

RV College of Engineering®
(Autonomous Institution Affiliated to VTU, Belagavi)

**Department of Information Science and
Engineering**



**“INTRUSION DETECTION AND SERVILLANCE
SYSTEM”**

Project Report

On

DESIGN THINKING LAB (MIT401L)

Sem: 1

Submitted by

Nikitha Santiago J-RVCE24MSE010
Thripurasri S -RVCE24MSE012

MTech (Software Engineering)

Submitted to

Prof. Rashmi R
Assistant Professor
Dept. of ISE

2024-25

(ODD Semester)

Table of Contents

Chapter No	Contents	Page no
1	Intrusion Detection and Surveillance System	1 - 2
	1.1 Introduction	1
	1.2 Objectives	2
2	Tools and Technology Used	3
3	System Architecture	4
	3.1 Methodology	4 - 5
	3.2 System design	6
	3.3 Sequence Diagram	7
4	Implementation of Five Phases of Design Thinking	8 - 9
	4.1 Empathize Phase	8
	4.2 Define Phase	8-9
	4.3 Ideate Phase	9
	4.4 Prototype Phase	9
	4.5 Test Phase	9
5	Results	10 - 17
	5.1 Snap Shots	11 - 14
6	INFERENCES	15
7	Future Scope	16
8	Conclusion	17

CHAPTER 1

Intrusion Detection and Surveillance System

1.1 Introduction

In today's rapidly evolving digital and physical environments, ensuring security has become a top priority. Traditional surveillance systems often require constant human monitoring and fail to provide immediate responses to unauthorized access or suspicious activity. To address these limitations, this project presents a Web-Based Intrusion Detection and Surveillance System that utilizes artificial intelligence and real-time communication technologies to enhance security through automation, accuracy, and instant alerting.

This system allows users to register securely, after which a live video feed is activated upon login. The system continuously captures frames from the webcam feed and processes them using YOLOv8, a real-time object detection model, to detect the presence of humans. If a person is detected who is not part of the authorized access, the system immediately sends an alert via SMS (using Twilio) and email (using SMTP) to the registered user.

Built using Flask as the backend, WebRTC for video streaming, and SQLite for secure data storage, this system ensures a reliable and responsive user experience. All user credentials are securely handled using hashed password storage, and no cloud-based data sharing is involved, ensuring data privacy and security.

This project aims to create a simple, effective, and automated security solution that can be used for homes, small offices, and personal spaces, offering real-time surveillance with intelligent detection and alerting mechanisms.

1.2 Objectives

- 1 Real-Time Intrusion Detection: Detects unauthorized access using YOLOv8 and face recognition.
- 2 Face Recognition-Based Access Control: Allows access only to registered users; sends alerts for unknown faces.

Intrusion Detection and Surveillance System

- 3** Instant Alerts via SMS & Email: Sends SMS (Twilio) and Email (SMTP) when an intruder is detected.
- 4** Live Video Streaming & Monitoring: Uses WebRTC to provide a real-time webcam feed on the web app.
- 5** Secure Data Storage: Stores user data, intruder logs, and captured images securely in a database.
- 6** User-Friendly Web Dashboard: Simple UI to view live feeds, intrusions, and manage settings.
- 7** AI-Powered Object Detection: Uses YOLOv8 to detect humans and suspicious objects in the frame.

CHAPTER 2

Tools and Technology Used

This project employs the following key technologies:

Backend Technologies

- Flask (Python Web Framework) – Manages authentication, video streaming, and API requests.
- SQL Alchemy (Database ORM) – Stores user data and intrusion logs.
- Twilio API – Sends SMS alerts to users.
- SMTP (Email Protocol) – Sends email notifications upon detection.

Machine Learning & Computer Vision

- YOLOv8 (You Only Look Once) – Real-time object detection.
- Face Recognition Library – Matches registered users against live camera feed.

Frontend Technologies

- HTML, CSS, JavaScript, Bootstrap – Provides an interactive user interface.
- WebRTC & JavaScript – Streams real-time video.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 METHODOLOGY

The Intrusion Detection and Surveillance System follows a structured methodology to ensure real-time monitoring, object detection, and automated alerts. Below is a step-by-step breakdown of the implemented methodology, focusing on image capture during registration, real-time video streaming, and alert triggering.

- 1 User Registration with Image Capture:** The first step in the system is the user registration process, where users create an account and provide their credentials. During registration, the system captures the user's image through a webcam. This captured image is stored in the database and is used for logging purposes. The system ensures that the user provides mandatory details, including their username, password, email, and phone number. The user is required to register via a web-based form, and once the registration is completed, their data is securely stored in an SQLite database using SQLAlchemy. The image capture process is done using WebRTC, which enables the system to access the webcam from the web browser. The image is encoded and saved in the database, ensuring that every registered user has a unique profile image associated with their account. The registration process ensures that only verified users can log in to the system, adding an extra layer of security.
- 2 Live Feed Processing & Object Detection:** Once a user logs in, the real-time video streaming begins automatically. The system activates the webcam and streams a live video feed through WebRTC, which allows seamless communication between the camera and the web application. The live feed is continuously monitored, and each frame is processed using computer vision techniques. The system employs YOLOv8 (You Only Look Once), an advanced object detection model, to analyze the video feed and detect humans, objects, and potential intruders. YOLOv8 processes each captured frame and determines whether any unauthorized movement is occurring within the monitored area. If a human figure or an unrecognized object is detected in the frame, the system logs the event and prepares for

further action. The integration of OpenCV allows real-time frame extraction, which ensures that the system is continuously analysing the video feed. The object detection mechanism is optimized for fast processing to ensure minimal delay in recognizing potential security threats.

- 3 Intrusion Alert System (SMS & Email Alerts):** If an unknown person or object is detected in the monitored area, the system immediately triggers an alert mechanism. The intrusion detection process involves capturing an image of the detected subject, logging the event in the database, and sending notifications to the registered user. The system is designed to notify users via two communication channels: SMS and Email. For SMS alerts, the system utilizes Twilio API, which allows sending text messages to the user's registered phone number. As soon as an intrusion is detected, Twilio sends a message notifying the user of the potential breach, including relevant details such as the date, time, and type of detection. For email alerts, the system uses SMTP (Simple Mail Transfer Protocol) to send an email notification to the user. The email includes a captured image of the intruder, providing evidence of the intrusion. This ensures that users are instantly informed about security breaches, even if they are not actively monitoring the system. Both SMS and email alerts are logged in the database, allowing users to review past incidents and take necessary actions.
- 4 Data Storage & Security:** To ensure data security and integrity, the system employs SQLAlchemy and SQLite for database management. Every user registration, captured image, intrusion log, and system event is securely stored in the database. The system ensures that sensitive user information, such as passwords, is hashed using bcrypt encryption, preventing unauthorized access. Captured intruder images are also stored securely, ensuring that they can be reviewed later if needed. The system maintains timestamped logs of all security events, allowing administrators or users to track past incidents efficiently. Data security is a critical component of the methodology, and the system ensures that only authenticated users can access stored information. No raw passwords or sensitive user data are stored in plaintext format.

3.2 SYSTEM DESIGN:

The Fig 3.1 represents intrusion detection and surveillance system begins with user registration, where the system captures the user's image and personal details. After successful login verification, users access the dashboard, which streams a live video feed from the webcam. The system continuously analyses frames, comparing them with the stored image. If a match is found, monitoring continues normally. However, if an unrecognized person is detected, the system triggers an alert and immediately sends an SMS and email notification with the captured intruder's image. This process ensures real-time security monitoring, allowing users to detect unauthorized access instantly and enhance surveillance efficiency.

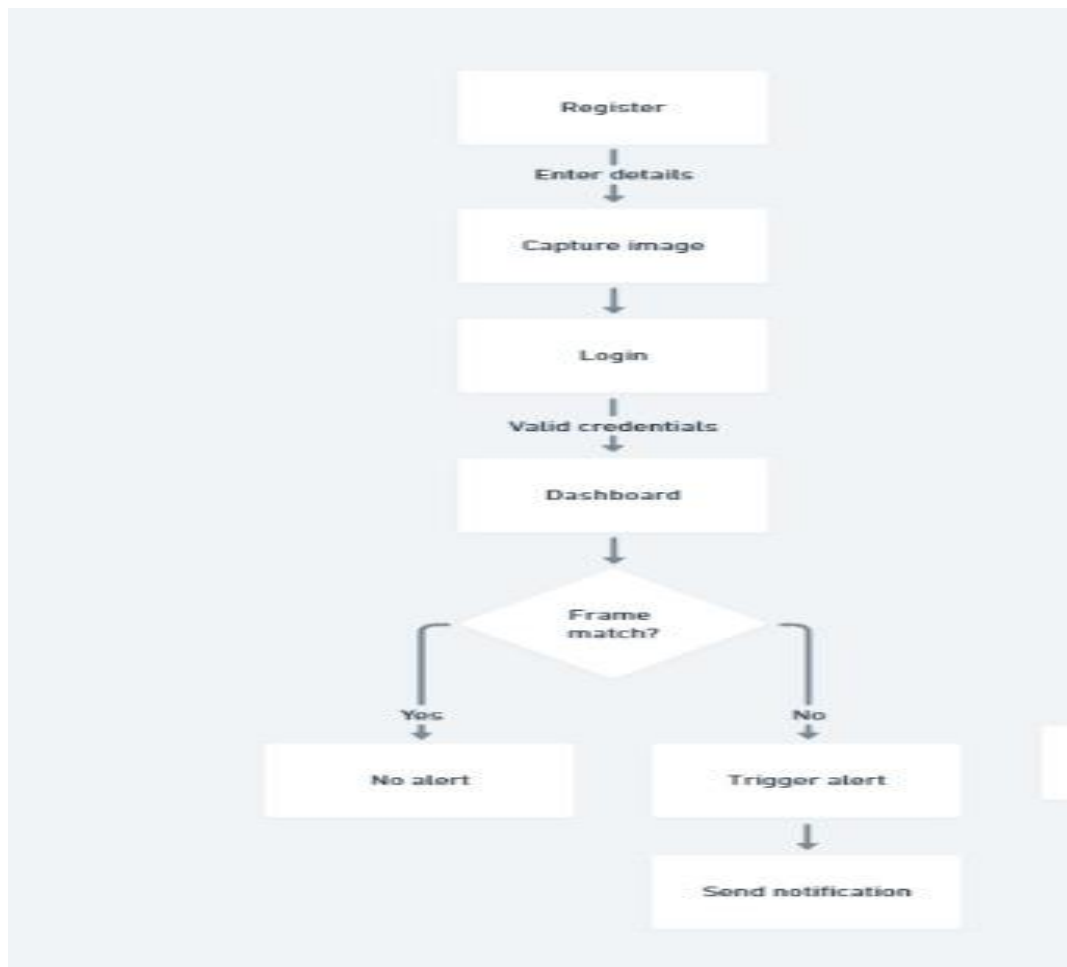


Fig 3.1: Flowchart

3.3 SEQUENCE DAIGRAM

The Fig 3.2 represents sequence diagram represents the interaction between the user, Flask backend, SQLAlchemy database, Twilio API, and SMTP server in the Intrusion Detection and Surveillance System. The user registers and logs in, triggering Flask to store and fetch user data using SQLAlchemy. When an intrusion is detected, Flask retrieves the user's phone number and email from the database. Twilio sends SMS alerts, while the SMTP server sends email notifications. This structured process ensures secure authentication, intrusion detection, and real-time alerts, allowing users to receive instant notifications via SMS and email when unauthorized access is detected.

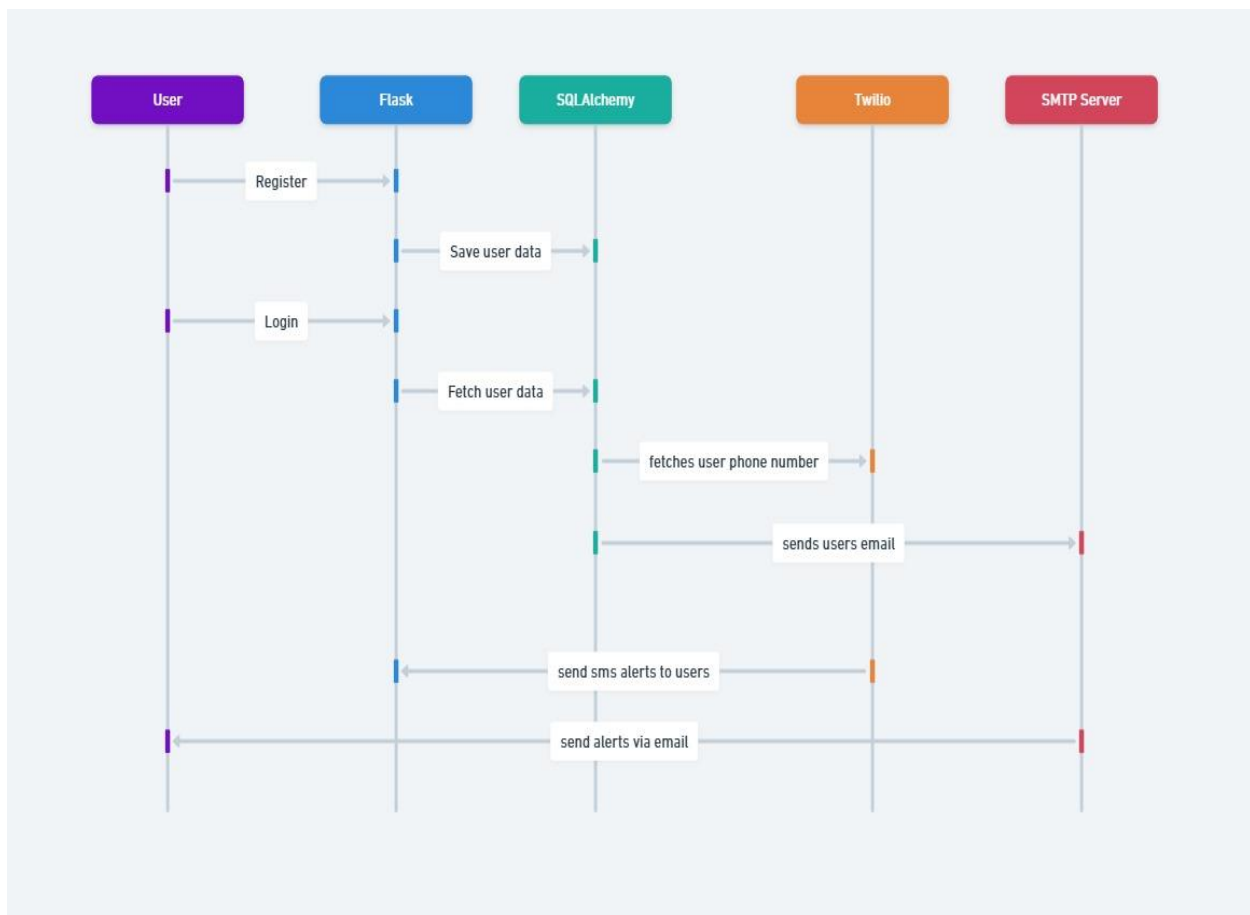


Fig 3.2: Sequence Diagram

CHAPTER 4

Implementation of the Five Phases of Design Thinking

4.1 Empathize Phase:

In the Empathize phase, we analyzed the security challenges faced by users and identified the key pain points that needed to be addressed.

1 Pain Points Analysis:

- Manual surveillance requires continuous human monitoring, which is inefficient.
- Lack of real-time automated intrusion detection leads to delayed security responses.
- No instant notification system to alert users when unauthorized access is detected.

2 Feature Identification:

- To solve these challenges, we designed the system with:
- Live webcam streaming for real-time monitoring
- YOLOv8-based object detection to identify potential intrusions
- Automated SMS & email alerts for immediate notification
- A web dashboard for live surveillance and intrusion logs

3 User Personas Creation:

- Home Users: Need a simple security solution for monitoring personal spaces.
- Small Businesses: Require an automated surveillance system to detect intrusions.
- Security Considerations:
- Ensured secure data handling, where user credentials and security logs are stored locally for privacy.
- Encrypted password storage to protect login information.

4.2 DEFINE PHASE

- 1 **Problem:** Users need automated security monitoring and instant alerts without human supervision.

- 2 **Solution Implemented:**

- Flask backend for managing system operations.
- YOLOv8 for object detection in video frames.
- WebRTC for real-time video streaming on the dashboard.
- Twilio SMS alerts & SMTP email notifications when an intrusion is detected.
- SQLite database for storing logs and user data.

4.3 Ideate Phase:

Initially considered **motion detection**, but it led to **false alerts** (e.g., light changes).

- Switched to YOLOv8 for accurate object recognition (specifically for human detection).
- Chose Flask for backend development due to its lightweight nature and flexibility.
- Integrated Twilio for SMS alerts and SMTP for email notifications with intrusion images.
- Implemented WebRTC for seamless webcam streaming within the web application.
- Stored intruder logs and images in SQLite for future reference.

4.4 Prototype Phase:

Developed Flask-based backend for handling video streaming, detection, and alerts.

- Integrated YOLOv8 to analyze frames from the webcam feed.
- Built a simple web dashboard to monitor live feeds and view alerts.
- Implemented WebRTC for real-time video streaming.
- Connected SQLite database to store user credentials and security logs.
- Configured Twilio & SMTP to send alerts instantly when an unauthorized person is detected.

4.5 Test Phase:

Tested YOLOv8's accuracy in detecting intrusions in live feeds.

- Optimized WebRTC's video performance to reduce streaming delays.
- Simulated intrusion scenarios to verify real-time alert triggering.
- Fixed notification delays to ensure instant SMS & email alerts.

CHAPTER 5

Results

The Intrusion Detection and Surveillance System was successfully implemented, focusing on real-time monitoring, user authentication, frame capturing, and enhanced security. Below are the key results achieved:

1. User Registration & Authentication:

- Users successfully registered with their details, including a captured webcam image.
- Login authentication was implemented using Flask and SQLite, ensuring only registered users could access the system.
- Passwords were securely hashed before storing them in the database for enhanced security.

2. Live Video Streaming & Frame Capture:

- WebRTC-based live video streaming worked efficiently, enabling real-time monitoring.
- The system successfully captured frames from the video feed at regular intervals.
- Captured frames were stored for reference and further processing if needed.

3. Intrusion Detection (Basic Frame Processing):

- YOLOv8 object detection was integrated for detecting human presence in the video feed.
- The system processed live frames, but no security log storage was implemented beyond capturing frames.
- The focus was on real-time monitoring and detection rather than maintaining an intrusion log.

4. Enhanced Security Measures:

- User passwords were hashed before being stored to prevent unauthorized access.

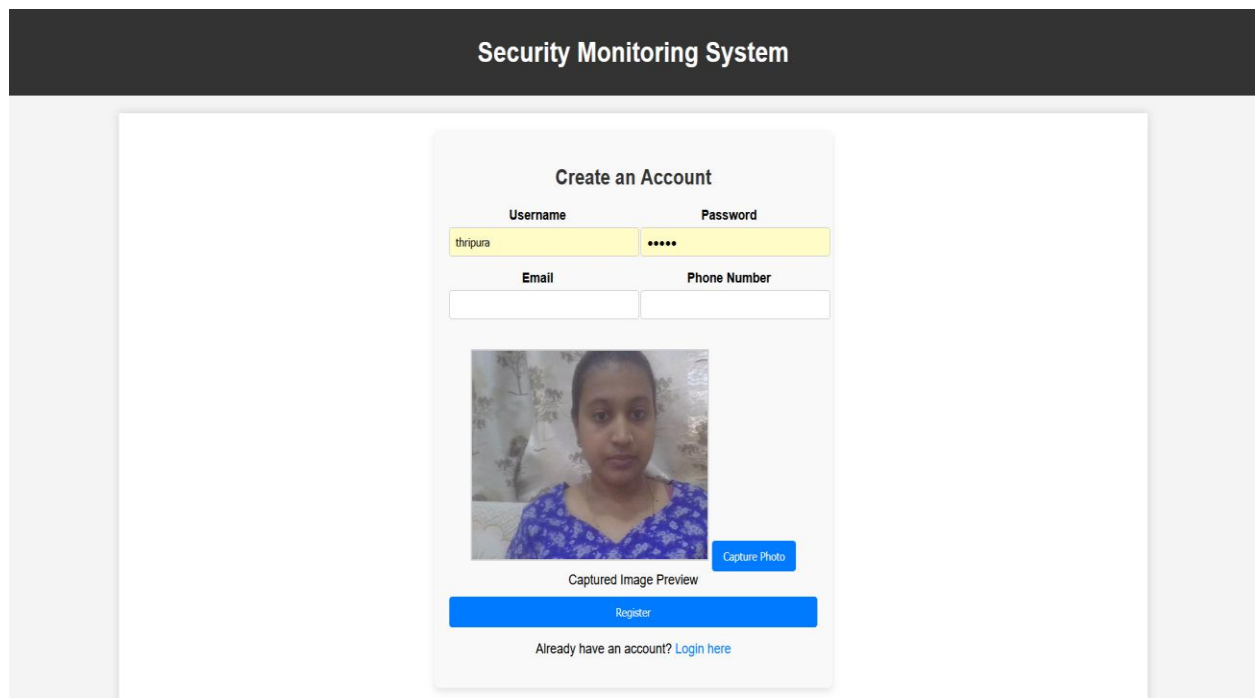
- No raw credentials were stored, ensuring data privacy and security.
- The system ran locally, ensuring no cloud-based security risks.

Overall Outcome:

The project successfully implemented secure user authentication, real-time video streaming, frame capturing, and enhanced data security using hashed passwords. While intrusion logs were not stored, the system effectively processed live frames and provided a secure monitoring environment.

5.1 Snapshots

The Fig 5.1 demonstrates the registration page of the web surveillance application. Which is the starting point for the users to get into the application. If the user is new to the application they can register and the most important thing is they should capture their image and because that frame will be used for face recognition and intruder alert process in later stages .



The screenshot displays the 'Security Monitoring System' registration interface. At the top, a dark header bar contains the title 'Security Monitoring System'. Below this, a white registration card is centered. The card has a title 'Create an Account' and four input fields: 'Username' (containing 'thripura'), 'Password' (masked with dots), 'Email', and 'Phone Number'. Below the inputs is a video feed showing a woman's face. A 'Capture Photo' button is positioned to the right of the video. Under the video is a 'Captured Image Preview' section. At the bottom of the card is a large blue 'Register' button and a link 'Already have an account? Login here'.

Fig 5.1:Registration page of the web application

Fig 5.2 shows the login page of the web application, after registering user can easily login into the application with there credentials.

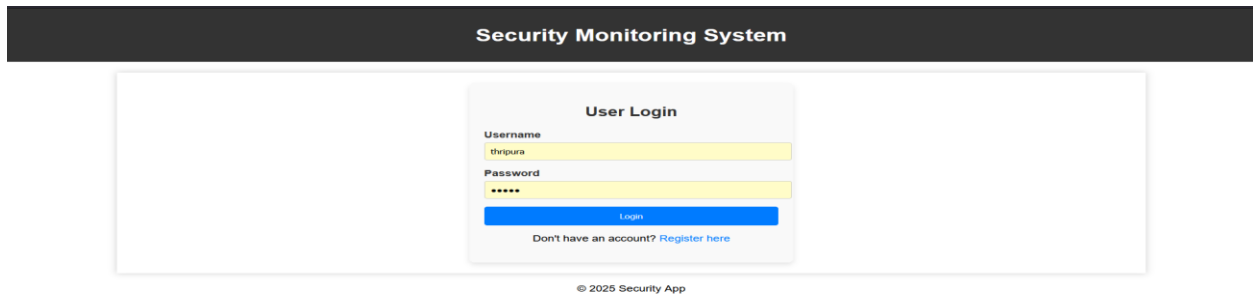


Fig 5.2: Login page of the web application

Fig 5.3 shows demonstrate the user dashboard, where user can see the live_vedio feed in real time .The images which was captured during registration will be stored in the database and frame generated from the live feed will be compared with frame in database . If the face is recognized, then intruder count will not rise . Till user logs out from the application the webcam and live_feed will be running which provides proper surveillance for the system.

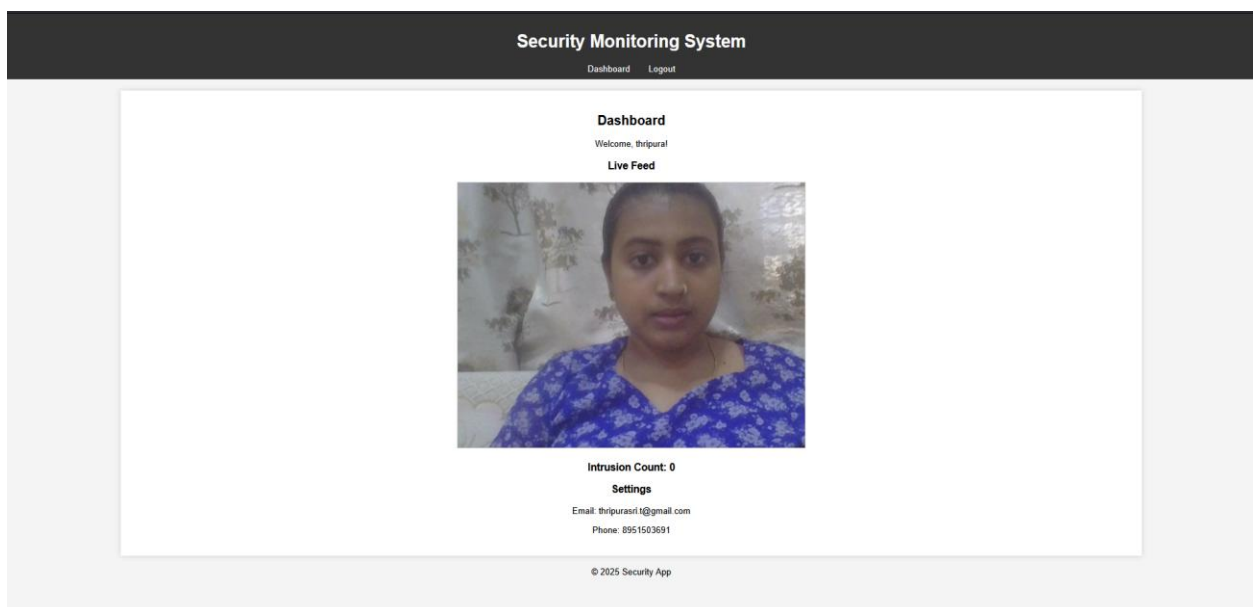


Fig 5.3: User dashboard

Intrusion Detection and Surveillance System

Fig 5.4 shows incrementation in intrusion count . This happens when the registered image is not same as the image / frame from the live feed . Hence this shows the intrusion detection.

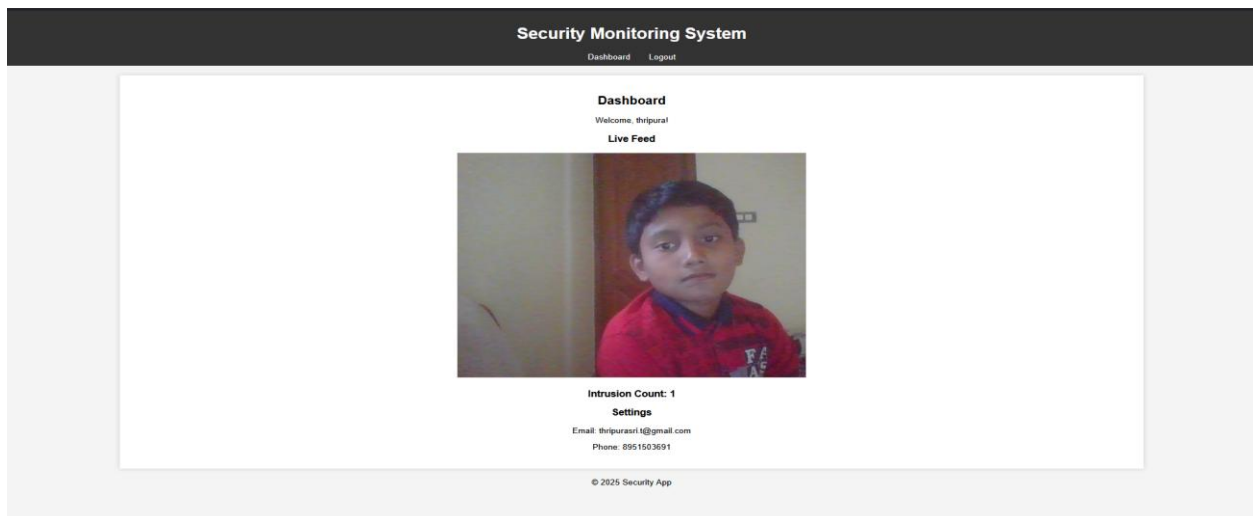


Fig 5.4: Intrusion detection

The Fig 5.5 shows the email sent to the registered user after detecting the intrusion.

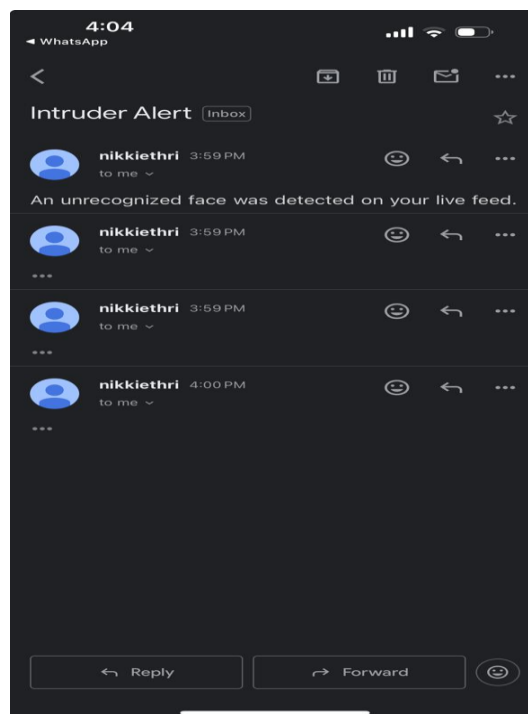


Fig 5.5:Email Alert Sent to Registered User

Fig 5.6 shows intrusion alerts through SMS using Twilio.

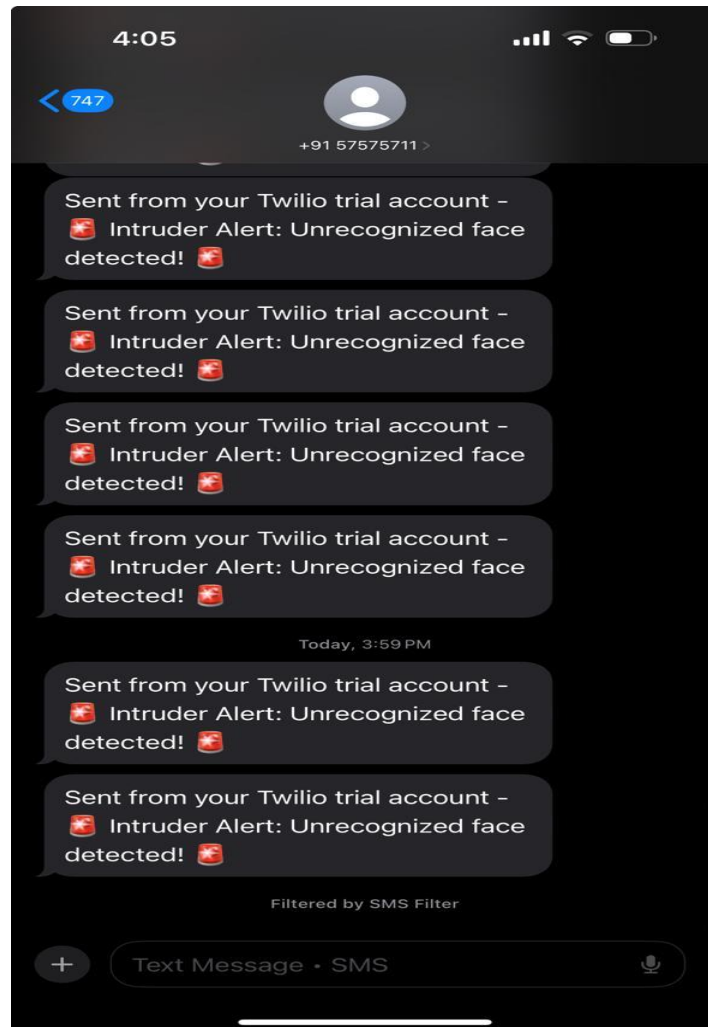


Fig 5.7: Intruder alerts through SMS

CHAPTER 6

Inferences

Inferences of the Intrusion Detection and Surveillance System:

- The system successfully implemented secure user authentication with hashed passwords for enhanced security.
- Live video streaming using WebRTC was effectively integrated, enabling real-time monitoring.
- Frames are captured from the live video feed, ensuring surveillance data is stored for analysis.
- YOLOv8 object detection was implemented, allowing accurate recognition of human presence.
- Face recognition was not implemented, meaning the system does not differentiate between registered and unregistered individuals.
- Structured intrusion logs were not maintained, limiting the ability to review past detection events.
- Twilio was successfully integrated for sending SMS notifications upon detecting an intruder.
- SMTP email alerts were implemented, sending real-time notifications with captured images of intruders.
- The system ensures real-time intrusion detection and automated alerts, enhancing security response time.
- All data is stored locally, preventing cloud-based vulnerabilities and ensuring user privacy.
- The project successfully met its core objectives of monitoring, detection, and alerting while maintaining security and reliability.

CHAPTER 7

Future Scope

Future Scope of the Intrusion Detection and Surveillance System

- Face Recognition for Enhanced Security – Implementing a face recognition system to differentiate between registered users and intruders for more precise intrusion detection.
- Intrusion Log Maintenance – Storing detection events with timestamps in a structured log, allowing users to review past security incidents.
- Cloud Storage Integration – Providing an option to store detected frames and logs on the cloud for remote access and backup.
- Mobile App Integration – Developing a mobile application to allow users to monitor live feeds and receive alerts on smartphones.
- Multiple Camera Support – Expanding the system to handle multiple webcams for monitoring multiple areas simultaneously.
- Advanced Object Detection – Enhancing YOLOv8 integration to recognize specific objects or behaviours beyond human presence, such as weapons or suspicious activities.
- Customizable Alert Preferences – Allowing users to adjust alert frequency, choose preferred notification methods, and set sensitivity levels.
- Intruder Response Mechanisms – Adding features like automated alarms, lights, or security locks triggered upon intrusion detection.
- AI-based False Alarm Reduction – Implementing AI-based filtering techniques to reduce false alerts caused by environmental changes or non-threatening movements.
- Scalability for Larger Systems – Upgrading the system for enterprise or business-level surveillance with a centralized monitoring dashboard.

The future enhancements will improve system accuracy, expand usability, and provide more flexible security options, making it more efficient and adaptable for different security applications.

CHAPTER 8

CONCLUSION

The Intrusion Detection and Surveillance System successfully integrates real-time monitoring, object detection, and automated alerts, providing an efficient and secure surveillance solution. By utilizing Flask for backend processing, WebRTC for live streaming, YOLOv8 for object detection, and Twilio/SMTP for instant notifications, the system ensures seamless and accurate intrusion detection. The implementation of hashed password storage and local data handling enhances security and privacy, making the system reliable and resilient against unauthorized access.

This project effectively addresses the need for automated security monitoring, eliminating the limitations of manual surveillance. The instant alert mechanism allows users to respond to potential threats in real-time, enhancing security awareness. While the system successfully meets its core objectives, future enhancements such as face recognition, multi-camera support, and AI-based false alarm filtering can further improve accuracy and scalability.

Overall, the project provides a strong foundation for smart surveillance technology, offering a cost-effective, scalable, and efficient solution for real-time security monitoring.