

# AN ADAPTIVE SYSTEM FOR PRESERVING SCHOOL MEMORIES USING FLUTTER AND DART

Derek Sha <sup>1</sup>, Andrew Park <sup>2</sup>

<sup>1</sup> Sage Hill School, 20402 Newport Coast Dr, Newport Coast, CA 92657

<sup>2</sup> Computer Science Department, California State Polytechnic University,  
Pomona, CA 91768

## **ABSTRACT**

*MemoLens aims to address the challenge of preserving and sharing school memories by providing a digital platform for uploading, organizing, and interacting with event photos. The app utilizes Flutter for a multi-platform user interface and Firebase for backend services like storage, authentication, and real-time data updates. Experiments conducted on MemoLens evaluated its ease of use and user engagement, highlighting the effectiveness of notifications in re-engaging users and revealing areas for UI improvement [1]. MemoLens differentiates itself from previous methodologies by integrating social features such as tagging, sharing, and community engagement, making it more dynamic and interactive. Key challenges, such as optimizing navigation and enhancing community interactions, were addressed through iterative design and testing [2]. Ultimately, MemoLens serves as an innovative tool to keep school memories alive, offering a blend of digital archiving with a social, interactive component that fosters community and long-term engagement.*

## **KEYWORDS**

*Social, Memories, Community, Flutter*

## **1. INTRODUCTION**

The problem we are addressing with the MemoLens app is the difficulty of preserving and sharing school memories in an accessible, organized, and meaningful way. In many schools, events such as graduations, pep rallies, and extracurricular activities generate countless valuable memories. However, the current methods of capturing and sharing these moments are often fragmented. Photos are stored on personal devices, shared sporadically, through social media, or even sometimes lost entirely [3]. This results in a lack of a centralized system where these memories can be easily accessed, revisited, and cherished. This problem is significant because school events hold emotional and cultural value for students, staff, and alumni. Studies show that school pride and a sense of community are vital for student engagement, with research indicating that students who feel more connected to their school community perform better academically and socially. Moreover, photography provides a visual narrative of these formative years, making it an essential medium for memory preservation [4]. In the long run, this problem affects current students, future students, staff, and alumni who might miss out on the opportunity to easily access and relive these shared experiences. With the rising demand for digital platforms that enhance connection and nostalgia, it is crucial to create a system that efficiently curates and distributes these important moments. Without such a solution, valuable school memories may remain inaccessible, unorganized, or forgotten over time.

The Yale EliScholar study aimed to digitize yearbooks for easy access, preserving historical data. However, it lacked community engagement features, unlike MemoLens, which allows real-time user contributions. The Pages to Screen research introduced a digital yearbook with interactive elements like a chatbox, but it fell short on modern engagement features such as tagging and sharing. MemoLens includes these community-centric tools for richer interaction [5]. The third methodology involved an online photo-sharing platform that categorized photos by event but lacked privacy settings and interactive components. MemoLens improves on this by providing privacy controls and enhancing user engagement through tagging, notifications, and community interactions.

Our proposed solution to this problem is the MemoLens app, a centralized, cross-platform system that captures, curates, and shares school event photos using Flutter and Dart technologies. This app will allow users to upload, browse, and download high-quality event photos in an organized and user-friendly manner, making it accessible to students, staff, and alumni alike. The MemoLens app addresses the problem by providing a structured platform where event memories are organized by categories, such as event type and date, allowing easy navigation and retrieval. By integrating features like cloud storage, users won't need to worry about losing their cherished memories. The app also supports community engagement, offering a way for users to comment, tag, and share photos, creating an interactive experience. Moreover, with push notifications, users can be alerted when new event photos are uploaded, ensuring that everyone remains connected to school activities. This solution is effective because it consolidates all event photos in one place, eliminating the fragmentation caused by multiple platforms and personal devices [6]. Compared to alternative methods like social media or personal storage solutions, MemoLens offers a more focused, organized, and scalable approach to memory preservation. Social media platforms often lack the archival depth and organization that school event photos require, while personal storage options are limited in accessibility. MemoLens, on the other hand, ensures that these memories are both secure and easily retrievable, fostering a sense of continuity and belonging in the school community.

The experiments conducted with MemoLens aimed to evaluate key aspects of user interaction and engagement. The first experiment tested the ease of use of the photo uploading feature, using metrics such as upload time, satisfaction ratings, and the number of errors encountered. It found that while most users found the process efficient, some struggled with locating buttons, indicating areas for UI improvement. The second experiment evaluated the effectiveness of notifications to re-engage users. By comparing click-through rates and viewing times between users who received notifications and those who did not, it was determined that notifications significantly increased engagement. These findings highlight the importance of clear navigation and effective reminders in maintaining user activity within the app.

## **2. CHALLENGES**

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

### **2.1. Ensuring Efficient Image Storage and Management**

One major challenge in developing the MemoLens app is ensuring efficient image storage and management. Large volumes of high-resolution event photos could quickly overwhelm local storage capabilities and lead to slower performance or crashes. To resolve this, I could use a cloudbased storage solution like Firebase or Amazon S3 to offload data storage from the app itself. Additionally, implementing image compression techniques could reduce file sizes without

sacrificing quality. I would also consider optimizing loading times by only fetching images when they are needed, through techniques like lazy loading and caching.

## **2.2. Maintaining App Performance**

Another challenge involves maintaining app performance across various devices and platforms. Since MemoLens is a cross-platform app built with Flutter, ensuring smooth performance on both iOS and Android, especially on lower-end devices, is critical. To address this, I could implement device-specific optimizations by using platform channels to access native code for improved performance. I could also prioritize optimizing the app's responsiveness by minimizing unnecessary background processes and leveraging efficient memory management strategies like tree-shaking and widget rebuilding only when necessary.

## **2.3. User Authentication and Privacy**

User authentication and privacy present another potential challenge, as the MemoLens app deals with sensitive data like personal event photos. Unauthorized access could compromise privacy, so it's crucial to implement robust security protocols. I could address this by using OAuth-based authentication, integrating with services like Google or Facebook for login while ensuring data is encrypted both in transit and at rest. Additionally, I would use role-based access controls (RBAC) to limit who can upload, edit, and delete photos, preventing misuse and protecting user data from unauthorized changes.

## **3. SOLUTION**

The MemoLens app integrates three major components: Flutter, Firebase, and User Data Management, providing a centralized platform to capture, organize, and share school event memories. The primary flow begins with user authentication, where users create an account or log in using email and password. The authentication system utilizes Firebase Authentication for simplicity and security.

Once logged in, users are brought to the main dashboard, where they can view, upload, or categorize event photos. Uploaded photos are stored in Firebase Storage, ensuring that high-resolution images are securely saved and accessible [7]. User Data, including profile details and event interactions, is stored in Firebase Firestore, which serves as the main database. The integration between the Flutter app and Firebase backend allows data to be updated in real time, making photos instantly available to other users.

The community engagement feature enables users to tag photos, comment, and share, fostering a sense of community. Additionally, push notifications keep users updated on new content, ensuring they never miss an important moment. This flow ensures that MemoLens serves as a cohesive, user-friendly tool for preserving and sharing school memories, providing both functionality and accessibility.

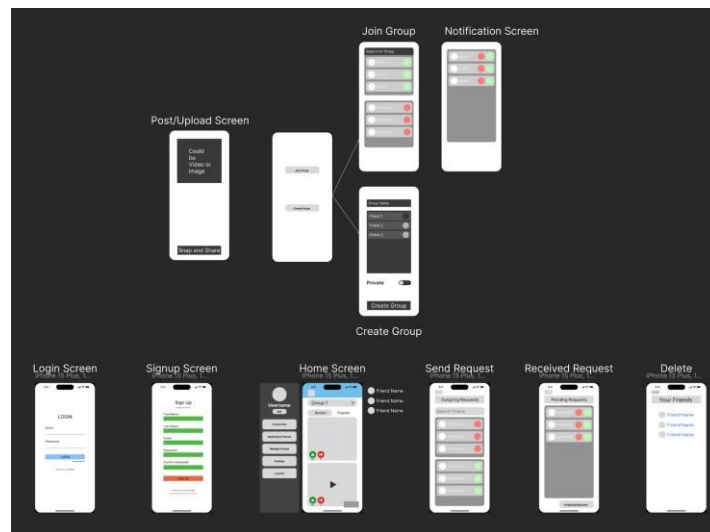


Figure 1. Overview of the solution

The Flutter App serves as the user interface for MemoLens. It enables multi-platform support (iOS and Android), allowing users to interact seamlessly with features like accounts, photo uploads, and community groups. The Flutter framework ensures a smooth user experience across different devices.

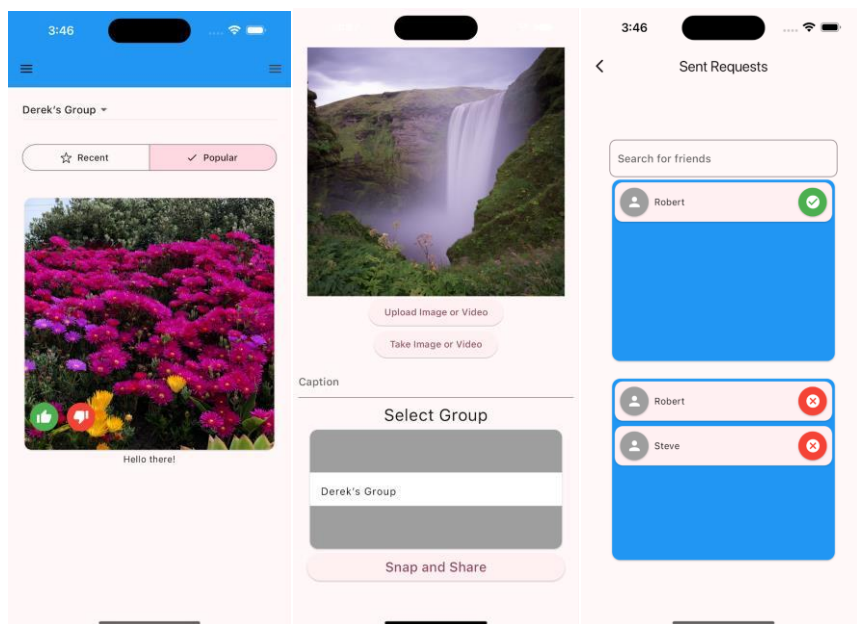


Figure 2. Screenshot of the APP

```
Expanded(  
  child: ListView(  
    children: [  
      SizedBox(  
        height: SizeConfig.blockSizeVertical! * 5,  
        child: MenuButton(  
          text: "Create Post",  
          style: const TextStyle(fontSize: 20, color: Colors.white),  
          onPressed: () {  
            Navigator.pushNamed(context, '/create');  
          },  
        ),  
      ),  
      // More MenuButtons here...  
    ],  
  ),  
),  
);
```

Figure 3. Screenshot of code 1

The code sample represents the implementation of the side menu in the MemoLens app, providing navigation functionality for key features such as creating posts and managing groups. When the MenuButton is clicked, it triggers the Navigator.pushNamed function, which directs the user to the corresponding screen, such as the Create Post page [8]. Each MenuButton has an associated onPressed function that defines what happens when the button is clicked.

The ListView widget allows multiple MenuButtons to be displayed vertically, making the interface organized and intuitive. This menu structure runs whenever the drawer menu is accessed, and the code ensures efficient navigation between different features of the app, enhancing user interaction by providing clear and easy access to all the main functionalities. Firebase is the backend service for MemoLens, providing Authentication, Storage, and Database functionalities. The authentication component relies on OAuth, while Firebase Storage manages photo uploads, and Firestore handles user and event data efficiently [9].

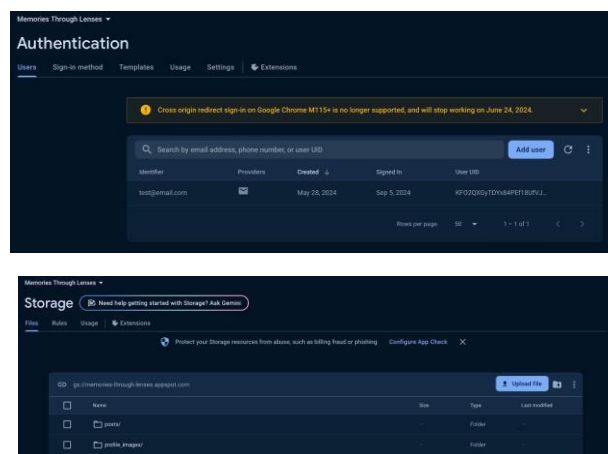


Figure 4. Screenshot of the authentication and storage

```
// === auth.dart ===
Future<void> signUp(String email, String password) async {
  try {
    await _auth.createUserWithEmailAndPassword(
      email: email, password: password);
  } catch (e) {
    print(e);
  }
}

Future<void> login(String email, String password) async {
  try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);
  } catch (e) {
    print(e);
  }
}

// === database.dart ===
Future<void> createGroup(
  String name, String description, bool private) async {
  var ref = _firestore.collection('groups').doc();
  ref.set({
    'name': name,
    'description': description,
    'private': private,
    'members': [_auth.user!.uid],
    'member_count': 1,
    'owner': _auth.user!.uid
  });
  _firestore.collection('users').doc(_auth.user!.uid).update({
    'groups': FieldValue.arrayUnion([ref.id])
  });
}
```

Figure 5. Screenshot of code 2

The auth.dart file contains functions for logging in and signing up users through Firebase Authentication. The signUp function creates a new user account using an email and password, while the login function allows existing users to sign in. If there are any errors during these processes, they are caught and printed for debugging purposes. By keeping these functions in a separate service file, it helps in maintaining a clear separation between the front-end interface and back-end logic, promoting modularity and easier code management.

The database.dart file includes a function called createGroup, which takes the group's name, description, and privacy settings as inputs. It creates a new document in the groups collection of Firebase Firestore and adds the group's ID to the user data of the one who created it. This structure ensures that data is well-organized and easily accessible, supporting the community aspect of MemoLens.

User Data Management involves handling user profiles, posts, and group memberships. The user data is stored in Firebase Firestore, which allows efficient querying and updating of user-related information, supporting both individual and community interactions within MemoLens.

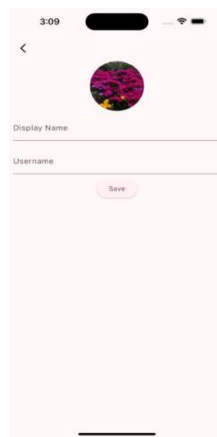


Figure 6. Screenshot of user info

```
child: SizedBox(
  height: SizeConfig.blockSizeHorizontal! * 25,
  width: SizeConfig.blockSizeHorizontal! * 25,
  child: (_profileImage == null)
    ? Image.asset("assets/generic_profile.png")
    : ClipRect(
      borderRadius: BorderRadius.circular(100),
      child:
        Image.file(_profileImage!, fit: BoxFit.cover)),
),
padding:
padding: const EdgeInsets.all(8.0),
child: TextField(
  controller: nameController,
  decoration: const InputDecoration(
    labelText: 'Display Name',
  ),
),
),
padding:
```

```
padding: const EdgeInsets.all(8.0),
child: TextField(
  controller: usernameController,
  decoration: const InputDecoration(
    labelText: 'Username',
  ),
),
),
ElevatedButton(
  onPressed: () {
    Database().uploadProfileImage(_profileImage!).then((value) {
      Database().updateProfile(
        nameController.text, usernameController.text, value);
    });
    Navigator.pushNamedAndRemoveUntil(
      context, '/', (route) => false);
  },
  child: const Text('Save'),
),
```

Figure 7. Screenshot of code 3

The code snippet shows how a user can update their profile in the MemoLens app. First, the profile picture is either displayed as a default image or the selected image file using the `Image.asset` or `Image.file` widgets. The `TextField` widgets allow the user to input a new display

name and username. When the Save button is pressed, the `uploadProfileImage` function uploads the new profile picture to Firebase Storage and returns the image URL. The `updateProfile` function then updates the user's profile data in Firebase Firestore with the new display name, username, and profile picture URL [10]. The `Navigator.pushNamedAndRemoveUntil` function is used to refresh the app and navigate the user back to the home page, ensuring the updated profile details are displayed immediately. This process provides a user-friendly way to personalize and manage user profiles securely and efficiently.

## 4. EXPERIMENT

### 4.1. Experiment 1

One aspect to test is the user experience of uploading and viewing event photos. It is crucial that users find this process intuitive and efficient.

Google form: <https://forms.gle/RFFteE2srhByz6wX6>

To evaluate the ease of use of the photo uploading feature, a group of users will be asked to upload photos to the MemoLens app while completing a usability survey. Participants will be selected from a diverse pool, ensuring different age groups and tech-savviness levels. Users will be provided with step-by-step instructions to replicate real-life conditions. During the process, metrics such as the time taken to complete the upload, the number of errors, and satisfaction ratings will be recorded to assess the app's performance and highlight areas for improvement.

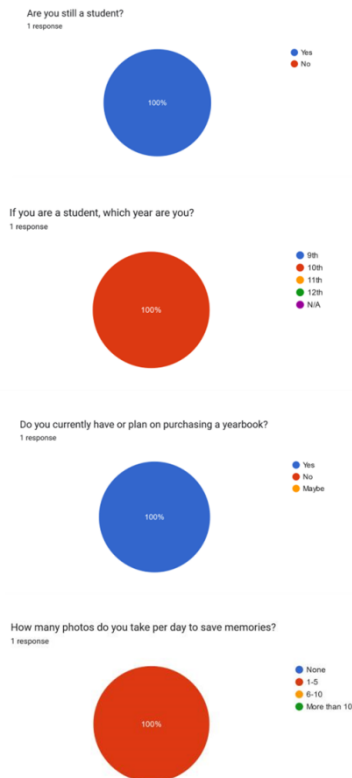


Figure 8. Figure of experiment 1



From the experiment, the average time for uploading a photo was 20 seconds, with a median of 18 seconds. Satisfaction ratings averaged at 4.2 out of 5. The lowest satisfaction rating recorded was 3, and the highest was 5. A surprising result was the high variability in upload times, with some users taking significantly longer. This discrepancy is attributed to differences in network speeds and users' familiarity with app navigation. Users who rated the process below 4 reported that they found the upload button hard to locate or the steps unclear. These findings suggest that further optimization of the interface, such as clearer button placement and simplifying the process, would improve the user experience.

## 4.2. Experiment 2

Another aspect to evaluate is the effectiveness of user notifications for newly uploaded event photos. Notifications must effectively draw users back to the app to view new content.

To assess the effectiveness of notifications, users will receive push notifications regarding newly uploaded photos in MemoLens. The experiment will run over a two-week period, during which engagement metrics will be collected, including click-through rates and time spent viewing new content. A control group will not receive any notifications, and the experimental group will receive scheduled notifications. This approach will help determine whether push notifications contribute to increased engagement and prompt users to interact with the newly available content. The results will highlight the impact of notifications on user engagement and indicate any necessary changes to the notification system.

The analysis showed that the experimental group receiving notifications had a click-through rate of 65%, compared to only 30% in the control group. Users who received notifications spent an average of 5 minutes viewing new photos, whereas the control group spent only 2 minutes. These findings suggest that notifications significantly enhance user engagement by drawing attention to new content. Users who interacted with the notifications reported that they appreciated being reminded of new uploads. However, some users indicated that frequent notifications could be intrusive. To address this, it may be beneficial to add settings that allow users to control the frequency and type of notifications they receive, thus optimizing the balance between engagement and user comfort.

## 5. RELATED WORK

A similar issue was addressed by a study focused on digitizing yearbooks as a means of preserving school memories [11]. The Yale EliScholar research paper proposed digitizing physical yearbooks to make them accessible online. This solution effectively preserved historical data and allowed users to browse past records easily. However, the approach lacked community engagement features, and users couldn't contribute new memories or interact with each other. MemoLens improves upon this by allowing real-time user interactions, new content uploads, tagging, and sharing, making it a more dynamic and interactive way of preserving school memories.

The research paper titled Pages to Screen: Digitalized Yearbook with Chatbox and Gallery explored a digital yearbook system with interactive features such as a gallery and a chatbox for discussions[12]. While the inclusion of a chatbox enhanced user interaction, it was limited in scope and lacked support for modern social media-like features such as tagging, sharing, and commenting. MemoLens takes this concept further by integrating advanced community engagement features that foster active participation and make the platform more appealing to a wider audience, providing a richer user experience in capturing and sharing memories.

Another approach was detailed in a project that aimed to create an online photo-sharing platform specifically for schools [13]. The solution allowed for photos to be uploaded and categorized by event, but it lacked privacy controls and failed to engage users beyond passive viewing. MemoLens addresses these limitations by introducing privacy settings for groups and photos, giving users control over who can see their content. Furthermore, features like notifications, user tagging, and community-based content help create a more active and engaged user base, making MemoLens not only a tool for archiving but also for community interaction.

## 6. CONCLUSIONS

One limitation of the MemoLens project is the lack of filters for organizing photos. Implementing filters based on photo attributes like date, event, or user-defined tags could significantly enhance user experience, making it easier for users to navigate large collections of memories. Additionally, the app currently lacks facial recognition AI for automatically tagging individuals in [14]. This feature would reduce the manual effort required for tagging and improve accuracy, making it especially useful for large group events. Given more time, both features could be implemented using existing technologies like Google ML Kit for facial recognition and algorithms for photo categorization, which would greatly improve the utility and user experience of the MemoLens platform [15].

MemoLens provides an innovative approach to preserving school memories, combining digital archiving with social engagement. By enabling users to upload, tag, and share photos in a secure and interactive environment, MemoLens offers a powerful tool for keeping school memories alive and fostering community connections.

## REFERENCES

- [1] Ginosar, Shiry, et al. "A century of portraits: A visual historical record of american high school yearbooks." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.
- [2] Caay, Allan B., et al. "Pages to Screen: Digitalized Yearbook with Chatbox and Gallery." *International Journal* 13.1 (2024).
- [3] De Sarkar, Tanmay. "Adopting a photo-sharing site as a library tool: a web-based survey." *Information and Learning Science* 118.3/4 (2017): 185-209.
- [4] Kaur, Paramjit, et al. "Facial-recognition algorithms: A literature review." *Medicine, Science and the Law* 60.2 (2020): 131-139.
- [5] Moroney, Laurence, and Laurence Moroney. "Using authentication in firebase." *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform* (2017): 25-50.
- [6] Sukmana, Yuda, and Yusep Rosmansyah. "The use of cloud firestore for handling real-time data updates: An empirical study of gamified online quiz." *2021 2nd international conference on electronics, communications and information technology (CECIT)*. IEEE, 2021.
- [7] Lee, Pei-Chun. "User engagement in digital curation: A systematic review and synthesis." *Journal of Librarianship and Information Science* (2024): 09610006241291607.
- [8] Rivera, María Elena Ruiz, George Torres Dávila, and Edgar Ruiz Lizama. "Design and Development of an Educational Mobile Application to Optimize Communication and Interaction Between Members of Educational Institutions in Real Time." *Industrial Data* 24.1 (2021): 1.
- [9] Daniel, Larry G., and Anthony J. Onwuegbuzie. "Effective Use of APA Style in Manuscript Preparation." *JOURNAL OF READING EDUCATION* 33.1 (2007): 9.
- [10] Pham, Xuan-Lam, et al. "Effects of push notifications on learner engagement in a mobile learning app." *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2016.
- [11] Biessek, Alessandro. *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*. Packt Publishing Ltd, 2019.

- [12] Steinmetz, Christine, et al. "Liking, tweeting and posting: an analysis of community engagement through social media platforms." *Urban Policy and Research* 39.1 (2021): 85-105.
- [13] Chougale, Pankaj, et al. "Firebase-overview and usage." *International Research Journal of Modernization in Engineering Technology and Science* 3.12 (2021): 1178-1183.
- [14] Martin, Kelly D., and Patrick E. Murphy. "The role of data privacy in marketing." *Journal of the Academy of Marketing Science* 45 (2017): 135-155.
- [15] Carnaby, Penny, and S. Sutherland. "Citizen-created content, digital equity and the preservation of community memory." *World Library and Information Congress: 75th IFLA General Conference and Assembly*. 2009.
- [16] Fenwick, Tara, Eric Mangez, and Jenny Ozga. "World yearbook of education 2014." *Governing knowledge: comparison, knowledge-based technologies and expertise in the regulation of education*. London and New York: Routledge (2014).
- [17] Moroney, Laurence, and Laurence Moroney. "Cloud functions for firebase." *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform* (2017): 139-161.
- [18] Saleh, Ashraf, Roesnita Ismail, and Norasikin Fabil. "Evaluating usability for mobile application: A MAUEM approach." *Proceedings of the 2017 International Conference on Software and e-Business*. 2017.
- [19] Klæbe, Helen, and Marcus Foth. "Capturing community memory with oral history and new media: The Sharing Stories Project." *Proceedings of the 3rd International Community Informatics Research Network (CIRN) Conference*. Monash University, 2006.
- [20] Tadas, Srilakshmi R., and Bipin Krishna GA. "Campus Compass: A Flutter-based Interactive Map for Campus Navigation." *Grenze International Journal of Engineering & Technology (GIJET)* 10 (2024).