An Industrial Oriented Mini Project Report on

AI BASED ATTENDANCE SYSTEM

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

LOKESH DAHAKE	**20BD1A05AE**
A KIRAN KUMAR	**20BD1A05A1**
R. NARSIMHA RAO	**20BD1A05BD**
UDITHYALA UDAY	**20BD1A05BR**

Under the guidance of

Ms. K Yasaswini
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.
Narayanaguda, Hyderabad, Telangana-29

2023-24

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.
Narayanaguda, Hyderabad, Telangana-29

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this is a bonafide record of the project report titled **"AI Based Attendance System"** which is being presented as the Industrial Oriented Mini Project report by

1. DAHAKE LOKESH	**20BD1A05AE**
2. A. KIRAN KUMAR	**20BD1A05A1**
3. R. NARSIMHA RAO	**20BD1A05BD**
4. UDITHYALA UDAY	**20BD1A05BR**

In partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering affiliated to the Jawaharlal Nehru Technological University Hyderabad, Hyderabad

Faculty Supervisor
(Ms. K Yasaswini)

Head of Department
(Mr. P. Upendar)

Submitted for Viva Voce Examination held on _____

External Examiner

Vision & Mission of KMIT

Vision of KMIT

- To be the fountainhead in producing highly skilled, globally competent engineers.
- Producing quality graduates trained in the latest software technologies and related tools and striving to make India a world leader in software products and services.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepares students to become successful professionals.
- To establish an industry institute Interaction to make students ready for the industry.
- To provide exposure to students on the latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities

PROGRAM OUTCOMES (POs)

PO1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution ofcomplex engineering problems.

PO2. Problem Analysis: Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

PO3. Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct Investigations of Complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The Engineer and Society: Apply reasoning informed by contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep learning, IOT, Data Science, Full stack development, Social Networks, Cyber Security, Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting to the changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES

P1: Accurately detect motion and control the appliances accordingly.

P2: Allow control of appliances remotely.

P3: Change the state of appliances instantly.

P4: Work seamlessly over the internet.

MAPPING PROJECT OUTCOMES WITH PROGRAM OUTCOMES

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	2	3	2	3	1	1	3	2	2	3	3	2
P2	3	2	3	1	3	2	1	3	1	2	2	3
P3	2	3	1	2	2	3	1	1	3	1	3	1
P4	3	3	2	2	1	1	2	2	1	3	1	2

L – 1 M –2 H– 3

PROJECT OUTCOMES MAPPING WITH PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	2	3
P2	2	1
P3	3	2
P4	3	3

PROJECT OUTCOMES MAPPING WITH PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1		2	3	2
P2	3	2	1	3
P3	2	3	2	1
P4	3	2	1	2

DECLARATION

We hereby declare that the results embodied in the dissertation entitled "AI Based Attendance System" has been carried out by us together during the academic year 2023-24 as a partial fulfillment of the award of the B.Tech degree in Computer Science and Engineering from JNTUH. We have not submitted this report to any other university or organization for the award of any other degree.

Student Name	Rollno.
DAHAKE LOKESH	20BD1A05AE
A. KIRAN KUMAR	20BD1A05A1
R. NARSIMHA RAO	20BD1A05BD
UDITHYALA UDAY	20BD1A05BR

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. B L Malleswari**, Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Founder & Director, **Mr. S. Nitin,** Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Ms. Deepa Ganu**, Director Academic for providing an excellent environment in the college.

We are also thankful to **Mr. P Upendar**, Head of the Department for providing us with time to make this project a success within the given schedule.

We are also thankful to our Faculty Supervisor **Ms. K Yasaswini**, for her/his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

Student Name	Roll no.
DAHAKE LOKESH	20BD1A05AE
A. KIRAN KUMAR	20BD1A05A1
R. NARSIMHA	20BD1A05BD
UDITHYALA UDAY	20BD1A05BR

ABSTRACT

The main purpose of this project is to build a face recognition-based attendance monitoring system for any educational or organizational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals is available to the respective faculty/authority and updates can be made by the authorized person only

LIST OF FIGURES

CONTENTS

CHAPTER-1

1. INTRODUCTION

1.1 Purpose of the project

The purpose of an AI-based attendance system project is to revolutionize the traditional process of recording attendance by leveraging artificial intelligence technologies. By employing computer vision and machine learning, the project aims to enhance efficiency, accuracy, and convenience in tracking attendance for educational institutions, offices, and other organizations.

The system's automated features reduce the administrative burden, providing real-time monitoring, improving security through biometric authentication, and generating valuable data insights. Beyond streamlining routine tasks, the project showcases a commitment to technological innovation, adaptability across various environments, and, especially in the context of health crises, contributes to contactless processes.

However, it is imperative to prioritize privacy, comply with regulations, and ensure transparency in data handling to address ethical considerations and build trust among users.

1.2 Problem Statement

The traditional attendance system where the attendance is marked manually so, the project aim is to automate it. It also enables an organization to maintain its records like in-time, out time, break time and attendance digitally. Digitization of the system would also help in better visualization of the data using graphs to display the no. of employees present today, total work hours of each employee and their break time. Its added features serve as an efficient upgrade and replacement over the traditional attendance system.

1.3 Objectives

- To develop a Smart Attendance System.
- To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.
- Have enough memory space to store the database.
- Able to recognize the face of an individual accurately based on the face database.
- Develop a database for the attendance management system.

- Provide a user-friendly interface for admins to access the attendance database and for non-admins to check their own attendance by mailing the attendance.
- Allow new students or staff to store their faces in the database by using a GUI.
- Allow the admin to edit the attendance just in any exceptional case of emergencies or permissions.

- Able to show an indication to the user whether the face-recognition process is successful or not.

1.4 Scope and Limitations

- To make the attendance system automatic.
- To save the time.
- To reduce the cost so that everyone can afford it.
- The limitation in our model is it cannot detect the all persons in the group and cannot take attendance.

CHAPTER-2

2. LITERATURE SURVEY

The literature survey for an AI-based attendance system project involves a comprehensive exploration of existing academic and professional literature pertaining to facial recognition, biometrics, computer vision, and machine learning. Studies on facial recognition algorithms, biometric technologies, and their ethical considerations provide insights into the strengths and limitations of identification methods. Research on computer vision techniques and object detection contributes to understanding the development of accurate and efficient facial recognition models. Examining machine learning models used in attendance systems helps in identifying suitable algorithms and training methodologies. Privacy and ethical considerations, user acceptance, security measures, real-time processing, and system performance are crucial aspects to explore. Comparative studies and research specific to the educational or organizational context provide valuable insights, while a focus on implementation challenges and regulatory compliance ensures a thorough understanding of potential issues and legal considerations. A literature survey is essential for establishing a solid foundation, addressing gaps, and informing the development of an effective and ethically sound AI-based attendance system.

Additionally, the literature survey should delve into user experience and acceptance, examining how individuals perceive and interact with AI-based attendance systems. Studies in this area can uncover user preferences, potential challenges to adoption, and strategies for improving the overall user experience. Exploring research on security measures is crucial to understanding methods for safeguarding against unauthorized access, data breaches, and potential vulnerabilities, contributing to the development of a robust and secure system. Moreover, an investigation into real-time processing and system performance metrics is essential to ensure that the AI-based attendance system can handle large datasets efficiently and provide timely responses. Comparative studies between different biometric methods or AI models offer valuable insights into the advantages and disadvantages of various approaches, aiding in the selection of the most suitable technology for the project. Ultimately, a thorough literature survey not only builds a strong theoretical foundation but also informs practical considerations and potential challenges that may arise during the implementation of the AI-based attendance system.

CHAPTER-3

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction to SRS:

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)
The SRS phase consists of two basic activities:

1. Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

2. Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.
The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

3.2 Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all
the parties involved in the system.

3.3 Requirements Specification Document:

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non-functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

3.4. Functional Requirements Specification:

Functional requirements in an SRS outline the specific functionalities and capabilities that a software system must possess to meet the specified needs of its users. These requirements define how the system should behave and what actions it should be able to perform. They are typically detailed, specific, and measurable, providing a clear understanding of the expected system behavior. Functional requirements serve as the foundation for system design, development, and testing, guiding the implementation process to ensure that the final software product meets user expectations. By delineating the essential features, interactions, and outcomes, functional requirements provide a comprehensive roadmap for the development team, enabling them to create a system that aligns with the user's operational needs and business objectives. The functional requirements for AI based attendance system are:

1 User Authentication
- The system shall support user authentication through unique usernames and passwords.

2 Face Recognition
- The system shall employ AI-based face recognition for identifying and verifying individuals.

3 Attendance Tracking
- The system shall record attendance based on face recognition results.

4 Reporting
- The system shall generate attendance reports for administrators.

5 User Management
- The system shall allow administrators to add, modify, and delete user profiles.

By fulfilling these requirements, the project aims to offer an effective and streamlined solution for automated helmet and number plate detection, contributing to enhanced road safety.

3.5. Non-Functional Requirements:

In a Software Requirements Specification (SRS), non-functional requirements describe the attributes and qualities that define the overall behavior and performance of a software system, beyond its specific functionalities. These requirements encompass aspects such as performance, reliability, usability, security, scalability, and maintainability. Unlike functional requirements, which outline what the system must do, non-functional requirements focus on how well the system must perform or adhere to certain constraints. They provide critical guidelines for evaluating the overall success of the software, ensuring that it meets user expectations, industry standards, and regulatory compliance. Non-functional requirements play a vital role in shaping the user experience, determining system reliability, and influencing architectural and design decisions throughout the software development lifecycle. The non-functional requirements for AI based attendance system are

1 Performance
- The system shall be able to handle concurrent attendance tracking for a minimum of 100 users.
- Response time for face recognition should be less than 2 seconds.

2 Reliability
- The system should have a 99.9% uptime.

3 Scalability
- The system should scale to accommodate additional users without significant performance degradation.

4 Usability
- The user interface shall be intuitive and user-friendly.

5 Security
- User data and attendance records should be securely stored and transmitted.

6 Availability
- The system should be available 24/7, with scheduled maintenance communicated in advance.

7 Compatibility
- The system should be compatible with common web browsers (Chrome, Firefox)

8 Maintainability
- The system should allow for easy updates and modifications.

9 Portability
- The system should be platform-independent and accessible from various devices.

10 Accuracy
- Face recognition accuracy should be at least 95%.

11 Response Time
- The system's response time for user interactions should be within acceptable limits.

12 Resource Utilization
- The system should utilize hardware resources efficiently.

13 Compliance
- The system should comply with relevant data protection and privacy regulations.

By addressing these non-functional requirements, this project aims to deliver a reliable, scalable, and user-friendly solution with high accuracy and performance.

3.6. Performance Requirements:

Performance requirements in a Software Requirements Specification (SRS) define the criteria that a system must meet regarding its speed, responsiveness, scalability, and resource utilization. These requirements detail the expected performance characteristics of the software under various conditions. They are crucial for ensuring that the system can handle the anticipated workload, deliver timely responses, and efficiently use resources. Performance requirements guide the design and implementation phases, influencing architectural decisions and optimizations to meet the specified criteria. By providing measurable benchmarks, performance requirements enable stakeholders to assess the software's responsiveness, throughput, and scalability, contributing to the overall success of the system. The performance requirements for AI based attendance system are:

1. Real-time Processing:
- The system should be able to process attendance for 100 users simultaneously.
- The face recognition algorithm should achieve a minimum accuracy of 95%.

2. Concurrent User Handling:
- The application should efficiently handle multiple concurrent users uploading and processing video files simultaneously.

3. Scalability:
- The system should scale effectively to accommodate an increasing number of users and handle growing volumes of video data.

4. Response Time:
- The application should provide timely responses to user interactions, and the response time for object detection and classification should be within acceptable limits.

5. Resource Utilization:
- Optimize resource usage to ensure efficient utilization of computing resources during video processing, minimizing system resource bottlenecks

3.7. Software Requirements:

1. Python: Python is a high-level, general-purpose programming language. It's the core language used to execute the code and manage dependencies.

2. OpenCV (cv2): OpenCV, or Open-Source Computer Vision Library, is a library for computer vision and image processing tasks. It's used for video processing and object detection in the code.

3. Tkinter: Tkinter is a Python library for creating graphical user interfaces (GUIs), commonly used in AI-based attendance systems for designing interactive and user-friendly interfaces to manage and display attendance-related information.

4. PIL (Python Imaging Library): PIL, or the Python Imaging Library, is a library for opening, manipulating, and saving image files. It's used for working with images in the code.

5. Pandas: Pandas is a Python library for data manipulation and analysis, providing data structures like Data Frame for efficient handling and processing of structured data. Pandas can be utilized for tasks such as organizing and analyzing attendance data, facilitating data preprocessing, and creating informative reports for administrators.

6. NumPy: NumPy is a fundamental Python library for numerical operations and array handling. It's often used for efficient data manipulation in scientific and engineering applications.

7. TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It's used for training and deploying machine learning models, and in this code, it's used to load a pre-trained facial recognition model.

8.face_recognition library: face recognition is a Python library for facial recognition that simplifies the process of detecting and identifying faces in images, commonly used in AI applications for attendance systems to recognize and verify individuals based on their facial features.

3.8. Hardware Requirements:

1. Computer: Any computer or server capable of running Python scripts is sufficient.

2. CPU: The Central Processing Unit (CPU) is used to execute the code. The code should run on a standard CPU, but the performance may vary based on the CPU's processing power.

3. GPU (Optional): A Graphics Processing Unit (GPU), particularly an NVIDIA GPU, can be used for GPU acceleration in deep learning tasks. It can significantly speed up the execution of machine learning models.

4. GPU Drivers (Optional): If a GPU is used, you need to install the appropriate GPU drivers to ensure compatibility and optimal performance.

5. Operating System: The code should work on various operating systems, including Windows, Linux, and macOS, as long as the required libraries and dependencies are properly installed.

CHAPTER-4

4. SYSTEM DESIGN

4.1 Introduction to UML:

Unified Modeling Language (UML) is a standardized visual modeling language used in software engineering for specifying, visualizing, constructing, and documenting the artifacts of a software system. It provides a common set of notations and semantics that allow developers, analysts, designers, and stakeholders to communicate and understand the structure and behavior of a system.

1. Key Features of UML:
   Standardization: UML is standardized by the Object Management Group (OMG), ensuring consistency and interoperability. It provides a common language for software development, making it easier for teams to collaborate.
2. Abstraction:
   UML allows developers to create models at different levels of abstraction. This includes high-level conceptual models as well as more detailed models focusing on specific aspects of the system.
3. Visual Representation:
   UML uses graphical notations to represent various elements of a system, making it easier to understand complex structures and behaviors. These visual representations are known as UML diagrams.

4.2 UML DIAGRAMS:

UML diagrams are graphical representations of different aspects of a software system. There are several types of UML diagrams, each serving a specific purpose in modelling various aspects of a system. Here are some commonly used UML diagrams:

1. Class Diagram:

Depicts the static structure of a system by showing classes, their attributes, methods, and relationships.

2. Use Case Diagram:

Illustrates the interactions between a system and its external actors, focusing on the system's functionality from a user's perspective.

3. Sequence Diagram:

Shows the interactions between objects over time, emphasizing the order of messages exchanged between them.

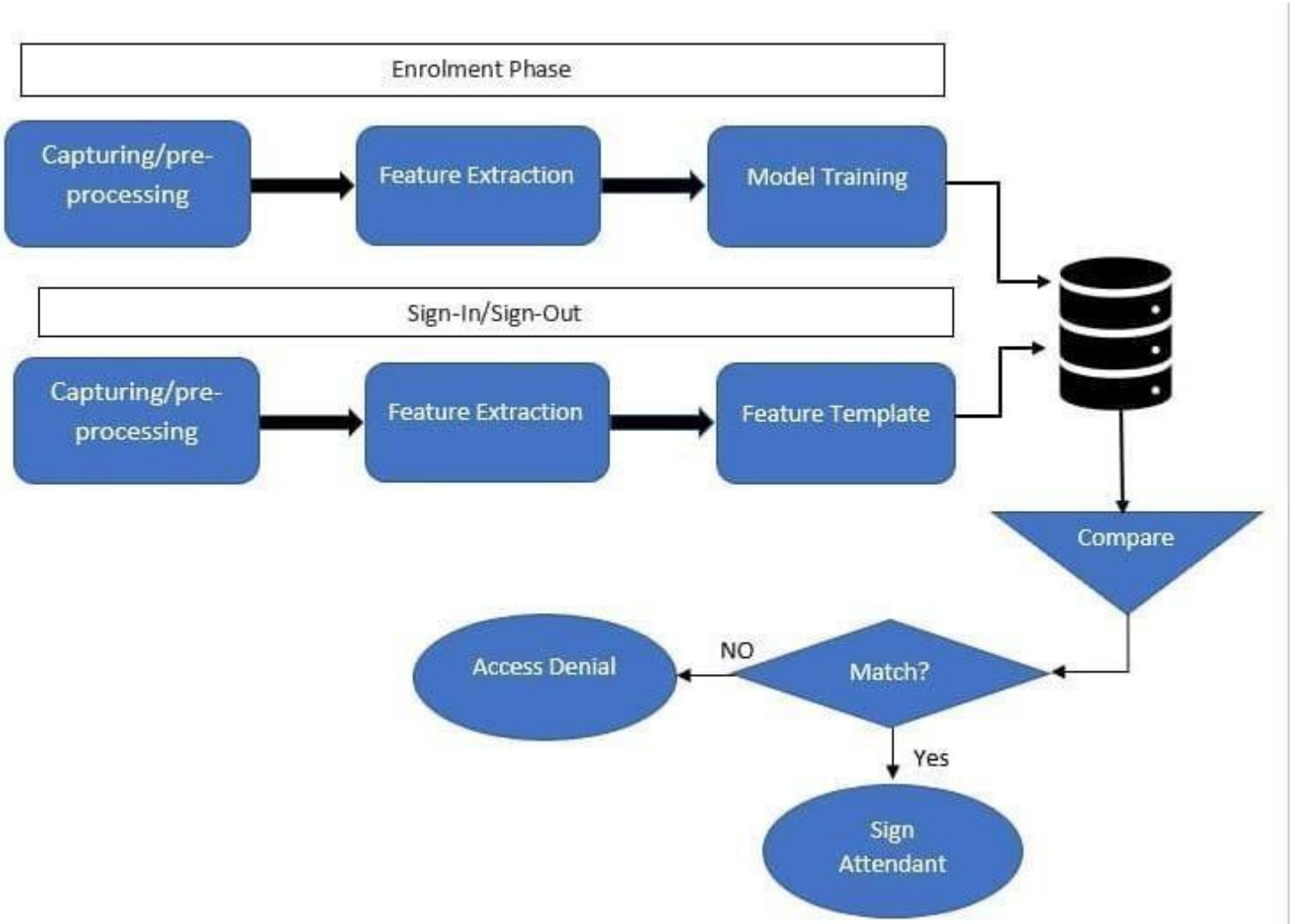
4. Activity Diagram:

Represents the flow of activities within a system, detailing the actions and decision points in a process.

4.2.1 ACTIVITY DIAGRAM

Activity diagrams focus on the workflow and business processes within the system. They describe the flow of activities, decision points, and parallel or concurrent behavior in a system. We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram

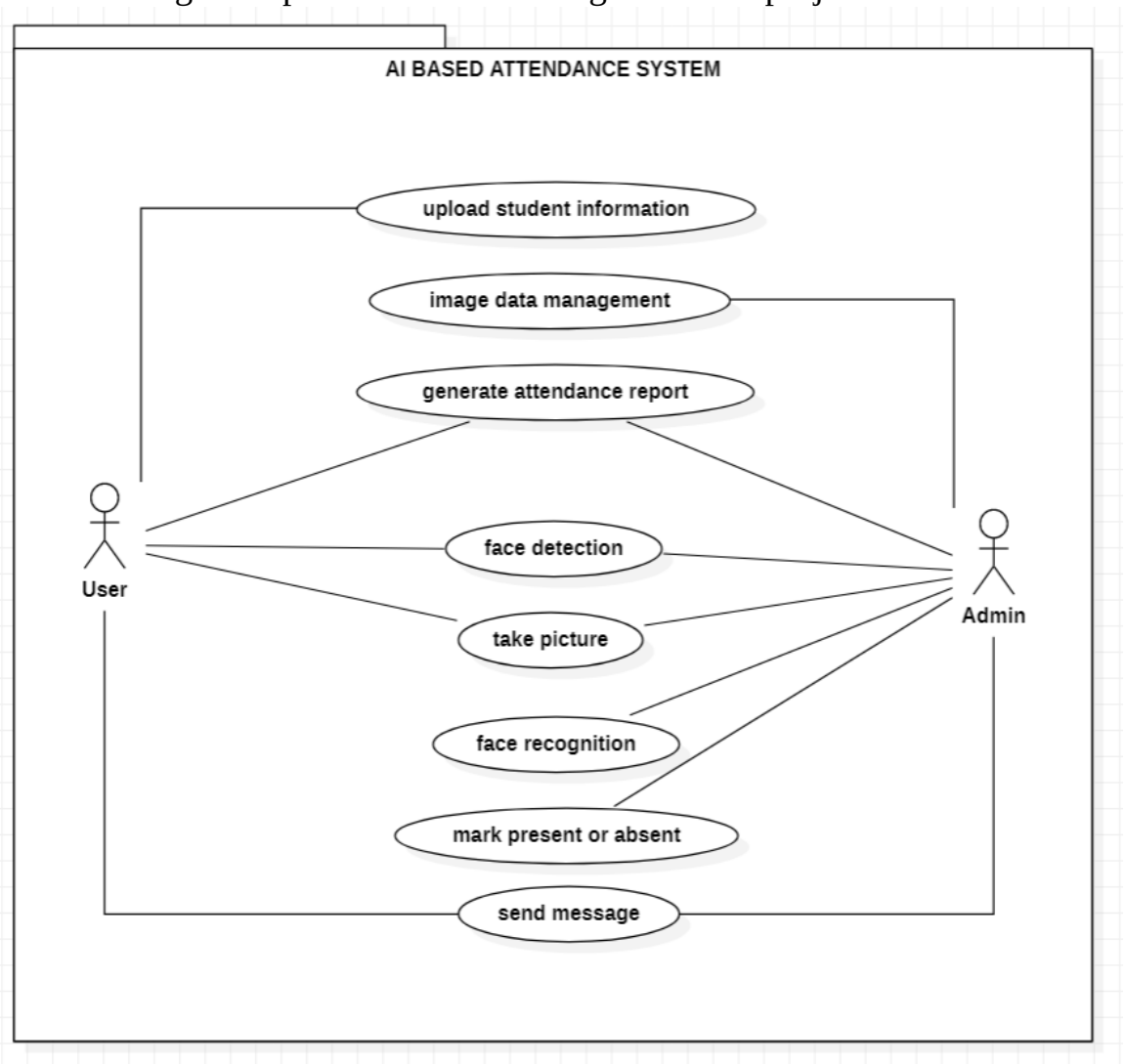
The Activity diagram of our mini project is shown in the below figure.



4.2.2 Use Case Diagram

A use case diagram in UML (Unified Modeling Language) models the behavior of a system. It describes the high-level functions and scope of a system. It also identifies the interactions between the system and its actors. A use case diagram consists of: Actors: The users of the system Use cases: The actions, services, and functions that the system needs to perform Relationships: The connections between the actors and use cases. A single use case diagram captures a particular functionality of a system. To model the entire system, a number of use case diagrams are used. To build a use case diagram, you'll use a set of specialized symbols and connectors. The notation for a use case is an ellipse. The notation for using a use case is a connecting line with an optional arrowhead showing the direction of control.

The below figure depicts the use case diagram of the project.

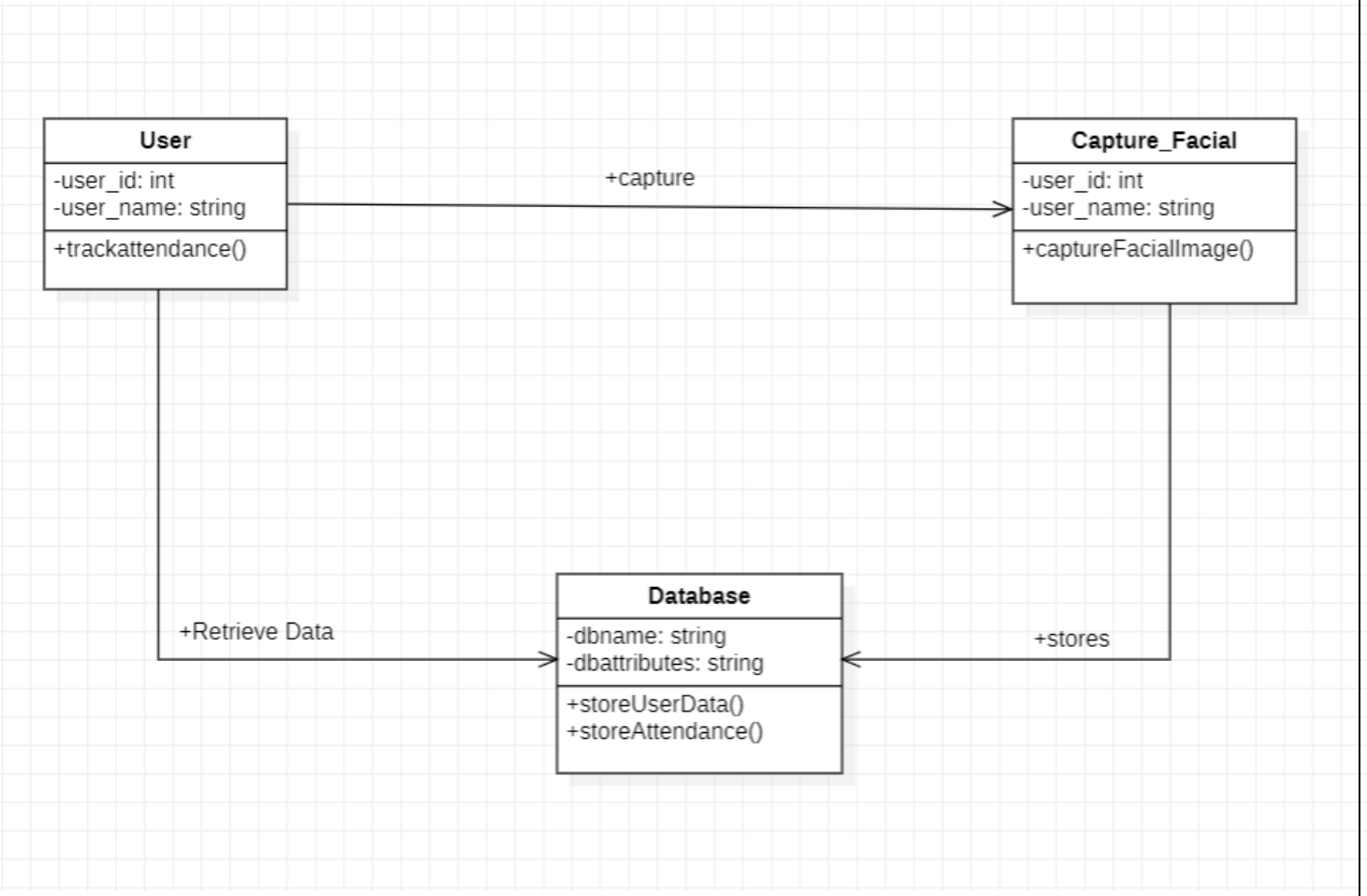


4.2.3 Class Diagram

A class diagram in the Unified Modelling Language (UML) is a static structure diagram that describes the
structure of a system. It shows the system's:

- Classes
- Attributes
- Operations (or methods)
- Relationships among objects

Class diagrams are widely used in the modelling of object-oriented systems. They are the only UML diagrams that can be mapped directly with object-oriented languages.
The below figures depict the class diagrams of the project



4.2.4 Sequence Diagram

A sequence diagram is a Unified Modelling Language (UML) diagram that shows the sequence of messages between objects in an interaction. Sequence diagrams are used to:

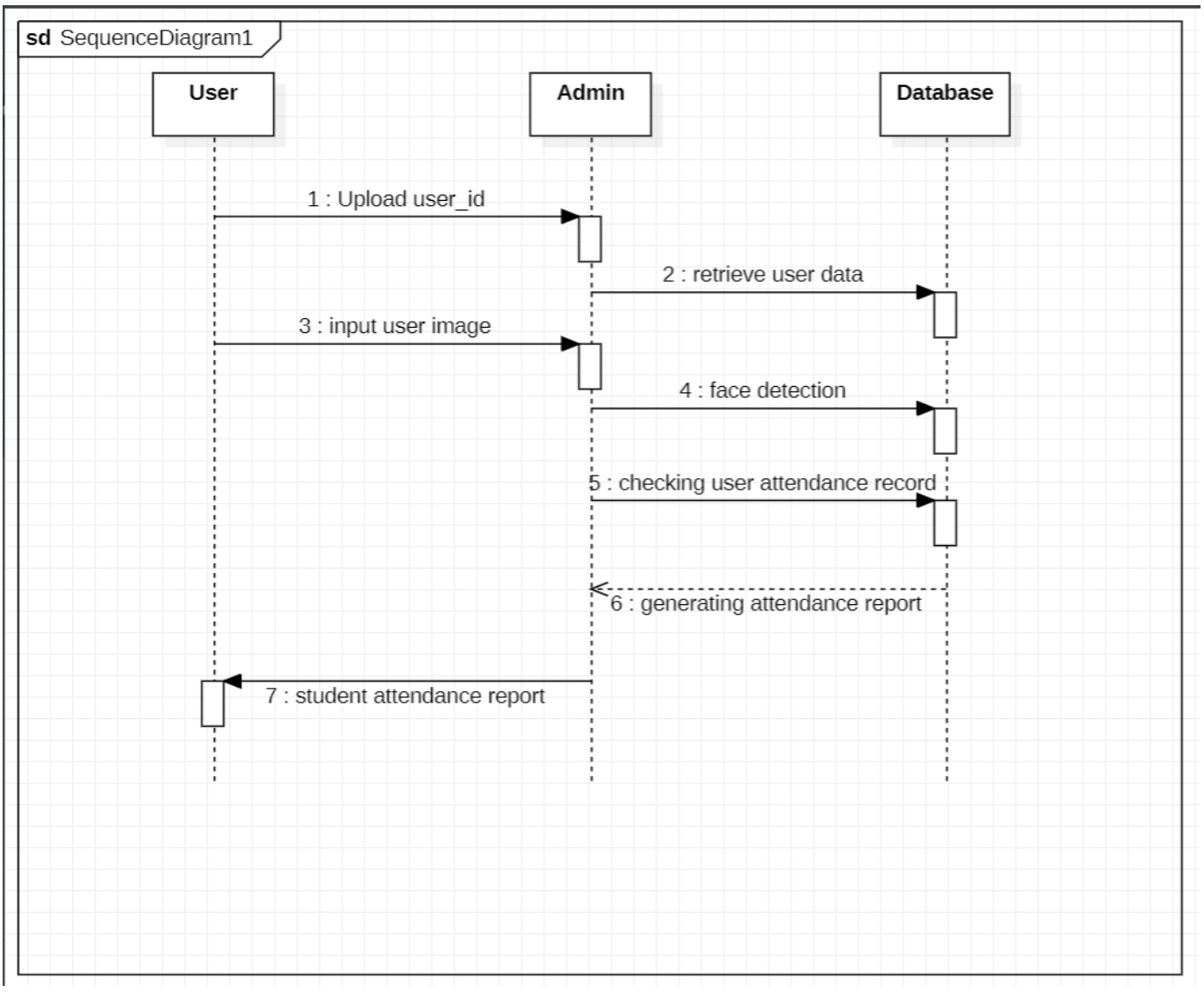
- Explore real-time applications
- Depict the message flow between objects
- Design, document, and validate the architecture, interface, and logic of systems
- Represent the scenario or flow of events in one single use case

Sequence diagrams consist of:

- A group of objects that are represented by lifelines
- The messages that the objects exchange over time during the interaction

The vertical axis of a sequence diagram represents time. The vertical position of an object indicates when it is instantiated. For example, if an object is “alive” from the beginning of the entire sequence, you should place it at the top of the diagram and its lifelines should extend to the bottom.

The Below diagrams depict the sequence diagram of the project



4.3 TECHNOLOGY USED

1. Python:

Python is a high-level, general-purpose programming language. It's the core language used to execute the code and manage dependencies.

2. OpenCV (cv2):

OpenCV, or Open-Source Computer Vision Library, is a library for computer vision and image processing tasks. It's used for video processing and object detection in the code.

3. Tkinter:

Tkinter is a Python library for creating graphical user interfaces (GUIs), commonly used in AI-based attendance systems for designing interactive and user-friendly interfaces to manage and display attendance-related information.

4. PIL (Python Imaging Library):

PIL, or the Python Imaging Library, is a library for opening,
manipulating, and saving image files. It's used for working with images in the code.

5. Pandas:

Pandas is a Python library for data manipulation and analysis, providing data structures like Data Frame for efficient handling and processing of structured data. Pandas can be utilized for tasks such as organizing and analysing attendance data, facilitating data preprocessing, and creating informative reports for administrators.

6. NumPy:

NumPy is a fundamental Python library for numerical operations and array handling. It's often used for efficient data manipulation in scientific and engineering applications.

7. TensorFlow:

TensorFlow is an open-source machine learning framework developed by Google. It's used for training and deploying machine learning models, and in this code, it's used to load a pre-trained facial recognition model.

8.face_recognition library:

face recognition is a Python library for facial recognition that simplifies the process of detecting and identifying faces in images, commonly used in AI applications for attendance systems to recognize and verify individuals based on their facial features.

CHAPTER-5

5. IMPLEMENTATION

5.1 Project Execution:

Implementing an AI-based attendance system involves several key steps:

1.Requirement Gathering: Define the specific requirements of the attendance system. Determine the desired features, such as facial recognition, biometrics, or RFID-based identification.

2.Technology Selection: Choose the appropriate AI technologies for the system. This may involve selecting facial recognition algorithms, biometric sensors, cameras, or RFID readers.

3.Data Collection and Preparation: Gather a dataset of faces or biometric information to train the AI model. Ensure the data is diverse and representative to improve accuracy.

4.Model Development: Develop or use pre-existing AI models for facial recognition or biometric identification. Train the model using the collected data to recognize faces or biometric patterns accurately.

5.System Integration: Integrate the AI model with the attendance system infrastructure. This involves connecting cameras, sensors, or other hardware devices to the AI model and attendance database.

6.Testing and Validation: Thoroughly test the system in different scenarios and environments to ensure accuracy, reliability, and security. Validate the system's performance against various use cases.

7.Deployment: Implement the system in the intended environment. Ensure proper installation of hardware, software, and network configurations.

8.Monitoring and Maintenance: Continuously monitor the system's performance, address any issues that arise, and update the AI model periodically to improve accuracy and security.

9.Compliance and Data Security: Ensure that the system complies with relevant privacy and data protection regulations. Implement measures to secure sensitive data collected by the attendance system.

5.2 Code Structure and Architecture

The code structure and architecture of an AI-based attendance system typically involve multiple components that work together to achieve accurate attendance tracking using AI technologies. Here's a high-level overview of the structure and architecture:

1.Data Collection:

Data Acquisition: Gather data from various sources like cameras, biometric sensors, or RFID readers.

Data Preprocessing: Clean and prepare the data for training. This includes image processing (if using facial recognition) or feature extraction (for biometric data).

2.AI Model Development:

• Algorithm Selection: Choose the appropriate AI algorithms for facial recognition or biometric identification (e.g., Convolutional Neural Networks (CNN) for facial recognition, machine learning models for biometric identification).

• Model Training: Train the AI model using labelled data. For facial recognition, use datasets of faces with corresponding identities; for biometric identification, use data with associated features.

3.Attendance Tracking:

• Face/Biometric Detection: Implement code to detect and recognize faces or biometric patterns from live or recorded data.

• Matching and Identification: Use the trained AI model to match detected faces/biometric patterns with stored identities in the database.

4.Database Management:

• Attendance Record Storage: Store attendance records securely in a database along with relevant metadata (e.g., date, time, identity).

• Database Integration: Ensure seamless integration between the AI model and the attendance database for storing and retrieving attendance information.

5.User Interface (UI) and Application Layer:

• User Interaction: Develop a user-friendly interface for administrators or users to interact with the system.

• System Control: Enable functionalities such as system configuration, reporting, and real-time monitoring.

6.Security and Compliance:

• Data Encryption: Implement encryption techniques to secure sensitive data stored in the database.

• Access Control: Set up permissions and access control mechanisms to manage who can interact with different parts of the system.

• Compliance: Ensure compliance with data protection regulations and privacy laws.

7.System Integration and Deployment:

• Hardware Integration: Connect cameras, sensors, or other devices to the system.

• Cloud or On-Premises Deployment: Deploy the system either in the cloud or on-premises based on infrastructure requirements.

5.3 Dataset Information:

1.Facial Recognition System:

• Images/Video Data: A collection of facial images or video frames containing faces.

• Labelled Data: Each image or video frame should be labelled with the identity of the person (name, ID, or unique identifier).

• Diversity: The dataset should encompass diverse facial characteristics (ethnicity, age, gender, expressions, lighting conditions, angles, and occlusions) to enhance the model's generalization.

• Data Size: Larger datasets often contribute to better model performance, so a substantial dataset with numerous identities is preferred.

• Quality and Resolution: High-quality images with adequate resolution to capture facial details are crucial.

2.Biometric System (e.g., Fingerprint, Iris, etc.):

• Biometric Data: Specific biometric information (fingerprint scans, iris images, etc.).

• Labelled Data: Each biometric sample should be linked to the respective person's identity.

• Variability: Similar to facial recognition, the dataset should cover variations within biometric traits (for instance, multiple fingerprints from different angles and pressures for fingerprint recognition).

• Data Integrity: Ensure data quality and accuracy to avoid false matches or identification errors.

5.4 Code Screenshot:

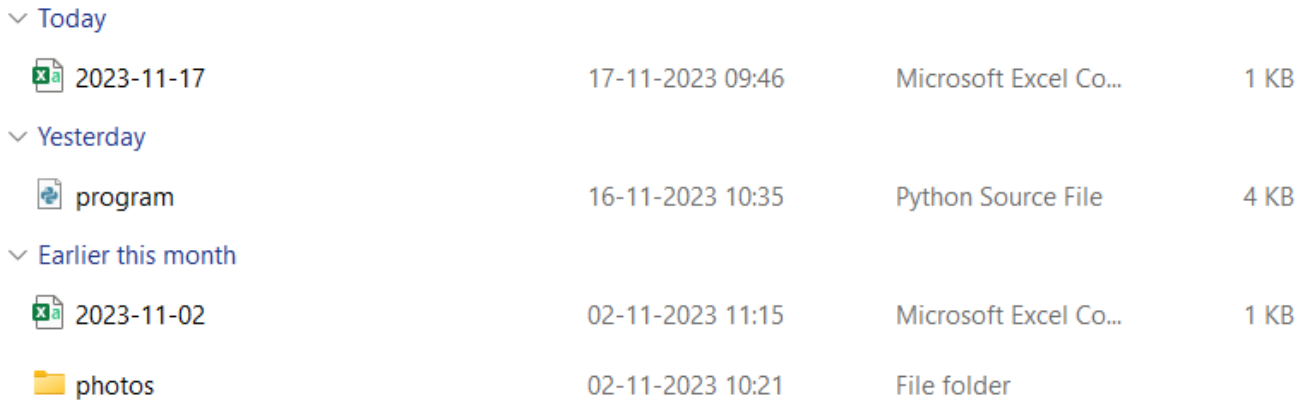
```
import face_recognition
import cv2
import numpy as np
import csv
import os
from datetime import datetime
import openpyxl

video_capture = cv2.VideoCapture(0)

narsimha_image = face_recognition.load_image_file("photos/narsimha.jpg")
narsimha_encoding = face_recognition.face_encodings(narsimha_image)[0]

uday_image = face_recognition.load_image_file("photos/uday.jpg")
uday_encoding = face_recognition.face_encodings(uday_image)[0]

lokesh_image = face_recognition.load_image_file("photos/lokesh.jpg")
lokesh_encoding = face_recognition.face_encodings(lokesh_image)[0]

known_face_encoding = [
narsimha_encoding,
uday_encoding,
lokesh_encoding
]

known_faces_names = [
"Narsimha",
"Uday",
"Lokesh"
]

students = known_faces_names.copy()

face_locations = []
face_encodings = []
face_names = []
s=True
```

```
now = datetime.now()
current_date = now.strftime("%Y-%m-%d")

f = open(current_date+'.csv','w+',newline = '')
str=current_date+'.csv'
csv_file = str
excel_file = 'f'
# workbook = openpyxl.load_workbook(excel_file)
# sheet = workbook.active
lnwriter = csv.writer(f)
l1=[]
```

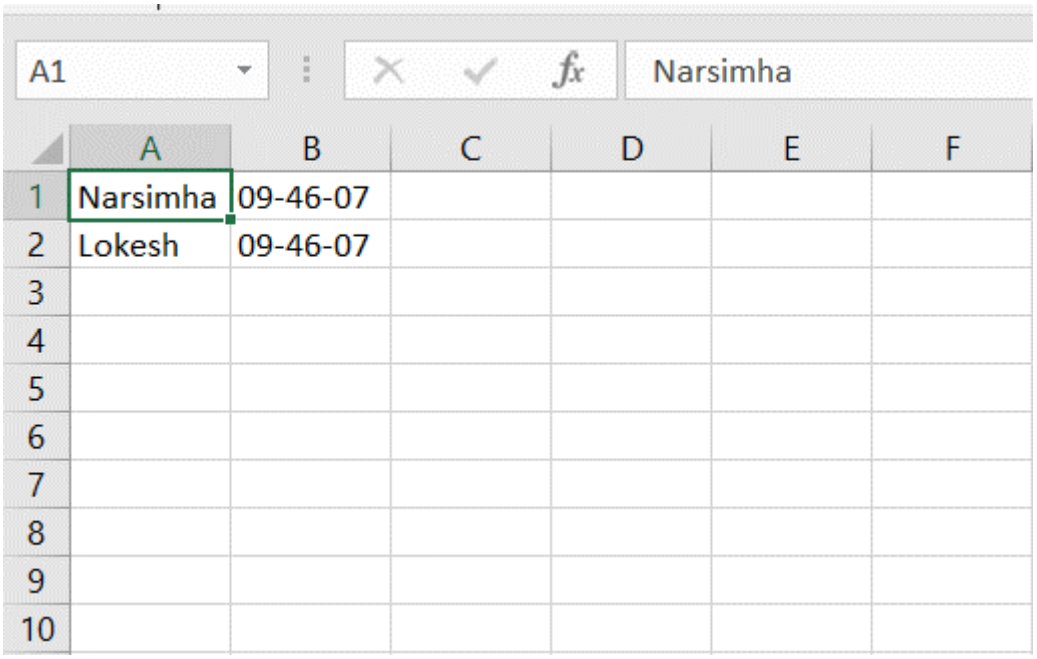
```
while True:
    _,frame = video_capture.read()
    small_frame = cv2.resize(frame,(0,0),fx=0.25,fy=0.25)
    rgb_small_frame = small_frame[:,:,::-1]
    if s:
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame,face_locations)
        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(known_face_encoding,face_encoding)
            name=""
            face_distance = face_recognition.face_distance(known_face_encoding,face_encoding)
            best_match_index = np.argmin(face_distance)
            if matches[best_match_index]:
                name = known_faces_names[best_match_index]

            face_names.append(name)
            if name in known_faces_names:
                font = cv2.FONT_HERSHEY_SIMPLEX
                bottomLeftCornerOfText = (10,100)
                fontScale              = 1.5
                fontColor              = (255,0,0)
                thickness              = 3
                lineType               = 2

                cv2.putText(frame,name+' Present',
                    bottomLeftCornerOfText,
                    font,
                    fontScale,
                    fontColor,
                    thickness,
                    lineType)

                if name in students:
                    students.remove(name)
                    print(students)
                    current_time = now.strftime("%H-%M-%S")
```

```
                if name in students:
                    students.remove(name)
                    print(students)
                    current_time = now.strftime("%H-%M-%S")
                    new_data = [
                        [ name,current_time]
                    ]
                    print(name,current_time)
                    l1.append([name,current_time])
                    # lnwriter.writerow([name,current_time])
                    with open(csv_file, mode='a', newline='') as file:
                        writer = csv.writer(file)
                        for row in new_data:
                            writer.writerow(row)
    cv2.imshow("attendence system",frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        print(l1)
        break
#workbook.save(excel_file)
video_capture.release()
cv2.destroyAllWindows()
f.close()
```

5.5 UI SCREENSHOTS:




```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH TERMINAL OUTPUT
[Running] python -u "c:\Users\Lokesh Dahake\Downloads\BackUp\program.py"
['Uday', 'Lokesh']
Narsimha 09-46-07
['Uday']
Lokesh 09-46-07

[Done] exited with code=1 in 121.083 seconds
```


A1
Narsimha
A B C D E F
1 Narsimha 09-46-07
2 Lokesh 09-46-07
3
4
5
6
7
8
9
10

CHAPTER-6

6. SOFTWARE TESTING

6.1 INTRODUCTION

So, our proposed method is Automating the attendance system.

• In order to automate we use OpenCV for capturing an image from cam before capturing the students face, they should enter their roll no and name, so that it feeds the database (CSV file)

• After the model is trained when student comes to class before he needs to mark his attendance by face recognizing.

• The wait time of the model is 100 sec. After successful scanning of the students face the dialogue box appears that attendance marked successfully.

• Faculty can easily take the attendance excel sheet where it provides student data and time, he signed into class based on that attendance can be uploaded into college database.

6.1.1 Testing Objectives:

The primary objectives of software testing for this project include:

Verification of Object Detection: Confirm the correct detection of faces through the AI model.

Integration Testing: Ensure smooth integration between the AI model to recognition face by the processing pipeline.

Performance Evaluation: Assess the system's ability to process real-time faces and maintain accuracy under various scenarios

6.1.2 Testing Strategies:

To achieve the testing objectives, the following strategies will be employed:

Unit Testing: Validate individual components, such as the model, face detection CNN, and image processing functions, to ensure their correctness.

Integration Testing: Confirm the seamless integration of different components to guarantee proper communication and functionality.

System Testing: Evaluate the system's overall performance, accuracy, and responsiveness in real-time processing.

Performance Testing: Assess the system's efficiency under different loads, video resolutions, and frame rates.

User Acceptance Testing (UAT): Gather feedback from users to ensure the system meets their expectations and requirements

6.1.3 System Evaluation:

System evaluation will focus on:

Accuracy: Validate that detected objects align with ground truth, and the face detection model correctly identifies faces.

Robustness: Test the system's resilience to variations in lighting and face quality.

Real-time Processing: Assess the system's ability to process video frames in real-time without significant delays.

6.2 Test Cases:

Test Cases for the face Detection and Recognition:

Test Case 1:

Input: A photos folder containing a photo of a students.

Expected Output:

- The face should be clear.
- And it compares the face with photo which contain in folder. After that it marked attendance

CONCLUSION

- In conclusion, the development of an AI-based attendance system represents a transformative approach to traditional attendance tracking, offering efficiency, accuracy, and enhanced user experiences.
- Through an extensive literature survey, we have gained insights into key areas such as facial recognition algorithms, biometrics, computer vision, machine learning, privacy considerations, and user acceptance.
- As the project progresses, several potential future enhancements have been identified, ranging from multimodal biometrics and continuous authentication to mobile integration and AI-driven insights.
- These enhancements not only aim to improve the system's functionality but also to address emerging technological trends and user needs. As we move forward, it is imperative to remain mindful of ethical considerations, user privacy, and legal compliance.
- The successful integration of these advancements, coupled with a commitment to ongoing improvement and user feedback, will contribute to the creation of a robust, secure, and future-ready AI-based attendance system, positively impacting educational institutions, workplaces, and other organizational settings.
- In the broader context, the development of an AI-based attendance system not only streamlines administrative processes but also aligns with the broader trajectory of technological advancement.
- By embracing cutting-edge technologies like facial recognition, machine learning, and biometrics, this project contributes to the ongoing evolution of smart and efficient systems in educational and organizational environments.
- Furthermore, the exploration of diverse modalities, such as behavioural biometrics and augmented reality, underscores a commitment to innovation and adaptability in the face of evolving technological landscapes.
- As the project advances, collaboration with stakeholders and continuous engagement with the latest research will be pivotal in ensuring the system remains at the forefront of advancements.
- Striking a balance between technological sophistication, ethical considerations, and user-centric design principles will be key in realizing the full potential of an AI-based attendance system, fostering a paradigm shift in how attendance is managed and contributing to a more seamless and secure operational landscape.

FUTURE ENHANCEMENTS

Several potential future enhancements can be considered for an AI-based attendance system to further improve its functionality, efficiency, and user experience. Here are some ideas for future development:

1.Multimodal Biometrics:
- Integrate multiple biometric modalities such as facial recognition, fingerprint scanning, and voice recognition for enhanced accuracy and reliability.

2.Edge Computing:
- Implement edge computing capabilities to process data locally on the device, reducing dependency on centralized servers and improving system responsiveness.

3.Continuous Authentication:
- Explore the integration of continuous authentication methods to verify identity throughout the duration of a session, providing an extra layer of security.

4.Behavioral Biometrics:
- Consider incorporating behavioural biometrics, such as typing patterns or mouse movements, to enhance the uniqueness and security of user identification.

5.Predictive Analytics:
- Implement predictive analytics to forecast attendance patterns based on historical data, enabling proactive measures for addressing potential attendance issues.

6.Mobile Integration:
- Develop a mobile application for users to easily check in and access attendance records using their smartphones, enhancing convenience and accessibility.

7.Natural Language Processing (NLP):
- Integrate natural language processing to enable voice commands for attendance checking and reporting, providing a more natural and user-friendly interface.

8.Automated Notifications:
- Implement automated notifications for users and administrators regarding attendance-related information, such as low attendance alerts or personalized messages.

9.AI-driven Insights:
- Use AI algorithms to derive actionable insights from attendance data, helping educators and administrators make informed decisions about resource allocation and scheduling.

These future enhancements can help elevate the AI-based attendance system, making it more versatile, secure, and user-friendly while also expanding its capabilities to address evolving needs in educational and organizational settings.

REFERENCE

- https://ieeexplore.ieee.org/abstract/document/9297643
- https://link.springer.com/chapter/10.1007/978-981-16-8512-5_39
- https://www.python.org/
- "RFID based Student Attendance System" (Hussain, Dugar, Deka, Hannan, 2014