WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Women's Safety System

C. Sanjay[a,*], S. R. Rithish[a], R. Sakthi Saran[a], Ms. S. Vijayalakshmi[b]

[a]Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, Coimbatore-641062, Tamil Nadu, India
[b]Assistant Professor, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, Coimbatore-641062, Tamil Nadu, India

A B S T R A C T :

Personal safety remains a critical concern for women during daily commutes, late-night travel, and movement through unfamiliar or poorly lit environments. In threatening situations, victims often experience panic and cognitive overload, which makes standard smartphone emergency actions difficult to carry out. Unlocking the phone, searching contacts, dialing authorities, finding coordinates, and typing a message take time that a person in danger may not have, and national helpline numbers such as 112, 1091, and 181 are often forgotten under duress. This paper presents the Smart Women's Safety System, a web platform that unifies emergency operations in a single dashboard. With one tap, the application triggers an SOS workflow that captures live coordinates through the HTML5 Geolocation API, generates a real-time Google Maps link, and attaches it to pre-crafted WhatsApp distress templates (Immediate Help, Unsafe Vehicle, and Route Tracking). The platform also provides one-touch speed dialers for official emergency services and an Emergency Family Circle of verified guardian and parent contacts. It is built with HTML5, responsive CSS3, and vanilla JavaScript, needs no app-store installation, and follows an offline-first design using browser localStorage with optional Google Identity Services and Firebase Cloud Firestore synchronization. The system is a foundation for fast emergency coordination, and further field evaluation is required to measure its reliability and usability in real-world conditions.

Keywords: Women's Safety, SOS Alert, HTML5 Geolocation API, Emergency Helplines, WhatsApp Distress Message, Offline-First Web Application, localStorage, Firebase Cloud Firestore.

Introduction

Women frequently travel to work, college, and home at times and through places where they may feel unsafe. In such moments the most valuable resource is time. A person who is frightened or threatened cannot be expected to follow a long sequence of steps on a phone, yet this is what a typical emergency requires: unlock the device, find the right contact, open a map to learn the location, type a message, and separately dial a helpline.

Modern browsers now expose capabilities that were once limited to native applications, including geolocation, local data storage, and links that open the telephone dialer and messaging applications with prefilled content. Combined, these features make it possible to build a safety tool that runs entirely in a standard web browser and compresses the emergency response into a single action.

1.1. Existing challenges in emergency response

Existing options include calling a helpline, messaging relatives manually, sharing a live location from a map application, and installing dedicated safety applications. Each of these covers only part of the need. Calling a helpline does not tell the contacts where the user is, and manual messaging requires typing while under stress. Many safety applications must be downloaded from an app store, which excludes devices where installation is not possible and users who have not installed the application before an incident.

A second challenge is recall. National numbers such as 112 (emergency), 1091 (women helpline), and 181 (women helpline) are easy to forget or to mistype under duress. Finally, tools that depend on constant network access may fail in areas of poor connectivity, which are common in transit corridors and unfamiliar locations.

1.2. Role of web technologies

The HTML5 Geolocation API allows a web page, with the user's permission, to obtain the device's coordinates. A coordinate pair can be inserted into a Google Maps URL, which any recipient can open to see the position. A WhatsApp link can open a conversation with a prefilled text message, and a telephone link can open the dialer with a number already entered. Browser localStorage can keep small amounts of data on the device so that it remains available without a network connection.

These features have limits. Geolocation requires user permission and a secure (HTTPS) context, accuracy depends on the device and environment, and a web page cannot send a message silently on the user's behalf: the user still has to confirm the message in the messaging application. The system is therefore designed to minimise the number of steps, not to remove the user from the loop.

1.3. Problem statement and objectives

Emergency operations on a smartphone are fragmented across contacts, maps, messaging, and dialer applications, and this fragmentation costs time when it matters most. There is a need for a lightweight, universally accessible coordination layer that brings these operations together with a minimum of user effort.

The main objective of this work is to propose a web-based Smart Women's Safety System that unifies SOS alerting, live location sharing, distress messaging, helpline dialing, and trusted-contact management in a single dashboard. The specific objectives are:

i. To provide a single-tap SOS workflow that starts all required emergency actions.
ii. To capture live coordinates using the HTML5 Geolocation API and generate a real-time Google Maps link.
iii. To attach the link to pre-crafted WhatsApp distress templates for Immediate Help, Unsafe Vehicle, and Route Tracking.
iv. To provide one-touch speed dialers for the official emergency numbers 112, 1091, and 181.
v. To maintain an Emergency Family Circle with dedicated, verified phone numbers of primary guardians and parents.
vi. To follow an offline-first design in which contact numbers, medical profile information, and emergency guidelines are stored in browser localStorage.
vii. To support optional Google Identity Services sign-in and Firebase Cloud Firestore synchronization.
viii. To evaluate the application through functional, integration, and usability testing.

Materials and Methods

1.4. Development approach

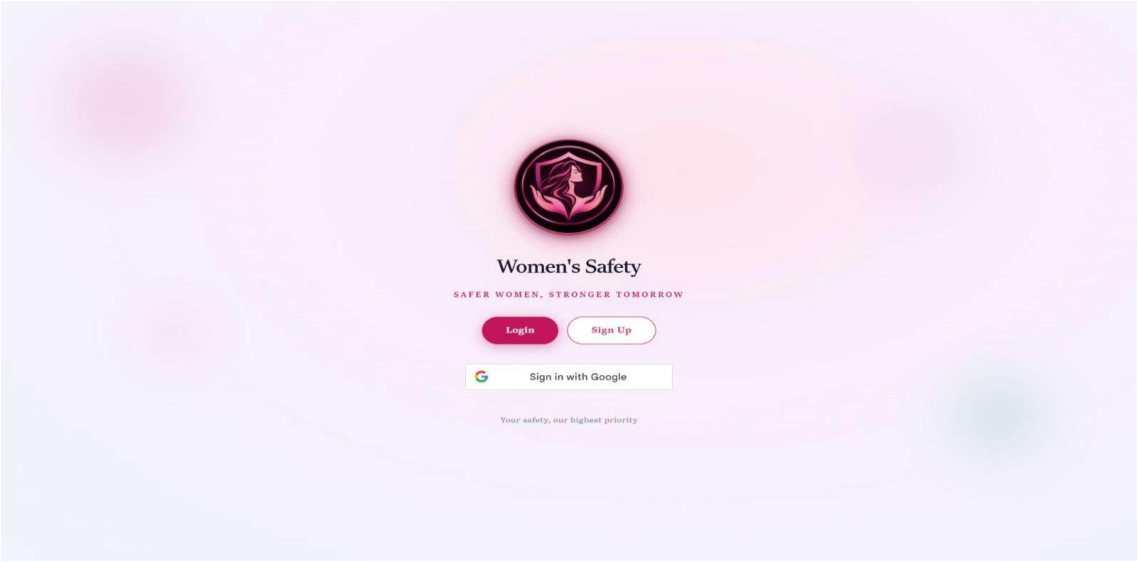
The project follows a structured development approach consisting of problem identification, requirement analysis, system design, implementation, integration, testing, and evaluation. During problem identification, the difficulties of performing emergency actions under stress and the fragmentation of existing tools are considered.

Requirement analysis identifies the main functions: SOS triggering, location capture, message generation, speed dialing, trusted-contact management, local storage of personal data, and optional cloud synchronization. System design defines how the dashboard, the SOS logic, the device services, and the storage components relate to one another. Implementation develops these modules, integration connects them, and testing identifies errors and areas requiring improvement.

1.5. System architecture

The system is organised into layers: a presentation layer (the responsive dashboard), an application logic layer (the SOS workflow and message builder), a device services layer (Geolocation API, telephone links, and WhatsApp links), a local storage layer (browser localStorage), and an optional cloud layer (Google Identity Services and Firebase Cloud Firestore).

The presentation layer displays the SOS control, the message templates, the helpline buttons, and the Emergency Family Circle. The application logic layer coordinates the steps triggered by a tap. The device services layer hands control to the browser and the operating system to obtain coordinates, open the dialer, or open the messaging application. The local storage layer is the primary data source, and the cloud layer is used only when the user chooses to sign in and synchronize. Because the layers are logically separate, a component such as the template set or the storage module can be changed without redesigning the whole application.



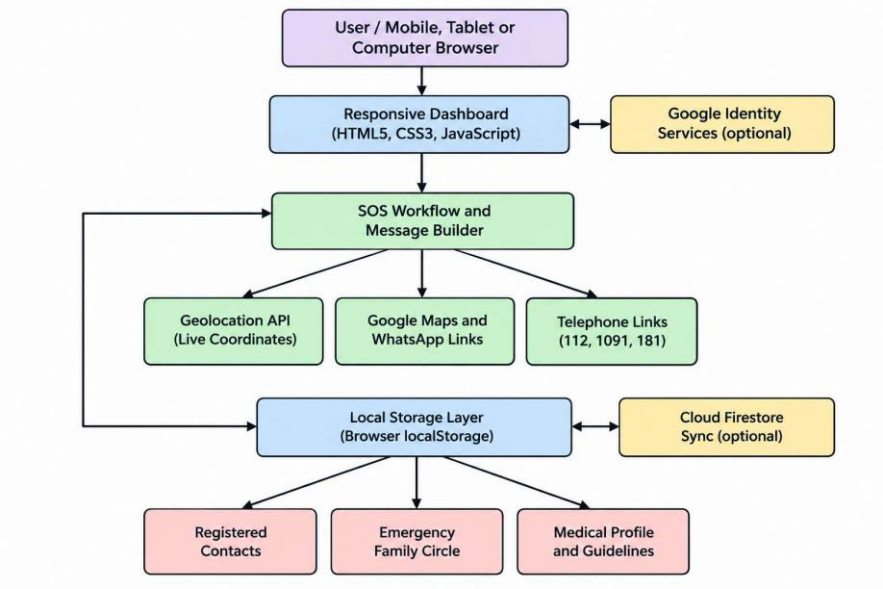


Fig. 1 - System Architecture of the Proposed Smart Women's Safety System.

1.6. Technology stack and development environment

The platform is built with HTML5 for structure, modern responsive CSS3 for layout across phones, tablets, and computers, and vanilla JavaScript for behaviour, without a heavy framework. This keeps the application small and quick to load on modest devices and slow connections. Because it runs in the browser, no app-store download or device-specific installation is required.

The HTML5 Geolocation API provides coordinates after the user grants permission. Location features require the page to be served over HTTPS. Google Maps URLs are used to produce a shareable location link, WhatsApp click-to-chat links are used to open a conversation with prefilled text, and telephone (tel:) links are used for the speed dialers. Google Identity Services and Firebase Cloud Firestore are optional components used for sign-in and synchronization. Development and testing can be performed with a standard code editor and current web browsers on Android, iOS, and desktop systems.

1.7. Data collection and processing

The system handles a small set of personal data: the contact numbers of the Emergency Family Circle, optional medical profile information, emergency guidelines, and, at the moment of an SOS, the device's coordinates. Coordinates are requested on demand when the SOS workflow is started and are placed in the message that the user sends.

Personal information should be collected only when necessary for the intended functions. Medical and contact information is sensitive, so the application keeps it on the user's device by default and uses cloud synchronization only when the user opts in. Inputs such as phone numbers should be validated before they are stored, and errors such as permission denial, positioning timeouts, and an unavailable network should be handled with clear messages to the user.

1.8. Data storage design

The storage design follows an offline-first approach. Browser localStorage holds the registered contact numbers, the medical profile, and the emergency guidelines as small text records, so the dashboard can be loaded and the stored information used even with low or no connectivity. When the user signs in with Google Identity Services and enables synchronization, the same records can be written to Firebase Cloud Firestore so that they can be restored on another device.

localStorage is limited in capacity, is stored per browser and per site, and is readable by anyone who can open the browser on that device. Firestore security rules should therefore restrict each user's records to that user, and the application should avoid storing more personal information than the functions require.

1.9. SOS workflow and message generation

When the user taps the SOS control, the application requests the device's position through the Geolocation API. On success, the latitude and longitude are inserted into a Google Maps link. The user's selected template (Immediate Help, Unsafe Vehicle, or Route Tracking) is combined with the link to form the final text, which is passed to a WhatsApp link that opens the chosen conversation with the message already written. The user then confirms the message in WhatsApp.

If the position cannot be obtained, for example because permission was denied or the request timed out, the application should say so clearly and still offer the template and the helpline dialers, rather than leaving the user without a way to act. Table 1 compares the steps of a conventional manual response with those of the proposed system.

Table 1 - Comparison of conventional emergency steps and the proposed workflow.

Step	Conventional manual response	Smart Women's Safety System
Reaching the alert function	Unlock phone, find the right application	Open the dashboard and tap SOS
Finding the location	Open a map, read or copy the position	Captured automatically by the Geolocation API
Writing the message	Type a description under stress	Pre-crafted template selected
Choosing recipients	Search the contact list	Emergency Family Circle contacts
Calling authorities	Remember and dial the number	One-touch dialers for 112, 1091, and 181

1.10. Application workflow

The workflow begins when the user opens the dashboard, which loads the stored contacts, profile, and guidelines from localStorage. The user can then tap SOS, choose a template, or press a helpline button. During an SOS, the application obtains coordinates, builds the Maps link, assembles the message, and opens the messaging application. The user may also dial a helpline in one touch at any point. When the user is signed in, changes to the stored data can be synchronized to Firestore in the background.

After each action the application should give clear feedback about its status, for example distinguishing a message prepared with a location link from one prepared without it, so that the user knows what is being sent.

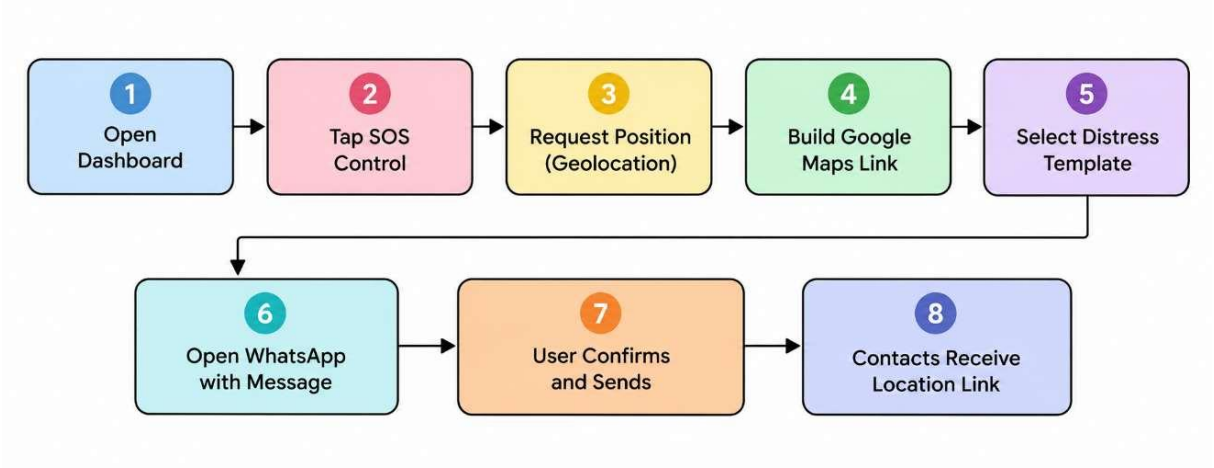


Fig. 2 - Workflow of the Proposed Smart Women's Safety System.

1.11. Testing and evaluation

Testing is required to evaluate the correctness and reliability of the system. Unit testing can examine individual functions such as the message builder and the link generator, and integration testing checks the interaction among the dashboard, the device services, local storage, and the optional cloud layer.

System testing evaluates the whole workflow on different devices and browsers, including permission handling, template selection, dialing, and offline loading. Usability testing examines whether users can find and start the SOS action quickly, for example with the System Usability Scale [9]. Any quantitative claim about response time, location accuracy, or usability should be reported only when supported by actual measurements.

Results and Discussion

1.12. Proposed system functionality

The proposed application brings the main emergency functions into a centralized interface: single-tap SOS, live location sharing through a Google Maps link, WhatsApp distress templates, speed dialers for 112, 1091, and 181, an Emergency Family Circle, and offline-first storage of contacts, medical profile information, and guidelines. Optional sign-in and Firestore synchronization extend this to multiple devices.

The integration of these functions aims to reduce the number of steps and the amount of thought needed to ask for help. The actual performance of the application must be established through testing of the implemented features and their interactions.

1.13. Functional evaluation

Functional evaluation examines whether each module behaves according to its requirements. Representative test scenarios are summarised in Table 2. Results should be documented from observed behaviour, and features that have not been tested should be identified as pending evaluation.

Table 2 - Representative functional test scenarios for the proposed system.

Test ID	Module	Test scenario	Expected behaviour
T1	SOS trigger	Tap SOS with location permission granted	Coordinates captured, Maps link generated, message prepared
T2	Geolocation	Deny permission or let the request time out	Clear message shown; templates and dialers remain available
T3	Templates	Select each of the three templates	Correct text with the location link inserted
T4	Speed dialers	Tap the 112, 1091, and 181 buttons	Dialer opens with the correct number
T5	Family Circle	Open and use a guardian contact	Correct verified number called or messaged
T6	Offline storage	Reload the page with the network off	Contacts, profile, and guidelines still available
T7	Synchronization	Sign in and sync, then open on a second device	Stored data restored; local data intact if offline
T8	Responsiveness	Open on a phone, a tablet, and a computer	Layout adapts and functions work on each

Any failed tests should be investigated and repeated after corrections. Location accuracy and the time from tap to message ready should be measured on several devices and in different environments, and reported per device type rather than as a single figure.

1.14. Comparison with conventional approaches

Conventional approaches rely on several separate steps and applications: unlocking the phone, reaching a contact, finding the location, composing a message, and dialing a helpline. Dedicated safety applications can automate some of this, but they require installation beforehand. The proposed system aims to bring these actions together in a browser-based dashboard that needs no installation, potentially improving accessibility and speed of use.

The benefit of this approach depends on measured performance. The proposed system should not be considered faster or more reliable than existing tools without comparative testing.

1.15. Discussion

The proposed design shows how widely available web features can be combined into a practical emergency tool. The offline-first design keeps the most important information on the device, and the absence of an installation step makes the tool available on almost any device with a modern browser.

The system also has limitations that should be stated plainly. It is a web page, so it generally requires the device to be unlocked and the page to be open, and it cannot run reliably in the background. The message still has to be confirmed by the user, and sending it requires WhatsApp and an internet connection even though stored data is available offline. Location accuracy varies with the device and environment, and an accidental tap may send a false alert. Contacts and medical information stored in localStorage are visible to anyone with access to the browser on that device. The system is meant to complement, and not replace, official emergency services.

Further evaluation should measure the speed of the workflow, the accuracy of location sharing, and the usability of the interface with real users, including users who are stressed or in a hurry. These measurements would provide stronger evidence of effectiveness.

Conclusion

The Smart Women's Safety System presents a web-based approach to personal safety that unifies SOS alerting, live location sharing, pre-crafted WhatsApp distress templates, one-touch helpline dialing, and an Emergency Family Circle in a single dashboard. Built with HTML5, responsive CSS3, and vanilla JavaScript, it is lightweight, cross-platform, and free of app-store dependence, and its offline-first design with optional Google Identity Services and Firebase Cloud Firestore synchronization keeps critical information available in low-bandwidth conditions.

The practical effectiveness of the system depends on correct implementation and testing of each component and on how users behave in real emergencies. Future development can focus on background or lock-screen triggering, delivery of alerts to several contacts at once, live location tracking during a journey, support for regional languages, and protection of locally stored data with encryption.

In conclusion, the proposed application offers a foundation for a practical, accessible safety tool. With careful validation, sound privacy safeguards, and user evaluation, it can contribute to reducing the time and effort needed to ask for help.

Acknowledgments

The authors would like to thank Sri Shakthi Institute of Engineering and Technology for providing the academic environment and facilities required for this project. The authors also express their sincere gratitude to the Head of the Department of Information Technology, the project guide, and the faculty members for their guidance, encouragement, and support throughout the development of the project.

REFERENCES

[1] Mozilla Developer Network. Geolocation API. Available at: https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API
[2] Mozilla Developer Network. Web Storage API (localStorage). Available at: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
[3] Google. Maps URLs. Google Maps Platform Documentation. Available at: https://developers.google.com/maps/documentation/urls/get-started
[4] WhatsApp. Click to Chat (wa.me links). WhatsApp Help Center. Available at: https://faq.whatsapp.com/
[5] Google. Google Identity Services for Web. Available at: https://developers.google.com/identity/gsi/web
[6] Google Firebase. Cloud Firestore Documentation. Available at: https://firebase.google.com/docs/firestore
[7] Government of India. Emergency Response Support System (ERSS), 112 India. Available at: https://112.gov.in
[8] National Crime Records Bureau. Crime in India. Available at: https://ncrb.gov.in
[9] J. Brooke, "SUS: A quick and dirty usability scale," in *Usability Evaluation in Industry*, P. W. Jordan et al., Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.
[10] OWASP Foundation. OWASP Application Security Guidance. Available at: https://owasp.org/