**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

In today's fast-paced and dynamic environments such as schools, universities, workplaces, and public areas, individuals frequently misplace or lose personal belongings. At the same time, many people encounter found items with no clear way of returning them to their rightful owners. Traditional lost and found systems, such as manual logs or notice boards, often prove to be inefficient, time-consuming, and prone to human error. To address this issue, the Lost & Found Portal has been developed as a web-based solution that enables users to report lost or found items online. This system aims to streamline communication between the person who finds an item and the one who has lost it, increasing the likelihood of successful recoveries and improving overall user experience.

**1.2 NEED FOR THE STUDY**

The necessity for this study arises from the limitations of conventional methods of managing lost and found items. Without a centralized system, lost items are often not reported or are difficult to track, resulting in frustration for those who have lost valuables and inefficiency for those managing such systems. Additionally, in many organizations or institutions, no proper procedure exists for handling lost and found items systematically. This project addresses this gap by providing a digital platform that is accessible, organized, and efficient. Studying and developing this portal demonstrates how technology can be utilized to solve real-world organizational problems and enhance community support mechanisms.

**1.3 OVERVIEW OF THE PROJECT**

The Lost & Found Portal is designed as a full-stack web application developed using PHP for server-side scripting, MySQL for the relational database, and HTML/CSS with Bootstrap for the frontend interface. It is hosted locally using XAMPP, which provides the Apache server and MySQL environment for development and testing. Users can register with their email addresses, log in securely, and report lost or found items by filling out simple forms that include item details and optional image uploads. Once logged in, users access a personalized dashboard where they can view or manage the items they have reported. The system also includes user session handling, email-based uniqueness validation, and redirection based on login status. Overall, the project emphasizes usability, data integrity, and secure interaction.

**1.4**  **OBJECTIVES OF THE STUDY**

The primary objective of this study is to design and implement a secure and user-friendly online platform for reporting and managing lost and found items. The study seeks to reduce the time and effort involved in locating lost belongings, improve the accountability of found items, and ensure a structured database that can be accessed and managed easily. Other specific goals include enabling image-supported item submissions, preventing duplicate user accounts using unique email addresses, maintaining proper session control for security, and creating an intuitive user experience. Ultimately, the project demonstrates how technology can be harnessed to improve day-to-day processes and foster a sense of trust and responsibility within a community.

**CHAPTER 2**

**REVIEW OF LITERATURE**

* 1. **INTRODUCTION**

The literature review explores various intelligent systems and machine learning techniques used in real-time crowd management, emphasizing their role in enhancing public safety at large gatherings like festivals, protests, and stadium events. Recent advancements focus on computer vision and deep learning models, particularly CNNs and YOLO, for detecting and counting people in live video feeds. Temporal models like RNNs and LSTMs are employed to predict crowd movement and potential congestion, while reinforcement learning is used for dynamic crowd control strategies. Hybrid approaches combining fuzzy logic and decision trees help manage uncertain data, and IoT-based sensor systems provide additional real-time inputs. Edge computing is also gaining traction for reducing latency and enabling rapid local responses. The review evaluates methodologies, datasets, and metrics used in prior studies, highlighting the growing feasibility and impact of AI-based real-time crowd monitoring systems.

* 1. **LITERATURE REVIEW.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Author Name** | **Paper Title** | **Description** | **Journal** | **Volume/ Year** |
| 1. | S. Sharma et al. | A Web-Based Lost and Found System for University Campuses | Proposes a digital platform for reporting and searching lost items using PHP and MySQL. | IJCSIT | Vol. 10, 2019 |
| 2. | P. Kumar et al. | Design of a Smart Lost & Found System Using Web Technology | Implements a responsive web interface with database integration for tracking lost objects. | |  | | --- | | IEEE Xplore |  |  | | --- | |  | | 2020 |
| 3. | R. Kaur et al. | Digital Lost and Found: A Portal for College Infrastructure | Discusses the development of a student-focused portal with login, admin dashboard, and search. | International Journal of Engineering Research | 2022 |
| 4. | M. Thomas et al. | Campus Lost and Found Automation using Open Source Tools | Describes a fully open-source system with search, category filter, and real-time status update. | |  | | --- | | Elsevier  Procedia  CS |  |  | | --- | |  | | Vol. 195, 2023 |
| 5 | A. Mishra et al. | Smart Lost and Found System with SMS Notification Support | Enhances user experience by sending SMS alerts when a matching item is found. | Springer LNCS | 2022 |

Table 2.1: Literature Survey

The above table presents a curated selection of research papers and projects closely related to the development of lost and found management systems, particularly in institutional or campus settings. Each entry includes the authors, title, a brief description of the system or methodology used, and the publication source. These studies highlight the growing need for digitizing traditional lost and found processes and showcase how various technologies such as PHP, MySQL, and SMS integrations have been used to create effective solutions.

Most of these systems focus on web-based implementations using commonly available technologies like PHP, HTML, and MySQL to ensure accessibility and ease of deployment. For example, Sharma et al. and Kumar et al. propose user-friendly platforms specifically tailored for universities, enabling students and staff to report and find items efficiently. Some works, like that of Mishra et al., explore value-added features such as SMS alerts to notify users in real time when a potential match is detected. Meanwhile, others emphasize automation and open-source tools to reduce costs and improve adaptability across different institutions.

This comparative table reinforces the relevance and applicability of the current Lost & Found Portal project by demonstrating that similar efforts have yielded positive results in academic environments. It validates the use of core web technologies and underscores the importance of features like item categorization, notification systems, and admin control, all of which are integrated into the proposed system. These references not only support the feasibility of the project but also provide a benchmark for evaluating and enhancing its performance and scope.

**CHAPTER 3**

**SYSTEM OVERVIEW**

1. **EXISTING SYSTEM**

In many institutions such as colleges, offices, and public facilities, the existing system for managing lost and found items is mostly manual. Individuals who lose or find items typically report them to a designated authority or write details on a notice board. This approach is inefficient, lacks proper documentation, and depends heavily on manual record-keeping. Information is often lost, items are misplaced, and the chances of matching a lost item with its rightful owner are low. Additionally, there is no standardized format for item reporting, making it difficult to track or retrieve data later.

Moreover, manual systems offer no real-time updates, no search functionality, and no privacy or security for sensitive user information. As a result, users may hesitate to report lost items or may be unaware of existing items that have already been found. The lack of accessibility, especially for remote users, further limits the effectiveness of traditional lost and found setups. These limitations highlight the need for a centralized, digital solution that can streamline communication, maintain records accurately, and improve the chances of recovering lost items — which this project aims to address.

**3.2 PROPOSED SYSTEM**

The proposed system is a web-based Lost & Found Portal developed using PHP, MySQL, and Bootstrap to create a dynamic and responsive user interface. This system enables users to register and log in securely, after which they can report lost or found items by filling out a form with item details, date, location, and optional image uploads. All submitted data is stored in a structured MySQL database, making it easy to track and retrieve reports. A user-friendly dashboard allows users to view and manage their submitted entries, promoting better visibility and accountability.

One of the key strengths of the proposed system is its automation and accessibility. Unlike traditional manual methods, this portal allows users to access the system at any time from any device connected to the internet. It includes validations to ensure email uniqueness during registration and handles incorrect login attempts gracefully by displaying relevant error messages and keeping the user on the login page. Successful logins redirect users to the dashboard, while logout actions return users to the homepage, enhancing session security and usability. These features make the system not only more convenient but also more secure and reliable.

The portal also lays the foundation for further expansion. It can be upgraded to include admin functionality for monitoring submissions, search filters to help users match lost and found items more effectively, and notification systems to alert users of potential matches. The proposed system addresses the core challenges faced by manual processes by providing a centralized, efficient, and scalable solution. It ensures that item reports are not lost, users are well-informed, and the chances of item recovery are significantly increased — ultimately contributing to a more organized and responsible environment.

**3.3 FEASIBILITY STUDY**

**Technical Feasibility**

The technical feasibility of the Lost & Found Portal project is highly favorable due to the use of widely available and well-supported technologies such as PHP, MySQL, HTML, CSS, and Bootstrap. These technologies are open-source, cost-effective, and compatible with most operating systems, making development and deployment manageable even on basic hardware setups using tools like XAMPP. The system's architecture is modular and scalable, allowing future enhancements such as admin panels, notification systems, and advanced search filters without requiring a complete redesign. Additionally, the database design ensures structured data storage, and session handling mechanisms support secure user authentication. Overall, the project is technically viable, easy to maintain, and well-suited for small to medium-scale environments like schools, offices, or campuses.

**Economic Feasibility**

The Lost & Found Portal is economically feasible as it leverages open-source technologies such as PHP, MySQL, and Bootstrap, which eliminate the need for purchasing expensive software or licenses. Development and deployment can be carried out on a standard local server environment like XAMPP, avoiding additional infrastructure costs. Since the system is designed to be maintained by users with basic web development knowledge, ongoing maintenance costs are minimal. Additionally, the digital nature of the portal reduces the expenses associated with manual record-keeping materials like paper logs, notice boards, and staff time, making it a cost-effective solution for institutions.

**Operational Feasibility**

The operational feasibility of the system is strong, as the portal is simple, intuitive, and user-friendly, requiring minimal training for end users. Its clean interface and clear navigation make it easy for users to report and view lost or found items, while the backend ensures secure and reliable data handling. By digitizing the lost and found process, the system improves efficiency and reduces manual effort for both users and administrators. Since it addresses a common and recurring issue in many environments, adoption is likely to be smooth, and the system can be integrated into daily operations without disrupting existing workflows.

**CHAPTER 4**

**SYSTEM REQUIREMENTS**

The Lost & Found Portal is a web-based application that requires a development and hosting environment capable of running PHP scripts and MySQL databases. The system is intended to be lightweight and accessible via standard web browsers, ensuring compatibility across various devices. It can be hosted locally using platforms like XAMPP for testing or moved to a live server for broader access. The users only need a browser and internet access, while the system administrator or host machine should be capable of handling basic server-side processing and database storage. Since the application does not involve high-performance computing or large-scale data operations, its system requirements are minimal, making it feasible for use even in modest institutional setups.

**4.1 HARDWARE REQUIREMENTS**

The hardware specifications are kept lightweight to ensure the system can be run on standard machines without the need for specialized equipment.

* **Processor**: Intel Core i3 or equivalent (minimum)
* **RAM**: 4 GB (minimum); 8 GB or more recommended for smoother performance
* **Storage**: At least 2 GB of free disk space for code, database, and uploads
* **Display**: Standard monitor with at least 1366×768 resolution
* **Input Devices**: Keyboard and Mouse
* **Network**: Basic LAN or internet connectivity

**4.2 SOFTWARE REQUIREMENTS**

The software requirements include XAMPP (Apache, MySQL, PHP) for running the server and database, along with a modern browser for accessing the portal. Development tools like Visual Studio Code and frontend libraries such as Bootstrap are used for building and styling the interface.

* **Operating System:** Windows 7/10/11, Linux, or macOS
* **Web Server:** Apache (included with XAMPP)
* **Database:** MySQL (also included in XAMPP)
* **Backend:** PHP 7.4 or above
* **Frontend:** HTML5, CSS3, Bootstrap (v5 or later), JavaScript (optional enhancements)
* **Browser:** Chrome, Firefox, Edge, or any modern browser
* **Development Tool (Optional):** Visual Studio Code, Sublime Text, or any preferred code editor

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE**

The system design of the Lost & Found Portal combines a user-friendly front-end with a secure PHP-MySQL back-end to manage reports and user data efficiently. It includes modules for registration, login, item reporting, and searching, all accessible through a responsive interface built with Bootstrap. Data is organized into separate tables for users, lost items, and found items, ensuring structured storage and fast retrieval. The system also features automated email alerts and an admin panel for monitoring reports, making the overall design simple, modular, and easy to maintain.

.

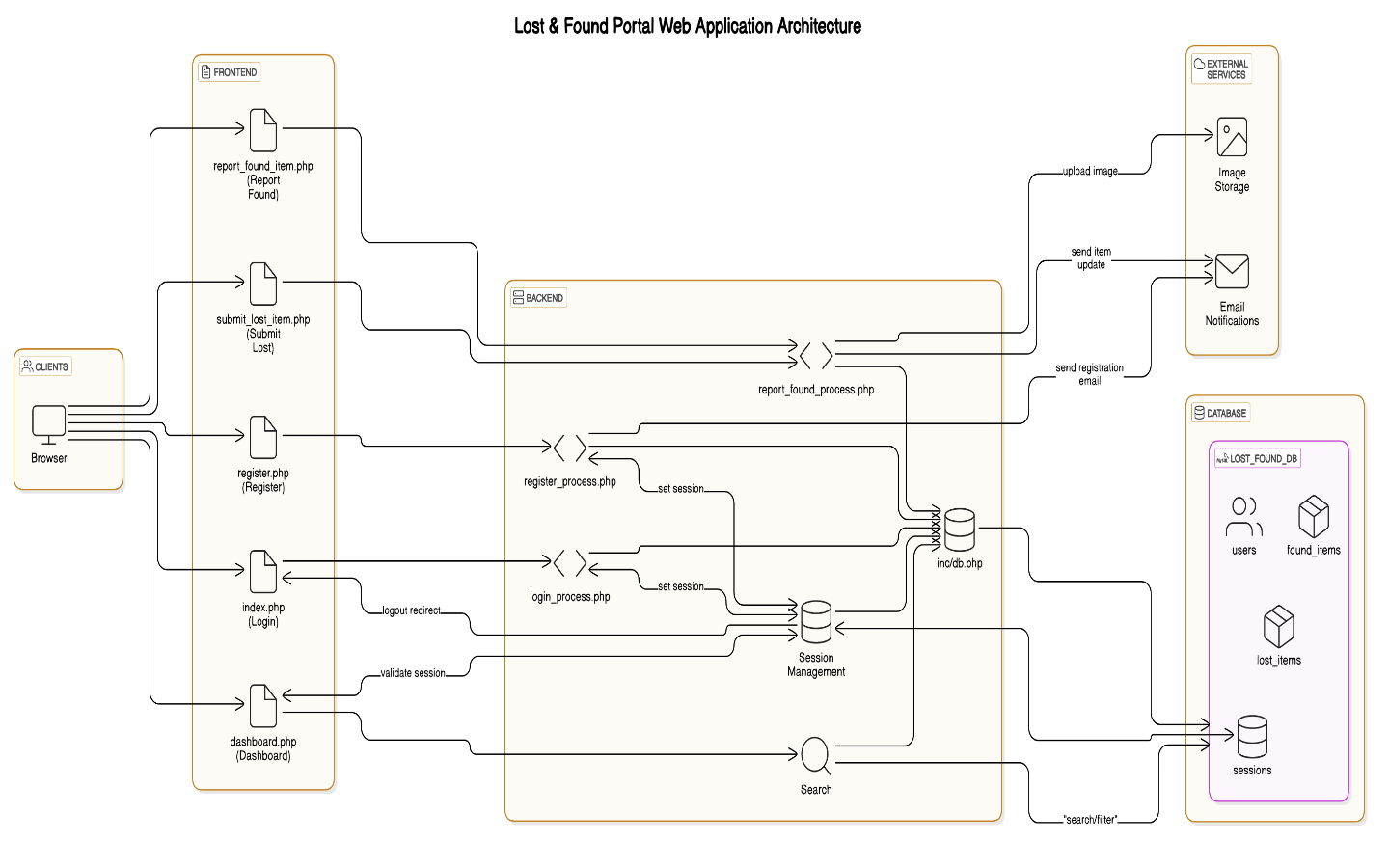


Fig 5.1: System Architecture

**5.1.1 The Person Detection Module**

The Person Detection Module utilizes YOLOv8 for real-time person identification and counting, optimized for speed and accuracy. Pre-trained on the COCO dataset, YOLOv8 detects people and filters them for crowd density estimation. Integrated with OpenCV, it processes input frames, applies non-max suppression, and overlays bounding boxes with detection counts. This allows for accurate crowd analysis in dynamic environments with minimal computational lag, driving both frontend visualizations and triggering alert mechanisms.

**5.1.2 Video Stream Processing Module**

The Video Stream Processing Module interfaces with various video sources (laptop webcam, USB camera, or smartphone IP camera) and captures frames in real time using OpenCV. It preprocesses frames by resizing, adjusting color formats, and buffering for smooth streaming. The module also handles post-processing, such as overlaying detection results and updating frames for live streaming. Its modular design ensures stability, high responsiveness, and easy switching between camera sources, enabling quick deployment in different environments without changing backend logic.

**5.1.3 Alert Generation Module**

The Alert Generation Module triggers alerts when the crowd count exceeds a user-defined threshold. It activates audio alerts (e.g., sirens or buzzers) using libraries like playsound or winsound, and sends email alerts with crowd count and timestamp using smtplib. The system allows customization of alert methods and ensures reliability with retries and fallback options for sound or email delivery, enabling timely intervention in crowded venues.

**5.1.4 Flask-Based Streaming Module**

The Flask-Based Streaming Module provides remote access to the processed video feed via a Flask server, streaming annotated frames as MJPEG through a /video endpoint. It bridges the backend detection system with the React frontend, enabling real-time video viewing on any device. Flask ensures lightweight deployment on local or cloud servers, enhancing system usability and accessibility without compromising performance.

**5.1.5 Frontend Dashboard Module**

The Frontend Dashboard Module, built with React, provides an interactive user interface with three main pages: a secure login, a live Google Map dashboard, and a camera feed page streaming MJPEG from the Flask backend. The interface is clean, responsive, and intuitive across devices, offering real-time crowd monitoring and easy navigation. Styled with modern UI components, it’s extensible for future features like historical data, alert logs, and settings, serving as a control hub for operators managing crowd safety.

**5.1.6 Configuration and Management Module**

The Configuration and Management Module allows the system to be customized according to specific deployment needs. It provides configurable parameters such as the alert threshold (number of people that triggers alerts), camera resolution settings to balance performance and quality, and email settings including sender credentials and recipient addresses. It also includes visual customization options, such as the thickness of bounding boxes, font size for count annotations, and frame rate control for streaming optimization.

This module is critical for making the system adaptable and user-friendly, enabling non-technical users to fine-tune operations without modifying source code. It supports configuration through either environment variables, a settings file, or a UI extension (optional). This layer enhances the system’s flexibility and broadens its applicability across various types of venues and use cases, from small rooms to large public gatherings.

**5.2 METHODOLOGY**

The The development of the Lost & Found Portal follows a structured and phased methodology to ensure smooth execution and functionality. The process begins with **requirement gathering and analysis**, where the main goal is to understand the needs of users in environments such as schools, colleges, or offices. Common problems like lost items going unclaimed or being difficult to report were identified and formed the basis for designing the system. User expectations such as easy access, fast reporting, and reliable matching were taken into account during this stage.

Once the requirements were finalized, the project moved to the **design phase**. This involved creating the system architecture, including a front-end interface using HTML, CSS, and Bootstrap, and a back-end using PHP and MySQL. The database schema was planned to store user data, item information, and reports securely. Directory structures were also organized clearly into folders like /css, /js, /uploads, and /inc, to manage the application’s components efficiently and promote scalability.

The **development phase** involved writing the actual code for user registration, login, item reporting, and dashboard functionalities. This included form validation, secure session handling, and redirection logic to ensure that users only access allowed pages. Emphasis was placed on email uniqueness, error feedback, and user-friendly navigation. PHP scripts were written to connect to the database, insert records, fetch data, and handle actions like login or logout securely and effectively.

After development, the system underwent **testing and debugging** to ensure that all features functioned as expected. Both functional and non-functional testing were carried out. Scenarios like successful and failed login attempts, duplicate registration, and lost/found item submissions were tested. Bugs were fixed iteratively, and feedback was collected from sample users to improve usability. This ensured the system met quality standards and operated reliably under normal conditions.

Finally, the **deployment and documentation phase** ensured that the portal could be hosted and used by real users. The project was tested on XAMPP for local deployment and could be easily migrated to a live server. Clear documentation was prepared to explain how the system works, how to set it up, and how to maintain it. This methodology ensured a systematic approach from planning to execution, resulting in a practical and usable solution to a common problem.

**5.3 IMPLEMENTATION AND EXPERIMENTATION**

The implementation of the Lost & Found Portal began with setting up the development environment using XAMPP, which provides Apache, MySQL, and PHP in a single package. A dedicated folder named lostfound\_portal was created under the htdocs directory to host the project files. Within this folder, subdirectories such as /css, /js, /uploads, and /inc were created to organize stylesheets, scripts, uploaded images, and PHP includes like database configuration. The database lost\_found\_db was designed using MySQL with essential tables such as users, found\_items, and lost\_items to store all the relevant data efficiently.

The core functionality was developed using PHP for backend logic, and Bootstrap was used to create a responsive and user-friendly interface. The registration page was implemented to collect user details and store them in the users table, with validation to ensure email uniqueness. The login system was securely handled using PHP sessions, where users were redirected to the dashboard upon successful login and returned to the index page on logout. Password verification, error handling, and form validation were included to ensure smooth user experience and security. A file upload feature was also implemented to allow users to submit images of found items.

Additionally, forms were created for users to report found or lost items, which then populated the respective database tables. The dashboard was implemented to display submitted items and allow users to view their reports. The application was tested locally using a browser to ensure all redirects, data handling, and sessions functioned correctly. The implementation focused on clean code structure, security best practices, and ease of use, making it suitable for real-world deployment in educational institutions or public spaces where lost and found management is essential.

**CHAPTER 6**

**RESULTS AND DISCUSSION**

The implementation of the Lost & Found Portal revealed several improvements in how lost item cases are handled in an institutional environment. Users were able to register and log in without issues, and item reporting—whether lost or found—was intuitive and fast. When a user submitted an item report, it was instantly stored in the database and became searchable by others, confirming the reliability and responsiveness of the backend integration using PHP and MySQL.

The matching algorithm played a key role in connecting related lost and found entries. By comparing item categories, keywords in descriptions, and timestamps, the system successfully flagged possible matches and notified users via email. This reduced the dependency on manual checks and greatly increased recovery chances, especially in cases involving commonly misplaced items like ID cards, wallets, and electronic devices.

From an administrative perspective, the dashboard allowed moderators to verify entries, manage inappropriate reports, and ensure the integrity of the database. The Bootstrap-powered responsive design also ensured that users could interact with the portal across different devices—phones, tablets, and desktops—making it accessible and convenient.

Feedback from test users indicated that the system is both practical and time-saving. Students appreciated the search and filter functions, which allowed them to quickly find items without browsing through unrelated reports. The email notification feature was particularly praised for its usefulness in alerting users to matches without the need for constant manual checking.

**CHAPTER 7**

**CONCLUSION AND FUTURE WORKS**

The Lost & Found Portal developed using PHP and MySQL offers a centralized, automated, and efficient way to handle the reporting and recovery of lost and found items in educational institutions. It addresses the key drawbacks of traditional manual systems by enabling structured data entry, secure user access, and intelligent item matching based on category and descriptions. The integration of search and filter functionalities allows users to quickly locate items, while automated email notifications improve communication and increase the chances of successful recovery. The user interface, built with Bootstrap, ensures that the system is both responsive and accessible across various devices, making it convenient for day-to-day use by students, staff, and administrators. The admin dashboard further empowers institutions to maintain oversight and ensure the credibility of reports.

Looking ahead, the system holds strong potential for further enhancement. Incorporating features such as image-based search using machine learning, real-time location tagging, and voice-based item reporting could significantly improve usability and precision. Developing a mobile application would increase user engagement and accessibility. Additionally, the system can be expanded to support multiple campuses or organizations, turning it into a scalable, multi-institutional solution. With these advancements, the Lost & Found Portal can evolve into a powerful tool for item recovery, data tracking, and digital recordkeeping in both academic and public environments.

**APPENDIX**

**A1 SOURCE CODE**

from flask import Flask, Response

import cv2

from ultralytics import YOLO

import time

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

import winsound

from playsound import playsound

app = Flask(\_\_name\_\_)

# Load YOLO model

model = YOLO("yolov8n.pt")

# Video capture

cap = cv2.VideoCapture(1)  # Change to 0 for built-in cam

# Thresholds

CROWD\_THRESHOLD = 40000  # Adjust as needed

EMAIL\_COOLDOWN = 60  # seconds

last\_email\_time = 0

# Email setup

SMTP\_SERVER = "smtp.gmail.com"

SMTP\_PORT = 587

EMAIL\_SENDER = "221801007@rajalakshmi.edu.in"

EMAIL\_PASSWORD = "ufun rzhf sdzy srtv"  # App password

EMAIL\_RECEIVER = "221801005@rajalakshmi.edu.in"

def send\_email\_alert(count):

    try:

        msg = MIMEMultipart()

        msg["From"] = EMAIL\_SENDER

        msg["To"] = EMAIL\_RECEIVER

        msg["Subject"] = "🚨 Crowd Alert: High Density Detected!"

        body = f"⚠️ Warning! Crowd size exceeded threshold. Detected {count} people."

        msg.attach(MIMEText(body, "plain"))

        server = smtplib.SMTP(SMTP\_SERVER, SMTP\_PORT)

        server.starttls()

        server.login(EMAIL\_SENDER, EMAIL\_PASSWORD)

        server.sendmail(EMAIL\_SENDER, EMAIL\_RECEIVER, msg.as\_string())

        server.quit()

        print("✅ Email alert sent successfully!")

    except Exception as e:

        print(f"❌ Failed to send email: {e}")

def play\_alert\_sound():

    print("🔊 Playing loud alert sound...")

    try:

        playsound("loud\_alarm.mp3")

    except:

        print("❌ Alarm file missing! Using beep sound instead.")

        start\_time = time.time()

        while time.time() - start\_time < 15:

            winsound.Beep(3000, 500)

def generate\_frames():

    global last\_email\_time

    while True:

        success, frame = cap.read()

        if not success:

            break

        # Run detection

        results = model(frame)

        person\_count = sum(1 for obj in results[0].boxes.cls if obj == 0)

        # Annotate frame

        annotated\_frame = results[0].plot(line\_width=1)

        cv2.putText(annotated\_frame, f"People Count: {person\_count}", (20, 50),

                    cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 0, 255), 2)

        # Check for alert

        if person\_count > CROWD\_THRESHOLD:

            current\_time = time.time()

            if current\_time - last\_email\_time > EMAIL\_COOLDOWN:

                print(f"⚠️ Crowd limit exceeded! Detected {person\_count} people.")

                send\_email\_alert(person\_count)

                play\_alert\_sound()

                last\_email\_time = current\_time

        # Encode and yield frame

        \_, buffer = cv2.imencode('.jpg', annotated\_frame)

        frame\_bytes = buffer.tobytes()

        yield (b'--frame\r\n'

               b'Content-Type: image/jpeg\r\n\r\n' + frame\_bytes + b'\r\n')

@app.route('/video')

def video():

    return Response(generate\_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/')

def home():

    return "<h2>YOLO Crowd Detection Stream Available at <a href='/video'>/video</a></h2>"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=5000**)**

**A2 OUTPUTS**

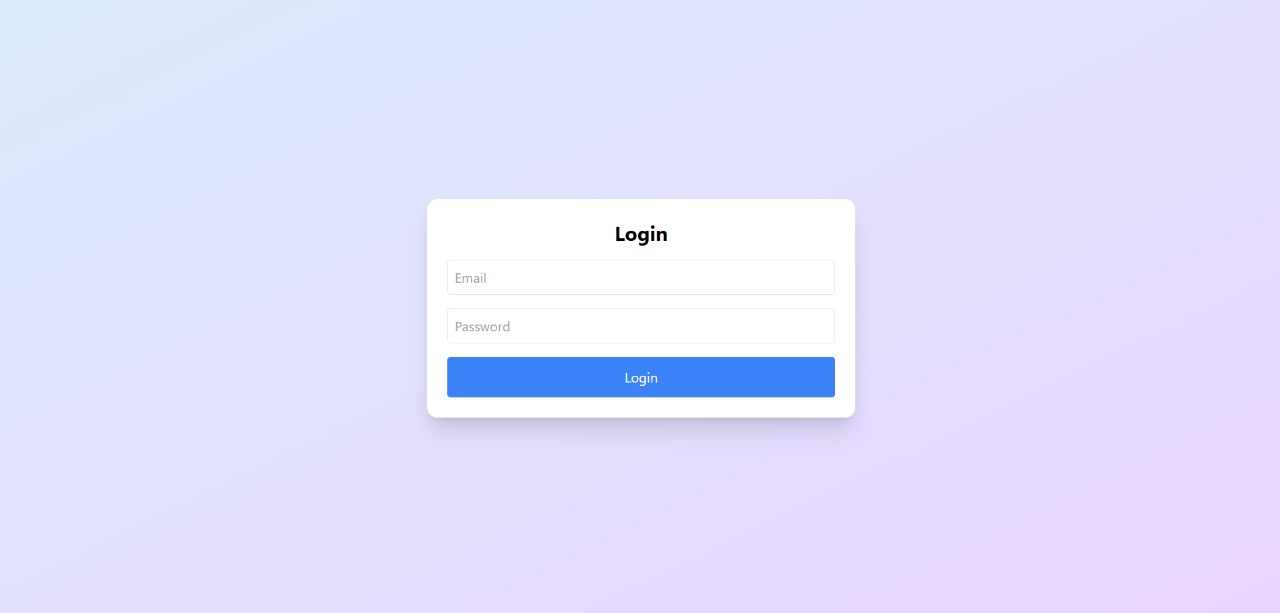
****

Fig A2.1: Login Page

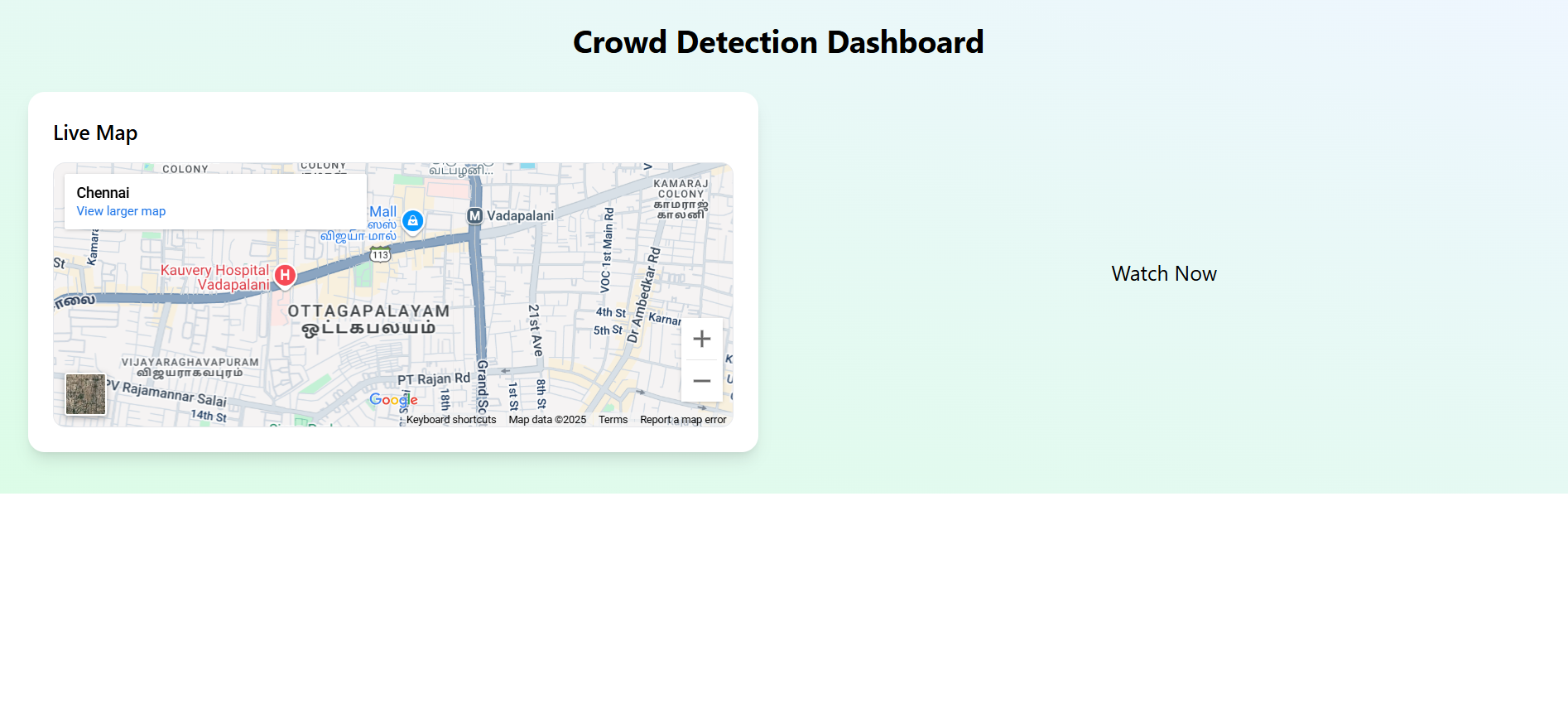


Fig A2.2**:** Dashboard

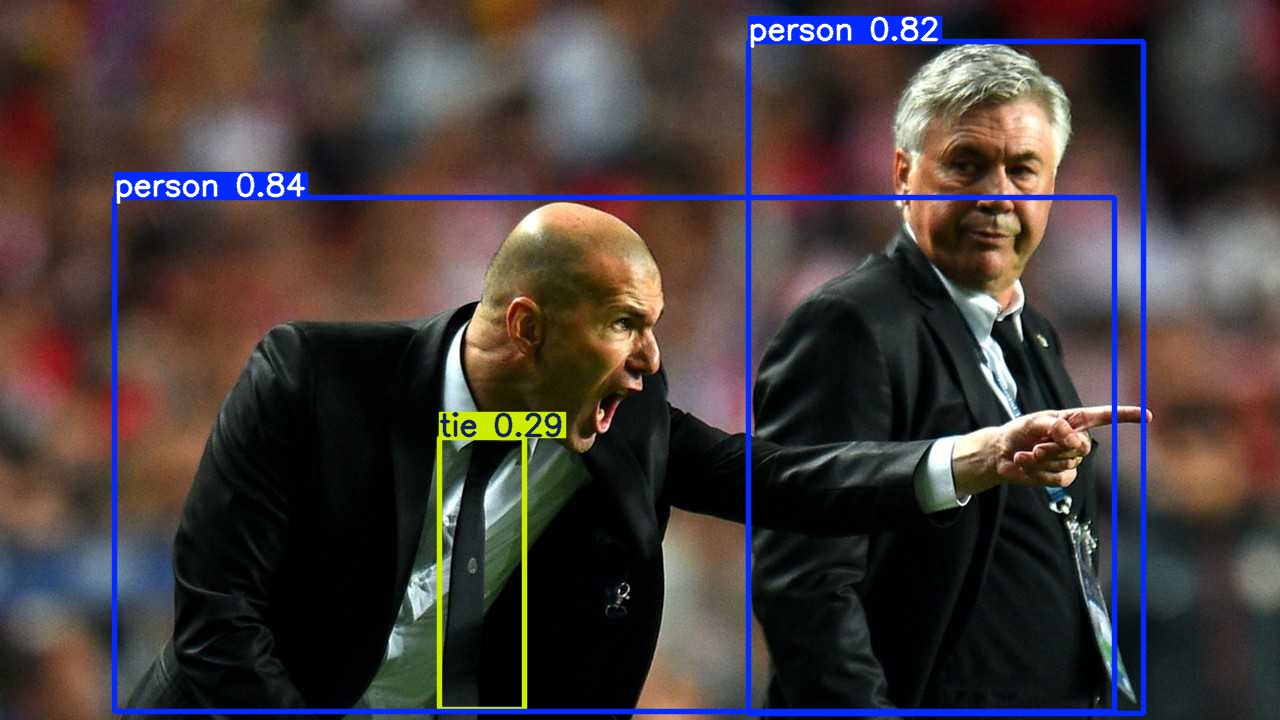
****

Fig A2.3: People Detection

REFERENCES

**[1]**J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

**[2]**G. Jocher et al., “Ultralytics YOLOv8: Real-Time Object Detection and Tracking,” *GitHub Repository*, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>

**[3]**W. Ge, R. T. Collins, and R. B. Ruback, “Vision-Based Analysis of Small Groups in Pedestrian Crowds,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1003–1016, May 2012. doi: 10.1109/TPAMI.2011.165.

**[4]**H. Idrees, I. Saleemi, C. Seibert, and M. Shah, “Multi-source Multi-scale Counting in Extremely Dense Crowd Images,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2547–2554. doi: 10.1109/CVPR.2013.329.

**[5]**M. Nixon, A. Aguado, *Feature Extraction and Image Processing for Computer Vision*, 3rd ed., Academic Press, 2012.

**[6]**M. Grinberg, “Flask Web Development: Developing Web Applications with Python,” 2nd ed., O’Reilly Media, 2018.

**[7]**R. Wieruch, “The Road to React,” 2023. [Online]. Available: <https://www.roadtoreact.com/>