

# School of Computer Sciences & Engineering

**Department of Computer Science & Application**

**Community Engagement Project**

**Synopsis**

**On**

**‘Crime report safety system’**

By

1. **Shravani Hingmire. (230105011150)**
2. **Kunal Dorik. (230105011151)**
3. **Namratra Khandale. (230105011152)**

Class & Semester:SY BCA. DIV-B(2)

Under the Guidance of

**Prof. Deepali chaudhari.**

**Academic Year: 2024-25 odd Semester**

**Crime Report Safety system.**

# •Abstract

The **Crime Report Safety system** aims to provide a secure, anonymous, and realtime solution for crime reporting, designed for use by the general public and law enforcement agencies. Citizens can submit detailed crime reports, including type, location, time, and description, through an intuitive and user-friendly interface. Anonymity is a key feature of the app, allowing users to report crimes without revealing their identity, reducing the fear of retaliation or exposure.

Once submitted, crime reports are stored in a centralized and secure SQL database, allowing law enforcement to access, manage, and track cases efficiently. The app offers the functionality of real-time updates, ensuring that users and authorities are always aware of the status of a reported case. Law enforcement can update the status of crimes, allowing citizens to follow the progress of their reports.

This app enhances public safety by improving the efficiency of crime reporting and response times, fostering a collaborative relationship between the community and law enforcement. By enabling quick action and informed responses to crime, the app contributes to making communities safer.

* **Key Features of the Crime Report Safety App:**
* **Crime Reporting:** Citizens can submit detailed crime reports with critical data.
* **Anonymity:** Optional anonymous reporting, ensuring privacy for users.
* **Centralized Database:** All crime reports are stored and accessible by authorized law enforcement.
* **Real-time Updates:** Provides citizens and authorities with status updates on crime reports.
* **Location Tracking:** Optional geolocation features to track crime locations and map them for authorities.

# • Introduction

**Background and Problem Definition:**

* Crime is a pervasive issue in societies worldwide, and the ability to report crime efficiently is a cornerstone of maintaining public safety. Traditional methods of reporting crimes, such as calling police hotlines or visiting police stations, are often cumbersome and inconvenient, leading to delayed responses. Furthermore, many individuals hesitate to report crimes due to fears of retaliation or concerns about privacy.
* Additionally, law enforcement agencies often face challenges in managing crime data, tracking cases, and ensuring timely responses. Crime reports are often submitted in a fragmented manner, making it difficult for authorities to analyze and act on them swiftly. The **Crime Report Safety App** addresses these problems by offering a streamlined, user-friendly platform for reporting crimes.

**Scope and Purpose of the Project:**

* The **Crime Report Safety App** is a mobile/web-based application that allows citizens to report crimes easily, quickly, and securely. The app seeks to empower individuals to contribute to crime prevention by eliminating barriers such as anonymity concerns and complex reporting procedures. For law enforcement, the app provides a centralized, efficient system for tracking, managing, and responding to crime reports.
* The purpose of this app is to enhance the relationship between law enforcement agencies and the community. The app increases transparency, fosters public trust, and accelerates the response time to criminal activities. With features such as real-time status tracking and anonymous reporting, the app enables faster and more effective crime management, ultimately contributing to safer neighborhoods.

# •System Architecture

The system architecture is divided into three major components:

**Client Interface (CLI):**

o Function: The user interacts with the application through a Command-Line Interface (CLI), entering inputs like crime reports, crime statuses, and safety tips.

* Features:
  + Input options for crime report submission, safety tip access, and viewing crime report statuses.
  + Display of crime reports and safety tips retrieved from the database. o Technologies: Python-based CLI, utilizing standard input/output methods.

**Backend Layer (Python Application):**

* Function: The backend serves as the business logic layer of the application. It handles interactions between the CLI and the database, including CRUD (Create, Read, Update, Delete) operations for crime reports and safety tips.
* Features:
  + Crime Reporting: Adding new crime reports, including fields such as title, description, location, and date.
  + Crime Report Status Update: Admin functionality to update the status of crime reports to either "Resolved" or

"Unresolved."

* + Safety Tip Management: Adding and retrieving safety tips.
  + Database Communication: Interacts with the database to store, retrieve, and modify crime and safety data. o Technologies:
  + Python (sqlite3 library for database interactions).
  + SQLite for a lightweight, self-contained database.

**Database Layer (SQLite Database)**:

* Function: The database is responsible for storing crime report data and safety tips, maintaining data integrity, and supporting efficient retrieval. o Schema:
  + CrimeReports Table: Stores crime details such as title, description, location, status, and timestamps.
  + SafetyTips Table: Stores safety tips, including titles and descriptions.
* Technologies: SQLite (embedded database engine).

# • Objectives

1. **Enhancing Community Safety**: The system streamlines crime reporting and management, allowing users to report incidents, track case statuses, and access safety tips to promote a safer environment.
2. **Efficient Crime Management**: Administrators can review, update, and resolve cases while leveraging secure SQLite storage for data integrity and future scalability, including authentication and analytics.

# Hardware & Software requirements

**Hardware Requirements:**

**Client System (for users to interact with the application):**

* **Processor**: Any modern processor (Intel or AMD, 1.5 GHz or higher). o **Memory (RAM)**: 2 GB or higher.
* **Storage**: At least 50 MB of free space for the application and SQLite database files. o **Input/Output Devices**: o Keyboard (for data entry).
* Monitor (for CLI interface).

**Server System (if scaling or handling a large number of reports):**

* **Processor**: Multi-core processor (Intel i5 or better). o **Memory (RAM)**: 4 GB or more (depending on scale).
* **Storage**: 1 GB or more (as the database will grow with crime reports and safety tips). o **Network**: Internet connection (if scaling for online use or integrating with external services like email notifications).

**Software Requirements:**

**Operating System**:

* **Windows** (7 or higher), **Linux** (any modern distribution), or **macOS**.
* For the server-side (if using more advanced features), Ubuntu or CentOS is recommended for a production environment.

**Programming Language**:

* **Python 3.x**: Python serves as the primary backend language for handling business logic and database interactions.

**Database**:

* **SQLite**: A lightweight, serverless, self-contained SQL database that stores crime reports and safety tips. It is ideal for this application since it's easy to set up and manage. o If the system needs to scale, consider migrating to **MySQL** or **PostgreSQL**.

**Python Libraries**:

* **sqlite3** (comes with Python for interacting with SQLite databases). o **os**: To manage file and directory operations. o **datetime**: For handling timestamps for crime reports.

# • Conclusion

The **Crime Reporting and Safety Awareness App** provides a simple yet powerful platform for users to report crime incidents, track their statuses, and stay informed about safety tips. It empowers citizens to contribute to the safety of their community by creating an accessible and straightforward method for crime reporting and safety awareness.

By leveraging Python and SQLite, the app offers a lightweight, efficient, and scalable solution for both small-scale and future expanded use cases. The design is intentionally simple, using a Command-Line Interface (CLI) to interact with the system, making it accessible to users without requiring any technical expertise. The app's ability to store and manage data securely through a relational database ensures that valuable crime and safety information is kept intact and easily retrievable.

Key benefits of the system include:

* **Community Empowerment**: By allowing users to report crimes and share safety tips, the app strengthens the collective responsibility toward crime prevention.
* **Real-Time Reporting and Updates**: Administrators can quickly resolve or monitor reported crimes, improving the efficiency of crime management and response.
* **Safety Awareness**: Through a repository of safety tips, users are equipped with the knowledge to protect themselves and others from potential crimes.
* **Scalability**: The architecture is designed to support future growth, with potential to expand to web or mobile platforms, integrate notifications, or add advanced analytics.

# References :

**Flask Documentation**: Flask, https://flask.palletsprojects.com/

**Django Documentation**: Django, <https://www.djangoproject.com/>

**SQLite Documentation**: SQLite, https://www.sqlite.org/docs.html

**MySQL Documentation**: MySQL, <https://dev.mysql.com/doc/>

**Google Maps API Documentation**: Google, https://developers.google.com/maps/documentation

**Bootstrap Documentation**: Bootstrap, <https://getbootstrap.com/>