Advance Employee Attendance System

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ICFAI University,Triprua
INDIA – 799210
December – 2023

Advance Employee Attendance System

Report submitted to
ICFAI University,Triprua
for the award of the degree

of

Bachelor of Technology

by

Mrinmoy Biswas(21IUT0020080)
Hrishit Bhattacharjee(21IUT0020083)
Ajay Saha(21IUT0020086)
Subrata Das(21IUT0020090)

Under the guidance of
Mr. Sanjib Debnath
Assistant Professor
ICFAI University, Tripura

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ICFAI University, Tripura
INDIA–799210
DECEMBER 2023

REPORT APPROVAL

Date:..........................

Certified that the report entitled ***‘Online Examination System ’*** submitted by ***Chanchal Bawara(20UCS0133), Raju Kumar(20UCS148),Sushma Debbarma(20UCS143),Ruman Debbarma(20UCS010)*** to the National Institute of Technology, Agartala, for the award of the degree of ***B*****achelor of Technology** in ***C*****omputer Science and Engineering**

Mr. Sanjib Debnath
(**Project Supervisor**)

Dr. Saptarshi Chakraborty
(**Head of the Department, CSE**)

CERTIFICATE

This is to certify that the report entitled ***'Advance Employee Attendence System'***, submitted by **Mrinmoy Biswas(21IUT0020080)**, **Hrishit Bhattacharjee(21IUT0020083)**, **Ajay Saha(21IUT0020086)** and **Subrata Das(21IUT0020090)** to the ICFAI University, Tripura, is a record of bonafide research work, carried out under our supervision and this work has not been submitted elsewhere for a degree.

Mr. Sanjib Debnath
Project Supervisor
Assistant Professor
ICFAI University, Tripura

Acknowledgments

We would like to take this opportunity to express our deep sense of gratitude to all who helped us directly or indirectly during this thesis work. Firstly, we would like to thank our supervisor, **Mr. Sanjib Debnath**, for being a great mentor and the best adviser we could ever have. His advise, encouragement and critics are source of innovative ideas, inspiration and causes behind the successful completion of this report. The confidence shown on us by him was the biggest source of inspiration for us. It has been a privilege working with him from last one year. We are highly obliged to all the faculty members of Computer Science and Engineering Department for their support and encouragement. We also thank **(Dr.) Saptarshi Chakraborty**, H.O.D, ICFAI University,Tripura CSED for providing excellent computing and other facilities without which this work could not achieve its quality goal.

- Mrinmoy Biswas **- Hrishit Bhattacharjee**

- Ajay Saha **- Subrata Das**

ICFAI UNIVERSITY, TRIPURA
Tripura, India - 799210

Contents

List of Figures

List of Tables

Abstract

In colleges, universities, organizations, schools, and offices, recording attendance is a critical daily task typically carried out manually through methods such as calling names or using roll numbers. This project aims to revolutionize this manual process by creating an automated Face Recognition-based attendance system. The project aligns with the need for modernization in attendance management and time efficiency.

The system is implemented in classrooms, where student information, including names, roll numbers, class, section, and photographs, is trained into the system. Image extraction is performed using OpenCV. Prior to the commencement of each class, students can approach the device, which captures and compares their facial images with the qualified dataset. The Logitech C270 web camera and NVIDIA Jetson Nano Developer kit are utilized for image capture and processing.

The image processing involves face identification using a Haarcascade classifier, followed by face recognition using the LBPH (Local Binary Pattern Histogram) Algorithm. The histogram data is then checked against the established dataset, and the system automatically marks attendance. An Excel sheet is generated and updated every hour with information provided by the respective class instructor.

In summary, this project streamlines attendance tracking by employing facial recognition technology, enhancing the efficiency of the process and meeting the demands for modernization and effective time management in educational institutions and workplaces.

Chapter 1

Introduction

In the dynamic landscape of modern workplaces, the conventional methods of employee attendance tracking have become outdated and inefficient. The manual processes of taking attendance not only lack attractiveness but also suffer from consistency issues. Recognizing the need for a trans formative solution, we present the Cloud-Based Face Recognition Attendance Management System, a cutting-edge approach poised to revolutionize the way organizations handle employee attendance.

Gone are the days of manual attendance-taking, where employees were accounted for through laborious processes. The Cloud-Based Face Recognition Attendance Management System stands as a beacon of innovation, providing a seamless and efficient alternative. By leveraging facial recognition technology, this system transcends traditional methods, offering a secure and user-friendly means of identifying and verifying employees.

The system operates on the cloud platform, introducing a new dimension to attendance management. Cloud technology ensures that the attendance data is not only securely stored but also accessible from anywhere with an internet connection. This accessibility enhances real-time data management, enabling organizations to monitor and analyze attendance trends effortlessly.

In the realm of employee attendance, consistency is paramount. The Face Recognition technology, a biometric marvel, ensures accuracy and reliability in the identification process. Unlike traditional methods, it requires minimal effort from the user end, contributing to a more streamlined and efficient attendance management process.

Machine learning, a cornerstone in the realm of technology, plays a pivotal role in this system. By providing the system with a dataset as input, machine learning algorithms enhance the accuracy and effectiveness of attendance tracking. This technology optimizes the overall functionality of the Cloud-Based Face Recognition Attendance Management System.

The Haar Cascade method, employed for image analysis, complements the facial recognition process, contributing to the system's proficiency in recognizing and verifying employee faces. This further solidifies the reliability of the system, mitigating concerns related to fake attendance and providing an additional layer of security.

Time is of the essence in the fast-paced corporate world, and this system addresses the challenges related to time management seamlessly. The transition to a cloud-based infrastructure not only saves time but also adds a layer of flexibility and scalability, making it adaptable to the evolving needs of organizations.

In conclusion, the Cloud-Based Face Recognition Attendance Management System marks a significant leap forward in employee attendance tracking. By amalgamating facial recognition technology, machine learning, and cloud computing, this system promises to redefine the efficiency, accuracy, and security of attendance management in the workplace. Join us on this journey toward a future where the management of employee attendance is not just a task but a seamlessly integrated, secure, and intelligent process.

1.1 Background

In the realm of daily professional life, the importance of face recognition cannot be overstated, as it serves as a vital tool for identifying familiar faces among colleagues and employees. Often, we might overlook the intricate steps involved in this process. Human intelligence effortlessly processes information by interpreting visual cues, and the recognition process begins with the reception of light-form images on the retina.

The work of Robinson-Riegler and Robinson-Riegler (2008) highlights that, after visual processing, our minds engage in classifying the shape, size, contour, and texture of objects to analyze the received information. This analyzed data is then compared to stored representations in our memory for recognition. While this is a complex task for automated systems, computers equipped with extensive memory, high processing speed, and computational power become indispensable in overcoming human limitations.

The human face, being a unique identifier, forms the basis of biometric face recognition. This method involves identifying individuals by comparing real-time captured facial images with stored images in a database (Margaret Rouse, 2012).

In contemporary workplaces, the Face Recognition Attendance Management System has gained traction for its simplicity and remarkable performance. Applications range from enhancing security at the workplace to streamlining attendance tracking and access control. Major corporations, such as Intel and Apple, have integrated face recognition to bolster security measures and simplify user authentication.

The historical evolution of face recognition research, which began in the 1960s, saw pioneering work by Woody Bledsoe, Helen Chan Wolf, and Charles Bisson. Their system required

administrators to manually locate facial features, setting the groundwork for subsequent developments. Goldstein, Harmon, and Lesk expanded this in 1970 by incorporating additional features like hair color and lip thickness. By 1988, Kirby and Sirovich proposed using Principal Component Analysis (PCA) for automated face recognition. Since then, continuous research efforts have propelled the field to new heights (Ashley DuVal, 2012).
Today, face recognition has become integral to various aspects of employee management. From ensuring workplace security to facilitating user-friendly authentication processes, this technology continues to redefine our interactions and perceptions in the ever-evolving professional landscape. As technology advances, face recognition stands as a cornerstone, reshaping the way we identify and engage with our colleagues in the workplace.

1.2 Goal

The overarching goal of implementing an advanced Employee Attendance Management System is to elevate the efficiency, accuracy, and security of attendance tracking within our organization. By embracing automation and leveraging technologies such as face recognition, our aim is to streamline administrative tasks, eliminating manual inefficiencies associated with traditional attendance methods. This system is designed to enhance accuracy in recording work hours, ensuring precise payroll calculations and promoting transparency in employee attendance records.
In the contemporary landscape of diverse work scenarios, including remote work and flexible schedules, our goal is to create an adaptable system that aligns with modern work environments. The introduction of real-time accessibility to attendance data and seamless integration with HR systems empowers our administrators with dynamic insights for informed decision-making.
Security is a paramount consideration, and the implementation of advanced security measures, including face recognition technology, is intended to fortify the system against identity fraud and unauthorized access. We are committed to safeguarding sensitive employee data, adhering to privacy regulations, and maintaining the highest ethical standards in the use of biometric information.
This initiative is driven by a commitment to user-friendliness, aiming to foster widespread adoption by both employees and administrators. Scalability is a key aspect, ensuring that our attendance management system remains effective and adaptable as our organization evolves.
In evaluating the total cost of ownership, our goal is to implement a cost-efficient solution that provides long-term value through improved operational efficiency and effectiveness in attendance management. This commitment aligns with our vision for responsible and modern workforce management practices.

1.3 MOTIVATION

The motivation behind introducing an advanced Employee Attendance Management System is deeply rooted in our commitment to organizational excellence and the well-being of our workforce. Recognizing the inefficiencies and limitations of traditional manual attendance tracking, our aspiration is to propel our operations into the future by embracing cutting-edge technology. This initiative is fueled by the desire to enhance efficiency, accuracy, and adaptability within our workplace.

Efficiency lies at the forefront of our motivation, as we seek to alleviate the administrative burdens associated with manual attendance tracking. By automating this process, we aim to empower our administrative teams to focus on more strategic tasks that contribute to the overall success of the organization. The pursuit of precision and accuracy in attendance records is driven by our dedication to fostering a workplace culture built on transparency and fairness, where every employee's contribution is accurately reflected in payroll calculations.

In response to the evolving landscape of work practices, our motivation extends to creating a system that seamlessly adapts to modern work scenarios. This includes catering to remote work arrangements, flexible schedules, and alternative work hours. Moreover, our commitment to security is a driving force, compelling us to implement advanced security measures, such as face recognition, to safeguard sensitive employee data and ensure compliance with privacy regulations.

The introduction of real-time data access aligns with our motivation to equip administrators with timely insights, facilitating informed decision-making. Simultaneously, the user-friendly design of the system is motivated by the understanding that technology adoption is most successful when it minimizes resistance to change. By prioritizing a positive user experience, we aim to instill confidence in our employees and administrators, fostering widespread acceptance of the new attendance management system.

Looking ahead, our motivation is guided by a future-ready perspective, considering scalability to accommodate organizational growth and changing needs. Additionally, the pursuit of cost-efficiency underscores our commitment to financial responsibility, ensuring that the implementation of this system not only reduces operational costs but also provides enduring value.

In essence, our motivation is deeply rooted in the belief that this advanced Employee Attendance Management System will not only enhance operational efficiency but also contribute to creating a work environment that is technologically advanced, adaptable, and employee-centric. This initiative embodies our commitment to continuous improvement and the harnessing of technology to elevate both our organizational processes and the experiences of our valued workforce.

1.4 Contribution of Advance Employee Attendance System

The introduction of an advanced Employee Attendance Management System is poised to make a substantial contribution to our organization across various dimensions, enhancing efficiency, transparency, and overall workforce management. Several key contributions can be anticipated from the implementation of this innovative system:

1.4.1 Operational Efficiency

The foremost contribution lies in the significant enhancement of operational efficiency. By automating attendance tracking processes, manual efforts will be minimized, freeing up valuable time for administrators to focus on more strategic tasks. This efficiency gain translates to increased productivity across the organization.

1.4.2 Accuracy and Payroll Precision

The system's contribution to accuracy in attendance tracking ensures precise payroll calculations. Eliminating the potential for errors associated with manual methods contributes to fair and transparent compensation practices, fostering trust among employees.

1.4.3 Adaptability to Modern Work Practices

The system contributes to the adaptability of our organization to modern work practices. In an era where flexible schedules and remote work are prevalent, the system ensures seamless attendance tracking regardless of the work scenario, supporting a diverse and dynamic workforce.

1.4.4 Security and Compliance

The implementation of advanced security measures, including face recognition technology, contributes significantly to data security and compliance. Safeguarding sensitive biometric data ensures the organization remains in adherence to privacy regulations, fostering trust and compliance with ethical standards.

1.4.5 Real-Time Decision-Making

The introduction of real-time data access contributes to more informed decision-making. Administrators gain immediate insights into attendance patterns, allowing for proactive responses to workforce trends and facilitating strategic decision-making processes.

1.4.6 User Adoption and Satisfaction

The system's user-friendly interface contributes to high user adoption rates and overall user satisfaction. By prioritizing positive user experiences, the system encourages widespread acceptance among both employees and administrators, minimizing resistance to technological change.

1.4.7 Scalability and Future-Readiness

The contribution to scalability ensures the system's effectiveness over the long term. As the organization evolves, the system can adapt and scale to meet changing needs, providing a future-ready solution that aligns with organizational growth.

1.4.8 Cost-Efficiency and Value Creation

The system contributes to cost-efficiency by minimizing the total cost of ownership. This contribution extends beyond immediate operational savings, creating enduring value through improved efficiency, reduced administrative costs, and enhanced workforce management practices.

1.4.9 Enhanced Data-Driven Insights

The system's contribution to data-driven insights empowers administrators with a deeper understanding of attendance patterns. This contributes to strategic workforce planning, helping the organization make informed decisions related to resource allocation and talent management. In essence, the implementation of an advanced Employee Attendance Management System is not merely a technological upgrade but a transformative initiative that contributes to the overall success, adaptability, and employee satisfaction within our organization.

1.5 Problem Statement

In the traditional landscape of employee attendance management, manual methods have long been the norm, requiring time-consuming processes such as roll call or manual entry. This conventional approach, while prevalent, is fraught with inefficiencies, inaccuracies, and security concerns. The need for a more sophisticated and streamlined system has become increasingly evident, prompting the exploration of automated solutions.

One of the primary challenges faced by organizations is the inconsistency and time-consuming nature of manual attendance tracking. Relying on manual methods not only consumes valuable time for both employees and administrators but also opens the door to errors, be it through miscommunication or intentional falsification of attendance records. This inconsistency can lead to inaccurate payroll calculations, creating challenges in maintaining an equitable and transparent work environment.

Moreover, in the current landscape where workplaces are evolving, embracing remote and flexible work arrangements, traditional attendance methods fall short. Remote work scenarios, varying work hours, and flexible schedules demand a more dynamic and adaptable attendance management system. The need for a solution that accommodates diverse work arrangements and ensures accurate attendance tracking has never been more crucial.

Security concerns also come to the forefront, especially in environments where sensitive information is handled. Traditional methods of attendance tracking, relying on physical identifiers like cards or manual sign-ins, are susceptible to identity fraud or unauthorized access. Ensuring the security and authenticity of attendance records is paramount, and the conventional approaches often lack the robust mechanisms required to address these concerns effectively.

In light of these challenges, there is a pressing need for an advanced Employee Attendance Management System that leverages technology such as face recognition, cloud computing, and machine learning. This system should not only automate attendance tracking but also address the issues of inconsistency, adaptability to modern work environments, and enhanced security. Developing and implementing such a system becomes essential for organizations striving for efficiency, accuracy, and security in their employee attendance management processes.

Chapter 2

LITERATURE REVIEW

In this paper, "Individual Stable Space: An Approach to Face Recognition Under Uncontrolled Conditions" by Xin Geng, it is mentioned that most face recognition systems require faces to be input based on specific rules, such as controlled illumination, a particular position, a specific view angle, and without any obstacles. These systems are known as face recognition under controlled conditions. However, these rules limit the use of face recognition in many real-time applications as they cannot meet these requirements. Real-time applications need techniques that do not require strict control over individuals to recognize faces. Systems of this type require face recognition under uncontrolled conditions. Therefore, this paper proposes such a system, but the system has a drawback in that it requires an image as input and one person per image, which hinders its use in real-time applications like attendance systems.[1].

The authors have proposed a model for an automated attendance system. The model focuses on how face recognition, combined with Radio Frequency Identification (RFID), identifies authorized students and keeps a count as they enter and exit the classroom. The system maintains an authentic record for each registered student. Additionally, it stores the data of every student registered for a specific course in the attendance log and provides necessary information as required.[2].

In this paper, the authors have designed and implemented an attendance system that utilizes iris biometrics. Initially, participants were instructed to register their details, including their unique iris template. During attendance, the system automatically captured the eye image of each participant, recognized their iris, and searched for a match in the created database to record class attendance. The prototype was web-based[3].

In the paper titled "Anti-Cheating Presence System Based on 3WPCA-Dual Vision Face Recognition," Edy Winarno proposed a system that can anticipate cheating in a facial recognition-based system, such as using a photograph of an authorized person or an image similar to the authorized person. They utilized a dual vision camera, also known as a stereo vision camera, which produces one image from each of its two lenses. After obtaining the two images, they used a half-join method to combine the left half of the left image and the right half of the right image of a person into a single image of the person. This composite image then undergoes extraction using the 3WPCA method. The system achieved a 98% accuracy rate in recognizing cheating using this approach[4].

In this paper, the author has designed and explained the improvement of a picture-based attendance system that captures the faces of many students and may be the next generation to all the biometric devices that are currently prevalent. The human face, being a distinctive feature, exhibits a high degree of variability, requiring a fast and accurate method for detecting students' facial structures. The system's processing involves registering students by capturing their images and then using them for attendance tracking. Continuous registration is necessary to achieve high and precise accuracy. This paper outlines the system, and supporting evidence will be provided to reinforce its effectiveness. The project can be utilized in online examinations for certification, aiding in the identification of the student taking the test[5].

The researchers explored and evaluated the best facial recognition algorithms, namely Eigenface and Fisherface, provided by OpenCV 2.4.8. They conducted experiments comparing the Receiver Operating Characteristics (ROC) curve and subsequently implemented the chosen algorithm in the attendance system. According to the results obtained in this study, the ROC curve demonstrated that Eigenface outperformed Fisherface. The system, implemented using the Eigenface algorithm, achieved an accuracy rate ranging from 70% to 90%[6].

In 2019, Nandhini R. introduced an Attendance System based on face recognition. This system captures the video of the students, converts it into frames, and stores it in the database. Additionally, the Convolutional Neural Network (CNN) algorithm is employed to detect faces. This system contributes to enhancing accuracy and speed[7].

The authors proposed a method for a student attendance system in the classroom using face recognition techniques by combining Discrete Wavelet Transforms (DWT) and Discrete Cosine Transform (DCT). These algorithms were employed to extract the features of the student's face, followed by applying Radial Basis Function (RBF) for classifying the facial objects. This system achieved an accuracy rate of 82%[8].

Chapter 3

Methodology and Experimental Setup

3.1 Objective

The primary objective of implementing a modern face recognition-based attendance system is to overcome the limitations and inefficiencies associated with traditional attendance marking methods in educational institutions and organizations. The key objectives include:

1. **Automation of Attendance Recording:** Replace manual attendance marking with an automated system that utilizes face recognition technology to identify and record the presence of individuals.

2. **Time Efficiency:** Streamline the attendance process to save time for both faculty members and employees. The automated system aims to be faster and more efficient than traditional methods.

3. **Accuracy Improvement:** Enhance the accuracy of attendance records by leveraging face recognition algorithms, reducing the likelihood of errors and instances of proxy attendance.

4. **Minimization of Administrative Burden:** Alleviate the administrative burden on faculty members, teachers, and HR personnel by automating the attendance management process.

5. **Security Enhancement:** Implement a secure and reliable system that utilizes biometric data for identification, reducing the chances of fraudulent attendance practices.

6. **Integration with Database Systems:** Integrate the attendance system with databases to maintain a centralized and organized record of attendance data, making it easily accessible for reporting and analysis.

7. **User-Friendly Interface:** Develop a user-friendly interface for students, employees, and administrators to interact with the attendance system seamlessly.

8. **Cost-effectiveness:** Evaluate and implement a cost-effective solution that provides long-term benefits in terms of time and resource savings.

9. **Adaptability:** Design the system to be adaptable and scalable for implementation in various educational institutions, organizations, and diverse settings.

10. **Compliance with Privacy Regulations:** Ensure that the implementation of the face recognition attendance system complies with privacy regulations and guidelines to protect individuals' sensitive information.

By addressing these objectives, the aim is to create a more efficient, accurate, and secure attendance management system that aligns with modern technological advancements.

3.2 Research Methods

Face detection involves the process of categorizing image windows into two classes: one containing faces and the other representing the background (clutter). This task is challenging due to the variations in age, skin color, and facial expressions among faces. Complications arise from diverse lighting conditions, varying image qualities and geometries, as well as the potential for partial occlusion and disguises. An ideal face detector should be capable of identifying faces under any lighting conditions and against any background.

The face detection task can be divided into two steps. The first step is a classification task that takes an arbitrary image as input and outputs a binary value (yes or no) indicating whether faces are present. The second step is the face localization task, aiming to take an image as input and provide the location of any faces through bounding boxes (x, y, width, height).

After capturing an image, the system will compare it with pictures in its database, delivering the most relevant result. The hardware components include the NVIDIA Jetson Nano Developer Kit, Logitech C270 HD Webcam, and the programming will be done using the Python language with the OpenCV platform.

3.3 Modal Implementation:

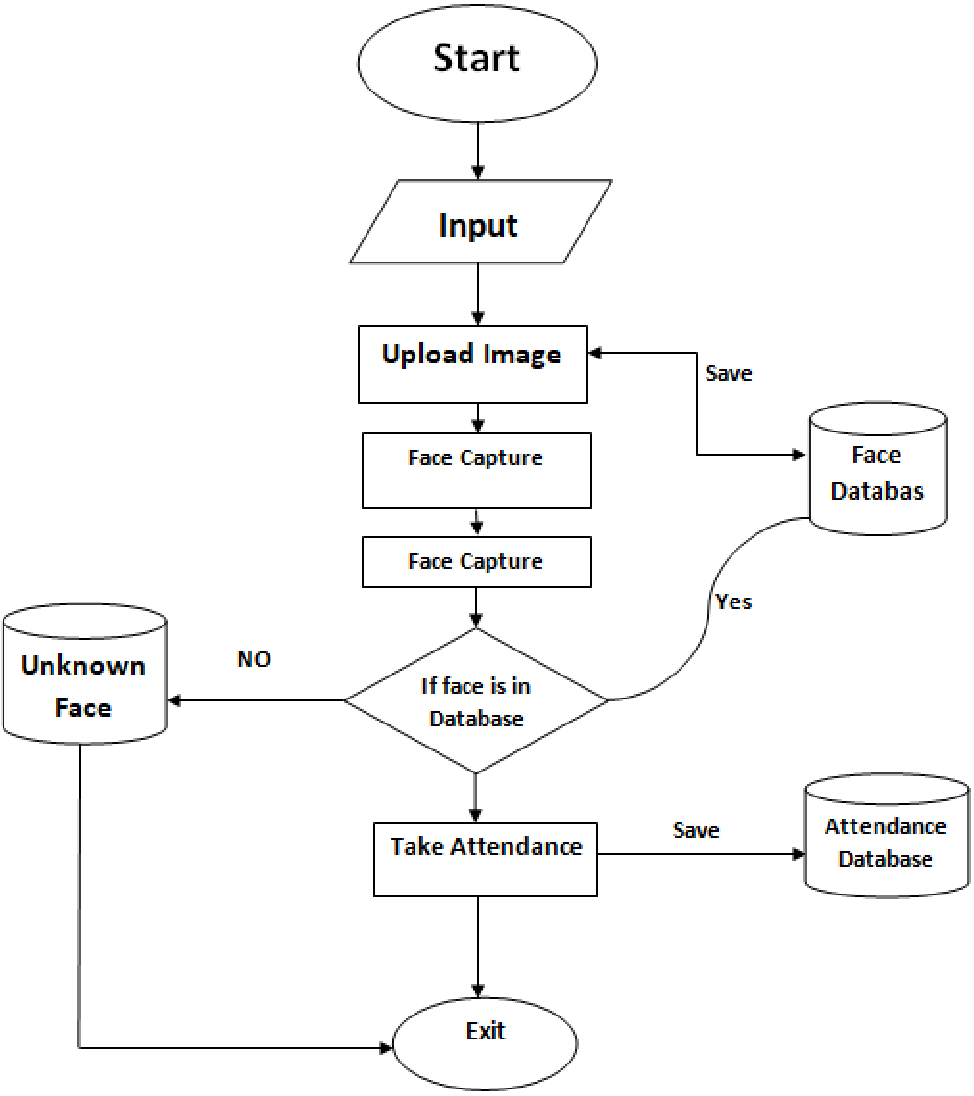


Figure 3.1: **Model Implementation**

3.4 Existing System Flow Chart

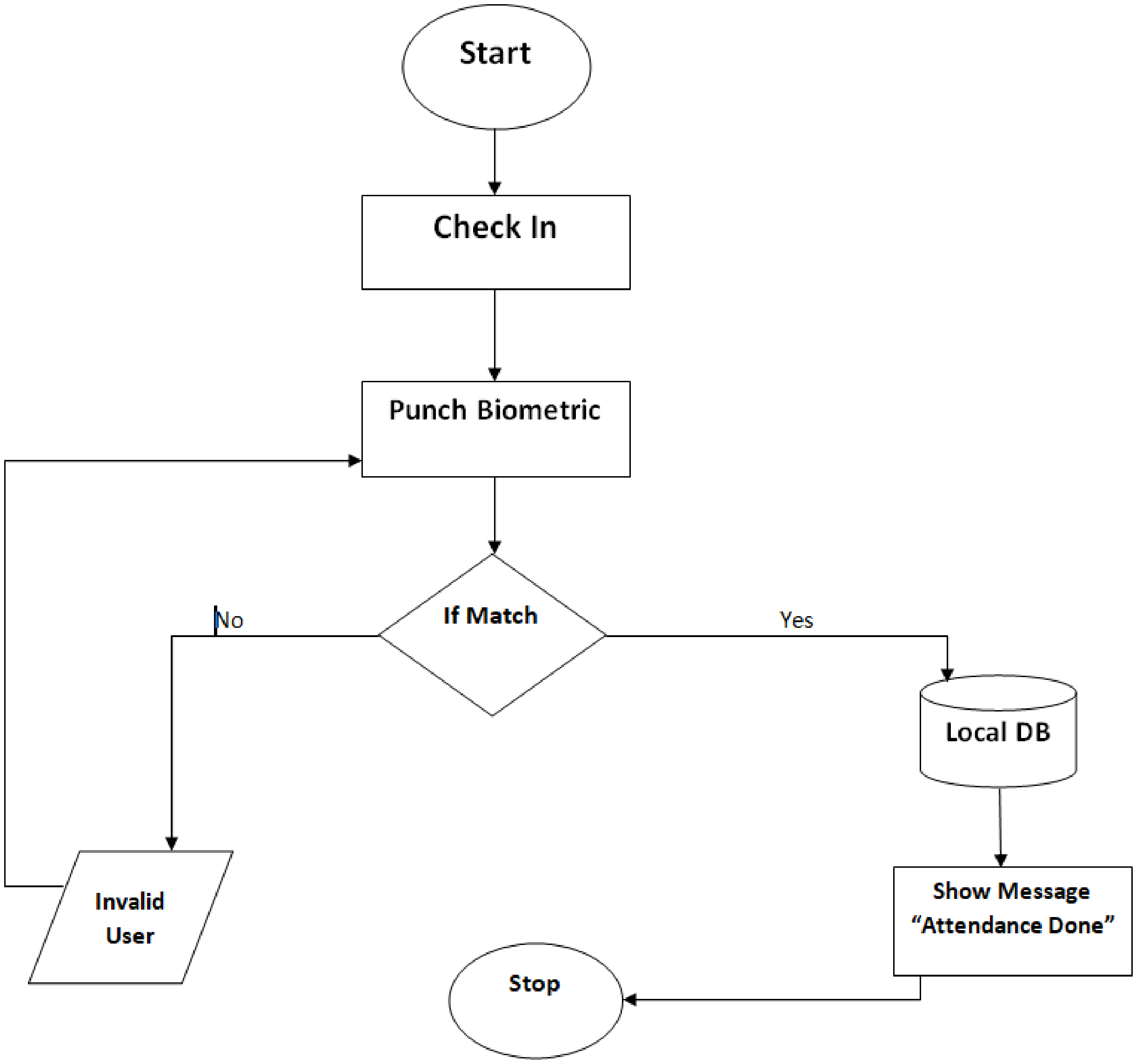


Figure 3.2: **Flow Chart of Existing System**

3.5 Detriments of Existing System

A register-based attendance system is a traditional method of tracking attendance in educational institutions or workplaces. In this system, attendance is manually recorded in a physical register or logbook. Here's an overview of how a register-based attendance system typically works:

1. **Manual Entry:** When individuals enter a class or workplace, they sign or mark their attendance in a register.
2. **Unique Identifiers:** Each person may have a unique identifier such as an employee ID, student ID, or name, which is used to associate the attendance record with the individual.
3. **Date and Time:** The date and time of entry are usually recorded alongside the person's identifier.
4. **Accuracy and Verification:** The accuracy of this system relies on individuals accurately marking their attendance. Verification may be challenging, and the system is susceptible to errors, intentional or unintentional.
5. **Processing:** At the end of a specific period (daily, weekly, etc.), the attendance data from the register may be compiled and processed to generate reports.

While register-based systems are simple and easy to implement, they have several drawbacks:

- **Manual Labor:** The process is labor-intensive and requires someone to manage and maintain the physical register.
- **Prone to Errors:** There is a risk of errors in data entry, and individuals may mark attendance on behalf of others (proxy attendance).
- **Limited Accessibility:** Accessing real-time attendance information or generating reports may be time-consuming.

3.6 Proposed System

The proposed system aims to streamline and enhance the attendance management process through a comprehensive cloud-based solution. The system encompasses the following steps:

1. **Registration of Attendance in Cloud-Based System:** Employees initiate the attendance registration process through a cloud-based system. This centralized system ensures real-time access and management of attendance data from any location. Employees can easily log their attendance by accessing the system using secure credentials.

2. **Role and Location Update:** The system allows employees to update their roles and locations as needed. This feature ensures that the attendance data is contextually relevant, accommodating scenarios where employees may have different roles or work from multiple locations.

3. **Image Upload for Face Recognition:** Employees are required to upload their facial images during the registration process. These images serve as the basis for face recognition, a robust and secure method for verifying identity. The system ensures the confidentiality and integrity of uploaded images.

4. **Registration Number Input:** After the initial registration, employees are assigned unique registration numbers. This number acts as a key identifier for the face recognition system. Employees input their registration numbers when prompted, facilitating a seamless connection between their identity and the stored facial image.

5. **Face Recognition through Camera:** The system utilizes a camera for real-time face recognition during the attendance marking process. When employees present themselves to the camera, the system captures and analyzes their facial features. The pre-uploaded facial image is compared with the real-time facial data obtained from the camera.

6. **Attendance Marking based on Recognition:** If a match is successfully identified through face recognition, the attendance is automatically marked for the corresponding employee. This automated process eliminates the need for manual attendance tracking and minimizes errors associated with traditional methods.

7. **Non-Matching Scenarios:** In cases where the facial features do not match the pre-registered image, the attendance remains unmarked. This non-intrusive approach ensures that attendance records are accurate and reliable, with potential false positives minimized.

The proposed system combines the convenience of cloud-based technology with the accuracy of facial recognition, providing an efficient and secure solution for attendance management. By incorporating these steps, the system aims to optimize the overall attendance tracking process while ensuring a user-friendly experience for employees.

3.6.1 Advance Attendance Management System Algorithm

```
Step 1: Register Attendance in Cloud-Based System
function registerAttendance(employeeDetails):
    # Store employee details in the cloud database
    cloudDB.storeEmployeeDetails(employeeDetails)

Step 2: Update Role and Location
function updateRoleAndLocation(employeeID, newRole,
   newLocation):
    # Update employee's role and location in the cloud
       database
    cloudDB.updateEmployeeRoleAndLocation(empID,newRole,
       newLoc)

Step 3: Upload Image
function uploadImage(employeeID, facialImage):
    # Upload employee's facial image to the cloud database
    cloudDB.uploadFacialImage(employeeID, facialImage)

Step 4: Registration Number Input
employeeRegistrationNumber = input("Enter your registration
   number: ")

Step 5: Camera Open Face Recognition
capturedFaceImage = camera.captureFaceImage()

Step 6: Face Recognition Process
function recognizeFace(employeeRegistrationNumber,
   capturedFaceImage):
    # Retrieve employee's facial image from the cloud
    storedFacialImage = cloudDB.getFacialImage(employeeReg No
       )

    # Perform face recognition
    similarityScore = faceRecognitionAlgorithm(capFaceImg,
       storedImg)
    # Step 7: Attendance Marking
    if similarityScore > similarityThreshold:
        markAttendance(employeeRegistrationNumber)
        logAttendance(employeeRegistrationNumber, "Attendance
            marked")
    else:
        logAttendance(empid, "Attendance not marked")

Face Recognition Algorithm
function faceRecognitionAlgorithm(capFaceImg,FacialImg):
    # Implementation of face recognition algorithm
    # Compare facial features and calculate similarity score
    # Return similarity score

Step 7: Attendance Marking
function markAttendance(employeeRegistrationNumber):
    # Mark attendance with timestamp
    timestamp = getCurrentTimestamp()
    cloudDB.markAttendance(employeeRegistrationNumber,
       timestamp)

# Logging Function
function logAttendance(employeeRegistrationNumber, logMessage
   ):
    # Log attendance details with the provided message
    logging.log(logMessage)

# Utility Function to Get Current Timestamp
function getCurrentTimestamp():
    # Implementation to get the current timestamp
    return currentTimestamp
```

3.6.2 Advance Attendance Management System Flow Chart

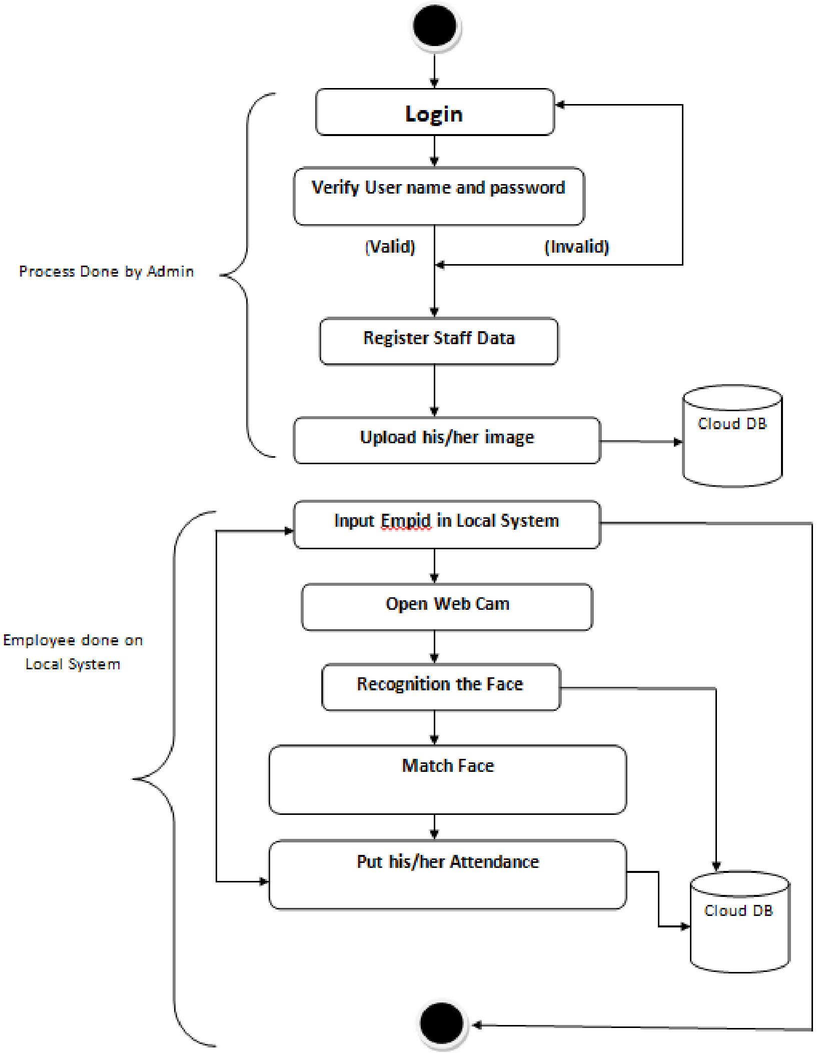


Figure 3.3: **Flow Chart of Advance Attendance System**

3.7 Proposed System Architecture

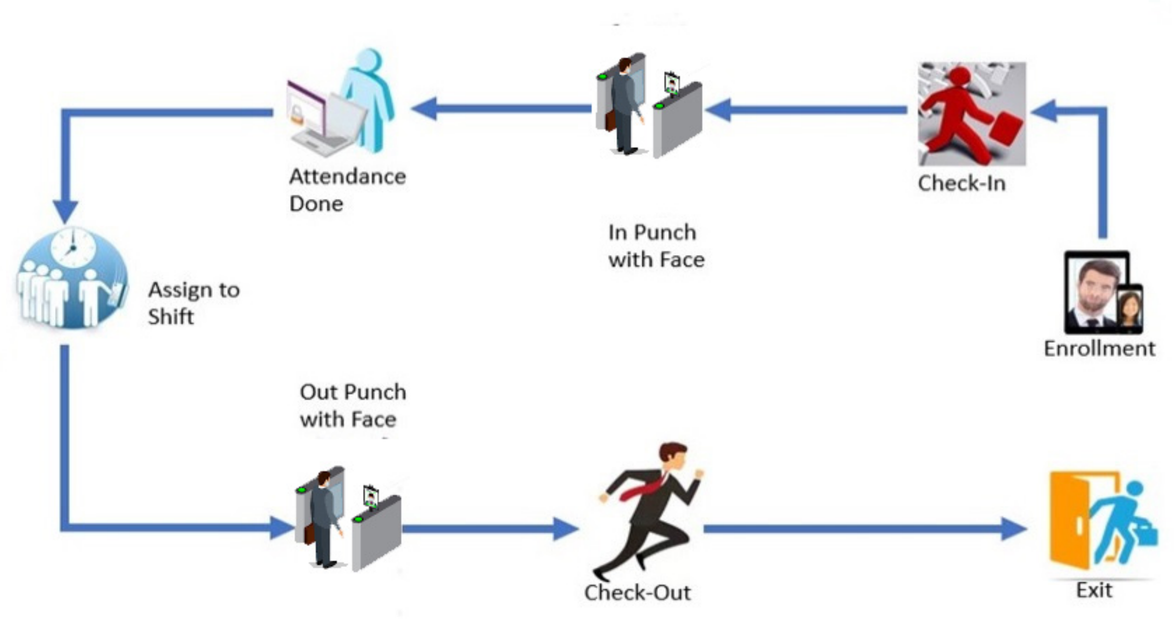


Figure 3.4: **System Architecture of Advance Attendance System**

3.8 Equipment Requirement for Cloud

- Core i5 (Processor)
- 2 GB RAM
- 512 KB Cache Memory
- Hard disk 40 gb

3.9 Technology Requirement

- Operating System : Windows
- Web-Technology: PHP

- Front-End:HTML,CSS,JAVASCRIPT
- Back-End: MySQL
- Web Server: Apache SERVER.
- Python 3.9

3.10 Library Requirement

- **Numpy:** The NumPy library is a fundamental requirement for various tasks related to face recognition in Python. NumPy provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. Face recognition libraries, such as face recognition, often utilize NumPy for efficient handling and processing of image data.
  Following are some common ways in which NumPy is used in face recognition tasks:

  1. Image Representation: Images are represented as NumPy arrays in face recognition systems. Each pixel's color information is stored in a NumPy array, allowing for easy manipulation and analysis.
  2. Array Operations: NumPy provides a wide range of array operations that are essential for processing and manipulating the pixel data in facial images. Operations like cropping, resizing, and normalization are commonly performed using NumPy functions.
  3. Vectorized Operations: Face recognition algorithms often involve complex mathematical operations on image data. NumPy's vectorized operations significantly improve the efficiency of these calculations, making the processing faster and more scalable.
  4. Matrix Operations: Linear algebra operations, such as matrix multiplication and eigenvalue decomposition, are employed in certain face recognition techniques. NumPy's support for these operations is crucial for implementing such algorithms.

- **Face_Recognition:** The face_recognition library is a popular Python library used for face recognition tasks. It simplifies the process of face recognition by providing a high-level interface built on top of the dlib library. Here are some reasons why face_recognition is commonly used for face recognition:

1. Simplicity and Ease of Use: face_recognition provides a simple and easy-to-use interface for face recognition tasks. It abstracts away many of the complexities involved in face recognition algorithms, making it accessible for developers without deep knowledge of computer vision.
2. Integration with dlib: Under the hood, face_recognition utilizes the dlib library, which is a powerful toolkit for machine learning and computer vision. dlib includes implementations of various machine learning algorithms, including facial landmark detection and deep metric learning, which are essential components of face recognition systems.
3. Facial Landmark Detection: face_recognition includes functionality for detecting facial landmarks, such as the locations of eyes, nose, and mouth. These landmarks are crucial for accurately aligning faces and extracting facial features.
4. Pre-trained Models: The library comes with pre-trained models for face recognition. These models have been trained on large datasets and can recognize faces in various conditions, making it convenient for developers to get started without training their models from scratch.
5. Community Support and Documentation: face_recognition has gained popularity in the developer community, leading to extensive documentation, tutorials, and community support. This makes it easier for developers to find resources and solutions when working on face recognition projects.
6. Cross-Platform Compatibility: The library is designed to work on multiple platforms, including Linux, macOS, and Windows, making it versatile for a wide range of applications.

It's worth noting that while face_recognition provides a convenient interface for face recognition, the underlying dlib library is a powerful tool for various computer vision tasks beyond face recognition. Developers can leverage the capabilities of dlib for more advanced applications if needed.

- **dlib:** The dlib library is a C++ toolkit that includes machine learning algorithms and tools for computer vision tasks. It is commonly used for various applications, and in the context of face recognition, dlib provides several essential functionalities. Here are some reasons why dlib is often used as a requirement for face recognition:

   1. **Facial Landmark Detection:** dlib includes a facial landmark detector that can identify key points on a face, such as the locations of eyes, nose, mouth, and other facial features. This information is crucial for aligning faces and extracting discriminative features.

2. **Deep Metric Learning:** The library incorporates deep metric learning techniques, which are essential for creating embeddings that represent faces in a high-dimensional space. These embeddings enable accurate face comparisons and recognition.
3. **Support for Modern Machine Learning Techniques:** dlib supports various machine learning techniques, including Support Vector Machines (SVM) and deep learning. These techniques are employed for training face recognition models on large datasets.
4. **Pre-trained Models:** dlib comes with pre-trained models for face recognition, allowing developers to leverage these models without the need to train a recognition system from scratch. These models are often trained on diverse datasets to handle different lighting conditions, poses, and facial expressions.
5. **Cross-Platform Compatibility:** dlib is designed to be cross-platform, making it compatible with different operating systems such as Linux, macOS, and Windows. This ensures that face recognition applications built using dlib can be deployed on a variety of platforms.
6. **Community and Research Support:** dlib has gained popularity in the research and developer communities. It is actively maintained, has extensive documentation, and is used in a wide range of applications beyond face recognition.

While dlib is a powerful library, it's worth noting that it is primarily written in C++. Python bindings are provided to make it accessible in Python projects, and the face_recognition library, which builds on dlib, offers a more user-friendly interface specifically for face recognition tasks in Python.

- **mysql.connector:** To implement a cloud-based employee attendance system using MySQL and the mysql.connector library in Python, you can follow these steps:

1. **Database Setup:** Begin by setting up a MySQL database in the cloud. This involves creating a database and tables to store employee information and attendance records. Define the necessary fields such as employee ID, name, date, and time.
2. **Python Script:** Use the mysql.connector library in Python to connect to the MySQL database from your script. Ensure that you have the library installed by running pip install mysql-connector-python. **Attendance Registration:** Write functions in your Python script to handle attendance registration. This includes capturing employee information, verifying identity (e.g., through face recognition), and recording attendance details in the database.

3. **Cloud Deployment:** Host your Python script and the associated database in a cloud environment. Services like AWS, Google Cloud, or Azure can provide the infrastructure needed to deploy and run your cloud-based attendance system.
4. **Security Measures:** Implement security measures to protect sensitive employee data. Utilize secure connections (HTTPS) when communicating with the database and employ encryption for data at rest.
5. **Scalability:** Design your system to be scalable, considering the potential growth in the number of employees and attendance records. Cloud services typically offer scalability features that allow you to adapt to changing demands.
6. **Monitoring and Logging:** Implement monitoring and logging mechanisms to track system performance, errors, and user activities. Cloud platforms often provide tools for monitoring and logging that can be integrated into your system.

- **datetime:**In a cloud-based employee attendance system, the utilization of datetime plays a crucial role in accurately recording and managing attendance data. The datetime functionality is instrumental in capturing and storing the precise date and time when an employee registers their attendance. Here's how datetime is typically incorporated into such a system:

  1. **Timestamping Attendance Records:** When an employee registers attendance, the system utilizes the datetime module to generate a timestamp that includes both the date and time of the registration. This timestamp is then associated with the employee's record in the database. The accuracy of this timestamp is vital for tracking attendance trends and ensuring the reliability of the system.
  2. **Date-Based Operations:** The system often performs date-based operations using datetime functionalities. For example, it may retrieve attendance records for a specific date range or calculate the total hours worked by an employee within a given period. These operations rely on datetime objects to manipulate and compare dates effectively.
  3. **Real-Time Attendance Tracking:** In a cloud-based environment, where multiple employees may be registering attendance simultaneously, datetime helps in maintaining real-time accuracy. The system captures the current date and time when attendance is registered, allowing for immediate and precise updates to the attendance records.
  4. **Handling Time Zones:** Cloud-based systems are likely to cater to a diverse workforce spread across different geographic locations. The datetime module facilitates the handling of time zones, ensuring that attendance records accurately reflect the local time of each employee. This is crucial for maintaining consistency and fairness in attendance tracking.

5. **Auditing and Compliance:** The datetime information associated with each attendance record contributes to the system's auditing and compliance capabilities. It provides a detailed chronological history of attendance events, which can be crucial for regulatory compliance, audit trails, and resolving any discrepancies that may arise.
6. **Integration with Cloud Services:** Cloud platforms often offer services for managing datetime information efficiently. These platforms may provide tools for synchronizing time across distributed systems, handling daylight saving time changes, and ensuring the accuracy of timestamps in a cloud-based environment.

In summary, datetime operations are fundamental to the accurate and reliable functioning of a cloud-based employee attendance system. They contribute to real-time tracking, date-based analytics, and compliance, ensuring that the system operates seamlessly and provides trustworthy attendance data.

- **speech_recognition** The speech_recognition module in Python is a versatile and powerful tool for integrating speech recognition capabilities into applications. It provides a straightforward interface for converting spoken language into text, making it useful in a variety of applications, including voice-controlled systems, virtual assistants, and accessibility features. Here is an overview of the key aspects and functionalities of the speech_recognition module:

  1. **Audio Input Handling:** The module supports various audio sources, allowing developers to capture audio input from different devices, such as microphones or audio files. This flexibility enables the integration of speech recognition into diverse applications.
  2. **Compatibility with Multiple Recognition Engines:** speech_recognition is designed to work seamlessly with various automatic speech recognition (ASR) engines, including Google Web Speech API, Sphinx, Wit.ai, and others. This allows developers to choose the engine that best suits their application requirements.
  3. **Easy Integration with Cloud-Based Services:** Cloud-based ASR services often provide enhanced accuracy and support for multiple languages. The module facilitates integration with popular cloud-based speech recognition APIs, enabling developers to harness the capabilities of services like Google Cloud Speech-to-Text.
  4. **Support for Multiple Languages:** Developers can leverage the module's support for multiple languages, making it suitable for applications catering to diverse linguistic audiences. This multi-lingual capability enhances the accessibility and inclusivity of applications.

5. **Dynamic Adjustments and Configuration:** The module allows developers to dynamically adjust recognition parameters, such as energy threshold and dynamic energy adjustment, to optimize performance in different environments. This adaptability is crucial for achieving reliable recognition results.

6. **Integration with Language Understanding Services:** In addition to converting speech to text, developers can integrate the module with natural language understanding services. This extends functionality beyond simple transcription to include comprehension and interpretation of spoken commands.

7. **Real-Time and Offline Processing:** Whether operating in real-time or processing pre-recorded audio, the speech_recognition module accommodates both scenarios. This flexibility makes it suitable for applications with varying latency requirements.

8. **Error Handling and Exception Reporting:** The module includes features for handling errors and reporting exceptions. Developers can implement robust error-handling mechanisms to ensure the graceful degradation of the application in case of recognition challenges.

9. **Extensibility and Community Support:** Being an open-source module,speech_recognition benefits from a supportive community and regular updates. This extensibility encourages collaboration and ensures that the module stays aligned with evolving industry standards.

10. **Cross-Platform Compatibility:** The module is designed to be compatible with different operating systems, including Windows, macOS, and Linux. This cross-platform support enhances its versatility and applicability across a broad range of environments.

In conclusion, the speech_recognition module provides a user-friendly and comprehensive solution for integrating speech recognition capabilities into Python applications. Its versatility, compatibility with multiple recognition engines, and support for various languages make it a valuable tool for developers seeking to enhance the interactivity and accessibility of their applications through speech-based interfaces.

- **pyttsx3:** The pyttsx3 library in Python serves as a text-to-speech (TTS) conversion engine, allowing developers to incorporate speech synthesis into their applications. This library is particularly useful for creating applications that require spoken output, such as voice assistants, accessibility features, and interactive systems. Here is an overview of the key features and functionalities provided by pyttsx3:

1. **Cross-Platform Text-to-Speech:** pyttsx3 is designed to work across different operating systems, including Windows, macOS, and Linux. This cross-platform

compatibility makes it a versatile choice for developers targeting diverse environments.

2. **Straightforward Interface:** The library offers a simple and easy-to-use interface for converting text into spoken words. Developers can initiate text-to-speech conversions with minimal code, making it accessible for both beginners and experienced programmers.
3. **Adjustable Speech Rate and Volume:** pyttsx3 allows developers to control the speech rate (speed) and volume of the synthesized voice. This flexibility enables fine-tuning to match the preferences and requirements of the application or end user.
4. **Support for Multiple TTS Engines:** The library supports various text-to-speech engines, providing developers with options to choose the engine that best suits their needs. This adaptability allows for improved voice quality and language support.
5. **Asynchronous Speech Synthesis:** Developers can choose to run the text-to-speech conversion asynchronously, preventing the application from being blocked during speech synthesis. This is particularly useful for creating responsive and interactive applications.
6. **Event Handling and Callbacks:** pyttsx3 supports event handling, allowing developers to set up callbacks for specific events during the speech synthesis process. This includes events like the start and completion of speech, providing fine-grained control over the user experience.

   Language and Voice Selection: The library supports the selection of different voices and languages, enabling developers to tailor the spoken output to the desired linguistic and stylistic characteristics.
7. **Rich Documentation and Community Support:** pyttsx3 comes with comprehensive documentation, making it easier for developers to understand and implement text-to-speech functionality. Additionally, being an open-source library, it benefits from community contributions and updates.
8. **Customizable Speech Properties:** Developers can customize various speech properties, such as pitch and rate, to achieve the desired intonation and pacing in the synthesized speech. This level of customization enhances the naturalness of the spoken output.
9. **Compatibility with Python 3:** pyttsx3 is compatible with Python 3, ensuring that developers can leverage the latest features of the language while incorporating text-to-speech capabilities into their applications.

In summary, pyttsx3 is a versatile and user-friendly library that empowers developers to integrate text-to-speech functionality seamlessly into their Python applications. Its flexibility, cross-platform support, and rich set of features make it a valuable tool for creating applications with spoken output, contributing to improved accessibility and user interaction.

- **requests** The requests library in Python is a popular and versatile HTTP library that simplifies the process of sending HTTP requests and handling responses. Developed to be user-friendly and efficient, requests is widely used for various web-related tasks, including making API calls, fetching web pages, and interacting with web services. Here are key features and functionalities of the requests library:

  1. **Simplified HTTP Requests:** requests abstracts the complexities of HTTP, providing a clean and straightforward API for sending HTTP requests. Developers can perform common HTTP methods such as GET, POST, PUT, DELETE, etc., with minimal code.
  2. **User-Friendly API:** The library is designed with simplicity in mind, making it accessible to both beginners and experienced developers. Its intuitive API allows users to focus on their specific tasks without the need for intricate HTTP protocol details.
  3. **Support for HTTP Verbs and Methods:** requests supports a variety of HTTP methods, including GET for retrieving data, POST for submitting data, PUT for updating resources, and DELETE for removing resources. This versatility accommodates a wide range of web-related scenarios.
  4. **URL Parameter Handling:** The library enables developers to include parameters in the URL easily, facilitating the customization of requests based on specific requirements. This is especially useful when interacting with APIs that accept query parameters.
  5. **Request Headers and Cookies:** Developers can set custom headers and handle cookies within requests. This feature is valuable for scenarios where authentication tokens, custom headers, or session cookies need to be included in the HTTP requests.
  6. **Response Handling:** requests provides straightforward methods for handling HTTP responses, including accessing response content, status codes, headers, and other relevant information. This allows developers to extract and process data from the server's response efficiently.
  7. **Session Management:** The library supports sessions, allowing developers to persist certain parameters across multiple requests within the same session. This is particularly beneficial for scenarios that require maintaining state, such as handling user authentication.

8. **File Uploads and Downloads:** requests simplifies the process of uploading files by providing easy-to-use methods for including files in multipart/form-data requests. Similarly, it facilitates downloading files from the web effortlessly.
9. **Timeouts and Retries:** The library allows developers to set timeouts for requests, preventing indefinite waiting for a response. Additionally, it supports automatic retries for failed requests, contributing to enhanced reliability in scenarios with intermittent connectivity issues.
10. **SSL/TLS Certificate Verification:** requests supports SSL/TLS certificate verification by default, ensuring secure communication with HTTPS endpoints. Developers can also customize certificate handling based on specific security requirements.
11. **Proxy Support:** The library seamlessly integrates with proxies, enabling developers to route requests through proxy servers. This is useful in situations where requests need to be made from behind a firewall or for anonymizing web traffic.
12. **Community Support and Documentation:** requests benefits from an active community of developers and comprehensive documentation. This makes it easy for users to find support, examples, and guidance for using the library effectively.

In summary, the requests library in Python provides a powerful and user-friendly solution for handling HTTP requests and responses. Its simplicity, flexibility, and extensive features make it a go-to choice for developers involved in web development, API integration, and data retrieval tasks.

Chapter 4

Design Analysis

4.1 Architecture of Advance Attendance System

The architecture of an advanced attendance system encompasses a sophisticated integration of hardware and software components designed to streamline and enhance the attendance tracking process. Leveraging cutting-edge technologies, such a system typically comprises biometric recognition devices, cloud-based platforms, and intelligent algorithms.

At its core, biometric recognition serves as the primary means of identity verification. Advanced attendance systems often employ technologies such as fingerprint scanning, facial recognition, or iris scanning to uniquely identify individuals with a high level of accuracy. This ensures the reliability and security of the attendance data.

The hardware components, including biometric devices, are seamlessly integrated with a cloud-based system. This cloud infrastructure facilitates real-time data storage, accessibility, and analysis. Cloud-based platforms enable administrators to manage attendance records, update employee information, and generate reports from anywhere with an internet connection.

Intelligent algorithms play a crucial role in face recognition and matching processes. These algorithms are designed to adapt to variations in lighting conditions, facial expressions, and other environmental factors, ensuring robust performance in diverse scenarios. Machine learning techniques may also be incorporated to continuously improve recognition accuracy over time.

The architecture also involves a user-friendly interface for both administrators and employees. Employees can easily register attendance using biometric devices, while administrators have access to a comprehensive dashboard for monitoring attendance trends, managing employee information, and generating insights.

Security is a paramount consideration in the architecture. Encryption protocols are implemented to secure data transmission between biometric devices and the cloud. Additionally, stringent access controls are put in place to protect sensitive attendance data from unauthorized access.
In summary, the architecture of an advanced attendance system is a sophisticated blend of biometric recognition technologies, cloud-based infrastructure, intelligent algorithms, and user-friendly interfaces. This integrated approach aims to deliver accuracy, efficiency, and security in attendance tracking for modern organizations.

4.2 System Architecture

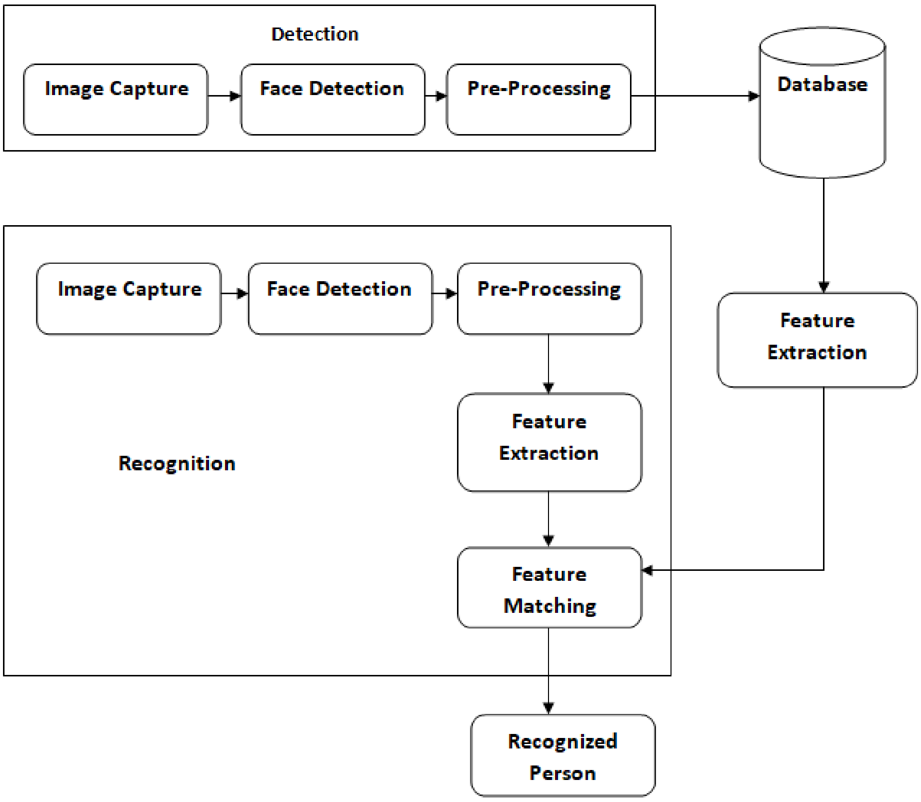


Figure 4.1: **System Architecture**

4.3 Data Tables

4.3.1 Role Table:

This table are designed for identify the all role or designation are available on this organization. table are as follows:

Sl no	**Fields Name**	**Type**	**Size**	**Key**
1	Role	varchar	50	Primary
2	User	varchar	100	–
3	doe	timestamp	–	–

Table 4.1: Roledb Table

4.3.2 Location Table:

This table is designed for which area the organization office is available to identifying this. Table structure are as follows:

Sl no	**Fields Name**	**Type**	**Size**	**Key**
1	Outlet_name	varchar	50	Primary
2	Outlet_Address	varchar	50	–
3	User	varchar	100	–
4	doe	timestamp	–	–

Table 4.2: Locationdb Table

4.3.3 Staff Table:

This table are designed for registration the staff with his/her image . In future this image required to model and used for face reconnigation. The table structure are as follows:

Sl no	Fields Name	Type	Size	Key
1	empid	varchar	50	Primary
2	Name	varchar	50	–
3	Ph	varchar	50	–
4	address	varchar	50	–
5	gender	varchar	50	–
6	email	varchar	50	–
7	role	varchar	50	–
8	Location	varchar	50	–
9	qualification	varchar	50	–
10	image_path	varchar	50	–
11	User	varchar	100	–
12	doe	timestamp	–	–

Table 4.3: staffdb Table

4.3.4 Attendance Table:

This table is designed for to take the attendance all the employes in which time he/she login or logout . Table structure are as follows:

Sl no	Fields Name	Type	Size	Key
1	empid	varchar	50	Primary
2	name	varchar	50	–
3	logindate	date	–	–
4	logintime	varchar	100	–
5	logouttime	varchar	100	–
3	User	varchar	100	–
4	doe	timestamp	–	–

Table 4.4: Attendance Table

4.3.5 Login Table:

This table is designed for to login admin or employee. Table structure are as follows:

Sl no	**Fields Name**	**Type**	**Size**	**Key**
1	user	varchar	50	Primary
2	pass	varchar	50	–
3	type	varchar	100	–

Table 4.5: Login Table

4.4 ER Diagram

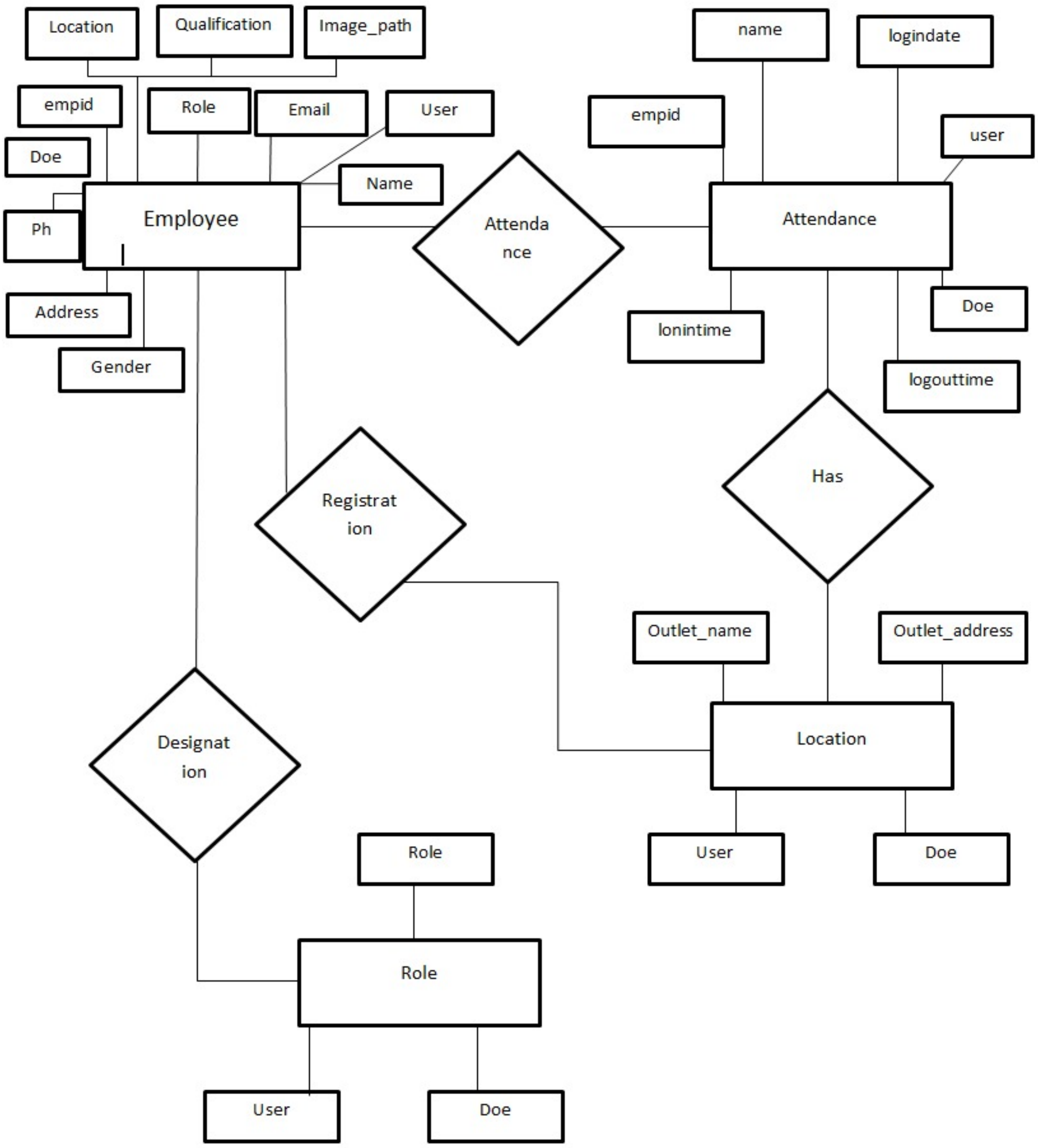


Figure 4.2: **ER Diagram for Advance Attendence System**

4.5 Data Flow Diagram

A Data Flow Diagram (DFD) for an Advance Attendance System illustrates the flow of data within the system components. Here's a simplified representation of the DFD for an Attendance system:

4.5.1 0 Level DFD

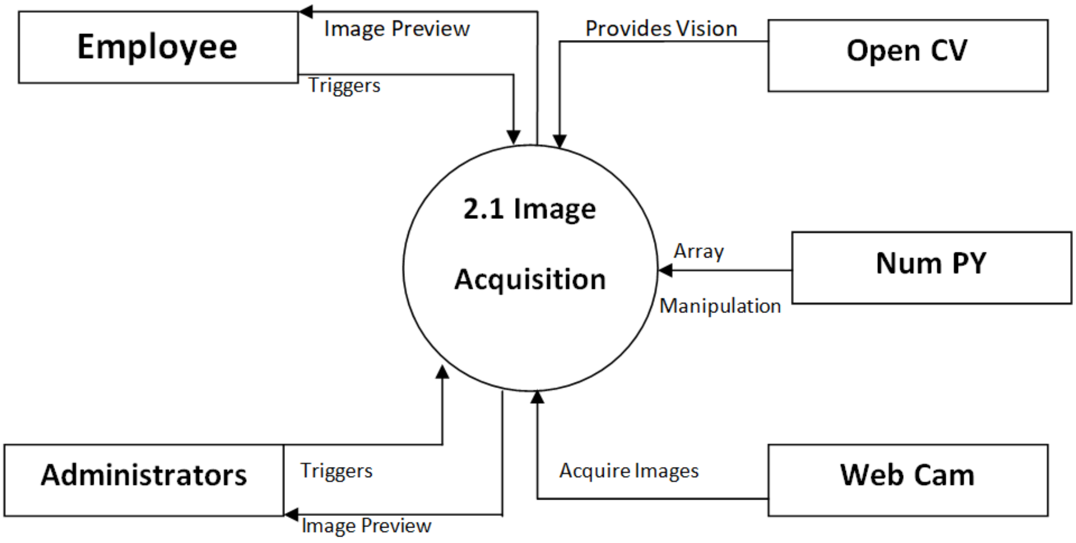


Figure 4.3: **0 Level DFD**

4.5.2 1 Level DFD

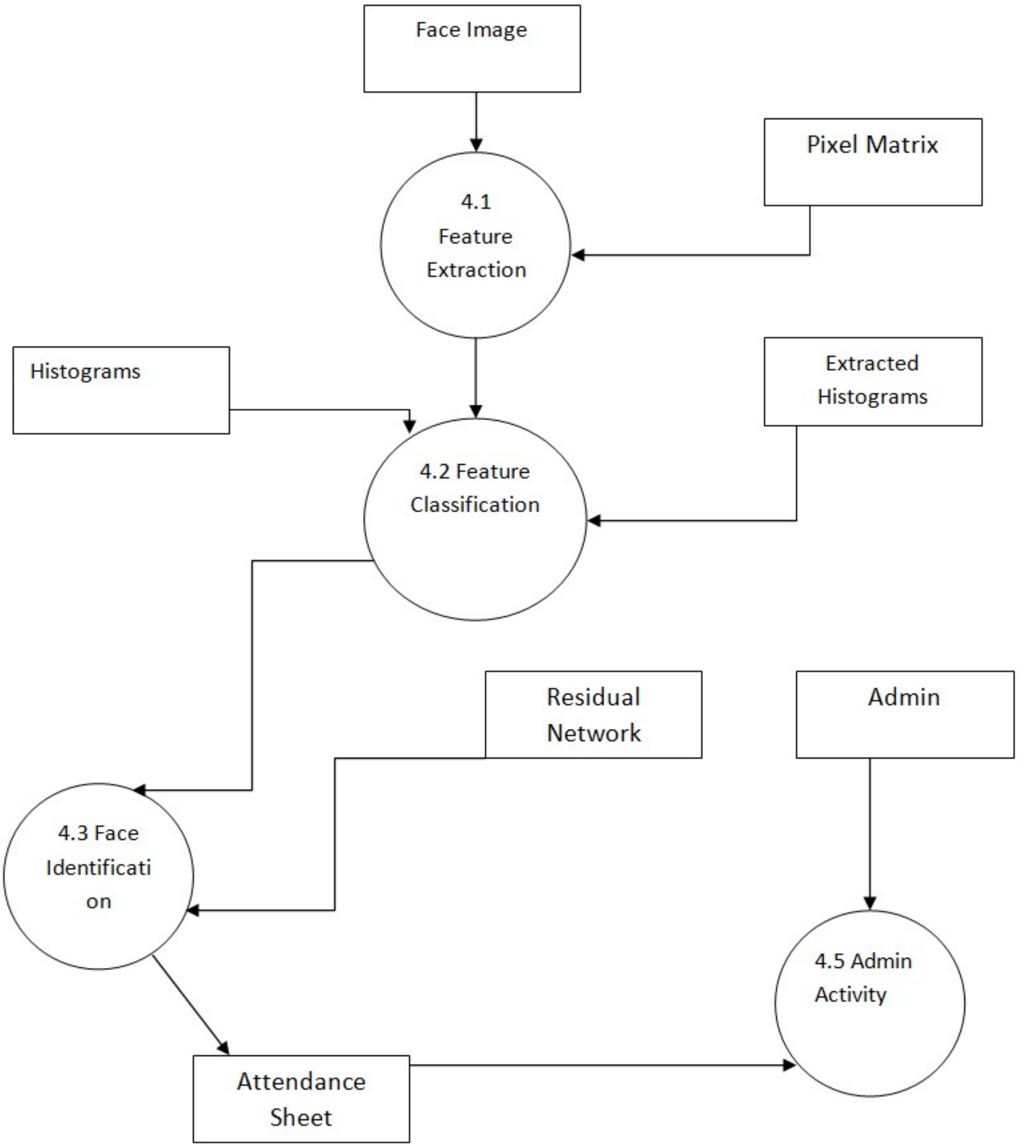


Figure 4.4: **1 Level DFD**

4.5.3 2 Level DFD

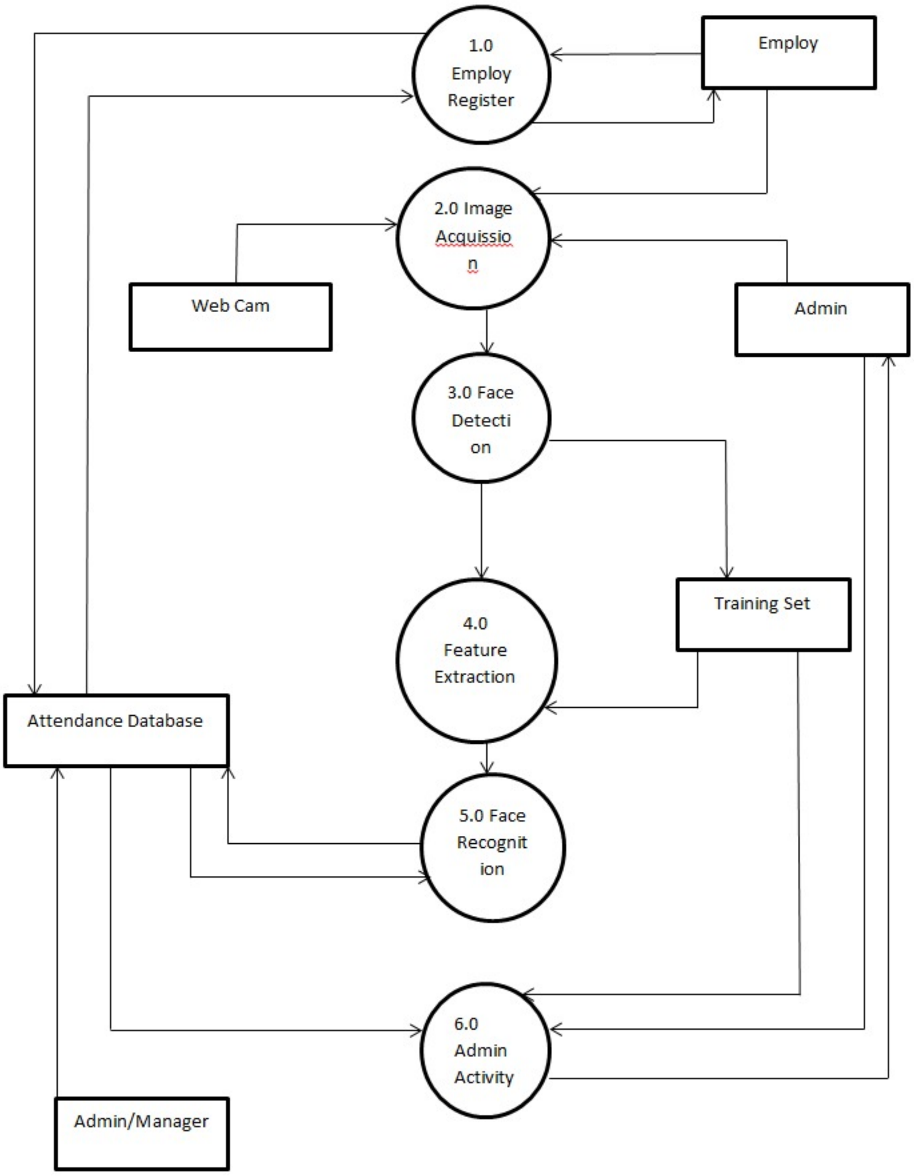


Figure 4.5: **2 Level DFD**

Chapter 5

Functioning of the System

The employee attendance system, utilizing face recognition technology and cloud-based infrastructure, streamlines the attendance tracking process. Employees register by providing essential details and a facial image, which is securely stored in the cloud. Upon arrival, employees input their unique registration number, prompting the system to initiate face recognition using the built-in camera. Successful matches result in the automatic marking of attendance, with timestamps recorded and updated in the cloud-based system. Non-matches trigger appropriate feedback, ensuring accurate attendance records. The system's integration with HR and payroll systems, coupled with real-time updates, enhances overall workforce management. With its efficient and secure approach, this advanced attendance system optimizes the tracking of employee presence.

5.1 Key Features and Functionalities

The functioning of an employee attendance system using face recognition typically involves several steps:

5.1.1 User Registration and Authentication

- Employees are registered in the system by providing essential details like their name, employee ID, and role.
- A facial image of the employee is captured and stored in the system along with the other details.

5.1.2 Cloud-Based Storage:

- Employee data, including facial images, is stored securely in a cloud-based system.
- This ensures accessibility and centralizes the attendance management process.

5.1.3 Updating Role and Location

- Employees may have the option to update their facial image for recognition purposes.
- This helps to keep the facial database up-to-date.

5.1.4 Registration Number Input:

- When employees arrive at work, they input their registration number into the system. This could be a unique identifier assigned to each employee.

5.1.5 Face Recognition

- The system opens the camera for face recognition when an employee inputs their registration number.
- The captured facial image is compared to the stored facial images in the database.

5.1.6 Attendance Marking

- If a match is found, the system marks the employee's attendance with a timestamp.
- The attendance data is updated in the cloud-based system.

5.1.7 Non-Match Handling

- If no match is found, the system may prompt the employee to retry or may indicate that attendance was not successfully recorded.

5.1.8 Attendance Record Management

- The system maintains a record of attendance, including dates, times, and employee details.
- This information is useful for generating reports and tracking attendance trends.

5.1.9 Security Measures

- Security measures are implemented to prevent unauthorized access or attempts to manipulate the system.

5.1.10 Integration with Other Systems

- The attendance system may be integrated with other HR or payroll systems for comprehensive workforce management.

5.1.11 Real-Time Updates

- The system provides real-time updates on attendance, allowing administrators to monitor employee presence instantly.

By combining cloud-based storage, facial recognition technology, and a user-friendly interface, this type of attendance system offers an efficient and secure way to manage employee attendance.

5.2 Scalability and Integration

The scalability and integration of the employee attendance system are pivotal aspects contributing to its effectiveness. Designed with scalability in mind, the system accommodates organizational growth seamlessly. Whether a company expands its workforce or integrates additional features, the architecture allows for effortless scalability, ensuring the system remains robust and responsive. Furthermore, the system is designed for seamless integration with other organizational modules, including HR, payroll, and employee management systems. This integration enhances overall efficiency, as attendance data seamlessly flows into broader organizational processes. The system's adaptability to evolving organizational needs, coupled with its smooth integration capabilities, positions it as a versatile and future-ready solution for businesses of varying sizes and structures.

5.3 Advance Employee Attendance System Benefits

The following benefits are available for using Advance Employee Attendance system

5.3.1 Automation

- Reduces administrative workload by automating attendance tracking processes.
- Minimizes errors associated with manual data entry.

5.3.2 Accuracy in Attendance Tracking:

- Ensures precise recording of employee attendance, reducing discrepancies.
- Eliminates the potential for time fraud or manipulation.

5.3.3 Payroll Accuracy

- Enhances payroll accuracy by providing reliable attendance data.
- Streamlines payroll processing and minimizes errors.

5.3.4 Compliance

- Facilitates compliance with labor regulations and work hour requirements.
- Helps organizations adhere to legal standards related to attendance.

5.3.5 Real-time Monitoring

- Provides real-time insights into workforce attendance.

- Enables quick decision-making based on up-to-date attendance data.

5.3.6 Security Enhancement

- Integrates biometric or facial recognition technology to enhance security.
- Prevents proxy attendance and ensures a secure work environment.

5.3.7 Transparency

- Promotes transparency in attendance management.
- Reduces the likelihood of fraudulent practices.

5.3.8 Efficient Response

- Facilitates swift response to attendance-related issues.
- Allows for quick resolution of discrepancies or concerns.

5.3.9 Productivity Improvement

- Streamlines attendance processes, allowing employees to focus on core tasks.
- Enhances overall workplace productivity.

5.3.10 Improved Work Environment

- Fosters a transparent and accountable work environment.
- Enhances employee satisfaction and trust in attendance management.

These benefits collectively contribute to creating a more efficient, secure, and compliant workplace.

5.3.11 Environmental Sustainability

Environmental sustainability in the context of an employee attendance system refers to the implementation of practices and technologies that minimize the system's impact on the environment throughout its lifecycle. The adoption of eco-friendly measures aligns with the broader goal of corporate responsibility and reducing the ecological footprint. Here are key aspects:

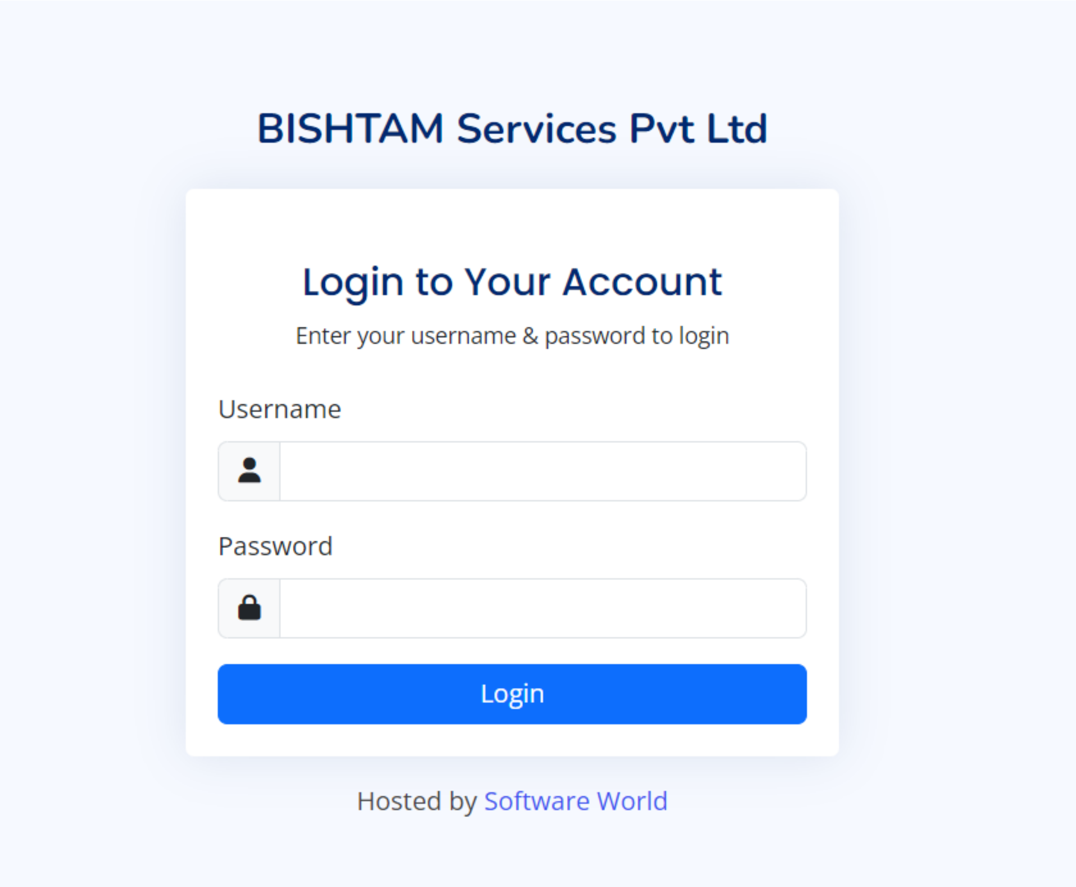
1. **Paperless Operations:** The system minimizes or eliminates the need for paper-based attendance tracking, reducing paper consumption and waste. Digital processes, such as biometric or facial recognition, contribute to a paperless environment.
2. **Energy Efficiency:** The system, including hardware and servers, is designed to be energy-efficient. This helps in conserving energy resources and lowering the overall carbon footprint. Employing energy-efficient components and practices reflects a commitment to sustainability.
3. **Remote Work Support:** By facilitating remote work options, the attendance system contributes to reduced commuting and associated carbon emissions. This aligns with the promotion of flexible work arrangements and the reduction of the organization's overall environmental impact.
4. **Minimal Hardware Footprint:** Efforts are made to ensure that the hardware components of the attendance system are sourced responsibly and are designed for longevity. Minimizing electronic waste through proper disposal and recycling practices is prioritized.
5. **Cloud-Based Solutions:** Utilizing cloud-based attendance systems reduces the need for extensive on-premises infrastructure. Cloud solutions often optimize resource usage, leading to more efficient data storage and processing.
6. **Regular System Audits:** Regular assessments and audits of the attendance system are conducted to identify areas for improvement in environmental sustainability. This includes evaluating energy usage, server efficiency, and waste reduction strategies.
7. **Employee Awareness Programs:** Organizations may implement awareness programs to educate employees on the environmental benefits of sustainable practices, such as using digital methods for attendance. Encouraging responsible behavior contributes to a culture of sustainability.
8. **Eco-Friendly Technologies:** Exploration and adoption of eco-friendly technologies, such as renewable energy sources for powering the attendance system infrastructure, contribute to a greener and more sustainable operation.

By integrating environmental sustainability into the design, implementation, and ongoing management of the employee attendance system, organizations demonstrate their commitment to reducing their environmental impact and contributing to a more sustainable future.

Chapter 6

EXPERIMENTAL RESULTS

6.1 Admin Login Section



BISHTAM Services Pvt Ltd

Login to Your Account

Enter your username & password to login

Username

Password

Login

Hosted by Software World

Figure 6.1: **Login Screen**

6.2 Admin Dashboard

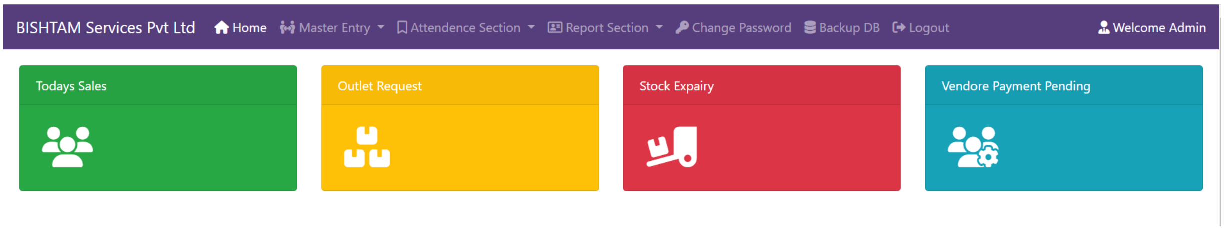
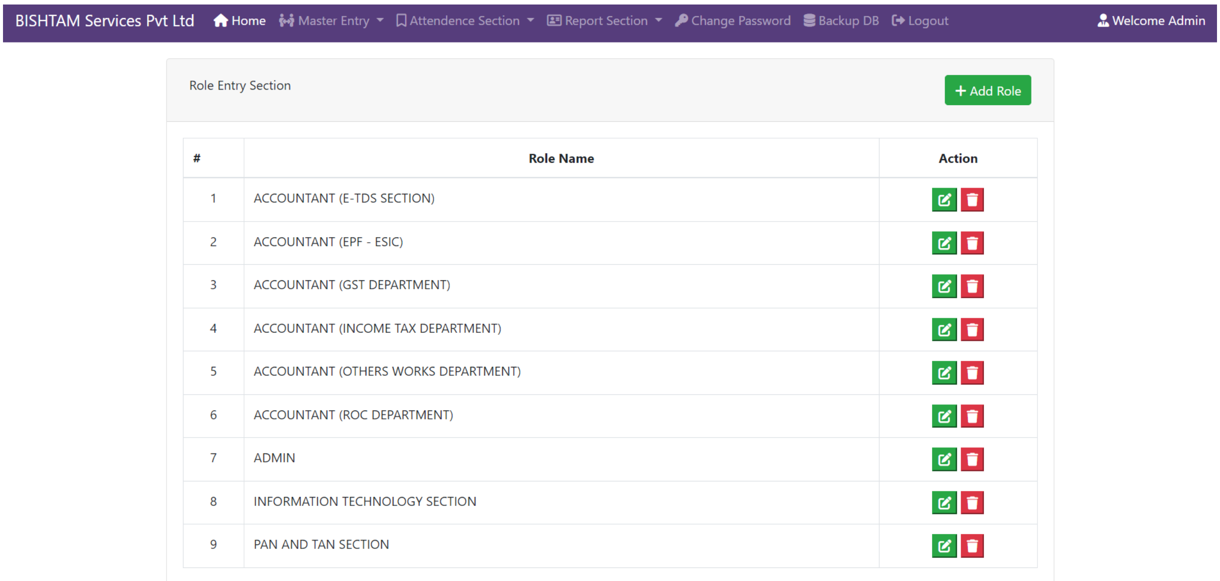


Figure 6.2: **Admin Dashboard**

6.3 Role Entry Section

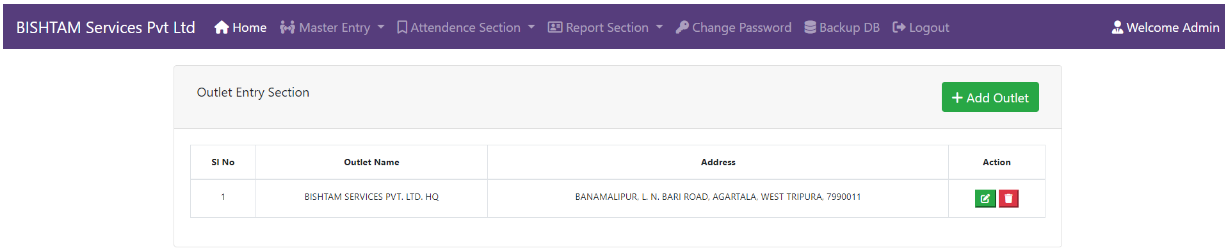


Role Entry Section

#	Role Name	Action
1	ACCOUNTANT (E-TDS SECTION)	✎ 🗑
2	ACCOUNTANT (EPF - ESIC)	✎ 🗑
3	ACCOUNTANT (GST DEPARTMENT)	✎ 🗑
4	ACCOUNTANT (INCOME TAX DEPARTMENT)	✎ 🗑
5	ACCOUNTANT (OTHERS WORKS DEPARTMENT)	✎ 🗑
6	ACCOUNTANT (ROC DEPARTMENT)	✎ 🗑
7	ADMIN	✎ 🗑
8	INFORMATION TECHNOLOGY SECTION	✎ 🗑
9	PAN AND TAN SECTION	✎ 🗑

Figure 6.3: **Role Entry Section**

6.4 Branch Section

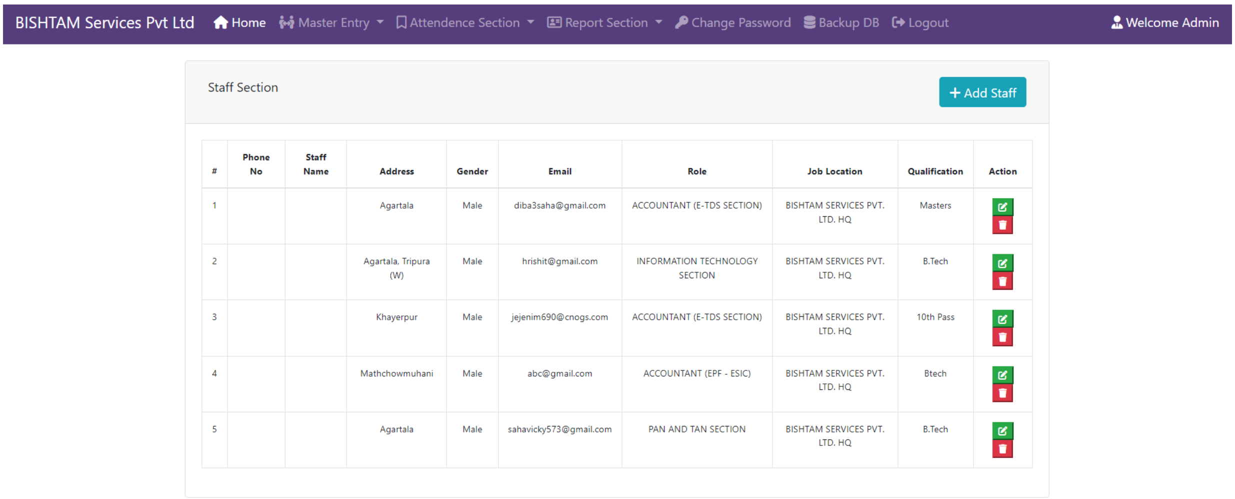


Outlet Entry Section

Sl No	Outlet Name	Address	Action
1	BISHTAM SERVICES PVT. LTD. HQ	BANAMALIPUR, L. N. BARI ROAD, AGARTALA, WEST TRIPURA, 7990011	

Figure 6.4: **Branch Section**

6.5 Staff Section

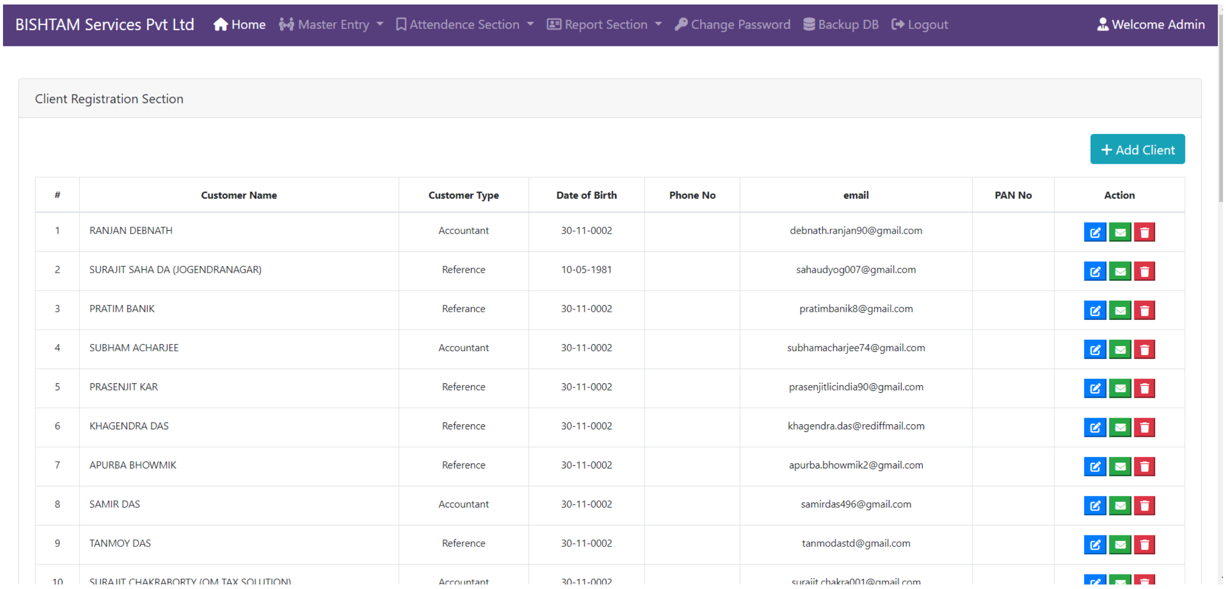


Staff Section

#	Phone No	Staff Name	Address	Gender	Email	Role	Job Location	Qualification	Action
1			Agartala	Male	diba3saha@gmail.com	ACCOUNTANT (E-TDS SECTION)	BISHTAM SERVICES PVT. LTD. HQ	Masters	✎ 🗑
2			Agartala, Tripura (W)	Male	hrishit@gmail.com	INFORMATION TECHNOLOGY SECTION	BISHTAM SERVICES PVT. LTD. HQ	B.Tech	✎ 🗑
3			Khayerpur	Male	jejenim690@cnogs.com	ACCOUNTANT (E-TDS SECTION)	BISHTAM SERVICES PVT. LTD. HQ	10th Pass	✎ 🗑
4			Mathchowmuhani	Male	abc@gmail.com	ACCOUNTANT (EPF - ESIC)	BISHTAM SERVICES PVT. LTD. HQ	Btech	✎ 🗑
5			Agartala	Male	sahavicky573@gmail.com	PAN AND TAN SECTION	BISHTAM SERVICES PVT. LTD. HQ	B.Tech	✎ 🗑

Figure 6.5: **Staff Section**

6.6 Client Section

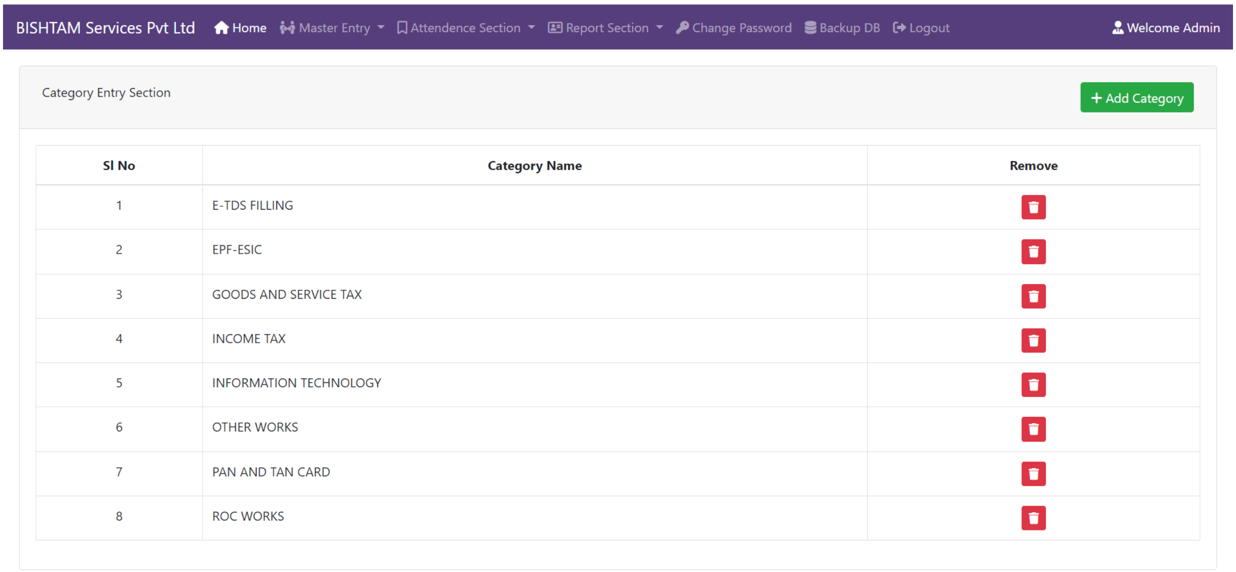


Client Registration Section

#	Customer Name	Customer Type	Date of Birth	Phone No	email	PAN No	Action
1	RANJAN DEBNATH	Accountant	30-11-0002		debnath.ranjan90@gmail.com		✎ ✉ 🗑
2	SURAJIT SAHA DA (JOGENDRANAGAR)	Reference	10-05-1981		sahaudyog007@gmail.com		✎ ✉ 🗑
3	PRATIM BANIK	Referance	30-11-0002		pratimbanik8@gmail.com		✎ ✉ 🗑
4	SUBHAM ACHARJEE	Accountant	30-11-0002		subhamacharjee74@gmail.com		✎ ✉ 🗑
5	PRASENJIT KAR	Reference	30-11-0002		prasenjitlicindia90@gmail.com		✎ ✉ 🗑
6	KHAGENDRA DAS	Reference	30-11-0002		khagendra.das@rediffmail.com		✎ ✉ 🗑
7	APURBA BHOWMIK	Reference	30-11-0002		apurba.bhowmik2@gmail.com		✎ ✉ 🗑
8	SAMIR DAS	Accountant	30-11-0002		samirdas496@gmail.com		✎ ✉ 🗑
9	TANMOY DAS	Reference	30-11-0002		tanmodastd@gmail.com		✎ ✉ 🗑
10	SURAJIT CHAKRABORTY (OM TAX SOLUTION)	Accountant	30-11-0002		surajit.chakra001@gmail.com		✎ ✉ 🗑

Figure 6.6: **Client Section**

6.7 Category Section

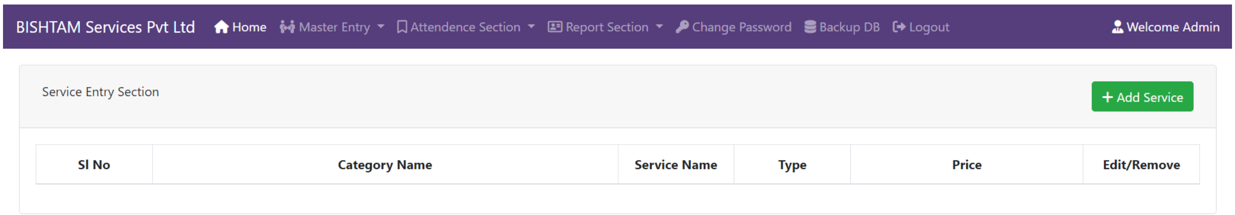


Category Entry Section

Sl No	Category Name	Remove
1	E-TDS FILLING	🗑
2	EPF-ESIC	🗑
3	GOODS AND SERVICE TAX	🗑
4	INCOME TAX	🗑
5	INFORMATION TECHNOLOGY	🗑
6	OTHER WORKS	🗑
7	PAN AND TAN CARD	🗑
8	ROC WORKS	🗑

Figure 6.7: **Category Section**

6.8 Client Section

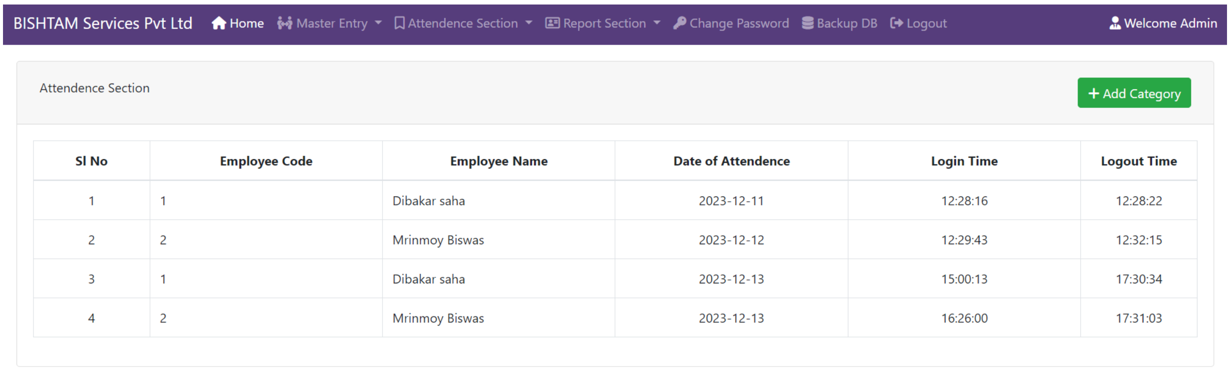


Service Entry Section

Sl No	Category Name	Service Name	Type	Price	Edit/Remove

Figure 6.8: **Service Section**

6.9 Attendance View Section

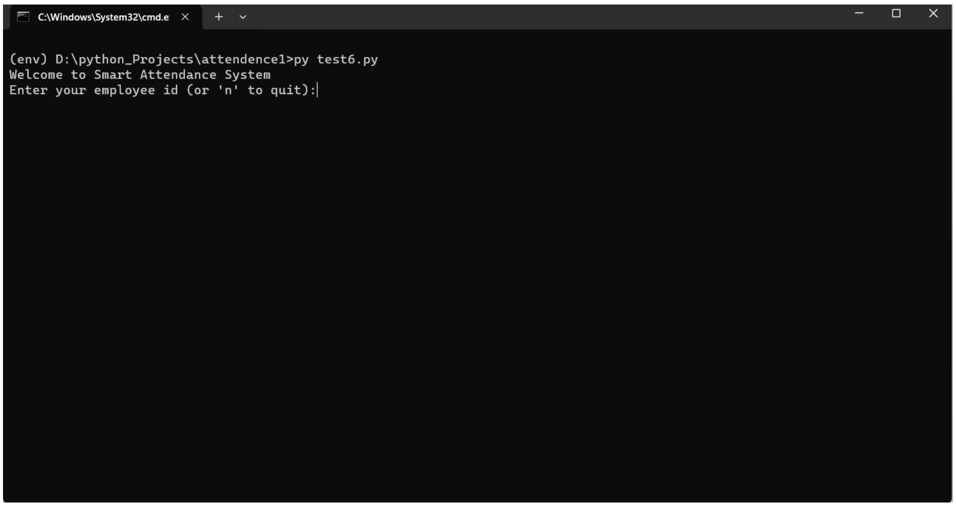


Attendence Section

Sl No	Employee Code	Employee Name	Date of Attendence	Login Time	Logout Time
1	1	Dibakar saha	2023-12-11	12:28:16	12:28:22
2	2	Mrinmoy Biswas	2023-12-12	12:29:43	12:32:15
3	1	Dibakar saha	2023-12-13	15:00:13	17:30:34
4	2	Mrinmoy Biswas	2023-12-13	16:26:00	17:31:03

Figure 6.9: **Attendance View Section**

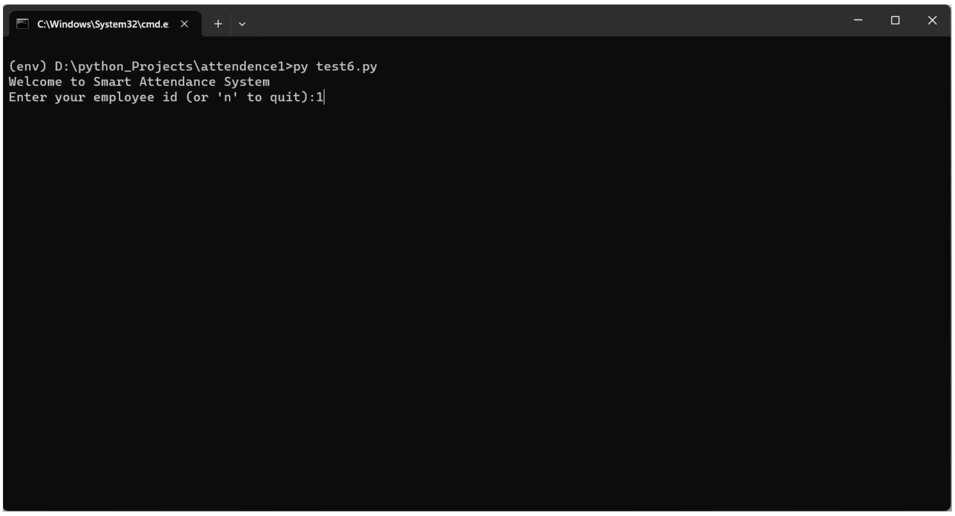
6.10 Attendance Local Portal



```
(env) D:\python_Projects\attendence1>py test6.py
Welcome to Smart Attendance System
Enter your employee id (or 'n' to quit):
```

Figure 6.10: **Attendance View Section**

6.11 Put His/Her Employeeid



```
(env) D:\python_Projects\attendence1>py test6.py
Welcome to Smart Attendance System
Enter your employee id (or 'n' to quit):1
```

Figure 6.11: **Put His/Her Employeeid**

6.12 Take Attendance with Image

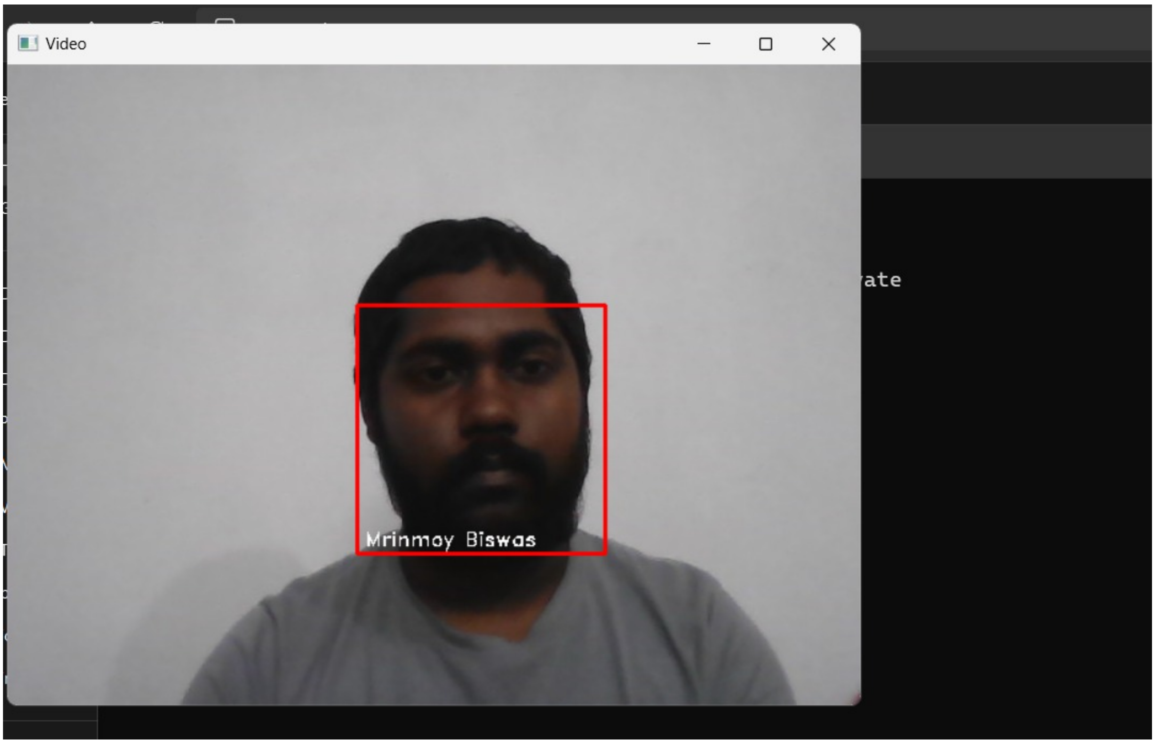
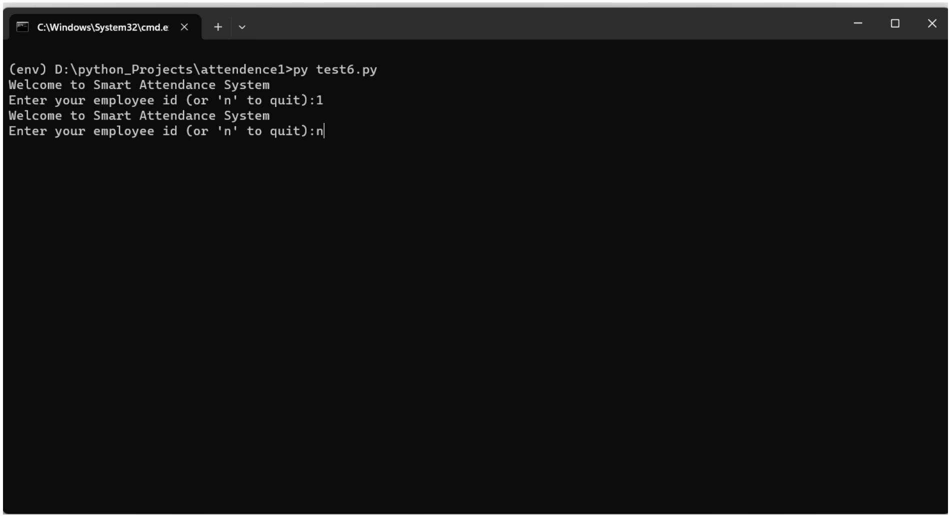


Figure 6.12: **Take Attendance with Image**

6.13 Close Local System



```
(env) D:\python_Projects\attendence1>py test6.py
Welcome to Smart Attendance System
Enter your employee id (or 'n' to quit):1
Welcome to Smart Attendance System
Enter your employee id (or 'n' to quit):n
```

Figure 6.13: **Take Attendance with Image**

6.14 Accuracy Testing

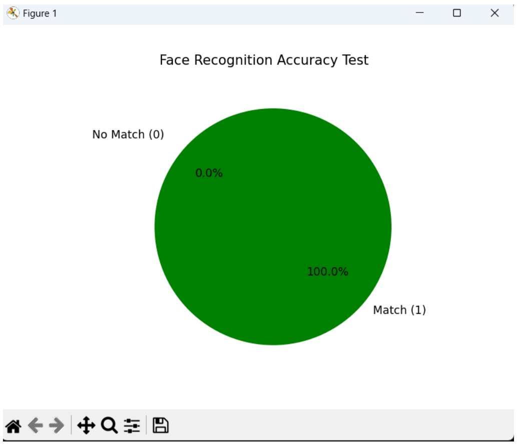


Figure 6.14: **Accuracy Testing**

Chapter 7

Conclusion and Future Direction of Work

7.1 Conclusion

In conclusion, the implementation of an advanced cloud-based employee attendance system brings about a transformative shift in the way organizations manage workforce tracking. Leveraging cutting-edge technologies, such as facial recognition and cloud computing, this system offers numerous advantages in terms of accuracy, efficiency, and overall operational effectiveness.

The utilization of face recognition algorithms ensures a secure and streamlined attendance tracking process. This not only mitigates concerns related to fraudulent practices but also enhances the overall precision of attendance records. The cloud-based architecture adds an extra layer of flexibility, enabling real-time data access from anywhere, facilitating remote work scenarios, and accommodating the dynamic nature of modern workplaces.

Furthermore, the integration of additional features, such as role and location updates, along with the centralization of attendance data in the cloud, contributes to a more comprehensive employee management system. The ease of registration, coupled with the seamless recognition process, enhances the overall user experience for both employees and administrators.

The scalability and integration capabilities of the system empower organizations to adapt to evolving needs, whether it be accommodating a growing workforce or integrating with other business systems. The use of biometric data and the cloud introduces a level of sophistication that goes beyond traditional attendance tracking methods.

While focusing on technological advancements, it is crucial not to overlook the environmental impact. Implementing eco-friendly practices, such as reducing paper usage, promoting remote work, and adopting energy-efficient solutions, aligns the attendance system with broader sustainability goals.

In essence, the advanced employee attendance system represents a pivotal step towards modernizing workforce management. Its multifaceted benefits, ranging from enhanced security and accuracy to improved operational efficiency and sustainability, position it as a valuable asset for organizations striving to stay ahead in today's dynamic business landscape.

7.2 Future Direction of work

The future direction of work in the field of advanced employee attendance systems holds exciting prospects for continued innovation and improvement. Several key areas are likely to shape the evolution of these systems:

1. **Integration with Emerging Technologies:** As technology continues to advance, integrating employee attendance systems with emerging technologies such as artificial intelligence (AI) and machine learning (ML) will likely enhance recognition accuracy and provide more robust data analytics.

2. Enhanced Security Measures: Future systems will likely focus on incorporating even more robust security measures to safeguard sensitive biometric data. This could include the exploration of blockchain technology to ensure the integrity and privacy of attendance records.

3. **User Experience Optimization:** Continuous efforts will be made to improve the user experience by refining the registration and recognition processes. User feedback and usability studies will play a crucial role in shaping interfaces that are intuitive and user-friendly.

4. **Adoption of Wearable Technology:** The integration of wearable devices for attendance tracking could become more prevalent. Wearables equipped with biometric sensors could offer a convenient and unobtrusive way to monitor attendance while providing additional health and well-being data.

5. **Predictive Analytics for Attendance Management:** Utilizing data analytics and predictive modeling, future systems might offer insights into attendance patterns, helping organizations anticipate attendance trends, plan resources more effectively, and identify potential issues proactively.

6. **Global Accessibility:** With an increasingly globalized workforce, future systems may emphasize global accessibility, accommodating different languages, cultural norms, and legal requirements related to data privacy and employee tracking.

7. **Continuous Software Updates:** To stay ahead of security threats and technological advancements, future systems will likely incorporate mechanisms for seamless and automatic software updates. This ensures that organizations always have access to the latest features and security patches.

8. **Environmental Sustainability:** The future of employee attendance systems may see a heightened focus on environmental sustainability. This could involve minimizing the carbon footprint associated with data centers, adopting energy-efficient hardware, and promoting eco-friendly practices.

9. **Regulatory Compliance:** Given the evolving landscape of data protection laws, future systems will need to stay abreast of regulatory changes and ensure strict compliance. This includes adhering to privacy regulations and industry-specific requirements.

10. **Customization and Scalability:** Recognizing the diverse needs of different organizations, future systems might prioritize customization and scalability. This allows businesses to tailor the attendance system to their specific requirements and seamlessly scale it as their workforce grows.

In essence, the future of advanced employee attendance systems lies in a dynamic interplay between technological innovation, user-centric design, and a commitment to security and compliance. Organizations that embrace these advancements will likely find themselves better equipped to navigate the complexities of modern workforce management.

7.2.1 Enhanced User Profile Features

Enhanced user profile features are crucial components of modern digital platforms, providing users with a personalized, interactive, and comprehensive experience. The following key features contribute to an enriched user profile: Profile Customization:

1. Profile Customization: Users can personalize their profiles with profile pictures, cover photos, and custom themes, allowing for self-expression and individuality.

2. Bio and About Me Section: A dedicated space for users to write a bio or "About Me" section, providing an opportunity to share personal interests, aspirations, and a brief introduction.

3. Contact Information: Users can include various contact details, such as email addresses, phone numbers, and social media links, facilitating easy communication and networking.

4. Professional Details: Users have the option to showcase their professional details, including job titles, company affiliations, and industry expertise, enhancing their credibility.

5. Education and Work History: Comprehensive sections for educational background and work history allow users to highlight their academic achievements and career progression.

6. Skills and Endorsements: Users can list their skills, and connections can endorse them, adding a layer of validation to their professional capabilities.

7. Projects and Portfolio Showcase: A dedicated area to display projects, portfolios, or work samples enables users to exhibit their skills and showcase their professional achievements.

8. Certifications and Achievements: Users can feature certifications, awards, and notable accomplishments, adding a layer of recognition and expertise to their profile.

9. Publications and Contributions: For users involved in research or content creation, a section to highlight publications, articles, or contributions further establishes their authority in their field.

10. Recommendations and Testimonials: Colleagues, clients, or collaborators can provide recommendations and testimonials, offering insights into the user's professional relationships and performance.

11. Activity Feed: An activity feed displays recent user interactions, such as posts, comments, or updates, fostering engagement and keeping the profile dynamic.

12. Privacy Settings: Robust privacy controls empower users to manage who can view specific elements of their profile, ensuring a balance between visibility and data protection.

13. Integration with Social Media: Seamless integration with social media platforms allows users to import content and updates, enhancing the overall visibility and connectivity of their profile.

14. Interactive Widgets: Widgets like calendars, task lists, or upcoming events contribute to a productive user experience directly from the profile interface.

15. Gamification Elements: Gamified elements such as badges, achievements, or completion milestones incentivize user engagement and contribute to a more interactive platform.

16. Responsive Design: User profiles are designed to be responsive, ensuring a consistent and visually appealing experience across various devices.

These enhanced user profile features collectively contribute to a more engaging, informative, and user-friendly platform, fostering meaningful connections and interactions within the digital community.

7.3 Integration with HR and Payroll Systems

The integration of the face recognition-based employee attendance system with HR and payroll systems presents a transformative approach to workforce management. By seamlessly connecting attendance data with HR records, the system streamlines payroll processing and employee management. This integration ensures accurate and timely compensation for employees based on their attendance records. HR professionals can leverage the system's real-time insights into employee attendance patterns, enabling them to make informed decisions regarding workforce planning and resource allocation. The automated synchronization reduces manual data entry, minimizing errors and enhancing overall efficiency. Furthermore, this integration facilitates compliance with labor regulations and company policies, as attendance data becomes an integral part of HR workflows. The cohesive connection between attendance, HR, and payroll systems creates a unified platform that fosters operational synergy, empowers decision-makers, and ultimately contributes to the overall productivity and success of the organization.

7.4 Future Scope

The future scope of an employee attendance system using face recognition technology encompasses several exciting possibilities and advancements. Here are some potential areas for future development and enhancement:

1. **Advanced Facial Recognition Algorithms:**Continued research and development in facial recognition algorithms can lead to improved accuracy, efficiency, and adaptability, even in challenging scenarios like low light or diverse facial expressions.

2. **Machine Learning Integration:** Integration with machine learning models can enhance the system's ability to adapt and learn from diverse datasets, improving its recognition capabilities over time.

3. **Real-Time Analytics:**Implementing real-time analytics can provide valuable insights into attendance patterns, employee behavior, and workplace dynamics, enabling better

decision-making for organizational management.

4. **Behavioral Analysis:** Incorporating behavioral analysis, such as gait recognition or emotion detection, can provide additional layers of information about employee well-being, engagement, and overall satisfaction.

5. **Multimodal Biometrics:** Exploring the integration of multiple biometric modalities, such as fingerprint or iris recognition, alongside facial recognition, can enhance overall security and accuracy.

6. **Edge Computing for Decentralization:** Leveraging edge computing technologies can decentralize processing, enabling faster and more efficient recognition directly on the devices, reducing latency and dependency on centralized servers.

7. **Blockchain for Security:** Implementing blockchain technology can enhance the security and integrity of attendance records, ensuring tamper-proof and transparent data storage.

8. **Integration with IoT Devices:** Connecting the attendance system with Internet of Things (IoT) devices, such as smart cameras or wearables, can provide a more comprehensive and interconnected approach to employee management.

9. **Mobile Application Enhancements:** Improving mobile applications for attendance tracking can offer more flexibility and convenience, allowing employees to mark attendance or receive notifications on their smartphones.

10. **Automated Compliance Management:** Developing features that automate compliance management, ensuring that the attendance system adheres to legal and privacy regulations, can be a key focus for future enhancements.

11. **Voice Recognition for Authentication:** Integrating voice recognition technology alongside facial recognition can provide an additional layer of authentication and convenience for users.

12. **Augmented Reality (AR) Integration:** Exploring AR applications for attendance tracking, such as virtual check-ins or AR-based employee directories, can add a futuristic and engaging dimension to the system.

13. **Global Accessibility and Localization:** Adapting the system to cater to global requirements and cultural differences, including language localization and compliance with diverse regulations, can broaden its applicability.

14. **Employee Feedback Mechanisms:** Implementing feedback mechanisms within the system can allow employees to provide input on the attendance process, fostering a user-centric approach to system development.

15. **Integration with HR and Payroll Systems:** Strengthening integration with HR and payroll systems can streamline administrative processes, ensuring seamless communication between attendance records and other HR functions.

16. **Continuous User Training and Awareness:** Providing ongoing training and awareness programs for users and administrators can help in maximizing the benefits of the system and ensuring its effective utilization.

By exploring these avenues, the future of face recognition-based employee attendance systems holds the potential for increased efficiency, accuracy, and user satisfaction, contributing to the overall advancement of workplace technology.

Chapter 8

Reference

1. Xin Geng, Zhi-Hua Zhou, & Smith-Miles, K. (2008). Individual Stable Space: An Approach to Face Recognition Under Uncontrolled Conditions. IEEE Transactions on Neural Networks

2. M. S. Akbar, P. Sarker, A. T. Mansoor, A. M. Al Ashray and J. Uddin, "Face Recognition and RFID Verified Attendance System," 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 2018, pp. 168-172, doi: 10.1109/iCCECOME.2018.8658705.

3. Okokpujie, Kennedy O., et al. "Design and implementation of a student attendance system using iris biometric recognition." 2017 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2017.

4. E. Winarno, W. Hadikurniawati, I. H. Al Amin and M. Sukur, "Anti-cheating presence system based on 3WPCA-dual vision face recognition," 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 2017, pp. 1-5, doi: 10.1109/EECSI.2017.8239115.

5. E. Winarno, W. Hadikurniawati, I. H. Al Amin and M. Sukur, "Anti-cheating presence system based on 3WPCA-dual vision face recognition," 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 2017, pp. 1-5, doi: 10.1109/EECSI.2017.8239115.

6. A. R. S. Siswanto, A. S. Nugroho and M. Galinium, "Implementation of face recognition algorithm for biometrics based time attendance system," 2014 International Conference on ICT For Smart Society (ICISS), Bandung, Indonesia, 2014, pp. 149-154, doi: 10.1109/ICTSS.2014.7013165.

7. Nandhini R, Duraimurugan N, S.P Chollalingam “ FaceRecognition Attendance System” in IJEAT in 2019. https://www.ijeat.org/wpcontent/uploads/papers/v8i3 S/C

8. Lukas, Samuel, et al. "Student attendance system in classroom using face recognition technique." 2016 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2016

Chapter 9

Codding

9.1 Python codding for Face Recognition

```
 import face_recognition
import cv2
import mysql.connector
from datetime import datetime
import speech_recognition as sr
import pyttsx3
import requests
import numpy as np

def speak(text):
    engine = pyttsx3.init()
    engine.setProperty('rate', 150)
    engine.say(text)
    engine.runAndWait()

# Function to encode the facial image
def encode_face(image_path):
    image = face_recognition.load_image_file(image_path)
    face_encodings = face_recognition.face_encodings(image)
    if len(face_encodings) > 0:
        return face_encodings[0]
    else:
        return None

def is_attendance_record_exists(db_cursor, cid, current_date)
    :
    # Check if attendance record for the given employee ID
        and current date exists
    sql = 'SELECT * FROM attendance WHERE cid=%s AND date=%s'
    db_cursor.execute(sql, (cid, current_date))
    attendance_record = db_cursor.fetchone()
    return attendance_record is not None

# Function to recognize, compare faces, and mark attendance
def recognize_faces_and_mark_attendance(video_capture,
    reference_image, cid):
    # Encode the reference face image
    #reference_face_encoding = encode_face(reference_image)
    reference_face_encoding = face_recognition.face_encodings
        (reference_image)[0]

    # Connect to the MySQL database
    db_connection = mysql.connector.connect(
        host='217.21.95.154',
        user='u333956788_image_root',
        password='Soft@2017',
        database='u333956788_imagedb'
    )
    db_cursor = db_connection.cursor()

    while True:
        ret, frame = video_capture.read()

        # Find face locations and encodings in the frame
        face_locations = face_recognition.face_locations(
            frame)
        face_encodings = face_recognition.face_encodings(
            frame, face_locations)
```

```
for face_encoding in face_encodings:
    # Compare the captured face with the reference
       face
    name = "Unknown"

    matches = face_recognition.compare_faces([
       reference_face_encoding], face_encoding)
    if matches[0]:
        sql = ’SELECT name FROM reg WHERE cid=%s’
        db_cursor.execute(sql, (cid,))
        records = db_cursor.fetchall()
        for row in records:
            name = row[0]

        # Store attendance data in the database
        current_date = datetime.now().strftime(’%Y-%m
           -%d’)
        current_time = datetime.now().strftime(’%H:%M
           :%S’)

        if not is_attendance_record_exists(db_cursor,
            cid, current_date):
            db_cursor.execute(’’’
                INSERT INTO attendance (cid, name,
                   date, time)
                VALUES (%s, %s, %s, %s)
            ’’’, (cid, name, current_date,
               current_time))
            db_connection.commit()
        else:
            db_cursor.execute(’’’
                UPDATE attendance
                SET logout_time = %s
                WHERE cid = %s AND date = %s
            ’’’, (current_time, cid, current_date))
            db_connection.commit()
            # Display the result on the frame
            top, right, bottom, left = face_locations[0]
            cv2.rectangle(frame, (left, top), (right, bottom)
                , (0, 0, 255), 2)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6),
               font, 0.5, (255, 255, 255), 1)

        # Display the frame with recognized faces
        cv2.imshow(’Video’, frame)

        # Press ’q’ to exit
        if cv2.waitKey(1) & 0xFF == ord(’q’):
            break

    # Release the camera and close the window
    video_capture.release()
    cv2.destroyAllWindows()
    db_connection.close()

def download_image_from_url(url):
    response = requests.get(url)
    if response.status_code == 200:
        image_array = np.asarray(bytearray(response.content),
            dtype=np.uint8)
        image = cv2.imdecode(image_array, cv2.IMREAD_COLOR)
        return image
    else:
        return None

def save_image_locally(url, file_path):
    with open(file_path, ’wb’) as f:
        response = requests.get(url)
        if response.status_code == 200:
            f.write(response.content)
            return True
        else:
            return False

if __name__ == "__main__":
    # Connect to the database and retrieve image paths from
       the table
    while True:
        db_connection = mysql.connector.connect(
            host='217.21.95.154',
            user='u333956788_image_root',
            password='Soft@2017',
            database='u333956788_imagedb'
        )
        db_cursor = db_connection.cursor()
        print("Welcome to Smart Attendance System")
        speak("Welcome to Smart Attendance System")
        speak("Enter your employee id please")
        x = input("Enter your employee id (or 'n' to quit):")

        if x.lower() == 'n':
            break

        sql = 'SELECT image_path FROM reg WHERE cid=%s'
        db_cursor.execute(sql, (x,))
        image_paths = [record[0] for record in db_cursor.
           fetchall()]
        db_connection.close()
        reference_image_url = image_paths[0]


        # Download and save the reference image locally
        reference_image_local_path = 'img/1.jpg'
        if not save_image_locally(reference_image_url,
           reference_image_local_path):
            print("Failed to download the image from the URL.
                Skipping employee.")
            continue
        else:
            save_image_locally(reference_image_url,
                reference_image_local_path)

        # Load the reference image from the local file path
        reference_image = cv2.imread(
            reference_image_local_path)



        # Initialize the camera
        video_capture = cv2.VideoCapture(0)  # Use the
            appropriate camera index if not the default one
        # Capture attendance for the current employee
        recognize_faces_and_mark_attendance(video_capture,
            reference_image, x)
        # Release the camera and close the window after
            capturing attendance
        video_capture.release()
        cv2.destroyAllWindows()
```

9.2 Checking the accuracy

```
    import face_recognition
import cv2
import mysql.connector
from datetime import datetime
import speech_recognition as sr
import pyttsx3
import requests
import numpy as np
from PIL import Image
from io import BytesIO
import matplotlib.pyplot as plt

def speak(text):
    engine = pyttsx3.init()
    engine.setProperty(’rate’, 150)  # Adjust speech rate (
       words per minute)
    engine.say(text)
    engine.runAndWait()

# Function to encode the facial image
def encode_face(image_path):
    image = face_recognition.load_image_file(image_path)
    face_encodings = face_recognition.face_encodings(image)
    if len(face_encodings) > 0:
        return face_encodings[0]
    else:
        return None

def is_attendance_record_exists(db_cursor, cid, current_date)
   :
    # Check if attendance record for the given employee ID
       and current date exists
    sql = ’SELECT * FROM attendance WHERE cid=%s AND date=%s’
    db_cursor.execute(sql, (cid, current_date))
    attendance_record = db_cursor.fetchone()
    return attendance_record is not None

# Function to recognize, compare faces, and mark attendance
def recognize_faces_and_mark_attendance(video_capture,
   reference_image, cid):
    # Encode the reference face image
    reference_face_encoding = face_recognition.face_encodings
       (reference_image)[0]

    # Connect to the MySQL database
    db_connection = mysql.connector.connect(
        host=’217.21.95.154’,
        user=’u333956788_image_root’,
        password=’Soft@2017’,
        database=’u333956788_imagedb’
    )
```

```
db_cursor = db_connection.cursor()

while True:
    ret, frame = video_capture.read()

    # Find face locations and encodings in the frame
    face_locations = face_recognition.face_locations(
       frame)
    face_encodings = face_recognition.face_encodings(
       frame, face_locations)

    for face_encoding in face_encodings:
        # Compare the captured face with the reference
           face
        name = "Unknown"

        matches = face_recognition.compare_faces([
           reference_face_encoding], face_encoding)
        if matches[0]:
            sql = ’SELECT name FROM reg WHERE cid=%s’
            db_cursor.execute(sql, (cid,))
            records = db_cursor.fetchall()
            for row in records:
                name = row[0]

            # Store attendance data in the database
            current_date = datetime.now().strftime(’%Y-%m
               -%d’)
            current_time = datetime.now().strftime(’%H:%M
               :%S’)

            if not is_attendance_record_exists(db_cursor,
                 cid, current_date):
                db_cursor.execute(’’’
                    INSERT INTO attendance (cid, name,
                       date, time)
                    VALUES (%s, %s, %s, %s)
                    ’’’, (cid, name, current_date,
                       current_time))
                    db_connection.commit()
                else:
                    db_cursor.execute(’’’
                        UPDATE attendance
                        SET logout_time = %s
                        WHERE cid = %s AND date = %s
                    ’’’, (current_time, cid, current_date))
                    db_connection.commit()

            # Display the result on the frame
            top, right, bottom, left = face_locations[0]
            cv2.rectangle(frame, (left, top), (right, bottom)
               , (0, 0, 255), 2)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6),
               font, 0.5, (255, 255, 255), 1)

        # Display the frame with recognized faces
        cv2.imshow(’Video’, frame)

        # Press ’q’ to exit
        if cv2.waitKey(1) & 0xFF == ord(’q’):
            break

    # Release the camera and close the window
    video_capture.release()
    cv2.destroyAllWindows()
    db_connection.close()

def download_image_from_url(url):
    response = requests.get(url)
    if response.status_code == 200:
        image_array = np.asarray(bytearray(response.content),
            dtype=np.uint8)
        image = cv2.imdecode(image_array, cv2.IMREAD_COLOR)
        return image
    else:
        return None

def save_image_locally(url, file_path):
    with open(file_path, ’wb’) as f:
        response = requests.get(url)
        if response.status_code == 200:
            f.write(response.content)
            return True
        else:
            return False

def encode_face_from_url(image_url):
    image = download_image_from_url(image_url)
    if image is not None:
        face_encodings = face_recognition.face_encodings(np.
           array(image))
        if len(face_encodings) > 0:
            return face_encodings[0]
    return None

def calculate_accuracy_from_url(reference_image_url,
   test_image_url):
    reference_image = encode_face_from_url(
       reference_image_url)
    test_image = encode_face_from_url(test_image_url)

    if reference_image is None or test_image is None:
        return None

    matches = face_recognition.compare_faces([reference_image
       ], test_image)
    return matches[0]

def test_accuracy_from_url(reference_image_url,
   test_image_urls):
    accuracy_results = []
    print(reference_image_url)
    print(test_image_urls)

    for test_image_url in test_image_urls:
        accuracy = calculate_accuracy_from_url(
           reference_image_url, test_image_url)
        accuracy_results.append(accuracy)

    return accuracy_results

def plot_accuracy_pie_chart(accuracy_results):
    labels = ['Match (1)', 'No Match (0)']
    colors = ['green', 'red']
    explode = (0.1, 0)  # explode the 1st slice (Match)

    # Count the occurrences of 1s and 0s
    count_match = accuracy_results.count(1)
    count_no_match = accuracy_results.count(0)

    # Check for invalid values in counts
    if np.isnan(count_match) or np.isnan(count_no_match):
        print("Invalid values in counts. Cannot plot pie
           chart.")
        return

    sizes = [count_match, count_no_match]

    plt.pie(sizes, explode=explode, labels=labels, colors=
       colors, autopct='%1.1f%%', startangle=140)
    plt.title('Face Recognition Accuracy Test')
    plt.show()



if __name__ == "__main__":
    # Connect to the database and retrieve image paths from
```

```
   the table
while True:
    db_connection = mysql.connector.connect(
        host='217.21.95.154',
        user='u333956788_image_root',
        password='Soft@2017',
        database='u333956788_imagedb'
    )
    db_cursor = db_connection.cursor()
    print("Welcome to Smart Attendance System")
    speak("Welcome to Smart Attendance System")
    speak("Enter your employee id please")
    x = input("Enter your employee id (or 'n' to quit):")

    if x.lower() == 'n':
        break

    sql = 'SELECT image_path FROM reg WHERE cid=%s'
    db_cursor.execute(sql, (x,))
    image_paths = [record[0] for record in db_cursor.
       fetchall()]

    db_connection.close()
    reference_image_url = image_paths[0]

    # Download and save the reference image locally
    reference_image_local_path = 'img/1.jpg'
    if not save_image_locally(reference_image_url,
       reference_image_local_path):
        print("Failed to download the image from the URL.
            Skipping employee.")
        continue
    else:
        save_image_locally(reference_image_url,
           reference_image_local_path)

    # Load the reference image from the local file path
```

```
        reference_image = cv2.imread(
           reference_image_local_path)

        # Initialize the camera
        video_capture = cv2.VideoCapture(0)  # Use the
           appropriate camera index if not the default one
        # Capture attendance for the current employee
        recognize_faces_and_mark_attendance(video_capture,
           reference_image, x)
        # Release the camera and close the window after
           capturing attendance
        video_capture.release()
        cv2.destroyAllWindows()

    # Test accuracy
    accuracy_results = test_accuracy_from_url(
       reference_image_url, image_paths[0:])
    print(accuracy_results)
    # Plot the results
    plot_accuracy_pie_chart(accuracy_results)
```

9.3 Database Connection File

```
     <?php
// *** Create the connection file ***
$con=new mysqli("localhost","u333956788_image_root","
   Soft@2017","u333956788_imagedb") or die(mysqli_error($con)
   );
?>
```

9.4 Database class to perform CURD Operation

```
   <?php
class database
{
//   private $host="localhost";
//   private $user="root";
//   private $pass="";
//   private $dbname="bishtamdb";
//   private $result=array();
//   private $mysqli="";
//   private $con=false;

private $host="localhost";
 private $user="u333956788_image_root";
 private $pass="Soft@2017";
 private $dbname="u333956788_imagedb";
 private $result=array();
 private $mysqli="";
 private $con=false;

 public function __construct()
 {
  if(!$this->con){
    $this->mysqli=new mysqli($this->host,$this->user,$this->
       pass,$this->dbname);
    $this->con=true;
    if($this->mysqli->connect_error)
    {
     array_push($this->result,$this->mysqli->connect_error);
    }
  }
 }
 //  *** function to insert the databse ***
 public function insert_db($table, $data)
 {
  $fields = array_keys($data);
 $values = array_values($data);
 $sql = "INSERT INTO $table (" . implode(’,’, $fields) . ")
    VALUES (’" . implode("’,’", $values) . "’)";
 if ($this->mysqli->query($sql) === TRUE)
 {
   echo "Data insertion successfully complete...";
   //return true;
 }
 else
 {
  echo "Error: " . $sql . "<br>" . $this->mysqli->error;
  //return false;
 }
}

// *** function to update row in a database ***
public function update_db($table, $data,$where)
{
 $sql = "UPDATE $table SET ";
 foreach ($data as $key => $value) {
   $sql .= "$key = ’$value’,";
 }
 $sql = rtrim($sql, ’,’);
 $sql .= " WHERE $where";
 if ($this->mysqli->query($sql) === TRUE) {
    echo "Data update Successfully....";
    //return true;
 } else {
    //return false;
  }
}

// *** function to Delete row in a databse ***
public function delete_db($table, $data)
{
 $idd=$data[’idd’];
 $sql="delete from $table where id=’".$idd."’";
```

```
  $res=$this->mysqli->query($sql);
  echo "Data deleted successfully...";
  //return true;
 }

 // *** Function to view the database  */
 public function view_db($table)
 {
  $sql="select * from $table";
  $res=$this->mysqli->query($sql);
  $output=array();
  while($row=$res->fetch_assoc())
  {
   $output[]=$row;
  }
  echo json_encode($output);
 }

//  *** Condition Statement ***
public function select_db($table,$data,$where)
 {
  $sql="select * from $table where $where";
  $res=$this->mysqli->query($sql);
  $output=array();
  while($row=$res->fetch_assoc())
  {
   $output[]=$row;
  }
  echo json_encode($output);
 }

 // close the connection
 public function __destruct()
 {
  if($this->con){
    if($this->mysqli->close()){
      $this->con=false;
      //return true;
    }
  }
  else{
    //return false;
  }
 }
}
?>
```

9.5 Login Page

```
    <?php
session_start();
include 'connect.php';
$user=$_POST['user'];
$pass=$_POST['pass'];
$sql="select * from login where user='".$user."' and pass='".
   $pass."'";
$res=mysqli_query($con,$sql) or die(mysqli_error($con));
$c=mysqli_num_rows($res);

if($c==1){
  while($row=mysqli_fetch_assoc($res)){
    $type=$row['type'];
  }
  if($type=='admin'){
    $_SESSION['type']=$type;
    $_SESSION['user']=$user;
    echo "<script>window.location.href='../../admin';</script
       >";
  }

  else if($type=='outlet'){
    $_SESSION['type']=$type;
    $_SESSION['user']=$user;
    echo "<script>window.location.href='../../outlet';</
       script>";
  }
}
else{
    $_SESSION['type']='';
    echo "<script>alert('Sry Userid and Password Mismatch')
       ;</script>";
    echo "<script>window.location.href='../../index.php';</
       script>";
}
```

```
mysqli_close($con);
?>
```

9.6 Controller Section

```
    var app = angular.module("myApp", ["ngRoute", "ngCookies
       "]);
app.config(function ($routeProvider) {

    $routeProvider
        .when("/", {
            templateUrl: "pages/dashboard.php",
        })
        .when("/dash", {
            templateUrl: "pages/dashboard.php",
        })
        .when('/role', {
            templateUrl: 'pages/role.html',
            controller: 'dashCtrl'
        })
        .when('/shop', {
            templateUrl: 'pages/shop.html',
            controller: 'dashCtrl'
        })
        .when('/staff', {
            templateUrl: 'pages/staff.php',
            controller: 'dashCtrl'
        })

        .when('/client', {
            templateUrl: 'pages/client.html',
            controller: 'dashCtrl'
        })

        .when('/category', {
            templateUrl: 'pages/category.html',
            controller: 'dashCtrl'
        })

        .when('/service', {
            templateUrl: 'pages/service.html',
            controller: 'dashCtrl'
        })

        .when('/attendance', {
            templateUrl: 'pages/attendence_view.html',
            controller: 'dashCtrl'
        })



        .when('/cp', {
            templateUrl: 'pages/change_password.html',
            controller: 'dashCtrl'
        })
        .otherwise({
            template: "<h1>Error 404! Not Found </h1>"
        });
});

app.controller('dashCtrl', function ($scope, $http) {

    //Role Entry Section
    $scope.roleentry = function () {
        $http({
            url: "../assets/api/role_section.php",
            method: "POST",
            data: {
                "role": $scope.role,
                "op": "insert"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.role = null;
                $scope.roleview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    }
    //    *** View the role ****
    $scope.roleview = function () {
        $http({
            url: "../assets/api/role_section.php",
            method: "POST",
            data: {
                "op": "view"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.role_data = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.role_edit = function (type) {
        $http({
            url: "../assets/api/role_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.role = data.data[0][’role’];
                $scope.idd = type;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
};


// *** Edit the role section ****
$scope.role_update = function (type) {
    $http({
        url: "../assets/api/role_section.php",
        method: "POST",
        data: {
            "id": $scope.idd,
            "role": $scope.role,
            "op": "edit"
        }
    }).then(
        function (data) {
            console.log(data.data);
            alert(data.data);
            $scope.roleview();
        },
        function () {
            alert("Error! Fetching Problem in sendData()
               ");
        });
};

// **** Delete any role ***
$scope.role_delete = function (type) {
    $http({
        url: "../assets/api/role_section.php",
        method: "POST",
        data: {
            "idd": type,
            "op": "delete"
        }
    }).then(
        function (data) {
            console.log(data.data);
            $scope.roleview();
        },
        function () {
            alert("Error! Fetching Problem in sendData()
               ");
        });
};

// *** Outlet section CURD Operation ****

//Outlet Entry Section
$scope.outletentry = function () {
    $http({
        url: "../assets/api/outlet_section.php",
        method: "POST",
        data: {
            "oname": $scope.sname,
            "address": $scope.address,
            "op": "insert"
        }
    }).then(
        function (data) {
            console.log(data.data);
            alert(data.data);
            $scope.shop = null;
            $scope.address = null;
            $scope.outletview();
        },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    }
    //    *** View the outlet ****
    $scope.outletview = function () {
        $http({
            url: "../assets/api/outlet_section.php",
            method: "POST",
            data: {
                "op": "view"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.sdata = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.outlet_edit = function (type) {
        $http({
            url: "../assets/api/outlet_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.sname = data.data[0]['oname'];
                $scope.address = data.data[0]['address'];
                $scope.idd = type;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };


    // *** Edit the outlet section ****
    $scope.outlet_update = function (type) {
        $http({
            url: "../assets/api/outlet_section.php",
            method: "POST",
            data: {
                "id": $scope.idd,
                "oname": $scope.sname,
                "address": $scope.address,
                "op": "edit"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.outletview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    // **** Delete any outlet ***
    $scope.outlet_delete = function (type) {
        $http({
            url: "../assets/api/outlet_section.php",
            method: "POST",
            data: {
                "idd": type,
                "op": "delete"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.outletview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };


    // *** Staff section ***

    //staff Entry Section
    $scope.staffentry = function () {
        $http({
            url: "../assets/api/employee_section.php",
            method: "POST",
            data: {
                "phone": $scope.phone,
                "sname": $scope.sname,
                "address": $scope.address,
                "gender": $scope.gender,
                "email": $scope.email,
                "role": $scope.role.trim(),
                "location": $scope.location.trim(),
                "quali": $scope.quali,
                "op": "insert"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.phone = null;
                $scope.sname = null;
                $scope.address = null;
                $scope.gender = null;
                $scope.email = null;
                $scope.role = null;
                $scope.location = null;
                $scope.quali = null;
                $scope.staffview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    }
    //    *** View the outlet ****
    $scope.staffview = function () {
        $http({
            url: "../assets/api/employee_section.php",
            method: "POST",
            data: {
                "op": "view"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.staff_data = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.staff_edit = function (type) {
        $http({
            url: "../assets/api/employee_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.phone = data.data[0]['phone'];
                $scope.sname = data.data[0]['sname'];
                $scope.address = data.data[0]['address'];
                $scope.gender = data.data[0]['gender'];
                $scope.email = data.data[0]['email'];
                $scope.role = data.data[0]['role'].trim();
                $scope.location = data.data[0]['location'].
                   trim();
                $scope.quali = data.data[0]['quali'];
                $scope.idd = type;

            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };


    // *** Edit the outlet section ****
    $scope.staff_update = function (type) {
        $http({
            url: "../assets/api/employee_section.php",
            method: "POST",
            data: {
                "id": $scope.idd,
                "phone": $scope.phone,
                "sname": $scope.sname,
                "address": $scope.address,
                "gender": $scope.gender,
                "email": $scope.email,
                "role": $scope.role,
                "location": $scope.location,
                "quali": $scope.quali,
                "op": "edit"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.staffview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    // **** Delete any outlet ***
    $scope.staff_delete = function (type) {
        $http({
            url: "../assets/api/employee_section.php",
            method: "POST",
            data: {
                "idd": type,
                "op": "delete"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.staffview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    //Outlet Entry Section
    $scope.categoryentry = function () {
        $http({
            url: "../assets/api/category_section.php",
            method: "POST",
            data: {
                "cname": $scope.cname,
                "op": "insert"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.cname = null;
                $scope.categoryview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    }
    //    *** View the outlet ****
    $scope.categoryview = function () {
        $http({
            url: "../assets/api/category_section.php",
            method: "POST",
            data: {
                "op": "view"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.cdata = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.category_edit = function (type) {
        $http({
            url: "../assets/api/category_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.sname = data.data[0]['oname'];
                $scope.address = data.data[0]['address'];
                $scope.idd = type;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };


    // *** Edit the outlet section ****
    $scope.category_update = function (type) {
        $http({
            url: "../assets/api/category_section.php",
            method: "POST",
            data: {
                "id": $scope.idd,
                "cname": $scope.sname,
                "op": "edit"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.categoryview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    // **** Delete any outlet ***
    $scope.category_delete = function (type) {
        $http({
            url: "../assets/api/category_section.php",
            method: "POST",
            data: {
                "idd": type,
                "op": "delete"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.categoryview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };


    // *** Service Section ***
    $scope.serviceentry = function () {
        $http({
```

```
        url: "../assets/api/service_section.php",
        method: "POST",
        data: {
            "cname": $scope.cname,
            "iname": $scope.iname,
            "type": $scope.type,
            "price": $scope.price,
            "op": "insert"
        }
    }).then(
        function (data) {
            console.log(data.data);
            alert(data.data);
            $scope.cname = null;
            $scope.iname = null;
            $scope.type = null;
            $scope.price = null;
            $scope.serviceview();
        },
        function () {
            alert("Error! Fetching Problem in sendData()
              ");
        });
}
//    *** View the outlet ****
$scope.serviceview = function () {
    $http({
        url: "../assets/api/service_section.php",
        method: "POST",
        data: {
            "op": "view"
        }
    }).then(
        function (data) {
            console.log(data.data);
            $scope.idata = data.data;
            $scope.categoryview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                  ");
            });
    };

    $scope.service_edit = function (type) {
        $http({
            url: "../assets/api/service_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.sname = data.data[0]['oname'];
                $scope.address = data.data[0]['address'];
                $scope.idd = type;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                  ");
            });
    };


    // *** Edit the outlet section ****
    $scope.service_update = function (type) {
        $http({
            url: "../assets/api/service_section.php",
            method: "POST",
            data: {
                "id": $scope.idd,
                "cname": $scope.sname,
                "op": "edit"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.serviceview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    // **** Delete any outlet ***
    $scope.service_delete = function (type) {
        $http({
            url: "../assets/api/service_section.php",
            method: "POST",
            data: {
                "idd": type,
                "op": "delete"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.serviceview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    // *** Client Section ***

    $scope.clientsave = function () {
        $http({
            url: "../assets/api/client_section.php",
            method: "POST",
            data: {
                "name": $scope.name,
                "ctype": $scope.ctype,
                "dob": $scope.dob,
                "ph": $scope.ph,
                "email": $scope.email,
                "pan": $scope.pan,
                "op": "insert"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.name = null;
                $scope.ctype = null;
                $scope.dob = null;
                $scope.ph = null;
                $scope.email = null;
                $scope.pan = null;
                $scope.clientview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                  ");
            });
    };

    $scope.clientview = function () {
        $http({
            url: "../assets/api/client_section.php",
            method: "POST",
            data: {
                "op": "view"
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.client_data = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.client_edit = function (type) {
        $http({
            url: "../assets/api/client_section.php",
            method: "POST",
            data: {
                "op": "select_condition",
                "idd": type
            }
        }).then(
            function (data) {
                console.log(data.data);
                $scope.email = data.data[0][’email’];
                $scope.name = data.data[0][’name’];
                $scope.ctype = data.data[0][’ctype’];
                $scope.dob = data.data[0][’dob’];
                $scope.ph = data.data[0][’ph’];
                $scope.pan = data.data[0][’pan’];
                $scope.idd = type;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.client_update = function (type) {
        $http({
            url: "../assets/api/client_section.php",
            method: "POST",
            data: {
                "id": $scope.idd,
                "name": $scope.name,
                "ctype": $scope.ctype,
                "dob": $scope.dob,
                "ph": $scope.ph,
                "email": $scope.email,
                "pan": $scope.pan,
                "op": "edit"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.clientview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.client_delete = function (type) {
        $http({
            url: "../assets/api/client_section.php",
            method: "POST",
            data: {
                "idd": type,
                "op": "delete"
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.clientview();
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    };

    $scope.emaild = function (type) {
        $scope.to = type;
    }

    $scope.email_send = function () {
        $http({
            url: "../assets/api/email_client.php",
            method: "POST",
            data: {
                "to": $scope.to,
                "subject": $scope.subject,
                "txt": $scope.txt
            }
        }).then(
            function (data) {
                console.log(data.data);
                alert(data.data);
                $scope.to = null;
                $scope.subject = null;
                $scope.txt = null;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                   ");
            });
    }


    // *** Attendence Section ***
    $scope.attendence_view1 = function (type) {
        $http({
            url: "../assets/api/attendence_view.php",
            method: "GET"
        }).then(
            function (data) {
                console.log(data.data);
                $scope.adata = data.data;
            },
            function () {
                alert("Error! Fetching Problem in sendData()
                    ");
            });
    };

    $scope.attendence_view = function () {
        setInterval(function () {
            $scope.attendence_view1();
        }, 2000);
    };




});
```