

Smart Attendance System using Face Recognition

J Pradyumna¹, Touqeer Khan², Kiran Kumar³

^{1,2}Student, BMS College of Commerce and Management

³Assistant Professor, BMS College of Commerce and Management

1. INTRODUCTION

Punch cards, log books, fingerprint systems, barcodes, Faces, and RFID are some of the several types of attendance systems that have been invented, yet they still have many issues, like giving users the wrong information. The desktop-based attendance system's goal is to computerize the conventional method of keeping track of attendance and offer a simple, effective way to do so in modern institutions. This project will aid institutions in recording attendance through a desktop-based application. By lessening the burden of the traditional attendance system and the resources needed to mark the pen-paper and Register Books, the Smart Attendance System is dependable, quick, and accurate. Institutions of higher learning are now worried about students' attendance patterns. This is primarily caused by pupils' general academic performance.

You will learn about the issue with facial recognition in this essay, as well as how deep learning techniques can perform at a superhuman level.

If you read this post, you will understand:

Face recognition is a complex issue that makes it difficult to recognize or confirm individuals in images and movies.

The steps involved in face recognition are detection, alignment, feature extraction, and task recognition.

For facial recognition tests, deep learning models initially came close to then surpassed human performance.

The technology automates attendance by utilizing a person's identity and face's distinctiveness. Calling out names or roll numbers to take attendance is a time-consuming and ineffective practice. The attendance procedure is automated with the aid of facial recognition technology, saving both students and tutors considerable time.

Considering

Purpose:-

The goal of this paper is to outline the specifications for the "Face Track: Facial Recognition Attendance System" development. This system uses facial recognition technology to automate the tracking of attendance in a company.

Scope:- The goal of the project is to create an automated attendance system that uses temperature and face recognition. By eliminating the need for manual recording, cutting down on time, and addressing safety

issues during the COVID-19 epidemic, the initiative seeks to improve the traditional attendance method. Employees or students will be able to mark their attendance using this technology by having their faces scanned. Accurate attendance recording will be possible thanks to the system's capture and analysis of facial features to match them with saved profiles.

Document Conventions

- The words "must," "shall," and "will" are used to denote necessary needs.
- The words "should" and "may" are used to denote desired or optional requirements.

Intended Audience:-

This guide is meant for the Face Track system's development team, stakeholders, and anybody else involved in those processes.

The following elements are included in the scope:

1. Face detection: The system uses image processing methods to precisely identify the position and area of the face.
2. Face Recognition: The system matches the detected face with the faces contained in the database using face recognition algorithms, which enables precise identification of students.
3. Temperature Detection: The project involves instruments or sensors that measure people's body temperatures. After identifying a person's face, the system checks their body temperature.
4. Recording of Attendance: When a student's face is recognized and their body temperature is within the usual range, the system automatically records their attendance.
5. Database Management: The project comprises building and maintaining a database with student data, including faces and other pertinent information needed for tracking attendance.

The scope of the project focuses on automating the attendance process, improving efficiency, and ensuring the safety of students and staff by incorporating face recognition and temperature detection technologies. The project aims to provide a reliable and time-saving solution for educational institutions and other organizations that require attendance management.

1.1 History of Facial Recognition: The first facial recognition pioneers were Charles Bisson, Helen Chan Wolf, and Woody Bledsoe. In 1964 and 1965, they started experimenting with utilising computers to recognise human faces. They manually delineated the lips, nose, and other prominent facial landmarks like the centres of the eyes. Later, they rotated mathematically on the computer to account for pose variance. To verify the identity, the distances between the facial landmarks were automatically calculated and compared with the image. The era of facial recognition had just begun. Facial recognition became a workable biometric for business when Sirovich and Kirby introduced linear algebra to the problem. Less than 100 values were needed in a system they created dubbed "Eigenface" 5 to accurately code the facial image. Discovering the existence of

Importance of Facial Recognition System: Applications using facial recognition systems are widespread. They are applied in security systems, authentication systems, verification systems, surveillance systems, etc. We are interacting with face recognition systems without even realizing it. Many Businesses are using facial recognition systems for authentication, verification, and security. There are diverse applications of this system. Countries such as United States, United Kingdom, and Australia are now installing facial recognition technologies in different public spaces such as airports, cafes, shopping areas, factory areas, and government buildings. A large retail company like Alibaba is working on the utilization of pay-by-face technology. Workspaces are using this technology to record the clock in and clock out time of the employees. Law enforcement agencies are installing cameras with facial recognition systems to identify criminals and search for missing persons. As facial recognition technology and algorithms advance, we would see it being implemented more and more in our society.

Challenges of Facial Recognition System: A facial recognition technology has the potential to transform how organizations and institutions deal with the public. However, there could be risks with this technology if it is not used appropriately. The possibility of sensitive information being used improperly is quite real. Before deploying this technology, businesses and organizations need to make sure that there are adequate checks and balances and adequate protection. The unique biometrics of each person whose face is scanned by this technique are recorded in a database. The security measures in place to safeguard the database and who owns it will determine whether or not information can be misused, leaked, or taken without the owner's permission. Systems for facial recognition are not flawless. Human data are gathered and used to train the algorithms. If there aren't enough data

1.2 Project Objective:

For both the teacher and the student in an educational setting, attendance is extremely crucial. Therefore, keeping a record of attendance is crucial. When we consider the conventional method of recording attendance in a classroom, the issue appears. Calling a student's name or roll number for attendance causes problems with both time and energy use. So, the automatic attendance system can address all of the aforementioned issues. There are a few automated systems for keeping track of attendance that are now being employed by several institutions. RFID technology and biometrics are two examples of such systems. Despite being automatic and a step up from the conventional method, it falls short of the deadline. The learner must stand in line before receiving.

The main goal of this project is to create a system that uses facial reorganisation to track student attendance. It's an intriguing idea to use authentication methods to automate the conventional attendance system. The traditional technique necessitates the manual marking of each student's attendance in a register, which takes time. Thus, the suggested concept does away with the necessity to keep an attendance log.

System:

- Improved security be provided.
- The system's upkeep is simple and affordable.

- Produce the outcome swiftly.
- Data should be accurate and useful.
- Friendly to users.

1.3 CHARACTERISTIC OF PROPOSED SYSTEM

- One-stop solution for attendance calculation
- User-friendly
- Reports are quickly generated
- Very little paper effort

1.4 Problem Statement

Every school, college, and university maintains a record of each student's attendance. According to empirical evidence, there is a strong relationship between students' academic success and their attendance. Additionally, it was asserted that pupils with bad attendance histories typically have poor retention. As a result, educators must keep accurate attendance records. The manual approach of keeping attendance records is ineffective and takes more time to organize the records and determine each student's average attendance. The need for a system to handle student record organization and average attendance calculation arises as a result. Facial recognition is one option for automating the student attendance system.

1.1. MOTIVATION

Real-time access to attendance data, identification and location verification, frequent changes to working hours, hardware and maintenance, attendance and departure reports, and department of sections and branches are all significant issues in the majority of institutions. Consequently, a pressing requirement for:

- (i) creating a quick, precise attendance system that depends on speed to save time and effort and attain social distance.
- (ii) Reduce friction and ensure worker safety by integrating databases to store attendance and present outcomes with Faces of Respective Students.
- (iii) For institutions to be fair and transparent, weekly and monthly attendance analysis is done..

1.2. DATABASE USED

Any type of data can be stored, maintained, and accessed using databases. They gather data on individuals, locations, or objects. It is gathered in one location so that it can be seen and examined. You might think of databases as a well-organized collection of data. To keep track of student information and attendance, we use a MySQL database.

Scope of the project •

We can automatically record attendance in the classroom using this system, and the output is obtained in

an excel sheet as needed in real-time. •However, in order to create a specialised system that can be used in educational institutions, a highly effective algorithm that is insensitive to the classroom lighting conditions must be created..

- Also a camera of the optimum resolution has to be utilized in the system.
- The creation of an online attendance database and automatic updating of the attendance is a crucial area where we can make progress..

Methodology :

We suggested the attendance management system based on face detection in the lecture hall in order to gather the attendance, positions, and face images. The technique uses constant observation and recording to estimate each student's attendance and placement. The methodology utilised to obtain the various weights of each focused seat is the basis for current work. These days, technology tries to impart fantastic knowledge-based technological advancements. One of the fascinating fields is deep learning, which allows a machine to train itself using various datasets as input and produce the right results during testing by using various learning methods. Today, attendance is valued as a key component for both students and teachers in educational institutions. Using the

Manual Attendance System (MAS): A teacher who is manually concerned with a student's attendance is using a manual student attendance management system. Manually recording attendance may be seen of as a time-consuming method, and occasionally the teacher may miss someone, or pupils may answer repeatedly regarding the absence of their friends. Therefore, when we consider the conventional method of taking attendance in the classroom, a difficulty occurs.

Automated Attendance System (AAS): The Automated Attendance System (AAS) uses face recognition technology to track student attendance in the classroom. Additionally, it is feasible to tell if a student is dozing off or awake during class, and it is also possible to use it in exam sessions to guarantee the student's attendance. A real-time camera installed at the entrance will be used to record pupils' attendance as they enter or leave the classroom. The camera has been trained to distinguish between photographs and shadows. The two popular methods for identifying faces in photographs are,

- 1.Feature-Based approach
- 2.Brightness Based approach

Feature-Based approach: The feature-based technique, sometimes referred to as a local face recognition system, is used to identify the most important facial traits.

Brightness Based approach: The global face recognition system, also known as the brightness-based technique, is utilized to identify every component of the image.

Feasibility Analysis

A feasibility study assesses the project's likelihood of success; as a result, perceived objectivity plays a significant role in the research's credibility with regard to possible backers and lenders. It must therefore be carried out objectively and impartially in order to give data on which judgments can be made. Here, we

go over the three main feasibility studies that are necessary for our project.

Functionality in Operation The ability of a proposed system to solve user problems is known as operational feasibility. Operational feasibility entails estimating whether the system will be used if it is designed and implemented, and it depends on the human resources available for the project. Since most teachers and employees these days are comfortable with digital technology, the project is operationally practical for the users.

Financial viability Financial

— Face Recognition Using Eigenfaces, 1991.

The 2018 paper titled “Deep Face Recognition: A Survey,” provides a helpful summary of the state of face recognition research over the last nearly 30 years, highlighting the broad trend from holistic learning methods (such as Eigenfaces), to local handcrafted feature detection, to shallow learning methods, to finally deep learning methods that are currently state of the art.

The holistic approaches dominated the face recognition community in the 1990s. In the early 2000s, handcrafted local descriptors became popular, and the local feature learning approach were introduced in the late 2000s. [...] [shallow learning method] performance steadily improves from around 60% to above 90%, while deep learning boosts the performance to 99.80% in just three years.

2018 survey titled "Deep Face Recognition: A Survey"

Deep learning techniques for face recognition underwent a frenzy of research and publications in 2014 and 2015 as a result of AlexNet's 2012 success for the less complex challenge of picture categorization. On a typical face recognition dataset, capabilities soon reached near-human-level performance, then exceeded human-level performance within a three-year timeframe, which is an astonishing rate of improvement given the past decades of effort.

These advancements were sparked by arguably four seminal systems on deep learning for face recognition: DeepFace, the Deep series of systems, VGGFace, and FaceNet. Let's talk about each briefly.

Yaniv Taigman, et al. from Facebook AI Research have described a system called DeepFace that uses deep convolutional neural networks.

2. LITERATURE SURVEY

In order to recognize relatives, friends, or other people we are familiar with, face recognition is essential in daily life. We might not realize that numerous steps have been done to recognize human faces. Our ability to acquire information and analyses it throughout the recognition process is a result of human intellect. Every organization needs a reliable and consistent way to track student attendance. Every organization has a unique system for keeping track of participants; some utilize a piece of paper and a call-out system to write down names of listeners during lectures, while others use biometric tools like fingerprint readers, RFID card readers, and iris scanners. It takes a lot of time to call each student by name individually as is customary. While other biometrics, including voice, iris, or finger print identification, each have their own drawbacks and are not completely accurate[1][9]. The clever method of the

attendance management system uses face recognition to track attendance. Compared to other systems, face recognition is more precise, quicker, and minimises the likelihood of proxy attendance. Face recognition allows for passive identification, which means that the individual who has to be identified doesn't have to do anything to prove who they are. [2]. Face recognition is a two-step process that starts with the identification of faces and moves on to matching those faces with those in the database. Numerous face detection and identification techniques have been developed. Face recognition uses geometric features such as the eyes, nose, eye brows, and cheeks as well as the appearance-based method, which covers the features of the entire face.[3]. With the use of machine learning, our method employs face identification to minimise the shortcomings of the current system. To record student photographs, a high-quality camera is needed, and the detection process is carried out using a histogram of oriented gradient. and performance recognition using deep learning. An IPC (Inter Personal Communication) bridge is created to connect the frontend side (client side), which is composed of GUI and is based on electron JS, and the backend side, which is composed of logic and Python (server side). The photos taken by the camera are uploaded to the system for further processing, where they are compared to a collection of student reference photographs to determine each student's attendance. In [4] Jomon Joseph, K. P. Zacharia introduced a system based on Matlab that makes use of image processing, PCA, Eigen faces, and a microcontroller. Their system can only process front-facing images, so an appropriate technique that takes into account the system's orientation is required. In [5], Ajinkya Patil and his colleagues introduced a face recognition system for attendance marking that makes use of the Viola Jones algorithm, Haar cascades to identify faces in images, and the Eigen face method to recognise faces. In order to make the attendance system simple and secure, the author of [6] proposed a system that makes use of artificial neural networks. They employed PCA to extract facial images, neural networks for testing and training, and their system functions in a variety of orientations. MuthuKalyani.K. and VeeraMuthu suggested a 3D facial recognition method for an attendance management system. They should keep track of attendance as well as each student's monthly progress, said A [7]. To better recognize oriented faces, a different algorithm is needed. With the aid of the PCA algorithm, an effective attendance management system was created [8]. Although it has attained accuracy levels of up to 83 percent, its system performance suffers from tiny variations in lighting.

[10] Information is transmitted to us in the form of light via the image that is projected onto the retina of our eyes. Electromagnetic waves take the shape of light, which is projected to human vision after being transmitted from a source onto an object. According to Robinson-Riegler, G., and Robinson-Riegle, B. (2008), when the human visual system has processed an image, we really identify an object based on its shape, size, contour, and texture in order to understand the data. In order to identify the items or faces, the examined data will be compared to other representations of those things that already exist in human memory. [11] In reality, creating an automated system that can detect faces with the same accuracy as a human is a difficult challenge. To recognise diverse faces, though, we need a strong memory. For instance, because there are so many students in universities of all racial and gender backgrounds, it is impossible to remember every one of their faces without making a mistake. Face recognition systems leverage computers with nearly unlimited memory, fast processing speed, and power to overcome human limitations. The human face is a special illustration of personal identity. Face recognition is thus described

as a biometric technique that identifies a person by comparing a real-time captured image with previously stored photographs in that person's database (Margaret)

In this work, they compare various face recognition algorithms. Overall, the method used to keep attendance records was good. The suggested method is intended to reduce the shortcomings of the current manual system by automating the attendance of various organizations. When the time for a certain subject arrives, the system automatically starts collecting pictures and checks to see whether there are any human faces in the photograph. This is how the system calculates attendance by subject. Administrators manually input student and subject data. We calculated and compared 128-d face features for face identification using deep learning techniques and the Histogram of Oriented Gradient for face detection. After faces are found and recognized using the database at hand, the algorithm calculates attendance.

1. Face recognition is significant in many facets of daily life, including social interactions, access control, surveillance, security systems, and personal identification. It makes it possible to identify people based on the characteristics of their faces, enhancing ease and security across a range of applications.
2. Human Visual Processing: The human visual system analyses visual data, such as object shape, size, contour, and texture, that is received by the eyes. These visual clues play a key role in identifying and understanding faces. Humans can categorise and analyse facial traits, which enables them to distinguish and identify various people.
3. Face recognition challenges: There are a number of difficulties in developing an automated face recognition system that mimics human abilities. Variances in lighting, facial emotions, position, occlusions (such as glasses or facial hair), and intra-class variations (changes in the same person's face over time) are a few of these difficulties. Complex algorithms and strategies are required to overcome these difficulties.
4. Face recognition is regarded as a biometric technique, which means that it makes use of particular behavioural or physical traits of individuals to identify them. Face recognition uses a database of saved photos to match and identify individuals based on the distinctive traits of their faces.
5. Comparison with Alternative Identifying Systems: The literature review briefly examines the drawbacks of other identification methods, including voice, iris, fingerprint, RFID card, and other biometric and RFID card-based systems. Regarding ease of use, effectiveness, accuracy, and privacy issues, each of these systems has pros and problems of its own. Face recognition has benefits include the ability to capture facial photos without being noticed and the possibility for high accuracy.
6. Systems for Monitoring Infrared Temperature: The survey briefly addresses an IoT-based application for infrared temperature monitoring systems. For the purpose of gathering current temperature information, these systems use infrared thermometer sensor modules. A web application that provides access to the collected data and a history of temperature progression is available. This illustrates how face recognition technology could be incorporated into other Internet of Things applications.

Face recognition has recently attracted a lot of attention as one of the most effective uses of picture analysis and understanding, particularly during the past several years. This trend can be attributed to at

least two factors: the first is the broad range of applications in business and law enforcement, and the second is the availability of workable technology after 30 years of study. Even if contemporary machine recognition systems have developed to a certain level of maturity, the restrictions placed on them by numerous practical applications limit their potential. For instance, it's still difficult to recognize faces in photographs of people taken outside with changing lighting and/or poses. In other words, current systems still fall far short of what the human perception system is capable of. This essay offers

3, SYSTEM REQUIREMENTS SPECIFICATION

- It is necessary to specify requirements using a specification language. Natural languages are now most frequently utilized to define needs, even when formal notations are available to specify precise system attributes. When formal languages are utilised, they are frequently used as a component of the larger SRS to specify certain attributes or for particular elements of the system.
- A clean and succinct document must contain every need for a system, whether they are expressed in formal notation or common English. The requirements document must be appropriately organized in order to do this. Here, we'll talk about how the IEEE framework was used to organize the specifications for the software needs.
- The IEEE standards take into account the possibility that different projects may need to organize their requirements.

3.1 HARDWARE COMPONENTS

- ❖ Processor- Intel Core i5
- ❖ Memory required :1GB or higher
- ❖ Storage: 1GB or higher
- ❖ Wireless Camera(with more mega pixels and high resolution)

3.2 SOFTWARE REQUIEIMENTS

- ❖ Windows 10 or next version
- ❖ Visual Studio Code (with Anaconda Virtual Environment)
- ❖ MySQL
- ❖ Python

- Mysql-connector-python
- PyQt5 framework
- Pandas
- Face-Recognition Python Module
- Opencv-python
- Numpy
- Dlib

3.3 THE PYTHON MODULES USED IN PROJECT:-

- MySQL-connector-python – MySQL Connector/Python enables Python programs to access MySQL databases, using an APWE that is compliant with the Python Database APWE Specification
- PyQt5 framework – PyQt5 is the latest version of a GUWE widgets toolkit developed by Riverbank Computing. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUWE library. PyQt5 is a blend of Python programming language and the Qt library.
- Face-Recognition - Use the world's easiest face recognition library to recognise and modify faces from Python or from the command line. developed using the most advanced facial recognition technology available from dlib. The model's precision on the Labeled Faces in the Wild benchmark is 99.38 percent. Additionally, a straightforward face recognition command line programme is provided, allowing you to do face recognition on a folder of photographs directly from the command line!
- Dlib – its strong object and facial recognition capabilities. Dlib is a cutting-edge C++ toolkit that includes machine learning techniques and tools for developing sophisticated software to address real-world issues. It is employed in a variety of fields, including robots, embedded technology, mobile phones, and huge high performance computer environments, in both industry and academics. Dlib can be used for free in any application thanks to its open source licencing. Subscribe to the development of dlib on github to keep up with it or get involved. If you plan to contribute code to the project, make sure to read the how to contribute page as well. Use these directions to build dlib to rapidly begin utilising it.
- Opencv – Cameras are displayed on the window frames to connect. Face scanning. ready-made OpenCV packages for Python that are CPU-only. If you want to compile the bindings from source to enable more modules like CUDA, look at the manual build section.
- Pandas – supplying practical data structures and operations to handle structured data with efficiency. and employ for export. Pandas is a Python-based open source data analysis and manipulation

tool that is quick, strong, adaptable, and simple to use. Now go install pandas!

- Numpy – to analyse the frame's or location's location in relation to the stored array value for the QR Code. The cornerstone Python module for scientific computing is called NumPy. A multidimensional array object, various derived objects (like masked arrays and matrices), and a variety of routines for quick operations on arrays are provided by this Python library. These operations include discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more.

3.4 INPUT/OUTPUT

Input: Face of student

Output: CSV sheet

3.5 NON-FUNCTIONAL REQUIREMENTS

1. The attendance must be recorded no later than two minutes after the activity in question.
2. Any device and all platforms must be able to access the application.
3. User privacy is protected, and the programmer responds in less than 3 seconds.

3.6 FUNCTIONAL REQUIREMENTS:

- 1) The method must cut down on the amount of time needed to take instructor attendance.
- 2) The software must be able to mark and record the student's attendance for the lectures.
- 3) The student can monitor the attendance real time.

Performance

- 1) The system must be able to support numerous concurrent users.
- 2) Each user's face recognition should be completed in [X] seconds.
- 3) A low rate of false positives and false negatives should be present in the system.

Security

- 1) User biometric data should be securely stored and encrypted.
- 2) The system has to establish safe administrator authentication procedures.

Usability

- 1) The user interface ought to be simple to use and navigate.
- 2) During the attendance marking procedure, the system should give users clear instructions.

Reliability

- 1) High availability is desired, with little downtime required for maintenance or upgrades.

- 2) In the event of a failure, the system must to include backup and recovery procedures.

User Registration:-

1. Users must be able to register their facial biometric data on the system.
2. Users should enter their name, employee or student ID, department, and class.
3. The registered facial biometrics should be securely stored by the system.

Attendance Marking

- 1) Users' faces must be captured by the system during attendance tracking.
- 2) Real-time face detection and tracking is expected from the technology.
- 3) The system must compare recorded biometrics with facial photos that have been acquired.
- 4) For users who are known to the system, attendance should be accurately recorded.

Attendance Management

1. For administrators, the system must offer a centralised interface for managing attendance.
2. The ability for administrators to examine and export attendance records is necessary.
3. Reports summarizing attendance information should be generated by the system.

3.7 About Python:

- Python is a popular computer programming language used to create software and websites, automate processes, and analyse data. Python is a general-purpose language, which means it may be used to make many various types of applications and isn't tailored for any particular issues.
- Python is frequently used to support software developers in a variety of capacities, including build control and management, testing, and many others. For build control, use SCons. For automated continuous testing and compilation, use Buildbot and Apache Gump. For project management and bug tracking, use Roundup or Trac.
- Python's syntax is simple and is similar to that of English. Python's syntax distinguishes itself from a number of other programming languages by allowing developers to create applications with fewer lines of code. The interpreter system used by Python enables the immediate execution of written code.
- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics that was developed by Guido van Rossum. In 1991, it was initially made accessible. The name "Python" is designed to be both straightforward and entertaining, paying homage to the British comic troupe Monty Python.
- Python may be used to create a wide range of applications because it works on cross-platform operating systems, including online applications, gaming applications, enterprise-level programmers, ML

applications, image processing, text processing, and much more.

- Guido van Rossum was reading the published scripts from the 1970s BBC comedy series "Monty Python's Flying Circus" while he started using Python. Van Rossum chose the name Python for the language because he felt it needed to be short, distinct, and a little mysterious.
- The majority of startups employ Python-based backend frameworks like Django, Flask, and NodeJS (JavaScript). These are likewise regarded as the best programming languages for novices to master. The most well-liked and effective programming languages that will be in demand in 2023 are listed below.
- Python is an interpreted, object-oriented, high-level, dynamically semantic programming language. Let's examine each paradigm individually. Programming language classification is based on paradigms and features.

3.8 About Visual Studio Code IDE:

Microsoft created the extremely well-liked source code editor known as Visual Studio Code (VS Code). Due to its efficiency, flexibility, and wide range of features, it has become incredibly popular among developers. The following are some intriguing features of Visual Studio Code.:

- (1) **Cross-Platform Compatibility:** VS Code is made to function flawlessly across a variety of operating systems, including Windows, macOS, and Linux. With this cross-platform compatibility, developers may utilise the operating system of their choice and still take use of the same robust features and productivity improvements.
- (2) **Lightweight and Fast:** Despite having a wide range of features, Visual Studio Code is renowned for being lightweight. Even when managing big codebases, it starts up swiftly and runs well. The responsiveness and fluidity that this efficiency assures boosts developer productivity.
- (3) **Extensibility and Customization:** Developers can tailor their coding environment to meet their unique needs using the excellent extension marketplace that VS Code provides. Developers may add new programming languages, interface with outside tools, and increase efficiency through additional functionalities thanks to the thousands of extensions that are readily available.
- (4) **Integrated Debugging and Version Control:** Developers may step through their code, create breakpoints, and analyse variables thanks to the built-in debugging features that Visual Studio Code offers for a number of programming languages. It also smoothly interfaces with well-known version control programmes like Git, facilitating effective teamwork and simplified code administration.
- (5) **IntelliSense and Code Navigation:** IntelliSense, a potent code completion function that offers context-aware suggestions as you type and cuts down on errors, is a feature that VS Code contains. Additionally, it provides powerful code navigation capabilities like symbol search, Go to Definition, and Find All References to help developers more quickly browse and comprehend codebases.
- (6) **Task Automation and Terminal Integration:** The built-in task runner in VS Code enables

developers to specify and automate repetitive operations. It supports a number of task runners and build systems, which makes it simpler to compile code, run tests, and carry out other project-specific tasks. Additionally, switching between external terminals is not necessary because the built-in terminal offers a handy command-line interface within the editor itself.

(7) **Live Share Collaboration:** Developers can collaborate in real-time with Visual Studio Code's Live Share plugin. It enables remote pair programming, coaching, and collaborative troubleshooting while allowing numerous people to work in the same codebase and edit and debug code simultaneously.

(8) **Active Community and Ecosystem:** A thriving and engaged developer community supports the ongoing growth and development of the editor in Visual Studio Code. The community produces and maintains a wide variety of extensions, themes, and practical tools to make it simpler for developers to learn about and improve their coding skills.

Developers working in a variety of fields and coding languages now frequently choose Visual Studio Code. It is the perfect tool for individual developers, teams, and organisations looking for a productive and adaptable coding environment because of its versatility, performance, and wide ecosystem.

3.9 About Anaconda:

A well-liked open-source distribution of the Python and R programming languages is called Anaconda. It offers a complete platform for scientific computing, machine learning, and data science. Here are a few crucial features of Anaconda.:

(1) **Package Management:** By offering a robust package management system called conda, Anaconda makes it easier to handle Python and R packages. Conda makes it simple to replicate and transfer projects between other systems by enabling users to establish isolated environments with particular package versions and dependencies..

(2) **Extensive Package Ecosystem:** A large number of pre-installed packages and libraries used in data science, machine learning, and scientific computing are included with Anaconda. Users save time and effort thanks to this broad package ecosystem because they can easily access and use well-known technologies like NumPy, pandas, scikit-learn, TensorFlow, and many others.

(3) **Cross-Platform Compatibility:** Because Anaconda runs on Windows, macOS, and Linux, users on various operating systems will have the same experience. Users may easily move their projects and workflows between machines thanks to this cross-platform interoperability without encountering any problems.

(4) **Integrated Development Environment (IDE):** A user-friendly and feature-rich integrated development environment called Anaconda Navigator is included with Anaconda. Users that prefer a visual interface over the command line can access Navigator since it offers a graphical user interface for managing packages, environments, and launching programmers.

(5) **Jupyter Notebook Integration:** Jupyter Notebook, a popular web-based interactive computing environment, is seamlessly integrated with Anaconda. Jupyter Notebook allows users to create and share documents that combine live code, visualizations, explanatory text, and equations. It is widely used for

exploratory data analysis, prototyping, and interactive data visualization.

(6) **Data Science Libraries and Tools:** Anaconda is the best option for data analysts and scientists because it provides a complete collection of data science libraries and tools. It includes tools for data processing like pandas, numerical computations like NumPy, machine learning tools like Scikit-Learn, data visualization tools like Matplotlib and Seaborn, and many more. These libraries offer a quicker workflow for data-centric tasks when used in conjunction with Anaconda's environment management.

(7) **Community Support:** In order to help users take advantage of the platform's full potential, Anaconda has a sizable and vibrant community of users and developers. This community offers support, tutorials, and other materials. The community regularly updates and maintains packages to give users access to the most recent developments in the fields of data science and scientific computing.

(8) **Enterprise Solutions:** Anaconda provides business options for corporations in addition to its free distribution. These products offer improved security, scalability, and collaboration capabilities that make it possible for teams to work together on data science projects and roll out products in real-world settings.

For data scientists, academics, and developers working on data-intensive projects, Anaconda has emerged as a go-to platform. It is a potent tool for creating, maintaining, and sharing data science processes and applications due to its package management features, wide ecosystem, and user-friendly interface.

3.10 DATABASE

The database of student attendance is managed using MySQL. The database stores Student ID, names, Department, year, and attendance status and much more. In here 2 students' data were stored in this database The database table is shown in below:

Figure 3. Database for a particular Student Detail

```
mysql> select * from classroom;
```

sid	name	Department	year	attendance	sex	father_name	mother_name	email	phone	address	dob	pin	religion	nationality	caste	state	city	age
1002	touqeer	BCA	2020	Absent	M	muneer	noorjahan	touqeer@example.com	7624839440	#22, bikasipura yelchenahalli 7th main 10th cross	2001-11-23	560070	islan	indian	obc	karnataka	banglore	21
1003	pradyuma	BCA	2020	Present	M	jithendra	jayalakshmi	pradyuma@example.com	6363703079	#1064, south end c cross jayanagar 9th block east end	2001-08-25	560069	hindu	indian	obc	karnataka	banglore	21

2 rows in set (0.04 sec)

4 .SYSTEM ARCHITECTURES

4.1 FLOW CHART

The flow chart of the problem is shown in the figure below.

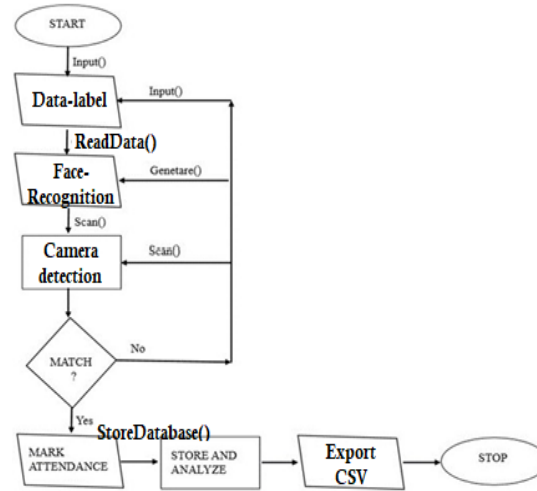


Figure 5. Flowchart of the application system

The system architecture is explained in this flowchart. Then, we show the Face to the camera so that it can scan the Face. First, we enter the student's information into the Face, which is stored in a database. Here, we build a method to retrieve all of the information from the Faces in the database. And we also modify the content scanner, which scans and stores the database's data content in files. Indicated as present if it is true; otherwise, marked as absent. It also produces the result.

4.2. USE CASE

The various actors of the system along with their functionality are described in the

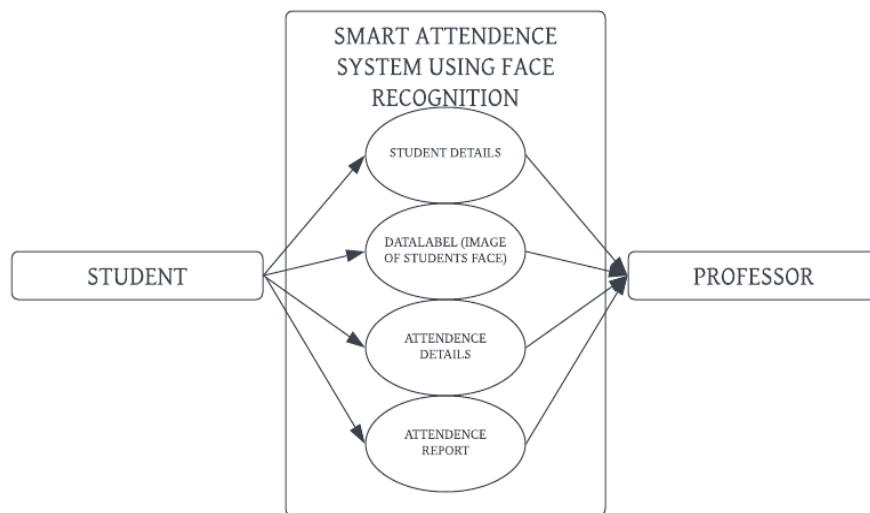


Figure 1. Use Case Diagram

We can see the many system actors in the diagram above, along with their functions. The face, which contains the student details, must be held by the student. To record his attendance in the classroom, he only needs to scan the front of the camera. The database of student information must be managed by the professor. Additionally, he must ensure that no student is engaging in questionable behaviour to indicate their attendance, such as using an attendance proxy. One student who is a friend of another student who is missing may carry the be handed to his friend to mark the attendance if the absent student has not shown up for class.

4.3. Process of Automatic Face Recognition

The challenge of recognising or authenticating faces in a photo is known as face recognition.

The following is a generic formulation of the issue of facial recognition by machines: Use a database of faces to recognise or confirm one or more people in a situation from still or video photos.

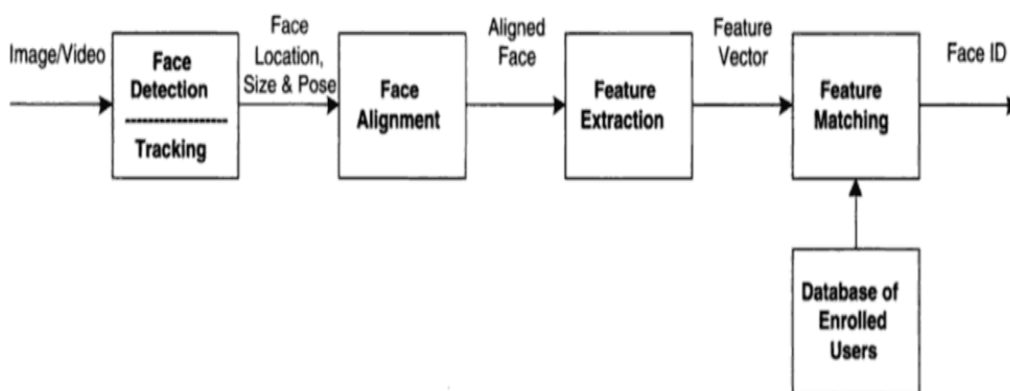
From 2003's Face Recognition: A Literature Survey.

Face detection, face alignment, feature extraction, and finally face recognition are the first four steps in many descriptions of the process of face recognition.

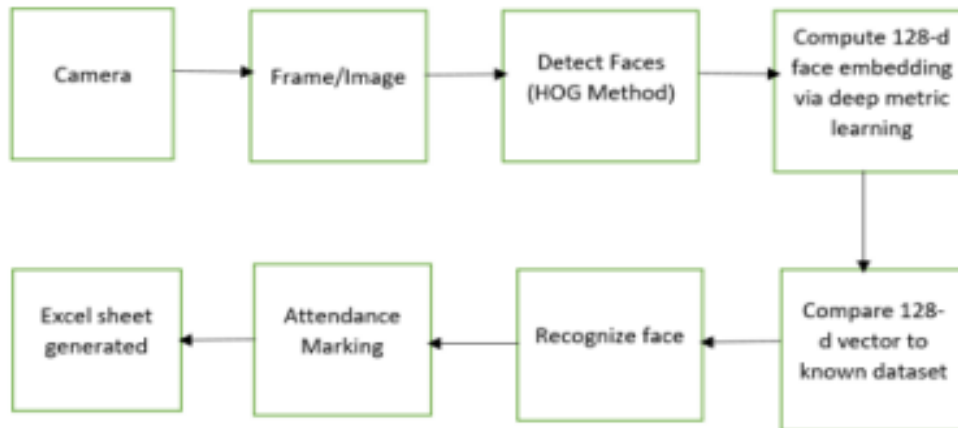
1. **Face Recognition. Determine which faces are present in the image and mark them with a bounding box.**
2. **Aligning the face. Normalize the face so that its photometrics and geometry are in line with the database.**
3. **Extraction of Features. Extract from the face features that can be applied to the task of recognition.**
4. **Recognition of faces. Match the face against one or more known faces in a database that has been prepared.**

Unlike in the past, a system may integrate any or all of the processes into a single process instead of having a separate module or programmer for each phase.

The following book, "Handbook of Face Recognition," offers a helpful summary of this procedure.:



4.4. Face recognition process



1. **Face Detection and Extraction:** Face recognition is crucial because it allows the system to recognise the human faces in an image that was acquired using a camera. Several image processing algorithms have been developed to help identify faces in images as well as their locations. We utilised the HOG approach to find people in the provided image.

2. **Face Positioning:** There are 68 different spots on a human face. Another option is to state "68 facial landmarks." The main goals of this step are to locate the image and identify face landmarks. Python scripts are used to automatically locate the face and position it as exactly as possible without distorting the image.

3. **Face Encoding:** After faces have been located in the sample image, the following step is to extract the distinguishing facial feature that can be used to identify each image. The incredibly accurate 128-d key facial points are typically retrieved for each image input whenever we obtain face localization and stored in data files for face recognition.

4. **Face matching:** The process of facial recognition ends at this phase. Deep metric learning, which is one of the greatest learning techniques and produces feature vectors with real value, is employed in this study. Our approach creates a 128-d embedding (ratification) for each face in order to validate it. To determine the Euclidean distance between a face in an image and every other face in the dataset, the internally compare faces function is used. The current image will proceed to attendance marking if it matches the existing dataset by at least 60%.

4.5. DATA FLOW DIAGRAM

The data flow diagram of the system is given in the diagram:

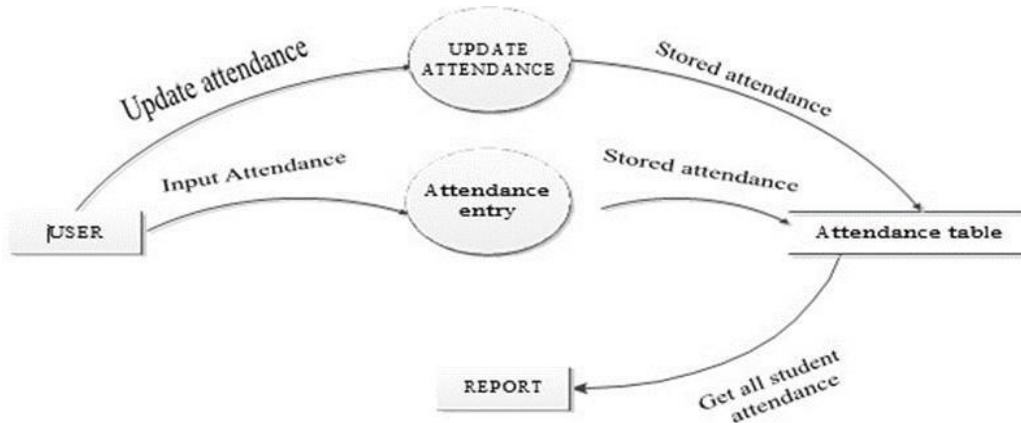


Figure 2. Use Case Diagram

Here, we can see a graphic showing how the user entered the data for the attendance entry and stored it in the attendance table. Then he exports the report after updating the attendance and storing it in the attendance table.

4.6. SEQUENCE DIAGRAM

The overall sequence diagram of the goal system is given below

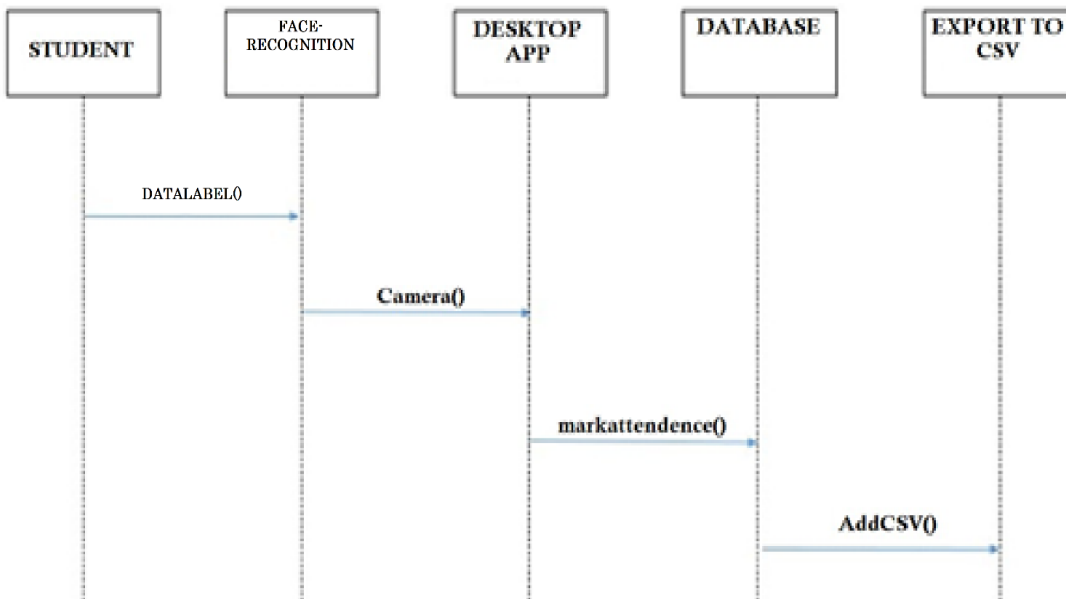


Figure 4. Sequence diagram of the application system

We may see the above diagram here. In order, the system's objective. First, we obtain student information from a database, Data-label the image, and disseminate it to the appropriate students. They are then required to appear in front of the camera to sign up for class. The desktop application that connects to a webcam and supports a student database in the classroom

5. SYSTEM IMPLEMENTATION

5.1 HOME PAGE

Here is a picture of our project's graphical user interface. Present Sir is the name of our programme, and the buttons to carry out different tasks include DASHBOARD, MANAGE STUDENT, TAKE ATTENDANCE, EXPORT TO CSV, and EXIT. The Smart Attendance Management System is easy to use and effective. Once each student has registered using the administration's established ID, the system starts operating automatically. This is our attendance system's home page. It is built using the Python framework PyQt5. The following modules make up it:

- Student Registration
- Face Recognition
- Attendance sheet generation and import to Excel(CSV) format.

The administrator must first enter the student's name, department, and roll number in order to initialise the system. For testing purposes, we have constructed a training dataset of 6 kids, each with a total of 120 photos.

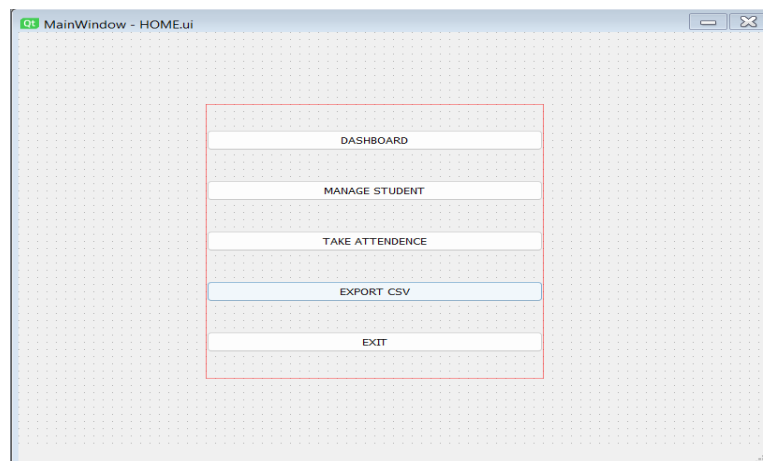


Figure 6. Home Page GUI

5.2. MANAGEMENT PAGE:

Student data can be added, updated, and deleted here; to delete a student, simply choose the column containing the row's name and click Remove Student. The data set can also be viewed in the table widgets on this Management page. With front-end GUI, it is used to delete student information from databases. It is linked to the Management page to make it more user-friendly. A unit test case for the class and its related functionality in the main module is contained in the code you submitted. You can run this script to run the tests, and the module will automatically find and use the class' stated methods. Feel free to add more test methods to account for the class's additional behaviours and functionalities. the exam

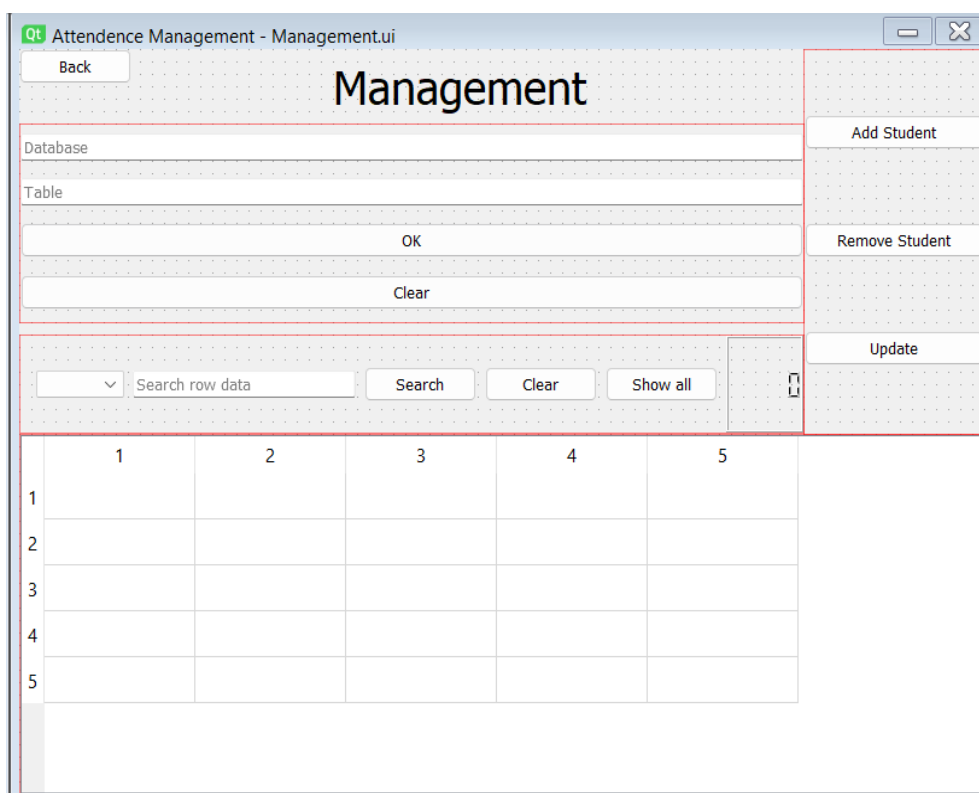
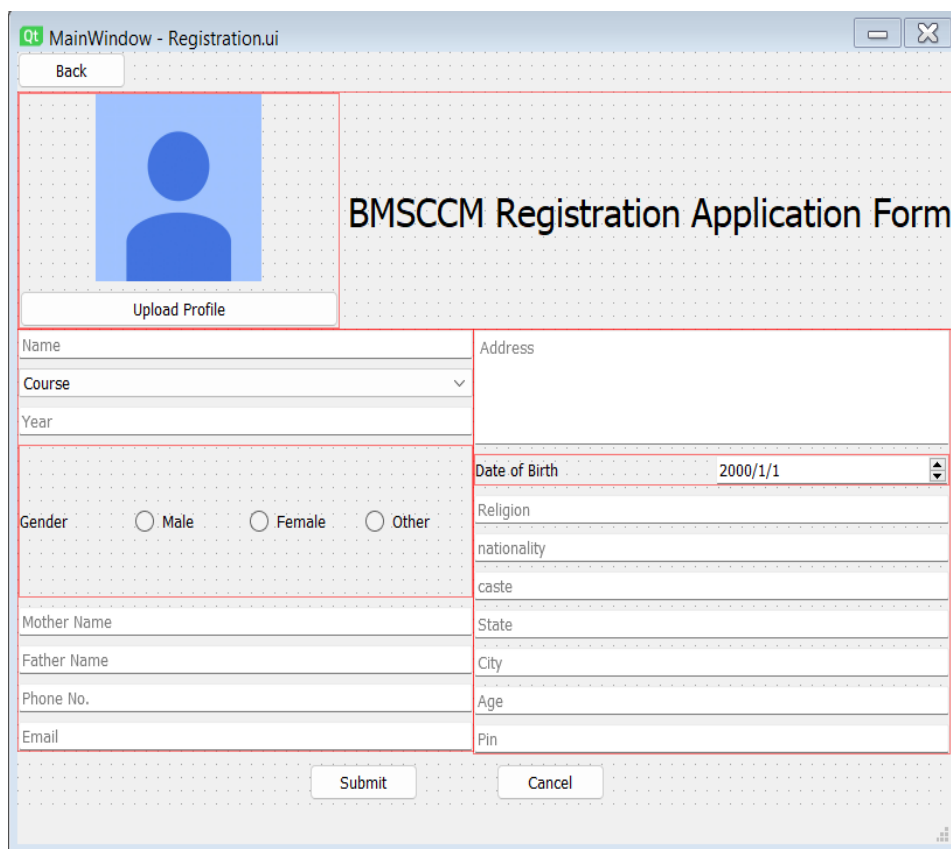


Figure 7. Attendance Management Page GUI

5.3. REGISTER PAGE:

Here, students can enter their information and submit images. Images are used to keep data labels or to save folders in which faces can be recognised. The front-end GUI is used to add student information to databases. It is linked to the Management page to make it more user-friendly. The code you offered serves as a unit test case for the main module's Register Screen class and related functionality. You can run this script to run the tests, and the module will find and run the test methods declared in the class automatically. Please feel free to add extra test methods to the Register Screen class to cover other features and behaviours.



Qt MainWindow - Registration.ui

Back

Upload Profile

BMSCCM Registration Application Form

Name	Address
Course	
Year	
Gender <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	Date of Birth 2000/1/1
Mother Name	Religion
Father Name	nationality
Phone No.	caste
Email	State
	City
	Age
	Pin

Submit Cancel

Figure 7.1. Registration Page GUI

5.4. UPDATE PAGE:

Students' details may be updated or changed here. If there are any errors in the database, it is utilised to modify the student information using the front-end GUI. The code you offered serves as a unit test case for the main module's UpdateScreen class and related features. You can run this script to run the tests, and the module will find and run the test methods declared in the class automatically. You are allowed to add more test methods to cover additional UpdateScreen class behaviours and functionality.

The test case makes sure that when the update method of the app object is called, it is called with the appropriate arguments. This test's objective is to

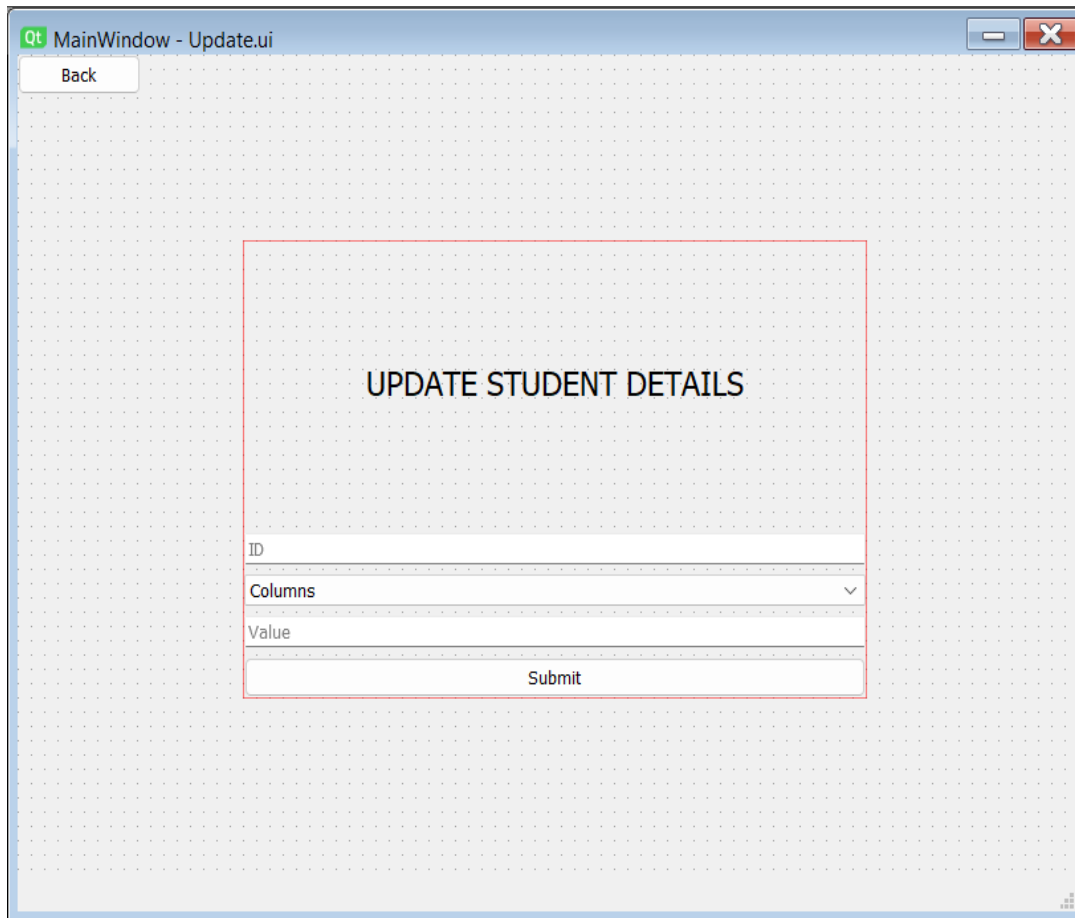


Figure 7.3. Modification Page GUI

5.5. DASHBOARD PAGE:

Through a pie chart, you can see here who is in the class and how many students there are.:



Figure 8. Dashboard Page GUI

5.6. SCANNING WINDOW:

The output of the scanned face is seen in the screenshot, which also includes the names of the students and a rectangle indicating who is there. When the time for the corresponding subject arrives, the system begins to capture images, detects faces, compares the faces with existing databases, marks attendance, and generates an excel sheet for identifying students. This is the subject folder. Subjects are to be filled out in accordance with the timetable.

3.6. EXPORTING ATTENDANCE TO CSV/XLS FILE

The teacher needs to choose the subject code in order to export the total attendance sheet in CSV format. Images may be recognised by the attendance system in a variety of lighting and angle situations. Unknown faces are those that are not included in our training dataset. Real-time attendance tracking is done for students with identifiable photos. and automatic system saving and import to an Excel sheet. The system immediately generated an Excel sheet with the relevant subject ID, date, and time.

Here is a screenshot of generated CSV file

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
StudentID	Name	Age	Departme	Email	Year	Sex	Father Na	Mother N	Phone	Address	Date of Birth	PIN	Religion	Nationalit	Caste	State	City	attendance	
1002	touqeer	21	BCA	touqeer@	2020	M	muneer	noorjahan	7.62E+09	#22, bikasi	11/23/2001	560078	islam	indian	obc	karnataka	banglore	Absent	
1003	pradyumni	21	BCA	pradyumni	2020	M	ijthendra	jayalakshn	6.36E+09	#1864, sov	8/25/2001	560069	hindu	indian	obc	karnataka	banglore	Present	

Figure 9. Classroom Data in CSV

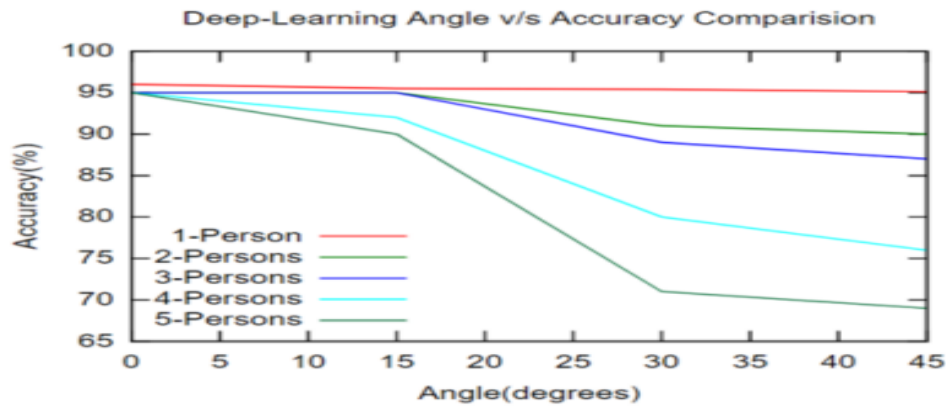


Figure 10. Accuracy graph in terms of Angles

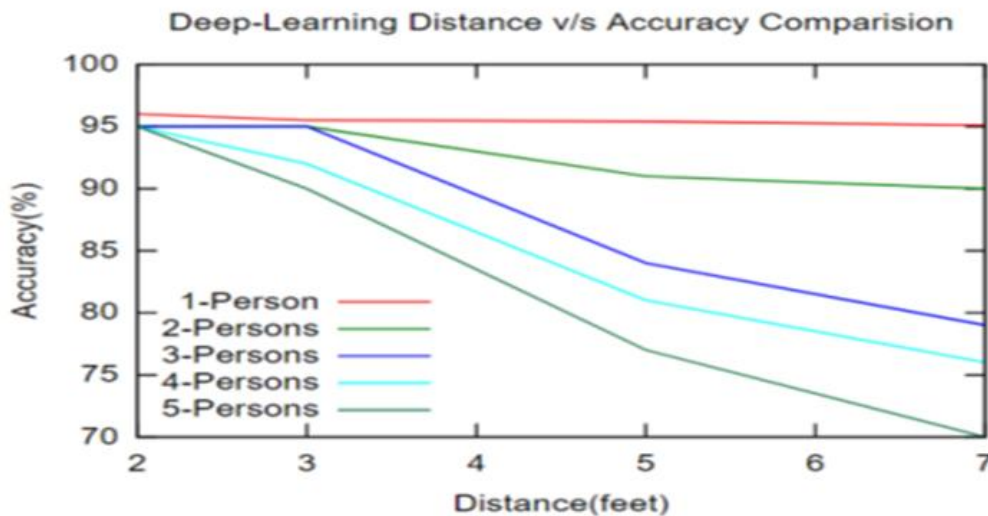
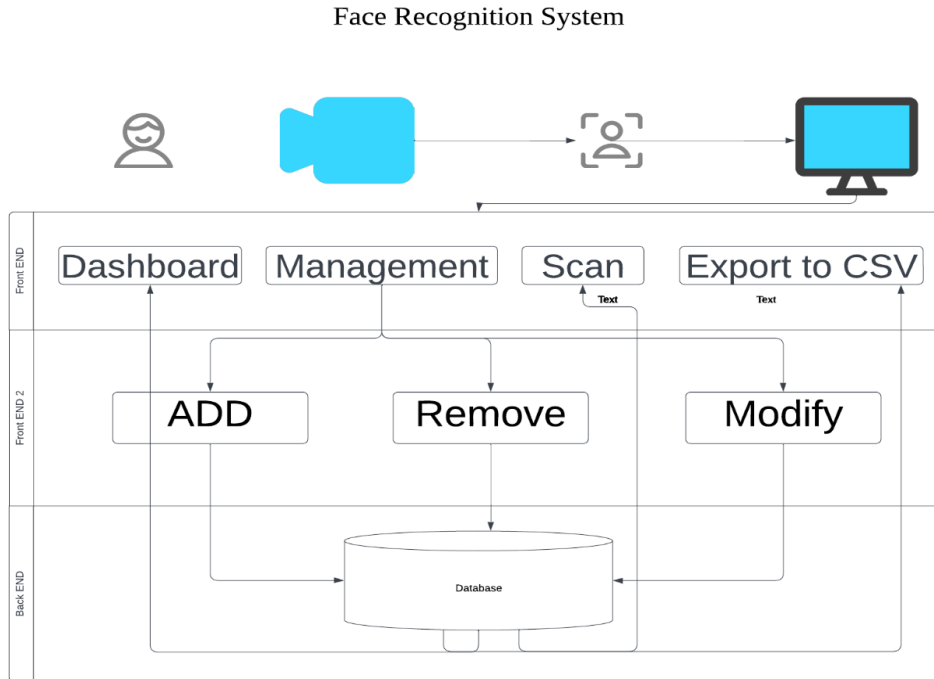


Figure 11. Accuracy graph in terms of distance



Many experiments are conducted with different distances, angles, and people; we maintain one value variable and the other two constant. the impact of increasing angle variation on system accuracy

6. IMPLEMENTATION



❖ FRONT END USED

PyQT5 Framework :- It is a Python binding to the free and open-source Qt widget toolkit, which also serves as a foundation for developing cross-platform applications. For creating GUWE apps for all the main desktop, mobile, and embedded platforms, a well-liked C++ framework is Qt (supports Linux, Windows, MacOS, Android, iOS, Raspberry Pi, and more)

Since PyQt is one of the most popular GUWE Frameworks for Python, it has a sizable community and a tonne of excellent documentation. The QWidget class, which implements a Widget, is one of PyQt's fundamental classes.

The most recent version of Riverbank Computing's GUWE widgets toolkit is PyQt5. Qt, one of the most effective and well-liked cross-platform GUWE libraries, has a Python interface. A combination of Python programming is PyQt5.

Relation of Backend and Front-End

With the use of machine learning, our method employs face identification to minimise the shortcomings of the current system. To record student photographs, a high-quality camera is needed, and the detection process is carried out using a histogram of oriented gradient. and performance recognition using deep learning. The backend side is composed of logic and Python, whereas the frontend side (client side) consists of a GUI built on PyQt5. The camera's photos are uploaded to the system for additional analysis, where they are compared to a set of student reference photographs to determine attendance.

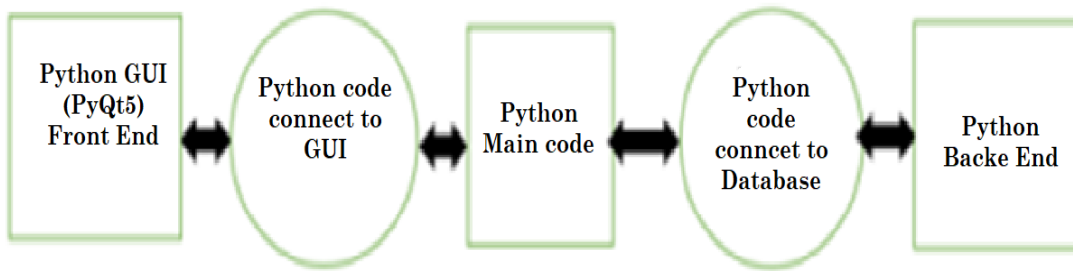


Figure 13. Communication Between Front End And Back-End

▪ **SOME OF THESE ARE:-**

- A set of GUWE widgets.
- An XML parser.
- QScintilla, a rich-text editor widget based on Scintilla.
- SVG support.
- Data-aware widgets.
- Classes to access SQL databases.
- Classes to embed ActiveX controls on Windows.

❖ **BACK END USED**

o Database using MySQL: It might be anything, such as a straightforward grocery list, a photo gallery, or the enormous amount of data in a business network. A database management system, such as MySQL Server, is required to add, access, and process data contained in a computer database.

Languages related to databases include SQL and MySQL. While MySQL is an open-source database solution that implements the SQL standard, SQL is a computer language used to interact with data in relational databases.

The relational database management system MySQL is built on the SQL (Structured Query Language) language. Data warehousing, e-commerce, and logging applications are just a few of the many uses for the application. MySQL, however, is most frequently utilised as an online database

7. RESULTS AND TESTING

- **Testing By Dlib**
 - o a logger object inspired by the well-known Java logger log4j that is thread safe

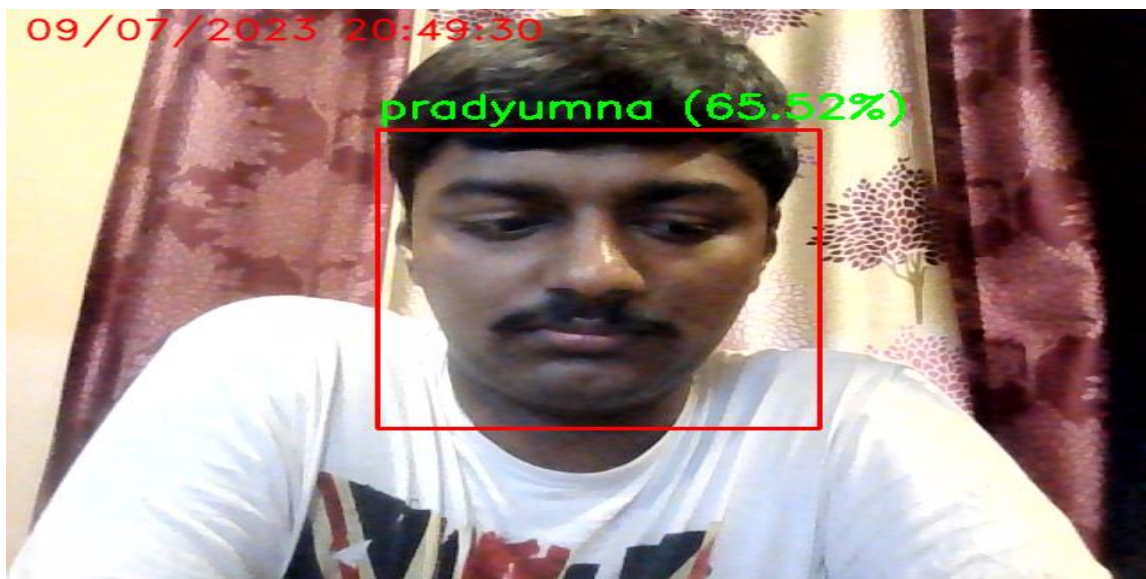
- a framework for modular unit testing several assert macros for precondition testing

UNIT TESTING:

A unit of source code is tested using the Python unittest module. Consider that you must test your project. You are aware of the type of data that the function will output. You must verify that the output is accurate after writing a lot of code. You can create and run unit tests for your code using the built-in testing infrastructure provided by Python's unittest module. It offers a collection of classes and methods for creating test plans, executing tests, and claiming desired results.

Here's a brief overview of the key components and concepts in the unittest module:

- **Test Case:** A test case is a single testing unit. It stands for a collection of test procedures gathered together to examine a selected feature or pattern in your code. Test cases are referred to as unittest subclasses. Class TestCase.
- **Test Method:** A test method is a function within a class of test cases that executes a particular test. Usually, the term "test" is at the start. Each test method should test a distinct area of your code and stand alone.
- **Assertions:** Statements used to verify the anticipated results of your tests are called assertions. unit testing. For checking conditions and ensuring that your code is valid, the TestCase class has a variety of assertion methods, including assertEquals(), assertTrue(), assertFalse(), assertRaises(), and more.
- **Test Suite:** A test suite is a set of test cases that have been compiled for the purpose of running as a single unit. Multiple test cases can be organised and executed simultaneously using test suites.
- **Test Runner:** A test runner is in charge of finding, carrying out, and reporting the tests. A built-in test runner is offered by the unittest module and can be launched using the unittest.main() function.



- **Face Scanner:-** We employ these modules—OpenCV, NumPy, and Face Recognition—to identify and scan faces. To turn on and off a webcam or other camera, or to show the frames on a window where the camera is pointed, use OpenCV. And the outcome is here.

```
PS D:\facechecker> python -m unittest Camera_test.py
['pradyumna']
1
pradyumna
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
Already Present
D:\facechecker\face_recog.py:33: ResourceWarning: unclosed file <_io.TextIOWrapper name='attendance.txt' mode='r' encoding='cp1252'>
  self.mark = Mark()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
Button clicked is: OK
OK clicked
Database connected
['touqeer', 'pradyumna']
0 record updated.
touqeer is absent
0 record updated.
pradyumna is Present
Button clicked is: OK
OK clicked
Button clicked is: OK
OK clicked
.Button clicked is: OK
OK clicked
..
-----
Ran 3 tests in 41.115s
OK
```

- **Add Student: -** The front-end GUI is used to add student information to databases. It is linked to the Management page to make it more user-friendly. The code you offered serves as a unit test case for the main module's RegisterScreen class and related functionality. You may run this script to run the tests, and the unittest module will instantly locate and carry out the test methods specified in the TestRegisterScreen class. You are allowed to add more test methods to cover additional RegisterScreen class behaviours and functionality.

```
PS D:\facechecker> & 'c:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.6.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '56054' '--
'D:\facechecker\Camera_test.py'
Button clicked is: OK
OK clicked
Database connected
1 record inserted.
Button clicked is: OK
OK clicked
.Button clicked is: OK
OK clicked
-----
Ran 1 test in 4.213s
OK
```

Modify Student:- If there are any errors in the database, it is utilised to modify the student information using the front-end GUI. The code you offered serves as a unit test case for the main module's UpdateScreen class and related features. When you run this script to run the tests, the unittest module will automatically locate and carry out the test methods specified in the TestUpdateScreen class. You are allowed to add more test methods to cover additional UpdateScreen class behaviours and functionality.

The test case makes sure that when the update method of the app object is called, it is called with the appropriate arguments. This test's objective is to ensure that the update procedure behaves as anticipated.

```
PS D:\facechecker> d:; cd 'd:\facechecker'; & 'C:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.6.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '56679' '--' 'D:\facechecker\camera_test.py'
Button clicked is: OK
OK clicked
Database connected
1 record updated.
Button clicked is: OK
OK clicked
Button clicked is: OK
OK clicked
-----
Ran 1 test in 4.199s
OK
```

- Delete Student: - With front-end GUI, it is used to delete student information from databases. It is linked to the Management page to make it more user-friendly. The code you gave is a unit test case for the main module's DeleteScreen class and related functions. When you run this script to run the tests, the unittest module will automatically locate and carry out the test methods specified in the TestDeleteScreen class. You are free to include additional test methods to cover the DeleteScreen class's various features and behaviours. The test case makes sure that when the delete method of the app object is called, it is invoked with the appropriate arguments. This test checks to see if the delete action operates as intended.

```
PS D:\facechecker> d:; cd 'd:\facechecker'; & 'C:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.6.1\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '56882' '--' 'D:\facechecker\Remove_test.py'
Button clicked is: OK
OK clicked
Database connected
0 record delete.
Button clicked is: OK
OK clicked
Button clicked is: OK
OK clicked
-----
Ran 1 test in 4.028s
OK
PS D:\facechecker> █
```

Marking Attendance:- We retrieve all the data from the databases using the mysql-connector-python-module, and we compare it to the text file's entry for the student's name. and enter text into a file. attendance.txt is a text file. If both values are the same, they will be indicated as present; otherwise, they will be marked as absent. This indicates that the value of the face that the scanner recognised will be saved in a text file. The identical file will be compared. The code you offered is a unit test case for the Attendance module's Mark class. The markattendance approach is being tested.

When you run this script to run the tests, the unittest module will automatically find and run the test methods specified in the TestMark class.

Be sure to substitute Attendance in for Mark's Attendance import.ore test methods to cover other functionalities and behaviors of the **Mark** class.

```
PS D:\facechecker> d;; cd 'd:\facechecker'; & 'C:\Users\user\anaconda3\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.python-2023.6.1\pythonFiles\Lib\pytho
n\debugpy\adapter\..\debugpy\launcher' '57595' '--' 'D:\facechecker\demo1.py'
['pradyumna']
Button clicked is: OK
Database connected
['tougeer', 'pradyumna', 'John']
0 record updated.
tougeer is absent
0 record updated.
pradyumna is Present
0 record updated.
John is absent
0 record updated.
Button clicked is: OK
OK clicked

-----
Ran 1 test in 2.591s

OK
PS D:\facechecker>
```

- **Login page:** - When a user logs in, they see this page. This is the project's primary programme, and it connects to each home page. A unit test case for your module's Main App class is shown in the code you supplied. It makes use of the unittest. Several test methods are defined in the TestCase class to test various elements of the functionality of the MainApp. When you run this script to run the tests, the unittest module will automatically find and run the test methods specified in the MainAppTestCase class. Make careful to substitute the correct name or path to the module containing the MainApp class for your module in from your module import MainApp. Please feel free to add extra unit tests to the MainApp class to cover other methods and behaviours.

```
PS D:\facechecker> python -m unittest Login_test.py
.Button clicked is: OK
OK clicked
..
-----
Ran 3 tests in 2.587s

OK
```

Open File Dialog: - When a user clicks Upload Image, a file selection dialogue appears. This method lets you open a folder and upload pictures. Your code tests the openFileDialog function of the MainApp class as part of a unit test scenario. It makes use of the unittest. There is a test function named test openFileDialog in the TestCase class. When you run this script to run the tests, the unittest module will automatically locate and carry out the test methods specified in the TestMainApp class. Make sure to substitute the appropriate name or path to your MainApp class implementation for main in from main import MainApp.

You are free to include extra test methods to cover the MainApp class's additional features and behaviours.

```
PS D:\facechecker> python -m unittest OpenFile_test.py
.
-----
Ran 1 test in 0.211s

OK
```


8. CONCLUSION

Nowadays, it's imperative to stay current with emerging technologies and fashions, particularly in the computer industry. Educational institutions are eager to improve the educational process by utilising the most recent technologies and software to cut down on manual labour. We considered adopting QR technology to make the most of the full allotted lecture time after taking note of the current scenario. When courses get larger and more crowded, it might be thought of as a waste of lecture time when teachers take the time to take attendance. Instead of wasting time on attendance, this technique dedicates the entire session to the lecture. This opens the door to the "DIGITAL WORLD." In conclusion, of all of the possibilities, taking attendance using Face is the most affordable and flexible.

References

1. Kar, Nirmalya, et al. "*Study of implementing automated attendance system using face recognition technique.*" International Journal of computer and communication engineering 1.2 (2012): 100.
2. RoshanTharanga, J. G., et al. "*Smart attendance using real time face recognition (smart-fr).*" Department of Electronic and Computer Engineering, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka (2013)
3. Selvi, K. Senthamil, P. Chitrakala, and A. Antony Jenitha. "*Face recognition based attendance marking system.*" Corresponding Author: S. Rajkumar*, Email: rajkumarsrajkumar@ gmail. com (2014)
4. Joseph, Jomon, and K. P. Zacharia. "*Automatic attendance management system using face recognition.*" International Journal of Science and Research (IJSR) 2.11 (2013): 327- 330.
5. Patil, Ajinkya, and Mrudang Shukla. "*Implementation of classroom attendance system based on face recognition in class.*" International Journal of Advances in Engineering & Technology 7.3 (2014): 974
6. Kanti, Jyotshana, and Shubha Sharm. "*Automated Attendance using Face Recognition based on PCA with Artificial Neural Network.*" International journal of science and research IJSR(2012).
7. MuthuKalyani, K., and A. VeeraMuthu. "*Smart application for AMS using face recognition.*" Computer Science & Engineering 3.5 (2013): 13.
8. Deshmukh, Badal J., and Sudhir M. Kharad. "*Efficient Attendance Management: A Face Recognition Approach.*" (2014).
9. Mehta, Preeti, and Pankaj Tomar. "*An Efficient Attendance Management Sytem based on Face Recognition using Matlab and Raspberry Pi 2.*" International Journal of Engineering Technology Science and Research IJETSRS 3.5 (2016): 71-78
10. Robinson-Riegler, G., & Robinson-Riegle, B." *the image projected into our eyes*" Robinson-Riegler, G., & Robinson-Riegle, B. (2008)
11. Margaret Rouse. "*Big Data and Data Storage*"(Margaret Rouse, 2012)
12. Wagh, Priyanka, et al. "*Attendance system based on face recognition using eigen face and PCA*

- algorithms*." 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, 2015.
13. Bhattacharya, Shubhobrata, et al. "Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment." 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT). IEEE, 2018.
 14. Li, Xiang-Yu, and Zhen-Xian Lin. "Face recognition based on HOG and fast PCA algorithm." The Euro-China Conference on Intelligent Data Analysis and Applications. Springer, Cham, 2017.
 15. Arsenovic, Marko, et al. "FaceTime—Deep learning based face recognition attendance system." 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 2017.
 16. Rekha, N., and M. Z. Kurian. "Face detection in real time based on HOG." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 3.4 (2014): 1345-1352.
 17. Kwolek, Bogdan. "Face detection using convolutional neural networks and Gabor filters." International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2005.
 18. Ashwini, C., et al. "An Efficient Attendance System Using Local Binary Pattern and Local Directional Pattern." Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org 8.4 (2018).
 19. Karnalim, Oscar, et al. "Face-face at classroom environment: Dataset and exploration." 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, 2018.
 20. Mian, Ajmal. "Realtime face detection and tracking using a single pan, tilt, zoom camera." 2008 23rd International Conference Image and Vision Computing New Zealand. IEEE, 2008.
 21. [PyQt5 · PyPI](#)
 22. [Python | Introduction to PyQt5 - GeeksforGeeks](#)
 23. [PyQt5 Tutorial \(tutorialspoint.com\)](#)
 24. [PyQT5 - Python Tutorial \(pythonspot.com\)](#)
 25. [Qt Designer Manual](#)
 26. <https://www.youtube.com/watch?v=A6464U4bPPQ&t=325s&pp=ygUXZmFjZSB5ZWNVZ25pdGlubiBweXRob24%3D>
 27. [face-recognition · PyPI](#)
 28. [face_recognition/examples/facerec_from_webcam_faster.py at master · ageitgey/face_recognition · GitHub](#)