ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-2/19-27
Samiha Sarwar *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

Secure Her : A Voice Controlled Women Safety App

A Vasavi Sujatha[1],Rithisha Aritakula[2],Samiha Sarwar[3],Shaik Mahin[4]
[1]Associate Professor; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India
[2,3,4]B.Tech Students; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India
Mail Id; aritakularithisha18@gmail.com[2], samihasarwar1234@gmail.com[3], mahinshaik1206@gmail.com[4]
Accepted 06-04-2026

Abstract
Ensuring personal security, especially for women, has become an increasingly important concern in modern society. To address this challenge, this project proposes an intelligent women safety application designed to provide immediate assistance during emergency situations. The system integrates modern web technologies, real-time communication, and artificial intelligence to create a responsive and reliable safety platform. Users can activate emergency alerts either manually through an SOS button or through voice commands, ensuring accessibility even when physical interaction with the device is difficult.The application utilizes technologies such as Firebase Authentication, the Geolocation API, and Google Maps services to enable secure user login and precise location identification. When the SOS function is triggered, the system automatically captures the user's real-time geographical coordinates and generates a shareable map link. This information is instantly transmitted to pre-registered emergency contacts through an automated SMS service, enabling faster response and accurate location tracking during critical situations.To enhance usability, the system includes an AI-based voice recognition component capable of detecting emergency phrases like "HELP" or "SOS." This functionality allows the application to operate in a hands-free mode, which can be essential in threatening circumstances. Additionally, the platform incorporates a real-time monitoring interface supported by Folium that visually displays the user's movement on an interactive map, improving situational awareness for emergency responders and trusted contacts.By combining cloud authentication services, geospatial technologies, artificial intelligence, and automated communication tools, the proposed solution enhances emergency response efficiency and promotes proactive personal safety. The system demonstrates a scalable and user-friendly approach to modern safety applications, encouraging the integration of smart technologies into everyday personal security practices.

Keywords: *Women Safety, SOS Alerts, Voice Recognition, Real-Time Tracking, Geofencing, Crime Data Analysis, AI Safety Scoring, Danger Zone Detection, Safer Route Suggestions, Live Location Sharing, Emergency Contacts, Flutter, Firebase Authentication, Google Maps API, Twilio SMS Service.*

Introduction

In the modern world, personal security has become a significant concern, particularly for women, due to the rising number of crimes and unsafe situations in many regions. Women frequently encounter risks while commuting, working late hours, or visiting public spaces where immediate assistance may not always be available. In such circumstances, the ability to quickly alert others and share accurate location information becomes extremely important. Conventional methods of requesting help, such as phone calls or seeking assistance from nearby individuals, are often inefficient during emergencies because they require time, physical interaction, and the presence of supportive people nearby. These limitations highlight the need for an intelligent and easily accessible technological solution capable of providing rapid assistance.This research proposes "SecureHer: A Voice-Controlled Women Safety and Emergency Response System," a smart application designed to support users during emergency situations. The platform enables individuals to send immediate alerts, share their live location, and notify trusted contacts with minimal interaction. The system is implemented as a web-based application to ensure accessibility and ease of use across multiple devices without requiring extensive technical knowledge from users.A major feature of the proposed system is the SOS alert mechanism, which allows emergency alerts to be activated either manually through a panic button or automatically through voice commands. In situations where the user cannot physically access the device, predefined voice keywords such as "HELP" or "SOS" can trigger the alert mechanism. This hands-free

capability makes the system particularly valuable during high-risk situations where manual operation may not be possible. Once activated, the system automatically captures the user's real-time geographic location and prepares an emergency notification.The application utilizes several modern technologies to ensure secure and accurate functionality. Firebase Authentication is employed for user verification and secure data management, while the Geolocation API is used to capture precise real-time location information. In addition, Google Maps services are integrated to generate location links that allow emergency contacts to easily navigate to the user's position. The combination of these technologies ensures reliability, accuracy, and security within the system.Beyond emergency alerts and tracking, the system also includes an automated communication feature that sends notifications to pre-registered emergency contacts. The alert message contains essential information such as the user's emergency status and a direct map link showing their exact location. This automated communication reduces the need for manual typing or multiple steps, which can be difficult during stressful situations.To further enhance safety awareness, the system incorporates a community-based incident reporting feature. Through this functionality, users can report unsafe locations or suspicious activities, helping other individuals avoid potentially dangerous areas. This collective reporting mechanism encourages community participation and contributes to building a safer environment for everyone.

Purpose of the Project

The primary objective of this project is to develop a smart and dependable women safety system capable of providing immediate support during emergency situations. The project aims to leverage modern technologies to reduce response time and ensure that assistance can be delivered quickly and effectively.The system is designed to allow users to send emergency alerts using both manual activation and voice-based commands. By integrating features such as SOS alerts, live location sharing, and automated communication with emergency contacts, the application enables users to quickly inform trusted individuals about their situation.Overall, the project aims to enhance personal security, improve emergency response efficiency, and provide a practical technology-based solution that can help protect individuals and potentially save lives.

Existing System

Current women safety systems primarily rely on traditional methods such as emergency phone calls, basic mobile applications, or manual alert mechanisms. While several safety applications provide functionalities such as sending SMS notifications or sharing location information, many of them require multiple user interactions to activate these features.In most cases, users must unlock their device, open the application, and manually trigger an alert. During emergency situations, this process can be difficult or even impossible, especially if the user is under stress or unable to access the device quickly. As a result, delays in sending alerts may reduce the chances of receiving timely assistance.Therefore, there is a clear need for a more advanced system that can provide faster response, improved accessibility, intelligent features, and enhanced usability during emergency situations.

.

Related Work

Survey

Women's safety has become a critical issue in modern society, encouraging researchers and developers to design technological solutions that can provide immediate assistance during emergency situations. Various safety systems have been proposed that integrate mobile applications, GPS-based tracking, GSM communication, and artificial intelligence to enhance personal security. These systems aim to reduce response time and improve communication between individuals in distress and their trusted contacts or emergency responders.Several safety applications rely on a panic-button mechanism that enables users to send emergency notifications to pre-registered contacts. When activated, these systems typically transmit an SOS message along with the user's location using GPS technology. This approach helps emergency contacts quickly identify the user's position. However, such systems depend heavily on manual interaction, which may not always be possible during dangerous or stressful situations.Other studies have explored the use of combined Global Positioning System (GPS) and Global System for Mobile Communication (GSM) technologies to ensure reliable communication during emergencies. In these systems, when an alert is triggered, the device sends location coordinates to multiple contacts through SMS messages. This approach ensures that alerts can still be delivered even in areas where internet connectivity is weak or unavailable. Although this method improves communication reliability, most of these systems lack advanced automation or intelligent decision-making capabilities.More recent applications have introduced live location tracking and continuous monitoring features that allow emergency contacts to track the user's movement in real time. Mapping technologies enable the visualization of location data on interactive maps, making it easier for responders to identify the exact position of the individual requiring assistance. Despite these advantages, continuous tracking may

consume additional device resources and often requires stable network connectivity.Some safety platforms also emphasize preventive safety measures by providing information about nearby police stations, hospitals, or safe locations. These features help users take precautionary actions before a risky situation occurs. Although these tools improve situational awareness, they cannot fully replace an effective emergency response system.Recent research has also explored the integration of voice recognition technologies into safety applications. Voice-based activation enables users to trigger emergency alerts using spoken commands, eliminating the need for manual device interaction. Speech-to-text technologies can recognize keywords such as "help" or "SOS" and initiate emergency protocols automatically. However, such systems may face challenges in noisy environments Despite the progress made by existing solutions, several limitations remain. Many applications rely heavily on manual activation, which may not be practical during emergencies. Some systems also lack proper integration between essential functionalities such as communication, location tracking, and automation. In addition, issues such as delayed notifications, complex user interfaces, and reliance on internet connectivity can reduce the overall effectiveness of these applications.To address these challenges, the proposed system integrates voice-based emergency activation, real-time location tracking, automated SMS alerts, and community incident reporting into a single platform. By minimizing reliance on manual interaction and improving response speed, the system aims to provide a more efficient and user-friendly safety solution. Overall, the review of existing technologies indicates that while several safety applications exist, there is still a need for a more integrated and intelligent system capable of delivering faster and more reliable emergency assistance.

Requirement Analysis

Requirement analysis plays an important role in defining the functionalities and performance expectations of the proposed Women Safety Application. Based on the evaluation of existing systems and their limitations, the requirements for the SecureHer system have been identified. These requirements describe both the functional capabilities and the operational characteristics necessary for the system to function effectively.

Functional Requirements

Functional requirements define the essential operations that the system must perform in order to provide effective safety support. The application should allow users to register and log in securely through an authentication mechanism that ensures only authorized users can access personal data such as emergency contacts and location information. The system must include an SOS alert mechanism that can be activated either manually through a panic button or through voice commands. Once triggered, the alert process should begin immediately so that emergency contacts can be notified without delay.The application must also support real-time location sharing by capturing the user's geographic coordinates through GPS services. The system should generate a shareable map link that allows emergency contacts to locate the user quickly. Additionally, the application should provide continuous location tracking through an interactive map interface, enabling contacts to monitor the user's movement during emergencies.Another important requirement is voice command support, where the system detects predefined keywords such as "HELP" or "SOS" and automatically activates the emergency alert process. The application should also support automated communication by sending SMS notifications containing the user's location details to registered emergency contacts. Furthermore, users should be able to manage their emergency contacts by adding, updating, or removing contact information as needed.The system should also include a community incident reporting feature that enables users to report unsafe areas or suspicious activities. This functionality can help raise awareness among users and improve overall community safety.

Non-Functional Requirements

Non-functional requirements define the performance characteristics and quality attributes of the system. The application should be scalable so that it can handle a growing number of users without affecting performance. It should also provide fast system response, particularly during emergency situations when quick alert generation and location sharing are critical.Usability is another important requirement, meaning that the interface should be simple and intuitive so that users of different age groups and technical backgrounds can operate it easily. The system must also be reliable, ensuring that emergency features function consistently whenever they are needed.Security is essential for protecting sensitive information such as user location data and emergency contact details. Therefore, the system should implement secure authentication and data protection mechanisms. Additionally, the application should maintain high availability so that it remains accessible during emergency situations.Maintainability and extensibility are also important considerations. The system architecture should allow developers to update or improve the application easily without affecting existing functionalities. It should also support future enhancements such as AI-based threat detection or integration with wearable safety devices.

Computational Resource Requirements

The proposed system requires both software and hardware resources to operate efficiently. From the software perspective, the application can run on operating systems such as Windows, macOS, or Linux. Python is used as the primary programming language for implementing the system, and development can be performed using integrated development environments such as Visual Studio Code or PyCharm. The frontend interface is built using the Streamlit framework, which provides a web-based user interface, while backend operations are handled using Python-based frameworks.The system also integrates artificial intelligence technologies such as Natural Language Processing for voice recognition functionality. Frameworks like OpenAI API and LangChain may be used to support advanced AI features. Databases such as ChromaDB, JSON, or SQLite are used to store user data and application information. Additional libraries including Pandas, NumPy, and Matplotlib may be used for data processing and visualization. Users can access the application through modern web browsers such as Google Chrome or Mozilla Firefox, and version control is maintained using Git and GitHub.From the hardware perspective, the system requires a computer with a quad-core processor operating at 3.0 GHz or higher, a minimum of 8 GB RAM, and approximately 1 TB of storage space using either HDD or SSD

System Design

System Architecture

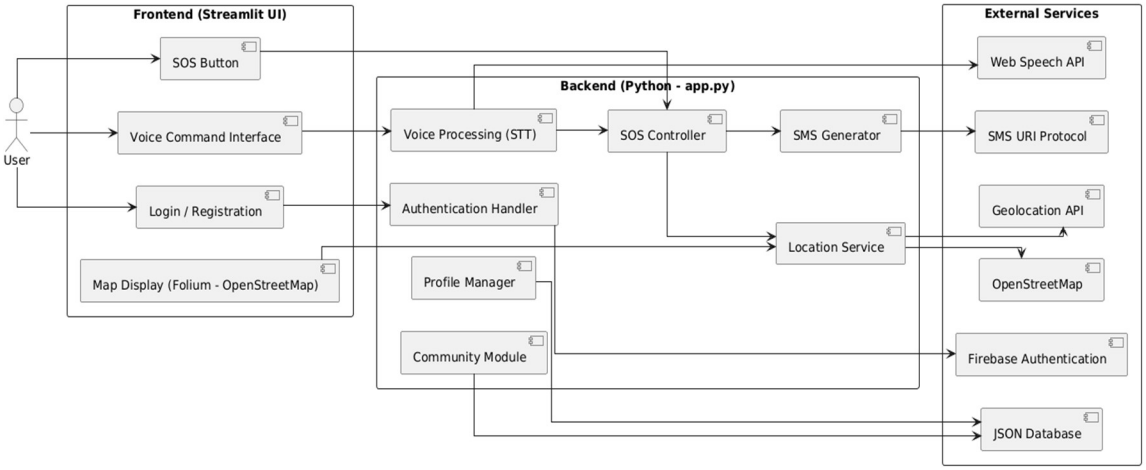


Fig. 1 System Architecture

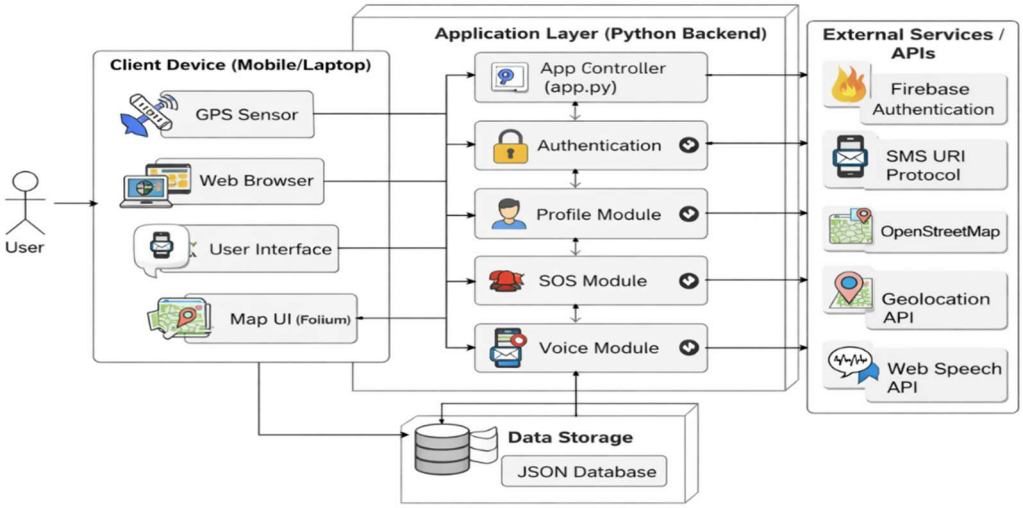


Fig. 2 Technical Architecture

The system architecture illustrates how different components of the Women Safety Application interact to provide emergency assistance. In the proposed architecture, users interact with the application through a web-based interface developed using Streamlit. Through this interface, users can perform various actions such as registering, logging in, activating SOS alerts, issuing voice commands, and

viewing their current location on a map.User requests are processed by a Python-based backend application responsible for handling authentication, voice command processing, profile management, and emergency alert operations. The SOS controller serves as the central component that coordinates emergency response activities. When an SOS alert is triggered, the system retrieves the user's real-time location using the Geolocation API and generates a map link that can be shared with emergency contacts.The backend system communicates with external services such as Firebase Authentication for secure login management, Web Speech API for voice recognition functionality, and mapping services for location visualization. These integrated components work together to ensure efficient communication, accurate location tracking, and rapid emergency response.

UML Diagrams

Unified Modeling Language (UML) diagrams are used to visually represent the structure and behavior of the system. These diagrams help developers understand system interactions and workflow processes.The Use Case Diagram illustrates how users interact with the application and identifies key system functionalities such as user authentication, SOS alert activation, location sharing, and incident reporting. The Class Diagram represents the static structure of the system by defining classes, attributes, methods, and relationships that support application functionality. The Sequence Diagram shows the order of communication between system components during operations such as emergency alert generation. The Activity Diagram models the workflow of the system by illustrating the sequence of activities involved in processing emergency alerts and location updates.

System Modules

To improve organization and maintainability, the application is divided into multiple modules, each responsible for a specific functionality. The User Registration and Login Module manages authentication and ensures secure access to the system through Firebase Authentication services. The SOS Alert Activation Module provides the core emergency feature that allows users to trigger alerts through either manual interaction or voice commands.The Live Location Sharing Module retrieves the user's geographic coordinates and generates a shareable map link that can be sent to emergency contacts. The Real-Time Location Tracking Module continuously updates the user's position on an interactive map interface, enabling accurate monitoring during emergencies.The Voice-Controlled Emergency Support Module uses speech recognition technology to detect emergency keywords such as "HELP" or "SOS" and automatically trigger emergency alerts. The Emergency Communication Module sends SMS notifications containing the user's location and alert message to trusted contacts.

Implementation

Libraries and Technologies

The SecureHer system is implemented using a combination of frontend, backend, cloud, and artificial intelligence technologies. Firebase Authentication is used to provide secure login and registration functionality, ensuring that only authorized users can access the system. Firebase Firestore or Realtime Database is used to store user data, emergency contacts, and alert information in the cloud.The Geolocation API is used to retrieve the user's real-time geographic coordinates, enabling accurate location tracking. Google Maps API is integrated to display maps and generate location links that help emergency contacts navigate to the user's position.The SpeechRecognition library enables the system to detect voice commands and convert them into text, allowing voice-based SOS activation. Flask is used as the backend framework for handling server-side operations and integrating various APIs. The Twilio SMS API is used to send automated emergency notifications to registered contacts.Additionally, the Folium library is used to generate interactive maps that visually display the user's location, while JSON is used as a lightweight format for storing configuration data and user information.

Pseudocode

The system operation begins with user authentication. After successfully logging into the application, the system continuously monitors the user's device location and displays it on an interactive map interface. The application actively listens for emergency triggers such as manual panic button activation or predefined voice commands.

When an SOS alert is detected, the system retrieves the user's current geographic coordinates and generates a Google Maps link representing the location. An emergency message containing this link is then automatically prepared and sent to the user's registered emergency contacts through SMS communication. In addition to emergency alerts, the system also allows users to submit incident reports about unsafe locations, which are shared within the community to improve safety awareness.

Testing and Validation

Test Cases

Testing plays an essential role in ensuring the correctness, reliability, and efficiency of the SecureHer Voice-Controlled Women Safety Application. A structured set of test cases was developed to verify whether individual modules and the integrated system operate as expected under various conditions. The testing process primarily focused on critical system functionalities such as user

authentication, SOS alert activation, voice command detection, and real-time location sharing.The objective of these test cases was to confirm that the system performs accurately in both normal usage scenarios and emergency situations. Each module was evaluated independently before performing integrated system testing to ensure that the complete application delivers reliable emergency assistance.

Test Strategy

The testing strategy defines the overall methodology used to evaluate the system's functionality and stability. The approach includes validating both standalone modules and the fully integrated system to ensure that all components interact correctly.The strategy emphasizes testing important system features such as user registration, secure login, SOS alert activation, speech-based emergency commands, real-time location tracking, and SMS alert generation. Both manual testing and automated testing techniques were employed to verify system performance and reliability.Testing was conducted under normal operational conditions as well as simulated emergency scenarios to confirm that the system responds promptly when an SOS alert is triggered. Particular attention was given to real-time performance, communication reliability, and data security to ensure that the application functions effectively during critical situations.

Testing Approach

A hybrid testing approach combining manual evaluation and automated testing tools was adopted to ensure comprehensive system validation.

i. Test Environment Setup

The testing environment included a local development setup for backend operations and multiple devices for user interface evaluation. The following components were used during testing:

Local development environment for testing backend services built using Flask

Mobile and web browsers for validating user interface interactions

Firebase sandbox environment for authentication and database validation

Mock datasets to simulate emergency contacts, location coordinates, and SOS events

This setup allowed the system to be tested in a controlled environment before deployment.

Test Case Design

Different testing techniques were applied to evaluate the application thoroughly:

Black-boxtesting:

Used to evaluate system functionality from the user perspective. This method validated outputs such as SOS alerts, SMS notifications, and location sharing without analyzing internal code structure.

White-boxtesting:

Used to analyze internal program logic including voice command processing, API interactions, and backend workflow execution.

Exploratory testing:

Performed to examine unexpected system behaviors during emergency conditions and to identify usability issues.

Automation Tools

Automated tools were employed to improve testing efficiency and ensure code quality.

pytest / unittest: Used for testing backend logic and API responses

Pylint / Flake8: Used for static code analysis and enforcing coding standards

Firebase testing tools: Used to validate authentication workflows and database operations

These tools helped identify potential errors and ensured consistent system behavior.

Bug Reporting and Tracking

All issues identified during testing were carefully documented and tracked. A structured testing log was maintained to record executed test cases, detected errors, and implemented fixes. Bug tracking was managed through tools such as GitHub Issues or manual tracking systems to ensure that each reported issue was resolved systematically before deployment.

Unit Testing

Unit testing focuses on verifying the functionality of individual components within the application. Each module was tested independently to ensure that internal logic produced correct outputs for given inputs. Unit testing was performed after completing the implementation of each module but before integrating it with other system components.This testing approach ensured that business logic, configuration settings, and data processing operations functioned correctly according to system specifications.

Integration Testing

Integration testing evaluates whether different modules operate correctly when combined into a single system. After completing unit testing, modules such as authentication, location tracking, voice processing, and SMS communication were integrated and tested together.The primary goal of integration testing was to identify errors caused by interactions between different components and to ensure that data flows correctly across modules.

Functional Testing

Functional testing verifies whether the system performs operations according to defined requirements. This testing focused on validating key application functions, including:

Acceptance of valid user inputs

Proper rejection of invalid inputs

Correct execution of system features
Accurate generation of system outputs
Proper interaction with external services
Functional testing ensured that the application behaves as expected in accordance with its functional specifications.

System Testing

System testing evaluates the entire integrated application in an environment that closely resembles real-world conditions. The objective of system testing is to confirm that all modules function together as a unified system.This testing phase examined overall application workflow, external service integration, data handling, and user interactions. Multiple input scenarios—including valid, boundary, and invalid cases—were tested to ensure consistent system performance.

White Box Testing

White box testing examines the internal structure and logic of the application. Test cases were developed based on the program code to evaluate decision branches, loops, and control flow paths.This testing method helped identify logical errors, potential vulnerabilities, and inefficient code segments that might not be visible through external testing methods.

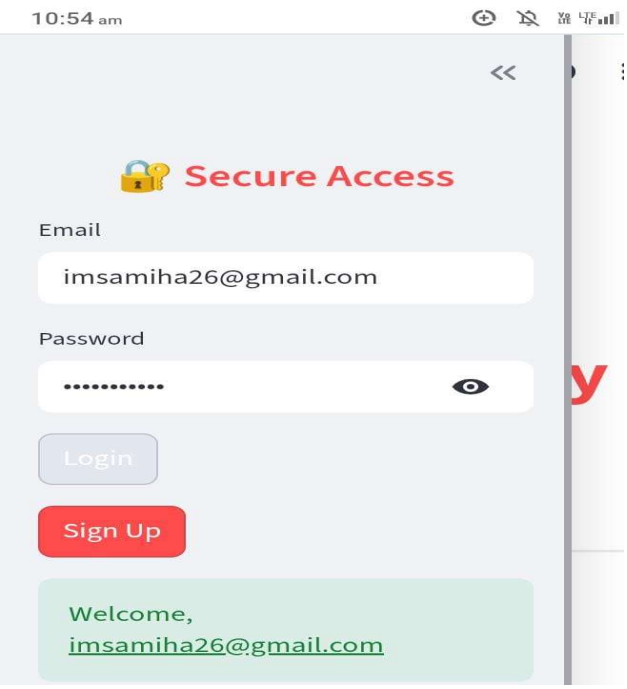
Black Box Testing

Black box testing evaluates the application from the user perspective without considering the internal implementation. In this method, the system is treated as a "black box," where inputs are provided and outputs are analyzed.This approach was particularly useful for validating user interface behavior, system responses, and overall usability.

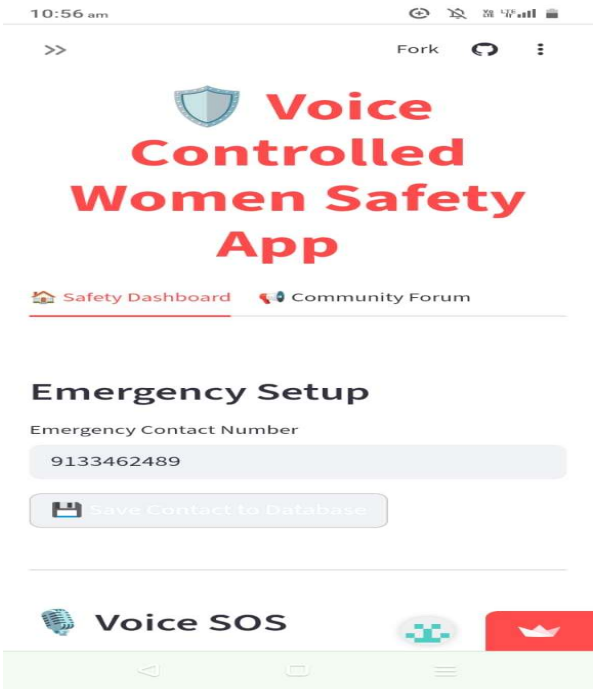
Result Analysis

The testing results indicate that the SecureHer Voice-Controlled Women Safety Application performs effectively across all core functionalities. The system demonstrated reliable operation in handling emergency alerts, voice command recognition, and real-time location sharing.The authentication module successfully ensured secure login processes and prevented unauthorized access to sensitive user information. User data such as emergency contacts and profiles were correctly stored and retrieved from the cloud database.Both manual and voice-based SOS activation mechanisms were tested under multiple scenarios. The system was able to detect predefined keywords such as "SOS" and "HELP" with high accuracy and initiate emergency actions promptly.he location tracking module accurately captured geographic coordinates and continuously updated the user's position. Emergency contacts received Google Maps location links correctly, enabling them to quickly determine the user's location during emergencies.

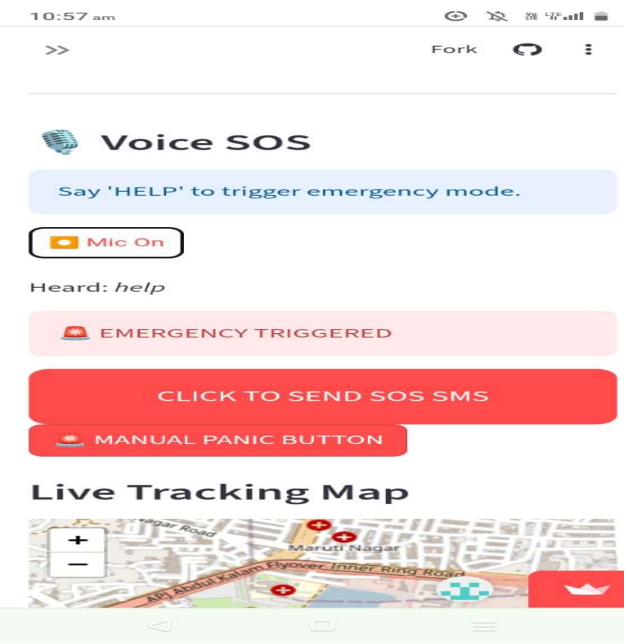
Screenshots



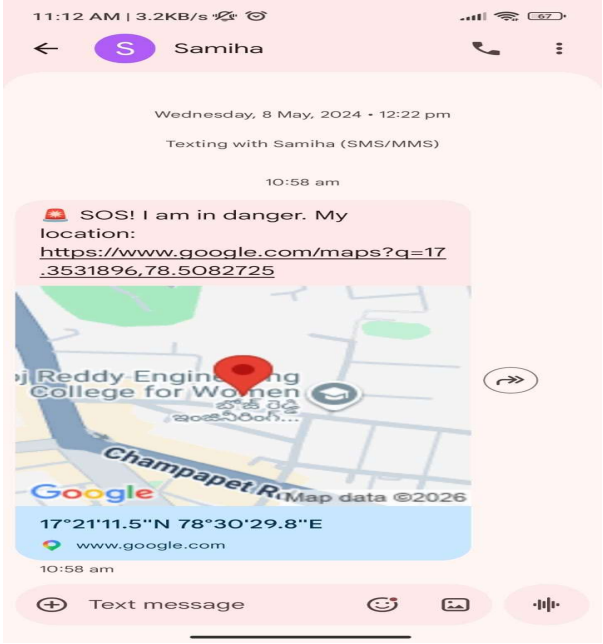
Screenshot 1: User Authentication Success



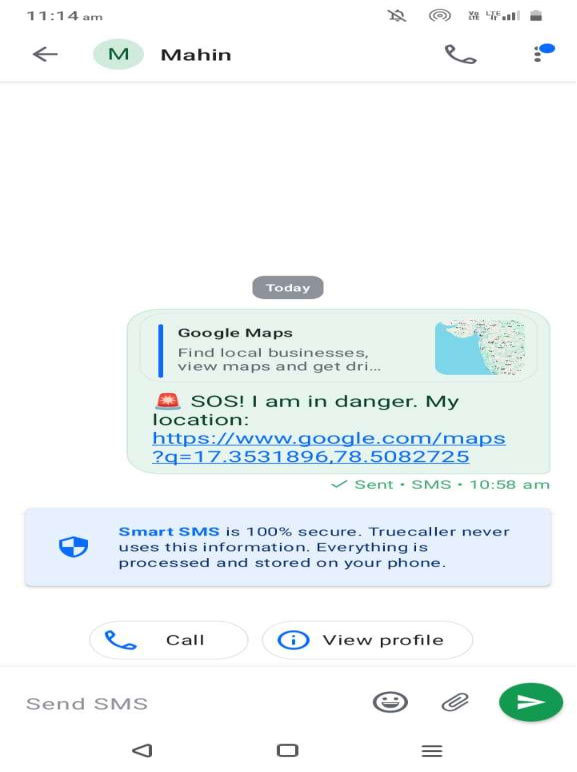
Screenshot 2: Emergency Number Entered



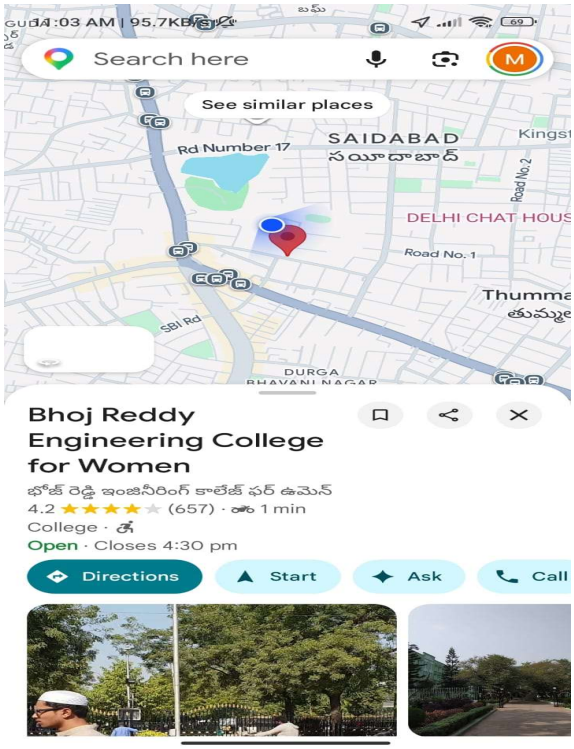
Screenshot 3: Emergency Triggered



Screenshot 5: SOS SMS Received



Screenshot 4: SOS SMS Sent



Screenshot 6: Live location

Conclusion

The SecureHer Voice-Controlled Women Safety Application successfully achieves its goal of providing an intelligent and dependable personal safety solution through the use of modern technologies. The system combines voice recognition, real-time location tracking, cloud-based data management, and automated emergency communication to support users during critical situations.One of the key advantages of the system is its ability to activate SOS alerts using both manual input and voice commands. This feature ensures that emergency assistance can be requested even when the user cannot physically interact with the device. Additionally, the instant sharing of live location information significantly improves response time during emergencies.The integration of technologies such as Firebase authentication, Geolocation APIs, Google Maps services, and Flask-based backend processing ensures a secure, efficient, and scalable system architecture.Overall, SecureHer demonstrates strong performance in terms of responsiveness, reliability, and user accessibility. The system can serve as an effective safety tool for real-world deployment and contributes toward improving personal security through technology-driven solutions.

Future Scope

Although the current system provides a reliable safety platform, several opportunities exist for future enhancements.One potential improvement is the integration of advanced artificial intelligence models capable of predicting potential threats by analyzing user behavior patterns and environmental factors. Machine learning algorithms could also improve the accuracy of voice recognition in noisy environments.Another enhancement involves integrating wearable devices such as smartwatches or IoT-based safety gadgets. This would enable users to trigger emergency alerts directly from wearable devices without relying on smartphones.The system could also incorporate real-time crime data analysis and geofencing techniques to identify high-risk areas and warn users before entering potentially unsafe locations. Additionally, AI-based route analysis could provide safer navigation suggestions.Future development may include integration with government emergency services, police departments, and healthcare facilities to ensure faster emergency response. Multi-language voice recognition support could also improve accessibility for users from diverse linguistic backgrounds.With these improvements, SecureHer can evolve into a more intelligent and comprehensive personal safety ecosystem.

References

[1] M. Villalba, “Artificial Intelligence and Natural Language Processing Applied to Design,” Proc. Doctoral Symposium on Natural Language Processing, Valladolid, Spain, 2024.
[2] X. A. Chen, T. Knearem, and Y. Li, “Exploring Generative UI Tools for UX Design Applications,” arXiv preprint arXiv:2501.13145, 2025.
[3] D. D. Patil et al., “Transformative Trends in Generative AI for Natural Language Understanding,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 4, pp. 309–319, 2023.
[4] R. Luera et al., “User Interface Design and Interaction Techniques in Generative AI Applications: A Survey,” arXiv preprint arXiv:2410.22370, 2024.
[5] D. R. Manchikanti, “AI-Powered Content Management Systems for Dynamic Web Applications,” ResearchGate Publication, 2025.
[6] S. V. Sheta, “Designing AI-driven Personalized User Interfaces for Enhanced User Experience,” ResearchGate Publication, 2025.
[7] S. V. Sheta, “A Systematic Review on Automatic Website Generation Technologies,” ScienceDirect, 2023.
[8] M. Villalba, “Applications of Artificial Intelligence in Design and Interaction Systems,” NLP-DS Conference Proceedings, 2024.
[9] H. Vandierendonck, B. Fraguela, and A. De Lorenzo, “Accelerating Dynamic Web Content Generation with Processor Support for PHP,” *IEEE ISPASS*, 2023.
[10] G. S. Gawande, A. R. Mote, and S. G. Deshmukh, “Automated Web Application Interface Generation Using Large Language Models,” *International Journal of Research Publication and Reviews*, vol. 5, no. 3, pp. 1186–1191, 2024.