AN INTELLIGENT MOBILE APPLICATION TO ASSIST IN MATHEMATICS LEARNING AND DISCUSSIONS USING ADVANCED SORTINGS FOR MULTIPLE PLATFORMS

Xi He[1], Yu Sun[2]

[1]Northfield Mount Hermon, One Lamplighter Way, Gill, MA 01354
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

There is a growing need for a modern platform for learning and discussing mathematics. With the development of mobile technology, mathematicians worldwide can communicate easily with each other. This paper develops an application using simple AI and data structure to build a social network for everyone to learn and appreciate math with like-minded people. We use Firebase as a backend service provider for Authentication and cloud storage with blocking posts and comments feature. The app is great for math lovers worldwide, with our experiments showing that the mean time for a complete browsing section is approximately 14.545 milliseconds and a median of 13.8 milliseconds across regions. In addition, our built-in blocking mechanism will ensure that people can feel the inclusivity and openness that allows math ideas to freely flow without fear or prejudice. For this application, we took reliability, usability, and openness into consideration. The application aims to be an open platform for every math lover and encourage people to look at math more interactively.

KEYWORDS

Flutter, Data sorting, Mobile development, Mathematics

1. INTRODUCTION

The importance of understanding mathematics is undeniable given how broad the application of math in different fields[1]. However, the level of public participation in math studies is not reaching the optimum level yet, and wider discrepancies can be found across countries[2]. Namely, David Post highlights the discrepancies between nations in terms of hours its citizens spend studying Math and how it affects the advancement of math proficiency among developed and developing countries[3]. Thus, math competitions are getting prominent as nations try to promote their competitiveness in math fields and demonstrate nations powers[4]. Curiously enough, some researchers conclude that the development of online math platforms will increase participation and further promote the level of understanding of mathematics [5]. Tausczik says that “Participation in an online mathematics community helps ideas to gather for groups to discuss and learn upon”, while Mailizar notes that “WhatsApp has become an essential part of everyday life and can be a transformative tool for teacher professional[math] development[6]. Moreover, Emma Johnson deduced that dedicated mobile applications help students from 12-15 to study math better[7]. With all that being said, it is surprising that there aren’t many mobile

David C. Wyld et al. (Eds): SE, SAIM, SIPM, CoNeCo, ICITE, ACSIT, CMIT, FCST, SNLP – 2024
pp. 81-91, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140808

applications dedicated to mathematical discussions, even though around 80% of many nations use a smartphone [9]. Additionally, math-related applications are mostly learning apps and the target audiences are usually schools and teachers[10]. Even applying technology to school curriculums, there are major portions of students who will not be excited about learning new knowledge, including those in mathematics[11]. Thus, M Zhang argues that a catchy mobile application will make students like to study mathematics more[12].

Stieglitz argues that clickbait captions will destroy meaningful content within apps [13]. Similarly, Lischka highlights problems with populating algorithmic content on the app [14]. For instance, The New York Times focuses on improving the quality and frequency of comments from new users for selection, ensuring fairness and inclusivity. In contrast, Odyma adopts a method akin to a free market, relying on random community preference, thus granting equal opportunities for all voices to be heard and valued. Facebook utilizes a sophisticated algorithm to gauge engagement through metrics like likes and comments, yet this approach raises concerns about privacy invasion and addiction. Conversely, Odyma eschews algorithms to maintain a pure, focused community centered on mathematics, devoid of irrelevant distractions [8]. R Lu and Q Yang's study of Twitter's trend analysis reveals reliance on the MACD indicator, which prioritizes clickbait and short videos, potentially undermining meaningful discourse. Odyma opts for user-driven interaction without succumbing to trends or shallow content, prioritizing substantive discussions on mathematics over fleeting distractions.

My solution is to develop an integrated application to provide an interactive and supportive social media platform centered around math olympiads equipped with a tool to help problem-solving for math lovers. The first part creates a platform for users to post pictures, words, and words of their interests with hashtags labeled geometry, algebra, combinatorics, number theory, and others. People can comment for discussion or like and collect the posts for their own or others' references. Furthermore, the second part of the application is to provide a handy tool for users when solving problems. We will provide a collection of famous problems including AMC, AIME, USAMO, and IMO with solutions saved for the users to pick and practice. For AMC and AIME whose problems have a definite single-value answer, the application will be able to check whether the user's answer is correct. Moreover, it can be set to exam mode with a timer for mock practice. Besides, particularly for proof-based problems like those from USAMO and IMO, the application (iPad) provides a writing space for users to draft their answers. After they save and submit the answer, the page will be divided into two sections with their answers on the left and the official solutions on the right. While they can make comparisons and check answers by themselves, the application also has artificial intelligence(AI) to interpret their solutions and official solutions and provide advice for users' reference. Under all circumstances, a timer will be provided for the users to control the amount of time spent on the problems. Last but not least, the application allows the users to input their own problems and allows them to make them into their personal collection of problems. In case they need hints for the problems, AI will provide some potential hints as guidance for them.

In analyzing browsing times across continents, we aimed to understand the factors influencing internet speed. We collected data on browsing times and device counts for each region. Surprisingly, Asia, despite having the highest browsing time at 16.4 milliseconds, also had the most devices, indicating potential complexities in internet infrastructure or cultural habits. The correlation between device count and browsing time suggests a relationship worthy of further investigation, considering factors like network infrastructure and website content. VPN usage may also impact browsing speeds due to changing IP addresses. Conversely, in analyzing blocked posts percentages, our goal was to assess moderation efficacy. Despite Asia's lower percentage at 87%, Europe had the highest at 92%. Cultural norms, legal frameworks, and language differences likely contribute to these variations. The sheer number of devices may also affect moderation workload and algorithm performance. Both analyses underscore the multifaceted nature of internet behaviors, influenced by technological, societal, and geographical factors, requiring tailored approaches for effective content moderation and ensuring positive user experiences.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Categorization Chaos

First is categorization chaos: Our current system might struggle with sorting projects into numerous, diverse categories. We need to consider options like faceted search, hierarchical structures, or AI-powered classification to ensure projects fall under the right umbrellas, making them easily discoverable. Relevance Roadblock: Simply displaying posts isn't enough. We must bridge the gap between user interests and content. Leveraging user profiles, browsing history, and even implicit signals like clicks and likes can power recommendation algorithms that curate a personalized feed, showcasing only the most relevant posts, and boosting engagement and user satisfaction. By tackling these twin challenges, we unlock the true potential of our app, empowering users to navigate seamlessly customizing users' experiences, and generating the feeling of uniqueness.

2.2. Scaling a Project

Scaling a project management or collaboration app for a growing number of users presents unique challenges. Information overload can hinder users from finding what they need and make users lose interest in the app, so robust search, customizable dashboards, and smart notifications are key. To streamline collaboration, it's vital to ensure clear task ownership, implement version control, and provide real-time communication channels. Performance optimization through database design, Firebase cloud platform optimization, load balancing, and caching will prevent slowdowns as usage increases. Lastly, strong security measures like role-based access control and data encryption are paramount to protect user privacy and sensitive project information as the application scales.

2.3. A Robust Content Moderation System

To safeguard users from invasive and unwanted content, a robust content moderation system is crucial. This system must address potential pitfalls like the subjectivity of problematic content, the constant evolution of harmful materials, and the need to balance protection with free expression. I could implement a multi-pronged approach, including keyword filtering, user reporting mechanisms, image recognition to detect inappropriate visuals, and potentially AI-powered analysis to flag harmful language patterns. To ensure effectiveness, this system requires continuous updates to address new threats. Careful consideration of potential biases is also essential to achieve equitable and effective content moderation.

3. SOLUTION

The Flutter app, built for cross-platform compatibility on Android and iOS, presents a well-organized framework for social interaction. The app uses the language of Flutter for cross-platform development and provides a better UI and UX for the users. The app's logic centers around three core components: search, main page listing, and profile/posting. The search tool employs string-matching algorithms to filter posts based on user input. The main page listing dynamically fetches and renders posts, potentially using pagination for efficient data loading and a smooth scrolling experience.It implements ranking or sorting logic to prioritize content based on relevance or recency. Users can see every other person’s posts and search for posts most relevant to them with ease. They also can choose to like the posts, block/report users, or post content that is offensive to them. The profile section stores user-specific data and handles the post-creation process. This involves input validation, potential image handling, and communication with a database or server to persist the new post. The program flow begins with fetching initial data for the main page listing. Search interactions would trigger queries to filter and re-render results. Navigation to a profile page would load user-specific information and previously created posts. The post-creation process would guide the user through input, validation, and submission. The app uses Firebase as a cloud platform to oversee and handle online data storage and the app’s authentication system. Overall, the app leverages Flutter's reactive nature to update the UI seamlessly in response to user actions and data changes.

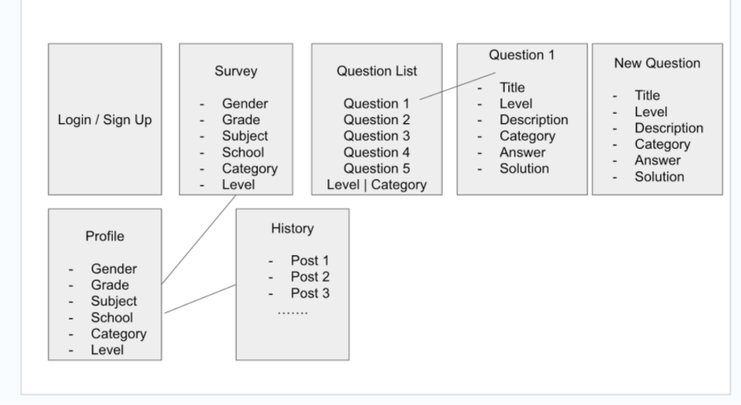


Figure 1. Overview of the solution

One of the central components of the application is the main page listing. In the app, a user can see all other people’s posts and make remarks within the posts. They also can reply to comments from other people as well. This system relies on interfacing with a Firebase Server that hosts all these comments and post info. To do so, it utilizes the Cloud Firestore platform and Flutter Streambuilder code to send and receive back posts and comments from different users.

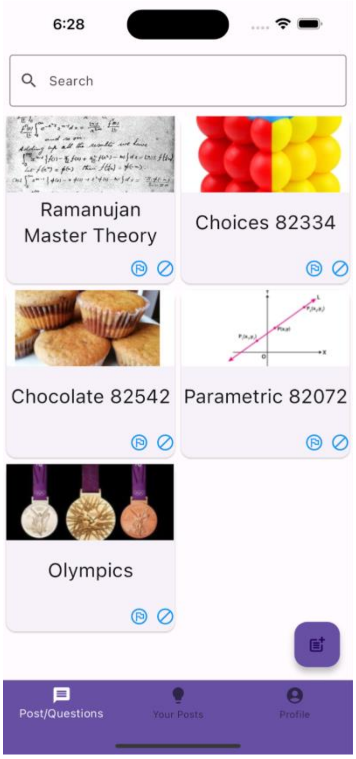
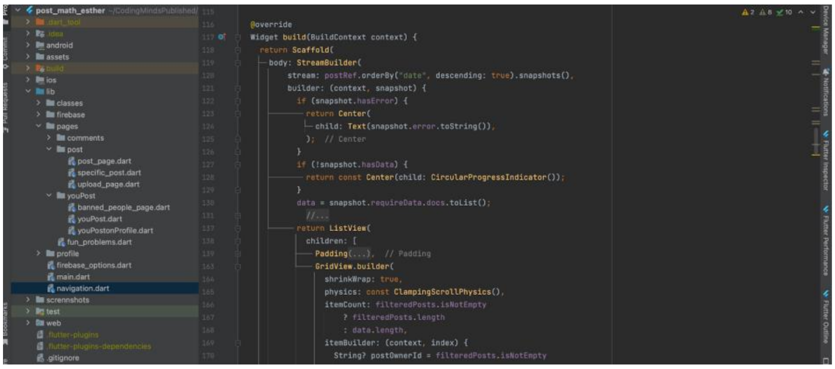


Figure 2. Screenshot of the images



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: StreamBuilder(
        stream: postRef.orderBy("date", descending: true).snapshots(),
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return Center(
              child: Text(snapshot.error.toString()),
            ); // Center
          }
          if (!snapshot.hasData) {
            return const Center(child: CircularProgressIndicator());
          }
          data = snapshot.requireData.docs.toList();
            //...
          return ListView(
            children: [
              Padding(...), // Padding
              GridView.builder(
                shrinkWrap: true,
                physics: const ClampingScrollPhysics(),
                itemCount: filteredPosts.isNotEmpty
                    ? filteredPosts.length
                    : data.length,
                itemBuilder: (context, index) {
                  String? postOwnerId = filteredPosts.isNotEmpty
```

Figure 3. Screenshot of code 1

To take information from the Cloud, this app uses Google Firebase as an information hosting platform. Firstly, our app uses StreamBuilder to create a data flow to ensure a smooth intake of real-time data. The data is organized in Cloud Database as Snapshots which uses Streambuilder code in Flutter to determine the state of the data. If the data has been successfully loaded into the snapshot, the code will modify the data and make it show on the screen. If the snapshot is loading or empty, the code will return a waiting sign on the screen. Lastly, if the data in the snapshot has an error, such as not having a uniform structure, the code will return a display error on the screen. The server of the database responds to the request from our app and sends the data by streams as JSON format data, which can be translated to information that can be displayed on the screen. Our app uses GridviewBuilder to display the data.

Another component is the posting page. The app connects with Firebase Firestore and sets up a mechanism to send information about the post, including the date, the image information, and the content of the post to the central storage system hosted on Firebase Cloud. Everyone will be able to see the contents and comment on them.

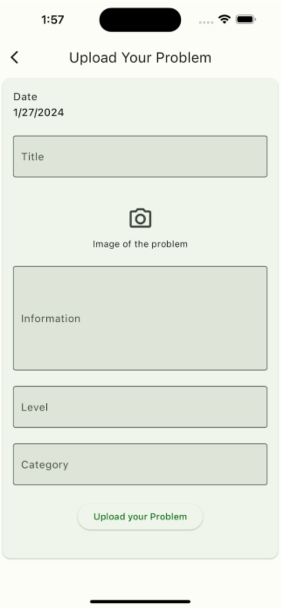
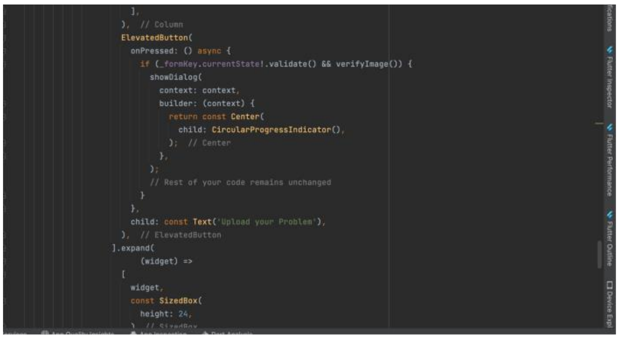


Figure 4. Screenshot of upload page

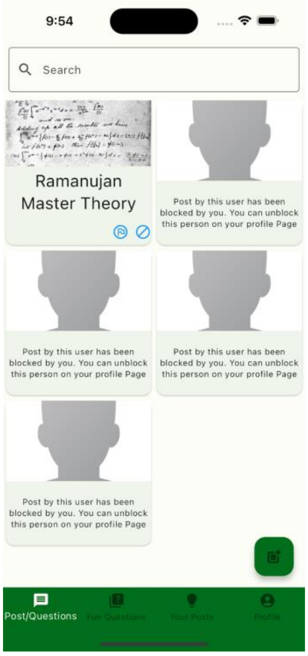


```
        ],
      ), // Column
      ElevatedButton(
        onPressed: () async {
          if (_formKey.currentState!.validate() && verifyImage()) {
            showDialog(
              context: context,
              builder: (context) {
                return const Center(
                  child: CircularProgressIndicator(),
                ); // Center
              },
            );
            // Rest of your code remains unchanged
          }
        },
        child: const Text('Upload your Problem'),
      ), // ElevatedButton
    ].expand(
        (widget) =>
      [
        widget,
        const SizedBox(
          height: 24,
```

Figure 5. Screenshot of code 2

The app asks for the users to upload posts, which are then connected to the Firebase. These posts contain different kinds of information, mainly pictures, and texts, which allow the users to post different types of data in different categories in an easy way. Once all the proper data fields are filled with user inputs, the app will send all this information to Cloud Firebase for storage purposes. In case other users want to read the content of the post, they will be able to see the comments on the posts as described in the previous section. To obtain the image, there are currently 2 ways for the users to choose: either via Camara taking or from Gallery. After the user chooses a picture, the app will upload the picture to the Cloud first for safekeeping and then generate a designated unique URL for the picture location to ensure that the pictures from the posts can be accessed by anyone who uses the Flutter image display algorithm.

Finally is the blocking mechanism. The app will look through the user's blocked list and make sure that their posts and contents will not be shown. This is done by using Streambuilder and Futurebuilder to hide posts from unwanted spammers.



Ramanujan Master Theory

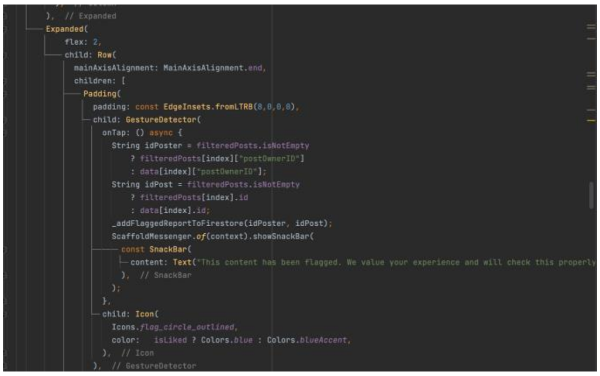
Post by this user has been blocked by you. You can unblock this person on your profile Page

Post by this user has been blocked by you. You can unblock this person on your profile Page

Post by this user has been blocked by you. You can unblock this person on your profile Page

Post by this user has been blocked by you. You can unblock this person on your profile Page

Figure 6. Screenshot of posts



```
  ),  // Expanded
  Expanded(
    flex: 2,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        Padding(
          padding: const EdgeInsets.fromLTRB(8,0,0,0),
          child: GestureDetector(
            onTap: () async {
              String idPoster = filteredPosts.isNotEmpty
                  ? filteredPosts[index]["postOwnerID"]
                  : data[index]["postOwnerID"];
              String idPost = filteredPosts.isNotEmpty
                  ? filteredPosts[index].id
                  : data[index].id;
              _addFlaggedReportToFirestore(idPoster, idPost);
              ScaffoldMessenger.of(context).showSnackBar(
                const SnackBar(
                  content: Text("This content has been flagged. We value your experience and will check this properly
                ),  // SnackBar
              );
            },
            child: Icon(
              Icons.flag_circle_outlined,
              color:   isLiked ? Colors.blue : Colors.blueAccent,
            ),  // Icon
          ),  // GestureDetector
```

Figure 7. Screenshot of code 3

To take information from Cloud, this app uses Google Firebase as an information hosting platform. Firstly, our app uses StreamBuilder to create a data flow to ensure a smooth intake of real-time data. The data is organized in Cloud Database as Snapshots which uses Streambuilder code in Flutter to determine the state of the data. If the data has been successfully loaded into the snapshot, the code will modify the data and make it show on the screen. If the snapshot is loading or empty, the code will return a waiting sign on the screen. Lastly, if the data in the snapshot has an error, such as not having a uniform structure, the code will return a display error on the screen. Furthermore, as the information filtered from the cloud database goes through, the data will go through a FutureBuilder to process which posts or contents belong to those who are on the blocked list. If they are then their contents will not be displayed and return a message saying to unblock this person first before going ahead. Our app uses GridviewBuilder to display the data.

4. EXPERIMENT

4.1. Experiment 1

One of the challenges for any application is the handling of users' activities in large numbers. Our math platform needs to be able to function smoothly across countries so that users can feel connected with like-minded people.

For this experiment, I set up different accounts in different countries by setting up VPNs for different accounts. With all the devices set up with different accounts, I make every device browse the app simultaneously, while I set up the phones to record the time it takes for browsing activities to work. The experiment includes browsing and posting one post at a time while finding comments and comments on them. I believe that 100 would be sufficient for the experiment since the experiment relies on the diversity of users' locations and 100 different users to different areas using VPNs will help test how users will interact simultaneously.

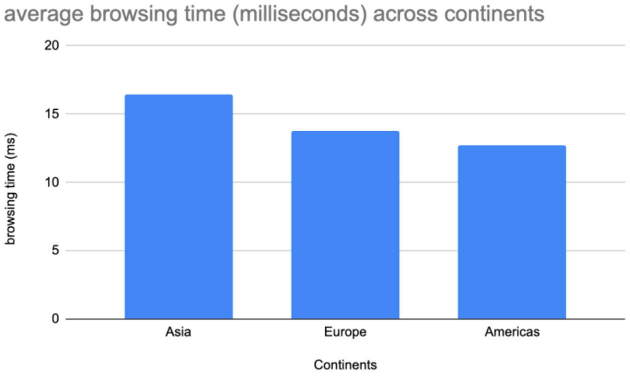


Figure 8. Figure of experiment 1

The analysis of the provided data reveals insightful patterns in average browsing times across continents. With a mean of approximately 14.545 milliseconds and a median of 13.8 milliseconds, browsing times vary across regions. Notably, the lowest browsing time of 12.7 milliseconds is recorded in the Americas, while Asia exhibits the highest at 16.4 milliseconds. Surprisingly, despite Asia's high browsing time, it also boasts the highest number of devices, indicating potential complexities in internet infrastructure or cultural internet usage habits. This disparity suggests that factors beyond mere device count influence browsing speeds, such as network infrastructure or website content. Nevertheless, the correlation between device count and browsing time hints at a possible relationship worthy of further investigation. This analysis underscores the multifaceted nature of internet browsing behaviors, shaped by a combination of technological, societal, and geographical factors. It also highlights that because of the VPN usage, the browsing time may be affected by the changing IP address, affecting the speed of the browsing mechanism.

4.2. Experiment 2

Another challenge is the blocking mechanism. For a public forum that values inclusivity and diversity, I believe that there must be no place for racist and bigotry comments presented on the screen and users should be the ones who decide who to block.

In an experiment aimed at enhancing user experience, participants were tasked with blocking unwanted comments from an application. The study sought to assess the efficacy of various comment moderation features and their impact on user satisfaction by using 100 different accounts across the world. Users were provided with options to block comments based on keywords, user profiles, or pre-defined categories. Results indicated varying success rates across methods, with keyword-based blocking showing the highest efficacy in filtering out undesirable content. Users who are proficient in different languages are encouraged to generate intentional profanity language to ask other users to block them. The users use VPNs to make this simulation worldwide, improving the test level of internationalization.

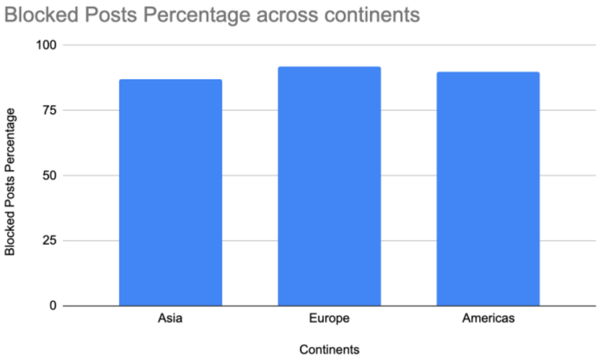


Figure 9. Figure of experiment 2

Upon analyzing the data on blocked posts percentages across continents, notable insights emerge. The mean blocked posts percentage stands at approximately 89.67%, with a median of 90%. Asia reports the lowest figure at 87%, while Europe records the highest at 92%. This discrepancy prompts consideration of various factors influencing moderation practices and content filtering efficacy. Despite Asia's larger device count, its comparatively lower blocked posts percentage raises questions about the region's moderation strategies or the nature of posted content. Cultural norms, legal frameworks, and differing user behaviors may contribute to these variations, as the language differences and the usage of normal words in special contexts may hinder the ability to autoblock such posts. Moreover, the sheer number of devices in each continent could affect moderation workload and algorithm performance, impacting the observed results. This analysis underscores the complexity of online content moderation and highlights the need for tailored approaches to ensure a positive user experience while addressing regional nuances and user expectations.

5. RELATED WORK

The New York Times comment section is picked based on an increase in the quality of first-time receivers' next approved comment, as well as the commenting frequency during commenters' early tenure on the site to ensure the quality of their picks of comments[15]. However, Odyma relies on random community preference, which is similar to how a "free market" runs without any intervention. This method of comment processing is better because it not only gives everyone an equal opportunity to voice their ideas but also gives everyone's opinions an equal chance to be heard, such that all of our users can get inspired, motivated, and rewarded.

The Facebook platform relies on a super algorithm that has four metrics to measure the volume of engagement with each post type, "likes," "comments," "poll votes," and "photo uploads." [16] However, we believe that this is not only an invasion of users' privacy, but also prone to the abuse of algorithms to make the users addicted to the app. Thus, in Odyma, we do not rely on any algorithm to control the information available to the users which prevents anyone from posting eye-catching things that are not relevant to math in the community in order to create a pure simple community for math lovers.

R Lu and Q Yang analyze Twitter trends and deduce that Twitter(now X) uses the MACD (Moving Average Convergence-Divergence) indicator to catch trends and use this to recommend users topics that eventually will generate trends [17]. We believe that our method is better since this algorithm relies a lot on users' interactivity, which in turn heavily relies on click-bait topics and short-duration videos. These will inevitably delude a meaningful discussion area, which we do not want to have on Odyma. As for our app, we believe in in-depth discussions in several math areas, thus we do not want to employ click-bait or snapping content that will take users away from discussing real issues with mathematics.

6. CONCLUSIONS

The first limitation is that we use a Google database which means the app does not work in China. As such, an independent Chinese app needs to be developed and published. Secondly, the aesthetics of the app needs further improvements to increase its interactive nature. For instance, 40% of the users suggest decreasing the space of the margins and increasing the room for content. Moreover, in the comment section, posting pictures and comment-liking functions are not available right now. However, 90% of the users would like to have these functions. In the future, these functions could be enabled to increase users' experience. A rewarding system could be created to increase the adhesiveness of users. Since 30% of the users complained about disinterested posts recommended to them, an algorithm that accurately detects the users' preference of topics such that they will be able to browse their interested topics in a short amount of time.

In conclusion, further efforts are needed to improve the app both in terms of its functions and aesthetics based on the users' feedback. Most of the functions in need are minor tools that can improve users' experience and aesthetic improvements are rather simple which can be done by increasing the highlights of the contents.

REFERENCES

[1] Reyna, Valerie F., and Charles J. Brainerd. "The importance of mathematics in health and human judgment: Numeracy, risk communication, and medical decision making." Learning and individual differences 17.2 (2007): 147-159.

[2] Sandefur, Justin. "Internationally comparable mathematics scores for fourteen African countries." Economics of Education Review 62 (2018): 267-286.

[3] Post, David, and Suet-Ling Pong. "Student labour and academic proficiency in international perspective." International Labour Review 148.1-2 (2009): 93-122.

[4] Campbell, James, and Kirsi Tirri. "Mathematics and science Olympiad studies: the outcomes of Olympiads and contributing factors to talent development of Olympians." International Journal for Talent Development and Creativity 5 (2017): 49-60.

[5] Tausczik, Yla R., and James W. Pennebaker. "Participation in an online mathematics community: differentiating motivations to add." Proceedings of the ACM 2012 conference on computer supported cooperative work. 2012.

[6] Mailizar, Mailizar, et al. "Mathematics teachers' interactions on online community of practices: A social network analysis." Education and Information technologies 27.5 (2022): 6533-6551.

[7] Johansson, Emma. "Applying Gamification to a Mobile Application to Motivate Children to Learn Math." (2021).

[8] Padayachee, Pragashni, and Anita L. Campbell. "Supporting a mathematics community of inquiry through online discussion forums: towards design principles." International Journal of Mathematical Education in Science and Technology 53.1 (2022): 35-63.

[9] Olson, Jay A., et al. "Smartphone addiction is increasing across the world: A meta-analysis of 24 countries." Computers in Human Behavior 129 (2022): 107138.

[10] Ariyanto, L., W. Kusumaningsih, and A. N. Aini. "Mobile phone application for mathematics learning." Journal of Physics: Conference Series. Vol. 983. No. 1. IOP Publishing, 2018.

[11] Mora, Richard. "" School is so boring": High-stakes testing and boredom at an urban middle school." Penn GSE Perspectives on Urban Education 9.1 (2011): n1.

[12] Zhang, Meilan, et al. "Using math apps for improving student learning: An exploratory study in an inclusive fourth grade classroom." TechTrends 59 (2015): 32-39.

[13] Jung, Anna-Katharina, et al. "Click me…! The influence of clickbait on user engagement in social media and the role of digital nudging." Plos one 17.6 (2022): e0266743.

[14] Lischka, Juliane A., and Marcel Garz. "Clickbait news and algorithmic curation: A game theory framework of the relation between journalism, users, and platforms." New Media & Society 25.8 (2023): 2073-2094.

[15] Wang, Yixue, and Nicholas Diakopoulos. "Highlighting High-quality Content as a Moderation Strategy: The Role of New York Times Picks in Comment Quality and Engagement." ACM Transactions on Social Computing (TSC) 4.4 (2022): 1-24.

[16] Edney, Sarah, et al. "Posts, pics, or polls? Which post type generates the greatest engagement in a Facebook physical activity intervention?." Translational behavioral medicine 8.6 (2018): 953-957.

[17] Lu, Rong, and Qing Yang. "Trend analysis of news topics on twitter." International Journal of Machine Learning and Computing 2.3 (2012): 327.