



CRIME RECORD MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

SEVENHILLSVASA S (2303811724321100)

in partial fulfillment of requirements for the award of the course

CGB1221-DATABASE MANAGEMENT SYSTEMS

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE-2025

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**CRIME RECORD MANAGEMENT SYSTEM**” is the bonafide work of **SEVENHILLSVASA S (2303811724321100)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T. AVUDAIAPPAN, M.E., Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.S. GEETHA, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on ...**04.06.2025**.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**CRIME RECORD MANAGEMENT SYSTEM**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS**.

Signature

SEVENHILLSVASA S

Place: Samayapuram

Date: 04.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E., Ph.D.**, Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S. GEETHA, M.E.**, Department of **ARTIFICIAL INTELLIGENCE**, for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Crime Record Management System is designed to efficiently manage crime case data using an SQLite database, ensuring reliable storage and retrieval. It enforces data integrity through primary keys and unique constraints to avoid duplicate records. The system supports core CRUD operations, enabling users to add, update, search, and delete crime cases seamlessly. The user-friendly interface, developed with Gradio, allows law enforcement personnel to interact with the system without technical difficulties. Additionally, the system offers data visualization via status distribution charts, providing quick insights into case progress. By digitizing crime records, the system improves data accessibility, accuracy, and operational efficiency compared to traditional manual methods. The modular design facilitates future enhancements such as advanced reporting, security features, and role-based access control. This system aims to support law enforcement agencies in effective crime management and decision-making processes.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5: BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The Crime Record Management System is designed to efficiently manage crime case data using an SQLite database, ensuring reliable storage and retrieval. It enforces data integrity through primary keys and unique constraints to avoid duplicate records. The system supports core CRUD operations, enabling users to add, update, search, and delete crime cases seamlessly. The user-friendly interface, developed with Gradio, allows law enforcement personnel to interact with the system without technical difficulties. Additionally, the system offers data visualization via status distribution charts, providing quick insights into case progress. By digitizing crime records, the system improves data accessibility, accuracy, and operational efficiency compared to traditional manual methods. The modular design facilitates future enhancements such as advanced reporting, security features, and role-based access control. This system aims to support law enforcement agencies in effective crime management and decision-making processes.</p>	<p>PO1 - 3 PO2 - 3 PO3 - 3 PO4 - 2 PO5 - 2 PO6 - 2 PO7 - 1 PO8 - 2 PO9 - 1 PO10 - 2 PO11 - 2 PO12 - 2</p>	<p>PSO1 - 3 PSO2 - 2</p>

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 SQL AND DATABASE CONCEPTS	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	3
3	MODULE DESCRIPTION	4
	3.1 DATA MANAGEMENT	4
	3.2 CASE OPERATIONS	4
	3.3 SEARCH AND FILTER	5
	3.4 DATA VISUALIZATION	5
4	CONCLUSION & FUTURE SCOPE	6
	APPENDIX A SOURCE CODE	7
	APPENDIX B SCREENSHOTS	12
	REFERENCES	14

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The primary objective of the Crime Record Management System is to develop a secure and efficient platform for storing and managing crime case data. It aims to simplify case entry, update, and retrieval through an intuitive interface. The system ensures data integrity by preventing duplicate records. It provides advanced search and filtering capabilities to quickly locate cases. Additionally, the system offers visual analytics for monitoring case status trends. Overall, it seeks to improve operational efficiency and support law enforcement decision-making.

1.2 OVERVIEW

The Crime Record Management System is developed to streamline the management of crime case data by digitizing records into a centralized database. It offers an intuitive interface that simplifies case entry, updating, searching, and deletion, making it accessible for non-technical users. The system ensures data accuracy and integrity through unique constraints and primary keys. Additionally, visual analytics provide quick insights into case statuses, aiding law enforcement agencies in efficient decision-making and resource allocation.

- Secure storage ensuring data integrity and accuracy.
- Efficient case addition, update, search, and deletion.
- Interactive interface designed for non-technical users.
- Visual charts displaying case status statistics

1.3 SQL AND DATABASE CONCEPTS

The project uses SQLite for lightweight and efficient data storage. Crime case details are stored in structured tables with constraints. Primary key enforces unique case identification for data integrity. SQL queries perform insert, update, delete, and fetch operations.

- UNIQUE constraint prevents duplicate cases by title and location.
- Prepared statements protect database from SQL injection attacks.
- Data retrieval uses SELECT queries filtered by case ID.
- Database connection ensures resource release after each operation.

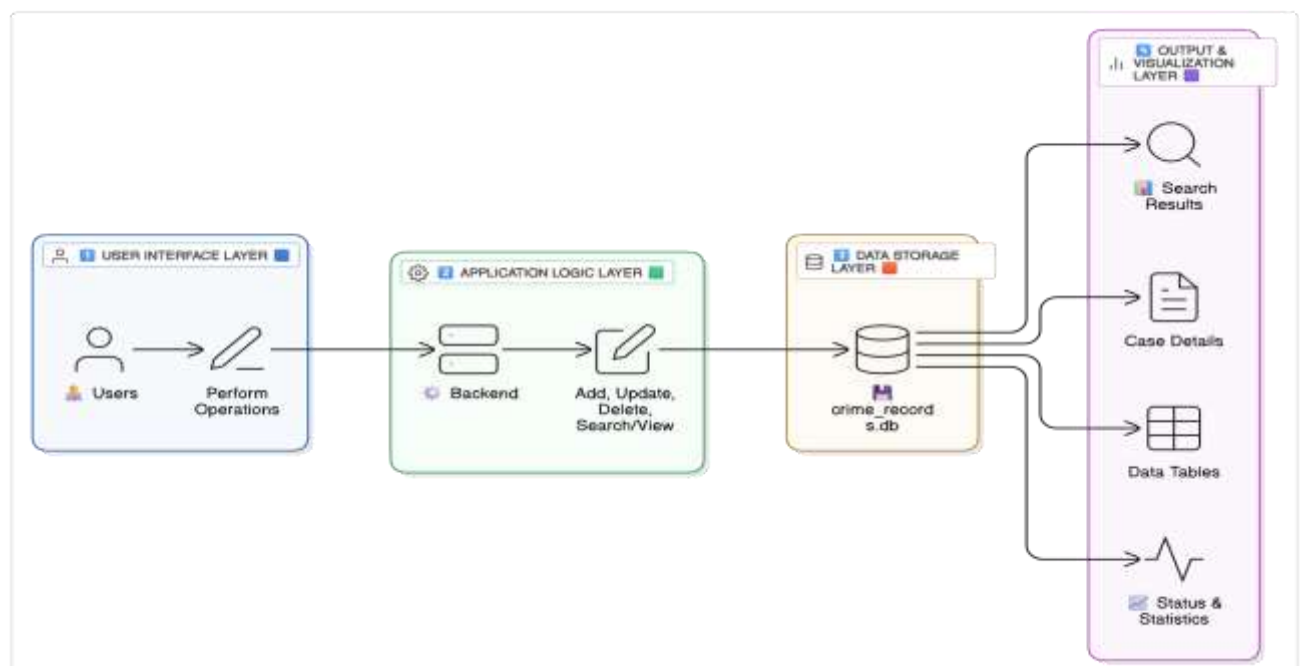
CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

The project aims to develop a digital Crime Record Management System to replace traditional manual record-keeping. It focuses on creating a reliable SQLite database with strict constraints to ensure data integrity and prevent duplicates. The system facilitates efficient management of crime cases through complete CRUD operations, allowing users to add, update, delete, and search records easily. An interactive user interface built using Gradio provides seamless access and manipulation of data. Additionally, integrated data visualization helps in monitoring case status trends, enabling better analysis and decision-making for law enforcement agencies. Security measures such as input validation and error handling are incorporated to maintain system robustness. The design also prioritizes user-friendly navigation for ease of use by personnel. Overall, the project enhances crime data accessibility and operational efficiency.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 DATA MANAGEMENT

This module initializes and manages the SQLite database connection for crime records. It defines the table schema with appropriate data types and constraints to ensure data accuracy and integrity. The module handles secure transaction commits and safe connection closure after each operation. It is crucial for reliable and persistent data storage.

Key Methods:

- `connect_db()`: Establishes and returns a database connection.
- `setup_db()`: Creates the `crime_cases` table with schema and constraints.

3.2 CASE OPERATION

This module implements CRUD operations for managing crime case data efficiently. It performs insertion with duplicate checks, updates existing records, fetches case details by case ID, and deletes records securely. The module ensures consistent and accurate manipulation of crime case information. It acts as the primary interface between the application and the database.

Key Methods:

- `insert_case()`: Inserts new crime case with duplicate prevention.
- `update_case()`: Updates existing case data based on case ID.
- `fetch_case_by_id()`: Retrieves case details for a given case ID.
- `delete_case()`: Deletes a case record by case ID.

3.3 SEARCH AND FILTER

This module enables keyword-based searching across multiple case fields like ID, title, location, and officer name. It uses SQL LIKE queries for flexible, case-insensitive matching to quickly locate relevant records. The module improves user experience by efficiently filtering large datasets. It returns results formatted for display or further processing.

Key Methods:

- `search_cases()`: Performs keyword search across all case fields.
- `view_all()`: Retrieves all crime cases from the database.

3.4 DATA VISUALIZATION

This module generates bar charts to visualize the distribution of crime cases by status using Matplotlib. It aggregates cases into categories such as Open, Closed, and Pending to show trends and workload. The visualization supports monitoring and decision-making processes. It safely handles empty datasets to prevent errors.

Key Methods:

- `plot_status_chart()`: Creates a bar chart of case status distribution.

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

This project efficiently developed a Crime Record Management System using Gradio and SQLite in Python. It offers user-friendly modules for adding, updating, searching, and deleting crime cases. Visual statistics and data export features enhance utility for law enforcement workflows. The modular design ensures future scalability and easy integration of advanced functionalities.

- Implement user authentication with role-based access control.
- Enable automated alerts for pending or critical case updates.
- Create a web/mobile version for remote system accessibility.
- Integrate facial recognition for suspect identification and tracking.
- Add location-based mapping using crime heatmaps and GIS.

APPENDIX A – SOURCE CODE

```
import gradio as gr
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

# ----- DB Setup -----
def connect_db():
    return sqlite3.connect("crime_records.db")

def setup_db():
    conn = connect_db()
    c = conn.cursor()
    c.execute("""
        CREATE TABLE IF NOT EXISTS crime_cases (
            case_id TEXT PRIMARY KEY,
            title TEXT,
            location TEXT,
            officer TEXT,
            status TEXT,
            description TEXT,
            UNIQUE(title, location)
        );
    """)
    conn.commit()
    conn.close()

setup_db()

# ----- DB Operations -----
def insert_case(case_id, title, location, officer, status, description):
    try:
        conn = connect_db()
        c = conn.cursor()
        c.execute("""
            INSERT INTO crime_cases (case_id, title, location, officer, status,

```

```

description)
    VALUES (?, ?, ?, ?, ?, ?);
    """', (case_id, title, location, officer, status, description))
    conn.commit()
    return "✅ Case added successfully!"
except sqlite3.IntegrityError:
    return "❌ Case already exists (duplicate ID or title+location)."
finally:
    conn.close()

def fetch_case_by_id(case_id):
    conn = connect_db()
    df = pd.read_sql_query("SELECT * FROM crime_cases WHERE
case_id = ?", conn, params=(case_id,))
    conn.close()
    return df.to_dict('records')[0] if not df.empty else None

def update_case(case_id, title, location, officer, status, description):
    conn = connect_db()
    c = conn.cursor()
    c.execute("""
        UPDATE crime_cases SET title = ?, location = ?, officer = ?, status =
?, description = ?
        WHERE case_id = ?;
    """, (title, location, officer, status, description, case_id))
    conn.commit()
    conn.close()
    return "✅ Case updated successfully!"

def delete_case(case_id):
    conn = connect_db()
    c = conn.cursor()
    c.execute("DELETE FROM crime_cases WHERE case_id = ?",
(case_id,))
    conn.commit()
    conn.close()
    return "🗑️ Case deleted successfully!"

```

```

def view_all():
    conn = connect_db()
    df = pd.read_sql_query("SELECT * FROM crime_cases", conn)
    conn.close()
    return df

def search_cases(keyword):
    conn = connect_db()
    like_kw = f"%{keyword}%"
    query = """
        SELECT * FROM crime_cases
        WHERE case_id LIKE ? OR title LIKE ? OR location LIKE ? OR
officer LIKE ? OR status LIKE ? OR description LIKE ?;
    """
    df = pd.read_sql_query(query, conn, params=[like_kw]*6)
    conn.close()
    return df

def plot_status_chart():
    df = view_all()
    if df.empty:
        return plt.figure()
    counts = df["status"].value_counts()
    fig, ax = plt.subplots()
    ax.bar(counts.index, counts.values, color="cornflowerblue")
    ax.set_title("Case Status Distribution")
    return fig

# ----- Gradio UI -----
with gr.Blocks(theme=gr.themes.Base(primary_hue="blue")) as app:
    gr.Markdown("## 🚔 Crime Record Management System")

    # Separate button for full DB view
    with gr.Row():
        view_db_btn = gr.Button("📁 View Database Table and Data",
size="sm")

```















```
Database", visible=False)
```

```
gr.Markdown("### 🌟 Quick Access")
nav = gr.Radio(
    ["➕ Add Case", "↻ Update Case", "🗑 Delete Case", "🔍 Search
Case", "📄 View All", "📊 Stats"],
Case", interactive=True
```

```
# Shared Inputs
case_id = gr.Text(label="Case ID 🆔 ", visible=True)
title = gr.Text(label="Title 📄 ", visible=False)
location = gr.Text(label="Location 📍 ", visible=False)
officer = gr.Text(label="Officer 👮 ", visible=False)
status = gr.Dropdown(["Open", "Closed", "Pending"], label="Status
🚩 ", visible=False)
lines=3, visible=False)
result = gr.Textbox(label="📄 Result", lines=1, interactive=False)
```

```
submit_btn = gr.Button("✅ Submit", visible=False)
update_btn = gr.Button("✏ Update", visible=False)
fetch_btn = gr.Button("🔍 Fetch Case", visible=False)
delete_btn = gr.Button("🗑 Delete", visible=False)
search_input = gr.Text(label="Search Keyword 🔍 ", visible=False)
search_btn = gr.Button("Search", visible=False)
data_table = gr.Dataframe(label="📄 Data Table", visible=False)
chart_output = gr.Plot(visible=False)
# Logic to switch UI components
def change_tab(tab):
    return {
        case_id: gr.update(visible=tab in ["➕ Add Case", "↻ Update
Case", "🗑 Delete Case"]),
Add Case", "↻ Update Case"]),
["➕ Add Case", "↻ Update Case"]),
gr.update(visible=tab in ["➕ Add Case", "↻ Update Case"]),
```

```

status: gr.update(visible=tab in [" Add Case", " Update Case"]),
description: gr.update(visible=tab in [" Add Case", " Update Case"]),
update_btn: gr.update(visible=tab == " Update Case"),
fetch_btn: gr.update(visible=tab == " Update Case"),
delete_btn: gr.update(visible=tab == " Delete Case"),
search_input: gr.update(visible=tab == " Search Case"),
search_btn: gr.update(visible=tab == " Search Case"),
data_table: gr.update(visible=tab in [" Search Case", " View All"], value=view_all() if tab == " View All" else None),
chart_output: gr.update(visible=tab == " Stats",
value=plot_status_chart() if tab == " Stats" else None),
nav.change(fn=change_tab, inputs=nav, outputs=[
    case_id, title, location, officer, status, description,
    submit_btn, update_btn, fetch_btn, delete_btn,
    search_input, search_btn, data_table, chart_output
])
# Action handlers
submit_btn.click(insert_case, inputs=[case_id, title, location, officer,
status, description], outputs=result)
fetch_btn.click(lambda cid: fetch_case_by_id(cid) or {},
inputs=case_id,
outputs=[title, location, officer, status, description])
update_btn.click(update_case, inputs=[case_id, title, location, officer,
status, description], outputs=result)
delete_btn.click(delete_case, inputs=case_id, outputs=result)
search_btn.click(search_cases, inputs=search_input,
outputs=data_table)
view_db_btn.click(fn=view_all, inputs=[],
outputs=view_db_table).then(

)

app.launch()

```

APPENDIX B – SCREENSHOTS

Crime Record Management System

[+ Add Case](#) [Update Case](#) [Delete Case](#) [Search Case](#) [View All Cases](#) [Statistics](#)

Case ID **Title**

ROI ROBBERY

Location **Officer**

TRICHY RAHAGAVAN

Status **Description**

Open NOT YET GATHERED

[Add Case](#)

Status

Case added successfully!

Crime Record Management System

[+ Add Case](#) [Update Case](#) [Delete Case](#) [Search Case](#) [View All Cases](#) [Statistics](#)

Search Keyword

1

[Search](#)

Search Results

case_id	title	location	officer	status	description
C001	Burglary at Mall	Downtown	Officer Smith	Open	Suspects broke into the mall after hours.
C010	Fraudulent Scheme	Financial District	Officer Garcia	Closed	Several suspects arrested for Ponzi scheme.
ROI	ROBBERY	TRICHY	RAHAGAVAN	Open	NOT YET GATHERED

Crime Record Management System

+ Add Case

✎ Update Case

🗑 Delete Case

🔍 Search Case

📄 View All Cases

📊 Statistics

📄 Load All Cases

📄 All Crime Cases

case_id	title	location	officer	status	description
C001	Burglary at Mall	Downtown	Officer Smith	Open	Suspects broke into the mall after hours.
C002	Vandalism in Park	Central Park	Officer Johnson	Pending	Graffiti found on several park benches.
C003	Hit and Run	5th Avenue	Officer Lee	Closed	Victim was hit by a vehicle which fled the scene.
C004	Robbery at ATM	City Bank	Officer Davis	Open	Robber stole money from ATM using explosives.
C005	Cybercrime Fraud	Internet	Officer Patel	Pending	Multiple victims reported unauthorized transactions.
C006	Assault Case	West Side	Officer Brown	Open	Victim was assaulted near the subway station.
C007	Missing Person	North District	Officer Wilson	Closed	Person found safe after 3 days of search.
C008	Drug Trafficking	Harbor Area	Officer Taylor	Open	Large shipment of illegal drugs seized.
C009	Arson at Warehouse	Industrial Zone	Officer Martinez	Pending	Warehouse caught fire under suspicious circumstances.
C010	Fraudulent Scheme	Financial District	Officer Garcia	Closed	Several suspects arrested for Ponzi scheme.
R01	ROBBERY	TRICHY	RAHAGAVAN	Open	NOT YET GATHERED



REFERENCES

1. **Gradio:** Build Machine Learning Web Apps in Python. Retrieved from <https://www.gradio.app/docs/>.
2. **Python Software Foundation:** Python 3 Documentation. Retrieved from <https://docs.python.org/3/>.
3. **Python Software Foundation:** Python library Documentation. Retrieved from <https://docs.python.org/3/>.
4. **SQLite Consortium:** SQLite Official Documentation. Retrieved from <https://www.sqlite.org/docs.html>.