International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

SentriXSOS: An AI-Assisted SOS and Emergency Support Companion

Snehal Dhananjay Jogdand

PROF. RAMKRISHNA MORE ARTS, COMMERCE & SCIENCE COLLEGE

Pradhikaran, Akurdi, Pune - 411044 (Maharashtra) India.

Email-snehalj2138@gmail.com

Dr. Kalyan C. Jagdale

PROF. RAMKRISHNA MORE ARTS, COMMERCE & SCIENCE COLLEGE

Pradhikaran, Akurdi, Pune - 411044 (Maharashtra) India.
kalyan.jagdale7@gmail.com

1. Abstract

In an emergency, people usually have only a few seconds to ask for help, and in that short time they are often scared, confused, or alone. Many SOS applications available today only send a basic "help me" message with GPS location to a few contacts, which is sometimes not enough for family members or police to really understand what is going on. This paper presents **SentriXSOS**, a web-based SOS companion that tries to reduce this communication gap. SentriXSOS allows users to save their safety profile and guardian details in advance, detects the current location and address from the browser, estimates a simple risk level from the situation description, and prepares a clear SOS message for guardians.

A small AI support chatbot is also included to answer common questions related to personal safety and basic first-aid, with voice input and output options. The prototype is mainly built using standard web technologies, and initial scenario testing with students shows that structured messages and live address information are easier to act on than traditional one-line alerts.

Keywords

Emergency alert, SOS system, web application, women safety, location sharing, AI chatbot, risk level

2. Introduction

2.1 Background and Motivation

Safety has become a daily concern for many groups of people in India, especially women travelling at night, college students returning from classes, and senior citizens living alone. News reports often show incidents where help arrived late because information about the location or the situation was incomplete or confusing. At the same time, almost everyone now carries a smartphone and has internet access, which means a browser-based application can reach a large population without requiring a separate app store installation.

During discussions with classmates and friends, it became clear that most of them know about at least one SOS app, but very few actually use it regularly or keep it configured. The main reasons they mentioned were: boring design, lack of trust that the app will really work when needed, and limited options to describe what is happening. This motivated the design of SentriXSOS as a slightly more "smart" and user-friendly alternative that still remains technically simple enough to implement as a student project.

2.2 Problem Statement

Most existing SOS mobile applications suffer from some common limitations:

- They treat every user in the same way and do not adapt to different profiles such as woman, child, elder, or general adult.
- The outgoing alert message is usually a fixed sentence with coordinates, without context about what exactly went wrong or how serious it appears.
- There is no built-in guidance before or after pressing the SOS button, for example for first-aid steps, helpline information, or legal awareness.
- Voice-based input and output are either missing or very limited, even though they are very useful when the user cannot type comfortably.

Because of these gaps, guardians receive very little information and often have to call the victim back to understand the details, which can waste precious time in the early minutes of an emergency.

2.3 Research Objectives

The main objectives of the SentriXSOS project are:

1. To design a browser-based SOS interface that lets users prepare their safety profile and guardian contacts in advance and re-use this data automatically during emergencies.
2. To build a simple risk-level estimation logic using everyday keywords from the incident description and show this clearly in the user interface.
3. To attach both raw coordinates and a human-readable address to the SOS alert by combining the geolocation API with a reverse-geocoding service.
4. To integrate a small AI support chatbot that can answer basic safety and first-aid questions using text as well as voice input and output.
5. To check, through scenario-based testing with students, whether such structured alerts are considered more useful and actionable than plain “help me” messages.

3. Methodology and Technology

3.1 Overall Architecture

SentriXSOS follows a straightforward three-layer design:

- **Presentation layer:** HTML, CSS, and JavaScript make up the user interface. The page is divided into panels for profile management, SOS controls, and alert history, with a floating button that stays visible on every scroll position.
- **Browser logic layer:** JavaScript handles local data storage, location access, risk classification, message generation, and the chatbot conversation.
- **Backend layer (future extension):** In the current prototype, alerts stay within the browser. In the planned version, a Node.js/Express backend will receive the JSON alert payload and then send SMS, WhatsApp, or email notifications using third-party APIs.

Even without the backend, the front-end prototype is useful for demonstrating interaction flow, documenting the idea, and collecting early user feedback.

3.2 Safety Profile Management

When SentriXSOS opens for the first time, the user is asked to pick a profile: **Woman**, **Child**, **Elder**, or **Adult**. This choice does not yet change the algorithm in a complex way but helps to label the alert properly and allows future profile-specific rules.

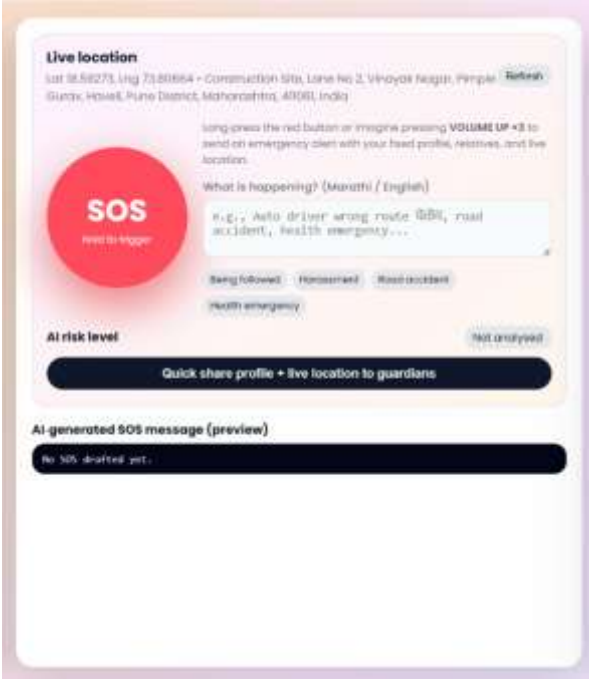
The user then fills in basic information:

- Name
- Optional medical information (for example “A+, asthma, BP medicine”)
- Primary and secondary guardian phone numbers
- Guardian email address
- Which groups to notify (Family, Friends, Police/Helpline)

This information is stored in the browser using localStorage in JSON format. The next time the user opens SentriXSOS, the page automatically loads and displays the last saved profile, so there is no need to re-enter data. This reduces friction and makes it more

realistic that the system will still be configured when needed in real life.

3.3 Location Capture and Address Resolution



To make the alert actually useful, SentriXSOS needs not only coordinates but also a readable address. For this purpose, two steps are performed:

1. The HTML5 Geolocation API asks for permission and, when allowed, returns latitude and longitude.
2. These coordinates are sent to a reverse-geocoding service such as OpenStreetMap Nominatim, which responds with a formatted address line containing street name, locality, and city.

The interface shows both:

- "Lat 17.75135, Lng 75.96955"
- "Shivaji Nagar, Solapur, Maharashtra, India" (for example)

The last retrieved location is stored in a temporary variable so that quick SOS actions can use it immediately without another geolocation call if the user presses the button again within a short time.

3.4 Risk-Level Estimation

Instead of asking the user to manually pick a risk category, SentriXSOS reads the incident description and applies straightforward keyword-based rules. If the text contains words linked to heavy injury or serious health problems such as "blood", "unconscious", or "chest pain", the risk is set to **Critical**. If the text mentions "harassment", "being followed", or "wrong route", the risk is set to **High**. In all other cases it remains at **Medium** by default.

The selected risk is then displayed in a pill component using different background colors. Even though this logic is not a full AI model, it still offers a clear visual hint to guardians and testers and can be gradually replaced by a trained model in later work.

3.5 SOS Message Generation and Alert Logging

The central SOS button reacts to a long-press gesture as a simple protection against accidental taps. The floating ⚡ button also triggers a quick SOS and simultaneously opens the chat popup. When a trigger is detected, the system performs the following steps:

1. Ensures that the latest profile and location data are available.
2. Computes the current risk level from the description.
3. Assembles a multi-line message that includes:
   - User name and profile type.
   - Location with address and coordinates.
   - Description of what is happening.
   - Assessed risk.
   - A short instruction asking guardians to call back or reach the location if possible.
4. Displays this message in the "AI-generated SOS message" preview box.
5. Inserts a compact summary into the **Recent alerts** list with time stamp and risk tag for quick scanning.

In a deployed version, the same JSON object that is logged in the UI can be posted to the backend to be forwarded via SMS or other channels.

3.6 AI Support Chatbot with Voice Features

The SentriXSOS chatbot lives in a side popup that appears near the bottom-right of the screen. It is opened and closed through the floating button or the close icon in the popup header. The design goals for this component were:

- Keep the interface minimal but friendly, with one scrollable area for messages and a simple input row at the bottom.
- Allow users to either type their doubts or speak using a microphone icon.
- Read out the responses so that the user does not always have to look at the screen.

Technically, the chatbot uses:

- SpeechRecognition (where available) to convert speech into text; if the browser does not support it, the microphone button is disabled.
- speechSynthesis to speak back the generated answer in an Indian English voice.
- Simple keyword matching on the text to decide which canned response to return, for example advice about bleeding, basic self-defence tips, or how to contact the national emergency number.

Although this is not yet a full LLM-based chatbot, it already shows how conversational help can be built into a safety tool without requiring a heavy server setup.

---

4. Results and Discussion

4.1 Scenario-Based Testing

To understand how SentriXSOS behaves in realistic situations, a few test cases were carried out with classmates who played different roles (woman travelling alone, school-going child, elder with health issues). Each participant configured their profile once and then triggered SOS for different descriptions, such as:

- "Auto driver taking wrong route, feeling unsafe."
- "Road accident, injuries, health emergency, chest pain."
- "Unknown person following me near hostel gate."

In the harassment and stalking cases, the system correctly classified the risk as **High**, while the accident-with-chest-pain scenario was flagged as **Critical**. The generated messages, when read by another student acting as a guardian, were generally rated as clear and informative, mainly because of the explicit mention of what was happening and the ready-to-use address line.

4.2 User Perception and Usability

Informal feedback sessions were held after the trials. Several patterns appeared:

- Students liked the idea of saving guardian numbers and medical information only once, instead of typing them every time.
- The live location block helped them feel that the app was actually using the phone's capabilities and not just sending a text.
- Some participants suggested that, in a real product, SOS should also start an audio recording or capture a short video clip for evidence, which is a potential extension.
- The AI assistant was mainly used for quick first-aid clarifications; a few users asked for clearer answers in simpler language, which shows that answer style matters as much as content.

These comments suggest that the overall direction of SentriXSOS is aligned with user expectations, but the depth and language of the chatbot content would need more refinement before public deployment.

4.2 Limitations

The current prototype has some known limitations:

- It depends on browser permission for location and microphone access; if the user denies these, functionality becomes limited.
- All "intelligence" is rule-based and runs on the client side. No training data or statistical models are involved yet, so risk classification can miss unusual phrases.

- There is no encryption of stored profile data in localStorage, which must be addressed before handling real personal information.
- Network calls for reverse-geocoding and, in the future, message delivery, make the system dependent on stable internet connectivity.

4.3 Safety and Privacy Considerations

While SentriXSOS stores profile data locally and does not yet transmit real alerts to servers, a production deployment must address:

- Secure storage and encryption of personal and medical data.
- Explicit user consent for location tracking and sharing.
- Rate-limiting and spam prevention to avoid misuse of SOS feature.
- Clear disclaimers that AI chatbot advice does not replace professional medical or legal consultation.

5. System Implementation & Interface Overview (Simple Points)

5.1 Technology Stack

- **Frontend:** HTML5, CSS3, vanilla JavaScript.
- **Browser APIs:** Geolocation, Web Speech (speech-to-text), Speech Synthesis (text-to-speech), LocalStorage.
- **Backend (planned):** Node.js, Express, third-party SMS/WhatsApp/email APIs, optional LLM API (OpenAI/Gemini).
- *(Screenshot placeholder: Code snippet view / architecture diagram – Figure 4)*

5.2 Key Modules

1. **Profile Management Module**
   - Loads and saves JSON profile objects in localStorage.
   - Provides validation for phone numbers and email fields.
2. **Location & Address Module**
   - Handles geolocation permissions and error messages.
   - Calls reverse-geocoding service and updates UI with address.
3. **Risk Assessment & Message Engine**
   - Parses incident description, computes risk level, and styles UI.
   - Assembles guardian message text and logs it into the alerts list.
4. **Alert Logging Module**
   - Prepends new alerts in reverse chronological order.
   - Maintains visual history for later reporting or screenshots.
5. **AI Chat Module**
   - Maintains conversation in the popup, handles send/receive, and integrates voice I/O.

6. Conclusion and Future Work

6.1 Conclusion

SentriXSOS tries to bridge a practical gap between very simple panic buttons and complex emergency-management systems. By combining a clean web interface, pre-saved safety profiles, automatic location and address fetching, basic risk analysis, and a built-in AI support chatbot, it offers a more informative SOS message to guardians with very little extra effort from the user. The project also shows that many useful features can be implemented using only front-end technologies, which makes it suitable as a student research and development project that can later grow into a more complete application.

In future versions, several directions can be explored:

1. **Real alert delivery:** Integrate Node.js backend APIs and external gateways so that the same JSON payload used in the UI is actually delivered as SMS, WhatsApp, or email to configured contacts**.**
2. **Data security and privacy:** Add client-side encryption for stored profiles, use HTTPS for all communication, and provide clear privacy settings.
3. **Smarter risk analysis:** Train a small supervised model or use a cloud-based LLM classification API on anonymized emergency texts to support more flexible phrasing and multiple languages.
4. **Richer chatbot:** Replace template-based replies with a controlled LLM service that can answer in English and Marathi, while keeping strict safety filters and disclaimers about medical and legal advice.
5. **Sensor-driven triggers:** Experiment with device sensors (accelerometer for sudden impact, inactivity for possible fainting) to auto-suggest SOS when unusual patterns are detected.
6. **User study at scale:** Conduct a structured questionnaire-based study with a larger sample of students and working women to measure perceived safety, message clarity, and willingness to use the app regularly.

If these enhancements are implemented, SentriXSOS could move from a college-level prototype to a more practical emergency companion that supports different user groups and works smoothly on low-end Android devices through the browser or as a lightweight progressive web app.

References :

1. Ahuja, R., & Bhatt, G. (2023). Design of a mobile SOS application for women safety using GPS and SMS gateway. *International Journal of Computer Applications*, 185(12), 10–18.
2. Al-Rubaie, M., & Chang, J. M. (2019). Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2), 49–58.
3. Boulis, A., & Ostendorf, M. (2005). A stochastic model for text categorization and clustering. *Computer Speech & Language*, 19(4), 465–496.
4. Chakraborty, S., & Dey, A. (2021). Location-based emergency alert system using Android and Google Maps API. *International Journal of Advanced Computer Science and Applications*, 12(6), 221–228.
5. Chatterjee, P., & Das, S. (2020). A review on mobile applications for women safety in India. *International Journal of Emerging*

Technologies and Innovative Research, 7(4), 560–566.

6. Council of Europe. (2019). *Guidelines on Artificial Intelligence and Data Protection*. Strasbourg: Council of Europe.

7. Das, R., & Roy, D. (2022). Context-aware emergency response system using smartphone sensors. *International Journal of Engineering Research & Technology*, 11(3), 95–101.

8. Government of India. (2022). *112 India – Single Emergency Helpline*. Ministry of Home Affairs.

9. Gupta, A., & Sharma, N. (2020). Progressive web applications: A modern approach for cross-platform mobile systems. *International Journal of Web Engineering*, 19(2), 45–59.

10. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

11. Jain, M., & Mehta, R. (2019). Real-time accident detection and reporting system using smartphone sensors. *International Journal of Intelligent Transportation Systems Research*, 17(4), 284–295.

12. Kaur, P., & Kaur, J. (2021). Women safety mobile applications: A comparative study. *International Journal of Scientific & Technology Research*, 10(1), 309–314.

13. Kumar, V., & Singh, A. (2022). Development of a smart SOS application for real-time emergency response. *International Journal of Novel Research in Computer Science and Software Engineering*, 9(5), 1–7.

14. Li, X., & Zhao, Y. (2020). A survey of emergency response support systems based on mobile and web technologies. *Journal of Network and Computer Applications*, 153, 102506.

15. Mishra, S., & Patil, R. (2023). Browser-based geolocation services and their applications in public safety. *International Journal of Web and Mobile Networks*, 15(2), 77–86.

16. OpenStreetMap Foundation. (2024). *Nominatim – OpenStreetMap Search and Reverse Geocoding Service*.

17. Saini, R., & Gupta, S. (2021). Voice-enabled chatbots for healthcare support: A review. *Health Informatics Journal*, 27(4), 1–15.

18. Sharma, P., & Verma, K. (2020). Sentiment and emotion analysis for emergency text classification. *Procedia Computer Science*, 167, 216–225.

19. Singh, R., & Yadav, P. (2019). Emergency SOS application: Design and implementation using Android. *International Journal of Engineering and Advanced Technology*, 8(6), 590–595.

20. SOS Emergency Alert and Assistance Mobile Application. (2020). *International Research Journal of Web Engineering*, 4(2), 15–21.

21. Srivastava, S., & Bansal, A. (2022). Sensor-driven emergency SOS app with real-time tracking and cloud storage. *International Journal of Recent Engineering Science*, 9(4), 47–53.

22. UNESCO. (2021). *Ethical principles for artificial intelligence in critical infrastructure*. Paris: UNESCO Publishing.

23. Wang, Y., & Li, H. (2022). User experience evaluation of safety and security mobile applications. *International Journal of Human–Computer Interaction*, 38(9), 851–866.

24. World Health Organization. (2018). *Violence against women: Key facts*. WHO Fact Sheet.