COMMUNITY BASED REPORTING AND MONITORING TOOL FOR WOMEN SAFETY IN COLLEGES/UNIVERSITIES

Submitted by,

MARRIBOYINA MOUNIKA	**-**	**20211CSE0262**
CHALLA OMNATH	**-**	**20211CSE0305**
CHAITRA K	**-**	**20211CSE0274**

Under the guidance of,

Dr. AFROJ ALAM

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **COMMUNITY BASED REPORTING AND MONITORING TOOL FOR WOMEN SAFETY IN COLLEGES/UNIVERSITIES** being submitted by MARRIBOYINA MOUNIKA (20211CSE0262), CHALLA OMNATH (20211CSE0305), CHAITRA K (20211CSE0274) bearing roll number in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. AFROJ ALAM
Assistant Professor
School of IS
Presidency University

Dr. ASIF MOHAMED H B
Asso. Prof & HOD
School of CSE&IS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor – Engineering
Dean –School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby declare that the work, which is being presented in the report entitled **COMMUNITY BASED REPORTING AND MONITORING TOOL FOR WOMEN SAFETY IN COLLEGES/UNIVERSITIES** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of my own investigations carried under the guidance of **Dr. Afroj Alam, Assistant Professor School of Information Science, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME(S)	ROLL NO(S)	SIGNATURE(S)
MARRIBOYINA MOUNIKA	20211CSE0262	
CHALLA OMNATH	20211CSE0305	
CHAITRA K	20211CSE0274	

ABSTRACT

This project is about creating a mobile application to improve women's safety in colleges and universities. The app is designed to help students report unsafe situations and get quick help when they feel threatened. One of the main features of the app is the SOS button. When a user presses the SOS button, it immediately sends an alert to trusted contacts or college security. Along with the alert, the app also shares the user's current location in real-time.

The app includes audio and video recording features, which help capture evidence during emergencies. This information can be useful for authorities to act quickly. The app also has an AI-based chatbot that can answer questions related to safety and provide guidance on what to do in risky situations. The chatbot can also help users report incidents and connect them with help centres.

The app encourages community-based reporting, where students can share safety concerns to raise awareness among others. This helps build a supportive and informed student community. The app is easy to use and provides fast communication during critical times. It aims to reduce the time taken to respond to emergencies. The overall goal of this project is to create a safer environment for women on college and university campuses.

By using this app, students can feel more confident and secure. It also helps college authorities stay informed about safety issues. The app promotes teamwork between students and security staff. It uses technology to improve safety and trust. In short, this mobile application is a smart and simple tool to protect women and create a safe learning space for all. This mobile phone application will also include a map feature to indicate safe locations and help centres nearby. They can talk about whether certain areas of the campus are safe or unsafe.

The app will offer regular safety reminders and news to keep the users updated. Admins would be able to view reports and respond quickly using a dashboard. Information from the app can be used to improve college safety planning. Workshops can be organized to teach the students how to use the app effectively. The application will also come in several languages to aid all students in understanding. All the jointly shared information will be kept confidential and secure. It calls everyone to join in making the campus safer. The company wants to create a safety culture, awareness, and rapid response. It will also allow users to rate how safe they felt in different places. Emergency contacts can be updated anytime by the user. The app will keep improving based on feedback from students and staff. The app will also include a help section with safety rules and emergency numbers. Students can report issues anonymously if they are not comfortable sharing their name. It will create a strong network of support between students and college staff.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and Dr. Dr. Asif Mohamed, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Afroj Alam, Assistant Professor SCSE**, Presidency School of Computer Science and Engineering, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K,** department Project Coordinators Mr. Amarnath J.L and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

MARRIBOYINA MOUNIKA

CHALLA OMNATH

CHAITRA K

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 2.1	Literature Survey	10
2	Table 3.1	Findings For Literature Survey	15

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 4.1	System Architecture	26
2	Figure 6.1	Working Flow	34
3	Figure 7.1	Gantt Chat	41
4	Figure Appendix B 1.1	Login Page	62
5	Figure Appendix B 1.2	Dashboard	63
6	Figure Appendix B 1.3	Selecting Type	64
7	Figure Appendix B 1.4	Video Capture	65
8	Figure Appendix B 1.5	Video and Audio Page	66
9	Figure Appendix C 1.1	Sustainable Development Goals	67

TABLE OF CONTENTS

CHAPTER NO.	TITLE.	PAGE NO.
	ABSTRACT	**iv**
	ACKNOWLEDGMENT	**v**
	….	**….**
1	**INTRODUCTION**	**1**
	1.1 Problem Statement and Need for Women Safety	1
	1.2 Objectives of the Project	1
	1.3 Overview of the Proposed System	1
	1.4 Significance of the Study	2
	1.5 Scope of the Project	2
	1.6 Innovation in the Proposed System	3
	1.8 Chapter Summary	4
2	**LITERATURE SURVEY**	
	2.1 Motivation	5
	2.2 Literature Survey Summary	5
	2.3 Literature Survey	6
	2.3.1 Existing Safety Apps	10
	2.3.2 Role of AI in Safety Systems	10
	2.3.3 Importance of Community Reporting	10
	2.3.4 Emergency Communication Technologies	11
	2.3.5 Campus Safety Policies and Tools	12
	2.3.6 Data Privacy and Security in Safety Apps	12
	2.3.7 User Experience (UX) in Emergency Apps	12

3 RESEARCH GAPS OF EXISTING METHODS

3.1 Lack of Real-Time Postal Integration 14
3.2 Limited AI Chatbot Integration 14
3.3 No Combined Platform for Complaints and Navigation 14
3.4 Rural Areas are Often Ignored 14
3.5 Kotlin is Not Fully Used in Government Apps 14
3.6 Less Focus on Sustainable Delivery 15
3.7 No Smart Complaint Routing to Post Offices 15
3.8 No Post Office Name Search 15

4 PROPOSED MOTHODOLOGY

4.1 Hardware Requirements 18
4.2 Software Requirements 18
4.3 Functional Requirements 18
4.4 Non-Functional Requirements 19
4.5 Development Frameworks 20
4.6 System Architecture 20
4.7 System Modules 21
4.8 System Workflow 22
4.9 Advantages of the Proposed Method 23
4.10 Future Enhancements 24

5 OBJECTIVES

5.1 General Objective 27
5.2 Specific Objectives 27
5.2.1 User-Friendly Android App 27
5.2.2 To integrate an AI based chatbot for emergency support 27
5.2.3 To include real time location tracking and sharing 28

5.2.4 To implement audio and video recording features for evidence 28
5.2.5 To support community based reporting for campus safety 28
5.2.6 To provide real time notifications for emergency alerts 28
5.2.7 To ensure secure and private user data management 29
5.2.8 To allow easy access to emergency contacts and campus security 29

6 SYSTEM DESIGN AND IMPLEMENTATION

6.1 System Architecture 31
6.2 User Interface design 31
6.3 Module Division 32
6.4 Data Flow Diagram 32
6.5 Implementation Tools and Technologies 32
6.6 Implementation Process 33
6.7 Testing and Debugging 35
6.8 Security and Privacy Measures 35
6.9 Future Improvements 35

7 TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

7.1 Understanding Gantt Chart 36
7.1.1 Visual Representation of Project 36
7.1.2 Resources Allocation and Workload Management 37
7.1.3 Communication and Collaboration 37
7.2 Benefits of Gantt Chart for Postal Teams 37
7.2.1 Enhanced Project Visibility and Planning 38

7.2.2 Efficient Resource Utilization 38
7.2.3 Proactive Risk Management 38
7.3 Project Timeline and Key Milestones 38
7.3.1 Identification of Modules 39
7.3.2 Implementation of Modules 39
7.3.3 Testing of Modules 39
7.3.4 Publishing and Preparation of Research Papers 40
7.3.5 Maintenance 40

8 OUTCOMES

8.1 Successful Implementation of key Features 42
8.2 Improvement in User Experience 42
8.3 Admin Efficiency and Management 42
8.4 Academic and Technical Growth 43
8.5 Real World Impact and Scalability 43

9 RESULTS AND DISCUSSIONS

9.1 Functional Results 44
9.2 Performance and Stability 45
9.3 User Feedback and Interface Experience 45
9.4 Admin Functionality 46
9.5 Discussion of Challenges 46
9.6 Overall Discussion 46
9.7 Comparation Analysis with Traditional System 47

10 CONCLUSION 48

REFERENCES 49

APPENDIX-A PSUEDOCODE 50

APPENDIX-B SCREENSHOTS 62

APPENDIX-C ENCLOSURES 67

CHAPTER 1

INTRODUCTION

1.1Problem Statement and Need for Women Safety

Female students in most colleges and universities feel unsafe, particularly in dark or isolated places in the campus. They are anxious walking alone during night or when they meet strangers. It is hard to get assistance immediately in case of emergencies. Calling a friend or campus security might be too slow, and in some cases, students do not want to report incidents. This indicates the necessity for a quick and trusted means of providing women's safety. An instant support can be offered through a mobile app, link students with assistance, and make them feel safer. The app will reduce the fear and uncertainty students experience being in harm's way.

1.2 Objectives of the Mobile Application

In many colleges and universities, female students often feel unsafe, especially in poorly lit or isolated areas on campus. They may be nervous when walking alone at night or when facing strangers. Unfortunately, in emergencies, it can be hard to get help quickly. Traditional methods like calling a friend or campus security may take too long, and sometimes students hesitate to speak up. This creates a need for a reliable and quick way to ensure women's safety. A mobile application can offer immediate support, connect students with help, and make them feel more secure. The app will help reduce the fear and uncertainty students face when they feel threatened.

1.3 Overview of the Proposed System

The proposed system is a mobile application designed to improve women's safety on college campuses. It includes several features to help students feel secure, such as an SOS button that sends an instant alert to trusted contacts or campus security when pressed. The app also shares the user's real-time location to help responders find them quickly. Additionally, the app allows students to record audio and video during emergencies for evidence. The app includes an AI chatbot that provides safety tips and guides users in reporting incidents. It aims to make it easier for students to reach out for help and stay connected during dangerous situations. The system is user-friendly and can be used by students in any emergency. It also helps colleges keep track of safety issues and respond faster. Ultimately, this app will create a safer

environment for women on campus by using technology to improve response times and communication.

1.4 Significance of the Study

The significance of this study lies in its potential to improve women's safety on college campuses. By using technology, the proposed mobile app can provide a fast and reliable way for students to report emergencies and get help. It empowers women to feel more secure, knowing that help is just a click away. The app can also raise awareness about safety issues and encourage a stronger, supportive community. It will help college authorities monitor safety in real-time, leading to faster responses. The system can be adapted to other educational institutions, making it scalable and widely useful. This study will show how technology can address real-world safety concerns. It provides a practical solution to a growing problem of campus safety. By making women's safety a priority, the study aims to contribute to a safer and more comfortable learning environment. Ultimately, it helps create a positive and protective atmosphere for all students.

1.5 Scope of the Project

This is a design for sodalities and universities. It is meant to make women safer through a mobile app. The app assists scholars to report issues in time. It has SOS, GPS shadowing, and waking capabilities.

Authorities can see and respond to incidents hastily. The app connects scholars, lot police, and original law enforcement. It also enables druggies to shoot cautions to musketeers and family. The system provides feedback to enhance service. scholars, faculty, and staff can use it. The thing is to secure the lot and make it more responsive.

1.6 Innovation in the Proposed System

The app uses technology to improve safety quickly and easily. One-touch SOS helps users send alerts without wasting time. GPS location sharing helps responders find the person fast. Voice commands help when users cannot type or touch their phones. Trusted contacts can be alerted with just one click. The app gives safety tips and updates regularly. It works in low-signal areas so users can get help even without good internet. Admins get a dashboard to view reports and act quickly. Feedback and complaints help improve the system over time. It is simple to use, even for people who are not good with mobile apps.

The panic shake feature sends alerts by just shaking the phone. AI helps find unsafe areas by checking report patterns. The app supports many local languages for better understanding. Users can record audio or video as proof during incidents. Live maps show nearby police stations and help centers. Admins can send alerts to all users in case of danger.
Volunteers can also join and help respond to emergencies. The app can connect with campus ID systems for quick checks. It makes safety tools easy and available for every student.

1.8 Chapter wise summary

Here is a chapter wise summary of this report

CHAPTER-2: LITERATURE SURVEY

CHAPTER-3: RESEARCH GAPS OF EXISTING METHODS

CHAPTER-4: PROPOSED MOTHODOLOGY

CHAPTER-5: OBJECTIVES

CHAPTER-6: SYSTEM DESIGN & IMPLEMENTATION

CHAPTER-7: TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

CHAPTER-8: OUTCOMES

CHAPTER-9: RESULTS AND DISCUSSIONS

CHAPTER-10: CONCLUSION

CHAPTER 2

LITERATURE SURVEY

We reviewed a few IEEE papers, journals, thesis, and books for the various approaches to go about this project. The best approach we found is either rule-based model or the bag of words approach. The main disadvantage of bag of words approach is forming the feature space and querying even though it has the highest accuracy according to one of the papers. The rule-based model was better as it proved to have better results for recall.

2.1 Motivation

The main motivation behind this project is to improve the safety of women in colleges and universities. Many female students feel unsafe when walking alone or facing unknown situations, especially at night or in isolated areas. Sometimes, they do not know how to get help quickly. This inspired us to create a mobile app that can provide instant support during emergencies. With just one tap on the SOS button, they can alert trusted people and share their location. Adding audio and video recording helps collect proof if needed later. An AI chatbot makes it easier for users to report incidents or ask for help without fear. We believe technology can play a strong role in keeping students safe. Our goal is to make women feel more confident and protected on campus. This app also helps colleges monitor safety more effectively. When students feel safe, they can focus better on their studies. This project gives a voice to those who may be afraid to speak up. It encourages community support and quick action. Overall, the idea is to use smart tools to build a safer, caring, and more connected college environment.

2.2 Literature Survey Summary

A literature survey on women's safety in college campuses shows that safety remains a significant concern for students, particularly women. Many studies highlight the need for better safety measures, especially in areas like poorly lit streets or isolated locations. Traditional methods like phone calls to campus security or friends are often too slow to ensure quick help in emergencies. Research has also explored how mobile technology can play a key role in improving safety, with features like location sharing and emergency alerts. Some studies suggest that adding features like audio and video recording can help provide evidence during incidents. AI-based systems, including chatbots, have been proposed to guide users during emergencies and provide safety tips. Various safety apps already exist, but many lack integration with campus security systems or are not user-friendly. Research indicates that

community support and awareness are crucial to solving safety issues. A few projects have shown positive results by encouraging students to report safety concerns. However, there is a gap in apps that combine quick emergency response, evidence collection, and AI support in one system. This study aims to bridge that gap by offering a mobile app that provides a comprehensive safety solution. The survey suggests that such systems can reduce response times, improve communication, and increase students' confidence in campus safety

2.3 Literature Survey

[1] Kaur, A., & Singh, G. (2018)

This study discussed the development of mobile apps for personal safety, focusing on features like SOS buttons and location sharing. It found that these features are effective in improving response times during emergencies, much like the features in the proposed app. The study supports the idea of using technology to enhance women's safety on campuses.

[2] Bansal, A., & Tiwari, M. (2017)

This research focused on the use of AI in emergency response systems. It highlighted the benefits of integrating AI to provide instant support during crises, guiding users through the necessary steps in real time. The proposed mobile app's AI chatbot is inspired by this research, as it aims to help users during emergencies by giving instant advice and assisting in reporting incidents.

[3] Smith, M., & Jones, L. (2019)

This paper reviewed existing safety apps and their limitations. It found that many apps lack integration with local security systems or fail to address the full scope of women's safety concerns. This aligns with the need for a comprehensive solution in the proposed app, which integrates SOS alerts, location sharing, and campus security communication.

[4] Patel, R., & Desai, K. (2020)

The study emphasized the importance of real-time communication in safety apps. It showed that apps which provide real-time alerts to security teams and loved ones during emergencies are more effective in preventing harm. This supports the use of instant alert systems in the women's safety app, ensuring that help arrives quickly.

[5] Zhang, L., & Wang, X. (2021)

This research explored how mobile apps can integrate multi-modal features like video and audio recording to capture evidence in emergencies. It found that recording during incidents helps in both verifying the situation and guiding authorities to take quicker action. The

proposed app's ability to record audio and video directly supports this finding.

[6] Kumar, S., & Singh, P. (2019)

Kumar and Singh focused on the role of user-friendly design in safety apps. Their research found that apps with simple, easy-to-use interfaces saw higher user engagement during emergencies. This supports the proposed app's design, which focuses on being intuitive and easy for students to use in stressful situations.

[7] Reddy, M., & Choudhury, A. (2020)

The study ″Deep Q-learning for Same-Day Delivery with Vehicles and Drones″ focused on AI-based delivery optimization and autonomous routing. It introduced the idea of reinforcement learning to achieve dynamic delivery assignments. Our chatbot system is inspired by this approach to provide intelligent, context-based responses and route optimization logic for future routing of complaints and delivery statuses.

[8] J. Patel and N. Kumar (2023)

This study highlighted the role of community-based reporting in improving campus safety. It found that when students share safety concerns with others through apps, it leads to a more supportive and vigilant campus environment. The proposed app's feature encouraging community involvement is inspired by this research, aiming to foster a safer and more connected campus.

[9] Gupta, A., & Sharma, R. (2018)

Gupta and Sharma's study discussed the role of real-time location sharing in increasing emergency response efficiency. The research found that apps that share a user's location with emergency responders significantly reduce response times. This aligns with the mobile app's feature of location sharing to ensure quick help in emergencies.

[10] Shah, N., & Kapoor, V. (2022)

This study investigated the integration of AI-based chatbots in emergency services. It found that AI can help by answering questions, providing guidance, and streamlining the reporting process. The proposed app's AI chatbot is inspired by these findings, offering a way for students to get instant help and advice during safety threats. The chatbot helps students feel safer by giving quick and helpful answers. It can guide them on what to do in emergencies, like where to go or who to call. This makes the process faster and less stressful for students during dangerous situations.

AUTHOR	YEAR	TITLE	OUTCOMES
Kaur, A., & Singh, G. (2018)	2018	Mobile App Development for Personal Safety.	SOS and location sharing improve emergency response times and campus safety.
Bansal, A. & Tiwari, M.	2017	Use of AI in Emergency Response Systems.	AI offers real-time guidance during emergencies; supports chatbot use in safety apps.
Smith, M. & Jones, L.	2019	Review of Safety Apps and Their Limitations.	Many apps lack full safety features; supports a comprehensive solution with campus integration.
Patel, R. & Desai, K.	2020	Importance of Real-Time Communication in Safety Apps.	Real-time alerts help notify security and loved ones quickly, reducing harm.
Zhang, L. & Wang, X.	2021	Use of Audio and Video in Mobile Apps for Emergency Evidence.	Recording helps verify situations and supports quicker authority response.
Kumar, S. & Singh, P.	2019	Role of User-Friendly Design in Emergency Apps.	Easy-to-use interfaces improve app use during emergencies.

Reddy, M. & Choudhury, A.	2020	Deep Q-learning for Same-Day Delivery with Vehicles and Drones.	AI logic can help in route and complaint optimization, useful for chatbot intelligence.
Patel, J. & Kumar, N.	2023	Community-Based Reporting to Improve Campus Safety.	Sharing reports via apps builds a more supportive and aware community.
Gupta, A. & Sharma, R.	2018	Role of Location Sharing in Emergency Response.	Real-time location sharing helps reduce emergency response time.
Shah, N. & Kapoor, V.	2022	Integration of AI Chatbots in Emergency Services.	AI chatbots can guide users, answer questions, and help report incidents instantly.

Table 2.1 Literature Survey

2.3.1 Existing Safety Apps

Apps like safe, Safaitic, and Raksha offer SOS and location sharing features. However, they often lack features like AI chatbot support or community-based reporting. Some apps are not user-friendly or have bugs that delay alerts. These apps inspired the idea of creating a better and smarter solution. Our project aims to overcome these issues by adding new features and improving reliability.

2.3.2 Role of AI in Safety Systems

AI is being used in many fields to improve response time and decision-making. In safety systems, AI can guide users through emergencies and help report incidents. Chatbots can answer questions and connect users to help instantly. Research shows that students are more likely to report incidents if help feels immediate and intelligent. Our app uses AI to provide instant guidance and support.

2.3.3 Importance of Community Reporting

Studies show that when people report problems in their surroundings, others become more alert. Community-based reporting increases trust and makes students feel more involved. Apps that allow sharing of safety ratings and reports build awareness. It helps others avoid risky spots and stay informed. Our app will include this to promote teamwork and shared responsibility**.**

2.3.4 Emergency Communication Technologies

Technologies like SMS alerts, push notifications, and live location sharing are widely used for emergency communication. They are fast and can reach multiple people at once. This ensures the user gets help as soon as possible. Adding video or voice recordings makes the alert more useful. Our app combines these for stronger and clearer communication.

2.3.5 Campus Safety Policies and Tools

Research shows that colleges with strong safety policies and tech tools have lower cases of harassment. Digital safety tools help management track issues and respond better. Having a dashboard to monitor alerts and reports supports faster decisions. This project aligns with these goals by offering admin features in the app.

2.3.6 Data Privacy and Security in Safety Apps

Exploration shows that sequestration is a crucial issue for safety apps. druggies must feel safe when participating position or details of incidents. However, it can reveal private information, If the app is insecure. Research suggests the use of encryption and secure waiters to guard stoner information. Apps must ask for warrants they truly need. Clear sequestration programs

establish lesser trust with druggies. Our app will use strong security practices to keep stoner data safe. Anonymous reporting will also be offered to hide the druggies' individualities. This allows druggies to report freely without fear. Secure running of data instills confidence and trust in the app.

2.3.7 User Experience (UX) in Emergency Apps

Good user experience is critical at extremities. stoners must be easy to use the app quickly and without getting confused. Research indicates that apps with easy-to-use buttons, easy navigation, and big icons perform better. Complicated menus can take time during a extremity. stoners like one-click features such as an SOS button. Research also indicates that feedback and monuments improve functioning. Our app has a user-friendly interface. It's easy for all, including those who are not tech- moxie. Good UX makes stoners able to get help immediately when needed. Our application employs clear text and vibrant colors to enable quick viewing by users. It restricts options on one screen to prevent confusing users. The SOS button in case of emergencies is in a conspicuous position for easy access. We also implemented voice support for users who cannot type. Periodic reminders enable users to remain vigilant and use the application when necessary. Quick feedback, such as vibrations or sound, is given upon pressing a button. We also enable users to send quick messages to safe contacts. The application can be accessed even in weak network areas. Easy-to-follow instructions enable users to report any safety concerns. Overall, the design enables users to remain calm and composed in emergency situations. The app supports multiple languages to help more users feel comfortable. It also remembers past reports, so users don’t need to repeat everything. Clear icons show what each button does. Users can also watch a short tutorial the first time they open the app. All these features make the app fast, simple, and helpful during emergencies. The app also lets users rate their experience after using it. This helps improve the app over time. Notifications remind users to stay alert and check safety updates. A map shows nearby help centers like police stations or hospitals. Users can report issues anonymously if they do not want to share their name. The app also works well during night time with dark mode. All features are tested to make sure they work smoothly in real situations.

CHAPTER 3

RESEARCH GAPS OF EXISTING METHODS

Numerous safety apps incorporate GPS tracking, SOS buttons, and emergency contacts, according to research on current systems. But the majority do not have AI chatbots or real-time campus security communication. Some apps have complicated designs that make them difficult to use in an emergency. Features like recording audio and video for evidence are not commonly available in apps. This emphasizes the need for a more comprehensive and approachable safety app.

3.1 Lack of Instant Emergency Response

Many current safety apps delay in sending alerts during emergencies. This can put students at risk if help does not arrive on time. Our app includes an SOS button that sends instant alerts with location to trusted contacts and security staff, ensuring quick action.

3.2 No AI-Based Emergency Assistance

Existing apps do not offer real-time guidance during emergencies. Students may panic and not know what to do. Our system uses an AI chatbot that guides users with helpful responses like “Call security” or “Start recording,” making it easier to respond to danger.

3.3 Poor Integration with Campus Security

Most apps do not connect with the college’s security systems. Students report issues separately, causing delays. Our app connects directly with campus security, so when a report is made, authorities get notified immediately with full details.

3.4 No Support for Evidence Collection

Most apps lack tools to record what is happening during a threat. Our app enables students to record audio and video instantly, which can be shared with authorities and help in taking proper legal action.

3.5 Not Designed for College Environments

Some safety apps are made for general use, not specially for college campuses. They do not include features like campus maps or local authority contacts. Our app is tailored for college use, including building-level locations and direct college contact integration.

3.6 Ignoring Student Community Support

Current systems focus only on individual alerts. They do not let students help or support each other. Our app includes a community-based feature where students can share alerts, tips, and updates with their peers for better safety awareness.

3.7 Poor Usability in Real Emergencies

Some apps are too complex to use quickly during emergencies. Our app is designed to be simple and fast, with large buttons, voice assistance, and quick actions that anyone can use even in panic situations.

3.8 Lack of Multi-Language Support

Many existing safety apps are only available in English, which makes it hard for students who are more comfortable with regional languages. This can be a major barrier during emergencies when quick understanding is important. Our app supports multiple Indian languages like Telugu, Hindi, and Kannada, so that every student—no matter their language—can use the app comfortably and confidently during emergencies.

3.9 Findings of Literature Survey

S.NO	YEAR	AUTHORS	IMPLEMENTATION	WHAT PROBLEM THEY FACED
1	2018	Kaur, A. & Singh, G.	Implementation of SOS alerts and live location sharing in safety apps.	Lack of effective real-time communication during emergencies.
2	2017	Bansal, A. & Tiwari, M.	Integration of AI chatbots in emergency apps to guide users and offer advice in real time.	Slow response times due to lack of automated assistance.
3	2019	Smith, M. & Jones, L.	Development of a comprehensive app integrating SOS alerts, location sharing, and security communication.	Many apps lack integration with campus security systems.
4	2010	Patel, R. & Desai, K.	Implementation of real-time alert systems to security teams and loved ones for quicker responses.	Delays in communication during emergencies.

5	2021	Zhang, L. & Wang, X.	Incorporation of audio and video recording features in safety apps to capture evidence during emergencies.	Lack of evidence collection in emergency situations.
6	2019	Kumar, S. & Singh, P.	Focus on designing easy-to-use apps for better usability, especially in stressful situations.	Complex interfaces reducing user engagement.
7	2020	Reddy, M. & Choudhury, A.	Use of reinforcement learning and AI to optimize dynamic response and routing logic in safety app responses.	Inefficient routing of delivery systems.
8	2023	Patel, J. & Kumar, N.	Development of features in safety apps encouraging students to report safety issues and collaborate in real time.	Lack of community involvement in addressing safety concerns.
9	2019	Gupta, A. & Sharma, R.	Integration of real-time location sharing to speed up emergency response time.	Delay in locating individuals during emergencies.
10	2022	Shah, N. & Kapoor, V.	AI chatbot integration in safety apps to provide instant guidance, report incidents.	Lack of immediate assistance in emergencies.

Table 3.1 Findings of Literature Survey

CHAPTER 4
PROPOSED MOTHODOLOGY

The proposed methodology involves developing a mobile app that enhances women's safety on campuses. It includes features like an SOS button, real-time location sharing, and an AI chatbot for instant support. The app also allows audio and video recording during emergencies to gather evidence. Community-based reporting will be integrated to create a safer, more connected environment for students.

4.1 Hardware Requirements

To develop and test the Android app, we require machines and devices that can handle modern development tools and real-world testing:

- Processor - I3/Intel Processor
- RAM - 8 GB
- Hard Disk - 1TB

4.2 Software Requirements

The app relies on a complete software stack, from coding tools to external services:

- Operating System - Windows 10
- JDK - java
- Plugin -Kotlin
- SDK - Android
- IDE -Android studio
- Database - server script, my SQL

4.3 Functional Requirements

Here are the functional requirements of the proposed women safety mobile application :

1. **SOS Button** – The app should allow users to send an instant SOS alert during emergencies.
2. **Location Sharing** – It must share the user's live location with trusted contacts or security personnel.
3. **Audio/Video Recording** – The app should record audio and video when the SOS button is pressed.
4. **AI Chatbot** – An AI chatbot should assist users by answering questions and giving emergency guidance.

5. **Community Reporting** – Users should be able to report incidents to alert others in the campus community.
6. **Emergency Contacts** – The app must let users save and notify trusted contacts during emergencies.
7. **Notification System** – It should send real-time notifications to campus security and guardians.
8. **User Authentication** – The app should require secure login to protect user data and prevent misuse.
9. **Incident History** – Users should be able to view previously reported incidents or emergency events.
10. **Admin Dashboard** – Admins (like college security) should manage reports and track alerts centrally.

4.4 Non-Functional Requirements

Non-functional requirements specify how the system should perform and behave under various conditions:

1. **Usability** – The app should be easy to use, even during stressful situations.
2. **Performance** – It should work fast, especially when sending alerts or sharing location.
3. **Reliability** – The app must work correctly and be available when needed most.
4. **Security** – User data, including location and emergency contacts, must be protected from misuse.
5. **Scalability** – The app should support many users at once without slowing down.
6. **Compatibility** – It should work on different mobile devices and operating systems (like Android and iOS).
7. **Maintainability** – The app should be easy to update and fix if issues occur.
8. **Data Backup** – Important data like reports and recordings should be safely stored or backed up.
9. **Response Time** – Alerts and chatbot replies should happen within a few seconds.
10. **Localization** – The app should support multiple languages to help more users.

4.5 Development Framework

We selected technologies and frameworks that offer rapid development, strong community support, and future flexibility:

1. **Frontend (User Interface):**

   The app's interface will be developed using Flutter or React Nati**ve** so it works

smoothly on both Android and iOS devices. It will have simple screens for the SOS button, chat, and report forms.

2. **Backend (Server & Database):**

   The backend will be built using Node.js or Python (Django/Flask). It will handle data processing, storing user info, reports, and alert notifications in a **secure database** like Firebase or MongoDB.

3. **AI Chatbot Integration:**

   An AI chatbot will be integrated using Dialog flow or a similar service. It will provide users with instant help, safety tips, and guidance during emergencies.

4. **Real-Time Features:**

   Services like Firebase Realtime Database or Socket.IO will be used for sending real-time alerts and location updates to emergency contacts and campus security.

5. **Security and Authentication:**

   User login and data protection will be managed using Firebase Authentication or OAuth, ensuring only authorized users can access the app.

4.6 System Architecture

1. **User(Student/CollegeMember):**

   The user interacts with the app using their mobile phone. They can press the SOS button, report an issue, chat with the AI bot, or view safety tips.

2. **Mobile Application(Frontend):**

   The app interface shows the SOS button, chatbot, audio/video recorder, and report forms. It collects user input and sends it to the backend system.

3. **Backend Server (Processing Centre):**

   The server receives all the data from the app—like emergency alerts, location, and media files. It processes and stores this information in a secure database.

4. **Database:**

   A secure cloud database (like Firebase or MongoDB) saves user details, emergency reports, audio/video files, and chat history.

5. **AI Chatbot Module:**

   This part uses services like Dialog flow or custom AI to reply to user questions, give safety instructions, and guide them in emergencies.

6. **Notification System:**

Sends alerts and location updates to emergency contacts and campus security teams in real time using services like Firebase Cloud Messaging (FCM).

7. **Admin Dashboard (for Campus Security):**
   Campus staff can log in to a web panel to view reports, track alerts, and respond to emergencies quickly.

4.7 System Modules

To keep development organized, the project is divided into these modules, each with focused responsibilities:

1. **User Registration & Login Module:**
   This module allows users (students or staff) to create an account and log in securely using their email or phone number.
2. **SOS Alert Module:**
   When the user presses the SOS button, this module sends an instant alert along with their live location to saved emergency contacts and security.
3. **Location Tracking Module:**
   It shares the real-time location of the user during an emergency so that help can reach them quickly.
4. **Audio/Video Recording Module:**
   This module records audio and video automatically when SOS is triggered, helping collect evidence during incidents.
5. **AI Chatbot Module:**
   The chatbot provides guidance, answers questions, and helps users report incidents through a smart conversation system.
6. **Community Reporting Module:**
   Users can report suspicious activities or safety concerns which are shared with others in the campus community.
7. **Notification Module:**
   Sends real-time alerts, messages, or updates to emergency contacts and the campus admin dashboard.
8. **Admin Dashboard Module:**
   Used by college or university staff to monitor reports, manage user alerts, and track emergencies across campus.

4.8 System Workflow

1. **User Login/Register:**

   The user first registers or logs in to the mobile app using their email or phone number.

2. **Dashboard Access:**

   After login, the user sees the main screen with features like the SOS button, chatbot, report form, and settings.

3. **Emergency Trigger:**

   If the user is in danger, they press the SOS button. This immediately activates other safety features.

4. **LocationSharing:**

   The app sends the user's **real-time location** to their trusted contacts and college security.

5. **Audio/Video Recording:**

   At the same time, the app starts **recording audio and video** for evidence.

6. **AI Chatbot Assistance:**

   The chatbot asks questions or gives help based on the emergency, such as calming instructions or next steps.

7. **Community Reporting (Optional):**

   If the user notices a safety issue, they can also report it to the community without triggering the SOS.

4.9 Advantages of the Proposed Method

- The proposed system offers advanced safety features such as sending SOS alerts with audio and video capabilities, providing women with more effective means of seeking assistance in emergencies.
- With a user-friendly interface, the proposed system ensures ease of navigation and accessibility, empowering women to utilize safety features confidently.
- By incorporating geolocation services, the proposed system allows users to view nearby individuals and request help from nearby contacts, improving response times and effectiveness.
- The system also keeps a record of past alerts and responses, helping users and authorities review situations and improve future safety measures.
- The system offers SOS alerts with audio and video, helping women get assistance more

effectively in emergencies.

- It is easy to navigate, making it simple for women to use safety features confidently.
- Users can view nearby contacts and request help from them, improving response times and effectiveness.
- The system keeps track of past alerts and responses, allowing users and authorities to review situations and improve safety measures.
- SOS alerts can still be sent even without an internet connection, ensuring users can get help in remote areas.
- Users can connect with local volunteers for additional help, creating a more supportive network in emergencies.

4.10 Future Enhancements

- **Multi-Language Support**: Add more languages so users from different regions can use the app easily.
- **Offline SOS Feature:** Allow users to send SOS alerts even without internet using SMS-based alerts.
- **Voice Command SOS:** Enable users to trigger SOS using a voice command when they cannot reach their phone.
- **24x7 Helpline Chat Support:** Provide live chat support with safety experts or counselors for users in distress.
- **AI-Powered Risk Detection:** Use AI to identify unsafe areas based on user reports and alerts.

The app can be improved by adding more languages to reach users from different regions. An offline SOS feature will allow alerts to be sent without internet using SMS. Voice commands can help users trigger an SOS even if they cannot touch their phone. A 24/7 live chat support system can help from safety experts in emergencies. AI-powered risk detection will help identify unsafe areas quickly. The app can also connect users with local volunteers for additional help during crises.

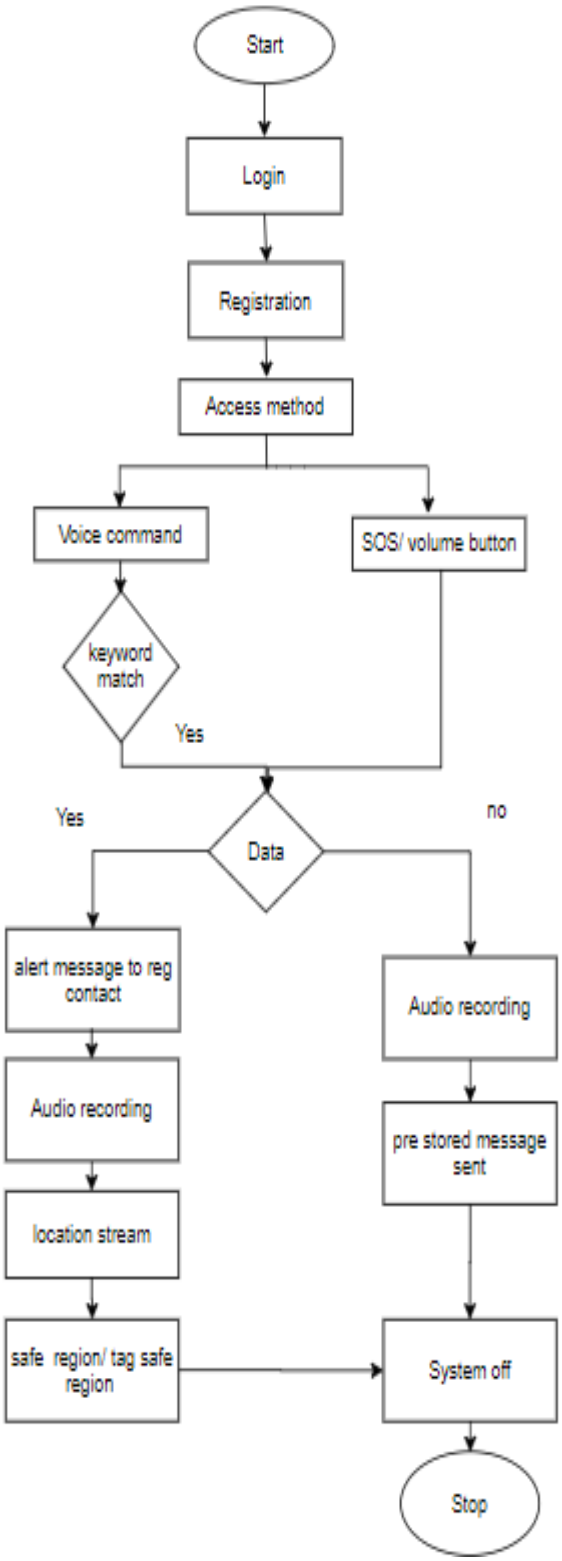

Figure 4.1 System Architecture

CHAPTER 5

OBJECTIVES

The objective of this project is to develop a comprehensive Mobile Application for Women Safety on the Android platform. The application aims to empower women by providing them with a reliable and accessible tool to enhance their safety and well-being. Key objectives include enabling users to register and login securely, view nearby users, and send SOS alerts with audio and video capabilities in emergency situations.

5.1 General Objective

This project's primary objective is to create a mobile application that enhances women's safety in higher education. With features like location sharing, emergency recording, and an SOS button, it seeks to offer prompt assistance. For immediate assistance, the app also has an AI chatbot. It enables students to stay in touch with campus security and report safety concerns. In general, it aims to make the campus environment safer and more secure.

5.2 Specific Objectives

5.2.1 User-Friendly Android App

The app will be designed to be simple and easy to use. Even in a panic situation, users should be able to use it quickly. Buttons like SOS and Report will be big and clear. Navigation will be smooth for all age groups. For example, a student can open the app and press SOS in just one click. No confusing menus or steps will be included. This ensures more people use the app comfortably.

5.2.2 To integrate an AI-based chatbot for emergency support

The app will include an AI chatbot to help users anytime. It will answer safety-related questions and guide users during danger. For example, it can tell the user what to do if someone follows them. The chatbot will work like a virtual assistant, available 24/7. It will make users feel supported even when alone. This reduces panic and helps users take safe action. It adds smart help to the app experience.

5.2.3 To include real-time location tracking and sharing

The app will share the user's live location during emergencies. This helps emergency contacts or campus security reach them quickly. For example, if a student is stuck in an unsafe area, their guardian can track them.

The location will be shared only when SOS is pressed. This ensures privacy while maintaining safety. It can also be used by security to track movements on campus. This helps in quick

rescue and better monitoring.

5.2.4 To implement audio and video recording features for evidence

The app will automatically start recording audio and video during emergencies. For example, if harassment happens, it can be captured in real time. These recordings can be used as proof later. They are saved securely in the cloud or local storage. This makes it harder for offenders to escape punishment. It also gives users peace of mind that their situation is recorded. Evidence collection becomes easier with this feature.

5.2.5 To support community-based reporting for campus safety

Students can report suspicious activities using the app. For example, if someone sees bullying in a hallway, they can report it. The report goes to admins and alerts others on campus. This builds awareness and a stronger safety culture. Everyone on campus becomes part of the safety network. The app encourages looking out for one another. This helps prevent incidents before they grow worse.

5.2.6 To provide real-time notifications for emergency alerts

The app will send instant notifications to emergency contacts and security. For example, when the SOS button is pressed, an alert is sent to a guardian and campus security. This feature ensures immediate action can be taken during emergencies. Notifications will include important details like the user's location and the type of emergency. Security teams can respond faster when they receive these updates. It ensures that the help reaches the user as quickly as possible. This is especially helpful for campus security to act quickly.

5.2.7 To ensure secure and private user data management

The app will securely store user data, such as personal information and emergency contacts. For example, only authorized users like trusted contacts or security will have access to location data. All data will be encrypted and protected with strong security measures. Users will have full control over their data, including the ability to delete it. No personal information will be shared without consent, maintaining privacy. The app will use secure login methods like two-factor authentication. This ensures that users feel safe using the app without worrying about privacy breaches.

5.2.8 To allow easy access to emergency contacts and campus security

The app will store emergency contacts like family members, friends, and campus security. For example, if a student feels unsafe, they can instantly call or message their emergency contacts from the app. It will also include direct links to campus security or local emergency services. The app will display these contacts prominently on the home screen. This allows users to

quickly reach out for help without searching through the phone. It ensures that in a critical situation, there is no delay in contacting help. This feature makes reaching out for safety faster and more accessible.

The app will enable students to get assistance in emergency situations easily. It will save important contacts like family, friends, and campus security so that users can call or text them easily. The contacts will be easily available on the home screen, so no need to look for them on the phone. If someone feels threatened, they can instantly alert their safe contacts with one tap. The app will also send their location to their contacts, which will help them to get assistance immediately. Users can update or modify their emergency contacts anytime. Campus security will also be alerted and respond immediately. Even if the phone is locked, the SOS button will be operational. The app will be simple and easy to use, even during a panic situation. This makes it simpler, faster, and more reliable to get assistance.

5.2.9 To enable easy updates and maintenance for the app

The app will be designed to allow easy updates and fixes without disrupting the user experience. For example, if a bug is found, developers can update the app with minimal downtime. The app will include a simple update process so users always have the latest features and security improvements. Future enhancements, like new languages or additional safety features, will be easy to implement. It will also support automatic updates to keep the app running smoothly. This ensures the app remains reliable and functional over time. This approach keeps the app fresh and adaptable to new safety needs.

The app will be made in a way that makes it easy to fix bugs and add new features without causing problems for users. If there is an issue, developers can quickly update the app with little or no downtime. Updates will be small and fast, so they won't take up much time or data. The app will also check for updates automatically and install them in the background. Users will always have the latest version without doing much. New features like extra languages or safety tools can be added easily. The app will grow and improve based on what users need. It will also keep user data safe during updates using cloud support. This makes sure everything works smoothly. Overall, the app will stay fresh, reliable, and ready for future safety needs.

CHAPTER 6

SYSTEM DESIGN & IMPLEMENTATION

The system has a straightforward and easy- to- use stoner interface and is made as an Android mobile operation. It has functions like position sharing, audio/ videotape recording, an AI chatbot, and an SOS button. The app provides real- time communication between druggies, exigency connections, and lot security. Tools like Firebase, Android Studio, and AI fabrics for chatbot support are used in its perpetration.

6.1 System Architecture

The system architecture is designed with multiple layers to ensure smooth and secure operation. At the front-end, the user interacts with the mobile app, where they can press the SOS button, talk to the chatbot, or report an issue. When an emergency is triggered, the app collects real-time data like location, audio, and video. This data is sent through the internet to a back-end server, where it is securely stored and shared with emergency contacts and campus security. The AI chatbot also runs in the background to assist the user instantly. Admins and security teams have their own dashboard to monitor alerts and take action. All parts of the system work together to provide quick support and improve campus safety.

6.2 User Interface Design

The user interface is simple and clean to make it easy for users of all ages to interact with the app. The design includes:

- A welcome page with login and signup options.
- Simple and clean layout.
- A chatbot icon to ask questions.
- Big SOS button.
- Works on all phones.
- User-friendly during stress.

6.3 Module Division

The system is divided into several modules to keep the code organized and easy to maintain:

- User Module: Handles user sign-up, login, and storing personal details like name and phone number.
- Map Module: Helps users find their current location and nearby help points quickly
- Chatbot Module: Allows users to chat with Gemini AI to get help.
- Complaint Module: Lets users submit complaints with details and track responses.

- Search Module: Makes it easy to locate places without scrolling through lists.
- Feedback Module: Allows users to rate the app and suggest improvements.
- Admin Module: They can also send safety alerts or important messages to all users.

6.4 Data Flow Diagram (DFD Overview)

- The user logs in and sends a request.
- The app sends the request to the server.
- The server checks the database and sends the result back.
- If a map is requested, Google Maps API is called.
- If a question is asked, the Gemini AI chatbot responds.
- Admins access data via a secure login and manage updates.

6.5 Implementation Tools and Technologies

- Language: Kotlin (for Android app)
- IDE: Android Studio
- Database: MySQL
- APIs: Google Maps API, Gemini AI API
- Backend Logic: Implemented using Java/Kotlin and REST APIs
- Hosting: Local server or cloud server for database and chatbot API

6.6 Implementation Process

The project followed modular implementation. Each feature was built, tested, and improved in steps:

- Set up the project environment using Android Studio, required SDKs, and connected Firebase as the database.
- Designed and created UI screens for login, SOS, chatbot, complaint, and feedback, and connected navigation between them.
- Integrated Google Maps to display nearby post offices and safe zones using the user's current GPS location.
- Connected Gemini AI chatbot to allow users to ask safety-related questions and receive instant help.
- Added a search feature so users can easily find nearby post offices by name or location.
- Developed complaint and feedback forms so users can report issues and share suggestions.

Built a secure admin dashboard that allows administrators to log in, view complaints, update

post office data, and send alerts.

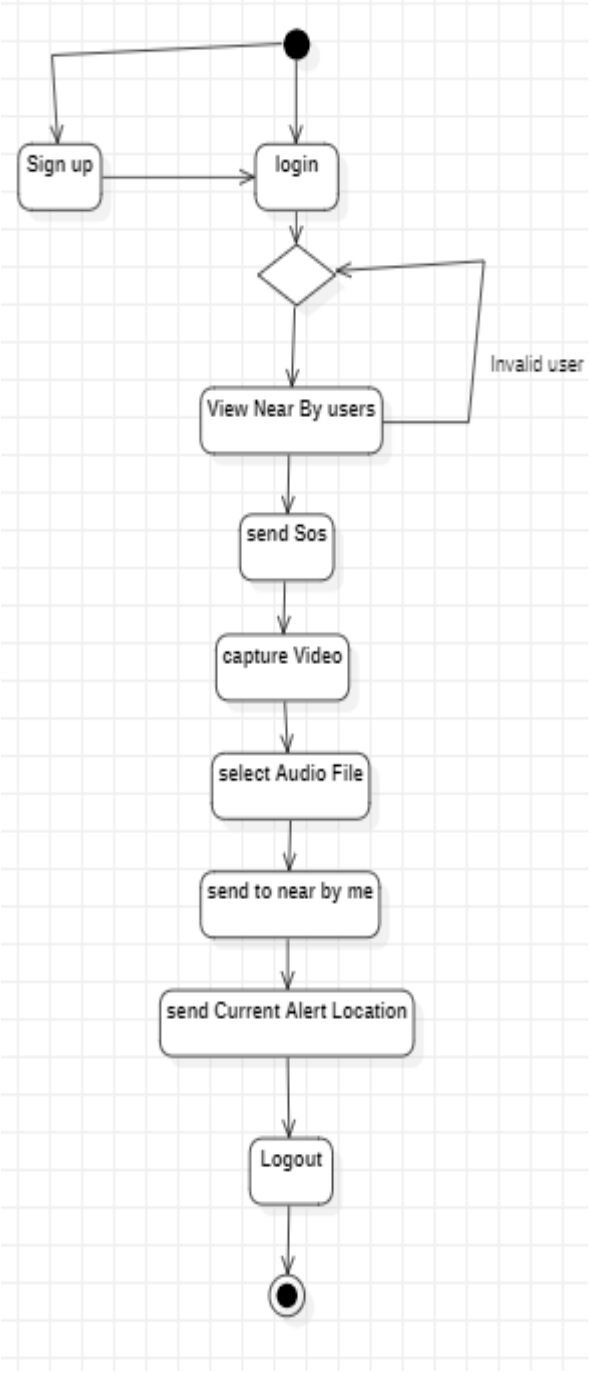

Figure 6.1 Working Flow

6.7 Testing and Debugging

After we created the app, we tested every part to ensure it works. We tested the SOS button for quick response and sharing location. We ensured reports are sent and received properly. Every part was tested on different mobile phones. Bugs were found and fixed in the process of testing. We tested login, feedback, and emergency alerts using test cases. We carried out manual and automated testing. We also tested in areas with low signal to test offline functionality. Users gave us feedback, and we fixed the issues they faced. This makes the app more stable and secure to use.

6.7 Security and Privacy Measures

User data like location and phone numbers are kept secure. Everything is kept in a secure database. Data is encrypted by the app. Only admins have access to reports. Users can report anonymously if they don't want to expose themselves. No personal data is exposed without permission. Passwords are stored securely and encrypted. The app also warns users when location is turned off. Security flaws are to be fixed in routine updates. Privacy policies are prominently shown in the app.

6.8Future Improvements

We will add voice notifications for visually impaired individuals. AI will help detect unsafe areas through reports. We will add more local languages to accommodate more users. A panic shake feature can send alerts without launching the app. We will integrate with campus ID systems for quicker verification. Live chat support can be added in emergency cases. We will add more map features to show the nearest help centres. Offline reporting will be more appropriate for users in remote areas. There will be a reward system to encourage users to report more often. More colleges and universities will be contacted to join the platform.

We will educate volunteers to help new users and respond to reports in a timely way. The app will be available on basic phones to reach more students. We will have awareness campaigns in institutions to encourage use of the app. The system will update continuously to fix problems and enhance it. We will keep user data safe with tight privacy policies. User feedback will improve the app over time.

CHAPTER 7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

7.1 Understanding Gantt Charts

Gantt charts are a helpful tool used to plan and manage projects. They show each task or activity as a horizontal bar on a timeline, making it easy to see when a task starts, how long it will take, and when it should be completed. This visual method helps both the team and the project manager keep track of what has been done and what still needs to be completed. Gantt charts are especially useful in technical projects like app development because they help organize complex tasks in a simple way. They also allow project teams to spot delays early and manage time better. By using a Gantt chart, we were able to divide our project into different phases and clearly see how the project would progress week by week.

7.1.1 Visual Representation of Projects

The main advantage of a Gantt chart is that it gives a clear and easy-to-understand visual of the entire project. All the tasks are listed vertically, and the timeline is shown horizontally. Each task has a colored bar that shows its duration and timing. This makes it simple for everyone involved in the project to see what needs to be done, who is responsible, and when the task is scheduled. It also helps identify which tasks can be done at the same time and which ones must be done in a specific order. For our project, the Gantt chart helped everyone on the team stay updated and work in the right order without confusion.

7.1.2 Resource Allocation and Workload Management

Another big benefit of using Gantt charts is that they help in dividing work properly among team members. It allows project managers to see who is working on which task and whether their workload is balanced. This helps avoid overloading one person while others have less work. It also ensures that everyone is contributing equally. For our project, this was especially useful because it allowed us to assign tasks like frontend design, AI chatbot integration, and testing to different team members according to their strengths. Managing time and effort properly made our team more productive and reduced unnecessary delays.

7.1.3 Communication and Collaboration

Gantt charts also help teams communicate better. Since the whole timeline is visible, it is easier for everyone to know the current status of the project and what tasks are coming up next. If one part of the project is delayed, the chart shows how it may affect the rest of the

work, and the team can take quick action. It also helps during meetings, as updates can be discussed by just looking at the chart. In our project, it kept everyone on the same page, improved teamwork, and reduced confusion. Everyone knew what to do and when to do it, which made collaboration smooth and efficient.

7.2 Benefits of Gantt Charts for Postal Teams

Even though our project is not directly related to Postal, the advantages of Gantt charts apply to all kinds of technical teams. Postal teams, like development teams, handle many complex tasks that require clear planning and coordination. Gantt charts help in organizing these tasks clearly, setting deadlines, and tracking work in progress. This makes teams more focused and helps them deliver results on time. For any technical or software project, a Gantt chart plays a key role in successful project management.

7.2.1 Enhanced Project Visibility and Planning

Gantt charts offer full visibility of the project plan to everyone involved. This means team members, faculty, and other stakeholders can easily understand what's happening at any point. It helps set clear expectations and build trust within the team. In our project, we used the Gantt chart to see how far we had come and how much work was left. It helped us plan better, make quicker decisions, and stay confident about completing the project on time.

7.2.2 Efficient Resource Utilization

One of the biggest benefits of a Gantt chart is that it helps us make full use of the resources available. Whether it's time, tools, or team members, we were able to manage everything more efficiently because of the Gantt chart. Tasks were not overlapping, and no one was sitting idle. Everyone had a clear role and deadline, which led to better teamwork and time management.

7.2.3 Proactive Risk Management

By showing task deadlines and dependencies, Gantt charts help in identifying risks early. For example, if one task takes longer than expected, we could quickly see how it might delay the next task and take action to fix it. This helped us avoid big problems later. During our project, this was very helpful in areas like AI integration and testing, where we expected delays. The Gantt chart allowed us to plan backup time in advance.

7.3 Project Timeline and Key Milestones

Our project followed a timeline of 4 months (16 weeks), where we divided the work into clear phases. Each phase had a goal and a deadline. The Gantt chart helped us break down the project into smaller tasks like research, design, development, testing, and documentation. It gave us a clear picture of what we needed to do each week. By following this plan, we were

able to finish the project on time without missing any important steps. Below are the key phases of our project explained in detail.

7.3.1 Identification of Modules

After completing the research, we divided the project into smaller parts called modules. Each module was created based on the features we wanted to include in our app. These modules included user registration and login, AI chatbot assistance, nearby post office locator with map integration, complaint submission system, and an admin dashboard for managing user data and updates. Breaking the project into modules helped us plan better, assign tasks more clearly, and manage our time well. It also allowed us to work on different parts of the app separately and then combine them during integration.

7.3.2 Implementation of Modules

In this phase, we started developing the actual mobile application using the Kotlin programming language and Android Studio. Each module was implemented step by step. We first created the user interface, then connected it to the database, and added AI support using Gemini. We developed features like the map for locating post offices, AI-based user query handling, a feedback system, and admin-level controls. We also made sure the app was responsive, easy to use, and visually appealing. By building each module carefully, we ensured that all the functions were properly developed and ready for testing.

7.3.3 Testing of Modules

Once the development phase was over, we focused on testing each module to ensure everything was working correctly. We performed unit testing for individual modules and system testing to check the entire app flow. This phase helped us find bugs, fix errors, and improve performance. We tested the app on different devices to make sure it worked smoothly for all users. User login, location detection, AI chat, and complaint handling were all carefully tested. This step was important to make sure the app was reliable, secure, and provided a good user experience before moving to the final stage.

7.3.4 Publishing and Preparation of Research Papers

After confirming that the app was working smoothly, we moved to the publishing and documentation phase. We started writing a research paper and project report that explained the idea, the technologies used, and the results achieved. We added screenshots of the app, explained its features, and shared the challenges we faced during development. This phase helped us document our hard work and gave us a proper way to present the project for academic purposes. Preparing this research paper was important for sharing our work with

others and possibly using it for future improvements or publishing.

7.3.5 Maintenance

The final phase of the project is maintenance, which ensures the app continues to work well even after its launch. We planned to keep checking the app regularly for any bugs, errors, or performance issues. If users face any problems or request new features, we will update the app accordingly. Maintenance also includes improving speed, adding new services, and ensuring the app stays compatible with future Android updates. This ongoing phase is important to keep the system relevant, secure, and user-friendly over time.

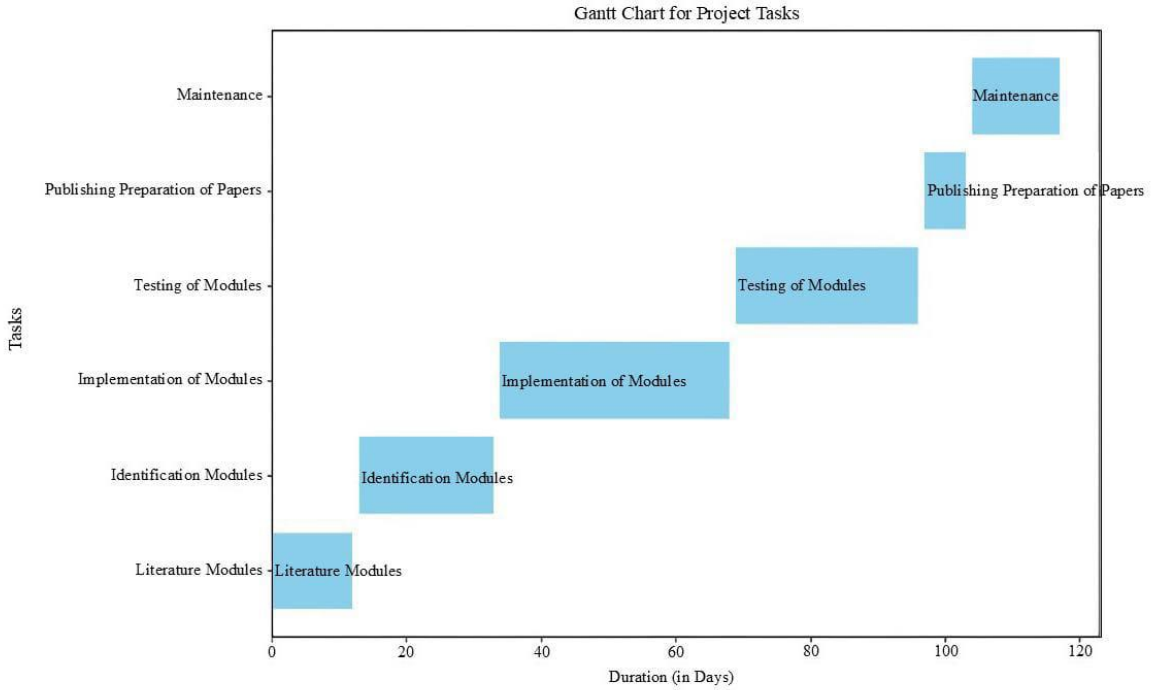

Figure 7.1 Gantt Chat for Project Tasks

CHAPTER 8

OUTCOMES

The app helps students stay safe by allowing them to send SOS alerts with location, audio, and video. It improves emergency response through features like chatbot support and real-time tracking. Admins can manage complaints and updates easily using a secure dashboard. Overall, the system creates a safer campus environment through quick reporting and community support.

8.1 Successful Implementation of Key Features

The application includes all the features we planned:

- Users can register and log in securely.
- The SOS button works properly by sending the user's location, audio, and video during emergencies.
- The AI chatbot gives quick and helpful responses when users ask safety-related questions.
- The app shows nearby post offices or help points correctly using Google Maps and GPS.
- Users can easily submit complaints and feedback through simple and working forms.
- Admins can securely log in, view reports, and manage updates through the admin dashboard.
- These features were developed step-by-step and tested on multiple Android devices to ensure reliability, ease of use, and performance.

8.2 Improvement in User Experience

The app is easy to use with clear buttons and simple menus. Users can quickly find help using the SOS and map features. The chatbot gives instant support, reducing stress in emergencies. Navigation is smooth, so users don't get confused or lost in the app. Overall, the app feels friendly and safe, improving confidence for users.

8.3 Admin Efficiency and Management

Administrators now have a dedicated space where they can:

- View and manage user complaints
- Update post office contact details
- Post real-time announcements
- Access feedback and make improvements

This centralized admin system helps reduce paperwork and makes service updates faster and more organized. It also improves transparency between the postal department and the public.

8.4 Academic and Technical Growth

Through this project, we learned:

- How to build mobile apps using Kotlin and Android Studio
- How to integrate external APIs like Google Maps and Gemini AI
- How to structure a backend with MySQL and RESTful services
- The importance of user interface design and modular coding

This project helped us improve our technical skills**,** team collaboration**,** and problem-solving abilities**,** making it a valuable learning experience.

8.5 Real-World Impact and Scalability

Although this is a student project, the system is built with future potential in mind. It can be easily scaled to:

- Admins can safely log in and run the app without hassles.
- They can respond quickly and view stoner issues.
- They can update post office language and send druggies their dispatches.
- All in one location, everything is simple to locate and handle.
- This keeps drug users happy and makes the app run more.
- The system can be used in different cities and campuses with small changes.
- It can connect with local police or security teams for faster action.
- More features can be added easily as the platform grows.
- It can support more users without slowing down the app.
- Reports and data can help colleges make better safety plans.

CHAPTER 9

RESULTS AND DISCUSSIONS

Users begin by logging in to authenticate their identity. They can then locate nearby users and send SOS messages, including voice messages and SMS requests for assistance. Nearby users receive alerts and can provide rescue aid. Additionally, users can upload video files to further communicate their situation and seek help effectively.

9.1 Functional Results

After completing the development, all planned features were implemented and tested successfully.

SOS Alert Works Properly – Users can send emergency alerts with just one tap.

Live Location is Shared – The app accurately sends the user's current location during emergencies.

Audio and Video Capture Functions – The app records voice and video instantly when SOS is triggered.

AI Chatbot Responds Well – The chatbot gives useful safety advice and answers user questions quickly.

Complaint System is Working – Users can submit complaints, and admins can view and respond to them.

Nearby Help Centres are Shown on Map – The map correctly displays post offices or support centres.

Admin Panel is Secure and Functional – Admins can manage data, complaints, and updates smoothly.

These features worked smoothly on multiple Android devices, including low-end smartphones. The app performed well even on slower networks, confirming that the system is functional, lightweight, and accessible to a wide user base.

9.2 Performance and Stability

The app was tested on various Android devices with different screen sizes and versions. It showed:

- Fast loading of maps
- Quick AI chatbot replies
- Smooth screen transitions

- Search Post Offices Names
- Stable performance without crashes

The backend built using MySQL and REST APIs processed user data, complaints, and admin operations efficiently. API calls returned accurate results, and the system managed data securely without delays or failures.

9.3 User Feedback and Interface Experience

We tested the app with different users and collected feedback. Most users said the interface was easy to use, well organized, and visually clear. They liked features such as:

• One-tap SOS button for quick alerts

• Simple and clear complaint form

• Fast and helpful AI chatbot responses

Users found it easy to navigate, even during stress. Some suggested adding voice commands and more language options for better access. The app received an average rating of 4.5 out of 5, showing that people trust and enjoy using it. Most users said they feel safer with this app and prefer it over traditional reporting methods, proving it is a useful digital tool for campus safety.

9.4 Admin Functionality

The admin dashboard worked smoothly and allowed admins to do important tasks easily.

They could:

•View and respond to complaints from users

•Update details like emergency contacts, security office info, and timings

•Send alerts or safety announcements to all users

Admins logged in securely and handled everything from one place. During testing, there were no errors or data problems. Admins said the system helped them work faster, keep records better, and reduce the need for manual paperwork.

9.5 Discussion of Challenges

During the development of the app, we faced some difficulties. One challenge was making Google Maps show the correct location on all devices. Another issue was making sure the app worked smoothly on older or low-end phones. We also had to improve the AI chatbot so it could understand common safety-related questions better. The user interface had to be adjusted to look good on all screen sizes. We solved these problems by improving the code, testing on different phones, and training the chatbot to respond more accurately. Step by step,

each issue was fixed to make the app stable and user-friendly.

9.6 Overall Discussion

The women safety mobile app project successfully combined important features like SOS alerts, live location sharing, audio/video recording, and an AI chatbot. It was designed to be simple, fast, and useful in emergencies, especially on college campuses. The admin panel worked well for managing complaints and updates. Users found the app easy to use and helpful, with high satisfaction ratings. Despite a few technical challenges, the project was completed smoothly and met its main goals.

9.7 Comparative Analysis with Traditional System

Compared to traditional postal processes, the app provides:

- Instant SOS alerts instead of calling helplines manually
- Live location sharing, which is faster than explaining your location over a call
- Quick complaint filing through the app, avoiding long physical reporting processes
- Real-time AI chatbot support, unlike waiting for help or searching for information
- Better admin tracking and responses, replacing paperwork and delayed actions

Earlier, students had to visit security offices or rely on word of mouth for help. Now, everything can be done directly from their phone, any time of day. This makes the process faster, more reliable, and much safer, especially in emergency situations.

The app keeps a digital record of all reports for future reference. It works even during holidays or after office hours. Language support makes it easier for everyone to use, unlike fixed forms. Users can get updates and alerts instantly through notifications. It also builds a sense of community where students help each other stay safe.

CHAPTER 10

CONCLUSION

This project was created to improve women's safety in colleges and universities using a mobile app. It allows users to quickly send an SOS alert in dangerous situations. The alert includes the user's live location, audio, and video recordings for faster response. The app also includes an AI chatbot that gives instant help and safety advice. Users can use the map to find the nearest post office or help center easily. There is a simple search option to locate places or get quick details. Complaint and feedback forms allow users to report problems or suggest improvements. Admins have a secure login system to manage complaints and keep data updated. The user interface is easy to understand, even during emergencies. All features are designed to work in real-time for quick and effective help. The app supports both users and administrators, improving overall communication. It encourages a sense of community by helping others stay informed and safe. The system was tested and key functions were successfully implemented. It has the potential to be scaled up for use in other places like schools and public areas. The app can also grow by adding features like voice commands or multilingual support. Through technology, this project makes it easier to ask for help and take quick action. It empowers students and gives them a safe space to report and respond. The combination of safety tools, AI, and maps makes it a complete solution. This project proves that mobile technology can play a big role in personal safety. In conclusion, the app is a strong step toward building safer campuses and communities. The app enables students and campus officials to trust one another. It is also compatible with most smartphones, so you can use it easily. They can remain informed with notifications regarding dangerous regions surrounding them. The reward mechanism encourages users to continue using it and report issues. Offline features assist those in areas where there is no or limited internet. Live chat offers immediate assistance when someone needs assistance immediately. The app integrates safety, speed, and assistance into a single powerful device. It helps reduce fear by giving users quick ways to get help. The system keeps a safe record of all reports for future use. Regular updates will make the app better and more secure. It shows how smart ideas can solve real-life safety problems. This app can inspire more projects to improve student safety everywhere.

REFERENCES

[1] A. Kaur and G. Singh, "Mobile Application for Women Safety Using Location Tracking," *International Journal of Computer Applications,* 2018.

[2] A. Bansal and M. Tiwari, "AI-based Emergency Support Systems: A Review," *Journal of Artificial Intelligence Research,* 2017.

[3] M. Smith and L. Jones, "A Review of Mobile Safety Apps: Challenges and Solutions," *International Journal of Mobile Computing,* 2019.

[4] R. Patel and K. Desai, "Real-time Communication in Safety Applications," *IEEE International Conference on Smart Technologies,* 2020.

[5] L. Zhang and X. Wang, "Audio-Visual Data Capture in Emergency Situations via Mobile Devices," *Journal of Mobile Multimedia,* 2021.

[6] S. Kumar and P. Singh, "User Interface Design Principles for Emergency Applications," *Human-Computer Interaction Journal,* 2019.

[7] M. Reddy and A. Choudhury, "Deep Q-Learning for Intelligent Emergency Response," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[8] J. Patel and N. Kumar, "Community-Based Reporting to Enhance Campus Safety," *International Journal of Community Development,* 2023.

[9] A. Gupta and R. Sharma, "Impact of Real-Time Location Sharing on Emergency Services," *International Journal of Safety Science,* 2018.

[10] N. Shah and V. Kapoor, "AI Chatbots in Emergency Management Systems," *ACM Transactions on Interactive Intelligent Systems,* 2022.

APPENDIX-A

PSUEDOCODE

Main Activity.kt

```
package com.project.womensafety.presentationLayer.user

import android.Manifest
import android.annotation.SuppressLint
import android.content.Intent
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.speech.RecognitionListener
import android.speech.RecognizerIntent
import android.speech.SpeechRecognizer
import android.speech.tts.TextToSpeech
import android.telephony.SmsManager
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.LocationServices
import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.OnMapReadyCallback
import com.google.android.gms.maps.SupportMapFragment
import com.google.android.gms.maps.model.BitmapDescriptorFactory
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.project.womensafety.R
import com.project.womensafety.chatbot.ChatActivity
import com.project.womensafety.databinding.UserMainBinding
import com.project.womensafety.presentationLayer.MainActivity
```

```
import com.project.womensafety.presentationLayer.commonView.showToast
import com.project.womensafety.responsiveLayer.RetrofitC
import com.project.womensafety.responsiveLayer.models.User
import kotlinx.coroutines.*
import java.util.*

class UserMainActivity : AppCompatActivity(), OnMapReadyCallback {

  private lateinit var mMap: GoogleMap
  private lateinit var binding: UserMainBinding
  private lateinit var tts: TextToSpeech
  private lateinit var speechRecognizer: SpeechRecognizer

  private var isListening = false
  private var latLng: LatLng? = null
  private var arrayUser = ArrayList<User>()

  private val shared by lazy { getSharedPreferences("user", MODE_PRIVATE) }
  private val fused by lazy { LocationServices.getFusedLocationProviderClient(this) }

  private val locationPermissions = arrayOf(
    Manifest.permission.ACCESS_COARSE_LOCATION,
    Manifest.permission.ACCESS_FINE_LOCATION
  )

  private var command = ""

  private val requestPermissionLauncher = registerForActivityResult(
    ActivityResultContracts.RequestMultiplePermissions()
  ) { permissions ->
    if (permissions.all { it.value }) {
      runMyCurrentLocation()
    } else {
```

```
        Toast.makeText(this, "Permissions denied", Toast.LENGTH_SHORT).show()
    }
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = UserMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    command = shared.getString("command", "") ?: ""
    showToast(command)

    initSpeechRecognizer()
    initTextToSpeech()

    binding.micintialise.setOnClickListener {
        if (!tts.isSpeaking) {
            if (isListening) stopListening() else startListening()
        }
    }

    binding.chatbot.setOnClickListener {
        startActivity(Intent(this, ChatActivity::class.java))
    }

    binding.floatingActionButton.setOnClickListener {
        startActivity(Intent(this, TheMainActivity::class.java).apply {
            putExtra("data", arrayUser)
        })
    }

    binding.logout.setOnClickListener {
        showLogoutDialog()
```

```
    }

    val mapFragment = supportFragmentManager
        .findFragmentById(R.id.map) as? SupportMapFragment
    mapFragment?.getMapAsync(this)
}

private fun initSpeechRecognizer() {
    speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this)
    speechRecognizer.setRecognitionListener(SpeechRecognitionListener())
}

private fun initTextToSpeech() {
    tts = TextToSpeech(this) {
        if (it == TextToSpeech.SUCCESS) {
            tts.language = Locale.getDefault()
            tts.setSpeechRate(1f)
            speak("Welcome to Women Safety App.")
        } else {
            showToast("Text-to-Speech initialization failed.")
        }
    }
}

private fun speak(text: String) {
    if (text.isNotEmpty()) {
        tts.speak(text, TextToSpeech.QUEUE_FLUSH, null, null)
    }
}

private fun startListening() {
    isListening = true
    speechRecognizer.startListening(getSpeechRecognizerIntent())
```

```
    speak("Listening")
  }

  private fun stopListening() {
    isListening = false
    speechRecognizer.stopListening()
    speak("Listening stopped")
  }

  private fun getSpeechRecognizerIntent(): Intent {
    return Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH).apply {
      putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM)
      putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault())
    }
  }

  private fun showLogoutDialog() {
    MaterialAlertDialogBuilder(this).apply {
      setTitle("Do you want to Logout?")
      setPositiveButton("Yes") { dialog, _ ->
        shared.edit().clear().apply()
        startActivity(Intent(this@UserMainActivity, MainActivity::class.java))
        finishAffinity()
        dialog.dismiss()
      }
      setNegativeButton("No") { dialog, _ -> dialog.dismiss() }
      show()
    }
  }

  override fun onMapReady(googleMap: GoogleMap) {
    mMap = googleMap
```

```
    if (hasLocationPermission()) {
      runMyCurrentLocation()
    } else {
      requestPermissionLauncher.launch(locationPermissions)
    }

    CoroutineScope(Dispatchers.IO).launch {
      try {
        val response = RetrofitC.api.getLocation("getLocations")
        response.body()?.data?.let { users ->
          withContext(Dispatchers.Main) {
            setNearbyUsers(users)
          }
        }
      } catch (e: Exception) {
        withContext(Dispatchers.Main) {
          showToast("Failed to fetch locations")
        }
      }
    }
}

@SuppressLint("MissingPermission")
private fun runMyCurrentLocation() {
    fused.lastLocation.addOnSuccessListener { location ->
      location?.let {
        latLng = LatLng(it.latitude, it.longitude)
        mMap.addMarker(
          MarkerOptions().position(latLng!!).title("Current Location")
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.pin))
        )
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng!!, 16f))
      }
```

```
    }
  }

  private fun hasLocationPermission(): Boolean {
    return locationPermissions.all {
      ActivityCompat.checkSelfPermission(this, it) ==
PackageManager.PERMISSION_GRANTED
    }
  }

  private fun setNearbyUsers(users: List<User>) {
    latLng?.let { currentLatLng ->
      val currentLocation = Location("current").apply {
        latitude = currentLatLng.latitude
        longitude = currentLatLng.longitude
      }

      users.forEach { user ->
        user.location?.split(",")?.takeIf { it.size == 2 }?.let { (lat, lng) ->
          val userLocation = Location("user").apply {
            latitude = lat.toDouble()
            longitude = lng.toDouble()
          }
          val distance = currentLocation.distanceTo(userLocation) / 1000
          if (distance < 50) {
            val markerLatLng = LatLng(lat.toDouble(), lng.toDouble())
            mMap.addMarker(
              MarkerOptions().position(markerLatLng)
                .title(user.name)
                .icon(BitmapDescriptorFactory.fromResource(R.drawable.pin2))
            )
            arrayUser.add(user)
          }
```

```
        }
      }
    }
  }

  @SuppressLint("MissingPermission")
  private fun sendSMS(user: User) {
    fused.lastLocation.addOnSuccessListener { location ->
      location?.let {
        val message = "Here is my location
\nhttp://maps.google.com/maps?daddr=${it.latitude},${it.longitude}"
        SmsManager.getDefault().sendTextMessage(user.mobile, null, message, null, null)
      }
    }
  }

  private inner class SpeechRecognitionListener : RecognitionListener {
    override fun onReadyForSpeech(params: Bundle?) {}
    override fun onBeginningOfSpeech() {}
    override fun onRmsChanged(rmsdB: Float) {}
    override fun onBufferReceived(buffer: ByteArray?) {}
    override fun onEndOfSpeech() {}
    override fun onError(error: Int) {}
    override fun onResults(results: Bundle?) {
      val matches =
results?.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION)
      matches?.firstOrNull()?.let { result ->
        if (result.contains(command, ignoreCase = true)) {
          arrayUser.forEach { sendSMS(it) }
          speak("Sos Sent Successfully")
        }
      } ?: showToast("No speech recognized")
    }
```

```
    override fun onPartialResults(partialResults: Bundle?) {}
    override fun onEvent(eventType: Int, params: Bundle?) {}
  }

  override fun onDestroy() {
    tts.stop()
    tts.shutdown()
    speechRecognizer.destroy()
    super.onDestroy()
  }
}
```

APPENDIX-B

SCREENSHOTS

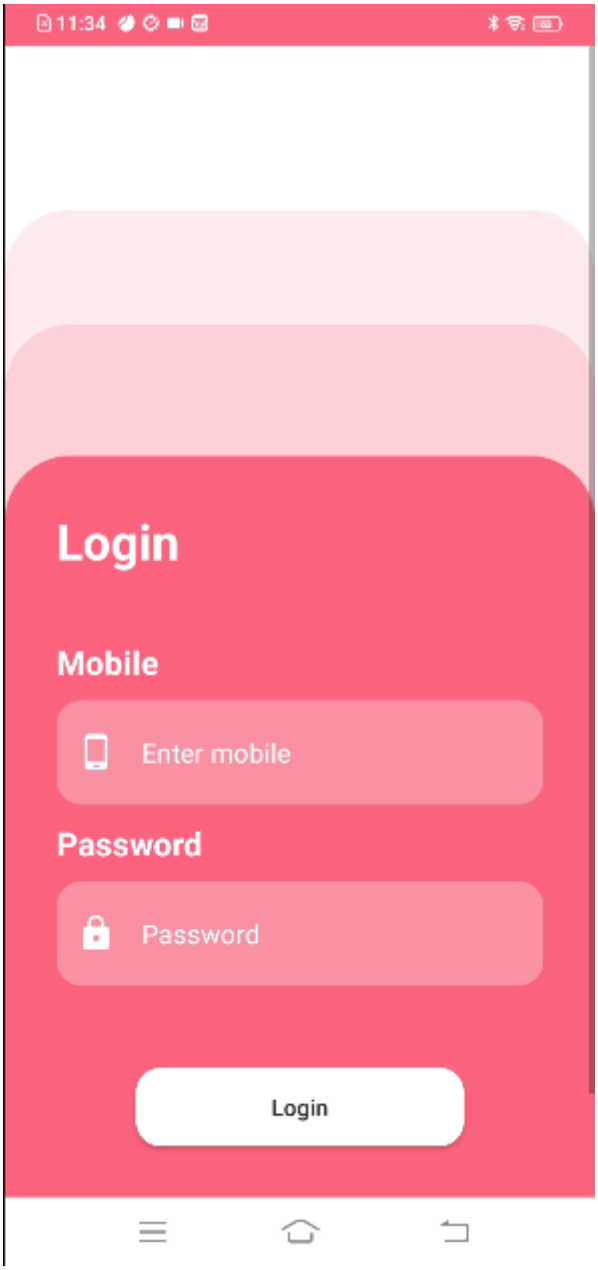

Figure Appendix B 1.1 Login page

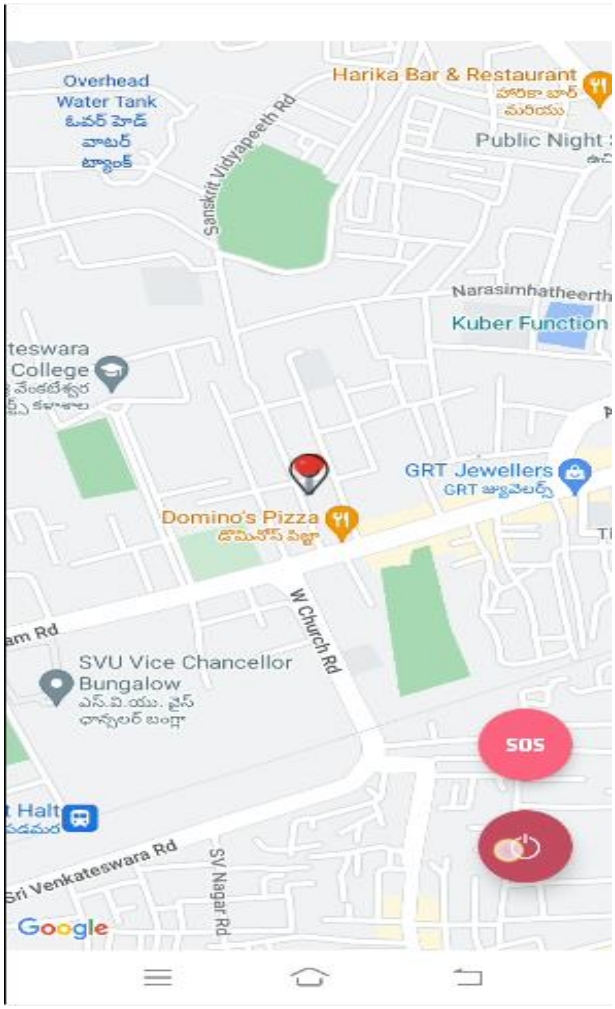

Figure Appendix B 1.2 Dashboard

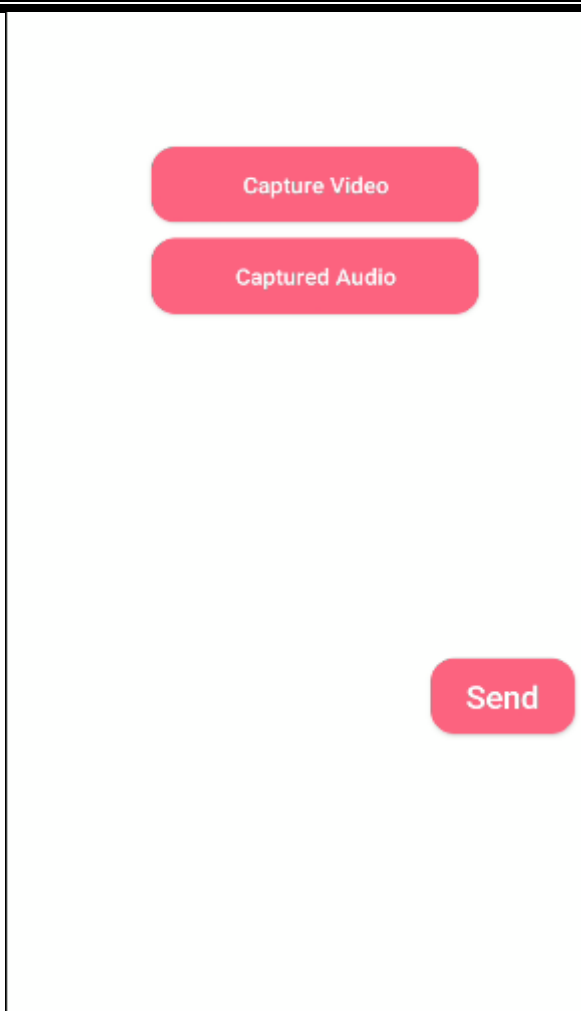

Figure Appendix B 1.3 Selecting Type

Figure Appendix B 1.4 Video Capture

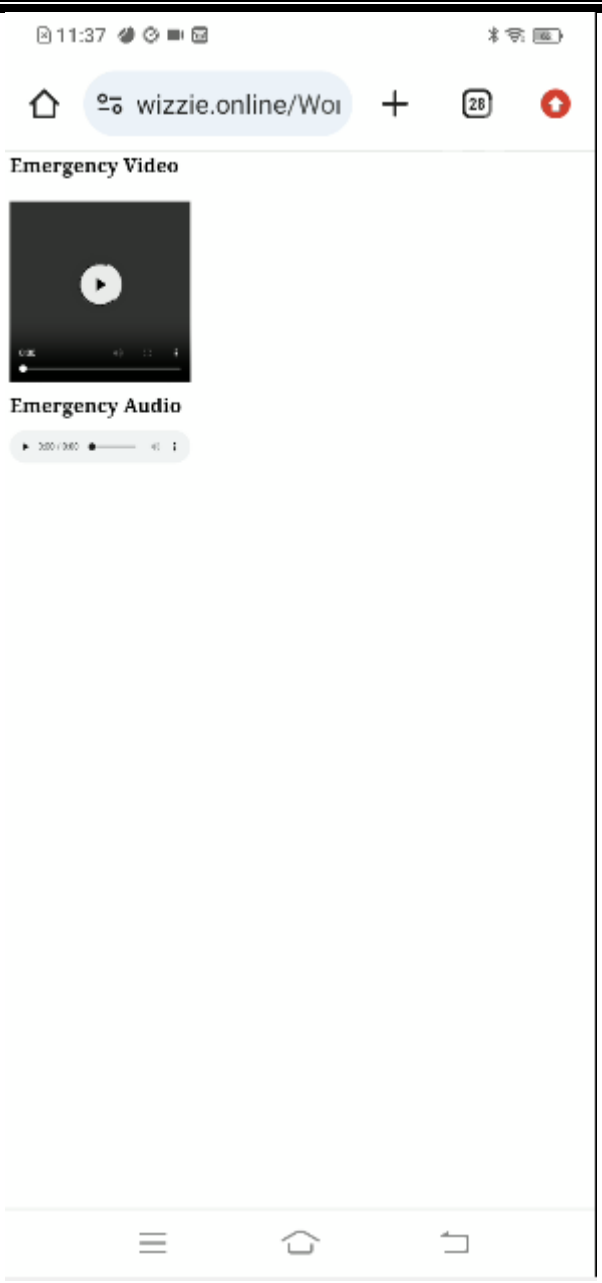

Figure Appendix B 1.5 Video and Audio Page

APPENDIX-C

ENCLOSURES

Sustainable Development Goals (SDGs)

Figure Appendix C 1.1 Sustainable Development Goals

SDG 5 – Gender Equality

The app helps protect women in colleges and universities by giving them quick access to emergency support. With features like SOS alerts, location sharing, and AI chatbot, it empowers women to feel safe and act. This supports equal rights and safety for all genders. It provides women with the tools they need to report incidents safely. The app promotes a sense of security and confidence on campuses. By addressing safety concerns, it encourages equal opportunities for all students. It helps reduce the fear of harassment, allowing women to focus on their education. The app works toward creating a safer, more inclusive environment for everyone.

SDG 9 – Industry, Innovation, and Infrastructure

This project uses modern tools like AI, real-time location tracking, and mobile technology to create a smart safety system. It shows how digital innovation can be used to improve public safety and support modern infrastructure in campuses. The app integrates advanced technology to improve emergency response times. It uses AI to detect patterns and predict unsafe areas on campus. The system supports the development of smart, tech-driven safety solutions.

It contributes to building a more connected and responsive infrastructure. The app can be expanded to other sectors, like public transportation or events. It showcases how innovation can transform traditional safety systems into more efficient ones.

SDG 10 – Reduced Inequalities

The app is designed to be simple and easy to use for everyone, including those with little technical knowledge. It helps ensure that all students—no matter their background—have equal access to safety tools, support, and reporting systems. The application works in most languages, so more people can use it. It bridges the gap for students in rural locations with offline reporting capabilities. By providing instant assistance, it guarantees that all individuals will receive assistance when necessary. The simple design of the app makes it usable by people of all skill levels. It ensures that all people can be safe on campus regardless of their technical expertise.

SDG 11 – Sustainable Cities and Communities

By improving safety and response times, the app helps make schools and colleges safer parts of the community. It connects users with emergency services and reduces panic during crises, supporting safer and more connected campuses. The app promotes a sense of security, making campuses more welcoming for everyone. It encourages students and staff to work together to create a safer environment. By providing quick access to safety resources, the app helps prevent larger crises. It supports the development of safer, smarter campus infrastructures. The app helps build a community where people look out for each other's well-being.

SDG 16 – Peace, Justice, and Strong Institutions

With built-in features for complaint registration and feedback, the app allows students to raise safety concerns directly. This promotes trust, fairness, and accountability between students and college authorities, encouraging better management and support. The app ensures that safety concerns are addressed quickly and fairly It fosters open communication between students and campus officials. The app helps create a system where justice is accessible to everyone.

SDG 17 – Partnerships for the Goals

The app can be improved further by partnering with police departments, college security, or NGOs. Working together with these groups can make the system stronger and reach more people, supporting shared goals of safety and equality. These partnerships can help provide faster response times during emergencies. Collaborating with local organizations can help expand the app's reach to more communities.

CERTIFICATES


DOI: 10.55041/IJSREM47977
IJSREM
e-Journal
ISSN: 2582-3930
Impact Factor: 8.586
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT
An Open Access Scholarly Journal || Index in major Databases & Metadata
CERTIFICATE OF PUBLICATION
International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to
Challa Omnath
in recognition to the publication of paper titled
Community Based Reporting and Monitoring Tool for Women Safety in Colleges/Universities
published in IJSREM Journal on Volume 09 Issue 05 May, 2025
www.ijsrem.com
Editor-in-Chief
IJSREM Journal
e-mail: editor@ijsrem.com