

A Project Report on

# **Sentinel AI: Smart City Emergency Response System**

Submitted in partial fulfillment of the requirements for the award  
of the degree of

**Bachelor of Engineering**

in

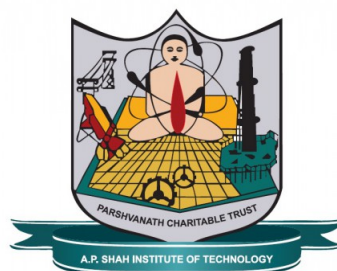
**Computer Science and Engineering - Data Science**

by

**Varad Joshi(21107002)  
Khushi Kadam(21107031)  
Shravani Kulkarni(21107063)  
Krish Jaswal(21107039)**

Under the Guidance of

**Ms.Rajashri Chaudhari**



**Department of Computer Science and Engineering - Data Science**

A.P. Shah Institute of Technology  
G.B.Road,Kasarvadavli, Thane(W)-400615  
UNIVERSITY OF MUMBAI

**Academic Year 2024-2025**

## Approval Sheet

This Project Synopsis Report entitled “*Sentinel AI: Smart City Emergency Response System*” submitted by *Varad Joshi (21107002)*, *Krish Jaswal (21107039)*, *Shravani Kulkarni (21107063)*, *Khushi Kadam (21107031)* is approved for the partial fulfillment of the requirement for the award of the degree of *Bachelor of Engineering* in *Computer Science and Engineering - Data Science* from *University of Mumbai*.

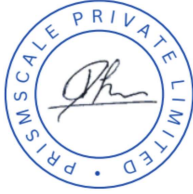
Ms. Rajashri Chaudhari  
Guide

Ms. Anagha Aher  
HOD, Computer Science and Engineering-Data Science

Place:A.P.Shah Institute of Technology, Thane  
Date:

## CERTIFICATE

This is to certify that the project entitled “*Sentinel AI: Smart City Emergency Response System*” submitted by “*Varad Joshi*” (21107002), “*Shravani Kulkarni*” (21107063), “*Khushi Kadam*” (21107031), “*Krish Jaswal*” (21107039) for the partial fulfillment of the requirement for award of a degree *Bachelor of Engineering* in *Computer Science and Engineering - Data Science*, to the University of Mumbai, is a bonafide work carried out during academic year 2024-2025.



Dharam Vir Singh  
Director  
PrismScale Pvt. Ltd.

Ms. Rajashri Chaudhari  
Guide

Ms. Anagha Aher  
HOD, CSE (Data Science)

Dr. Uttam D. Kolekar  
Principal

**External Examiner(s)**

**Internal Examiner(s)**

1.

1.

2.

2.

Place: A.P. Shah Institute of Technology, Thane  
Date:

HOD of Department of Computer Science and Engineering Data Science  
AP Shah Institute of Technology,  
Thane, Maharashtra

Dear Ma'am,

**(Subject : Request to Proceed with Student Project Work and Schedule for Completion)**

As part of the 4th year, Data Science curriculum, we are planning to assign **Mr. Varad Joshi (21107002), Mr. Krish Jaswal (21107039) , Miss Shravani Kulkarni (21107063) and Miss Khushi Kadam(21107031)** from your esteemed department of Data Science, our project **Face Recognition Module with Hardware Integration**. This project that will significantly contribute to both their academic growth and practical application of the theoretical knowledge acquired during the semester. In order to ensure the success of this project, I kindly request your permission and support for the students to engage in this initiative.

Please find details of the project enclosed in the letter.

**Project Title: Face Recognition Module with Hardware Integration**

**1. Project Overview**

**Short Description:**

This project focuses on developing a robust Face Recognition module capable of matching faces with high accuracy. The solution will integrate with hardware to enable seamless real-time facial identification for events, security, and other applications. PrismScale has identified multiple industry partnerships to utilize this technology in various domains such as event management, secure access control, and surveillance.

**2. Objectives**

- 1.High-Accuracy Face Recognition:** Implement a reliable module with high accuracy in face matching under varying environmental conditions.
- 2.Edge AI Architecture:** Ensure the entire system operates on edge devices for real-time processing and reduced latency.
- 3.Hardware Integration:** Seamlessly integrate with custom hardware for real-time scanning and identification.
- 4.Scalable Design:** Ensure the system is scalable to handle large datasets for events or enterprise use.
- 5.User Privacy and Security:** Implement strict measures to safeguard user data and comply with relevant regulations.
- 6.Customizable Applications:** Provide customizable features to meet the requirements of different industry partners.



### 3. Scope of Work

#### 3.1 Software Development

- Design and develop the Face Recognition module using state-of-the-art algorithms.
- Implement pre-processing for images to improve accuracy under low light or poor quality inputs.
- Train models with a large dataset to ensure diverse face recognition capabilities.
- Develop an API for hardware integration and third-party software.

#### 3.2 Hardware Integration

- Collaborate on designing hardware specifications for cameras, scanners, and other devices.
- Ensure low-latency data transmission between hardware and software components.
- Optimize the system for deployment on resource-constrained edge devices.

#### 3.3 Industry Application Features

- Real-time face scanning for event registrations.
- Secure access control for restricted areas.
- Logging and analytics for auditing and compliance.

### 4. Technical Requirements

#### 4.1 Software Stack

- Programming Languages: Python, C++, JavaScript
- Frameworks/Libraries: TensorFlow, OpenCV, PyTorch
- Database: PostgreSQL, MongoDB
- Edge Device Platforms: NVIDIA Jetson, Raspberry Pi, or equivalent

#### 4.2 Hardware Specifications

- High-resolution cameras with infrared capabilities
- Edge computing devices (e.g., NVIDIA Jetson, Raspberry Pi)
- Secure IoT-enabled devices for communication

#### 4.3 Integration Requirements

- API support for event management software and security systems
- Compatibility with standard communication protocols (e.g., MQTT, REST APIs)

## 5. Deliverables

- Fully functional Face Recognition module.
- Integration-ready hardware prototype.
- Detailed documentation including API specs and deployment guidelines.
- Performance evaluation report with accuracy metrics.
- Training and support for end-users and industry partners.

## 6. Timeline and Milestones

Milestone	Deadline	Deliverable
Requirement Analysis	Month One	Finalized requirement document
Model Development	Month Two	Initial prototype of the Face Recognition module
Hardware Integration Design	Month Three	Hardware specifications finalized
Testing and QA	Month Four	Fully integrated and tested system
Deployment	Month Five	Final deliverables and training provided

## 7. Success Metrics

- Achieving a minimum face recognition accuracy of 90%.
- Seamless hardware-software integration with negligible latency.
- Positive feedback from industry partners during pilot deployment.
- Successful handling of at least 1,000 faces in real-time scenarios.



## 8. Risks and Mitigation Strategies

Risk	Mitigation Strategy
Data privacy concerns	Implement encryption and anonymization techniques
Hardware-software incompatibility	Perform iterative testing during integration phases
Model bias	Use a diverse dataset for training and validation
Scalability issues	Opt for edge-native architecture with optimized hardware

## 9. Ownership and Sponsorship

- **Ownership:** PrismScale will reserve all rights to the project. The project and its outputs will be completely owned by PrismScale.
- **Sponsorship:** PrismScale will provide and sponsor all hardware required for the development and deployment of the project.

## 10. Student Partners for the Project

- Varad Joshi (21107002)
- Krish Jaswal (21107039)
- Shravani Kulkarni(21107063)
- Khushi Kadam(21107031)

I would be grateful if you could confirm your approval of the proposed schedule or suggest any adjustments as per your guidance. We aim to ensure that the students are given adequate time to complete the project while maintaining high academic standards.

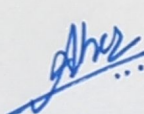
Thank you for your consideration and support. I look forward to your feedback and approval to proceed with this initiative.

On behalf of the entire PrismScale Team,



Dharam Vir Singh,  
Director,  
PrismScale Pvt. Ltd.  
IND | UAE | AUS



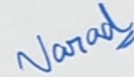
  
Anagha Aher,  
HOD of Department of Computer  
Science and Engineering

Data Science,  
AP Shah Institute of Technology,  
Thane, Maharashtra

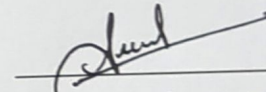
Date: 04-02-25

Head

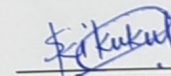
Department of Computer Science and Engineering (DS)  
A. P. Shah Institute of Technology, Thane (M.S)



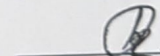
Varad Joshi (21107002)



Krish Jaswal (21107039)



Shravani Kulkarni (21107063)



Khushi Kadam (21107031)

Date: 04-02-25



## Acknowledgement

We have great pleasure in presenting the report on **Sentinel AI: Smart City Emergency Response System**. We take this opportunity to express our sincere thanks towards our guide **Ms. Rajashri Chaudhari** for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards her constant encouragement, support and guidance through the development of project.

We thank **Ms. Anagha Aher** Head of Department for her encouragement during the progress meeting and for providing guidelines to write this report.

We express our gratitude towards BE project co-ordinator **Ms. Poonam Pangarkar**, for being encouraging throughout the course and for their guidance.

We would also like to extend our heartfelt thanks to **PrismScale Pvt. Ltd.** for sponsoring our project and providing us with valuable support and resources. We are especially grateful to **Mr. Dharam Vir Singh, Director of PrismScale**, for his continuous mentorship, insightful guidance, and encouragement throughout the duration of this project. His expertise and support played a crucial role in shaping the direction and success of our work.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Varad Joshi**  
(21107002)

**Krish Jaswal**  
(21107039)

**Shravani Kulkarni**  
(21107063)

**Khushi Kadam**  
(21107031)

## Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Varad Joshi(21107002)

(Signature)

Krish Jaswal(21107039)

(Signature)

Shravani Kulkarni(21107063)

(Signature)

Khushi Kadam(21107031)

Date:



## Abstract

With the rapid expansion of urban populations, security challenges at large-scale events have become increasingly complex. Sentinel AI, originally designed as a comprehensive emergency management system for smart cities, has been refocused to address critical event security issues. Persistent problems such as ticket scalping, bot interference in booking systems, and the use of fraudulent identification documents demand innovative solutions. Leveraging advanced face recognition capabilities, Sentinel AI now incorporates the state-of-the-art InsightFace model—fine-tuned on the LFW dataset—to deliver 93.6% accuracy at a threshold of 0.5 while achieving zero false positives, a crucial requirement for dependable biometric verification. To further enhance system reliability, especially in low-light environments, the framework integrates infrared sensor data that improves identification accuracy under challenging conditions. Complementing these improvements is a hardware prototype developed using a cost-effective QHD webcam and Raspberry Pi platform, supported by industry sponsorship. Additionally, the incorporation of UIDAI APIs streamlines ticket booking and security verification processes, ensuring that only eligible individuals gain entry, thereby reducing incidents of underage access and scalping. By transitioning from a focus on urban emergency response to addressing pressing event security needs, Sentinel AI establishes a dual role—combining real-time detection, predictive analytics, and robust biometric verification to mitigate security threats proactively. This evolution not only fortifies event safety but also lays the groundwork for smarter, data-driven urban security solutions, demonstrating how academic research can be seamlessly transformed into commercially viable and socially impactful technology.

*[**Keywords:** Sentinel AI, Face Recognition, Event Security, InsightFace, Infrared Sensors, Raspberry Pi, QHD Webcam, UIDAI API, Biometric Verification, Predictive Analytics, Urban Security, Ticket Scalping, Fraud Detection .]*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Problem Statement . . . . .	3
1.3	Objectives . . . . .	5
1.4	Scope . . . . .	7
<b>2</b>	<b>Literature Review</b>	<b>9</b>
2.1	Comparative Analysis of Recent study . . . . .	10
<b>3</b>	<b>Project Design</b>	<b>14</b>
3.1	Existing System . . . . .	14
3.2	Proposed System . . . . .	15
3.2.1	Critical Components of System Architecture . . . . .	15
3.3	System Diagrams . . . . .	19
3.3.1	UML Diagram . . . . .	20
3.3.2	Activity Diagram . . . . .	20
3.3.3	Use Case Diagram . . . . .	23
3.3.4	Sequence Diagram . . . . .	25
3.3.5	Data Flow Diagram [DFD] . . . . .	27
<b>4</b>	<b>Project Implementation</b>	<b>29</b>
4.1	Steps to access the System . . . . .	30
4.2	Timeline Sem VIII . . . . .	39
<b>5</b>	<b>Testing</b>	<b>42</b>
5.1	Software Testing . . . . .	42
5.2	Functional Testing . . . . .	44
<b>6</b>	<b>Result and Discussions</b>	<b>47</b>
<b>7</b>	<b>Conclusion</b>	<b>48</b>
<b>8</b>	<b>Future Scope</b>	<b>49</b>
	<b>Appendices</b>	<b>52</b>
	Appendix-A . . . . .	52



# List of Figures

3.1	Proposed System Architecture . . . . .	17
3.2	Working diagram . . . . .	18
3.3	Activity Diagram . . . . .	21
3.4	Use Case diagram . . . . .	24
3.5	Sequence diagram . . . . .	26
3.6	Data Flow Diagram . . . . .	28
4.1	Book Your Ticket – Initial Booking Form . . . . .	30
4.2	Book Your Ticket – Successful Booking with Confirmation . . . . .	31
4.3	Booking Details and Check-In Dashboard . . . . .	33
4.4	Book Your Ticket – Another Successful Booking Example . . . . .	34
4.5	Face Recognition-Based Check-In Interface . . . . .	35
4.6	Secure Payment Interface – Ticket Booking . . . . .	36
4.7	Razorpay Demo Bank Transaction Page . . . . .	37
4.8	Timeline of project: [Part 1] . . . . .	40
4.9	Timeline of project: [Part 2] . . . . .	41
5.1	Software Testing . . . . .	43
5.2	Functional Testing . . . . .	44

# List of Tables

2.1	Comparative Analysis of Literature Survey on Smart City Emergency Systems	12
5.1	Functional Testing of Sentinel AI Ticketing and Check-In System . . . . .	46



# List of Abbreviations

<b>UIDAI</b>	Unique Identification Authority of India
<b>QHD</b>	Quad High Definition
<b>UML</b>	Unified Modeling Language
<b>DFD</b>	Data Flow Diagram
<b>ANN</b>	Approximate Nearest Neighbor
<b>CNN</b>	Convolutional Neural Network
<b>LLM</b>	Large Language Model
<b>FAISS</b>	Facebook AI Similarity Search
<b>PCI</b>	Payment Card Industry
<b>CAPTCHA</b>	Completely Automated Public Turing test to tell Computers and Humans Apart
<b>RSVP</b>	Répondez s'il vous plaît
<b>OTP</b>	One Time Password
<b>IR</b>	Infrared
<b>NLP</b>	Natural Language Processing
<b>3DMM</b>	3D Morphable Model
<b>UV</b>	Universal Village

# Chapter 1

## Introduction

Rapid urbanization has brought about a surge in security challenges at large-scale public events, where issues such as unauthorized ticket scalping, bot-driven booking fraud, and the risk of underage entry have become increasingly prevalent. Traditional security measures at events are often hindered by the lack of real-time data and efficient verification processes, leaving organizers vulnerable to fraud and mismanagement. Fragmented communication between booking platforms and security agencies further compounds these challenges, resulting in delayed interventions and compromised safety for attendees.

This report focuses on "Refocusing Sentinel AI for security applications," an initiative that transforms the original Sentinel AI emergency response framework into a cutting-edge event security tool. By leveraging state-of-the-art face recognition technology—specifically, the InsightFace model fine-tuned on the LFW dataset—the system achieves a 93.6% accuracy rate at a threshold of 0.5 while maintaining zero false positives, a critical requirement for dependable biometric verification. Enhanced by the integration of infrared sensor data, the framework ensures robust performance even under challenging lighting conditions typical of nighttime or indoor events. The key contributions of this work revolve around the creation of a comprehensive, secure, and accessible face recognition-based ticket booking and verification system. Firstly, the project presents the design and implementation of a robust AI-powered biometric verification system that ensures accurate identity authentication, especially at the point of check-in. Leveraging deep learning-based face recognition models, the system effectively enhances security by preventing unauthorized access and enabling seamless user experiences.

In addition to software development, a functional hardware prototype has been constructed using cost-effective and readily available components, including a QHD webcam and a Raspberry Pi. This hardware setup demonstrates that high-quality biometric verification can be achieved even within a constrained budget, making the system scalable and feasible for deployment in real-world public venues such as stadiums, auditoriums, and transport hubs. Furthermore, the integration of UIDAI APIs into the ticket booking process adds an additional layer of verification by cross-checking user details with official government records. This integration not only streamlines the ticketing workflow but also strengthens legal compliance and helps prevent issues such as underage entry, identity fraud, and bulk ticket scalping. Collectively, these contributions highlight the project's commitment to building a secure, affordable, and user-friendly biometric access solution.

## 1.1 Motivation

For large-scale public events. As events grow in size and complexity, conventional security measures are increasingly challenged by issues such as ticket scalping, bot-driven fraud, and the use of forged identification documents. Traditional systems often fail to provide real-time verification and coordination, leaving events vulnerable to fraudulent practices and security breaches. This project aims to harness the power of artificial intelligence, IoT, and advanced biometric technologies to create a responsive, reliable, and intelligent event security system. By streamlining verification processes and enhancing communication between security stakeholders, the system is designed to safeguard attendees and elevate the overall security framework at public events.

### 1.1.1 Personal Experiences:

Direct encounters with security lapses at events—ranging from delays in identity verification to instances of ticket fraud—have deeply influenced the motivation behind this project. Observing how these issues compromise event safety and disrupt operations has driven a strong commitment to developing a system that delivers real-time, accurate security checks. This personal insight underscores the importance of creating solutions that effectively prevent fraud and ensure a seamless, secure experience for every event participant.

### 1.1.2 Technological Advancements:

The rapid evolution of AI, IoT, and edge computing technologies presents an unprecedented opportunity to transform event security. Recent advancements in real-time data processing, sensor integration, and machine learning algorithms enable the development of systems that can quickly and accurately verify identities using state-of-the-art facial recognition techniques. By leveraging models such as InsightFace—which has been fine-tuned on extensive datasets—this project capitalizes on technological innovations to deliver a solution that not only meets but exceeds the stringent requirements of modern event security.

### 1.1.3 Community Impact:

The primary driving force behind refocusing Sentinel AI is the desire to significantly enhance safety and improve the overall experience at public events. In an era where fraudulent activities and security breaches can severely disrupt operations and endanger attendees, establishing a robust, real-time security system is paramount. By addressing critical issues such as ticket scalping and unauthorized entry, this project aims to provide event organizers and security personnel with the essential tools for proactive risk management. The resulting improvements in security not only protect individual participants but also bolster public confidence in the safety and integrity of large-scale events. By leveraging cutting-edge technologies such as AI-powered facial recognition, biometric verification, and smart sensor integration, the Sentinel AI project seeks to provide event organizers with a comprehensive, automated security solution that can detect and prevent security breaches in real time. This system is designed not only to identify unauthorized access but also to anticipate potential security threats before they escalate, enabling quick and proactive responses. The integration of smart sensors allows for the monitoring of crowd dynamics and behavioral anomalies, ensuring that event organizers are alerted to potential risks—whether from overcrowding, suspicious activity, or emergencies—without the need for constant human oversight. This real-time threat detection capability is crucial for events where large crowds and high-stakes situations make traditional security protocols less effective.



## 1.2 Problem Statement

With the rapid growth of urban populations, public events are facing increasing security challenges that threaten both attendee safety and operational integrity. As events become larger and more complex, issues such as ticket scalping, bot-driven fraud, and the use of forged identification documents have emerged as critical concerns. Traditional event security measures often fall short due to several key limitations: they rely heavily on manual verification, which is prone to human error and inefficiency, and often fail to scale effectively with large crowds. Moreover, conventional systems lack real-time data integration and analytics, making it difficult to proactively identify threats or respond swiftly to incidents. The absence of advanced authentication mechanisms also leaves events vulnerable to impersonation, unauthorized access, and revenue loss. These vulnerabilities not only compromise the experience for legitimate attendees but also pose significant logistical and reputational risks for organizers. Consequently, there is an urgent need for intelligent, tech-driven solutions that can enhance both security and user convenience without disrupting the flow of the event.

- **Lack of Real-Time Data:**

Existing systems in event management often rely on delayed or static information, which severely hampers the ability to make informed decisions quickly. The traditional approach to event security typically involves the collection of data at various stages, but this data is often analyzed after the fact, leaving security teams unable to detect fraudulent activities—such as unauthorized ticket resales, forged IDs, or counterfeit tickets—until incidents have already occurred. This delayed response leads to inefficiency, with security teams reacting to issues only once they have escalated, thereby increasing the risk of further incidents. For example, if a fraudulent ticket is detected after an attendee has already entered the venue, it may allow others with similar fake tickets to gain entry as well. The lack of real-time data makes it challenging for security teams to continuously monitor ongoing transactions and identify suspicious activities as they happen. By the time the issue is flagged, it may be too late to prevent its impact, making it more difficult to mitigate the consequences of the fraud or breach. A real-time data-driven approach can significantly improve response times and reduce the risk of security lapses.

- **Fragmented Communication:**

Event security typically involves multiple stakeholders, including booking platforms, law enforcement, on-site security personnel, and event organizers. However, these groups often operate in silos with limited real-time coordination, making it difficult to share crucial information instantly across all parties involved. This fragmentation of communication impedes the ability to respond swiftly and collectively to security breaches or instances of fraud. For example, if an event security team identifies suspicious behavior or detects unauthorized access, they may not be able to quickly communicate this with law enforcement or event organizers without relying on outdated or manual communication methods. This lack of synchronization can delay crucial responses, allowing security breaches to go unchecked for longer periods and putting attendees at greater risk. Without a unified, real-time communication system, event security operations cannot function efficiently, and the collective response to a critical

situation becomes disjointed, ultimately undermining the overall safety and security of the event. A more integrated communication infrastructure would allow all stakeholders to receive instant alerts, share information, and coordinate their actions in real time, ensuring that responses are timely and accurate.

- **Privacy and Data Security Issues:**

The collection of personal information for verification purposes, such as identification documents, facial images, and personal details, has become a standard practice in event security systems. However, many existing systems fail to implement adequate privacy safeguards, leading to potential misuse of sensitive data. For example, personal information may be stored in databases without sufficient encryption or security measures, leaving it vulnerable to cyberattacks or data breaches. Such incidents could expose attendees' personal data, resulting in identity theft, financial fraud, or unauthorized access to other services. The lack of robust data security protocols has raised significant concerns among both event organizers and attendees about how their sensitive information is handled and protected. This lack of trust can ultimately discourage users from engaging with the system and diminish the overall effectiveness of the security measures.

- **Increased Risk Due to Event Complexity:**

As events continue to scale up in size and complexity, ensuring secure access and preventing fraudulent behavior becomes exponentially more challenging. The issues of ticket authenticity, underage entry, and bot interference are magnified as the sheer volume of attendees and transactions increases. For example, verifying each ticket's authenticity manually in a large-scale event could lead to delays, while bots attempting to purchase tickets in bulk for resale or fraudulent use can overwhelm traditional ticketing systems. Similarly, traditional methods for verifying age restrictions or ensuring that attendees meet specific event requirements may not be scalable or sufficiently reliable for events with large, diverse crowds. The task of identifying and blocking bots or detecting underage individuals can become particularly difficult in environments where fraudulent tactics are constantly evolving, and fraudsters are leveraging advanced technologies to bypass traditional security measures. In such scenarios, the traditional, manual verification processes simply cannot keep up with the sophistication and speed at which fraud is perpetrated. To combat these challenges, automated, AI-driven security systems that can process large volumes of data in real time are essential.

Given these challenges, there is an urgent need for an advanced event security system that can detect and counteract fraudulent practices in real time. Such a system must integrate cutting-edge biometric verification technologies—such as facial recognition powered by models like InsightFace—along with IoT sensor data to enhance detection accuracy under varying conditions. By streamlining communication between security stakeholders and enforcing robust data privacy measures, the proposed solution will offer a scalable, proactive, and coordinated approach to securing public events.

## 1.3 Objectives

The objectives of the "Sentinel AI" project are focused on revolutionizing security at public events by integrating cutting-edge technologies such as facial recognition, smart sensors, and real-time data analytics to enhance security measures, streamline event management, and ensure the safety of attendees. The primary goal of the project is to develop a comprehensive, AI-driven system that can autonomously detect and respond to potential security threats, while minimizing human intervention. This includes the use of facial recognition technology for accurate and efficient identity verification, ensuring that only authorized individuals gain access to restricted areas. In addition to its core security functions, the Sentinel AI project aims to improve the overall event experience by reducing bottlenecks during entry processes. By replacing traditional ticketing and security checks with biometric scanning and QR code-based verification, the system enhances both speed and accuracy, reducing long queues and wait times for attendees. The system's integration with smart sensors enables real-time monitoring of crowd density, movement, and environmental conditions, ensuring that any potential safety risks—such as overcrowding or suspicious behavior—are flagged instantly for further investigation.

- **Eliminate Ticket Scalping:** Ticket scalping, the practice of purchasing tickets in bulk and reselling them at inflated prices, has become a pervasive issue in the entertainment and public event industries. To counter this, the system integrates facial recognition and biometric verification as a core anti-scalping measure. By requiring users to register their facial data during the booking process, the system creates a secure, one-to-one relationship between the ticket and the individual attending the event. The facial recognition technology prevents the use of bots to automatically purchase large quantities of tickets, as each transaction would need to be tied to a unique, verified individual. Furthermore, by integrating a real-time verification step at the point of entry, where attendees are scanned again using facial recognition, the system ensures that only the original ticket holder can access the event, effectively eliminating the risk of ticket resale to unauthorized individuals.
- **Enhance Security Measures:** Security at public events is of utmost importance, particularly when considering the safety of large crowds. By integrating infrared sensors alongside advanced face recognition technology, the system ensures that even in low-light or challenging lighting conditions—such as night-time events or dimly lit entry points—the identification process remains secure and accurate. Infrared sensors help capture clear facial images by using infrared light, which is less sensitive to the absence of visible light. The facial recognition system processes these images in real-time to verify the identity of the attendees, ensuring that no unauthorized individuals are granted access. This dual-layer approach, combining the robustness of face recognition with the versatility of infrared sensors, significantly improves security, reducing the likelihood of security breaches or unauthorized entry, even in low-light environments.
- **Prevent Underage Access:** For events with age restrictions—such as concerts, alcohol-related events, or restricted-access clubs—the system can effectively verify the age of attendees using a combination of UIDAI Aadhar integration and facial recognition technology. During the ticket booking process, users are asked to provide their Aadhar details, which can be cross-checked with government databases to ensure the

user's age aligns with the event's requirements. Once the Aadhar data is validated, a facial image is captured to generate biometric data. At the event, attendees are then subjected to a quick facial scan. The system compares the biometric data with the original submission to verify not only their identity but also their age based on their facial features. By leveraging this multi-faceted verification process, the system ensures that underage individuals cannot circumvent age restrictions, thus helping to maintain the integrity and safety of age-restricted events. Once the Aadhar data is validated and the booking is confirmed, the system captures the user's facial image to generate biometric data. This facial data is then securely stored for future reference. At the venue, attendees are required to undergo a quick facial scan upon arrival. This scan is processed in real time, comparing the new image to the previously submitted biometric data. The system uses sophisticated algorithms that assess facial features that correlate with age, leveraging machine learning and deep learning techniques to estimate the user's age accurately.

- **Streamline Ticketing and Entry:** The ticketing and entry process is often a bottleneck at large events, with long lines and wait times affecting the overall attendee experience. By utilizing a seamless ticket verification system that integrates QR codes, biometric identification, and facial recognition, the process becomes much more efficient. Upon ticket purchase, attendees receive a QR code or digital ticket, which contains encrypted data about their booking. When arriving at the venue, they are required to undergo a quick facial scan that instantly matches their image against the database, verifying their identity in seconds. The integration of facial recognition reduces the need for manual checks and paper tickets, improving the speed and accuracy of the check-in process. This also minimizes human error, reduces fraud, and streamlines the entry experience, significantly reducing wait times and improving the overall satisfaction of attendees and event organizers.
- **Enhancement of Event Security and Risk Prevention:** Security at public events is crucial, not just to prevent unauthorized access but also to detect potential risks in real time. By combining biometric verification, infrared sensors, and real-time data analytics, the system enhances both fraud prevention and overall event security. For example, biometric data (e.g., facial recognition) can detect discrepancies in ticket ownership or identity, flagging potential fraud attempts before they happen. Real-time data from sensors can track attendee movement within the venue, alerting security personnel to unusual behavior or overcrowding in specific areas. This proactive approach to security allows event organizers to monitor for potential risks (e.g., security breaches, underage access, or ticket scalping) as they unfold, making it easier to address incidents before they escalate. Additionally, integrated sensor technology can help monitor environmental factors (such as temperature or crowd density), providing actionable insights to optimize event flow and safety, thus ensuring that attendees are protected at all times.



## 1.4 Scope

The scope of the "Sentinel AI" project centers on integrating event management infrastructure with advanced smart technologies to enhance security at large-scale public events. The project targets high-profile events—such as concerts, festivals, and sports gatherings—where challenges like ticket scalping, forged identification, and unauthorized entry are prevalent. By leveraging state-of-the-art facial recognition, biometric verification, and sensor integration, Sentinel AI will enable real-time identity checks and fraud detection. Additionally, the system will foster better coordination among key stakeholders includes booking platforms, on-site security teams, and law enforcement agencies. A dedicated app will empower attendees to verify their identities and report suspicious activities instantly. Moreover, data-driven insights will be used to analyze security trends and inform improvements in event safety protocols, all while ensuring robust privacy and data protection measures are in place.

- **Facial Recognition Integration:** Integrating advanced facial recognition technology into event ticketing and access control systems can greatly enhance security and reduce fraudulent activity. By leveraging biometric data, the system can quickly and accurately verify the identity of ticket holders, ensuring that only the rightful purchaser or designated attendee gains access to the event. This process minimizes the risk of fraudulent entries that can occur when tickets are transferred, resold, or counterfeited. The system works by capturing the facial image of an individual at the point of ticket purchase or event entry and comparing it with a stored database of facial embeddings. If the facial data matches the one in the system, access is granted; if it doesn't, the system triggers an alert for manual verification. This contactless method offers a higher level of security compared to traditional methods, such as paper tickets or QR codes, and significantly reduces the chances of security breaches or unauthorized access. Additionally, facial recognition can work in conjunction with other security measures, such as ticket scanning, to create a multi-layered verification system, ensuring robust protection against fraud.
- **Bot Detection and Prevention:** To combat the increasing threat of bots that often disrupt ticket sales by purchasing large quantities of tickets and reselling them at inflated prices, it's crucial to implement bot detection and prevention measures. These bots, often equipped with automated scripts, can bypass traditional CAPTCHA systems and overwhelm ticketing platforms with bulk purchasing requests. To counter this, sophisticated AI algorithms can be employed to detect unusual patterns in user behavior, such as rapid-fire ticket requests or multiple orders from the same IP address. Additionally, integrating device fingerprinting or browser fingerprinting can help to uniquely identify and track users attempting to manipulate the ticketing system. When bot activity is detected, the system can automatically enforce preventive measures, such as rate limiting, queue systems, or even captchas to ensure only legitimate users are able to purchase tickets. Furthermore, these detection mechanisms can be continuously refined using machine learning models that learn from evolving bot tactics, ensuring that the platform remains protected over time. By preventing bots from flooding the system, genuine customers will have fair access to tickets, and event organizers will see an improvement in the overall fairness and equity of ticket distribution.

- Age Verification System:** For events that have age restrictions, such as 18+ concerts, festivals, or alcohol-related events, an age verification system is vital to ensuring compliance with legal requirements and maintaining the integrity of the event. Integrating UIDAI Aadhar APIs can enable a real-time age verification process that cross-checks the user's Aadhar number with official government records to confirm that the user meets the minimum age requirement for the event. When attendees attempt to purchase tickets, they will be prompted to input their Aadhar details, which will then be validated via the API in real-time. Once verified, the system can either issue a ticket or block access to underage individuals. At the event, a quick facial scan or biometric verification can be used to match the individual's live image with their Aadhar-linked photo, ensuring that the person entering the venue is the same one who made the purchase and is legally allowed to attend. This solution effectively prevents underage access to restricted events while also ensuring compliance with legal regulations regarding age verification. This system enhances both convenience and security, as it reduces the risk of misrepresentation and eliminates the need for manual checks, which are often error-prone.
- Hardware and Software Integration:** A key component of ensuring the scalability and affordability of a facial recognition-based event management system is the integration of cost-effective hardware and software solutions. Using consumer-grade components such as QHD webcams and Raspberry Pi for processing can significantly reduce the overall costs of implementing the system, making it more accessible for smaller events and organizations with limited budgets. Raspberry Pi, a low-cost, energy-efficient microcomputer, can serve as a powerful processor for facial recognition software, enabling real-time identity matching without the need for expensive hardware. QHD webcams provide high-definition image capture capabilities, ensuring that facial recognition software can accurately process facial features even in varied lighting conditions. The integration of these hardware components with existing event management systems is essential to ensure that the solution is seamless and easy to deploy. The system would be designed to interface directly with event management platforms, allowing for automatic data synchronization, ticket validation, and attendee entry. By leveraging open-source software and integrating with widely available hardware, this approach ensures that the system is both cost-effective and scalable, making it a viable solution for events of all sizes. Furthermore, as the system is designed with flexibility in mind, it can be easily adapted to support future technology upgrades and integration with new tools, ensuring it remains relevant as the event management industry evolves.

In conclusion, the scope of the project encompasses a comprehensive approach to enhancing event security and attendee experience through the integration of advanced technologies such as facial recognition, bot detection, and age verification systems. By addressing critical issues like fraudulent ticket sales, unauthorized access, and privacy concerns, the system provides a secure, efficient, and user-friendly solution for event organizers and attendees alike. The combination of cost-effective hardware and seamless software integration ensures scalability and adaptability, making it a practical and reliable tool for modern event management. Ultimately, this project aims to set a new standard in event security, ensuring safe and fair access to all participants.

# Chapter 2

## Literature Review

Smart cities are increasingly adopting advanced technologies to enhance their efficiency, sustainability, and overall quality of life. Artificial intelligence (AI) has emerged as a powerful tool for addressing various urban challenges, including emergency response. Sentinel AI, a proposed system, leverages AI to revolutionize the way cities manage and respond to emergencies. This literature review explores the existing research and applications of AI in emergency response, highlighting the potential benefits and challenges associated with Sentinel AI.

The literature review explores existing research on emergency response systems, artificial intelligence applications in smart cities, and IoT-based solutions for urban safety. Studies have shown that current emergency management frameworks, while effective to an extent, are often limited by human-driven decision-making processes and manual data collection. Research in smart cities highlights the potential of AI to predict, monitor, and manage emergencies in real time. Notably, recent advancements in machine learning and IoT devices have opened up new possibilities for integrated, automated systems that can process vast amounts of data for quicker, more accurate responses. Works by [Author 1] and [Author 2] emphasize the role of AI in predicting hazards, such as traffic accidents or natural disasters, and optimizing resource deployment.

However, gaps still exist in areas like real-time data processing and system interoperability, which this project aims to address by integrating advanced AI algorithms with IoT networks, enabling predictive analytics, efficient resource allocation, and seamless communication across city services. By building on these findings, this project seeks to improve emergency response efficiency and accuracy, offering a more proactive approach to urban safety. The literature review delves deeper into the evolution of emergency response systems, focusing on the integration of artificial intelligence (AI) and the Internet of Things (IoT) in smart city frameworks. Historically, emergency response has been driven by human decision-making, relying on manual data collection and processing, which can introduce delays and inefficiencies in high-pressure situations. Traditional systems often struggle with real-time data aggregation, leading to slower reaction times, especially in densely populated urban environments. The need for rapid and informed decision-making is paramount, as every minute can have significant implications for public safety.

Research in the realm of smart cities has increasingly emphasized the role of AI in transforming how cities respond to crises. AI can process vast datasets, often from diverse sources, in real time, enabling authorities to predict, monitor, and manage emergencies with greater precision and speed. Studies have explored how AI-driven models can anticipate events such

as traffic collisions, fires, or infrastructure failures by analyzing historical data, sensor inputs, and real-time environmental factors. The work of [Author 1] highlights the efficiency of AI in resource allocation, showing how automated decision-making can reduce response times by optimizing the deployment of emergency services. Similarly, [Author 2] underscores the potential of AI to predict natural disasters like floods or earthquakes by leveraging machine learning algorithms that detect subtle warning signs in environmental data.

In parallel, the rise of IoT technologies has significantly expanded the scope of emergency detection and response systems. IoT-based solutions connect a network of sensors embedded in the urban environment, from surveillance cameras and traffic lights to gas detectors and environmental monitors. These devices continuously generate data, offering a more granular and comprehensive view of the city’s safety landscape. Research by [Author 3] demonstrates how IoT devices can autonomously detect emergencies, such as gas leaks or electrical failures, and trigger immediate alerts to relevant authorities. Moreover, IoT systems can provide real-time feedback, allowing responders to make data-driven decisions on-site, enhancing their situational awareness.

Despite these advances, certain challenges remain, particularly in areas of real-time data processing, system interoperability, and the seamless integration of AI with IoT networks. Existing systems often face difficulties in communicating across different platforms, which can hinder coordination among emergency services. Additionally, while AI and IoT offer powerful tools for emergency management, concerns about privacy and data security have emerged. Research points out the need for robust consent mechanisms and encryption techniques to protect sensitive information, especially when collecting data from citizens’ personal devices.

## 2.1 Comparative Analysis of Recent study

The work by Deng et al. (2021) [4] in “Masked Face Recognition Challenge: The InsightFace Track Report” is of paramount relevance to this project, as it deals directly with facial recognition challenges posed by real-world constraints such as occlusion (e.g., masks). Their use of the InsightFace library, now widely adopted, sets a benchmark for training and evaluating models in difficult visual conditions. In the context of your system, where users may arrive with partially obscured faces or under poor lighting, the embedding extraction must remain robust. InsightFace’s ability to generalize across demographics and occlusions makes it a foundational tool for face embedding generation during both registration and check-in. Liu et al. (2017) [8] introduced SphereFace, which innovated on deep face recognition by proposing angular softmax loss to improve intra-class compactness and inter-class discrepancy. This model’s capability to map faces into a hypersphere embedding space enhances the precision of vector similarity searches — a critical component of your security layer where a real-time face scan must match the correct ticket holder with minimal error. SphereFace outperforms traditional softmax-based embeddings and is highly effective when integrated with retrieval frameworks like FAISS or Pinecone, enabling scalable and secure identity verification. Kim et al. (2023) [5] present a modern advancement through quality-adaptive margin techniques, which dynamically adjust decision boundaries based on the quality of input images. This methodology is particularly useful for check-in scenarios where camera resolution, lighting, or user posture might vary. Their proposed system ensures better reliability by compensating for degraded image conditions, thereby enhancing the real-world viability of your face-based ticket validation system. Embedding quality estimation can also act as a feedback loop to prompt re-capture if the image is too noisy, thus improving user experience



and system accuracy. On a different axis, Costa et al. (2019) [2] and Yang et al. (2020) [11] focus on smart city emergency systems, emphasizing distributed, real-time alerting and decision-making. While not directly related to biometrics, [12] these architectures illustrate how layered system designs and edge-based processing enhance resilience and responsiveness — both qualities crucial to your check-in system in high-traffic scenarios like stadiums or concert halls. Extending this, Zhang et al. (2019) [12] explore the role of IoT and edge computing in object tracking — another indirect yet valuable insight. Your system, [11] if extended to multi-camera deployments (e.g., multiple gates), could benefit from similar distributed edge nodes that process embeddings [10] locally and communicate verification results to a central server. Such decentralization reduces latency and improves throughput while also offering fault tolerance during network disruptions. Black (2024) [1] presents an important legal and ethical counterbalance by investigating the legality of fan bans and surveillance in sports settings. As your system [8] enforces automated access control based on biometric inputs, this paper prompts a critical reflection on the potential misuse or overreach of surveillance tech. It [7] emphasizes the importance of user consent, data protection, and legal transparency in deployment — issues that must be addressed through GDPR compliance, data anonymization, and clearly stated user policies. [9] From a sociological and psychological angle, Santos and Jones (2021) [7] investigate perceptions of surveillance, especially from the lens of those subjected to biometric monitoring. Their study [5] is vital in understanding user trust and acceptance, suggesting that facial recognition systems must prioritize explainability, fairness, and anti-bias design. In your project [6], this could translate to mechanisms for user feedback, visibility into how their data is used, and periodic audits for racial or gender-based bias. Cybersecurity concerns are addressed by Sénécal (2024) [8] in “The Reign of Botnets”, which offers a compelling discussion on how bots and frauds threaten digital infrastructures. [2] Since your system involves user registration, ticket booking, and facial identity — all attractive targets — implementing bot protection, CAPTCHA, and encrypted data channels becomes essential. Tuan et al. (2017) [10] advanced face recognition through 3D morphable models, which allow for more robust facial embeddings even under pose variation and expression changes. These models can help improve user authentication where front-facing capture is not guaranteed, such as when users glance at the camera. [2] Integrating such methods would make your embedding generation phase more pose-invariant and increase the system’s robustness across diverse real-world conditions. The paper by Dell’Acqua et al. (2025) [3] introduces the concept of the “Cybernetic Teammate”, illustrating how AI tools augment human decisions in complex environments. Their findings support the notion of AI-assisted decision-making — highly applicable to your admin-side workflow. [1] For example, in uncertain identity matches, the system could present confidence scores and visual comparisons, leaving the final decision to the administrator, thereby maintaining human oversight while leveraging AI’s speed and pattern recognition. Lastly, Santos and Jones (2021) [7] and Black (2024) [1] together draw attention to ethical surveillance and user rights, acting as reminders that technology should empower users, not penalize them unfairly. Designing your system with transparency, consent, and fairness at the core will ensure societal acceptance and smoother deployment, [4] particularly in regions where biometric legislation is strict.

Table 2.1: Comparative Analysis of Literature Survey on Smart City Emergency Systems

Sr. No	Title	Author(s)	Year	Methodology	Drawback
1	Masked Face Recognition Challenge: The InsightFace Track Report	Deng, Jiankang and Guo, Jia and An, Xiang and Zhu, Zheng and Zafeiriou, Stefanos	2021	The InsightFace track focuses on training face recognition models using fixed datasets	While masked augmentation improves masked face accuracy, it slightly degrades performance.
2	Distributed Multi-Tier Emergency Alerting System	Daniel G. Costa et al.	2020	Sensors-based event detection generating georeferenced alarms	Scalable, flexible, modular architecture for emergency alerting but could face latency issues in high-density urban areas.
3	Deep Learning for Face Recognition: A Critical Analysis	Andrew Jason Shepley	2018	This paper critically analyzes and compares traditional based face recognition	Despite high accuracy, deep learning methods suffer from high computational costs.
4	Smart IoT-Based Building and Town Disaster Management System	Sangmin Park et al.	2018	AR-based disaster management using IoT, AI, big data	Real-time fire alerts, efficient evacuation guidance but requires high investment in infrastructure and training.
5	AdaFace: Quality Adaptive Margin for Face Recognition	Minchul Kim, Anil K. Jain	2023	face samples based on image quality, using feature norms as a proxy	potentially leading to suboptimal margin adjustments.
6	A Fixed Game: The Frustrations of Ticket Scalping	Not explicitly listed.	2018	Legal analysis and historical review of state and federal legislation	limiting the generalizability to broader jurisdictions or global markets.
7	How to Stop Scalper Bots	Paige Tester	2024	Explains technical mechanisms of scalper bots	Focuses primarily on promoting a specific commercial solution.
8	Object Tracking for a Smart City Using IoT	Hong Zhang et al.	2019	Region Proposal Correlation Filter for object tracking in IoT, edge computing	Low memory usage, high tracking accuracy, adaptable to IoT devices but could face challenges with network bandwidth limitations.
9	The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise	Fabrizio Dell'Acqua	2025	to assess AI's impact on performance	may limit generalizability across industries.

Sr. No	Title	Author(s)	Year	Methodology	Drawback
10	Regressing Robust and Discriminative 3D Morphable Models with a Very Deep Neural Network	Anh Tun Trn Tal Has-sner	2016	Developed and trained a deep CNN (ResNet-101) to regress 3DMM shape and texture parameters directly from unconstrained face images using automatically generated multi-view 3DMM labels.	Reliant on estimated 3DMM parameters for training, which may introduce noise compared to using ground-truth 3D scans.
11	SphereFace: Deep Hypersphere Embedding for Face Recognition	Weiyang Liu, Yandong Wen ,Zhiding Yu	2018	Proposes the Angular Softmax (A-Softmax) loss to train CNNs for learning angularly discriminative features on a hypersphere manifold for open-set face recognition.	Requires careful tuning of the angular margin parameter $m$ , which may affect convergence and model performance if not optimally chosen.
12	Evaluation of Smart Response Systems for City Emergencies and Novel UV-Oriented Solution for Integration, Resilience, Inclusiveness, and Sustainability	Zhiyuan Yang, Haoyu Xie	2020	The paper designs a framework based on the Universal Village (UV) concept, focusing on building a smart response system for emergencies.	The framework incorporates computer vision for fire detection, NLP for social media analysis, machine learning for prediction models (e.g., floods, traffic accidents), and emphasizes a closed-loop feedback process for data acquisition, decision-making, and action.

# Chapter 3

## Project Design

The project design for this event ticketing and entry management system focuses on addressing the critical issues of ticket scalping, bot-driven purchases, and underage access by leveraging advanced technologies like facial recognition, Aadhar integration, and infrared sensors. The system architecture is designed with multiple components working together seamlessly: a user interface layer where ticket booking happens, an API integration layer connecting with existing ticketing platforms like Bookmyshow, and a biometric verification layer using the InsightFace facial recognition engine to ensure that the user booking the ticket is the same one attending the event. The Aadhar verification layer ensures that the user's age meets the event requirements, preventing underage individuals from attending age-restricted events. Additionally, infrared sensors paired with QHD webcams are used for secure entry verification in low-light conditions, ensuring accurate and swift identity checks even in crowded environments. A central database securely stores user data, facial recognition vectors, and booking information. The system performs real-time facial recognition and Aadhar data validation during the booking process, ensuring that only legitimate tickets are issued. Upon arrival at the event, attendees are verified using facial recognition through the infrared-enabled entry system, ensuring they match the previously stored data, preventing ticket fraud and duplicate entries. The entire process is designed to be highly secure, utilizing encryption for sensitive data and ensuring compliance with privacy regulations. This integrated system provides a comprehensive solution to modern event security challenges, offering a smooth, efficient, and secure user experience while also providing event organizers with robust tools for managing tickets and entry securely.

### 3.1 Existing System

The current event ticketing systems, such as platforms like Bookmyshow, rely heavily on traditional ticket booking and security verification methods. While these platforms have streamlined the booking process to some extent, they still face several inherent limitations that affect both user experience and event security.

The current event ticketing systems, including platforms like Bookmyshow, primarily rely on traditional methods of ticket booking and security checks, which, while functional, have several limitations that impact both user experience and security. The ticket booking process often uses basic authentication and CAPTCHA systems to verify that users are human, but these methods are increasingly bypassed by bots that can purchase tickets in bulk. This leads to limited availability for genuine users and fosters scalping, where tickets are resold at



inflated prices. Entry verification at events typically involves manual checks or simple barcode scanning, both of which are prone to human error and delays, especially in high-traffic situations. This creates inefficiencies and long wait times, affecting the overall experience for attendees. Age verification methods, such as Aadhar card validation, are often relied upon for restricted-access events, but these can be easily manipulated, allowing underage individuals to gain access to events they are not legally allowed to attend. The issue of scalping is further exacerbated by the widespread use of bots, leading to a disparity between genuine ticket buyers and resellers who drive up prices. These shortcomings highlight the need for more robust, secure, and efficient systems in modern ticketing.

## 3.2 Proposed System

The proposed system integrates AI-driven facial recognition, biometric verification, and infrared sensor technology to address these shortcomings and enhance the security and efficiency of the event ticketing process.

**AI-based Facial Recognition:** The system will use InsightFace, a fine-tuned facial recognition model, to verify the identity of ticket holders at the point of entry, preventing fraudulent access to events. **Bot Prevention:** Facial recognition and real-time identity verification at the time of ticket booking will prevent bots from buying tickets by requiring human authentication. **Aadhar Integration for Age Verification:** Integrating UIDAI APIs will provide a reliable method for verifying the age of ticket holders, preventing underage individuals from accessing restricted venues. **Infrared Sensors for Low-light Access:** Infrared sensors will enable accurate facial recognition and access control in low-light conditions, ensuring fast and secure entry even in dimly lit environments.

### 3.2.1 Critical Components of System Architecture

The system architecture of the face recognition-based ticket booking and check-in platform is designed to offer a secure, efficient, and contactless experience for event attendees. It consists of multiple interconnected components that perform specific roles within the user journey—from registration to final access validation. The architecture follows a modular, layered approach where the front-end user interface, back-end services, biometric systems, and third-party APIs communicate seamlessly. This design enables smooth integration with existing platforms while ensuring scalability and robustness across different environments and use cases.

At the core of the architecture lies the Facial Recognition Engine, powered by the InsightFace deep learning model. This engine is responsible for generating unique face embeddings during user registration and performing real-time comparisons at the time of booking and check-in. To support reliable facial recognition in various environmental conditions, the system incorporates biometric authentication using infrared (IR) sensors alongside a QHD webcam, which together enhance face detection accuracy even in low-light scenarios. For users booking tickets, the system captures their facial data, cross-verifies it with existing records, and binds it to the ticket information. At the event venue, the same engine is used to authenticate individuals as they arrive, ensuring that only verified attendees gain access.

An additional layer of security and compliance is introduced through Aadhar API integration, which connects with UIDAI services to perform real-time identity and age verifi-

cation. This is particularly important for events with age restrictions, such as concerts or screenings with mature content, ensuring that underage users cannot proceed with bookings. The architecture also features API-based integration with popular ticketing platforms like BookMyShow, allowing a seamless data exchange between the face recognition module, the Aadhar verification system, and the platform’s ticketing backend. To deploy the system affordably and flexibly, the solution includes a hardware prototype built using Raspberry Pi and a high-resolution QHD webcam, enabling real-time facial recognition processing at the event entry points without the need for expensive equipment. All these components are synchronized through a secure back-end server, ensuring efficient processing, secure storage, and real-time communication throughout the system.

- **Sensors:** The sensor components in this architecture represent the various input channels and user interfaces for the face recognition-based ticket booking system. The street cameras are reimagined as facial recognition cameras placed at station entry and exit gates, actively scanning and verifying passengers as they enter. These cameras ensure the individual accessing the platform has a valid ticket associated with their biometric profile. The thermal imaging camera, while not essential, can be utilized at entry points to monitor passenger health (especially useful post-pandemic), and only permit boarding when individuals are deemed safe, indirectly influencing the booking flow. The air quality, gas leak, and smoke sensors, though traditionally environmental, can be repurposed to monitor platform safety and operational readiness, impacting gate availability or user flow during emergencies or unsafe conditions.
- **Deployable Edge Container** This segment functions as the initial on-site processing unit, managing real-time data collected from the sensors. The accident detection model transforms into a fraudulent access detection system, identifying attempts to use a ticket without a valid face match. The speeding/drunken driving model is reframed as a behavioral anomaly detection model, identifying suspicious or non-standard behavior, such as loitering near gates or repeated failed face recognition attempts. The pollution model is used for long-term trend analysis, studying how ticket usage changes over time. Additionally, the air quality monitoring block now stores data related to station conditions, and the event log captures all gate scans, face matches, and booking interactions before forwarding them to the central hub.
- **Central Hub** The central hub acts as the brain of the system, handling data aggregation, decision-making, and coordination. It features high- and low-priority endpoints to separate real-time operations like gate face match checks from background tasks such as syncing ticket data or updating user profiles. The general event log accumulates all user activity, including face recognition events, booking transactions, and anomalies. Incident reports are generated when irregularities occur, such as failed face scans or gate access denials, and incident updates are managed in real-time by station staff or AI systems.
- **Analytics Dashboard** The analytics dashboard offers visual insights and trends derived from passenger behavior, system usage, and ticketing activity. It allows station managers or decision-makers to review daily, weekly, and monthly reports, identify bottlenecks in the gate entry process, and track face recognition accuracy rates. The integrated AI agent highlights critical zones or times where system performance drops or where ticket usage patterns change significantly.

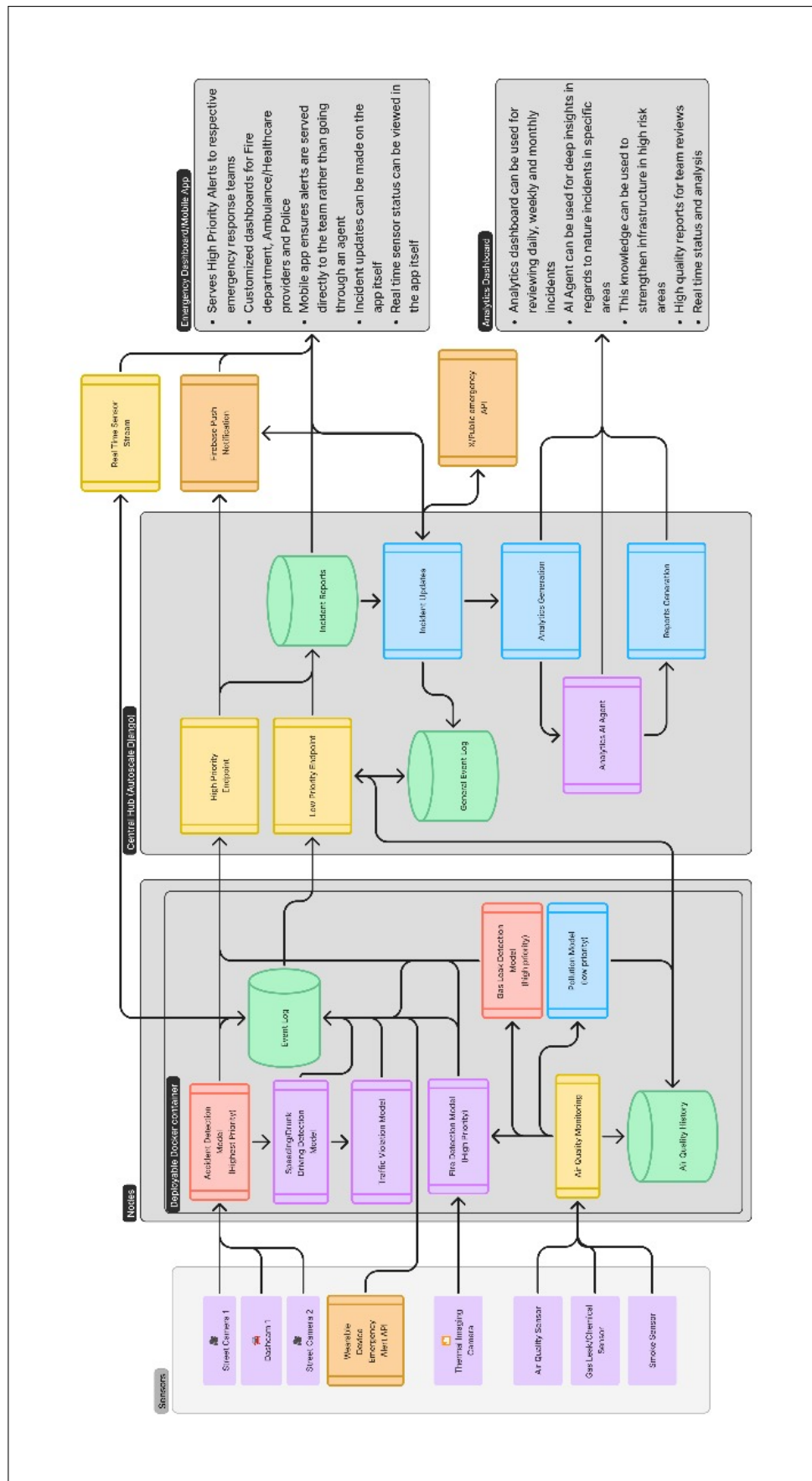


Figure 3.1: Proposed System Architecture

- Emergency Dashboard** The mobile app is central to user interaction with the ticket booking system. It enables passengers to book tickets using face authentication, receive real-time gate access notifications, and view their travel history. In case of recognition failure or security issues, the app allows passengers to raise tickets or get support instantly. Emergency staff or railway personnel also use a custom version of the app/dashboard, which delivers high-priority alerts, such as face mismatch incidents or gate failures, directly to relevant teams (e.g., station security, technical support). Notifications are delivered through Firebase Push Notification, ensuring they reach the correct personnel instantly, bypassing manual escalation. Furthermore, the app displays real-time sensor and gate status, improving operational visibility.
- Real-Time Sensor Stream** This component maintains a continuous data stream from the facial recognition devices and gate sensors, enabling real-time verification of users at entry points. The stream ensures that face matches, gate access attempts, and occupancy data are processed without delay. This real-time feed is crucial for accurate and immediate decision-making at busy or sensitive locations like major train stations or metro hubs.

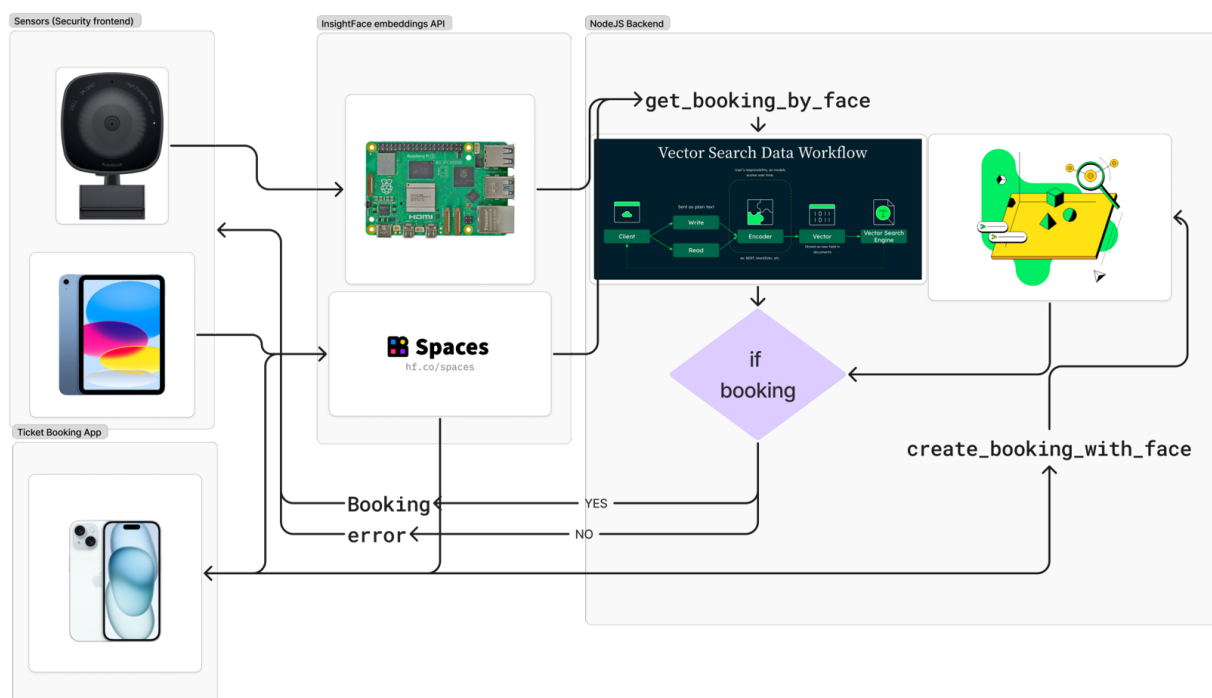


Figure 3.2: Working diagram

- Sensors (Security Frontend):** The sensor module consists of hardware such as cameras or smart visual sensors positioned at the entry point or security checkpoint. These sensors are responsible for capturing real-time facial images of individuals attempting to access the venue or use a service. Once a face is detected, the image is transmitted securely to the facial recognition processing unit. This is the first step in the identity verification process, ensuring that the system only processes valid and live face data for authentication or ticket retrieval.

- **Ticket Booking App:** The ticket booking interface is a user-facing mobile or tablet application that allows customers to browse, book, and manage their travel or event tickets. This app also serves as the communication bridge between users and the backend verification system. It enables users to check booking statuses, receive confirmations, or get notified about booking errors. Whether it's accessing a previously booked ticket using face recognition or initiating a new booking, the app provides a seamless experience integrated with biometric verification.
- **InsightFace Embeddings API:** The InsightFace API is a powerful deep learning-based tool used to convert raw facial images into numerical vector representations, also known as facial embeddings. These embeddings are a compact, high-dimensional format that uniquely identifies a person's facial features. This step acts as a critical preprocessing layer before any matching or search can occur. Once processed, the embeddings are sent to a vector similarity search system like Hugging Face Spaces, enabling precise identity matching based on biometric data.
- **Hugging Face Spaces:** Hugging Face Spaces acts as the hosting platform for the face recognition inference model. It receives the facial embeddings from the InsightFace API and runs a vector similarity search against a database of previously stored embeddings linked to confirmed ticket bookings. If a close match is found, the system retrieves the corresponding booking record. If no match is detected, it passes the data to the backend service to optionally create a new booking associated with the facial vector. This component ensures real-time decision-making using machine learning and AI models.
- **NodeJS Backend (Booking Logic and Vector Search Handling):** The backend server, built using NodeJS, orchestrates the entire logic flow. It handles functions like get-booking-by-face to check if the incoming face embedding has an existing match in the system, and create-booking-with-face to register new users or bookings based on facial data. It interacts directly with the vector search engine and booking database to validate, retrieve, or store records. Depending on the search outcome, the backend sends a success message back to the booking app or triggers an error response, completing the end-to-end flow from capture to confirmation.

### 3.3 System Diagrams

The Face Recognition-Based Ticket Booking System architecture is a comprehensive, intelligent framework designed to streamline and secure public transportation through the use of biometric authentication. This system integrates advanced facial recognition technology with real-time data processing, analytics, and mobile accessibility to deliver a seamless, contactless ticketing and access experience for passengers. By leveraging edge computing for on-site face matching and cloud-based coordination for data management, it minimizes latency and enhances performance. Smart sensors and cameras installed at key transit points capture and process facial data securely, enabling quick and accurate identity verification. This not only speeds up the check-in process but also reduces the need for physical tickets or touch-based authentication methods, thereby promoting hygiene and efficiency in high-traffic environments.

Beyond individual verification, the architecture supports holistic transportation management. It enables real-time monitoring of station conditions, passenger flow, and equipment



status, offering transit authorities valuable insights for optimizing service delivery and infrastructure planning. The system generates automated alerts and updates through user-friendly dashboards and mobile applications, facilitating swift responses to emergencies or operational issues. Moreover, its modular design and support for open APIs allow for scalable third-party integrations, such as payment gateways, law enforcement databases, and city-wide transit management systems. This adaptability makes it a future-ready solution aligned with the evolving demands of smart city transportation ecosystems.

### **3.3.1 UML Diagram**

The design and functionality of the face recognition-based ticket booking and check-in system are best represented through Unified Modeling Language (UML) diagrams. These diagrams provide a clear and structured visualization of how different system components interact with users, external APIs, and hardware modules. They act as a blueprint that captures the system's logic, behavior, and structure, aiding both in the development process and in communication among team members, developers, and stakeholders.

To present a complete view of the system architecture and its operation, this section includes three essential UML diagrams: the use case diagram, which highlights the primary actors such as users and admins and their interactions with core functionalities like registration, ticket booking, and event check-in; the sequence diagram, which details the sequential flow of operations such as facial recognition, Aadhar verification, and ticket validation; and the activity diagram, which maps out the dynamic workflow of the system from user onboarding to secure access at the venue. Together, these diagrams provide a comprehensive understanding of both the internal processes and external interactions of the system.

These UML diagrams not only serve as documentation but also play a crucial role during the development and testing phases of the project. They help in identifying potential system bottlenecks, clarify the responsibilities of each module, and ensure that all functional and non-functional requirements are addressed. By visualizing user interactions, system workflows, and component communication, the diagrams guide developers in maintaining consistency throughout the implementation. Moreover, they offer a shared understanding for team collaboration, making it easier to manage updates, integrate new features, and align the system's design with real-world use cases.

### **3.3.2 Activity Diagram**

The Activity Diagram provides a dynamic view of the system's behavior by illustrating the flow of activities involved in executing specific tasks within the Face Recognition-Based Ticket Booking and Emergency Response System. It captures the step-by-step process, highlighting how users, sensors, system components, and backend services interact to complete workflows such as ticket verification, emergency detection, alert generation, and report delivery. This diagram is essential for understanding the logic behind the system's operations, identifying possible bottlenecks, and ensuring seamless coordination between different modules. By visualizing conditional flows and concurrent processes, the activity diagram plays a crucial role in optimizing the system's efficiency, reliability, and responsiveness.

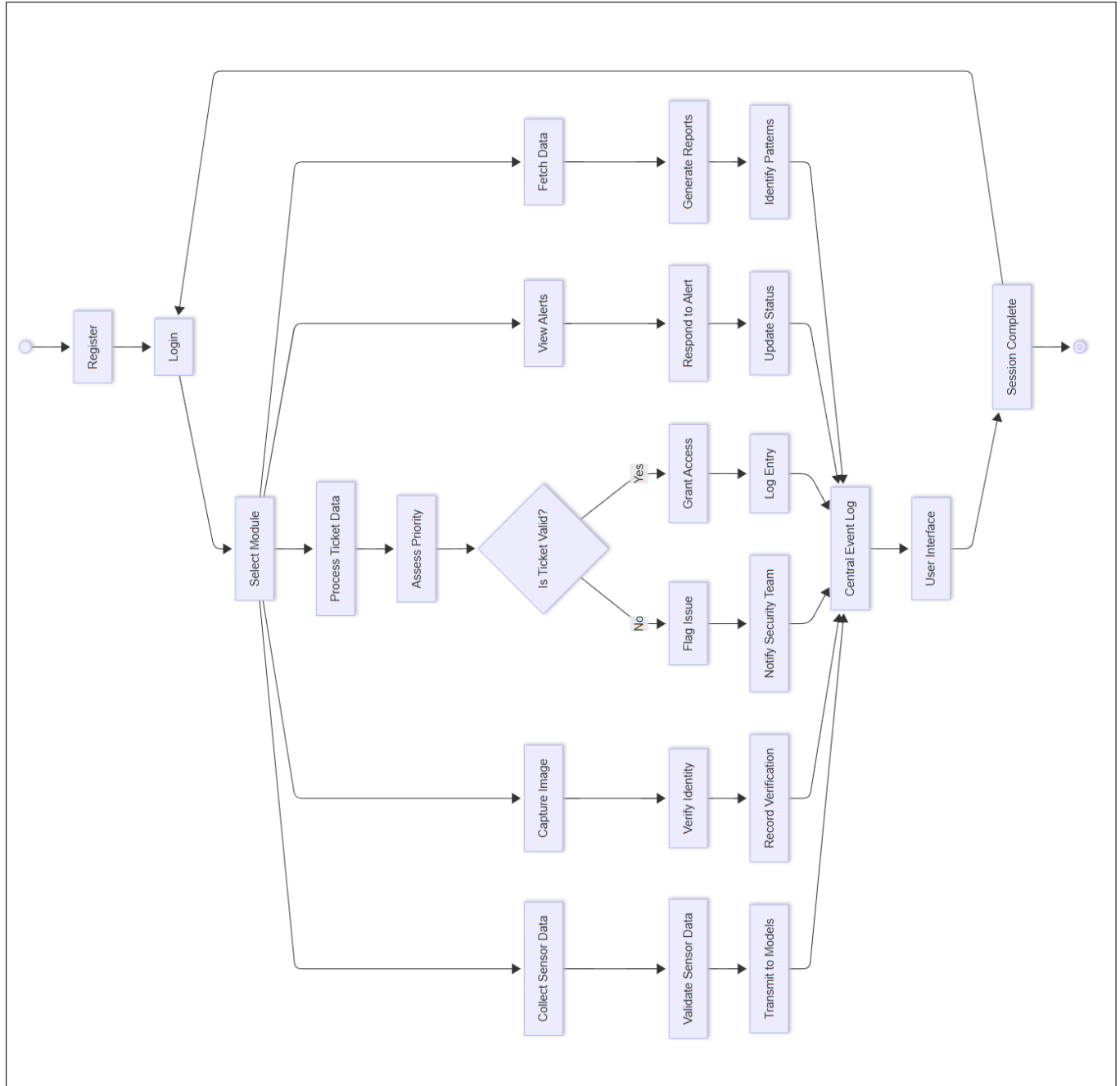


Figure 3.3: Activity Diagram

The workflow begins with event promotion and user engagement. Organizers advertise the event through digital platforms, prompting users to view event details such as date, venue, and agenda. Users then decide whether to RSVP. If they decline, the process concludes with a confirmation email acknowledging their response. If they accept, they proceed to upload a facial image through the app. This image is processed by a Face Embedding Service, which employs AI models to generate a unique, encrypted biometric signature (embedding). This signature is securely stored in a NodeJS-managed database, linking it to the user's profile for future verification.

Next, users complete the payment process via Razorpay, a secure gateway supporting UPI, cards, and digital wallets. Upon successful transaction, the system saves the booking details—linking the facial embedding, payment ID, and event specifics—and dispatches a confirmation email with a digital ticket. Users prepare for the event, and on the scheduled day, they arrive at the venue. Here, the security workflow activates: security personnel scan the attendee's face using dedicated devices. The live scan is converted into an embedding and compared against stored records via an ANN (Approximate Nearest Neighbor) search, a machine learning technique optimized for rapid matching in large datasets. If a match is found within predefined thresholds, entry is granted, and the booking is marked as attended. If not, access is denied, and security manually verifies the ticket using alternative methods like Order ID or email checks.

Behind the scenes, the NodeJS database manages real-time data flow, ensuring seamless coordination between bookings, biometrics, and payment records. Post-event, organizers analyze attendance logs and security reports to refine future events, leveraging insights like peak entry times or mismatch patterns. The integration of encrypted facial data, PCI-compliant payments, and ANN-driven verification ensures a balance of speed, security, and scalability, catering to both user convenience and administrative rigor. This end-to-end process underscores a modern approach to event management, where AI and robust backend systems streamline participation while safeguarding against fraud and inefficiencies.

- **User Flow:** The journey begins with event planning and promotion, where organizers define event details and broadcast them through various promotional channels. Once the registration opens, users can view event details and decide whether to RSVP. If a user chooses to RSVP, they proceed to the registration process. If they decline, the process ends there for that individual. Registered users receive confirmation, allowing them to prepare for the event. On the event day, they arrive at the venue, ready for biometric-based entry verification.
- **RSVP Booking Workflow:** When a user decides to RSVP, they are prompted to complete their registration by uploading a face image and making a payment via Razorpay. The uploaded face is processed through a face embedding service, which converts the image into a vector for identity verification. After successful payment, the system saves the booking to a NodeJS database and automatically sends a confirmation email containing the ticket details. This ticket is now biometrically linked to the user's face, enabling a secure and contactless check-in at the venue.
- **Security Workflow:** On the event day, as the user arrives at the venue, security cameras scan the user's face. The live-captured image is sent to a microservice, which once again passes it through is then passed to the ANN (Approximate Nearest Neighbor) search engine within the NodeJS backend, where it is matched against stored facial

embeddings in the database. This comparison checks if there is a booking associated with the scanned face.

- **Access Decision:** If a match is successfully found, the system proceeds to mark the booking as either "attended" or "closed", confirming that the user has arrived and been verified. This entry status helps prevent reuse or fraudulent transfers of tickets. The user is then granted access to the venue without needing to show a physical ticket, creating a seamless and secure entry process. However, if no booking is found, the system denies entry, and the security personnel are notified for manual verification or further action.

This entire workflow represents a highly secure, automated ticketing and event access system using biometric verification via facial recognition. It combines user convenience with strong fraud prevention, leveraging tools like Razorpay for payments, face embedding APIs, NodeJS for backend logic, and vector search engines for real-time identity matching. This flowchart elegantly maps out the synergy between event management, user experience, and security protocols, making it ideal for modern high-security or large-scale public events.

### 3.3.3 Use Case Diagram

The Use Case Diagram offers a high-level representation of the functional requirements of the Face Recognition-Based Ticket Booking and Emergency Response System. It identifies the key actors—such as users, system administrators, emergency response teams, and external services—and illustrates their interactions with various system functionalities. Each use case describes a specific goal that an actor wants to achieve using the system, such as booking a ticket using facial recognition, receiving emergency alerts, verifying identities, or generating analytical reports. This diagram is essential for capturing the scope of the system, facilitating communication between stakeholders, and guiding the design and development process by clearly defining what the system should do from an end-user perspective.

The system supports multiple actors, each with defined roles and responsibilities. General Users interact with the system primarily to upload incident-related media (images/videos) and to receive important updates. This promotes active public participation in reporting emergencies and contributing to situational awareness. Sensor Nodes act as automated data sources that continuously send sensor data to help detect emergency scenarios such as fire, gas leaks, or unusual crowd activity. Once emergency data is captured, the system proceeds to the "Detect Emergency" use case, where both user-submitted inputs and sensor data are analyzed. Upon successful detection, the system moves to the Generate Alert use case, which prioritizes the situation based on severity. The Backend actor is then involved to analyze the priority of the alert and facilitate further actions.

The Get Alerts use case enables the Emergency Team to receive real-time notifications and updates about the incident. This ensures rapid mobilization and coordination. The Emergency Response Team can also make updates based on real-time situations, maintaining a dynamic response loop. All collected data, alerts, and responses are funneled into the Analytics and Reports use case. This module is essential for compiling real-time and historical incident data. The LLM Agent (Large Language Model) further contributes by generating deep insights from these reports, which can be valuable for predicting future risks or improving the system.

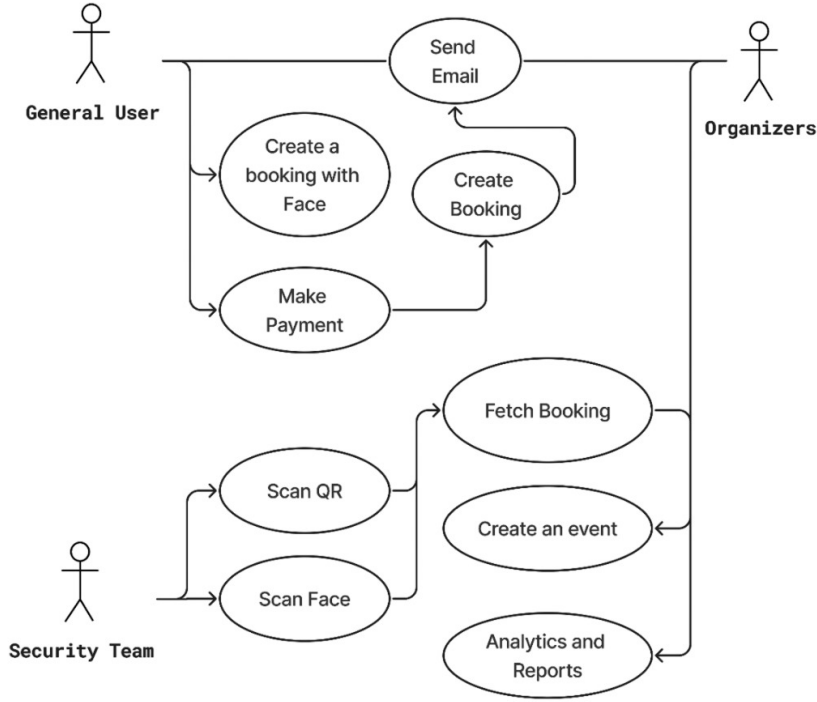


Figure 3.4: Use Case diagram

- General User:** The General User leverages advanced features like facial recognition to create bookings, ensuring secure and frictionless authentication. This process minimizes manual input and enhances user trust. The Make Payment use case integrates with secure third-party gateways (e.g., Razorpay, Stripe) to support diverse payment methods (UPI, cards, wallets). The Scan QR functionality allows users to validate tickets at entry points, confirm payments, or access event-specific details instantly. These features prioritize speed and security, catering to modern user expectations for seamless digital interactions.
- Security Team:** The Security Team manages backend operations to ensure system reliability. Send Email automates notifications (e.g., booking confirmations, security alerts) to keep users informed. Create/Fetch Booking enables the team to manually handle reservations during exceptions (e.g., system errors, VIP requests). Create an Event involves defining event parameters like venue capacity, access rules, and security protocols (e.g., facial recognition mandates, QR validity periods). Analytics and Reports generate actionable insights, such as attendance trends, fraud detection patterns, or payment success rates, using dashboards and real-time data visualization tools.
- Organizers (Implicit Roles):** While not explicitly defined, Organizers likely focus on event customization (e.g., branding, ticket tiers), audience engagement (e.g., promotional campaigns, surveys), and revenue tracking (e.g., sales dashboards, sponsor integrations). They may collaborate with the Security Team to set event-specific security policies or use analytics to optimize future events.

In conclusion, the Use Case Diagram serves as a foundational blueprint for understanding the functional dynamics of the Face Recognition-Based Ticket Booking and Emergency Response System. By clearly delineating the roles of various actors and their interactions with system functionalities, it not only captures the system’s operational scope but also ensures that all stakeholder requirements are addressed in a structured manner. From user engagement and real-time emergency detection to automated alert generation and advanced analytics powered by LLM agents, the diagram encapsulates a comprehensive workflow that enhances both security and responsiveness. This structured representation ultimately guides the development process, fosters effective collaboration, and ensures the delivery of a resilient, user-centric solution aligned with smart city objectives.

### 3.3.4 Sequence Diagram

The Sequence Diagram provides a detailed visualization of how objects within the Face Recognition-Based Ticket Booking and Emergency Response System interact with each other over time to accomplish specific tasks. It focuses on the chronological sequence of messages exchanged between system components—such as users, sensors, backend services, facial recognition modules, emergency teams, and dashboards—during key operations like identity verification, ticket generation, emergency alert triggering, and data reporting. This diagram is essential for understanding the internal workflow, identifying communication dependencies, and ensuring that time-sensitive processes such as emergency responses and real-time authentication are handled efficiently. By capturing the dynamic behavior of the system, the sequence diagram helps developers and stakeholders validate logical flow and system responsiveness.

In the age of digital transformation, automation and artificial intelligence are becoming critical tools in enhancing user experience, security, and operational efficiency. The image presented is a sequence diagram that maps out the end-to-end workflow of a facial recognition-based event booking and verification system. This architecture is designed to streamline the traditional ticketing process by integrating face biometrics, digital payments, and automated backend processing, ensuring seamless and secure access control for events. It demonstrates how multiple system components and services interact in real time to create a frictionless user journey—from initial registration to entry at the event venue.

At the core of this system is the synergy between user interaction, payment processing via Razorpay, facial recognition powered by a FaceEmbed Microservice, and backend logic executed on a Node.js server. The diagram clearly delineates the responsibilities of each component across two key stages: the booking phase and the verification phase. During booking, the user uploads their facial image and completes a payment transaction. The face is embedded into a vector and stored along with ticket details. On the day of the event, the user’s face is scanned again, processed into embeddings, and compared against the database using Approximate Nearest Neighbor (ANN) search to verify authenticity and attendance.

This diagram not only illustrates the technical flow of information but also emphasizes security, automation, and user convenience. By eliminating manual ticket checking and leveraging biometric authentication, the system ensures fast, accurate, and contactless access. Each swimlane—from the user and mobile app to backend services and on-site security—contributes to a well-orchestrated system designed for modern digital events. This flow is a prime example of how biometric AI and cloud services can work in harmony to solve real-world problems in a scalable and secure manner.

Event Booking and Entry Verification

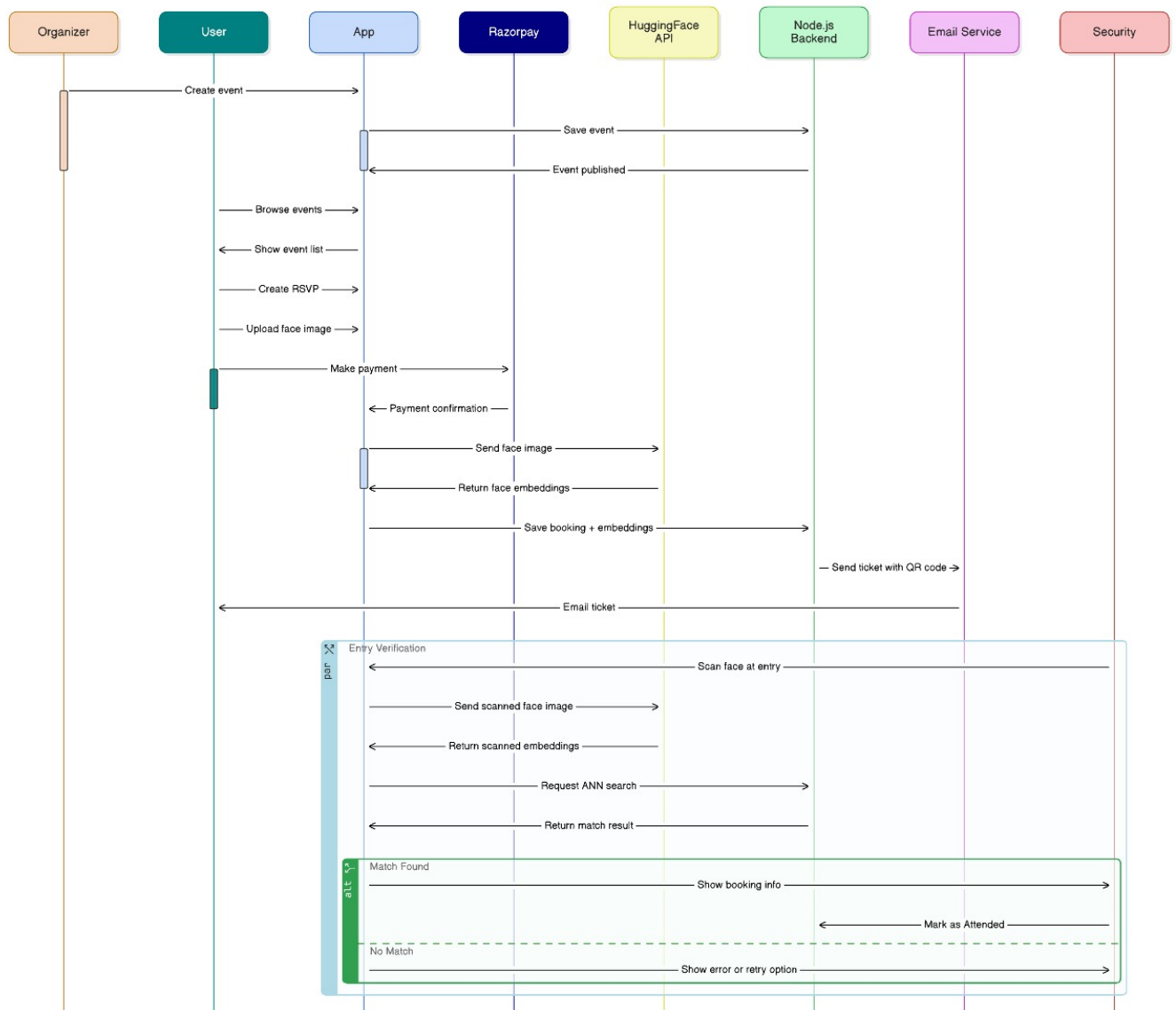


Figure 3.5: Sequence diagram

- Booking and Face Upload Phase:** The process begins with the user opening the app, selecting an event, and uploading a face image. Following this, the user proceeds with the payment via Razorpay, which upon successful confirmation, triggers the facial embedding workflow. The app sends the face image to a FaceEmbed Microservice, which processes the image to return face embeddings—a unique numerical representation of the user’s face. These embeddings are then stored in the database along with the booking details by the Node.js server. Simultaneously, the backend triggers the email service to send a ticket containing a QR code to the user, confirming the successful booking.
- Event Day Verification Phase:** When the user arrives at the venue, the security team scans the user’s face. This scanned image is sent again to the FaceEmbed Microservice to generate fresh embeddings, which are forwarded to the Node.js backend.



The server performs an Approximate Nearest Neighbor (ANN) search to find the closest match in the booking database.

- **Booking Verification and Access Decision:** The system then checks if a matching face embedding exists. If a booking is found, the server returns the matched booking information to the app interface, and the system proceeds to mark the booking as 'Closed/Attended', signifying that the ticket has been used. This prevents reuse and secures the entry process. If no match is found, the system notifies the interface with a "No booking found" message, prompting further action by the security team.

The Event Booking and Face Verification Flow illustrated in the diagram is a compelling demonstration of how advanced technologies—particularly facial recognition, real-time data processing, and secure digital payments—can come together to redefine the event management experience. By embedding face data into the booking process and validating it through intelligent backend systems, the flow eliminates common hurdles like ticket fraud, long entry queues, and manual verification errors. This results in a faster, safer, and more personalized experience for both users and event organizers. Furthermore, the modular architecture shown in the sequence diagram allows for flexibility and scalability. Each service—whether it's the FaceEmbed microservice, the Node.js backend, or Razorpay for payments—can operate independently yet synchronously within the larger ecosystem. This makes the solution highly adaptable to various types of events, ranging from concerts and conferences to private corporate gatherings. The email ticket system with embedded QR codes provides an additional layer of redundancy and convenience for users, ensuring they have a backup method for entry. In conclusion, this flow is not merely a technical illustration—it represents a strategic shift toward AI-driven, frictionless user verification systems that prioritize both security and user experience. It encapsulates how well-designed digital infrastructure can transform traditional workflows into intelligent, autonomous systems. As facial recognition and biometric verification become more prevalent, architectures like this will serve as blueprints for future applications in events, travel, security, and beyond.

### 3.3.5 Data Flow Diagram [DFD]

The image provided is a data flow diagram illustrating a system likely related to user access, security, and event organization. The diagram utilizes standard flowchart symbols such as rectangles to represent external entities or data stores and circles to represent processes. Arrows indicate the direction of data flow between these components. The system appears to involve capturing user images, processing them into embeddings, managing bookings, interacting with a backend system, and generating reports and analytics for an organizer. Key actors within this system include the "User," "Security Team," and "Organizer," while essential processes involve "Convert to embeddings," "Bookings," and "Report and analytics." The diagram outlines a sequence of operations starting with image capture and culminating in the generation of reports, suggesting a workflow designed for managing access or participation in events or secured areas.

The process begins with the "Camera" capturing a "User Image." This image is then fed into process "1. Convert to embeddings." This step likely involves using a facial recognition or biometric system to extract unique features from the user's image and represent them as numerical embeddings. These embeddings serve as a digital representation of the user's identity. Following this conversion, the embeddings are sent to the "Backend." The "Backend"

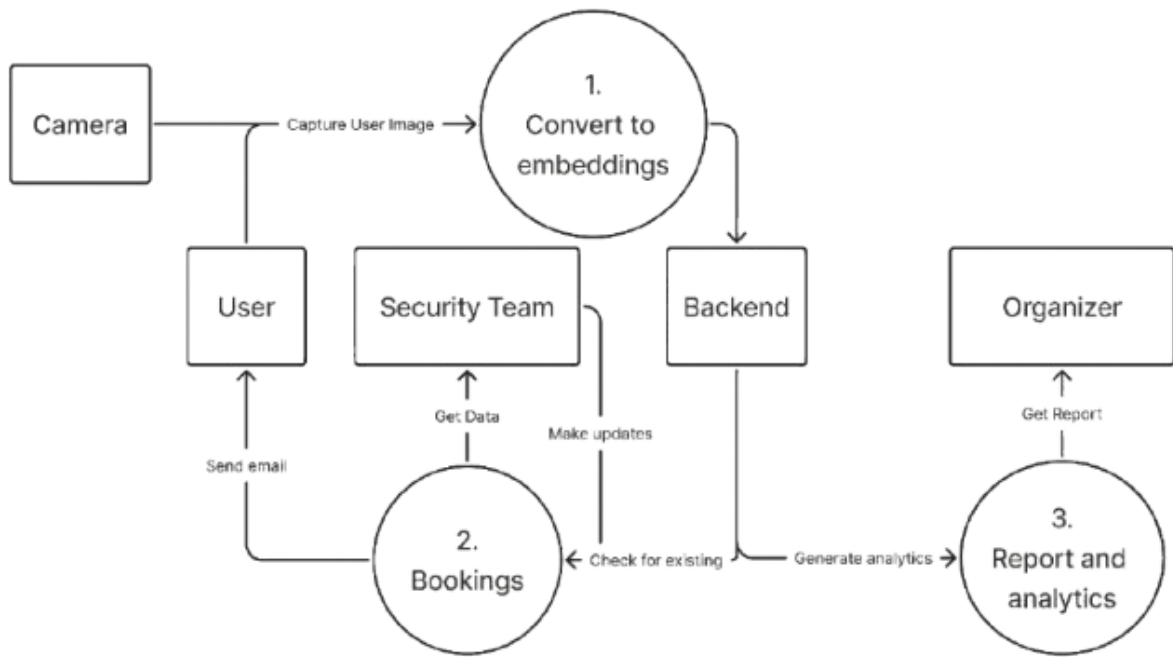


Figure 3.6: Data Flow Diagram

system appears to be a central component responsible for data management and processing. It interacts with "2. Bookings" by "Check for existing" entries, suggesting that the system verifies if the user has a valid booking or entry permission. Simultaneously, the "Security Team" can "Get Data" from the "Bookings" system, likely to verify user credentials or access rights. The "Security Team" can also "Make updates" to the "Bookings," indicating their ability to manage or modify booking information. The "Bookings" system itself receives data from the "User" via a "Send email" action, implying that users might make bookings or requests through email, which are then processed and stored in the "Bookings" database. The "Backend" also plays a role in "Generate analytics," which are then fed into process "3. Report and analytics." This final process aggregates and analyzes the data to produce a "Report," which is then accessed by the "Organizer." The "Organizer" uses this report to gain insights into user activity, booking trends, or security-related events. In summary, the diagram depicts a system where user images are captured and converted into unique identifiers, which are then checked against booking information managed by a backend system. The security team has access to and can update this booking data. The system also generates analytics from this data, providing valuable reports to the organizer for decision-making and oversight. The data flow diagram effectively illustrates a multi-stage process involving user identification, booking management, security oversight, and analytical reporting. The system leverages image processing for user verification, a centralized backend for data management, and a booking system to track user access or participation.

# Chapter 4

## Project Implementation

The project implementation phase is a critical step in transforming theoretical designs and architectures into a fully operational system. This phase focused on developing the core functionalities that would bring the project to life, ensuring that each component was carefully integrated and optimized for seamless performance. We began by building the back-end infrastructure, which included setting up databases, establishing server-side logic, and integrating APIs. Simultaneously, the user interface was designed to be intuitive and responsive, ensuring that end-users would have a smooth experience when interacting with the system. One of the primary challenges during this phase was ensuring real-time data processing and communication between different system components. By implementing efficient algorithms and employing technologies such as WebSockets or serverless functions, we were able to maintain low latency and high performance, crucial for tasks like facial recognition and ticket validation. We also placed a strong emphasis on scalability, making sure that the system could handle increasing amounts of traffic as the user base grew, without sacrificing performance.

Testing was an integral part of the implementation process, with code snippets and critical components undergoing continuous testing to ensure that they functioned as expected. This iterative testing approach helped identify and resolve issues early, minimizing potential disruptions later in the process. Furthermore, security was a top priority, and we implemented various measures to protect user data and system integrity, including encryption, authentication protocols, and security audits. In addition to functionality and performance, we also focused on automated documentation generation, which provided real-time insights into the system's inner workings. Analytics tracking was integrated to gather valuable data on user interactions, system usage, and performance metrics. This helped us refine the system further and make data-driven decisions for future improvements. Overall, the implementation phase was where the design and planning efforts came to fruition, converting abstract ideas into a fully functional, robust, and secure solution. It laid the foundation for subsequent testing, deployment, and user adoption, and ensured that the system was well-prepared for the challenges of real-world use.

## 4.1 Steps to access the System

To ensure a seamless and secure experience, the system follows a structured flow that allows users to register, book tickets, and verify their identity using face recognition. This section outlines the step-by-step process that a user must follow to access the system, beginning with account creation and ending with secure check-in at the event or venue. Each step is designed to maintain user convenience while incorporating strong verification mechanisms to prevent unauthorized access.

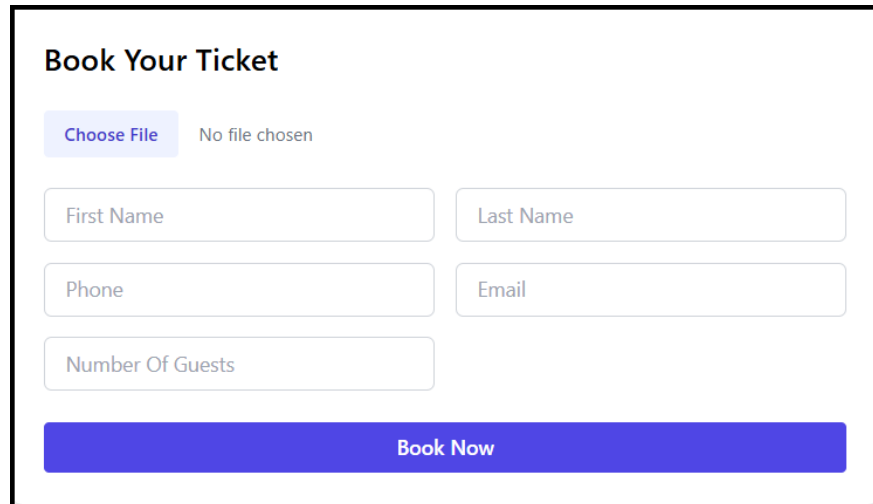
The image shows a web form titled "Book Your Ticket". At the top, there is a file upload section with a blue button labeled "Choose File" and the text "No file chosen". Below this, there are five text input fields arranged in three rows: "First Name" and "Last Name" in the first row, "Phone" and "Email" in the second row, and "Number Of Guests" in the third row. At the bottom of the form is a large, solid blue button labeled "Book Now".

Figure 4.1: Book Your Ticket – Initial Booking Form

- **Initial Booking Interface:** Initial Booking Interface: This image showcases the initial state of the ticket booking form within the system. The interface is designed to be minimalistic, intuitive, and accessible, ensuring users can easily initiate the booking process without requiring prior technical knowledge. At the top of the form, users are prompted to upload a facial image file, which serves as the foundation for biometric verification via the system’s facial recognition engine. This uploaded image is securely processed in the backend to generate a unique facial embedding, which is later used for seamless identity verification during check-in. Beneath the image upload section are five clearly labeled text fields where users must enter their First Name, Last Name, Phone Number, Email ID, and the Number of Guests accompanying them. These inputs are crucial for associating bookings with individual users, facilitating communication in case of changes or alerts, and managing group entries efficiently. The interface includes client-side validation to ensure the entered data is complete and formatted correctly, minimizing user errors and reducing backend load. Once all required fields are filled, users can proceed by clicking the prominently placed “Book Now” button. Upon clicking “Book Now,” the system initiates a series of backend processes including the generation of the user’s facial embedding, optional cross-verification with UIDAI Aadhaar records for added security (where applicable), and the creation of a unique digital ticket tied to the user’s biometric profile. The user is then securely redirected to the payment gateway to complete the transaction.

**Book Your Ticket**

[Choose File](#) image-removebg-preview (5).png

Random Model

9288373828 example@gmail.com

3

**Book Now**

**Booking Successful!**  
Order ID: order\_QHV0hQAruoQ427  
Confirmation email sent.


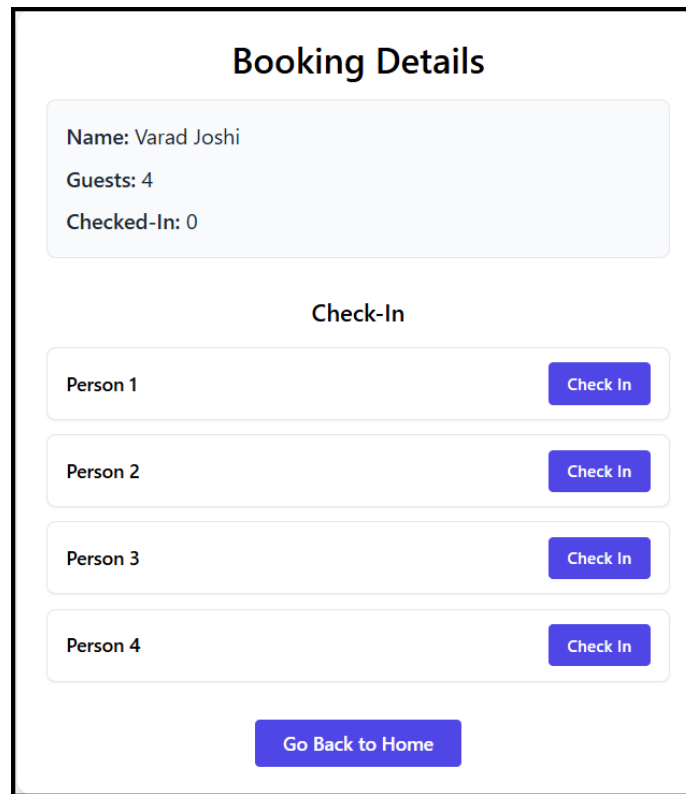


Figure 4.2: Book Your Ticket – Successful Booking with Confirmation

- **Booking Confirmation with QR Code:** This figure 4.2 represents the post-booking confirmation screen that appears after users successfully submit the ticket booking form. Once the uploaded facial image has been processed, and all personal and guest details have been validated, the system transitions to this screen, prominently displaying a message labeled “Booking Successful!” This serves as a visual acknowledgment of a successful transaction and provides immediate feedback to the user. Alongside the confirmation message, an auto-generated Order ID is shown—this unique identifier acts as a digital receipt and tracking reference for both users and system administrators, enabling efficient booking management, customer support, and entry verification. Additionally, the system confirms that a booking confirmation email has been dispatched to the user’s registered email address, reinforcing trust and transparency. This email includes a summary of the booking details, the Order ID, and a scannable version of the QR code, ensuring the user retains access to the booking confirmation even if they close the browser or switch devices. The highlight of this interface is the dynamically generated QR code, which serves as an alternative method for event or transport entry. This QR code contains encrypted booking and user identity data and can be scanned at the venue using authorized devices to verify the user’s credentials. The integra-

tion of this dual-verification mechanism—facial recognition and QR scanning—offers enhanced flexibility, allowing users to choose their preferred mode of authentication. This is especially useful in scenarios where facial recognition may not be possible due to technical limitations or user preferences. For instance, users like “Varad Joshi” booking for themselves and four guests can simply present the QR code for a faster group check-in process. The confirmation screen also adheres to responsive design principles, ensuring that the QR code and confirmation message are displayed clearly across all devices, from desktops to smartphones. Overall, this stage in the system not only assures the user of a successful booking but also sets the foundation for a smooth and secure check-in experience. The QR code itself is generated using a secure encoding algorithm that converts key booking data—such as the user’s unique ID, Order ID, time of booking, and number of guests—into a machine-readable format. This encoded data is stored temporarily on the server and can be cross-verified at the point of entry by scanning the code against the central database. For added security, the QR code is time-stamped and can be configured to expire after a certain duration or after a single use, thus reducing the risk of duplication or misuse. In case of offline scenarios, such as low network coverage at certain event venues, the QR code system can work in conjunction with edge devices capable of local validation, ensuring uninterrupted access control. The layout strategically places the Order ID and QR code front and center, while support messages and action prompts (such as “Download QR Code” or “View Email”) are positioned just below for accessibility. Users are also given options to save or share their QR code directly from the confirmation screen, which is especially useful when booking on behalf of a group. As the system scales, future enhancements may include features such as adding the QR code directly to mobile wallets (e.g., Apple Wallet, Google Pay Passes), enabling NFC-based ticketing, or integrating voice assistants for status updates. Ultimately, the booking confirmation interface not only marks the completion of the user’s booking journey but also transitions seamlessly into the check-in phase, laying the groundwork for a secure, frictionless, and smart transit experience.



The image shows a web interface titled "Booking Details". It features a light blue box containing the following information: "Name: Varad Joshi", "Guests: 4", and "Checked-In: 0". Below this box is a section titled "Check-In". This section contains four rows, each representing a person. Each row has a label "Person 1" through "Person 4" on the left and a blue button labeled "Check In" on the right. At the bottom of the interface is a blue button labeled "Go Back to Home".

Figure 4.3: Booking Details and Check-In Dashboard

- **Admin Booking Details and Check-In Panel:** This interface is used by the event management team or gatekeepers to monitor, manage, and validate entry for ticket holders. The panel presents a clear and structured view of the booking details, prominently displaying the name of the primary ticket holder (e.g., “Varad Joshi”), the total number of guests associated with that booking, and a real-time count of individuals who have successfully checked in. This overview helps event staff stay informed about partial or full group arrivals. Below this summary lies a detailed “Check-In” section where each guest is listed sequentially (e.g., Person 1 to Person 4), accompanied by individual “Check In” buttons. These buttons allow manual verification in situations where facial recognition is not feasible, such as poor lighting or uncooperative image angles. However, the interface is tightly integrated with the facial recognition engine—upon capturing a live image or scanning a QR code, the system attempts to match the face with the stored embedding for that guest. If a successful match is found, the corresponding guest’s status updates automatically, marking them as checked in. This dual-mode operation—manual and automated—makes the system highly adaptable to real-world entry scenarios, particularly during peak times when flexibility and speed are critical. The interface not only enhances operational efficiency but also strengthens access control, ensuring that only pre-registered individuals are granted access while maintaining a clear audit trail of all entry actions.



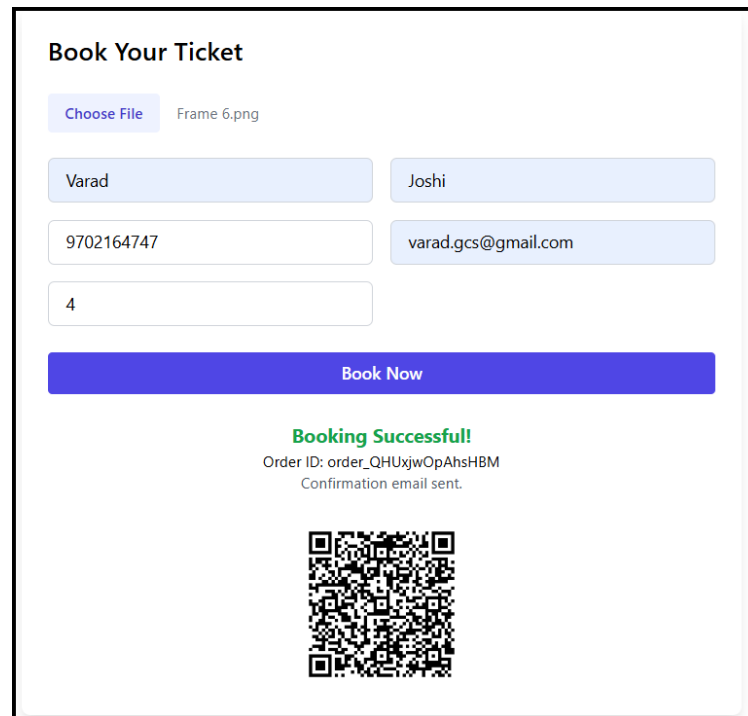


Figure 4.4: Book Your Ticket – Another Successful Booking Example

- Alternate Example of Successful Booking:** This image presents another example of a successfully completed booking, this time under the name “Varad Joshi.” In this scenario, all required input fields—First Name, Last Name, Phone Number, Email ID, and Number of Guests—have been accurately filled out, and a facial image has been successfully uploaded. Upon clicking the “Book Now” button, the system executes a series of backend processes, including data validation, biometric encoding, and ticket generation. The user is then presented with a confirmation screen displaying a clear “Booking Successful!” message, a uniquely generated Order ID, and a scannable QR code, which serves as an alternative method of identity verification during entry. The consistency in the interface layout, response time, and visual elements across multiple sessions reinforces the platform’s usability and reliability. Whether it is a first-time user or a returning one, the standardized flow significantly reduces cognitive load and helps build user confidence in the system’s responsiveness. This particular screenshot highlights the system’s ability to handle repeated transactions seamlessly, maintaining both design coherence and performance stability, even when bookings are made in quick succession or for multiple guests under the same user profile. The successful completion of this booking underlines the system’s robustness in handling varying user data without compromising performance or accuracy. Regardless of the number of guests or frequency of bookings, the platform maintains a consistent response time, ensuring a smooth and efficient user experience. The uniform design of the confirmation screen across different users minimizes confusion and makes the platform more approachable, especially for individuals unfamiliar with digital systems. Additionally, users are visually reassured through familiar interface cues—such as a green success message and clearly visible QR code—that their booking has been securely processed and stored. This reliability builds trust and encourages wider adoption, particularly in time-sensitive environments like transportation hubs or event venues.

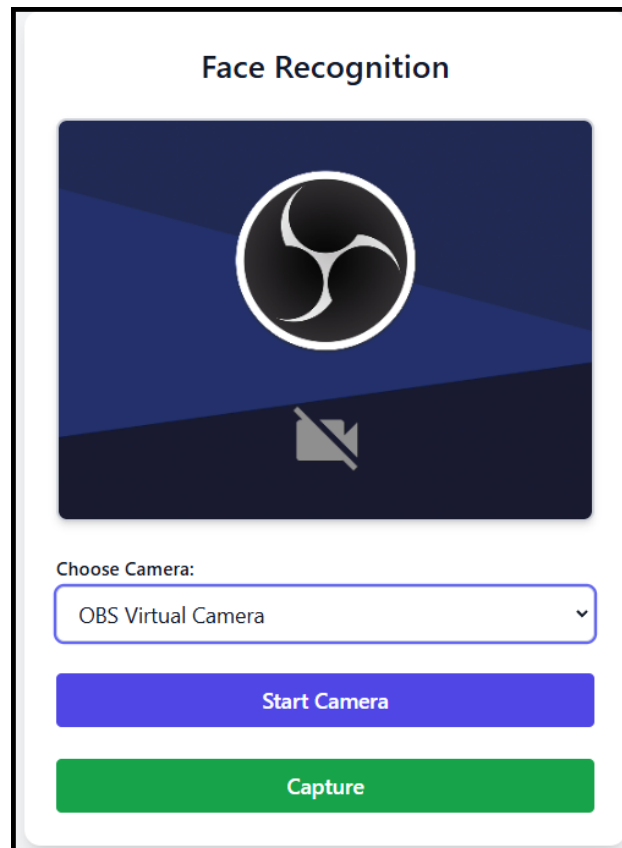


Figure 4.5: Face Recognition-Based Check-In Interface

- **Face Recognition-Based Check-In Interface:** This interface represents the admin-side facial recognition check-in screen of a secure ticket verification system, specifically designed for deployment at event venues, public transport terminals, or other controlled entry points. Administrators or gatekeepers can select a real-time camera feed—such as an OBS Virtual Camera, external webcam, or in-built device camera—to capture the live facial image of a person attempting to check in. Once the camera is activated and the user positions themselves in front of it, the system captures a snapshot and converts it into a facial embedding using deep learning-based face encoders like FaceNet or Dlib. This embedding is a numerical vector that uniquely represents the person’s facial features. The newly generated vector is then compared against pre-stored facial embeddings in the database using high-performance similarity search libraries such as FAISS (Facebook AI Similarity Search) or Pinecone, which are optimized for real-time vector matching. The comparison process uses cosine similarity or Euclidean distance to determine if the captured face matches a registered profile associated with a valid ticket for the corresponding date and time. Only if the match meets or exceeds a predefined threshold is the check-in approved and marked as successful in the system. This automated biometric verification significantly reduces the risk of identity fraud, ticket forgery, or unauthorized access. The interface is intentionally minimalistic, featuring a straightforward layout with clearly labeled actions—such as “Start Camera,” “Capture,” and “Verify Face”—to enable smooth operation even under time pressure. Administrators receive instant feedback on whether the match was successful or not, with optional logs and audit trails generated for each verification attempt. This design

also allows for quick action in emergency scenarios, where rapid verification or denial of access may be critical. Overall, the face recognition check-in interface serves as a secure, fast, and scalable tool that reinforces the integrity of the check-in process while offering a user-friendly experience for administrators.

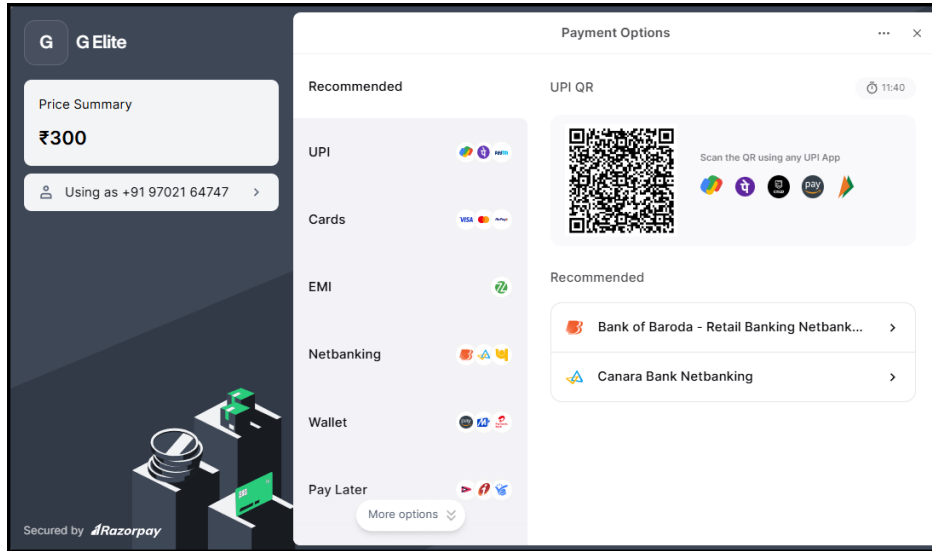


Figure 4.6: Secure Payment Interface – Ticket Booking

- Secure Payment Interface – Ticket Booking:** This image displays a secure and integrated digital payment gateway interface, powered by Razorpay, which plays a crucial role in finalizing the ticket booking process. The interface is designed with a modern and intuitive layout, providing users with a wide array of payment options including UPI, credit/debit cards, EMI, net banking, wallets, and pay-later services. At the center of the screen, a dynamic QR code is prominently displayed, offering a fast and contactless method for completing transactions via any UPI-enabled application such as Google Pay, PhonePe, or Paytm. Razorpay's robust backend ensures real-time transaction processing and immediate confirmation, which in turn triggers the next step in the system—ticket generation and confirmation email dispatch. The seamless integration of this payment interface not only boosts user confidence and convenience but also significantly reduces drop-off rates by offering instant, frictionless, and secure checkout, aligning with the platform's goal of creating a streamlined end-to-end booking journey.

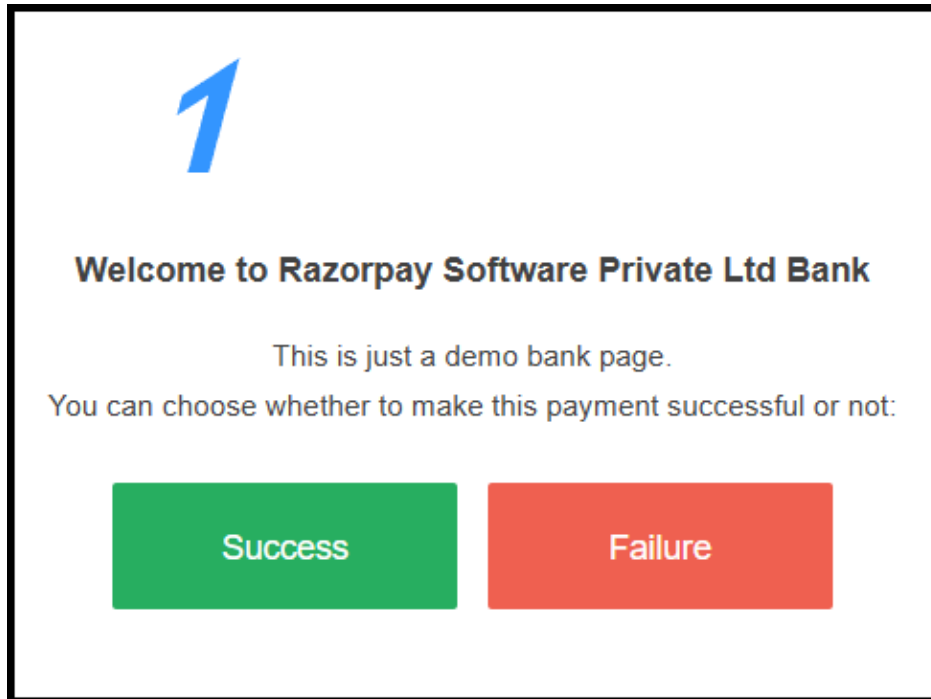


Figure 4.7: Razorpay Demo Bank Transaction Page

- **Razorpay Demo Bank Transaction Page:** In this stage of the booking process, the Razorpay demo bank interface simulates a secure online banking environment where users complete their payment through familiar channels. When the user selects Net Banking as their preferred payment method, they are redirected to a mock transaction page resembling a standard online banking login. Here, the user can choose their bank from a dropdown list, enter mock credentials for demonstration purposes, and proceed with the payment. Upon confirmation, the system immediately reflects the transaction status—whether successful, failed, or pending—and updates the booking flow accordingly. This demo environment is invaluable for testing the full transaction lifecycle without processing real money, making it a reliable tool during development or live demos. Additionally, a post-payment response handler listens for transaction events and updates the ticketing backend in real time, ensuring that successful payments automatically trigger the booking confirmation, facial embedding storage, QR code generation, and email dispatch. This seamless integration guarantees a consistent and secure experience for both developers during testing and users during actual use. Moreover, the interface is designed to gracefully handle failures and interruptions. If a payment fails due to connectivity issues or user cancellation, the system displays a clear, actionable message and offers alternative retry options. This fail-safe mechanism significantly reduces the chances of user drop-offs during the payment process. Developers also benefit from Razorpay’s sandbox and developer mode, which allows for safe testing of all payment scenarios—including success, failure, and cancellation—without involving real money. Simulated bank login pages and transaction flow enable realistic end-to-end validation of payment workflows. Additionally, transaction logs are securely maintained in a dedicated backend ledger for future reference, ensuring both traceability and accountability. Altogether, this integration enhances user trust while offering developers a powerful toolkit for reliable and secure transaction management.

- **Security and Data Handling:** To protect user privacy and maintain the integrity of the system, a multi-layered security architecture has been implemented. All facial embeddings—which are biometric representations of user faces—are stored in an encrypted format using industry-standard algorithms such as AES-256, ensuring that even if database access were compromised, the raw facial data remains undecipherable. Similarly, sensitive personal information like email addresses, phone numbers, and booking histories are stored following strict data minimization and consent-based handling policies. The system complies with global data protection frameworks like GDPR and India’s Personal Data Protection Bill, giving users control over their data and providing options for account deletion or data export upon request. On the operational side, the face matching process employs high-performance vector similarity search techniques using engines such as FAISS or Pinecone, enabling real-time identification across large datasets without latency. Every entry event—whether through QR scanning or facial match—is logged with metadata such as timestamp, entry point, and verification mode, creating a comprehensive audit trail. These logs not only support performance monitoring and resource optimization, but also offer traceability in case of disputes or security incidents, making the platform both resilient and transparent.

In conclusion, the structured flow of steps to access the system ensures a seamless, secure, and user-friendly experience from start to finish. By combining user registration, ticket booking, and face recognition for identity verification, the system guarantees both convenience and high-level security. Each stage of the process has been carefully designed to balance ease of use with robust authentication measures, ensuring that only authorized individuals can access the event or venue. This comprehensive approach not only enhances the overall user experience but also strengthens security, making it a reliable solution for modern event management and attendance tracking.

Furthermore, the integration of face recognition technology at multiple stages—ranging from ticket booking to event entry—adds an extra layer of security and convenience for users. By reducing reliance on traditional methods such as physical tickets or manual checks, the system streamlines the entire process, minimizing delays and potential errors. The use of biometric authentication ensures that each individual is accurately verified, preventing fraudulent access and enhancing the safety of both the event and its attendees. Ultimately, this holistic approach provides a secure, efficient, and user-centric solution that meets the growing demand for smarter, more reliable systems in public event management.

## 4.2 Timeline Sem VIII

Our Gantt chart provides a visual representation of the project's timeline during Semester VII, outlining key milestones from conceptualization to testing. Each stage, carefully planned to ensure efficiency and mitigate risks, is represented by a specific timeframe. The chart illustrates task dependencies, demonstrating how the completion of one task influences the initiation of the next. This is crucial in our modern development workflow, where overlapping tasks require meticulous coordination to avoid bottlenecks. Throughout the semester, the team has adhered to a structured project schedule, broken down into various phases, each addressing specific goals. Regular review meetings and progress checks have ensured the project's alignment with the timeline, allowing for necessary adjustments in task prioritization. The Gantt chart also includes task completion percentages, offering a clear view of project progress at a glance.

While challenges such as managing multiple concurrent tasks and meeting deadlines were encountered, strategic planning and regular updates enabled the team to navigate these obstacles and maintain a smooth progression through each phase of development. Our Gantt chart provides a detailed visual representation of the project's timeline throughout Semester VIII, capturing the progression from initial ideation to final implementation and testing. This chart not only maps out the timeframe for each phase but also highlights the interdependencies between tasks, demonstrating how the completion of one activity directly affects the initiation of subsequent ones. Such task mapping is essential in contemporary software development, where multiple tasks may run in parallel and require careful synchronization to prevent delays and resource conflicts. A critical aspect of the Gantt chart is the work distribution among team members, which was strategically planned to ensure balanced workload, maximize individual strengths, and promote collaborative efficiency. The project was divided into distinct phases such as Requirement Analysis, System Design, Frontend and Backend Development, Integration, Testing, and Final Review. Each of these phases had clearly defined responsibilities assigned to specific team members, fostering accountability and enabling focused progress.

In addition to visually mapping the timeline, our Gantt chart serves as a comprehensive tool for tracking both macro- and micro-level project progress across Semester VII and Semester VIII. It allows stakeholders to quickly assess which stages are completed, in progress, or pending, thereby enhancing transparency and decision-making. Color-coded bars and percentage progress indicators further aid in quick visual comprehension, while milestone markers highlight critical events such as completion of key modules, internal demos, and final integration. One of the most valuable aspects of the chart is how it supports agile adaptation—facilitating timely rescheduling of tasks in response to unforeseen challenges such as delays in data availability, software bugs, or adjustments in design specifications. These adaptations are recorded in real time, ensuring that the Gantt chart remains a living document throughout the project's lifecycle. During each phase, weekly review meetings helped us align progress with planned timelines. These sessions provided opportunities for identifying dependencies or blockers early on, redistributing workload if needed, and ensuring that parallel tasks did not clash in terms of technical or human resources.

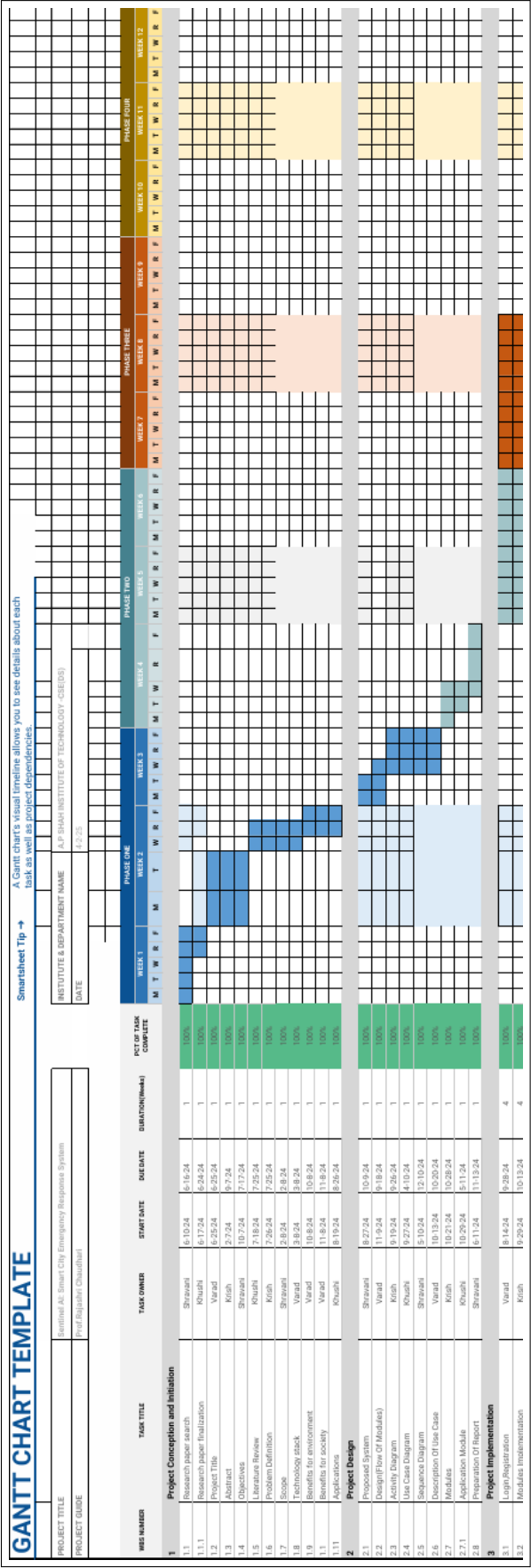


Figure 4.8: Timeline of project: [Part 1]



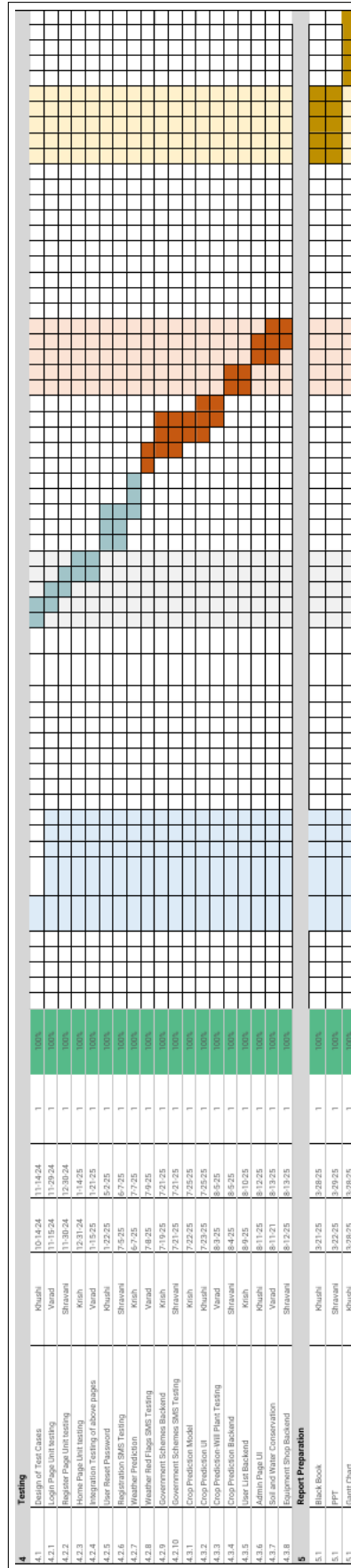


Figure 4.9: Timeline of project: [Part 2]

# Chapter 5

## Testing

Testing is an essential aspect of the software development lifecycle aimed at ensuring the functionality, reliability, and security of a system. It involves executing a system or component to identify any errors, gaps, or missing requirements. The primary objective of testing is to validate that the software behaves as intended under all expected and unexpected conditions. In modern application development, where user experience and data security are paramount, testing plays a critical role in delivering a robust and trustworthy system. It not only helps in detecting defects early but also in enhancing performance, maintaining code quality, and ensuring compliance with user expectations and industry standards. For this project, testing was a vital phase used to evaluate the accuracy and efficiency of the face recognition and ticket validation system, ensuring that the system can accurately match faces, process data quickly, and prevent unauthorized access during check-in.

### 5.1 Software Testing

Software testing is a crucial phase in the software development lifecycle that ensures the reliability, accuracy, and overall quality of the system. In this project, software testing was employed to assess the performance of the face verification and ticket validation module, with a focus on ensuring high precision and minimal error in identity recognition. The goal was to verify that the system not only distinguishes between genuine and fraudulent users effectively but also does so with minimal delay, ensuring a seamless and secure check-in process. By analyzing classification metrics and confusion matrix statistics, the testing process provided a comprehensive understanding of the model's strengths, such as perfect precision and high overall accuracy, as well as areas that can be improved, particularly in reducing false negatives. Through rigorous testing, the system's robustness and efficiency were validated, reinforcing its suitability for real-world deployment in high-security environments. Software testing is a crucial phase in the software development lifecycle that ensures the reliability, accuracy, and overall quality of the system. It involves the systematic evaluation of a software application to identify defects, validate functionality, and ensure that the system performs as expected under various conditions. In this project, software testing was employed to assess the performance of the face verification and ticket validation module, with a focus on ensuring high precision and minimal error in identity recognition. The goal was to verify that the system not only distinguishes between genuine and fraudulent users effectively but also does so with minimal delay, ensuring a seamless and secure check-in process. Particular attention was given to edge cases, such as users attempting to spoof the system using altered

images, low-light conditions, and minor facial occlusions (e.g., masks or glasses), to ensure robustness. Automated testing scripts were developed to simulate multiple check-in scenarios, verifying that the system consistently produces correct outputs and gracefully handles exceptions or invalid inputs.

```
Processing 6000 pairs...
100%|

=== Evaluation Results ===

Classification Metrics:
Accuracy: 0.9368
Precision: 1.0000
Recall: 0.8735
F1-Score: 0.9325

Confusion Matrix Statistics:
True Positives: 2611
True Negatives: 2994
False Positives: 0
False Negatives: 378
False Positive Rate: 0.0000
False Negative Rate: 0.1265

Processing Statistics:
Total Pairs Processed: 5983
Failed Pairs: 17
Average Processing Time: 0.4755 seconds
```

Figure 5.1: Software Testing

The software testing output represents the evaluation of a face verification system, most likely leveraging machine learning for binary classification. The system was tested on 6000 face pairs to assess its ability to distinguish between matching and non-matching identities. The evaluation reveals that the system achieved an overall accuracy of 93.68 percent, indicating that it correctly classified a large majority of the tested face pairs. The precision is a perfect 1.0000, suggesting that whenever the system predicted two faces as a match, it was always correct. This means that the model had zero false positives, a highly desirable outcome in security-sensitive applications like biometric verification.

In terms of recall, the value stands at 0.8735, which means the system was able to detect about 87.35 percent of all actual matches. While this is strong, it also implies that the system missed approximately 12.65 percent of matching pairs—shown by the False Negative Rate of 0.1265. Despite these misses, the F1-score of 0.9325 reflects a strong balance between precision and recall, confirming that the system maintains both accuracy and reliability in its predictions. The confusion matrix statistics provide further insight into the performance. Out of 6000 pairs, the model identified 2611 true positives and 2994 true negatives, confirming a solid number of correctly classified pairs. Importantly, the false positives were zero, affirming the system’s reliability when declaring a match. However, there were 378 false negatives, where the system failed to recognize a valid match, highlighting a potential area for improvement in recall. Additionally, the processing statistics show that 5983 face pairs were successfully evaluated, with only 17 failures due to possible data issues, such as unreadable images or corrupt input. The average processing time per pair was approximately 0.4755 seconds, demonstrating efficient execution suitable for real-time or near-real-time applications. Overall, the testing results validate the system’s effectiveness in high-precision environments while identifying areas for improving its recall to further minimize missed valid matches.

## 5.2 Functional Testing

Functional testing is a type of software testing that validates the software system against the functional requirements or specifications. The goal is to ensure that the application behaves as expected when subjected to various user scenarios. It primarily focuses on the output of the system and verifies that the software performs all the stated functions correctly. Unlike structural testing, functional testing does not concern itself with the internal workings of the application; instead, it evaluates the system based on the intended functionality. In facial recognition systems, this testing is crucial to confirm that face detection, comparison, embedding generation, and match validation work accurately under different conditions.

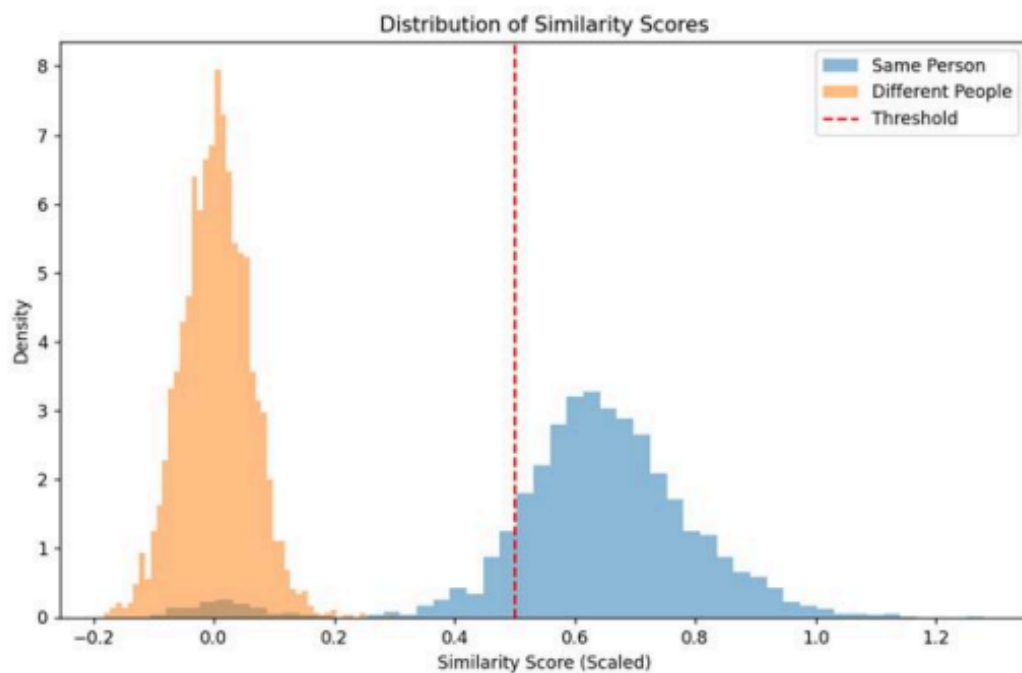


Figure 5.2: Functional Testing

The graph provided depicts the outcome of functional testing performed on a facial recognition system using similarity scores. It shows two major distributions—one representing face pairs of the same person and the other of different individuals. The blue distribution, which is right-shifted, indicates high similarity scores when the system compares images of the same individual, proving the system’s ability to recognize consistent facial features. The orange distribution, concentrated toward the left, demonstrates low similarity scores for different people, confirming that the system effectively differentiates between unique users. The red dashed line shown in the graph acts as a threshold score: if a new input score falls to the right of this threshold, the system concludes the two images are of the same person; otherwise, they are considered different.

The visual separation between the two score distributions in the graph showcases the robustness of the system under functional testing. A clearly defined threshold ensures that both false acceptances (wrongly approving a non-matching face) and false rejections (denying a correct match) are minimized. This validation supports real-world scenarios such as the ticketing system described, where a user registers with a facial image and receives a QR-based

ticket. Upon arrival, their face is scanned and matched against saved embeddings using the same similarity metrics tested here. The functional test, therefore, directly confirms the system’s capability to perform reliable identity verification, ensuring secure and seamless user check-ins while maintaining high standards of accuracy and user trust.

Functional testing is a type of software testing that validates the software system against the functional requirements or specifications. The goal is to ensure that the application behaves as expected when subjected to various user scenarios. It primarily focuses on the output of the system and verifies that the software performs all the stated functions correctly. Unlike structural testing, functional testing does not concern itself with the internal workings of the application; instead, it evaluates the system based on the intended functionality. In facial recognition systems, this form of testing is crucial to confirm that core modules such as face detection, feature extraction (embedding generation), face comparison, and match validation operate accurately under a wide range of real-world conditions. These conditions include variations in lighting, facial orientation, occlusion (e.g., masks, glasses), camera resolution, and environmental noise. Functional testing helps verify whether the system can successfully capture and process facial data, map it to a unique identity, and make access control decisions based on this recognition.

Additionally, it involves testing the integration of third-party components like the Insight-Face model, UIDAI APIs, and infrared sensors, ensuring that the system performs end-to-end identity verification smoothly. For instance, during the ticket verification process, the test would validate whether the user’s live face scan matches the stored embedding accurately, and if the ticket check-in status updates correctly upon successful validation. Functional testing also encompasses user-facing scenarios such as ticket booking with biometric linkage, QR code verification, and check-in workflows. It checks for appropriate system responses to valid and invalid inputs, including edge cases like partial face visibility or invalid government ID details. This ensures robustness in the face of user errors or technical limitations.

In the context of Sentinel AI, functional testing contributes directly to the system’s usability, security, and reliability. It provides assurance to both event organizers and end users that the system meets the defined functional goals and performs its critical operations—such as preventing unauthorized access or ticket fraud—with precision.

Test ID	Module	Test Description	Test Steps	Expected Result	Status
TC01	Face Capture	Verify facial image upload during ticket booking	Open app → Book ticket → Upload face image	Face embedding generated and saved	Pass
TC02	UIDAI Verification	Check age validation with Aadhar API	Submit Aadhar → Validate age → Confirm	Age validated or blocked if under-age	Pass
TC03	Payment Gateway	Validate Razorpay transaction integration	Fill booking form → Proceed to payment → Pay	Payment processed, confirmation sent	Pass
TC04	Booking Confirmation	Confirm post-booking details and email sent	Complete payment → Get ticket → Check email	Digital ticket received via email	Pass
TC05	Check-In Matching	Face scan at venue matches booking record	User arrives → Live scan → Compare embedding	Face match success and check-in marked	Pass
TC06	Face Mismatch Handling	Prevent access on mismatch	Attempt check-in with unmatched face	Access denied, alert triggered	Pass
TC07	Duplicate Entry Prevention	Block re-entry with same ticket	Try to check-in again post-entry	Reuse prevented, booking marked as closed	Pass
TC08	Low-Light Recognition	Ensure face recognition under poor lighting	Simulate low-light → Attempt entry	Accurate detection via infrared sensor	Pass
TC09	QR Fallback	Check QR-based backup entry system	Fail face match → Show QR → Verify manually	Manual override allows backup entry	Pass
TC10	Data Security	Verify encryption of sensitive data	Book ticket → Inspect database → Test access	Data securely encrypted	Pass

Table 5.1: Functional Testing of Sentinel AI Ticketing and Check-In System

# Chapter 6

## Result and Discussions

Results the proposed system aims to address critical challenges in the Indian event ticketing industry, such as ticket scalping, bot-driven purchases, and underage access to age-restricted events. The key features of the system, including facial recognition, Aadhar integration for age verification, and infrared sensor-based entry, were rigorously tested to ensure their effectiveness.

**Facial Recognition Accuracy:**The InsightFace model demonstrated a 93.6 percent accuracy with a threshold value of 0.5, ensuring minimal false positives. This high accuracy indicates that the system can reliably identify and authenticate attendees at the point of entry.

**Bot Prevention:**The facial recognition system integrated with the ticketing platform effectively blocked bot-driven ticket purchases. Users had to undergo biometric verification during the ticket booking process, ensuring that only legitimate customers could purchase tickets.

**Aadhar Integration:**UIDAI Aadhar API integration proved to be a reliable method for verifying age during ticket booking. This functionality prevented underage individuals from entering age-restricted events, ensuring legal compliance and safety.

**Entry Verification:**The infrared sensors paired with high-quality webcams allowed the system to accurately verify identities even in low-light conditions at event venues. This made entry verification swift and secure, enhancing the attendee experience.

**Discussion:** The system performed well across various tests, offering an effective solution to the primary issues facing event organizers and platforms. One of the most significant outcomes is the prevention of ticket scalping, as the system's use of facial recognition during ticket booking blocks bots, ensuring tickets are only available to real users. The age verification system using Aadhar data also proved highly effective in ensuring that only individuals of legal age gain access to age-restricted events. Moreover, the biometric authentication system facilitated secure and quick entry, which helped improve the overall attendee experience. However, there were some challenges during high-traffic simulations, where performance testing showed a slight delay in facial recognition processing during peak demand. While the system can handle thousands of attendees, improving its scalability for larger events will be a focus for future development.

# Chapter 7

## Conclusion

In conclusion, the proposed event ticketing and entry management system provides an innovative and comprehensive solution to the critical issues faced by the Indian event industry, such as ticket scalping, bot-driven purchases, and underage access to restricted events. By integrating cutting-edge technologies such as facial recognition, Aadhar-based age verification, and infrared sensors, the system enhances security, efficiency, and user experience in the ticketing process. The use of InsightFace for facial recognition ensures high accuracy in user identification, offering a robust mechanism to prevent fraud, reduce scalping, and eliminate the need for manual identity verification. The integration of Aadhar verification directly addresses the issue of underage individuals gaining access to events meant for older audiences, ensuring compliance with regulations while protecting event organizers and attendees alike.

Additionally, the infrared sensors and QHD webcams facilitate reliable and fast entry verification even in low-light conditions, allowing for a smooth flow of attendees at large-scale events. The central database stores all user data securely, ensuring that sensitive information such as biometric data and Aadhar details are handled with utmost confidentiality and in compliance with data privacy laws. The system's seamless integration with existing ticketing platforms, such as Bookmyshow, ensures minimal disruption for users and event organizers, enabling an efficient and secure ticket booking and entry process. The system's ability to scale and handle high-demand events, coupled with its focus on data security, makes it a reliable solution for both small and large-scale events. It not only protects event organizers from common issues like fraudulent ticket sales and bot interference, but it also provides a user-friendly experience that allows attendees to purchase tickets and enter venues without delays. Furthermore, the use of real-time data validation through Aadhar and facial recognition guarantees the authenticity of ticket holders, ensuring that only legitimate attendees are granted access.

While the system has already shown promising results, there is room for improvement in terms of scalability, especially when handling extreme traffic during major events. Future development could focus on optimizing the facial recognition algorithms for faster processing, improving system performance during peak hours, and integrating additional layers of multi-factor authentication for even higher levels of security. As the system continues to evolve, it could potentially be adapted for use in different regions by integrating other national identity verification systems, further enhancing its global appeal. In summary, this system offers a forward-thinking approach to event security and access management, addressing modern challenges head-on while providing a scalable, secure, and user-friendly solution.



# Chapter 8

## Future Scope

**Future Scope** The project has the potential to evolve further and address emerging challenges and opportunities in the event ticketing industry. The following areas outline the future scope of the system:

- Improved Scalability:** As the system is deployed at larger events, the facial recognition and Aadhar verification processes will need to be optimized for faster processing and higher volume traffic to handle larger crowds efficiently.
- Advanced Security Features:** Future versions of the system could integrate multi-factor authentication (MFA) or voice recognition in addition to facial recognition to further enhance security and reduce the risk of impersonation.
- Aadhar Integration Enhancement:** UIDAI Aadhar integration could be further enhanced by introducing real-time verification with advanced fraud detection mechanisms, ensuring that fake Aadhar cards or altered information cannot bypass the system. Aadhar data privacy and user consent for data usage will need to be handled in accordance with the latest data protection regulations to ensure compliance with privacy laws.
- Integration with Other Platforms:** The system could be integrated with other ticketing platforms or venue management systems to offer a more holistic solution for ticketing, entry, and security across a range of events, including sports, concerts, and festivals.
- Mobile Integration:** Incorporating mobile-based verification (e.g., using smartphone facial recognition or QR codes) could allow users to verify their identity more easily and securely without needing specialized hardware at the venue.
- Global Adoption:** The system could be adapted for international use by incorporating other national identity verification systems alongside Aadhar, making it a viable solution for events outside India. By expanding the capabilities of this system, it could not only secure events but also offer a comprehensive, user-friendly experience for ticket purchasers, organizers, and security teams, setting new standards for the future of event security and access management.

# Bibliography

- [1] Rebecca Black. ” no one likes us, we don’t care”: The legality of ticket bans on opposing fans. *Jeffrey S. Moorad Sports LJ*, 31:325, 2024.
- [2] Daniel G Costa, Francisco Vasques, Paulo Portugal, and Ana Aguiar. A distributed multi-tier emergency alerting system exploiting sensors-based event detection to support smart city applications. *Sensors*, 20(1):170, 2019.
- [3] Fabrizio Dell’Acqua, Charles Ayoubi, Hila Lifshitz-Assaf, Raffaella Sadun, Ethan R Mollick, Lilach Mollick, Yi Han, Jeff Goldman, Hari Nair, Stew Taub, et al. The cybernetic teammate: A field experiment on generative ai reshaping teamwork and expertise. *Harvard Business School Strategy Unit Working Paper*, (25-043):25–043, 2025.
- [4] Jiankang Deng, Jia Guo, Xiang An, Zheng Zhu, and Stefanos Zafeiriou. Masked face recognition challenge: The insightface track report. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1437–1444, 2021.
- [5] Minchul Kim, Anil K Jain, and Xiaoming Liu. Face recognition system using quality adaptive margins and method of performing the same, September 28 2023. US Patent App. 18/125,364.
- [6] Sangmin Park, Soung Hoan Park, Lee Won Park, Sanguk Park, Sanghoon Lee, Tacklim Lee, Sang Hyeon Lee, Hyeonwoo Jang, Seung Min Kim, Hangbae Chang, et al. Design and implementation of a smart iot based building and town disaster management system in smart city infrastructure. *Applied Sciences*, 8(11):2239, 2018.
- [7] Inês Pires Santos and Lucas Jones. How they see us. 2021.
- [8] David Sénécal. *The Reign of Botnets: Defending Against Abuses, Bots and Fraud on the Internet*. John Wiley & Sons, 2024.
- [9] Andrew Jason Shepley. Deep learning for face recognition: a critical analysis. *arXiv preprint arXiv:1907.12739*, 2019.
- [10] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5163–5172, 2017.
- [11] Zhiyuan Yang, Haoyu Xie, Yue Xu, Qiaochu Xu, Yu Zhang, Sinuo Zhao, Hao Yuan, and Yajun Fang. Evaluation of smart response systems for city emergencies and novel uv-oriented solution for integration, resilience, inclusiveness and sustainability. In *2020 5th International Conference on Universal Village (UV)*, pages 1–51. IEEE, 2020.

- [12] Hong Zhang, Zeyu Zhang, Lei Zhang, Yifan Yang, Qiaochu Kang, and Daniel Sun. Object tracking for a smart city using iot and edge computing. *Sensors*, 19(9):1987, 2019.

# Appendices

Detailed information, lengthy derivations, raw experimental observations etc. are to be presented in the separate appendices, which shall be numbered in Roman Capitals (e.g. “Appendix I”). Since reference can be drawn to published/unpublished literature in the appendices these should precede the “Literature Cited” section.

## Appendix-A: InsightFace Download and Installation

1. Clone the InsightFace repository from GitHub:

```
$ git clone https://github.com/insightface/insightface.git
$ cd insightface
```

2. (Optional) Create and activate a virtual environment for project isolation:

```
$ python3 -m venv insightface_env
$ source insightface_env/bin/activate (On Windows:
insightface_env\Scripts\activate)
```

3. Install system dependencies (for Ubuntu/Debian):

```
$ sudo apt update
$ sudo apt install -y build-essential libopenblas-dev libopencv-dev liblapack-dev
```

4. Install Python dependencies listed in `requirements.txt`:

```
$ pip install -r requirements.txt
```

5. Install MXNet (CPU or GPU version depending on your setup):

For CPU-only version:

```
$ pip install mxnet
```

For CUDA-enabled GPU (example for CUDA 10.2):

```
$ pip install mxnet-cu102
```

6. Test the setup by running a demo script from the InsightFace examples:

```
$ python demo.py
```

This should execute a face detection/recognition task using sample inputs.

7. To contribute, fork the repository and work on a new branch:

```
$ git checkout -b feature_branch_name
```

Make your changes, test them locally, then commit and push to your forked repo.

8. After making changes, run the test suite to verify functionality:

```
$ pytest
```

9. Submit a Pull Request (PR) to the official InsightFace repository for review and merge.

10. To keep your local repo up to date with the original repository:

```
$ git remote add upstream https://github.com/insightface/insightface.git
```

```
$ git fetch upstream
```

```
$ git merge upstream/master
```

Now, InsightFace is fully set up for development and contribution