

# Emergency Reporting System - Project Group 24

## Individual Contribution:

Everybody contributed equally to the project. 20% contribution of each member.

- Chowdhury Yasir (cya127): MD5 Crypto Hashing, show more info container styling
- Daiwik Marrott (drm11): Issue fixes across UI, bootstrap integration and UI enhancements
- Luvveer Singh Lamba (lsl11): Event & state management, leaflets Map API
- Priyansh Sarvaiya (pgs3): Linking events of table and map, Form Validation
- Sanchit Jain (sja164): Map marker events, Table of reports

## Project Overview

This project implements a web-based emergency reporting system for Metro Vancouver's 9-1-1 Emergency Call Answer Service. The application allows civilians to submit emergency reports and enables operators and first responders to monitor and review these reports. The system logs crucial information about each emergency, including witness details, emergency type, location, and additional comments.

## Getting Started

1. Clone the repository
2. Open `index.html` in a web browser
3. Use the form to submit emergency reports
4. View submitted reports on the map and in the list
5. Sort table by table header by clicking on a particular header
6. Filter reports by panning map to different locations

## Key Features

- Interactive map displaying emergency markers using Leaflet and OpenStreetMap
- Detailed emergency report submission form
- Dynamic list of emergency reports
- Map marker interaction to view emergency details
- Responsive web layout

## Technologies Used

- HTML5, CSS3, JavaScript

- Bootstrap 4.5.2 for responsive design
- Leaflet.js for map integration
- CryptoJS for MD5 hashing
- DOM Storage API for local data storage

## Project Requirements

1. Interactive map with emergency markers
2. Detailed emergency report submission
3. Passcode-protected modifications (to be implemented)
4. Local data storage using DOM Storage API
5. Input validation and error handling

## Methodology

Our team adopted an iterative development approach, focusing on implementing core functionalities first:

1. Set up the basic HTML structure with Bootstrap for styling
2. Implemented the emergency report submission form
3. Integrated Leaflet.js for map functionality
4. Developed the dynamic emergency list display

## Challenges Faced

1. First-time use of GitHub and git branching for most team members
2. Integrating Leaflet.js with dynamic report data
3. Implementing secure passcode protection for report modifications
4. Filtering reports based on map bounds
5. Handling filtered reports on map and additional information containers

## Lessons Learned

1. Importance of version control and collaborative coding using GitHub
2. Practical application of web APIs (Leaflet, DOM Storage)
3. Balancing functionality with user experience in web application design
4. State management in vanilla js
5. Always use SVG for icons to ensure consistency across devices
6. Create issue → assign members, add tags → Link commit to issue → Resolve/Comment → Merge

## Potential Issues and Future Improvements

1. Implement geolocation for more accurate emergency location reporting
2. Enhance security measures for passcode protection
3. Improve map interactivity with clustering for multiple reports in close proximity

4. Implement real-time updates for emergency reports
5. Club/Group emergency reports for same location

## **GitHub and Git Branch Usage**

This project marked the first time using GitHub and git branching for most team members. We learned to:

- Create and manage repositories
- Work with branches for feature development
- Resolve merge conflicts
- Collaborate effectively using pull requests