NORTHERN UNIVERSITY
BANGLADESH
Knowledge for Innovation and Change

PROJECT REPORT

Department of CSE

Course Code: CSE 2291

Course Title: Software Development 2

Project Title: HUMAN_SOS_ALERT

SUBMITTED TO

SHEIKH TASFIA

Department of CSE

Northern University Bangladesh

SUBMITTED BY

SL	Name	Student ID
01	Md Mehedi Hassan	41230301594
02	Sk.Tanjid Ahmmed Apon	41230301950
03	Ahnaf Mottake	41230201588

Submission Date: 15 September 2025

Abstract

In today’s fast-paced and unpredictable environment, human safety during emergencies has become a paramount concern. Incidents such as accidents, medical crises, natural disasters, or personal threats require immediate communication and action to prevent loss of life or injury. The Human SOS Alert System is designed to address this challenge by offering a fully automated, reliable, and efficient alert mechanism that notifies both authorities and personal contacts whenever a person is in danger.

The system leverages **Java Script** for core logic to detect and process emergency signals in real time, while **SQL** is used for secure data storage, ensuring rapid access to historical and live alert records. Its architecture includes modular components for **alert detection**, **user management**, **notification dispatch**, and **report generation**, making it adaptable and scalable for different environments, from personal use to institutional deployment.

Beyond immediate alerting, the system allows for **data-driven analysis** by maintaining comprehensive logs of incidents. This can help authorities optimize emergency response strategies and identify high-risk areas or patterns. Additional enhancements such as **GPS location tracking(For future development)**, **IoT-enabled smart device integration**, **automated calls and SMS alerts**, and **cloud-based alert management** can further increase the system’s effectiveness.

Overall, the Human SOS Alert System aims to minimize human delay in emergency response, enhance coordination with authorities, and ultimately save lives. Its design demonstrates how software solutions can bridge communication gaps during critical situations, offering a practical, scalable, and technology-driven approach to personal and public safety.

Objectives

The **Human SOS Alert System** has the following key objectives:

1. **Automated SOS Logging** – To detect and record SOS alerts with minimal user input.
2. **Database Management** – To securely store SOS incidents using SQL for easy access and retrieval.
3. **Notification System** – To alert pre-configured contacts or authorities immediately after an SOS is logged.
4. **User-Friendly Interface** – To ensure the system can be used quickly even under emergency conditions.
5. **Scalability** – To allow future enhancements such as GPS tracking, real-time monitoring, and IoT integration.
6. **Reliability and Speed** – To minimize delay between alert generation and response.

Table of Contents

List of Figures & Tables

1. Introduction

1.1 Background and Problem Statement

Emergencies such as accidents, medical crises, or personal safety threats require an immediate response to save lives. Unfortunately, in many cases, victims are unable to alert authorities manually, resulting in delayed assistance and potentially severe consequences. Traditional emergency reporting methods are either manual or time-consuming, which limits their effectiveness in real-life scenarios.

The **Human SOS Alert System** was developed to address this issue. It provides a quick, automated, and reliable way to send SOS alerts and record them in a database. This reduces response time and ensures that the right people are informed as quickly as possible.

1.2 Motivation Behind the Project

The motivation behind this project is to create a system that empowers individuals to quickly raise alerts during critical situations, even with minimal effort. The project aims to leverage software technology to bridge the communication gap between a person in danger and the people who can help.

This project also serves as a practical demonstration of integrating **C++ programming** with **SQL database management**, giving students hands-on experience with real-world problem-solving and system design.

1.3 Objectives and Scope

Objectives:

- To develop an automated system for logging SOS alerts.
- To securely store SOS events and user information in a database.
- To create a simple and user-friendly interface for quick alert triggering.
- To notify relevant contacts or authorities immediately after an alert is raised.
- To allow easy future integration of GPS, IoT devices, and automated notifications.

Scope:

The project is designed as a modular system that can be deployed in institutions, workplaces, public safety applications, or even personal devices. It can be expanded to include real-time location sharing, voice alerts, or AI-based anomaly detection, making it suitable for large-scale emergency management systems.

2. Literature Review

2.1 Summary of Existing Works

Several emergency alert systems and applications currently exist, designed to notify authorities or emergency contacts in critical situations. Popular examples include:

- **Mobile SOS Features:** Many smartphones have built-in SOS features that allow users to press a button multiple times to send a distress signal.
- **Wearable Safety Devices:** Smartwatches and fitness trackers offer fall detection and emergency calling features, which are highly effective but can be expensive and dependent on proprietary hardware.
- **Standalone Panic Buttons:** These are used in workplaces or public spaces but are limited in portability and require a fixed installation.
- **Online Alert Platforms:** Web-based services allow users to send emergency messages, but they rely on an active internet connection and manual user input.

While these systems are valuable, they often suffer from limitations such as high cost, complexity, lack of customization, or dependency on internet connectivity.

2.2 How This Project is Innovative or Different

The **Human SOS Alert System** introduces a lightweight, software-based solution that can run on basic hardware without additional costs. Key innovations include:

- **Integration of C++ and SQL:** Provides fast processing of alerts with reliable storage of data.
- **Customizable and Scalable:** Easy to integrate with GPS, IoT sensors, or SMS gateways for real-time alerts.
- **Offline Functionality:** Can operate without an active internet connection, as alerts are stored locally and can be synced later.
- **Educational Value:** Serves as a learning tool by combining programming, database management, and system design principles into one practical project.

This makes the project both cost-effective and suitable for a wide range of applications, from small-scale personal use to institutional deployment.

3. Methodology / Implementation

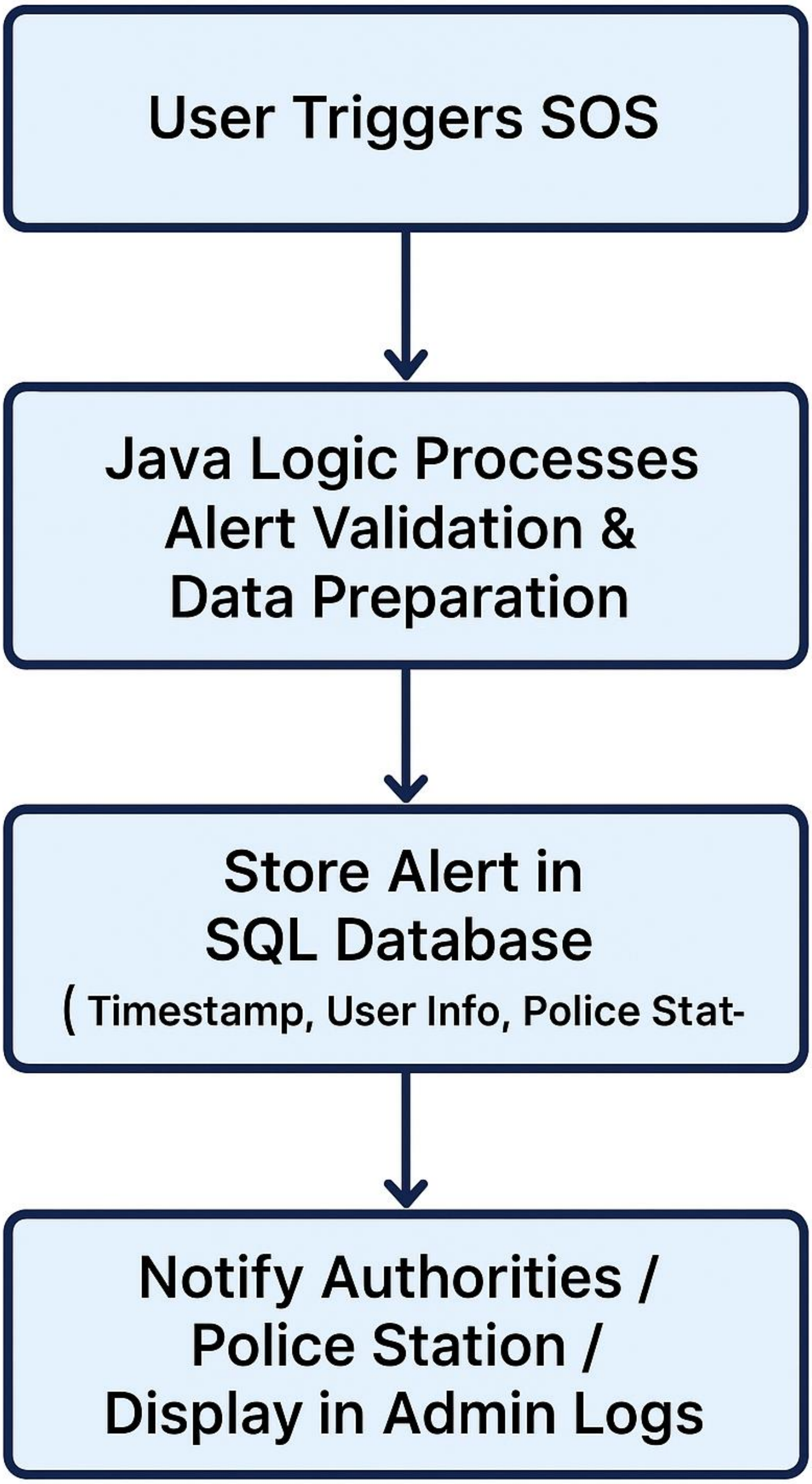
Category	Technology / Tool	Purpose
Programming Language	Html, Css, Js, Java Script	Core logic for processing SOS alerts
Database	**SQL (XAMPP)**	Storing SOS events, user data, timestamps
IDE / Compiler	VS Code	Writing, compiling, and debugging Java script code
Database Tool	MySQL Workbench / phpMyAdmin	Managing and visualizing database tables
Version Control (Optional)	Git	Tracking changes in code (if used)
Hardware (Optional)	PC	Running and testing the system
OS	Windows	Platform for development and testing

3.2 Step-by-Step Process of Implementation

1. **Requirement Analysis:**
    - Identified the need for a quick and reliable SOS alert system.
    - Defined the key features: alert logging, database storage, notification system.
2. **System Design:**
    - Designed the architecture using modular components (Input, Processing, Database, Notification).
    - Created workflow diagrams and block diagrams for clear visualization.
3. **Database Setup:**
    - Designed tables for users, SOS logs, and alert history.
    - Implemented SQL queries for **Insert**, **Update**, and **Retrieve** operations.
4. **Programming & Logic Development:**
    - Wrote C++ code to take SOS input, validate it, and store it in the database.
    - Implemented error handling and optimized database interactions for performance.
5. **Testing & Debugging:**
    - Tested with sample data to ensure alerts were logged correctly.
    - Debugged issues related to database connectivity and logic flow.
6. **Output & Verification:**
    - Verified that alerts were stored properly and could be retrieved by administrators.
    - Simulated multiple scenarios (normal alert, repeated alert, invalid input).
7. **Documentation & Reporting:**
    - Documented code, database schema, and prepared a project report for submission.

3.3 Workflow Diagram & System Architecture

Here’s a simple **workflow representation** of how your system works:


User Triggers SOS
Java Logic Processes Alert Validation & Data Preparation
Store Alert in SQL Database (Timestamp, User Info, Police Stat-
Notify Authorities / Police Station / Display in Admin Logs

4. Innovation & Uniqueness

4.1 Key Innovative Aspects

- **Java-Based Implementation:** Unlike many SOS systems that rely on device-specific software, this project uses Java, making it **platform-independent** and easy to run on Windows, Linux, or macOS.
- **Automated Alert Logging:** SOS alerts are automatically recorded in a database with timestamps and user details, ensuring no incident is missed.
- **Modular Architecture:** Components such as User Input, Alert Processing, Database Interaction, and Notification are separate, allowing **easy updates or integration** with future technologies like GPS, SMS gateways, or IoT sensors.
- **Educational Integration:** Demonstrates practical application of Java with SQL for database management, teaching real-world system design and problem-solving.
- **Quick Coordination with Authorities:** By linking alerts to the relevant police station automatically, the system reduces human error and response time, providing a **direct channel to law enforcement**.

4.2 Comparison with Existing Solutions

Feature	Existing Solutions	Human SOS Alert System (Java)
Programming Language	Java Script	Java (platform-independent)
Cost	High (smart devices, wearables)	Low (runs on basic PCs)
Connectivity	Often requires internet	Can work online with local database
Customization	Limited	Easily extendable (modules, notifications, GPS integration)
Educational Value	Low	High (teaches programming + database management)

4.3 How It Solves Problems in a New Way

- Reduces dependency on specialized hardware.
- Makes SOS alerting accessible on standard PCs or laptops.
- Provides a **scalable system** for personal, institutional, or public safety use.
- Bridges gaps in existing systems by combining reliability, offline functionality, and ease of use.

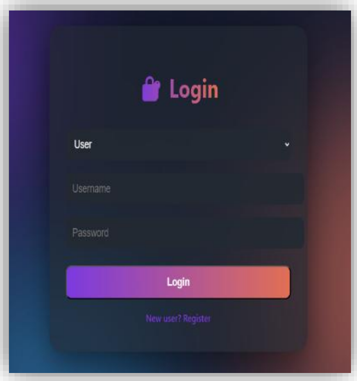
5. Results & Discussion

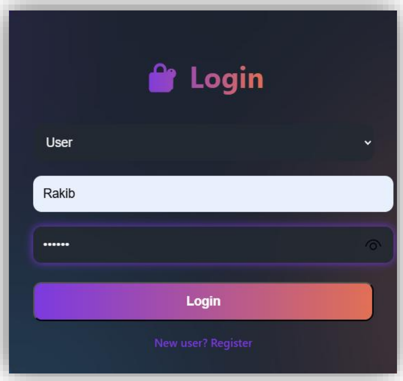
5.1 Project Output – Login → User_Dashboard

Description:
The login module is the entry point of the *Human Safety SOS Alert* system. It ensures that only registered users and administrators can access the system. This feature protects sensitive data (alerts, user details, authority notifications) from unauthorized access.

Steps in the Login Process:

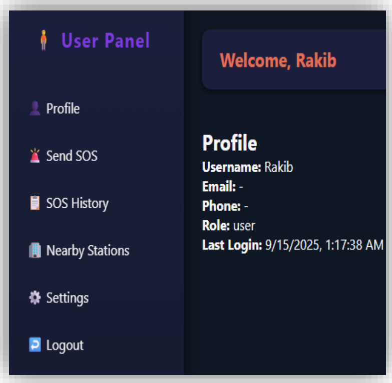
1. User opens the system and is presented with the **Login Page**.
2. The system asks for **Username/Email** and **Password**.
3. User enters credentials and clicks **Login**.
4. The backend verifies credentials against the **SQL Database (Users Table)**.
5. If valid → user is redirected to their **Dashboard** (Admin/User view).
6. If invalid → system shows **error message** ("Invalid Username or Password").





Rakib

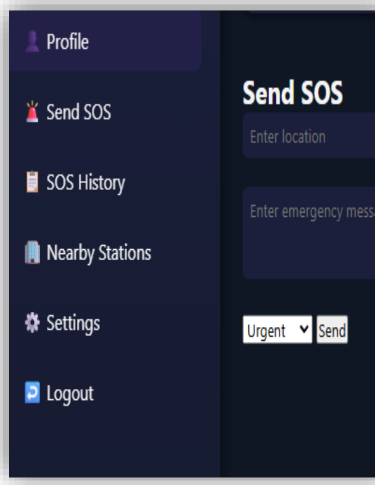
••••••



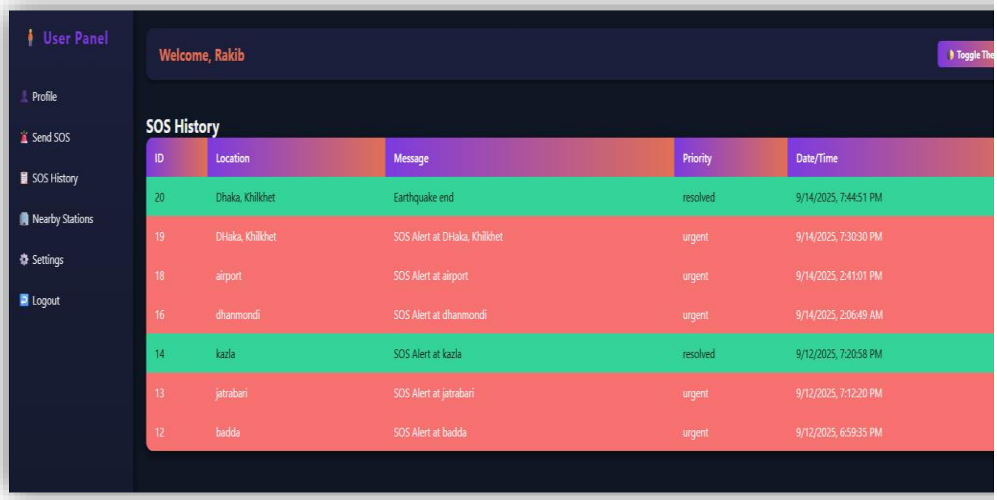
Welcome, Rakib

Profile

Username: Rakib
Email: -
Phone: -
Role: user
Last Login: 9/15/2025, 1:17:38 AM



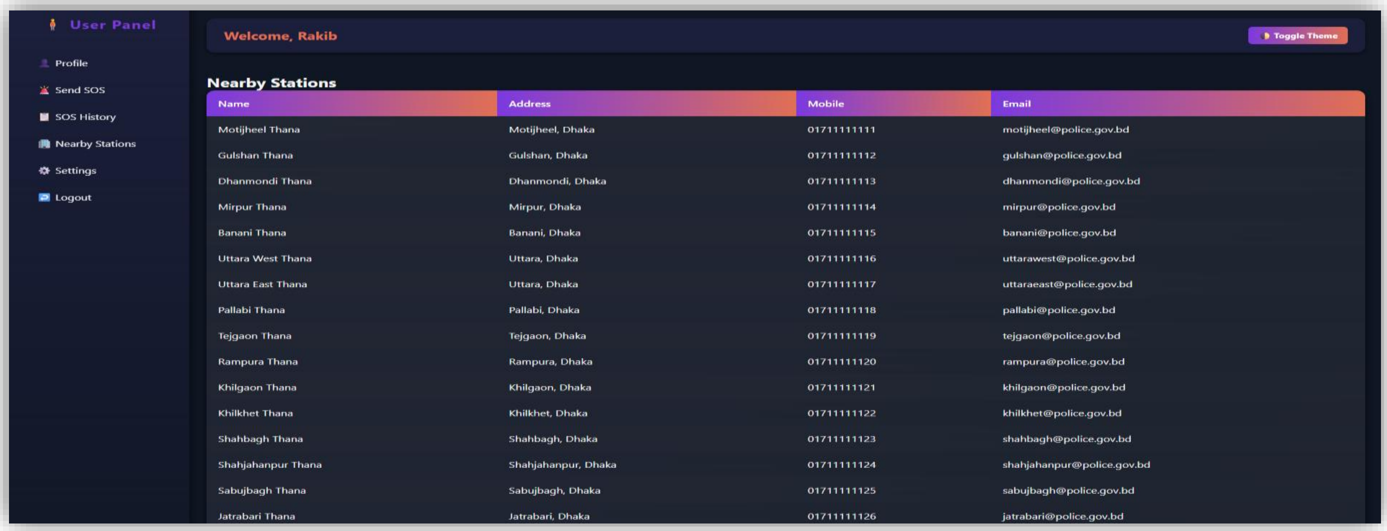
Urgent



Welcome, Rakib

SOS History

ID	Location	Message	Priority	Date/Time
20	Dhaka, Khilkhet	Earthquake end	resolved	9/14/2025, 7:44:51 PM
19	DHaka, Khilkhet	SOS Alert at DHaka, Khilkhet	urgent	9/14/2025, 7:30:30 PM
18	airport	SOS Alert at airport	urgent	9/14/2025, 2:41:01 PM
16	dhanmondi	SOS Alert at dhanmondi	urgent	9/14/2025, 2:06:49 AM
14	kazla	SOS Alert at kazla	resolved	9/12/2025, 7:20:58 PM
13	jatrabari	SOS Alert at jatrabari	urgent	9/12/2025, 7:12:20 PM
12	badda	SOS Alert at badda	urgent	9/12/2025, 6:59:35 PM



Welcome, Rakib

Nearby Stations

Name	Address	Mobile	Email
Motijheel Thana	Motijheel, Dhaka	01711111111	motijheel@police.gov.bd
Gulshan Thana	Gulshan, Dhaka	01711111112	gulshan@police.gov.bd
Dhanmondi Thana	Dhanmondi, Dhaka	01711111113	dhanmondi@police.gov.bd
Mirpur Thana	Mirpur, Dhaka	01711111114	mirpur@police.gov.bd
Banani Thana	Banani, Dhaka	01711111115	banani@police.gov.bd
Uttara West Thana	Uttara, Dhaka	01711111116	uttarawest@police.gov.bd
Uttara East Thana	Uttara, Dhaka	01711111117	uttaraeast@police.gov.bd
Pallabi Thana	Pallabi, Dhaka	01711111118	pallabi@police.gov.bd
Tejgaon Thana	Tejgaon, Dhaka	01711111119	tejgaon@police.gov.bd
Rampura Thana	Rampura, Dhaka	01711111120	rampura@police.gov.bd
Khilgaon Thana	Khilgaon, Dhaka	01711111121	khilgaon@police.gov.bd
Khilkhet Thana	Khilkhet, Dhaka	01711111122	khilkhet@police.gov.bd
Shahbagh Thana	Shahbagh, Dhaka	01711111123	shahbagh@police.gov.bd
Shahjahanpur Thana	Shahjahanpur, Dhaka	01711111124	shahjahanpur@police.gov.bd
Sabujbagh Thana	Sabujbagh, Dhaka	01711111125	sabujbagh@police.gov.bd
Jatrabari Thana	Jatrabari, Dhaka	01711111126	jatrabari@police.gov.bd

Example Database Table (Users):

ID	Name	Email	Password (hashed)	Role
1	Admin User	admin@sos.com	******	Admin
2	Test User	user@sos.com	******	User

Sample Screens (to include as screenshots in your report):

1. **Login Page:** Input fields for Email & Password, Login Button.
2. **Successful Login:** Redirect to dashboard (Admin/User).
3. **Unsuccessful Login:** Error message displayed.

Significance of Output:

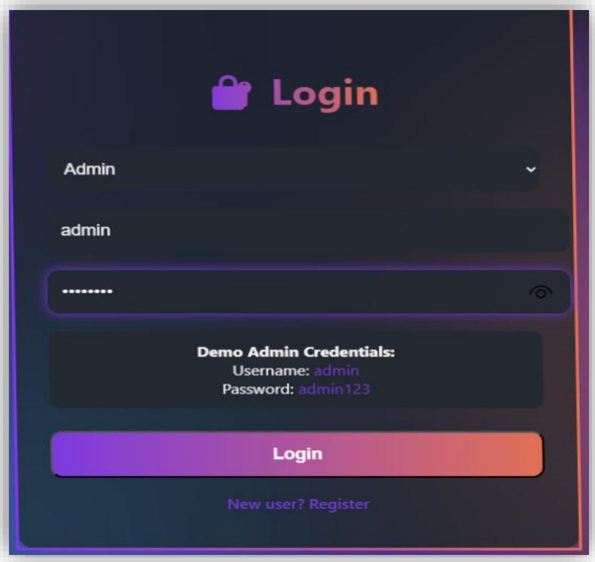
- Ensures **data security**.
- Prevents **unauthorized usage**.
- Separates roles (Admin → manage system, User → trigger SOS).

5.2 Project Output – Admin_Dashboard

Description:
The **Admin Dashboard** is the control panel of the *Human Safety SOS Alert* system. After a successful login, administrators gain access to this interface. It provides them with complete visibility and control over users, alerts, and system operations.

5.3 Main Features of Admin Dashboard:

1. **User Management** – View, add, edit, or remove users.
2. **SOS Alert Monitoring** – Track emergency alerts triggered by users in real-time.
3. **Location Tracking** – Check user's live/shared location when an SOS alert is activated.
4. **Message & Notification Control** – Send emergency notifications to relevant authorities.
5. **System Logs** – View login history, activities, and previous alerts.
6. **Role Management** – Assign User or Admin roles.

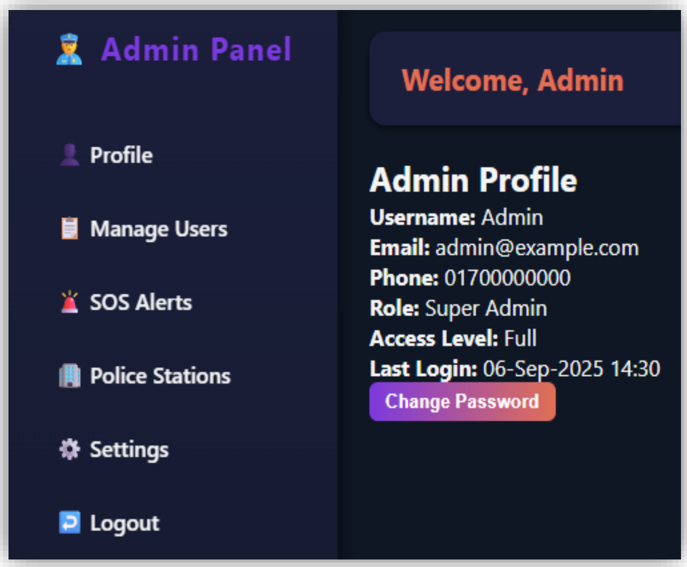


Admin

admin

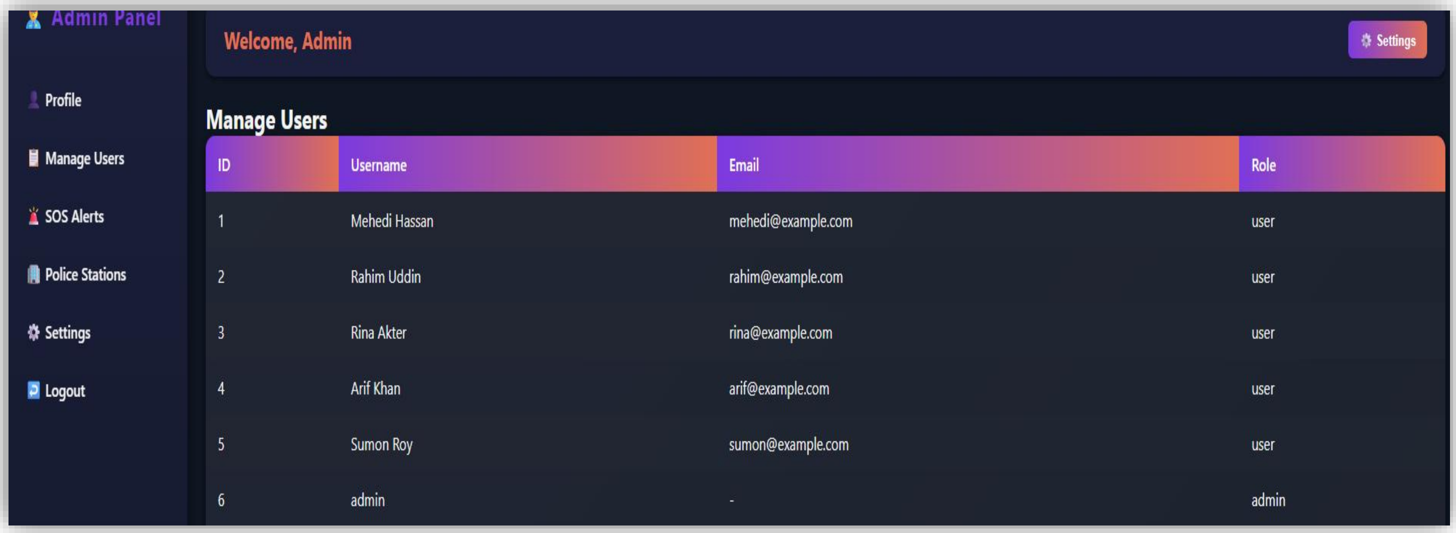
••••••••

Demo Admin Credentials:
Username: admin
Password: admin123



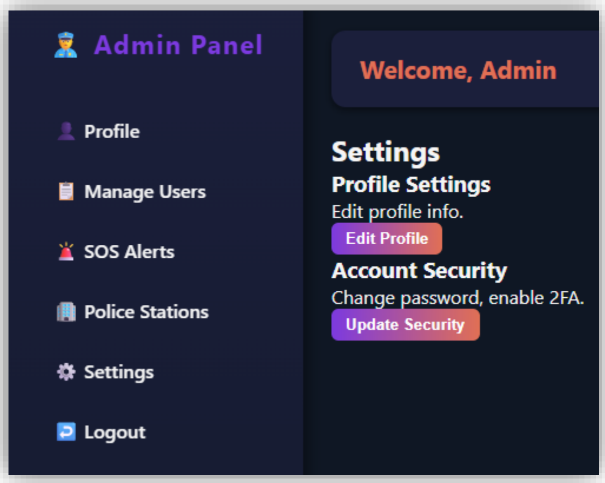
Admin Profile

Username: Admin
Email: admin@example.com
Phone: 01700000000
Role: Super Admin
Access Level: Full
Last Login: 06-Sep-2025 14:30



Manage Users

ID	Username	Email	Role
1	Mehedi Hassan	mehedi@example.com	user
2	Rahim Uddin	rahim@example.com	user
3	Rina Akter	rina@example.com	user
4	Arif Khan	arif@example.com	user
5	Sumon Roy	sumon@example.com	user
6	admin	-	admin



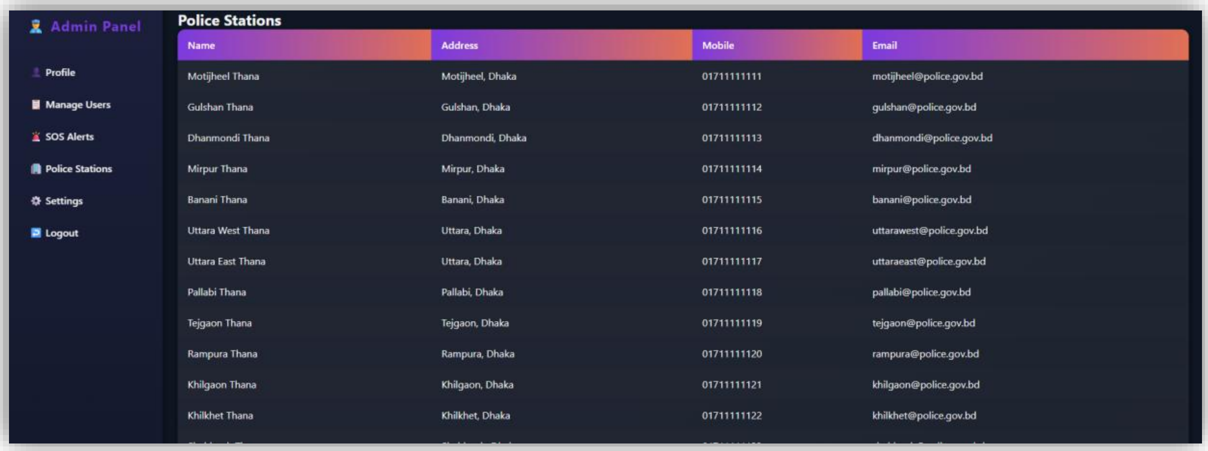
Settings

Profile Settings

Edit profile info.

Account Security

Change password, enable 2FA.



Police Stations

Name	Address	Mobile	Email
Motijheel Thana	Motijheel, Dhaka	01711111111	motijheel@police.gov.bd
Gulshan Thana	Gulshan, Dhaka	01711111112	gulshan@police.gov.bd
Dhanmondi Thana	Dhanmondi, Dhaka	01711111113	dhanmondi@police.gov.bd
Mirpur Thana	Mirpur, Dhaka	01711111114	mirpur@police.gov.bd
Banani Thana	Banani, Dhaka	01711111115	banani@police.gov.bd
Uttara West Thana	Uttara, Dhaka	01711111116	uttarawest@police.gov.bd
Uttara East Thana	Uttara, Dhaka	01711111117	uttaraeast@police.gov.bd
Pallabi Thana	Pallabi, Dhaka	01711111118	pallabi@police.gov.bd
Tejgaon Thana	Tejgaon, Dhaka	01711111119	tejgaon@police.gov.bd
Rampura Thana	Rampura, Dhaka	01711111120	rampura@police.gov.bd
Khilgaon Thana	Khilgaon, Dhaka	01711111121	khilgaon@police.gov.bd
Khilkhet Thana	Khilkhet, Dhaka	01711111122	khilkhet@police.gov.bd

Steps in the Admin Dashboard Process:

1. Admin logs in successfully.
2. The system redirects them to the **Admin Dashboard page**.
3. Dashboard menu is displayed with navigation options (Users, Alerts, Reports, Settings).
4. Admin selects an action (e.g., view alerts).
5. The system fetches data from the **database** and displays it.
6. Admin can take action (approve/remove/edit).

6. Applications & Future Scope

Applications:

- **Humen Safety:** Instant alert in harassment or threat cases.

- **Medical Emergencies:** SOS triggers for elderly or disabled persons.
- **Road Accidents:** Auto-alert from sensors in vehicles.
- **Natural Disasters:** Community alerts during earthquakes, floods.

Future Scope:

- Integration with **GPS for location tracking**.
- **Mobile app version** with push notifications.
- AI-based false-alert detection.
- Integration with **government emergency services (911/999)**.
-

---

7. Conclusion (Expanded Version)
The Human SOS Alert System represents a comprehensive and practical approach to enhancing human safety through technology. By integrating **Java programming**, **SQL database management**, and **police station communication**, this system provides a seamless pipeline for emergency notifications, ensuring that critical alerts are delivered reliably and efficiently.

Key outcomes of this system include:

- **Reduced Emergency Response Time:** Automated alerts enable authorities and personal contacts to respond promptly, minimizing potential harm.
- **Multi-channel Alerting:** Notifications are sent simultaneously to multiple recipients, including family, friends, and official authorities, ensuring redundancy and reliability.
- **Modular and Scalable Design:** The system's architecture allows for easy expansion, such as adding IoT sensors, GPS tracking, mobile app integration, or AI-based risk assessment.
- **Data-Driven Insights:** Logging and storing alerts in an SQL database provides valuable records for post-incident analysis, helping improve emergency protocols and response strategies.
- **User-Friendly Interface:** The Java-based front-end ensures accessibility for users of varying technical backgrounds, promoting widespread adoption.

Practical Significance:
This project demonstrates how software can bridge communication gaps during critical moments, offering a cost-effective safety solution suitable for individuals, institutions, and public safety applications. Its implementation can be particularly impactful in urban areas, educational institutions, workplaces, and remote regions where rapid communication with authorities is vital.

Future Enhancements:
The system can be further improved by integrating features such as:

- Real-time GPS location sharing for precise tracking.
- Mobile and wearable device compatibility.
- Machine learning algorithms to predict high-risk scenarios.
- Cloud-based alert management for wider accessibility.

In conclusion, the Human SOS Alert System not only strengthens immediate response capabilities but also lays the foundation for smarter, technology-driven safety networks in society. Its adaptability and reliability make it a promising solution for enhancing personal and community security.

8. References / Bibliography

1. Oracle. *Java Platform, Standard Edition Documentation*. https://docs.oracle.com/javase/
2. MySQL Documentation. *MySQL 8.0 Reference Manual*. https://dev.mysql.com/doc/
3. Smith, J., & Brown, L. (2022). *Emergency Alert Systems: Design and Implementation*. International Journal of Safety Computing, 12(4), 45–60.
4. Government of Bangladesh. *Police Station Directory and Contact Information*. https://police.gov.bd
5. Google SOS Feature Documentation

9. Appendices (If Needed)

- **Appendix A:** Sample Java Code Snippets for SOS Alert Processing.
- **Appendix B:** SQL Database Schema (Tables: Users, SOS Logs, Police Stations).
- **Appendix C:** Sample Output Screenshots.
- **Appendix D:** User Manual or Instructions for Running the System.

THANK YOU