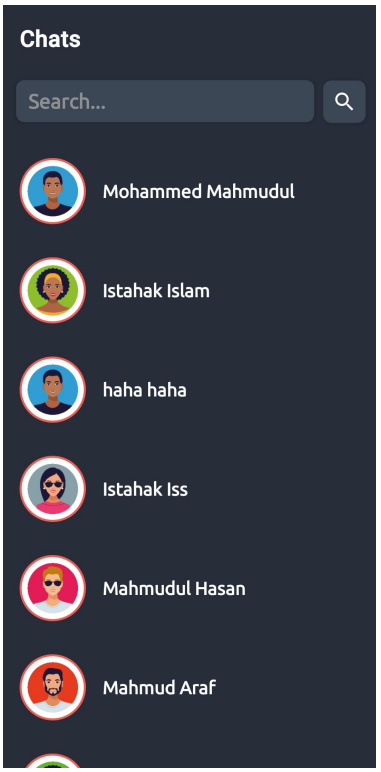
Mahmudul Hasan 4 days ago

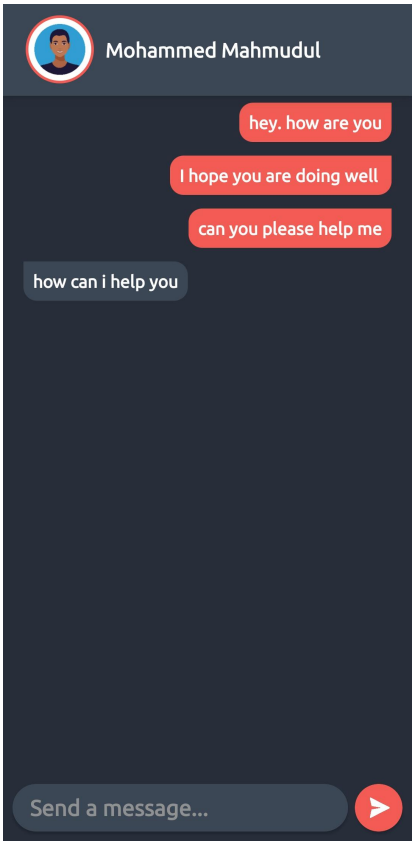
nice problem. you can do it using the idea of binary search

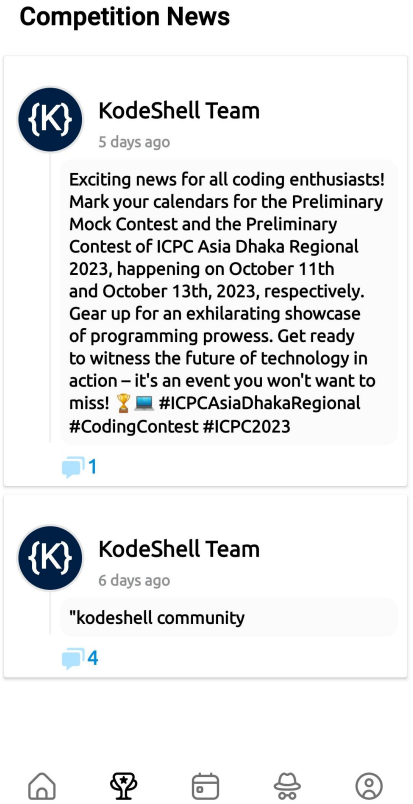
Type a comment...

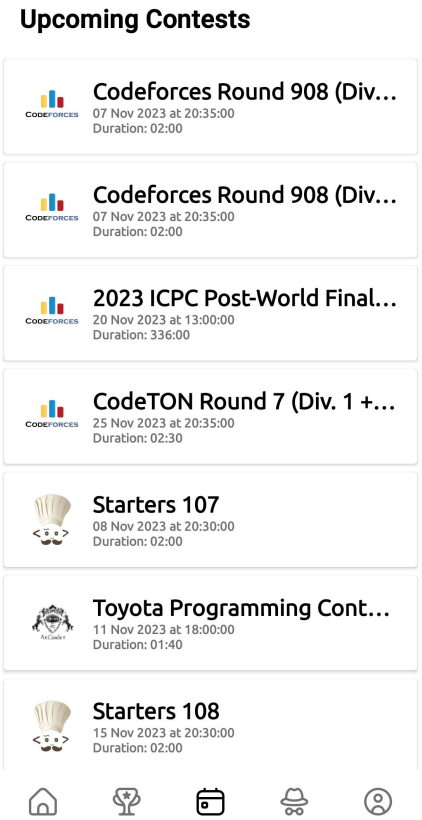

is
Istahak Islam
Istahak Iss

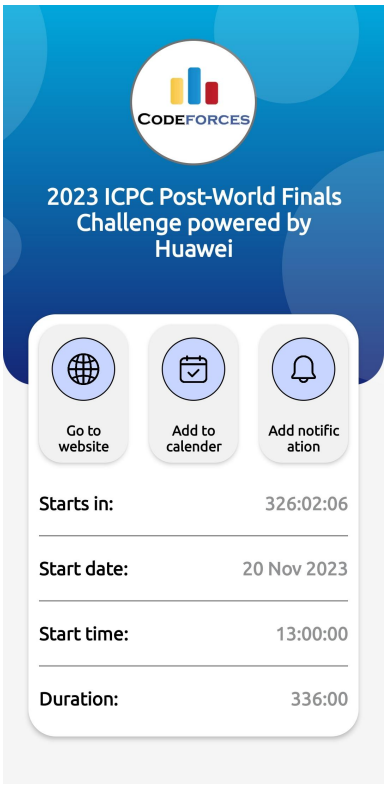

Create post
Post
Mahmudul Yeamim
What's going on your cp mind?

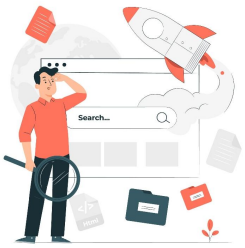

Chats
Search...
Mohammed Mahmudul
Istahak Islam
haha haha
Istahak Iss
Mahmudul Hasan
Mahmud Araf


Mohammed Mahmudul
hey. how are you
I hope you are doing well
can you please help me
how can i help you
Send a message...


Competition News
KodeShell Team
5 days ago
Exciting news for all coding enthusiasts! Mark your calendars for the Preliminary Mock Contest and the Preliminary Contest of ICPC Asia Dhaka Regional 2023, happening on October 11th and October 13th, 2023, respectively. Gear up for an exhilarating showcase of programming prowess. Get ready to witness the future of technology in action – it's an event you won't want to miss! #ICPCAsiaDhakaRegional #CodingContest #ICPC2023
1
KodeShell Team
6 days ago
"kodeshell community
4


Upcoming Contests
Codeforces Round 908 (Div...
07 Nov 2023 at 20:35:00
Duration: 02:00
Codeforces Round 908 (Div...
07 Nov 2023 at 20:35:00
Duration: 02:00
2023 ICPC Post-World Final...
20 Nov 2023 at 13:00:00
Duration: 336:00
CodeTON Round 7 (Div. 1 +...
25 Nov 2023 at 20:35:00
Duration: 02:30
Starters 107
08 Nov 2023 at 20:30:00
Duration: 02:00
Toyota Programming Cont...
11 Nov 2023 at 18:00:00
Duration: 01:40
Starters 108
15 Nov 2023 at 20:30:00
Duration: 02:00





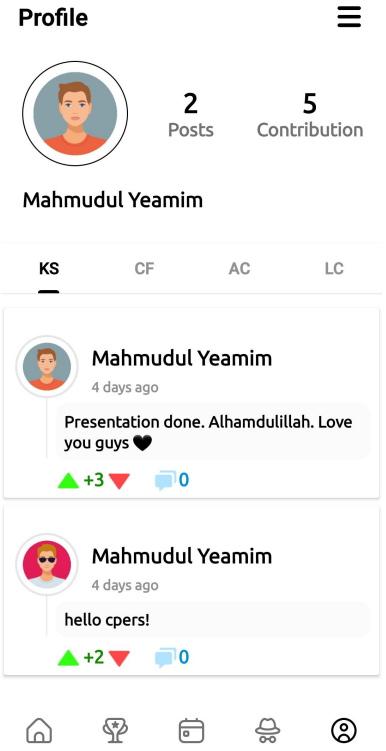
Stalk

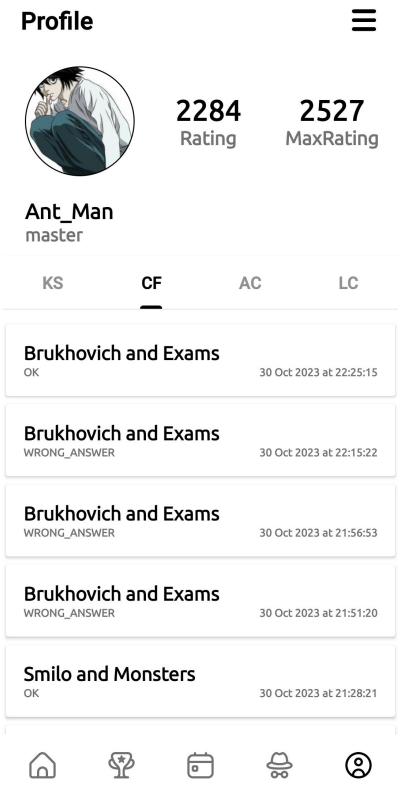
CodeForces Username

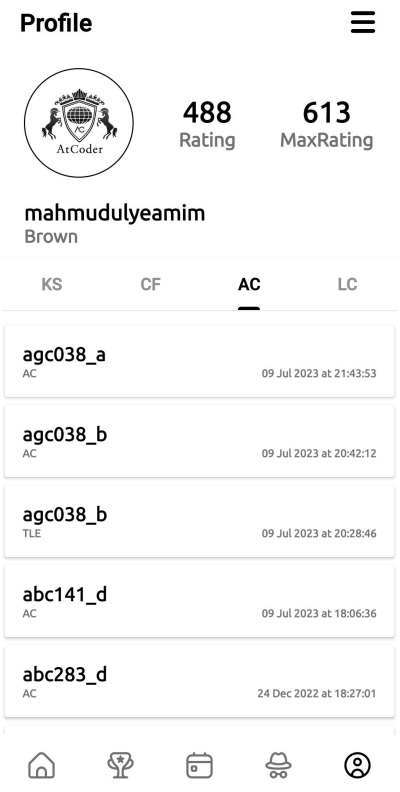
AtCoder Username

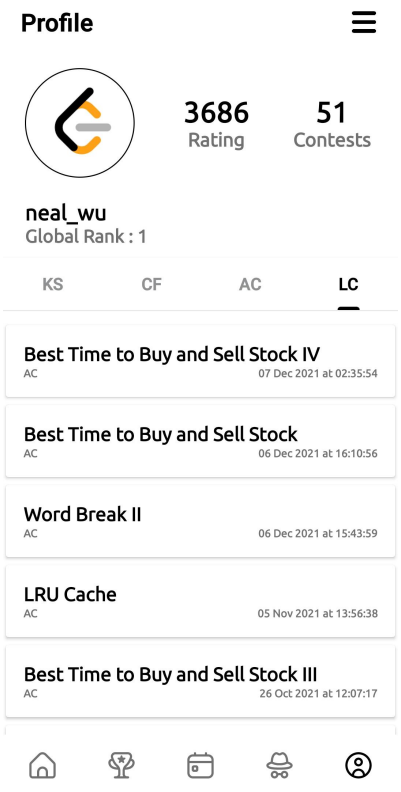
LeetCode Username

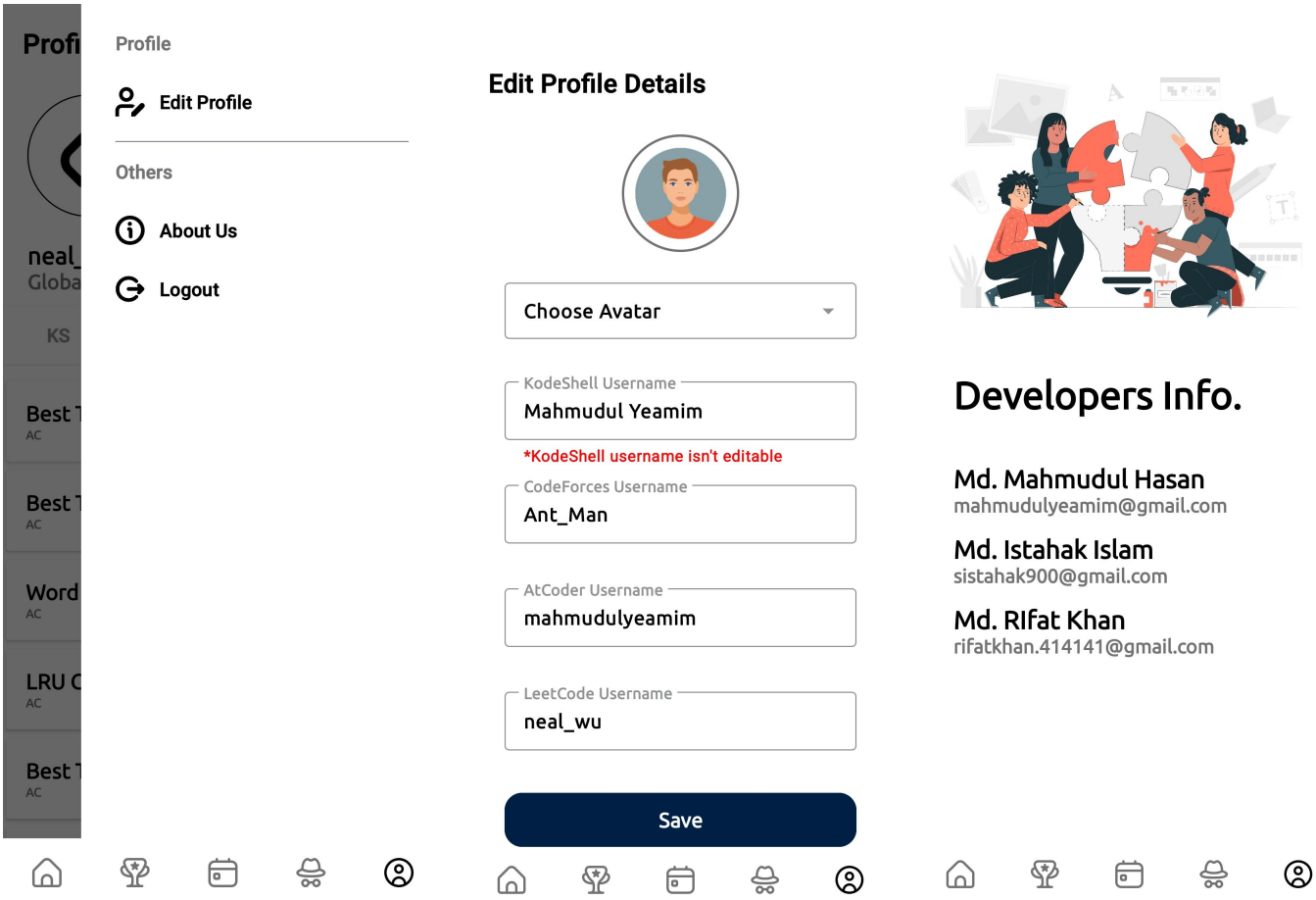
Search











2.2 Database Schema

We used Firebase Realtime Database for our android project. The Schema that we used for this project is described below:

1. User Data:

- Under the "user" node, user data is organized with a unique userID as the key.
- User information includes "atcoderuname," "avatarid," "codeforcesuname," "contribution," "firstName," "lastName,", “email”, "leetcodeuname" etc.

This part is used for user authentication and wherever we need to access data of a particular User.

2. Post Data:

- The "post" node stores information about posts with postID as the key.

- Each post includes details such as "avatarid", "comments", "content", "downVote", "id", "time", "upVote", "userID" etc.
- The comments node under the post node stores information about comments with commentID as the key.
- Each comments includes details such as "avatarid", "comment," "userName," etc.

This part handles data related to the post section.

3. Chat Data:

- Chats are organized under the "chats" node, with a unique chat ID as the key.
- The chatID is created by using the sender userID and the receiver userID so it ensures that the chatID is always unique for a particular sender and a receiver.
- Each chat contains a "messages" node with message IDs as keys, along with "message," "senderUID," "timeStamp." etc.

This part handles data related to chats. Handles which user sent a message to which user.

4. Admin Post Data:

- Admin posts are stored under the "adminpost" node, with post ID as the key.
- Admin posts include "comments" with comment IDs as keys, and each comment has "avatarid," "comment," "time," "username." etc.
- Additional admin posts can be added under the "adminpost" node as needed.

This part handles data related to the news section.

2.3 Code Repository

GitHub Link

2.4 Video Demo

YouTube

Conclusion

3.1 Challenges & Solutions

3.1.1 Md. Mahmudul Hasan

I encountered difficulties while trying to customize Android layouts and tab bars. I wanted to change certain attributes. But I didn't find any good resources to do so. Then, I decided to dive into the documentation to find solutions. After some research, I figured it out and made the necessary changes.

Our app's flow didn't initially match as we had thought, but we were able to resolve this issue by delving into the concept of app lifecycle. This understanding allowed us to fine-tune our app's behavior and make it work as intended.

We encountered a problem where fragments were being created before they appeared on the screen, resulting in unprepared data for display. To resolve this, we adjusted the data passing process by providing it during fragment creation rather than when the fragment was visible on the screen, ensuring that data was ready to be shown as intended.

Achieving consistent color theming, especially in dark or night mode, was initially challenging. Some issues arose, but by delving into the principles of color, I managed to resolve them.

I aimed to modify default behaviors of Android components but I was failing to do so. Then through learning, I discovered how to create custom designs and seamlessly integrate them.

3.1.2 Md Istahak Islam

During the development of this Android project, I encountered several challenges and implemented corresponding solutions:

Error Handling
Challenge: Errors can occur during network requests, such as server issues, network connectivity problems, or unexpected data formats.
Solution: I implemented a robust error-handling mechanism to provide informative error messages to users and log detailed error information for debugging. This enhanced user experience and simplified troubleshooting.

Data Synchronization:
Challenge: Ensuring the synchronization of data between the app and external platforms in real-time, especially during contests or live coding sessions, can be a significant challenge.

Solution: I implemented background services and periodic data updates to keep the app's data current. Additionally, I used WebSocket connections when available to receive real-time updates and contest notifications.
Addressing these challenges and implementing these solutions was crucial in creating a robust and user-friendly Android application that effectively connected with Codeforces, AtCoder, and LeetCode APIs.

3.1.3 Md. Rifat

Crafting an effective and scalable data structure was very difficult, requiring thoughtful planning to ensure it meets the project's needs. Managing real-time synchronization was very tricky, especially when dealing with simultaneous updates and conflicts. Testing the app on various Android devices was very tedious. Most of the time I faced errors regarding the database. However, each challenge offered valuable learning experiences.

Using the tools from android studio specially Logcat was very useful to point at the problem at hand. Using the internet to find the solution. Sometimes taking time to understand a concept before using it helped as well. Besides all these, communicating with team mates for their help regarding the problems helped me gain a lot of insights.

3.2 Lessons Learned

3.2.1 Md. Mahmudul Hasan

Holistic Understanding:
Holding dual roles in both UI/UX and backend development has provided a holistic perspective on app development, from design to implementation. This experience allows for informed decisions that consider the entire development process.

Effective Communication: Merging the frontend with the backend requires effective communication between these two domains. This experience has likely improved communication skills, ensuring a smooth collaboration between the design and development teams.

User-Centered Focus: The emphasis on UI/UX design and the integration of features like messaging and profile tracking demonstrate a commitment to creating a user-centered app.

Technical Proficiency: Being responsible for frontend and backend development has enhanced technical skills, making me a well-rounded developer capable of addressing various aspects of app development.

Quality Assurance Skills: Responsibility for quality assurance has provided an understanding of the importance of thorough testing and quality control in delivering a reliable app.

Problem-Solving: Multifaceted responsibilities have presented various challenges and problem-solving opportunities, fostering the ability to address complex issues and make informed decisions.
Team work: Working together on the app has shown us that when we collaborate and learn from each other, we can achieve great things. We've shared what we know, given feedback, and solved problems as a team. This has helped us improve our skills, communicate better, and come up with creative ideas together. We've supported each other, taken responsibility for our work, and found ways to solve any disagreements. In the end, working together has made us better and helped us create a successful app.

3.2.2 Md Istahak Islam

API Integration: I gained a strong understanding of how to interact with external APIs, enabling me to fetch and display data from different sources in our application.

Authentication: I acquired knowledge on implementing user authentication and securely handling API keys and tokens to ensure data privacy and security.

Error Handling: I developed skills in error handling and providing meaningful feedback to users in case of network or API-related issues.

Data Parsing: I improved my data parsing and manipulation abilities, allowing me to extract and present relevant information from the API responses.

User Experience: I gained insights into enhancing the user experience by efficiently presenting the fetched data and making it easily accessible and understandable.

Project Collaboration: I learned how to collaborate effectively with other team members, ensuring that my contributions seamlessly integrated with the rest of the project.

3.2.3 Md. Rifat

Real-time Synchronization: Navigating the complexities of real-time synchronization in the chat section enhanced our ability to manage concurrent updates seamlessly. Implementing Firebase listeners and conflict resolution strategies became second nature, contributing to a more responsive and engaging user experience.

Data Structure: The iterative nature of designing and refining a scalable data structure became apparent during the implementation of the chat and post sections. Adapting the structure as the project progressed underscored the importance of flexibility and planning for scalability.

Security and Access Control: Crafting and refining Firebase security rules for the authentication section imparted a deeper understanding of access control and safeguarding sensitive user information. This hands-on experience contributed to a more secure authentication process.

Error Handling: Dealing with errors in the authentication and chat sections honed our skills in implementing robust error-handling mechanisms. Providing users with clear and actionable feedback during unexpected situations became a priority for enhancing user satisfaction.

Collaborative Development Mindset: The collaborative nature of app development emerged as a key lesson. Regular team communication, effective documentation practices, and a commitment to continuous improvement became essential for overcoming challenges and streamlining the development process.

3.3 Future Plans

In the future, we would like to make our social media features more strong. There's plenty of features to be added in future like followers, making a strong algorithm for the post section. Besides them, we have also planned to add a dedicated help section where people can ask for doubts and others can reach out to them easily in order to help them. Features like end to end encryption in the chats and also provide a stronger privacy for every user. There's scope of adding AI tools where people can solve their doubts and fix their code bugs using these tools but this feature will take ample amount of time and effort from our end to be added but we will try to add this also even if it takes years. We would also like to shift our

database from free to commercial version to make the backend stronger than ever. We believe over time, we can provide a smooth and pleasing user experience for the users.