

COMPUTER VISION-BASED ATTENDANCE TRACKING SYSTEM WITH VOICE FEEDBACK

A Dissertation submitted in partial fulfillment of the requirements

For the award of the Degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By

Mohammed Irfan Ali (1604-19-735-015)

Mohammed Abdul Shoaib (1604-19-735-019)

Mohammed Sadiq Uddin (1604-19-735-030)



MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

BANJARA HILLS, HYDERABAD-500 034

(AFFILIATED TO OSMANIA UNIVERSITY)

2023



MUFFAKHAM JAH
COLLEGE OF ENGINEERING & TECHNOLOGY
(Est. by Sultan-Ul-Uloom Education Society in 1980)
(Affiliated to Osmania University, Hyderabad)
Approved by the AICTE & Accredited by NBA

CERTIFICATE

It is certified that the work contained in the project report titled "**Computer Vision-based Attendance Tracking System with Voice Feedback**" submitted by :

Mohammed Irfan Ali **(1604-19-735-015)**

Mohammed Abdul Shoaib **(1604-19-735-019)**

Mohammed Sadiq Uddin **(1604-19-735-030)**

has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Project Supervisor

Dr. Salma Fauzia

Electronics and Communication Engineering
 MJ College of Engineering & Technology
 Hyderabad – 500 034.

Signature of Head

Dr. Ayesha Naaz

Electronics and Communication Engineering
 MJ College of Engineering & Technology
 Hyderabad – 500 034.

External Examiner

ACKNOWLEDGEMENT

The austerity and satisfaction that one gets on completing a project cannot be fulfilled without mentioning the people who made it possible with gratitude.

We are grateful to the almighty God who helped us all the way throughout the project and also has molded us into what we are today. We express our sincere thanks to our parents who always encouraged us to achieve our goals.

We offer our sincere thanks to **MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY** for allowing us to do our major project in their esteemed institution.

We show gratitude to Prof. **BASHEER AHMED** (Advisor cum Director) for having provided all the facilities and support. We would like to thank **Dr. Ayesha Naaz**, (Head of the Department, Electronics and Communication Engineering Department) for her expert guidance and encouragement at various levels of project.

We are Thankful to our guide **Dr. Salma Fauzia** (Associate Professor, Electronics and Communication Engineering) for her sustained inspiring Guidance and cooperation throughout the process of this project report.

We express our deep sense of gratitude and thanks to all the Teaching and Non-Teaching Staff of our college who stood with us and helped us to make it a successful venture.

| | |
|------------------------------|--------------------------|
| Mohammed Irfan Ali | (1604-19-735-015) |
| Mohammed Abdul Shoaib | (1604-19-735-019) |
| Mohammed Sadiq Uddin | (1604-19-735-030) |

TABLE OF CONTENTS

| | |
|---|----|
| ABSTRACT | 7 |
| 1. INTRODUCTION | 9 |
| 1.1 Introduction | 10 |
| 1.2 Problem Statement | 11 |
| 1.3 Aim of thesis | 11 |
| 1.4 Motivation of thesis | 12 |
| 1.5 Facial Recognition | 12 |
| 1.6 OpenCV | 13 |
| 2. LITERATURE SURVEY | 14 |
| 3. HARDWARE DESIGN | 23 |
| 3.1 Introduction | 23 |
| 3.2 Webcam | 24 |
| 3.3 Flowchart | 25 |
| 3.4 Raspberry Pi | 26 |
| a. Introduction | 27 |
| b. Chip description | 28 |
| c. Raspberry Pi Pin Description | 29 |
| d. Specifications | 29 |
| e. Application | 30 |
| 3.5 3D printed face mask | 31 |
| 3.6 Oled display | 32 |
| 3.7 Arduino | 33 |
| a. Introduction | 34 |
| b. ATMega328P Chip | 35 |
| c. Arduino UNO Pin Description | 36 |
| d. Specifications | 38 |
| e. Application | 39 |
| 4. FACIAL RECOGNITION INTRODUCTION | 41 |
| 4.1 Introduction | 41 |
| 4.2 History | 42 |
| 4.3 Some common facial recognition techniques | 43 |
| 4.4 Benefits and Shortcomings | 43 |
| 5. SOFTWARE IMPLEMENTATION | 44 |
| 5.1 Introduction | 44 |
| 5.2 Arduino IDE | 45 |
| 5.3 Visual Studio Code | 46 |
| 5.4 Python | 47 |
| 5.5 Libraries | 48 |

| | |
|----------------------------------|--------|
| a. face_recognition | 49 |
| b. cv2 | 50 |
| c. numpy | 51 |
| d. pytsx3 | 52 |
| e. date_time | 53 |
| f. time | 54 |
| 5.2 Web Development | 55 |
| a. HTML | 56 |
| b. CSS | 58 |
| c. Javascript | 59 |
| d. jQuery | 60 |
| 5.7 Data Visualisation | 61 |
| 5.8 Web Hosting Configuration | 62 |
| 6. OUTPUT AND RESULTS | 64 |
| 6.1 Introduction | 64 |
| 6.2 Output | 65 |
| 6.3 Result | 66 |
| 7. CONCLUSION | 70 |
| 8. REFERENCES | 71 |
| 9. APPENDIX 1 | 73 |
| 10. APPENDIX 2 | 74 |

ABSTRACT

The applications of facial recognition technology are limitless, with potential uses ranging from law enforcement and security to marketing and retail. In this project, we have focused on implementing this technology for automated attendance taking in educational institutions. The traditional process of manual attendance taking in educational institutions has been a time-consuming and error-prone task for teachers, leading to inaccuracies in attendance records and difficulties in tracking student progress. The recent COVID-19 pandemic has highlighted the need for contactless attendance systems to ensure the safety of students and staff.

To address these challenges, we have developed a highly accurate and efficient attendance system that utilizes advanced facial recognition technology. The system is built on a Raspberry Pi 4B and incorporates a 3D printed face mask with OLED displays as eyes that blink accordingly by the signal sent from an Arduino whenever it talks, and speakers connected to it. We have implemented the system using Python and relevant libraries such as `face_recognition` and `cv2`, ensuring a robust and secure solution for automated attendance taking.

Facial recognition technology uses advanced algorithms and neural networks to identify unique features of an individual's face, providing a highly accurate and efficient method for identification. Our system leverages this technology to create a fast and reliable attendance system that saves time and resources for educational institutions. To ensure the accuracy of our system, we use a combination of facial landmarks and deep learning algorithms to create a facial encoding, which is a vector representation of a face. This encoding is then compared to a database of known encodings to identify the person in the image.

To ensure seamless integration with existing educational technology infrastructure, we have developed a web interface built using HTML, CSS, Javascript, and the data visualization library Plotly. This interface provides a user-friendly and customizable experience for educational institutions, allowing them to easily track attendance and view attendance reports.

The implementation of facial recognition technology in attendance systems raises important concerns around privacy and security. We have carefully considered these issues and implemented appropriate safeguards to ensure the responsible and

ethical use of this technology. Our system uses open-source libraries and provides a transparent and customizable solution for educational institutions.

In conclusion, our attendance system provides a highly accurate and efficient solution for automated attendance taking in educational institutions. By leveraging advanced facial recognition technology, we have created a fast and reliable system that saves time and resources while ensuring the safety and accuracy of attendance records. Our system is designed to handle a large number of students and can be easily integrated into existing educational technology infrastructure.

1. INTRODUCTION

1.1 Introduction

The project is a facial recognition-based attendance system using the Raspberry Pi 4B, which is designed to capture the attendance of students in a classroom setting using facial recognition technology. The system uses a camera to capture the image of a student's face and then identifies the student using facial recognition algorithms. The system then stores the attendance data in a database, which can be accessed and used for record-keeping purposes.

Facial recognition technology has been gaining traction in recent years due to its ability to accurately identify individuals based on their facial features. The technology is widely used in security applications such as surveillance and access control systems. The application of this technology in the educational sector is not widespread but can be extremely beneficial in improving attendance management and reducing the workload of teachers in terms of manual attendance taking.

The use of the Raspberry Pi 4B as the hardware platform for the system provides a low-cost and easily accessible solution for implementing the facial recognition attendance system. The Raspberry Pi is a popular single-board computer that can run a variety of applications and is widely used in education and DIY projects. The availability of GPIO pins on the Raspberry Pi also makes it easy to integrate with other hardware components, such as the camera module used in this project.

The project's software is implemented in the Python programming language using various libraries such as OpenCV, Numpy, Pytsxs3, and others. The OpenCV library is used for facial recognition, while Numpy is used for image processing. Pytsxs3 is used for text-to-speech conversion, which can provide audio feedback to the user. The web-based interface is created using HTML, CSS, JavaScript, and jQuery, and the web server is implemented using Node.js and Express.

The project's system workflow starts with the camera capturing the image of the student, which is then processed using facial recognition algorithms to identify the student. The attendance data is then stored in a database, and the student is marked as present in the system. The system provides a user-friendly interface for the teacher to view attendance records and generate reports. The text-to-speech feature provides an additional layer of accessibility for visually impaired users.

The project's implementation is expected to provide benefits such as reducing the workload of teachers in taking attendance manually, improving attendance management accuracy, and reducing the potential for attendance fraud. The system's ability to generate automated reports will also reduce the time and effort required in manual record-keeping.

In conclusion, the facial recognition-based attendance system using the Raspberry Pi 4B is a modern and innovative solution to the age-old problem of attendance management. The use of low-cost hardware and open-source software provides a cost-effective and accessible solution for educational institutions. The system's accuracy and automation capabilities will result in a more efficient and streamlined attendance management process. The project's potential to be further developed and expanded with additional features and capabilities makes it a valuable contribution to the education sector.

1.2. Problem Statement

Manual attendance taking in educational institutions is often inefficient, prone to errors, and time-consuming. To address this challenge, we have developed a facial recognition-based attendance system that automates the process, saving time and resources while ensuring the accuracy and safety of attendance records.

1.3. Objective

The project aims to provide a robust attendance system for educational institutions that can significantly reduce the manual workload and enhance the efficiency of the attendance taking process.

- 1) Develop a facial recognition-based attendance system that automates the attendance-taking process in educational institutions, eliminating the need for manual intervention, thus reducing the risk of human errors and saving time and resources.
- 2) Implement advanced facial recognition technology using Python libraries such as face_recognition and cv2 for accurate detection of students and prevent attendance manipulation by avoiding attendance taking for the second time on a single day.
- 3) Ensure the safety and privacy of students' data by using data validation techniques and implementing necessary security measures to protect the information from unauthorized access or cyber-attacks.
- 4) Utilize Pyttsx3 for voice conversion to make the attendance system more interactive, with 3D printed face masks and OLED displays as eyes that blink accordingly by signals sent from an Arduino and speakers connected to it.
- 5) Provide a user-friendly web interface using HTML, CSS, and JavaScript for easy access and monitoring of attendance records, enabling educational institutions to customize the system according to their requirements and generate attendance reports.
- 6) Use data visualization techniques and the Plotly library to represent the attendance data in a graphical form for better visualization, analysis, and monitoring of student attendance records.
- 7) Optimize the system's performance by using Raspberry Pi 4B for background processing, ensuring faster and smoother operation of the attendance system.

1.4. Motivation of thesis :

The motivation behind this project is to address the limitations and drawbacks of traditional attendance management systems that rely on manual attendance taking. Manual attendance taking can be time-consuming, error-prone, and tedious for both teachers and students. Additionally, traditional attendance systems do not provide real-time attendance records, which can lead to inaccurate reporting and inefficient monitoring of student attendance. Therefore, the motivation of this project is to develop an automated attendance system using facial recognition technology that can provide accurate, efficient, and real-time attendance records while reducing the workload of teachers and students. The project's ultimate goal is to optimize the attendance management process for educational institutions and improve the overall learning experience for students.

1.5. Facial Recognition :

Facial recognition technology is a type of biometric technology that uses software algorithms to identify and verify an individual's identity based on their facial features. This technology is used in a variety of applications, including security systems, authentication processes, and digital marketing.

Facial recognition technology works by capturing an image or video of a person's face and using complex mathematical algorithms to analyze and compare facial features against a database of known faces. The system analyzes factors such as the distance between the eyes, the shape of the nose, and the contours of the face to create a unique facial signature that can be used to identify the person.

One of the primary advantages of facial recognition technology is its speed and accuracy. The technology can quickly process large amounts of data and accurately match the person's face against a database of known faces. This makes it an ideal solution for security systems and authentication processes, as it can quickly identify unauthorized individuals and prevent security breaches.

Facial recognition technology has many practical applications in various fields. For example, law enforcement agencies use this technology to identify suspects in criminal investigations, while border control agencies use it to verify the identity of travelers. In healthcare, facial recognition technology can be used to help diagnose genetic disorders, while in retail, it can be used to analyze customer behavior and provide personalized recommendations.

Despite its many benefits, facial recognition technology is not without its limitations and controversies. One of the main concerns is privacy, as the technology involves the collection and storage of personal data. Additionally, the accuracy of the technology can be affected by factors such as lighting, facial expressions, and changes in appearance. Therefore, it is crucial to consider the ethical and legal implications of facial recognition technology and implement necessary security measures to protect personal data.

1.6. OpenCV :

cv2, or OpenCV (Open Source Computer Vision Library), is an open-source computer vision and machine learning software library used in a wide variety of applications. It

is designed to help developers create computer vision algorithms for real-time image and video processing.

cv2 provides a set of pre-built tools and functions to process and analyze images and videos in real-time. Some of the key features of cv2 include image and video processing, object detection and recognition, feature detection and extraction, and deep learning. cv2 also supports a wide range of programming languages, including C++, Python, and Java, making it highly accessible and versatile.

Some common applications of cv2 include face detection, object recognition, motion tracking, and image segmentation. It is widely used in fields such as robotics, surveillance, augmented reality, and medical imaging.

cv2 has become an important tool for researchers, developers, and engineers in the computer vision and machine learning fields. Its flexibility and extensive library of tools make it a popular choice for those looking to implement computer vision algorithms in their projects. With the increasing demand for intelligent systems and computer vision applications, cv2 is likely to continue to play a significant role in the development of new technologies in the future.

2. LITERATURE SURVEY

In [1] Kawaguchi introduced a lecture attendance system with a new method called continuous monitoring, and the student's attendance is marked automatically by the camera which captures the photo of a student in the class. The architecture of the system is simple since two cameras are equipped with the wall of the class. The first one is a capturing camera used to capture the image of a student in the class and the second camera is a sensor camera used to get the seat of a student inside the class and the camera capturing will snap the image of the student. The system compares the picture taken from a camera capturing images and faces in the database done much time to perfect the attendance. Another paper proposed by [2] introduced a real-time computer vision algorithm in the automatic attendance management system. The system installed the camera with non-intrusive, which can snap images in the classroom and compared the extracted face from the image of the camera capturing with faces inside the system. This system also used machine learning algorithms which are usually used in computer vision. Also, HAAR CLASSIFIERS used to train the images from the camera capturing. The face snap by the camera capturing will convert to grayscale and do subtraction on the images; then the image is transferred to store on the server and processed later. In 2012 N. Kar [19] introduced an automated attendance management system using face recognition technique which used the Principal Component Analysis To implement the system, using two libraries such OpenCV is a computer vision library and FLTK(Light Tool Kit. Both of these libraries helped the development such as OpenCV support algorithm[20] and FLTK [21] used to design the interface. In the system, there are Request Matching and Adding New facts to Database. In Request Matching, the first step is to open the camera and snap the photo after the extraction of the frontal face. The next step is recognizing the face with the training data and project the extracted face

onto the Principal Component Analysis. The final step displays the nearest face with the acquired images. Apart from that, adding a new face into the database is to snap the photo after that, extract the frontal face images and then perform the Haar cascade Method to find the Principal Component Analysis Algorithm. The final step is storing the information inside the face XML file. The system is focused on the algorithm to improve the face detection from acquired images or videos. In [3] the author also proposed a system which implements automatic attendance using face recognition. The system which can extract the object in the face such as the nose, mouth by using MATLAB with Principal Component Analysis (PCA). The system [7] designed to resolve the issues of attendance marking systems such as time consuming. As the result of the experiment shows in this paper, the system can recognize the dark background or different view of the face in the classroom. Jyotshana Kanti [4] proposed a smart attendance marking system which combines two differencing algorithms such as Principal Component Analysis and Artificial Neural Network. The purpose of the author is to solve the traditional attendance marking system and to resolve the time-consuming. In the system implemented with Principal Component Analysis, it does an extraction and identifies the similarities of the face database and acquires images. Artificial Neural Network is used to solve the problem of the input data or learn from the input data, and the expected value. In the system implemented by the author using a back propagation algorithm and combining with mathematical functions to perform in that system. As a result, written by the author's research, it shows that the system can be used to recognize in a different environment. In [22] Priyanka Thakare proposed a method using Eigenface and Principal Component Analysis which has the architecture as the following step. The camera needs to be installed in the front which can capture an entire face of the student inside the class. The first phase after the camera has been captured; the captured image was transferred into the system as an input. The image captured from the camera sometimes comes with darkness or brightness which needs an enhancement on it such as converting to a gray image. The next step, Histogram Normalization, is used in this system to remove the contrast of the image. It is easy to recognize when the student sits in the back row. The Median filter is used to remove noise from the image in case the camera is a high definition camera, but sometimes it still contains the noise. The author also implements skin classification which changes all the pixel to black except the pixel are close to the skin. [1]

Student Attendance System using Face Recognition: Samridhi Dev, Tushar Patnaikb(2020)

In this paper the system was tested on three different algorithms out of which the KNN algorithm proved to be better with the accuracy of 99.27 %. The system was tested on various conditions which include illumination, head movements, expressions, the distance of students from the camera. The system stands up to the expectations even when the image contains faces with beards and spectacles and without beard and spectacles. proposed system evinced to be magnificent to recognize faces having two years of difference. [2]

AUTOMATED SMART ATTENDANCE SYSTEM USING FACE RECOGNITION:

Kolipaka Preethi,swathy vodithala (2021) The proposed method consists of different stages to mark the attendance live A. Face Detection B. DataSet Creation and Training C. Face Recognition and Updating attendance [3] FAREC - CNN Based Efficient Face Recognition Technique using Dlib: Sharma S, Karthikeyan Shanmugasundaram, Sathees Kumar Ramasamy(2016) The paper used trained feature models from Convolutional Neural Network; model has the features of the entire labels of the face recognition systems. The test images are validated against these models and provide the maximum probability value among the labels and claims that to be the person. FAREC takes 20 epoch for converging learning rate from 0.01 and produce 96% accuracy for FRGC and False acceptance rate of 0.1% (1 in 100). The training losses are drastically reduced to 0 very soon as before 5th epoch. The following figure 9 and figure 10 showing the learning rate convergence and accuracy of FAREC.

[4] FaceTime – Deep Learning Based Face Recognition Attendance System: Marko Arsenovic, Srdjan Sladojevic, Andras Anderla, Darko Stefanovic (2017) The model was trained based on a small number of images per employee and using the proposed method of augmentation. This led to the enlargement of the initial dataset and the improvement of the overall accuracy. By analyzing the images stored in the database during the acquisition period, it could be seen that the light conditions influenced the recognition process. Most of the images predicted incorrectly were exposed to the daylight while the door was open. This could potentially be corrected by applying gradient transformation on the images. A small number of images affected by noise of the unknown cause were predicted correctly. The overall accuracy could be improved by applying on time interval automatic re-training of the embedding deep CNN together with the newly gathered images predicted by the model with the high accuracy rate. [5] Real Time Attendance System Using

Face Recognition Technique: Mayank Srivastava, Amit Kumar, Aditya Dixit, Aman Kumar (2020) In this, project experimented with 30 faces as a training set of 7 people for measurement of accuracy of the system. The Extract () function shows a sample binary image obtained with the help of face extracting frame work detection method by Paul – Viola. The results shows that with respect to face detection and recognition rate, on increasing the face angle, camera decreases. Introducing entry and exit times, the authors intend to develop an attendance management system for colleges which is based on facial recognition technology. Every student's attendance is collected by the system through constant observation at the entry and exit points. The results of our initial experiment performed better in performance assessment than traditional black and white display systems. This system is mainly developed for face recognition from images or video frames.

Tripathi et.al [1] claimed a real time system which can follow through the presence of the students in a classroom. The necessary supported images for this model was brought at a constant rate through a webcam until the system is turned off. The author scanned through several techniques in order for face detection and encourage them in recognition. Pupils are distinguished with the help of the Ada boost and Haar cascade classifier. Although for face exposer and recollection, the author made use of OpenCV libraries but still for in depth insight he made a quick use of PCA and LDA. The document also emphasized about the difference between LDA and PCA. In the end author confidently inclined towards the system's accuracy and noted that identification rate is entirely dependent on the database and the size of the used image. Ms. Pooja Humbe et.al [2] made use of 360-degree rotating camera for building the model which detects the pupils in the class. This system without the software such as XAMPP controller, NetBeans, Java Advance for the frontend and back-end with MySQL could have been impossible as stated by author. The characteristics of face are being brought by principal component analysis (PCA). Once registered, the record containing the names of students attended will be sent through email to parents and teachers. Shireesha Chintalapati et.al [3] defined the Viola Jones Face Detection Algorithm. The paper stated that this algorithm offers better results in various lighting conditions and the authors have clubbed multiple Haar classifiers to achieve better output rates up to 30-degree angles. The preprocessing phase relates to the histogram equalization of the facial image

obtained in which it is scaled down to 100x100. Images are converted to grayscale; the equalization of histograms is applied and images are scaled to size of 100x100. The system employed the LBPH algorithm to extract the characteristics and the SVM classifier for classification purpose. This document used a 80- person database (NITW database) with approximately 20 images of each individual collected for the project. This document sets out some performance evaluation conditions when combining LBPH and distance classifier, the false positive rate is 25 %, the object distance for correct recognition must be 4 feet, the training time being 563 milliseconds, 95 %of recognition percentage for static images, the recognition percentage (real-time video) was 78 %, the occluded faces 2.3% In Microsoft Visual C #and the EmguCV container the GUI is developed using the WinForms application. E. Varadharajan et.al [4] explained the automatic Attendance Management system based on Face Detection. The author describes how faces are sensed and then cut, before which background subtraction is performed on the image in order to improve system performance efficacy. The erudite authors recommend the use of Eigen face for its simplicity and quality of performance in facial recognition. The document also concluded with the observation that in the case of women, the detection and recognition rate of the face with a veil was 45% and 10%, while in the case of women it was 93% and 87% without the veil. The identification and recognition levels, on the other hand, were 79% and 65% for bearded men. Akshara Jadhav et.al [5] prompted face encounter algorithm Viola Jones and face recognition PCA algorithm with support for machine learning and SVM for extraction functionality. The author also incorporated reprocessing which includes the histogram equalization of the facial image extracted and is scaled to 100x100. The use of neural networks for facial recognition has been shown, and we can see the possibility of a semi-supervised learning approach that uses facial recognition support vector machines for satisfactory results. The process followed after the face is recognized is the subsequent processing in which attendance is generated weekly or monthly and can be sent to parents or guardians. Nirmalya Kar et.al [6] used Haar cascade front XML file for pinpointing a face and confirmation of faces using Eigen face. It was created using Open-CV Libraries. On the end of facial orientation, the test was prepared. Both detection and recognition levels were high when facial orientation was approximately 0 degrees with 98.7% and 95% respectively. The frequency decreased slowly as facial orientation rose from 0 degrees to 90 degrees. In

the end, the identification and recognition levels ranged from 0 to 90 degrees. Smit Hapani et.al [7] has magnified the system which approbated the model which contributes face distinguishing. Haar classifiers which uses cascade approach and followed by recognition which uses Fisher face. The system optimally offers efficacy up to 50% within 15 pupils when modelling with more than one face with respective to variations such as cap, spectacles. The proposed system makes use of classroom through video source, and these resulting frames are used to identify the faces. Thus, by following the procedures there by increasing the rate and accuracy of overall model. Krishna Dharavath et.al [8] has produced excellent preprocessing results on a noisy image. The methods suggested for pre-processing are face cropping, resizing, normalizing & filtering. A low pass filter is used to eliminate components of high frequency noise. PCA, DCT (Discrete Cosine Transform) and combined Spatial and Frequency Domain approach are compared before and after pre-processing. The proposed combined form has the highest rate of face recognition and is not much influenced by pre-processing. The major drawback is that facial detection is performed before the preprocessing of image. In multiple face recognition system, this is not expected as the image needs to be pre-processed first before any face detection or recognition. Priyanka Wagh et.al [9] multiple face identification system has been expressed using Viola Jones for face detection purposes and the Eigen face for face recognition. It described the face identification as an invariant of illumination, as it is a combined form of both Eigen face and PCA. While the face recognition rate as in the classroom is not established at a longer distance, the varying lighting conditions do not impact multiple face recognition. Nazare Kanchan Jayant et.al [10] executed an automatic attendance system. This system is based on the Viola Jones facial detection and face recognition algorithm. First the 20 student's database is created using various head poses for culminated recognition results. The face finding algorithm was then applied, and its efficiency was determined depending on the number of faces detected. The same process is followed for calculating the facial recognition algorithm's efficiency. Firoz Mahmud et.al [11] approbated use of 2 database types including UMIST database and ORL database. PCA and LDA both are used for face knowing purposes. The accuracy of the face recognition is determined using the above listed algorithms, depending on the face alignment. It is observed that front aligned faces have a much better accuracy of recognition than those of face side alignment. Refik Samet et.al

[12] has implemented a fully cell phone automatic attendance system. This is achieved using the Viola-Jones algorithm along with Ada-boost training for face finding, since according to the authors, they should work better in the real-life scenario. For the purposes of recognition, the Euclidean distance was determined for the 3 recognition methods, namely its Eigen face, Fisher face and LBP. A comparison of precision was made for all of the above-mentioned recognition techniques. The smartphone application was developed for the automatic attendance generating system. Sathyanarayana n et.al [13] launched Automated Attendance system using facial recognition. The system specifies algorithms such as Jones' Purple algorithm for face detection and MSE (medium square error) face recognition. The document stated and elaborated about the system's level of security and accuracy improves as the number of training images increases. The machine is also checked for different face angles and alignment up to 60 degrees can be identified. It is observed that when the system is tested with an image of six students, the system recognizes five students with 70% efficiency. D. Nithya et.al [14] has introduced Automated Attendance System which works on MATLAB. Extraction of the functions is accomplished by analysing the main components. The Eigen facial approach is utilized for its ease, speed and learning ability. The difference between the training values and the test image is calculated using the Euclidean distance. Rajashree P. Suryawanshi et.al [15] prescribed the system with hardware such as Raspberry PI and a wired camera, but the software also consisted of using Open-CV. The very first step in facial recognition is to detect a face in a given image, and afterwards proceed with the recognition only if there is a face there. The face pinning was performed using Haar Cascade Classifier and Face Recognition was based on PCA. K.L.P.M Liyanage et.al [16] prompted system having a separate application and a web-based application. The independent application deals with the process of facial recognition and the process of marking the attendance. The Web-based application mainly deals with the NLP process. Both applications link to a centralized database. Face detection is achieved using the Haar cascade method while face recognition is carried out using the PCM method. NLP is the other research framework developed in SMRT-FR for the processing and management of applications for employee licenses. Employees can easily request authorizations by sending an SMS or using the web interface and these requests for authorization are processed using the NLP application and the result of acceptance or refusal is generated in the light of

different conditions and rules. The system has been able to detect faces with 68% accuracy so far. Professor Arun Katara et.al [17] implemented a real time assistance system which can perform multiple facial recognition using the Raspberry PI model and Raspberry PI camera. For face pin-pointing it uses Open-CV libraries and for face recognition the combination of feature extraction methods such as principal component analysis along with LBP is implemented. Since the system can identify faces from a distance of 4 feet to 7 feet, the facial recognition efficacy is limited, and is suggested to be improved. Capture a video of classroom using a video camera and followed by processing images for facial recognition. Kennedy Okopujie et.al [18] describes a system that uses Viola Jones as a face detection tool and Fisher face algorithm for face recognition. Uses a webcam to build the database and to collect photos to process. It works well in good lighting conditions, but at different lighting conditions it decreases the face recognition rate (up to 54%). The system has access for the authority and the participants via the cell phone interface with the login credentials. Nilesh D. Veer et.al [19] an automatic attendance system has been developed in which a video is collected as input. frames are captured when there is human presence detected. For face detection, Viola Jones is used, and PCA is used for face recognition, which also uses LBP for threshold purposes. The facial recognition rate is nearly 100% for a small number of students and the attendance of the student is recorded along with the entry time of the student. A. Majumdar et.al [20] discussed how well they had done than PCA. To boost dispersion, they used the Fisher face subspace and LDA, and they also used KNN. Consequently, better results were obtained by using the PseudoFisher facial technique. This article examines several methods that various authors consider to improve the rate of detection and recognition. The results show that Viola Jones, who uses Haar Cascade, is consistent in all the papers reviewed and offers a good detection rate whereas Fisher Face's LDA algorithm provides better performance and faster results.

3. HARDWARE DESIGN

3.1 INTRODUCTION

The hardware design of the project includes the following components:

1. Webcam: A webcam is required for capturing images and videos of students in order to perform facial recognition and attendance tracking.
2. Raspberry Pi 4B: The Raspberry Pi is a single-board computer that is used as the main processing unit for the project. It is responsible for running the facial recognition algorithm, storing the attendance data, and interfacing with other components.
3. 3D Printed Face Mask: The 3D printed face mask is designed to cover the user's face and includes OLED displays as eyes. The mask is connected to an Arduino, which sends signals to the OLED displays to blink accordingly whenever it talks.
4. Speakers: The speakers are connected to the 3D printed face mask and are used to produce sound whenever the mask talks.
5. Arduino: The Arduino is used to interface with the 3D printed face mask and the speakers. It receives signals from the Raspberry Pi and sends signals to the OLED displays and speakers accordingly.
6. Miscellaneous Components: The project may also require additional components such as cables, connectors, breadboards, and power supplies, depending on the specific design.

The above components can be connected and integrated to form a complete hardware system for your project. The Raspberry Pi can be connected to the webcam and the Arduino, and the attendance data can be stored in a JSON file on the Raspberry Pi. Overall, this hardware design allows for the capture of facial recognition data, attendance tracking, and the display of information through the 3D printed face mask and speakers.

3.2 WEBCAM

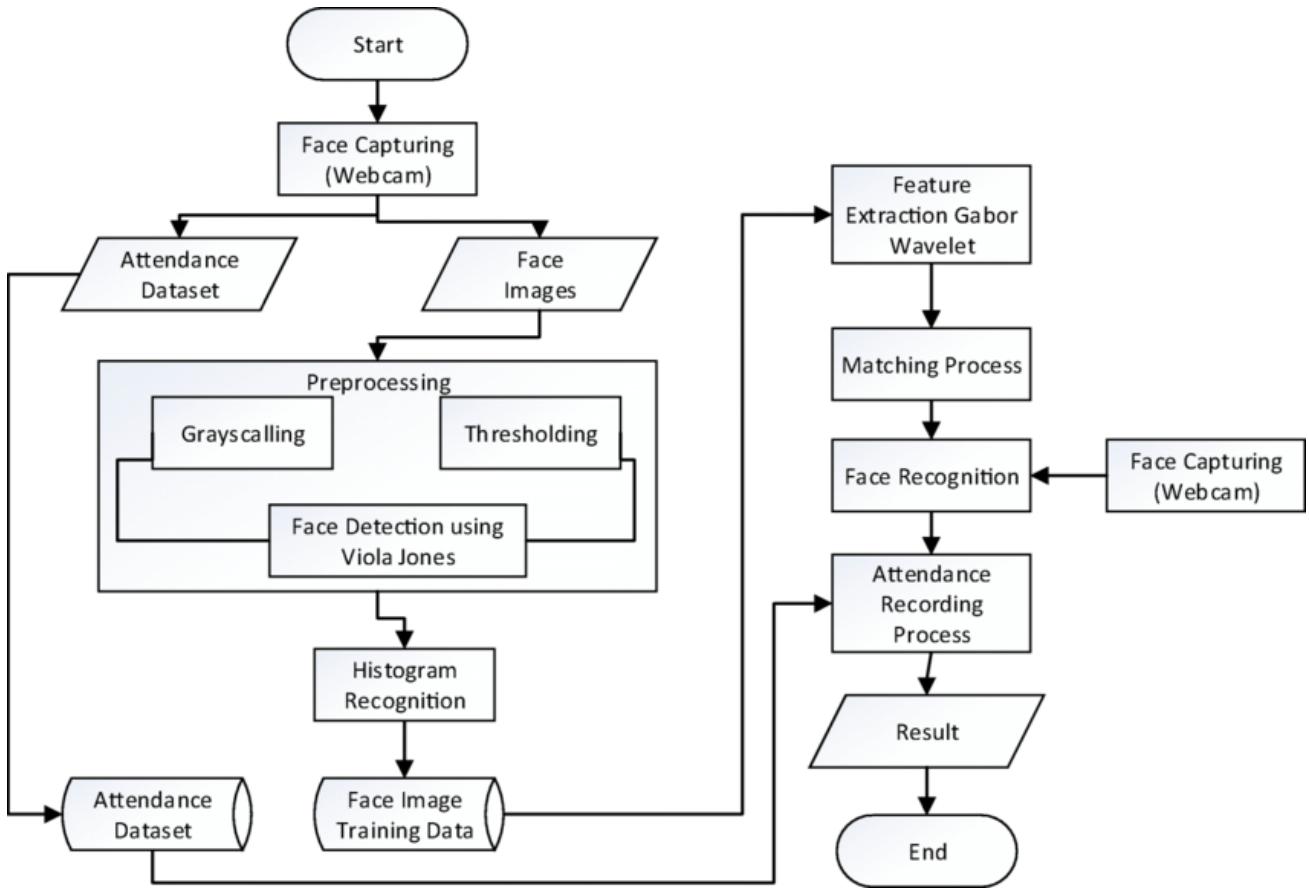
The webcam plays a crucial role in capturing images and videos of students for facial recognition and attendance tracking. The webcam captures images of the students' faces, which are then processed by the facial recognition algorithm to identify them. This allows the system to keep track of the students' attendance and generate reports.

The webcam is connected to the Raspberry Pi, which serves as the main processing unit for the project. The Raspberry Pi runs the facial recognition algorithm, which compares the captured images to the images stored in the system's database. When a match is found, the system records the attendance and updates the student's attendance record.

The webcam used in the project should have sufficient resolution and image quality to ensure accurate facial recognition. It should also have a wide-angle lens to capture the faces of all students in the frame. Additionally, it is important to ensure that the webcam is securely mounted and positioned to capture clear images of the students' faces.



3.3 Flowchart :



3.4 RASPBERRY PI 4B:

3.4.1 Introduction:

The Raspberry Pi 4B is a powerful and versatile single-board computer that is commonly used in a variety of projects due to its small size and low power consumption. In your project, the Raspberry Pi 4B serves as the main processing unit that handles the facial recognition algorithm and stores the attendance data.

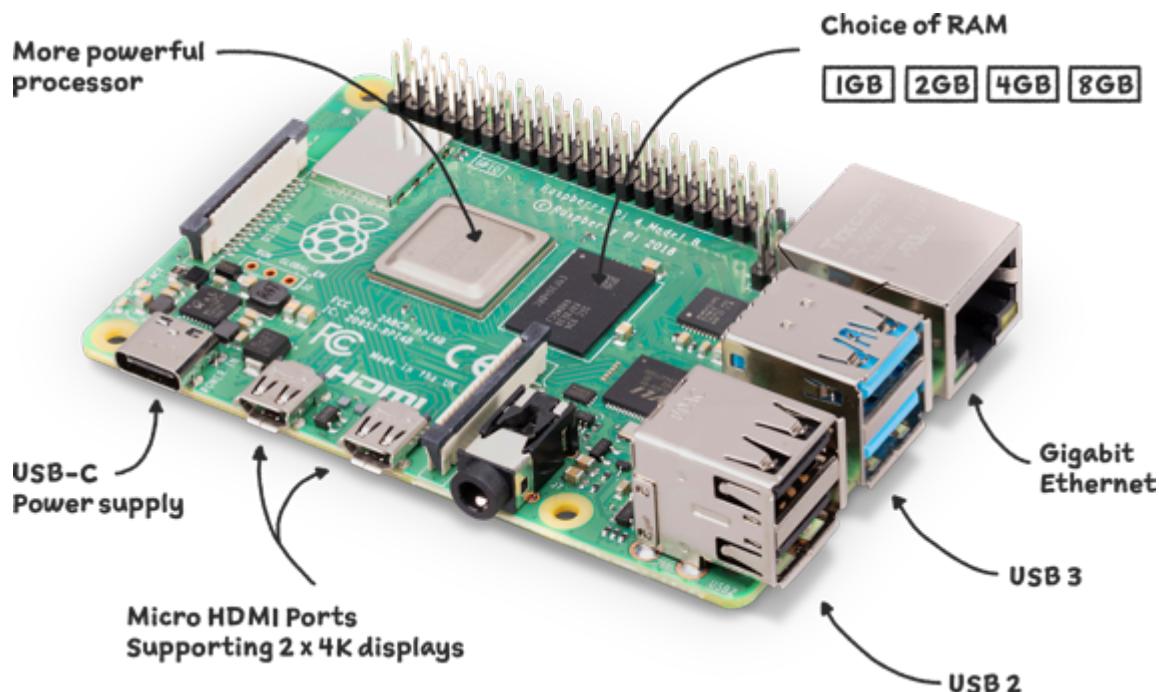
The Raspberry Pi 4B is equipped with a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC, which provides a significant improvement in performance compared to previous Raspberry Pi models. It has 2GB, 4GB, or 8GB of LPDDR4-3200 SDRAM (depending on the model), which allows it to handle multiple tasks and run complex programs.

In addition, the Raspberry Pi 4B has a variety of connectivity options, including Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, and BLE (Bluetooth Low Energy). It also has two USB 3.0 ports, two USB 2.0 ports, and two micro-HDMI ports that can support up to two 4K displays.

The Raspberry Pi 4B runs on a variety of operating systems, including Raspberry Pi OS (formerly known as Raspbian), Ubuntu, and other Linux-based distributions. This

makes it easy to customize and configure the system according to the project's requirements.

In your project, the Raspberry Pi 4B is used to control the system's hardware components, including the webcam and OLED display on the face mask. It also runs the facial recognition algorithm, which compares the captured images to the images stored in the system's database to identify the students and record their attendance. The attendance data is stored on the Raspberry Pi 4B in a JSON file, which can be easily accessed and processed.



3.4.2 CHIP DESCRIPTION :

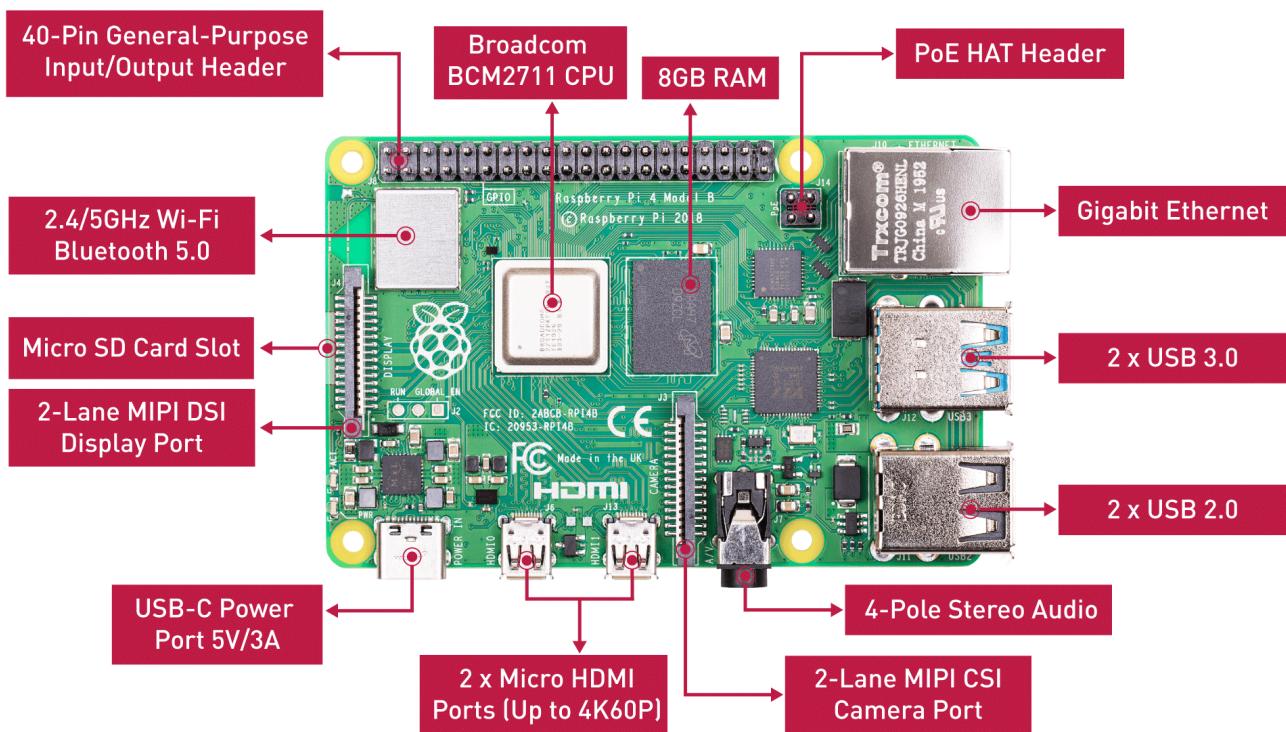
The Raspberry Pi 4B is powered by the Broadcom BCM2711 SoC (System on a Chip), which is a powerful quad-core ARM Cortex-A72 CPU clocked at 1.5GHz. The SoC also includes a Broadcom VideoCore VI GPU with support for OpenGL ES 3.x graphics, and hardware decoding for H.265 (4Kp60 decode), H.264 (1080p60 decode, 1080p30 encode), and MPEG-4 AVC (1080p30 decode) video formats.

In addition, the Raspberry Pi 4B features LPDDR4-3200 SDRAM in 2GB, 4GB, or 8GB configurations, depending on the model. The RAM is shared between the CPU and GPU, providing fast access to data and reducing latency. The SoC also includes a 1Gbit Ethernet controller, as well as support for dual-band 802.11ac wireless and Bluetooth 5.0.

The Raspberry Pi 4B has a variety of input/output (I/O) options, including two USB 3.0 ports, two USB 2.0 ports, two micro-HDMI ports (up to 4Kp60 supported), a 3.5mm audio jack, and a 40-pin GPIO header that provides access to various interfaces such as UART, I2C, SPI, and GPIO. It also has a microSD card slot for loading the operating system and storing data.

The Raspberry Pi 4B is designed to be a low-power device, consuming approximately 7.5W under typical load conditions. It is powered by a USB-C connector, which provides up to 15W of power for peripherals.

Overall, the Raspberry Pi 4B is a versatile and powerful single-board computer that provides a wide range of capabilities and I/O options, making it an ideal choice for a variety of projects.



3.4.3 PIN DESCRIPTION:

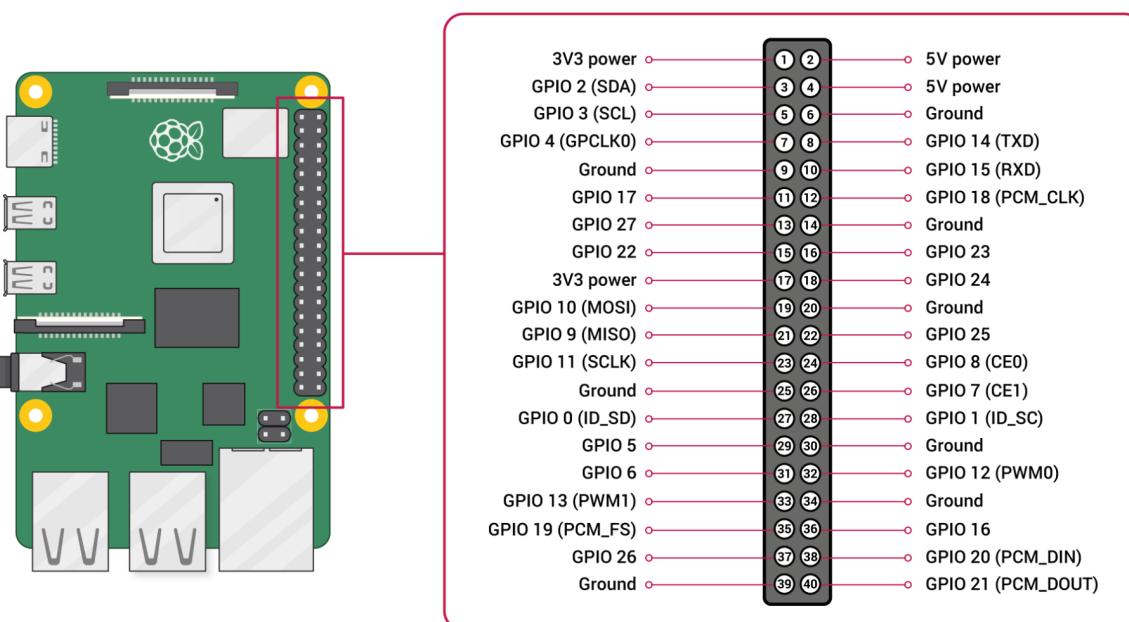
The Raspberry Pi 4B has a 40-pin GPIO (General Purpose Input/Output) header that provides access to a wide range of interfaces and peripherals. Here is a brief overview of the pinout:

- Pin 1 (3.3V) - power supply voltage (max 50mA)
- Pin 2 (5V) - power supply voltage (max 3A)

- Pin 3 (SDA1) - I2C data line 1
- Pin 4 (5V) - power supply voltage (max 3A)
- Pin 5 (SCL1) - I2C clock line 1
- Pin 6 (Ground) - ground
- Pin 7 (GPIO4) - general purpose input/output pin
- Pin 8 (UART0_TXD) - UART transmit data
- Pin 9 (Ground) - ground
- Pin 10 (UART0_RXD) - UART receive data
- Pin 11 (GPIO17) - general purpose input/output pin
- Pin 12 (GPIO18) - general purpose input/output pin (PWM)
- Pin 13 (GPIO27) - general purpose input/output pin
- Pin 14 (Ground) - ground
- Pin 15 (GPIO22) - general purpose input/output pin
- Pin 16 (GPIO23) - general purpose input/output pin
- Pin 17 (3.3V) - power supply voltage (max 50mA)
- Pin 18 (GPIO24) - general purpose input/output pin
- Pin 19 (SPI_MOSI) - SPI master output, slave input
- Pin 20 (Ground) - ground
- Pin 21 (SPI_MISO) - SPI master input, slave output
- Pin 22 (GPIO25) - general purpose input/output pin
- Pin 23 (SPI_SCLK) - SPI clock
- Pin 24 (SPI_CE0_N) - SPI chip select 0
- Pin 25 (Ground) - ground
- Pin 26 (SPI_CE1_N) - SPI chip select 1
- Pin 27 (ID_SD) - I2C ID EEPROM
- Pin 28 (ID_SC) - I2C ID EEPROM
- Pin 29 (GPIO5) - general purpose input/output pin

- Pin 30 (Ground) - ground
- Pin 31 (GPIO6) - general purpose input/output pin
- Pin 32 (GPIO12) - general purpose input/output pin (PWM)
- Pin 33 (GPIO13) - general purpose input/output pin (PWM)
- Pin 34 (Ground) - ground
- Pin 35 (GPIO19) - general purpose input/output pin (PWM)
- Pin 36 (GPIO16) - general purpose input/output pin
- Pin 37 (GPIO26) - general purpose input/output pin
- Pin 38 (GPIO20) - general purpose input/output pin
- Pin 39 (Ground) - ground
- Pin 40 (GPIO21) - general purpose input/output pin

The pinout provides access to a variety of interfaces, including I2C, SPI, UART, PWM, and general-purpose input/output pins. By connecting various sensors, actuators, and other components to the GPIO pins, you can create a wide range of projects with the Raspberry Pi 4B.



3.4.4 SPECIFICATION :

The Raspberry Pi 4B is a small single-board computer with the following specifications:

- Processor: Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Memory: 1GB, 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM (depending on model)
- Connectivity: Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, BLE
- Video and audio output: 2 × micro-HDMI ports (up to 4Kp60 supported), 2-lane MIPI DSI display port, 2-lane MIPI CSI camera port, 4-pole stereo audio and composite video port
- USB: 2 × USB 3.0 ports, 2 × USB 2.0 ports
- GPIO: Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
- Power: 5V DC via USB-C connector (minimum 3A), or via GPIO header (minimum 3A)
- Dimensions: 88 × 58 × 19.5 mm, 46 g

The Raspberry Pi 4B is a significant improvement over its predecessors, with a faster processor, more memory, and improved connectivity options. Its powerful processor and ample memory make it well-suited for a wide range of applications, including media centers, web servers, and IoT projects. The GPIO header and camera port make it easy to interface with a wide range of hardware peripherals, while the dual micro-HDMI ports provide convenient video output options. Its compact size and low power consumption also make it an attractive option for embedded applications.

3.4.5 APPLICATION :

The Raspberry Pi 4B has a wide range of applications in various fields, including:

1. Education: The Raspberry Pi 4B is an excellent platform for teaching computer science, programming, and electronics to students of all ages. It is an affordable

and versatile tool for teaching basic programming concepts and can be used to build a variety of projects.

2. Home automation: The Raspberry Pi 4B can be used to create smart home systems for controlling lighting, temperature, security, and other home appliances.
3. Media centers: With its powerful processor and support for video output, the Raspberry Pi 4B is a great choice for building media centers for streaming movies, music, and TV shows.
4. Internet of Things (IoT): The Raspberry Pi 4B can be used as a hub for IoT projects, collecting data from sensors and controlling actuators.
5. Robotics: With its GPIO header and camera port, the Raspberry Pi 4B is a popular choice for building robots and other DIY projects.
6. Gaming: The Raspberry Pi 4B can be used to build retro gaming consoles, as well as to play modern games with the help of emulation software.
7. Server hosting: The Raspberry Pi 4B can be used as a low-power server for hosting web applications, email servers, and other services.
8. Research: The Raspberry Pi 4B can be used in research projects that require low-cost computing and data collection.

Overall, the Raspberry Pi 4B is a versatile platform that can be used for a wide range of applications, from educational to commercial, from entertainment to research, and from home automation to robotics.

3.5 3D PRINTED FACE MASK :

A 3D printed face mask is a type of mask that is designed and produced using 3D printing technology. It is created using a digital design file that is sliced into layers and then printed layer by layer to create a physical object. 3D printed face masks can be customized to fit the face of the wearer, providing a more comfortable and secure fit than traditional masks.

In the context of this project, a 3D printed face mask was designed and produced to incorporate OLED displays as its eyes, which could blink according to the signals sent from an Arduino. The mask was designed to be worn by the user during the attendance recording process, adding a fun and interactive element to the process.



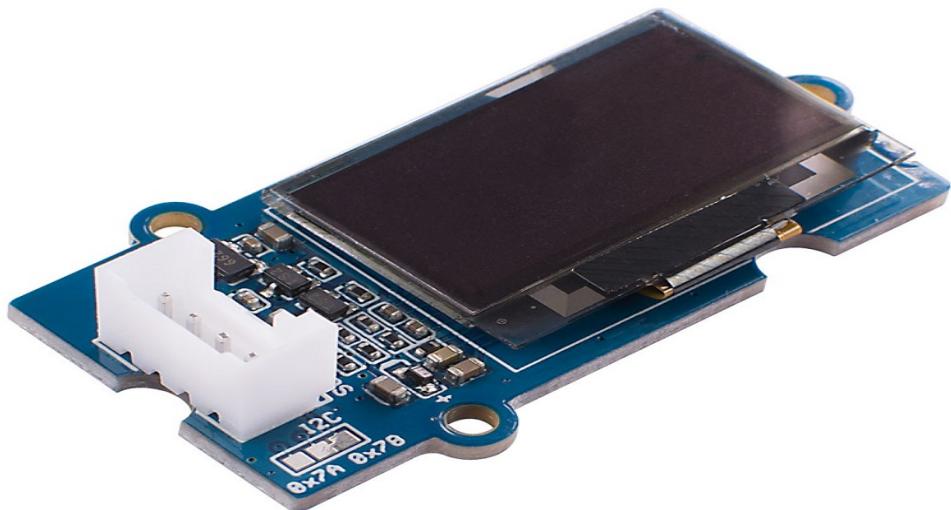
3.6 OLED DISPLAY :

OLED displays were used to serve as the eyes of the 3D printed face mask, adding a fun and interactive element to the attendance recording process. OLED stands for Organic Light Emitting Diode, which is a type of display technology that uses organic compounds to emit light when an electrical current is applied. OLED displays are known for their high contrast, vibrant colors, and low power consumption. They are also extremely thin and flexible, making them ideal for use in wearable technology and other compact applications.

In the context of this project, the OLED displays were programmed to blink in response to signals sent from an Arduino, adding a unique and engaging element to the attendance recording process. The displays were integrated into the 3D printed face mask, serving as the eyes of the mask.

OLED displays have found various applications in the electronics industry, including in smartphones, televisions, and wearable technology. They offer a high level of flexibility and customization, making them ideal for use in applications where space and power consumption are important factors.

Overall, the use of OLED displays in this project added a unique and engaging element to the attendance recording process, showcasing the potential of this display technology in a range of different applications.



3.7 ARDUINO UNO :

3.7.1 INTRODUCTION :

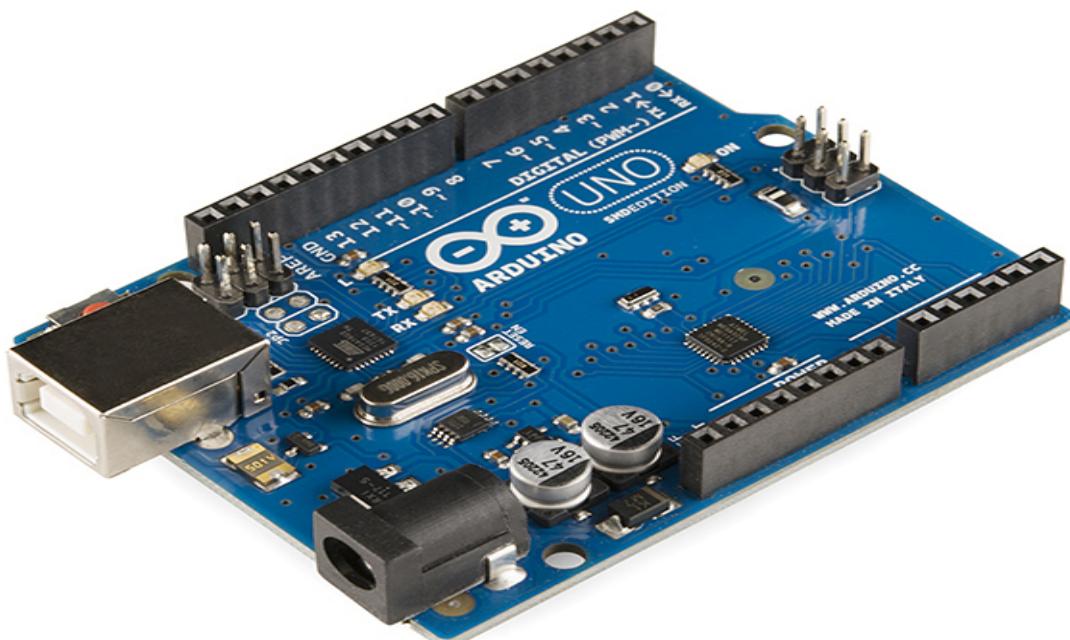
In this project, an Arduino Uno was used to send signals to the OLED displays on the 3D printed face mask. The Arduino Uno is a popular microcontroller board that is widely used in the electronics industry for its ease of use, flexibility, and compatibility with a wide range of sensors and other components.

The Arduino Uno is based on the ATmega328P microcontroller, which has 32KB of flash memory for storing program code and 2KB of RAM for data storage. It has 14 digital input/output pins, 6 analog input pins, and a variety of other features such as PWM (Pulse Width Modulation) output, interrupts, and serial communication ports.

In the context of this project, the Arduino Uno was programmed to send signals to the OLED displays on the 3D printed face mask in response to certain events, such as when the attendance was recorded or when a command was received from the Raspberry Pi. The signals were sent using the I2C (Inter-Integrated Circuit) protocol, which is a popular communication protocol used for connecting microcontrollers and other devices.

The Arduino Uno has a wide range of applications in the electronics industry, including robotics, automation, and sensor monitoring. It is a popular choice for hobbyists and professionals alike due to its ease of use and compatibility with a wide range of components and sensors.

Overall, the use of the Arduino Uno in this project demonstrated its versatility and flexibility, showcasing the potential of this popular microcontroller board in a range of different applications.



3.7.2 : ATMega328P

ATMega328P is a popular microcontroller chip that is widely used in various electronic devices due to its low cost, low power consumption, and high-performance characteristics. It is manufactured by Atmel Corporation, which is now a part of Microchip Technology Inc. The chip belongs to the AVR family of microcontrollers and is based on the Harvard architecture.

The ATMega328P chip has 32KB of flash memory, 1KB of EEPROM, and 2KB of SRAM. It has a clock speed of up to 20MHz and supports various communication protocols such as I2C, SPI, and UART. The chip also has 23 general-purpose

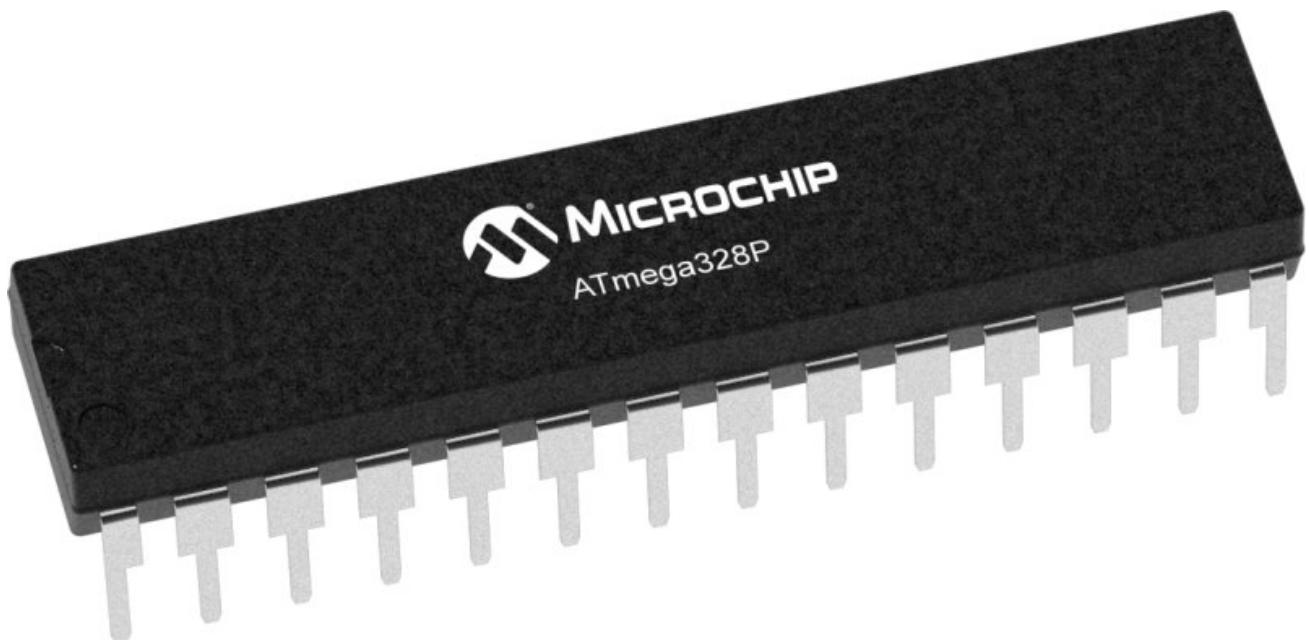
input/output (GPIO) pins, which can be used for various purposes such as digital input/output, analog input, PWM output, and more.

One of the key features of the ATMega328P chip is its built-in bootloader. The bootloader is a small piece of software that runs when the chip is powered on, and it allows the chip to be programmed over a serial interface without the need for an external programmer. This feature makes it easy to program and update the firmware of devices that use the ATMega328P chip.

The ATMega328P chip is used in a wide range of applications, from simple hobby projects to professional electronic devices. It is commonly used in microcontroller development boards such as the Arduino UNO and Nano, and it is also used in various sensors, motor controllers, and other electronic devices.

In addition to its low cost and high-performance characteristics, the ATMega328P chip also has a large and active community of developers and users. This community has created a vast library of code and resources that can be used to easily develop and implement various electronic projects using the ATMega328P chip.

Overall, the ATMega328P chip is a versatile and powerful microcontroller that is widely used in the electronics industry. Its low cost, low power consumption, and built-in bootloader make it an attractive choice for a wide range of applications, from simple hobby projects to professional electronic devices.



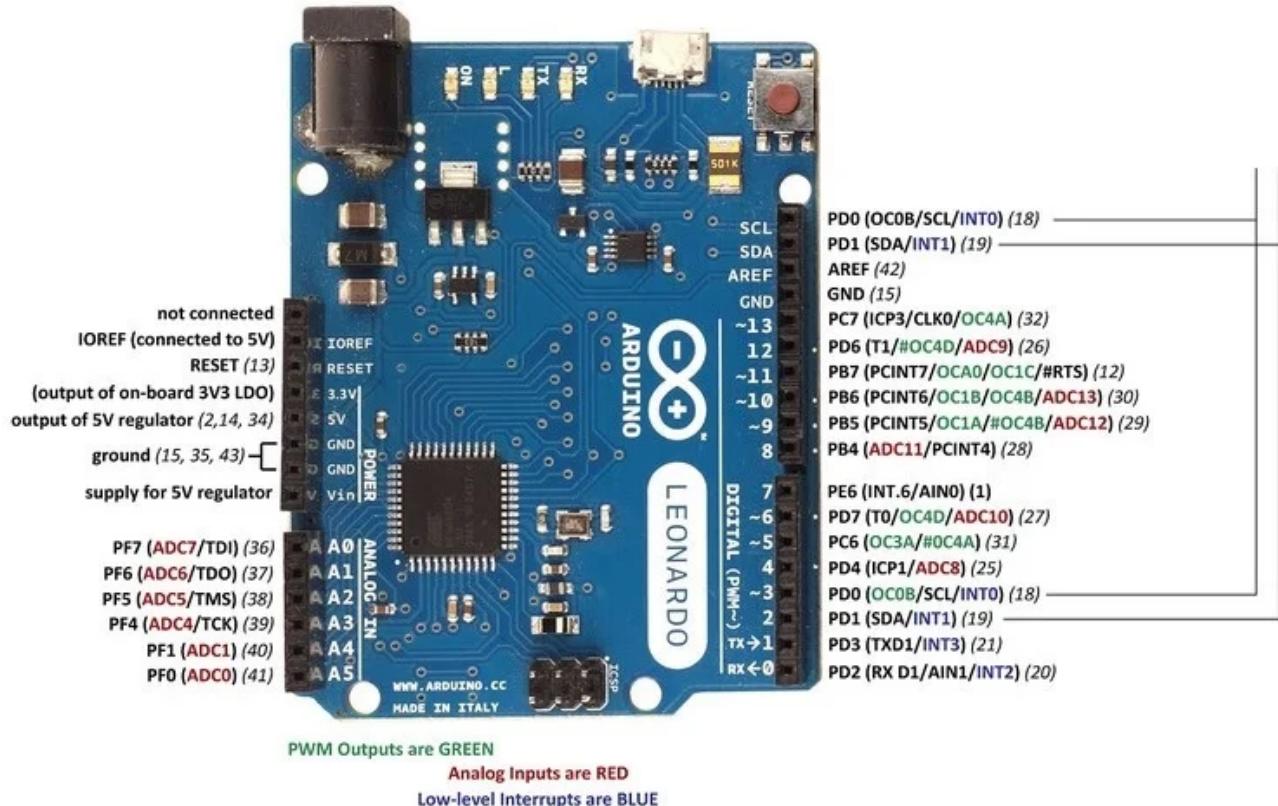
3.7.3 : ARDUINO UNO PIN DESCRIPTION

The Arduino Uno is a microcontroller board based on the ATmega328P chip. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, and an ICSP header. Here is a brief description of the pinouts of Arduino Uno:

1. Digital Pins: There are 14 digital pins numbered 0 to 13. They can be used as either input or output pins. They operate at 5 volts and can handle up to 20mA of current.
2. Analog Pins: The Arduino Uno has 6 analog input pins that can read values from 0 to 5 volts. These pins are labeled A0 to A5 and can also be used as digital pins.
3. Power Pins: The board has two power pins – 5V and 3.3V. The 5V pin is used to power the board, while the 3.3V pin is used to power external devices connected to the board.
4. Ground Pins: There are several ground pins on the board that are connected to the ground plane of the circuit. These pins are labeled GND.
5. Reset Pin: The reset pin is used to reset the microcontroller. It is labeled RESET and can be found near the USB connector.
6. ICSP Header: The ICSP (In-Circuit Serial Programming) header is used for programming the board with an external programmer. It consists of six pins – MOSI, MISO, SCK, RESET, VCC, and GND.
7. PWM Pins: The Arduino Uno has 6 pins that support Pulse Width Modulation (PWM) – 3, 5, 6, 9, 10, and 11. These pins can be used to control the brightness of LEDs or the speed of motors.
8. Serial Pins: The board has two pins – TX and RX – that are used for serial communication. They are connected to the ATmega16U2 USB-to-serial converter chip.
9. AREF Pin: The AREF pin is used to provide a reference voltage for the analog inputs. It can be connected to an external reference voltage if necessary.

Understanding the pinouts of the Arduino Uno is essential when designing a project or interfacing with external devices. Each pin has a specific function, and it is important

to use the correct pins for each task. The pin descriptions above provide a basic overview of the Uno's pins and their functions.



3.7.4 : ARDUINO UNO SPECIFICATION

The Arduino Uno is a microcontroller board that is based on the ATmega328P microcontroller chip. It is one of the most popular microcontroller boards for hobbyists and professionals alike due to its affordability, versatility, and ease of use. Here are the specifications of the Arduino Uno:

1. Microcontroller: ATmega328P

The ATmega328P is a high-performance, low-power microcontroller that runs at a clock speed of 16 MHz. It has 32 KB of flash memory for storing programs and 2 KB of SRAM for storing variables. It also has 1 KB of EEPROM memory for storing data that needs to be retained even when power is turned off.

2. Digital I/O Pins: 14

The Arduino Uno has 14 digital input/output pins, each of which can be configured as either an input or an output. These pins can be used for connecting sensors, actuators, and other digital devices.

3. Analog Input Pins: 6

The Arduino Uno has 6 analog input pins, which can be used for reading analog signals from sensors such as temperature sensors, light sensors, and potentiometers.

4. PWM Output Pins: 6

The Arduino Uno has 6 PWM (Pulse Width Modulation) output pins, which can be used for controlling the speed of motors, the brightness of LEDs, and other applications that require a variable output.

5. Operating Voltage: 5V

The Arduino Uno is designed to operate at a voltage of 5V. It can be powered either by a USB cable or by an external power supply.

6. Input Voltage (recommended): 7-12V

The recommended input voltage for the Arduino Uno is between 7V and 12V. If the input voltage is outside this range, the board may not function correctly.

7. Input Voltage (limits): 6-20V

The input voltage limits for the Arduino Uno are between 6V and 20V. However, it is recommended to stay within the recommended range to ensure proper operation.

8. Digital I/O Pin Current: 20 mA

Each digital I/O pin of the Arduino Uno can source or sink a maximum current of 20 mA.

9. DC Current per I/O Pin: 40 mA

The total DC current that can be sourced or sunk by all digital I/O pins of the Arduino Uno combined is 200 mA.

10. DC Current for 3.3V Pin: 50 mA

The 3.3V pin on the Arduino Uno can source a maximum current of 50 mA.

11. Flash Memory: 32 KB

The Arduino Uno has 32 KB of flash memory for storing programs.

12. SRAM: 2 KB

The Arduino Uno has 2 KB of SRAM for storing variables.

13.EEPROM: 1 KB

The Arduino Uno has 1 KB of EEPROM memory for storing data that needs to be retained even when power is turned off.

14.Clock Speed: 16 MHz

The Arduino Uno runs at a clock speed of 16 MHz.

Overall, the Arduino Uno is a powerful microcontroller board that is well-suited for a wide range of applications. Its ease of use and versatility make it a popular choice for hobbyists and professionals alike.

| Microcontroller | ATmega328 |
|--|---|
| Clock Speed | 16MHz |
| Operating Voltage | 5V |
| Maximum supply Voltage (not recommended) | 20V |
| Supply Voltage (recommended) | 7-12V |
| Analog Input Pins | 6 |
| Digital Input/Output Pins | 14 |
| DC Current per Input/Output Pin | 40mA |
| DC Current in 3.3V Pin | 50mA |
| SRAM | 2KB |
| EEPROM | 1KB |
| Flash Memory | 32KB of which 0.5KB used by boot loader |

3.7.4 : ARDUINO UNO APPLICATIONS

Arduino Uno is a popular microcontroller board used by hobbyists, students, and professionals alike due to its ease of use and flexibility. Some of the applications of Arduino Uno are:

1. Home automation: Arduino Uno can be used to control and automate various home appliances such as lights, fans, heaters, and air conditioners. This can be achieved by interfacing the board with different sensors and actuators.
2. Robotics: Arduino Uno can be used to control the movements of robots, as well as interface with sensors and actuators required for various robotic applications.
3. Data logging: Arduino Uno can be used to log data from various sensors such as temperature, humidity, and pressure sensors. This data can be stored in the

onboard memory or transmitted wirelessly to a computer or other devices for further analysis.

4. Internet of Things (IoT) applications: Arduino Uno can be used to build IoT devices that connect to the internet and can communicate with other devices. With the help of Wi-Fi or Ethernet shields, the board can be connected to the internet and used to build smart home devices, environmental monitoring systems, and many other applications.
5. Audio and music applications: Arduino Uno can be used to create music synthesizers, MIDI controllers, and other audio-related projects. With the help of shields and add-on boards, the board can interface with various audio components such as microphones, speakers, and amplifiers.
6. Education: Arduino Uno is an ideal tool for teaching electronics and programming to students due to its simplicity and ease of use. Many educational institutions use Arduino Uno in their courses to teach programming and electronics concepts to students.

Overall, Arduino Uno is a versatile microcontroller board that can be used in a wide range of applications, from simple hobby projects to complex industrial automation systems. Its flexibility, ease of use, and wide range of shields and add-on boards make it a popular choice among electronics enthusiasts and professionals alike.

4. FACIAL RECOGNITION

4.1 INTRODUCTION :

Facial recognition is a biometric technology that identifies individuals based on their unique facial features. It has gained immense popularity over the years due to its convenience, accuracy, and the wide range of applications it offers. Facial recognition technology works by capturing an image or a video of a person's face and then processing it to extract unique features like the distance between the eyes, the shape of the nose, and the curve of the lips. These features are then compared to a pre-existing database of facial images to identify the person.

Facial recognition technology is used in a variety of applications, including security, law enforcement, marketing, and personal identification. In the security sector, it is used for access control, surveillance, and to track individuals of interest. Law enforcement agencies use facial recognition technology to identify suspects and track down criminals. In the marketing sector, facial recognition technology is used for targeted advertising, analyzing customer behavior, and to personalize shopping experiences. Personal identification applications include unlocking mobile devices, and authorizing financial transactions.

Despite the numerous benefits of facial recognition technology, it has also faced criticism over issues such as privacy concerns, biases, and accuracy. The technology has been known to produce false positives and false negatives, particularly with people of color or those with non-traditional facial features.

Overall, facial recognition technology continues to be a fast-growing field with many potential applications, and it is constantly evolving with advancements in computer vision and artificial intelligence.

4.2 HISTORY :

Facial recognition technology has a relatively long history dating back to the 1960s. The first experiments in this field were conducted by Woodrow Wilson Bledsoe and his team at the Panoramic Research facility at Stanford University. They created a system that could recognize faces from different angles, which was a significant breakthrough at the time. However, due to limited computing power and the high cost of the required hardware, facial recognition technology remained experimental and impractical for many years.

In the 1990s, facial recognition technology began to attract more attention due to the increasing availability of powerful computers and the development of sophisticated algorithms. This led to the creation of several commercial facial recognition systems, which were primarily used for security and surveillance applications.

Since then, facial recognition technology has continued to improve and become more accessible. Today, it is used in a wide range of applications, from unlocking smartphones to tracking attendance in classrooms. As the technology continues to advance, it is expected to become even more ubiquitous in our daily lives. However, concerns about privacy and the potential for misuse of the technology have also emerged, leading to calls for greater regulation and oversight.

4.3 SOME COMMON FACIAL RECOGNITION TECHNIQUES :

There are several common facial recognition techniques used in the industry. Some of them are:

1. Eigenface: It uses the principal component analysis (PCA) technique to extract the features from the face images. The extracted features are then used to compare the faces.
2. Local Binary Patterns (LBP): LBP is a texture descriptor that extracts the texture information from an image. LBP technique uses the gray-scale information of an image to extract the features.
3. 3D Facial Recognition: It captures the facial features in 3D using the 3D sensors or multiple cameras. It provides a higher accuracy than the 2D facial recognition.
4. Deep Learning: It is the most advanced technique for facial recognition. It uses a neural network to extract the features from the face images. The neural network is trained on a large dataset to recognize the faces.

These techniques can be combined to create a more accurate facial recognition system. However, the choice of technique depends on the application and the accuracy required.

4.4 BENEFITS AND SHORTCOMINGS OF FACIAL RECOGNITION :

Facial recognition technology has many potential benefits, including improving security and efficiency in various industries. Some of the benefits are:

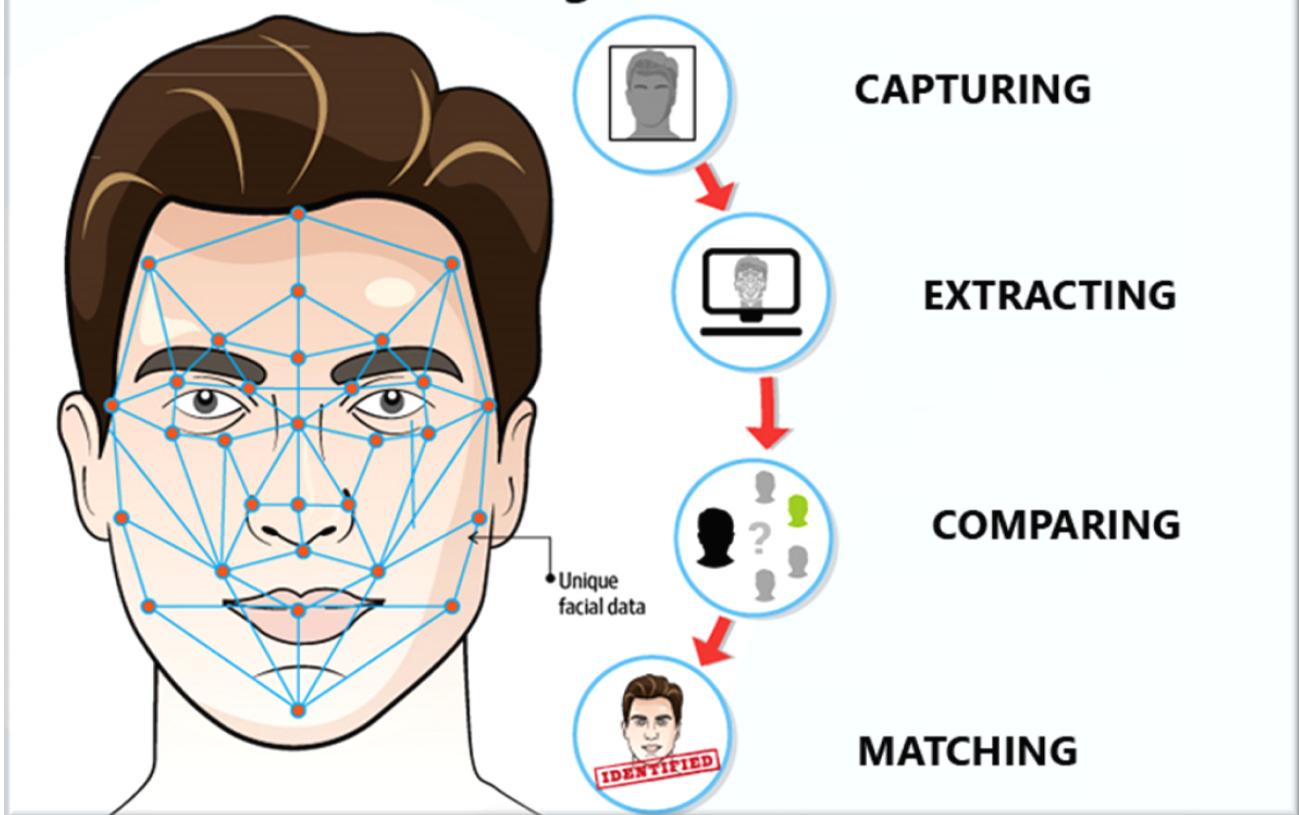
1. Enhanced security: Facial recognition can be used to identify individuals in high-security environments, such as airports, banks, and government facilities, which can help prevent unauthorized access.
2. Improved surveillance: Facial recognition technology can be used in surveillance systems to detect and track individuals of interest, such as criminals or suspects.
3. Increased efficiency: Facial recognition technology can automate many tasks that would otherwise require human intervention, such as passport control, ticketing, and payment processing.
4. Personalization: Facial recognition technology can be used to personalize customer experiences, such as recommending products or services based on a person's age or gender.

However, facial recognition technology also has several shortcomings and concerns, such as:

1. Privacy concerns: The use of facial recognition technology raises concerns about privacy and the potential for government or corporations to track and monitor individuals without their consent.
2. Accuracy issues: Facial recognition technology is not 100% accurate and has been shown to have higher error rates for people of color and women, which can lead to misidentification and false arrests.
3. Bias: Facial recognition technology is only as unbiased as the data it is trained on, and if the data contains biases, the technology can perpetuate those biases.
4. Ethical concerns: There are ethical concerns around the use of facial recognition technology, such as the potential for misuse, abuse, and violation of human rights.

It is important to consider both the potential benefits and shortcomings of facial recognition technology before implementing it in any application.

Biometrics Face Recognition - How does it Work?



5. SOFTWARE IMPLEMENTATION

5.1 ARDUINO IDE

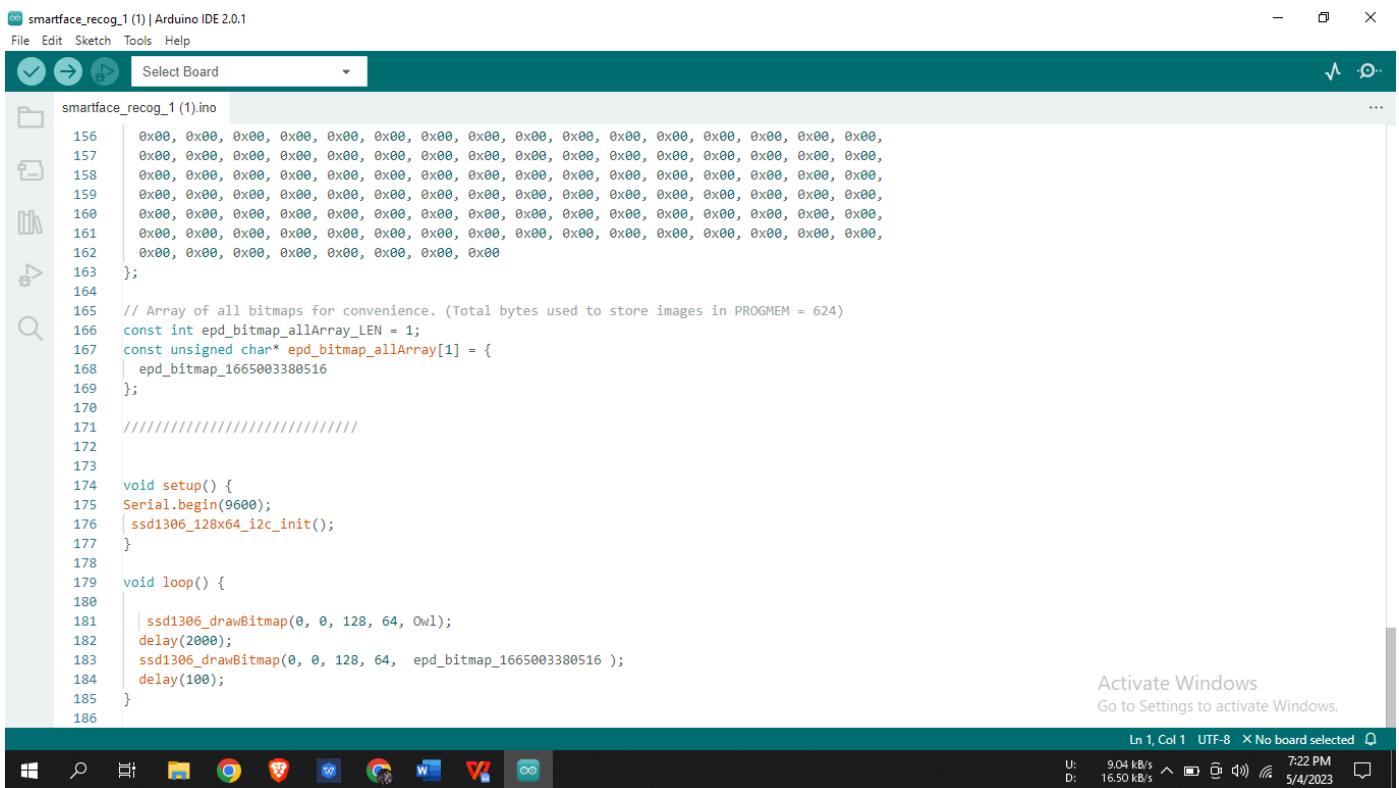
Arduino Integrated Development Environment (IDE) is a software application that is used to write, compile, and upload code to Arduino boards. It provides a user-friendly interface to the users for programming the Arduino boards. The Arduino IDE is an open-source software that is available for free to everyone. It supports various programming languages, including C and C++, and allows users to write code using a simple and intuitive syntax.

One of the main advantages of using the Arduino IDE is that it supports a wide range of Arduino boards, making it easy for users to choose the right board for their project. The IDE also has a built-in serial monitor that allows users to view the output of their code in real-time. Additionally, the Arduino IDE comes with a vast library of pre-built

code and examples, which makes it easier for beginners to get started with programming Arduino boards.

Moreover, the Arduino IDE also supports third-party libraries, which can be easily added to the IDE to extend its functionality. It also supports debugging tools, which can be used to find and fix errors in the code. The IDE is available for various operating systems, including Windows, Mac OS, and Linux, making it accessible to a broad range of users.

However, one of the drawbacks of the Arduino IDE is that it can be limited in terms of its functionality for more complex projects. In such cases, users may need to switch to more advanced programming tools. Additionally, the IDE can be overwhelming for beginners, and it may take some time for them to learn how to use it effectively.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** smartface_recog_1 (1) | Arduino IDE 2.0.1
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Save, Open, Run, Stop, and Select Board.
- Sketch Navigator:** Shows a tree view of the sketch structure, including a folder icon and the file name.
- Code Editor:** Displays the C++ code for the sketch. The code includes memory-mapped arrays for bitmaps and functions for setup and loop operations using the SSD1306 library.

```

156     0x00, 0x00,
157     0x00, 0x00,
158     0x00, 0x00,
159     0x00, 0x00,
160     0x00, 0x00,
161     0x00, 0x00,
162     0x00, 0x00
163 };
164
165 // Array of all bitmaps for convenience. (Total bytes used to store images in PROGMEM = 624)
166 const int epd_bitmap_allArray_LEN = 1;
167 const unsigned char* epd_bitmap_allArray[1] = {
168     epd_bitmap_1665003380516
169 };
170
171 /////////////////////////////////
172
173 void setup() {
174     Serial.begin(9600);
175     ssd1306_128x64_i2c_init();
176 }
177
178 void loop() {
179
180     ssd1306_drawBitmap(0, 0, 128, 64, Owl);
181     delay(2000);
182     ssd1306_drawBitmap(0, 0, 128, 64, epd_bitmap_1665003380516 );
183     delay(100);
184 }
185
186

```

- Status Bar:** Shows system information including network speed (U: 9.04 kB/s, D: 16.50 kB/s), system time (7:22 PM), and date (5/4/2023).

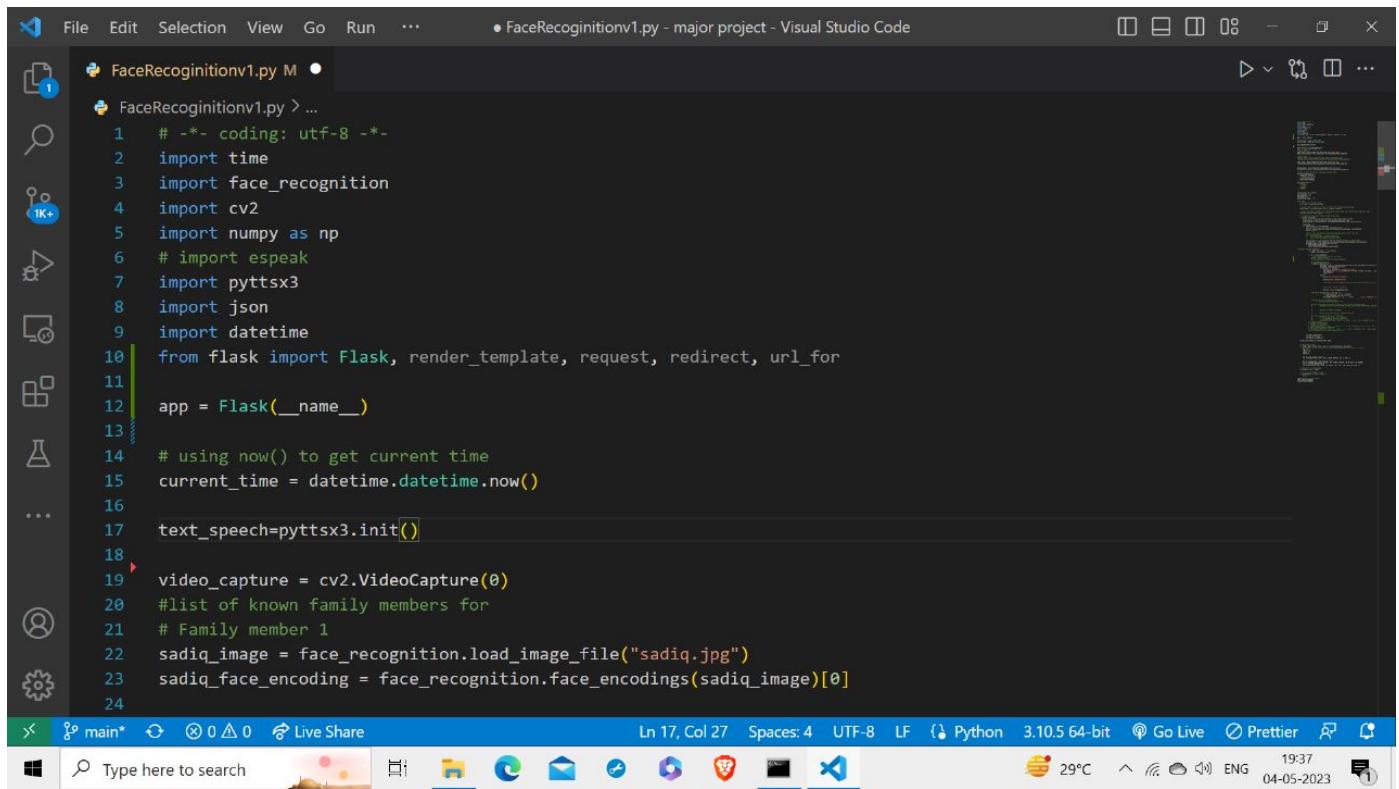
5.2 VISUAL STUDIO CODE

Visual Studio Code is a free and open-source code editor that is widely used by developers for building different types of software applications. In our project, Visual Studio Code is used as the primary code editor for writing and editing the Python code that runs on the Raspberry Pi.

Visual Studio Code offers many features that make it a popular choice for developers, such as a rich set of plugins, code completion, debugging tools, and source control integration. These features make it easier for developers to write and maintain high-quality code.

In our project, we use Visual Studio Code to write and debug Python code for our Raspberry Pi. We also use the platform to manage our source code using Git. This allows us to collaborate with team members and keep track of changes to our codebase.

Overall, Visual Studio Code is a versatile and powerful tool that helps developers to write efficient and reliable code. In our project, it plays a crucial role in helping us to build a robust and effective facial recognition attendance system.



The screenshot shows a Visual Studio Code interface with a Python file named 'FaceRecognitionv1.py' open. The code implements a basic facial recognition application using Flask, OpenCV, and pyttsx3 libraries. It loads a known family member's image, captures video from the camera, and compares faces in real-time.

```

1  # -*- coding: utf-8 -*-
2  import time
3  import face_recognition
4  import cv2
5  import numpy as np
6  # import espeak
7  import pyttsx3
8  import json
9  import datetime
10 from flask import Flask, render_template, request, redirect, url_for
11 app = Flask(__name__)
12
13 # using now() to get current time
14 current_time = datetime.datetime.now()
15
16 text_speech=pyttsx3.init()
17
18 video_capture = cv2.VideoCapture(0)
19 #list of known family members for
20 # Family member 1
21 sadiq_image = face_recognition.load_image_file("sadiq.jpg")
22 sadiq_face_encoding = face_recognition.face_encodings(sadiq_image)[0]
23
24

```

5.3 PYTHON

Python is a popular programming language used for facial recognition due to its simplicity, versatility, and vast collection of libraries. It offers a wide range of tools for processing images and videos, making it an ideal choice for implementing facial recognition algorithms. Python also allows developers to build complex systems and applications with relatively less coding, saving time and effort.

The `face_recognition` library in Python is one of the most commonly used libraries for facial recognition tasks. This library is built on top of OpenCV and DLib, two powerful computer vision libraries. It provides an easy-to-use interface to detect and recognize faces in images and videos. The library uses deep learning models trained on large datasets of face images to accurately identify faces.

Python's flexibility and simplicity make it an ideal choice for developing real-time facial recognition systems. The language can be easily integrated with other technologies, such as Raspberry Pi and web interfaces, enabling developers to build complex applications that can interact with the real world.

The ability to develop efficient and accurate facial recognition systems using Python has made it a popular choice in many industries, including security, surveillance, and marketing. Facial recognition can be used to identify criminals, track attendance, and personalize customer experiences. Its potential applications are vast and diverse, making it an exciting field for developers and researchers to explore.

Facial recognition technology is a rapidly evolving field that has garnered much attention in recent years due to its potential for numerous applications in security, marketing, and healthcare. In the context of our project, we have utilized Python programming language and relevant libraries like `face_recognition` and `cv2` for facial recognition, with the Raspberry Pi 4B serving as the main processor.

Python was chosen as the primary programming language due to its versatility, extensive libraries, and ease of use. The `face_recognition` library in Python provides an easy-to-use interface for performing facial recognition tasks, such as detecting and recognizing faces in an image or video stream. Additionally, the `cv2` library, which is a popular computer vision library, provides a wide range of functions to perform various image processing tasks such as image filtering, thresholding, and morphological operations.

Our project employs the use of a webcam for capturing images and video streams, which is an essential component of the facial recognition system. The webcam is a low-cost and widely available device that can capture high-quality images and video in real-time. Additionally, the Raspberry Pi 4B, which is a small and powerful single-board computer, serves as the main processing unit, providing the necessary computational power to perform real-time facial recognition tasks.

The use of Arduino UNO enables us to control the OLED displays present in the 3D printed face mask. Arduino UNO is an open-source electronic prototyping platform that provides a flexible and easy-to-use interface for controlling various electronic devices such as sensors, motors, and displays.

In summary, our project involves the integration of various technologies and hardware components to create a sophisticated and accurate facial recognition system that can recognize and record attendance of students. The use of Python, face_recognition and cv2 libraries, a webcam, Raspberry Pi 4B, and Arduino UNO provide the necessary computational and hardware infrastructure to achieve this objective.

5.4 LIBRARIES

5.4.1 FACIAL RECOGNITION

A facial recognition library is a set of pre-built software tools and algorithms that enable developers to easily and efficiently develop facial recognition applications. These libraries use deep learning algorithms to identify and match faces based on features such as facial contours, shape, and texture. Facial recognition libraries have a wide range of applications in various fields such as security, marketing, and healthcare. However, the use of facial recognition technology has also raised concerns around privacy and potential misuse of personal data. As such, it is important for developers to ensure that these technologies are developed and used responsibly, and that individuals are made aware of how their personal data is being collected and used. Facial recognition libraries are software packages that enable developers to integrate facial recognition capabilities into their applications. Some of the most popular facial recognition libraries used in Python are OpenCV, Dlib, Face Recognition, and DeepFace.

OpenCV is an open-source library that provides various algorithms and functions for computer vision tasks, including facial recognition. It has pre-trained models for face detection, eye detection, and facial landmark detection, making it a popular choice for facial recognition projects.

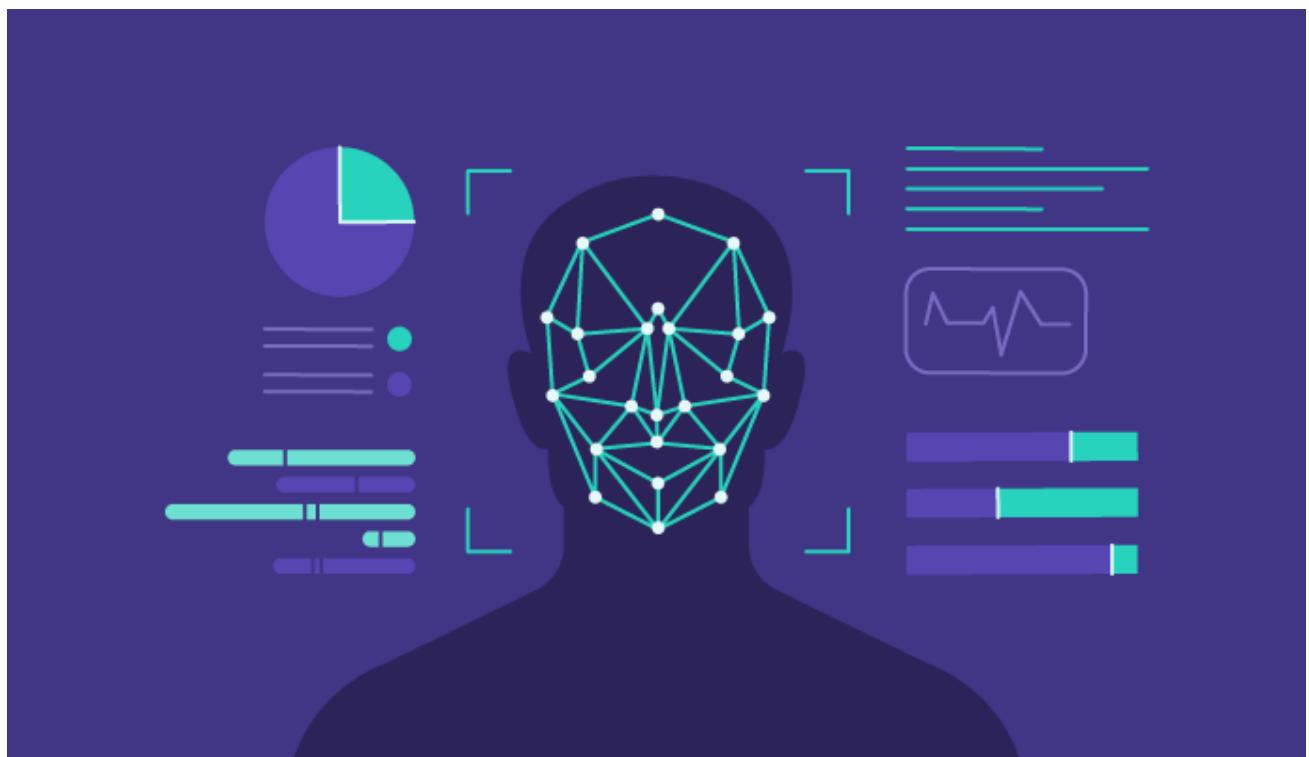
Dlib is another open-source library that provides facial recognition capabilities. It uses a machine learning algorithm called Histogram of Oriented Gradients (HOG) to detect faces and facial landmarks. Dlib is known for its high accuracy and speed, making it ideal for real-time facial recognition applications.

Face Recognition is a Python library that uses dlib and OpenCV to provide facial recognition capabilities. It has pre-trained models for face detection and recognition, as well as tools for training your own custom models.

DeepFace is a facial recognition library developed by Facebook's AI research team. It uses deep learning algorithms to recognize faces, and can even identify specific

individuals in a crowd. However, it requires a large amount of computing power and training data to achieve high accuracy.

These libraries can be used in combination with other Python libraries, such as NumPy and Pandas, to perform additional data analysis or processing tasks. They are essential tools for developers who want to add facial recognition capabilities to their applications.



5.4.2 CV2

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. The library has been designed to be cross-platform and it provides tools for developing real-time computer vision applications. OpenCV was initially released in 2000 and since then it has been widely used for a variety of applications ranging from facial recognition to robotics.

The cv2 library is a Python wrapper for OpenCV. It provides Python bindings for OpenCV and enables users to access many of the OpenCV functions in Python. This means that developers can use the same code base for computer vision applications in

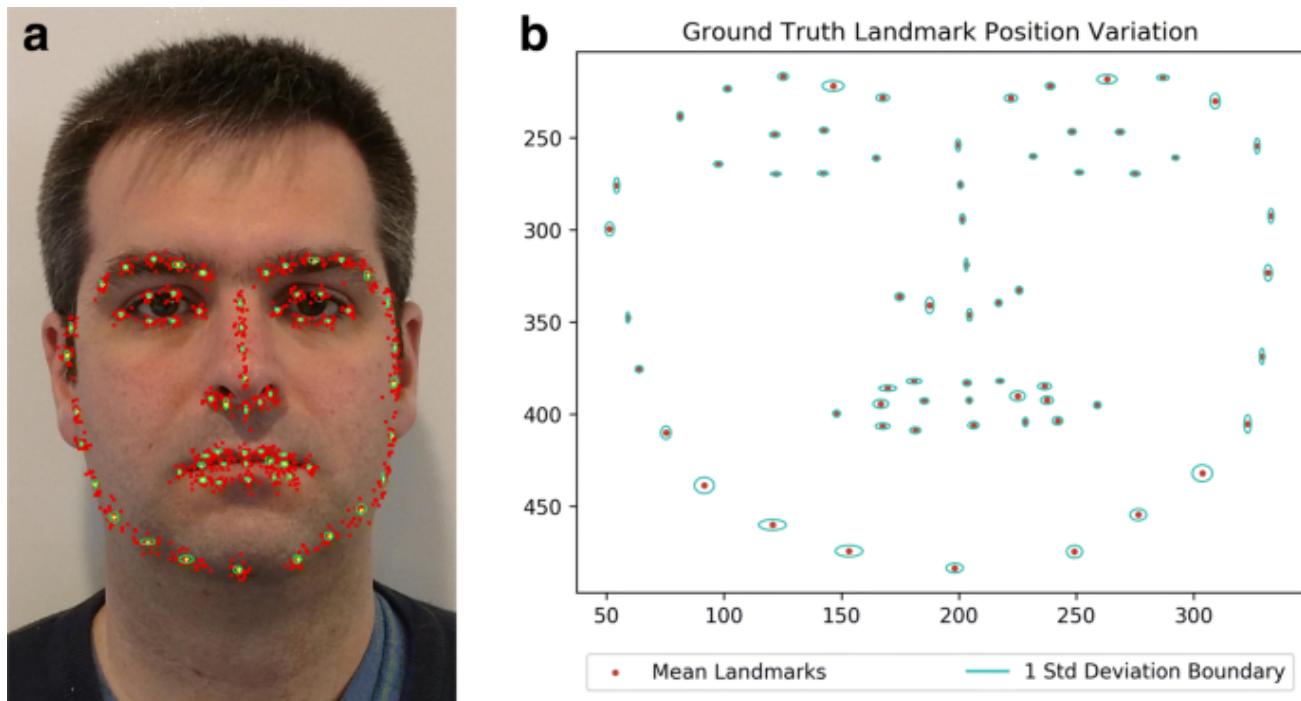
both C++ and Python. The cv2 library is widely used in computer vision applications because of its ease of use and its performance.

The cv2 library provides various modules that can be used for image processing and computer vision applications. Some of the most commonly used modules include:

1. Core: This module provides the basic functions for image processing such as reading and writing images, converting color spaces, and drawing shapes.
2. Image processing: This module provides functions for image filtering, thresholding, and edge detection.
3. Feature detection: This module provides functions for feature detection such as Harris corner detection and SIFT (Scale-Invariant Feature Transform).
4. Object detection: This module provides functions for object detection such as Haar Cascade Classifiers and HOG (Histogram of Oriented Gradients).

The cv2 library also provides support for video processing and computer vision applications in real-time. It includes functions for capturing video from a camera or a file, processing video frames, and displaying the results in real-time.

In conclusion, the cv2 library is an essential tool for computer vision and image processing applications in Python. Its easy-to-use interface and compatibility with OpenCV make it an ideal choice for developers who want to implement computer vision applications in Python.



5.4.3 NUMPY

NumPy (Numerical Python) is an open-source numerical computing library for Python that provides support for large, multi-dimensional arrays and matrices, as well as a large collection of high-level mathematical functions to operate on these arrays. It is a fundamental library for scientific computing with Python.

NumPy was first released in 2006 and has since become one of the most widely used libraries in the scientific Python ecosystem. It is a core library for data analysis, machine learning, and scientific computing.

NumPy provides a high-performance, memory-efficient implementation of arrays and matrices, which are the core data structures used in scientific computing. It includes functions for performing mathematical operations on these arrays, such as linear algebra, Fourier transforms, and random number generation. NumPy arrays are also interoperable with other libraries for scientific computing in Python, such as SciPy, Matplotlib, and Pandas.

One of the key benefits of using NumPy is its performance. NumPy arrays are implemented in C and are highly optimized for numerical operations, making them much faster than Python lists for large arrays. NumPy also provides a number of functions for efficient data manipulation, such as broadcasting, which allows operations to be performed on arrays of different shapes and sizes.

Overall, NumPy is a powerful library for numerical computing and is essential for scientific computing with Python. Its high-performance implementation of arrays and matrices, combined with a large collection of mathematical functions, make it a valuable tool for data analysis, machine learning, and scientific research.

NumPy is a widely used library in the field of data science, scientific computing, and engineering. It provides a powerful array data structure and tools for working with these arrays. Some of the key applications of NumPy are:

1. Numerical and Scientific Computing: NumPy provides a fast and efficient way to perform mathematical operations on large arrays of data. It is used in various scientific and numerical applications, such as signal processing, linear algebra, Fourier analysis, and statistical analysis.
2. Data Analysis and Data Visualization: NumPy is an essential library for data analysis and data visualization in Python. It provides tools for working with multidimensional arrays and enables users to perform complex data operations

such as filtering, sorting, and aggregation. It also provides support for plotting and visualizing data using libraries such as Matplotlib and Seaborn.

3. Machine Learning: NumPy is extensively used in machine learning for building and training models. It provides a foundation for other machine learning libraries such as Scikit-learn and TensorFlow. The library allows users to work with high-dimensional data and perform operations such as matrix multiplication and linear algebraic computations.
4. Image and Video Processing: NumPy is used in various applications related to image and video processing. It provides tools for image and video manipulation, such as resizing, cropping, and rotating. NumPy is also used in computer vision tasks such as object detection, face recognition, and image segmentation.
5. Computational Geometry: NumPy provides support for computational geometry tasks such as triangulation, Voronoi diagrams, and convex hulls. It is used in various applications such as computer graphics, robotics, and geographic information systems.

Overall, NumPy has become an essential library for scientific computing and data analysis in Python, with a wide range of applications in various fields.

5.4.4 PYTTSX3

Pyttsx3 is a Python library that provides a cross-platform Text-to-Speech (TTS) engine to produce artificial human-like voices. It enables the conversion of text to speech in real-time, making it an essential tool for various applications such as assistive technology for people with visual impairments, automated voice assistants, and chatbots.

The library is based on the Microsoft Speech API (SAPI) and uses different TTS engines such as eSpeak, nsss, and sapi5, making it a versatile tool that supports different platforms and operating systems.

Pyttsx3 offers a range of features and customizations such as setting the voice type, speed, and volume, and is easy to integrate with other Python libraries and applications. It also supports multiple languages, including English, French, German, Italian, and Spanish.

In our project, we used pyttsx3 to generate voice alerts for the users, such as providing information about the recognition process or notifying them about the attendance status. The library helped us provide a more user-friendly interface, and its customization options allowed us to create a personalized experience for the users. Overall, the pyttsx3 library is a valuable addition to any project that requires a TTS

engine and provides a high level of flexibility and control. The primary application of pytsxs3 is in the development of speech-enabled applications. For instance, it can be used in creating virtual assistants like Siri, Google Assistant, and Alexa, which rely on natural language processing and text-to-speech conversion to communicate with users.

In addition, pytsxs3 can also be used in creating assistive technology for individuals with visual impairments or reading difficulties. By converting text into speech, it can enable these individuals to access written materials like books, articles, and websites.

Moreover, pytsxs3 can be used in developing educational applications. For example, it can be used in creating e-learning platforms where the text content is converted to speech, making it easier for learners to comprehend the material.

Furthermore, pytsxs3 can be used in the development of entertainment applications such as games, where speech output can enhance the user experience by providing auditory feedback and dialogue.

In summary, pytsxs3 is a versatile library that can be used in various applications, including speech-enabled applications, assistive technology, e-learning, and entertainment applications.

```

1 import pytsxs3
2 engine = pytsxs3.init()
3 engine.say("I will speak this text")
4 engine.runAndWait()
```

5.4.5 DATETIME

The datetime module is a Python library that provides classes for working with dates and times. It offers various functions to manipulate dates and times in Python. The library has two main classes - datetime and timedelta.

The datetime class is used to represent a date and time combination. It has attributes for year, month, day, hour, minute, second, microsecond, and timezone information.

You can create a datetime object using its constructor, and you can also parse strings to create datetime objects using the strftime method.

The timedelta class is used to represent a duration, or the difference between two dates or times. You can create a timedelta object using its constructor, and you can perform arithmetic with timedelta objects to calculate new dates or times.

The datetime module is commonly used in web applications and data analysis tasks where date and time information is frequently processed. Some applications include calculating the age of a user from their birthdate, converting between timezones, or creating time-based plots.

For example, in our facial recognition project, we use the datetime module to keep track of attendance records for students. When a student is recognized, the current date and time are recorded using the datetime module and stored in a database. Later, this information can be used to generate attendance reports or to calculate the total number of hours a student has attended class.

5.4.5 TIME

The time library in Python provides various time-related functions that can be used in a program. One of the most commonly used functions from the time library is the sleep() function, which causes the program to pause execution for a specified amount of time.

The sleep() function takes a single argument, which is the number of seconds for which the program should pause. This can be useful in situations where the program needs to wait for a certain amount of time before continuing execution, such as when a delay is required between actions.

For example, in our college attendance system project, we use the time library to introduce a delay of 2 seconds between each face recognition process. This ensures that there is enough time for the system to capture and recognize the face before moving on to the next student.

The time library also provides other functions such as time() and localtime(), which can be used to retrieve information about the current time and date. This can be useful in a variety of applications, such as creating timestamps for data logging or generating time-based data for analysis.

Overall, the time library in Python provides a useful set of tools for handling time-related functionality in a program, and its use in our college attendance system project is a good example of how it can be utilized to achieve specific goals.

5.5 WEB DEVELOPMENT

5.5.1 HTML

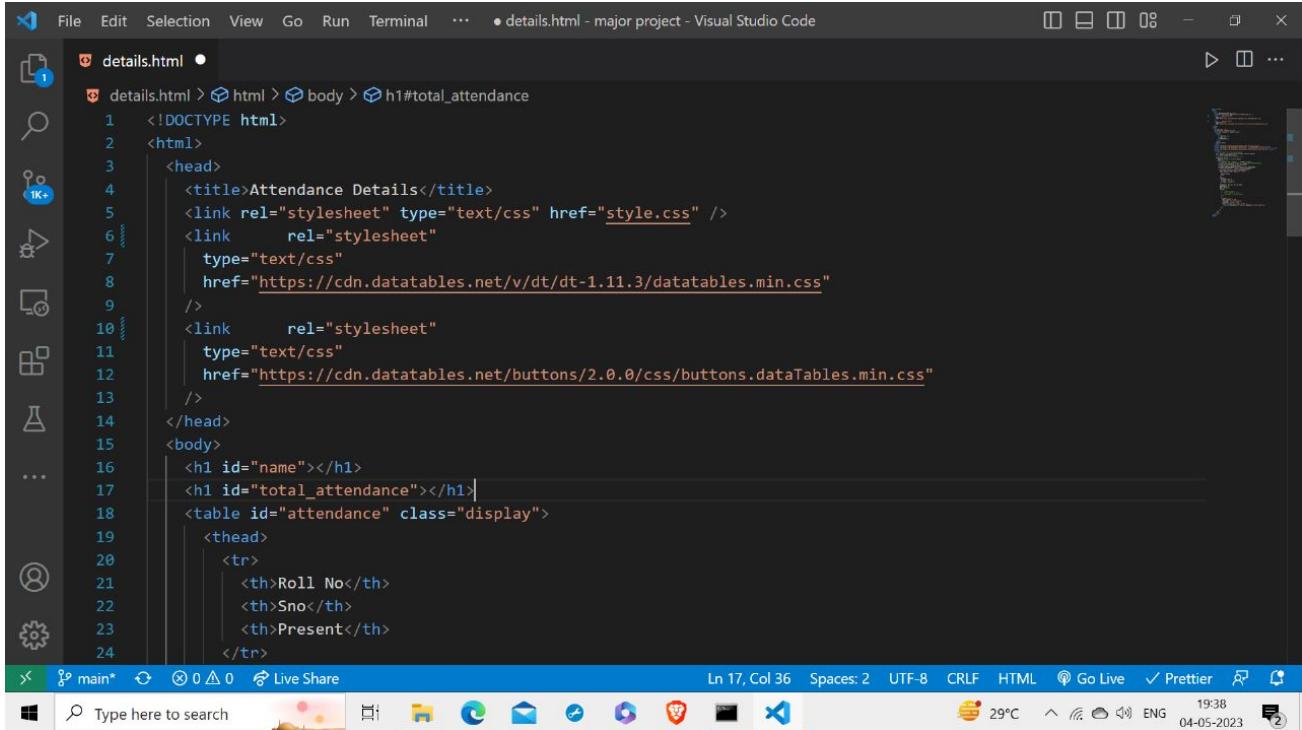
In our project, HTML (Hypertext Markup Language) is used to create the web interface for the attendance system. HTML is the standard markup language used to create web pages and it defines the structure and layout of a web page. The HTML code specifies the text, images, and other media that are included on a web page and how they are organized and displayed to the user.

HTML is used to create the basic structure of the web page, including the header, body, and footer sections. It is also used to add content to the page, including text, images, videos, and other multimedia elements. Additionally, HTML is used to create links between different pages on the website and to other websites.

The use of HTML in our project allows us to create a user-friendly interface for the attendance system that can be accessed through a web browser. This interface displays relevant information such as the attendance records of students and teachers, as well as real-time updates on the recognition process. It also provides easy navigation for the user to access different sections of the system.

Moreover, HTML is compatible with other web technologies, such as CSS and JavaScript, which allows us to enhance the functionality and appearance of the interface. The HTML code is easy to understand and modify, making it a flexible and powerful tool for creating web-based applications.

In summary, the use of HTML in our project enables us to create an intuitive and interactive web interface for the attendance system, providing users with a seamless experience while accessing the system's features and functionalities.



```

File Edit Selection View Go Run Terminal ... • details.html - major project - Visual Studio Code
details.html •
details.html > html > body > h1#total_attendance
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Attendance Details</title>
5      <link rel="stylesheet" type="text/css" href="style.css" />
6      <link rel="stylesheet" type="text/css"
7          href="https://cdn.datatables.net/v/dt/dt-1.11.3/datatables.min.css"
8      />
9      <link rel="stylesheet" type="text/css"
10         href="https://cdn.datatables.net/buttons/2.0.0/css/buttons.dataTables.min.css"
11     />
12   </head>
13   <body>
14     <h1 id="name"></h1>
15     <h1 id="total_attendance"></h1>
16     <table id="attendance" class="display">
17       <thead>
18         <tr>
19           <th>Roll No</th>
20           <th>Sno</th>
21           <th>Present</th>
22         </tr>
23       </thead>
24     </table>

```

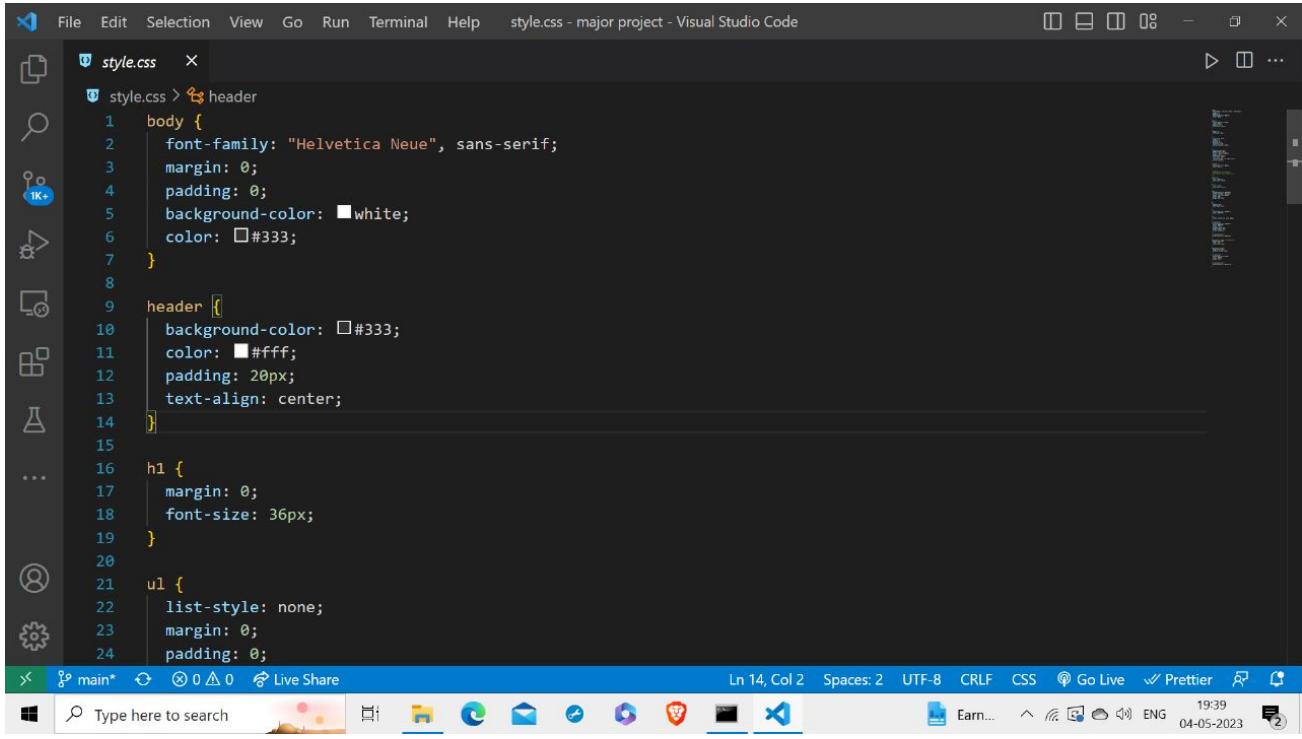
5.5.2 CSS

In our project, we have used Cascading Style Sheets (CSS) to enhance the appearance and visual appeal of our web interface. CSS is a style sheet language that is used to describe the look and formatting of a document written in HTML. It allows us to control the layout of our web pages, including elements such as text, images, and other media.

CSS allows us to define the visual aspects of our HTML elements, such as font size, color, background color, and margin, padding, border, and more. We have used CSS to create a clean and consistent look and feel throughout our web interface. This makes it easier for users to navigate and interact with our attendance system.

Furthermore, CSS enables us to create responsive design, which means that our web interface can adjust its layout based on the size of the user's device screen. This is important because users may access our attendance system on various devices with different screen sizes, such as smartphones, tablets, or laptops.

Overall, the use of CSS in our project has allowed us to create a visually appealing and user-friendly web interface for our facial recognition-based attendance system.



```

File Edit Selection View Go Run Terminal Help style.css - major project - Visual Studio Code
style.css > header
body {
    font-family: "Helvetica Neue", sans-serif;
    margin: 0;
    padding: 0;
    background-color: white;
    color: #333;
}
header {
    background-color: #333;
    color: #fff;
    padding: 20px;
    text-align: center;
}
h1 {
    margin: 0;
    font-size: 36px;
}
ul {
    list-style: none;
    margin: 0;
    padding: 0;
}

```

5.5.3 JAVASCRIPT

In our project, we have used JavaScript as a client-side programming language for developing the web interface. JavaScript is a high-level, interpreted scripting language that is used to create interactive web pages and web applications. It is a powerful tool for front-end web development and can be used to add interactivity, animations, and dynamic content to a website.

We have used JavaScript to create various interactive elements on our web interface, such as buttons, drop-down menus, and forms. We have also used JavaScript to send requests to our Flask backend and retrieve data from our database dynamically. This allows for a smoother and more user-friendly experience for the end-users.

Additionally, we have used JavaScript to integrate Plotly, a data visualization library, into our web interface. This allows us to create interactive graphs and charts to represent the attendance data collected by our system. With Plotly, we can create beautiful and customizable visualizations that make it easier to analyze and understand attendance trends.

Overall, JavaScript plays a crucial role in our project by allowing us to create a dynamic and interactive web interface that enhances the user experience and makes it easier to analyze attendance data.

5.5.4 JAVASCRIPT OBJECT NOTATION (JSON)

In our project, we have used JSON (JavaScript Object Notation) for data transfer between the server and client-side. JSON is a lightweight, text-based data format that is easy for humans to read and write and also easy for machines to parse and generate.

The data collected from the facial recognition system, such as the name of the recognized person and the date and time of their attendance, is stored in JSON format. This data is then sent to the server-side for storage and analysis.

One of the main advantages of using JSON is its simplicity and ease of use. It is a widely used data interchange format in modern web applications and can be easily parsed using built-in JavaScript functions. Additionally, JSON is platform-independent and can be used with any programming language that can parse text data.

By using JSON in our project, we have made the data transfer process more efficient and streamlined, allowing for easier data management and analysis.

```

File Edit Selection View Go Run Terminal Help data.json - major project - Visual Studio Code
data.json M
data.json > {} 0 > [ ]present
1 [
2   {
3     "name": "sadiq",
4     "rollno": "1604-19-735-030",
5     "total_attendance": 6,
6     "present": [
7       "27-4-2023",
8       "28-4-2023",
9       "29-4-2023",
10      "1-5-2023",
11      "2-5-2023",
12      "4-5-2023"
13    ],
14  },
15  {
16    "name": "irfan",
17    "rollno": "1604-19-735-015",
18    "total_attendance": 3,
19    "present": [
20      "27-4-2023",
21      "29-4-2023",
22      "4-5-2023"
23    ],
24  }
]

```

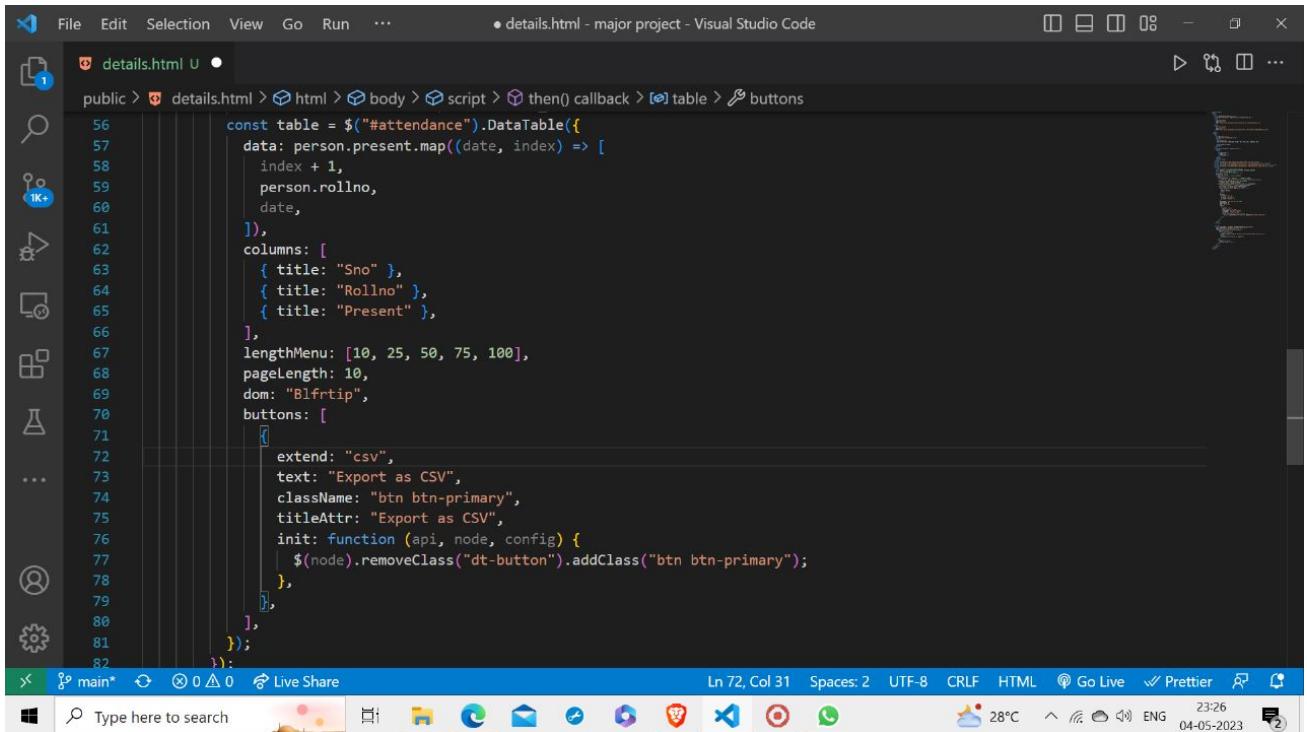
5.5.5 jQuery

In our project, we have utilized the jQuery library to streamline the process of handling events and manipulating the HTML DOM. jQuery is a fast, small, and feature-rich JavaScript library that simplifies the way of traversing HTML documents, event handling, and animating elements on a web page. It has a lot of useful functions that allow developers to easily manipulate and interact with web pages.

The main benefit of using jQuery in our project is that it simplifies the process of working with HTML and CSS, making it easier for developers to write concise and efficient code. It allows us to easily manipulate the DOM, handle events, and add animations to our web page with just a few lines of code.

For instance, in our project, we have used jQuery to handle form submission events, retrieve data from the server using AJAX requests, and dynamically modify the contents of the web page based on user input. We have also used the jQuery UI library to add various interactive components to our web page, such as sliders, progress bars, and date pickers.

In summary, the use of jQuery has allowed us to build a responsive and dynamic web interface for our project with minimal effort, making it more user-friendly and efficient.



```

public > details.html > html > body > script > then() callback > table > buttons
56 const table = $("#attendance").DataTable({
57   data: person.present.map((date, index) => [
58     index + 1,
59     person.rollno,
60     date,
61   ]),
62   columns: [
63     { title: "Sno" },
64     { title: "Rollno" },
65     { title: "Present" },
66   ],
67   lengthMenu: [10, 25, 50, 75, 100],
68   pageLength: 10,
69   dom: "Blfrtip",
70   buttons: [
71     {
72       extend: "csv",
73       text: "Export as CSV",
74       className: "btn btn-primary",
75       titleAttr: "Export as CSV",
76       init: function (api, node, config) {
77         $(node).removeClass("dt-button").addClass("btn btn-primary");
78       },
79     },
80   ],
81 });
82 });

```

Live Share

Ln 72, Col 31 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier

28°C ENG 04-05-2023

5.5.6 AUTHENTICATION

Node.js and Express are popular tools for building web applications. One important aspect of web applications is user authentication and session management. This is where the use of sessions and authentication comes into play.

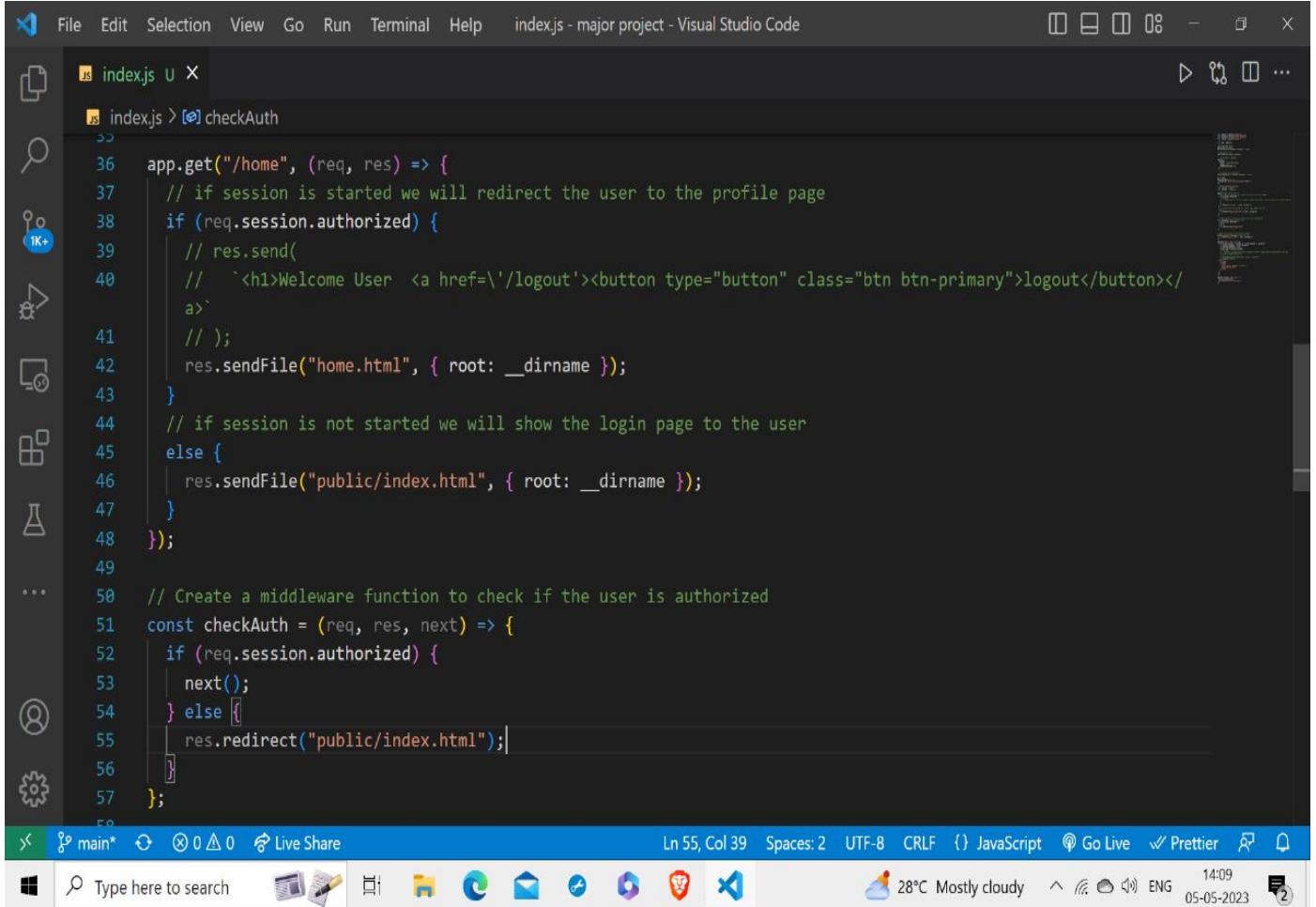
Sessions are used to maintain user state across multiple requests. When a user logs in, a unique session ID is created and stored on the server. This session ID is then sent back to the client in the form of a cookie. On subsequent requests, the session ID is sent back to the server and used to identify the user and retrieve their session data.

Authentication is the process of verifying that a user is who they claim to be. This is typically done by asking the user for a username and password. The server then checks these credentials against a database of users to determine if the user is authorized to access the requested resource.

In a Node.js and Express web application, there are several libraries available to handle sessions and authentication. One popular library is Passport.js, which provides a middleware layer for authentication. Passport.js supports a wide range of authentication strategies, including local authentication (username and password), OAuth, and OpenID.

Another popular library is express-session, which provides a middleware for handling sessions. With express-session, the session data is stored in a store, which can be a memory store, a file store, or a database store. This allows for flexibility in how session data is managed.

Overall, the use of sessions and authentication is essential for building secure web applications. With Node.js and Express, there are many libraries available to simplify the process and ensure that user data is kept safe.



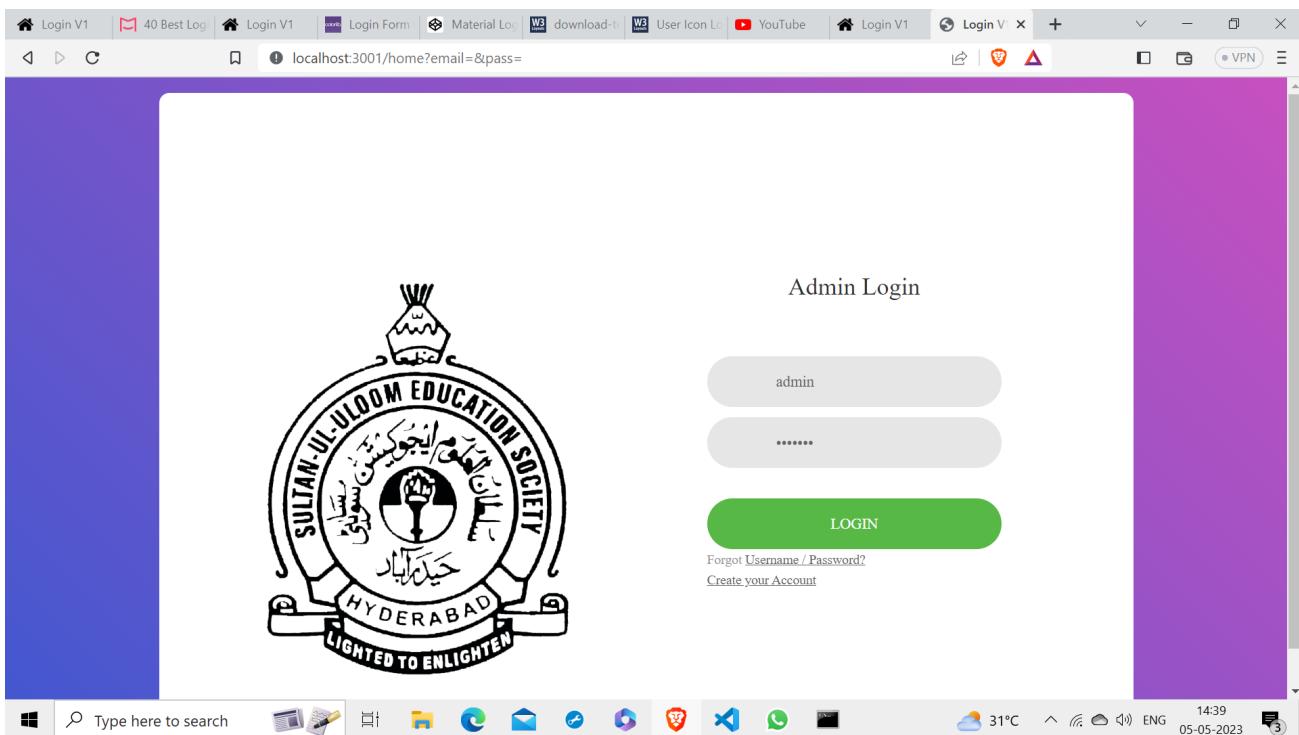
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** index.js - major project - Visual Studio Code.
- Code Editor:**

```

36 app.get("/home", (req, res) => {
37   // if session is started we will redirect the user to the profile page
38   if (req.session.authorized) {
39     // res.send(
40     //   `<h1>Welcome User <a href='/logout'><button type="button" class="btn btn-primary">logout</button></a>`
41     // );
42     res.sendFile("home.html", { root: __dirname });
43   }
44   // if session is not started we will show the login page to the user
45   else {
46     res.sendFile("public/index.html", { root: __dirname });
47   }
48 });
49
50 // Create a middleware function to check if the user is authorized
51 const checkAuth = (req, res, next) => {
52   if (req.session.authorized) {
53     next();
54   } else {
55     res.redirect("public/index.html");
56   }
57 };

```
- Sidebar:** Shows icons for file operations like copy, paste, search, and refresh.
- Bottom Bar:**
 - main* (active tab)
 - Live Share icon
 - Ln 55, Col 39, Spaces: 2, UTF-8, CRLF, JavaScript, Go Live, Prettier icons.
 - Type here to search input field.
 - Taskbar icons for file operations.
 - System tray: Weather (28°C), Cloudiness (Mostly cloudy), Date (05-05-2023), Time (14:09).



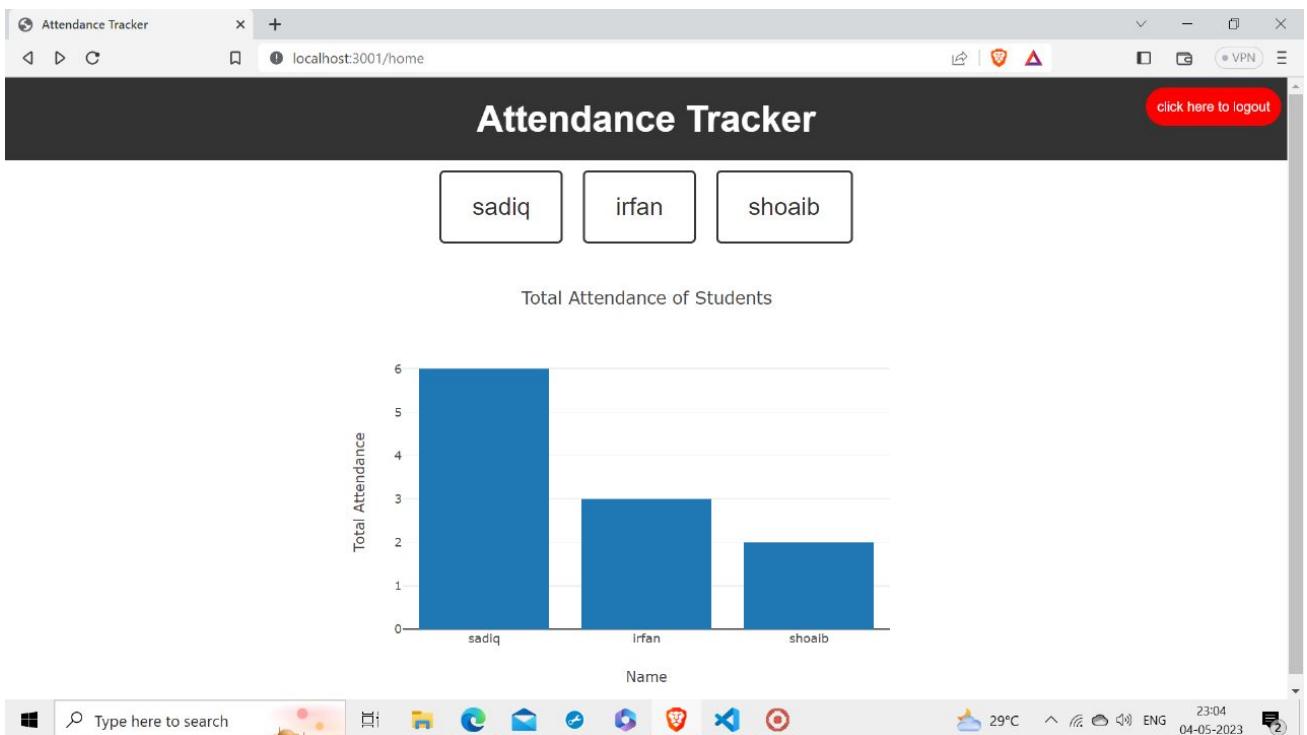
5.5.7 DATA VISUALISATION

The data in the project has been visualized using the Plotly library. Plotly is a data visualization library that allows the creation of interactive charts and graphs. The library provides an easy-to-use interface for creating visually appealing and informative visualizations.

In our project, Plotly has been used to create charts that display the attendance statistics of students in different subjects. These charts are interactive and allow users to hover over different parts of the chart to see more detailed information. For example, when the user hovers over a particular data point on the chart, the name of the student and their attendance percentage is displayed.

The charts are created using JSON data that is passed to the Plotly library. The library then generates the charts based on the data. This makes it easy to update the charts as new data becomes available.

Overall, the use of Plotly has allowed for the creation of visually appealing and informative charts that provide insights into the attendance patterns of students in different subjects.



6. OUTPUTS AND RESULTS

6.1 INTRODUCTION

The output and results of the project are the culmination of the different components and technologies utilized to achieve the objectives. The project's main goal is to automate the attendance system using facial recognition technology and display the attendance data on a web interface. The output includes the attendance data recorded and processed by the facial recognition algorithm, displayed through the OLED display and saved to a JSON file. The results of the project show how the system can effectively and accurately track attendance, eliminate the need for manual recording, and minimize the risk of errors.

Furthermore, the web interface displays attendance statistics and data visualizations generated using Plotly, making it easy for teachers and administrators to monitor attendance trends and patterns. The visualizations help to identify attendance issues that may require further investigation, such as frequent absences or tardiness. The system's use of a 3D printed face mask ensures the safety of students and staff by minimizing the risk of contact transmission of germs, especially during the COVID-19 pandemic.

Overall, the project's output and results demonstrate the efficiency, accuracy, and safety benefits of using facial recognition technology in attendance tracking, as well as the convenience and insights provided by the web interface and data visualizations.

6.2 OUTPUT

The output of a facial recognition-based attendance system would include the following:

1. Attendance records: The system will generate records of the attendance of each student or employee in a database or spreadsheet format.
2. Real-time status: The system will display the real-time status of the attendance, such as the number of students present, absent, and late.
3. Notification: The system may generate notifications via email or text message to the relevant parties, such as students, parents, or supervisors.
4. Analytics: The system may provide visual analytics of attendance data, such as graphs and charts, to help identify patterns and trends.
5. Verification: The system may provide verification of the attendance records using facial recognition technology, ensuring that the data is accurate and reliable.
6. Efficiency: The system will provide an efficient way to take attendance, reducing the need for manual tracking and reducing errors.

Overall, the output of the facial recognition-based attendance system will be a streamlined attendance process that saves time and increases accuracy.

6.3 RESULTS

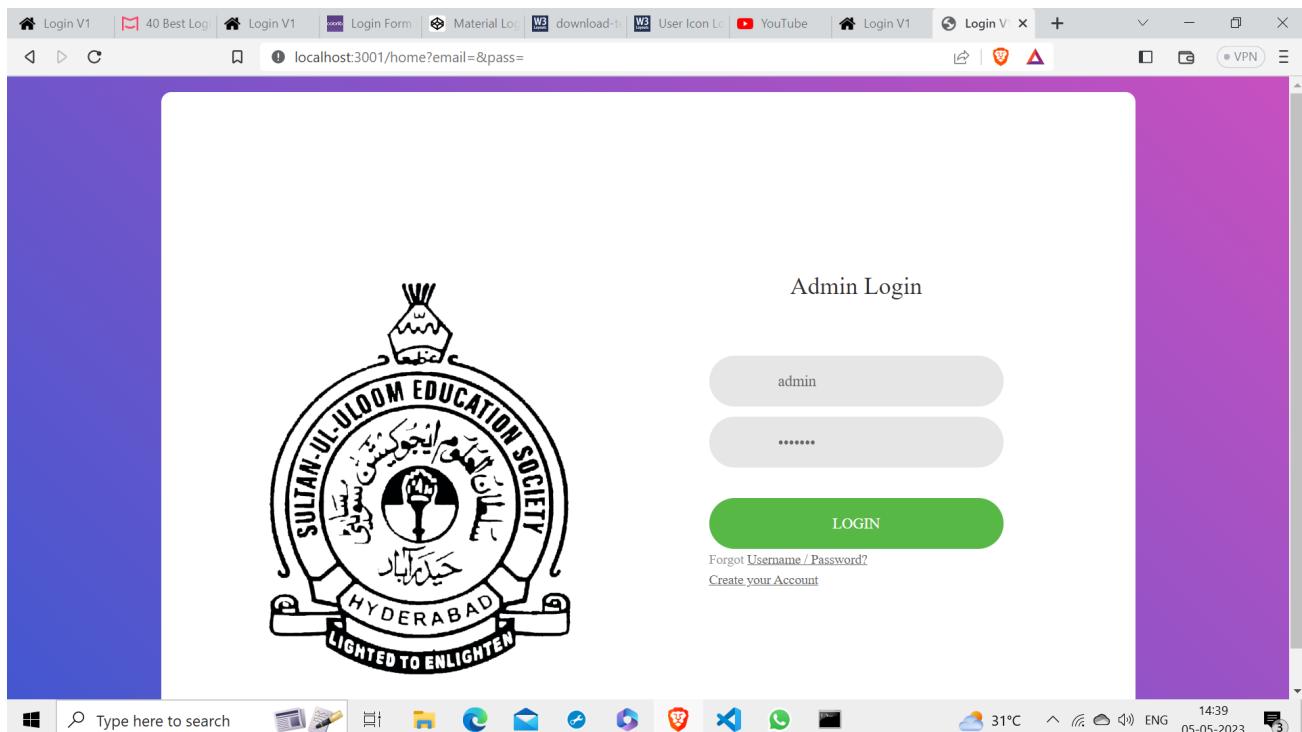
The expected results would be a facial recognition system that accurately detects and recognizes faces in real-time using the webcam connected to the Raspberry Pi 4B. The system performs tasks such as identifying registered faces and displaying their names on the OLED display.

The project also displays the date and time on the OLED display using the `DateTime` library, along with providing an audio output to notify users of the system's actions using the `pyttsx3` library. Additionally, the system should be able to save information

about recognized faces in a JSON file, which can be accessed later for analysis or to update the database of known faces.

The web interface should allow users to register new faces, view the list of registered faces, and delete any records if necessary. The use of HTML, CSS, JavaScript, and jQuery should ensure a user-friendly interface with a responsive design that works across different devices.

The successful implementation of these features would demonstrate the effectiveness of the system and validate the objectives of the project.



The screenshot shows the 'Attendance Tracker' application running on a Windows desktop. The browser window title is 'Attendance Tracker' and the URL is 'localhost:3001/home'. At the top right of the browser is a red button labeled 'click here to logout'. Below the header, three student names are displayed in separate rounded rectangular boxes: 'sadiq', 'irfan', and 'shoailb'. A chart titled 'Total Attendance of Students' is shown below the names. The Y-axis is labeled 'Total Attendance' and ranges from 0 to 6. The X-axis is labeled 'Name' and lists 'sadiq', 'irfan', and 'shoailb'. The bars show attendance counts of 6, 3, and 2 respectively. The Windows taskbar at the bottom includes a search bar, pinned icons for File Explorer, Edge, Mail, and others, and system status indicators like weather (29°C), date (04-05-2023), and time (23:02).

The screenshot shows the 'Attendance Details' page for student 'irfan' at the URL 'localhost:3001/details.html?name=irfan'. The main content displays the student's name in large blue text ('Student Name: irfan') and their total attendance ('Total Attendance: 3'). To the right is a blue button labeled 'Export as CSV'. Below this, there is a table with three entries. The table has columns for 'Sno', 'Rollno', and 'Present'. The entries are:

| Sno | Rollno | Present |
|-----|-----------------|-----------|
| 1 | 1604-19-735-015 | 27-4-2023 |
| 2 | 1604-19-735-015 | 29-4-2023 |
| 3 | 1604-19-735-015 | 4-5-2023 |

At the bottom left, it says 'Showing 1 to 3 of 3 entries'. On the right, there are buttons for 'Previous' (disabled), '1', and 'Next'. The Windows taskbar at the bottom is identical to the one in the previous screenshot.

Attendance Details +

localhost:3001/details.html?name=shoaib

Student Name: shoaib
Total Attendance: 2

[click here to logout](#)

Show 10 entries

| Sno | Rollno | Present |
|-----|-----------------|-----------|
| 1 | 1604-19-735-019 | 27-4-2023 |
| 2 | 1604-19-735-019 | 29-4-2023 |

Showing 1 to 2 of 2 entries

Previous 1 Next

Export as CSV

Search:



Attendance Details +

localhost:3001/details.html?name=sadiq

Student Name: sadiq
Total Attendance: 6

[click here to logout](#)

Show 10 entries

| Sno | Rollno | Present |
|-----|-----------------|-----------|
| 1 | 1604-19-735-030 | 27-4-2023 |
| 2 | 1604-19-735-030 | 28-4-2023 |
| 3 | 1604-19-735-030 | 29-4-2023 |
| 4 | 1604-19-735-030 | 1-5-2023 |
| 5 | 1604-19-735-030 | 2-5-2023 |
| 6 | 1604-19-735-030 | 4-5-2023 |

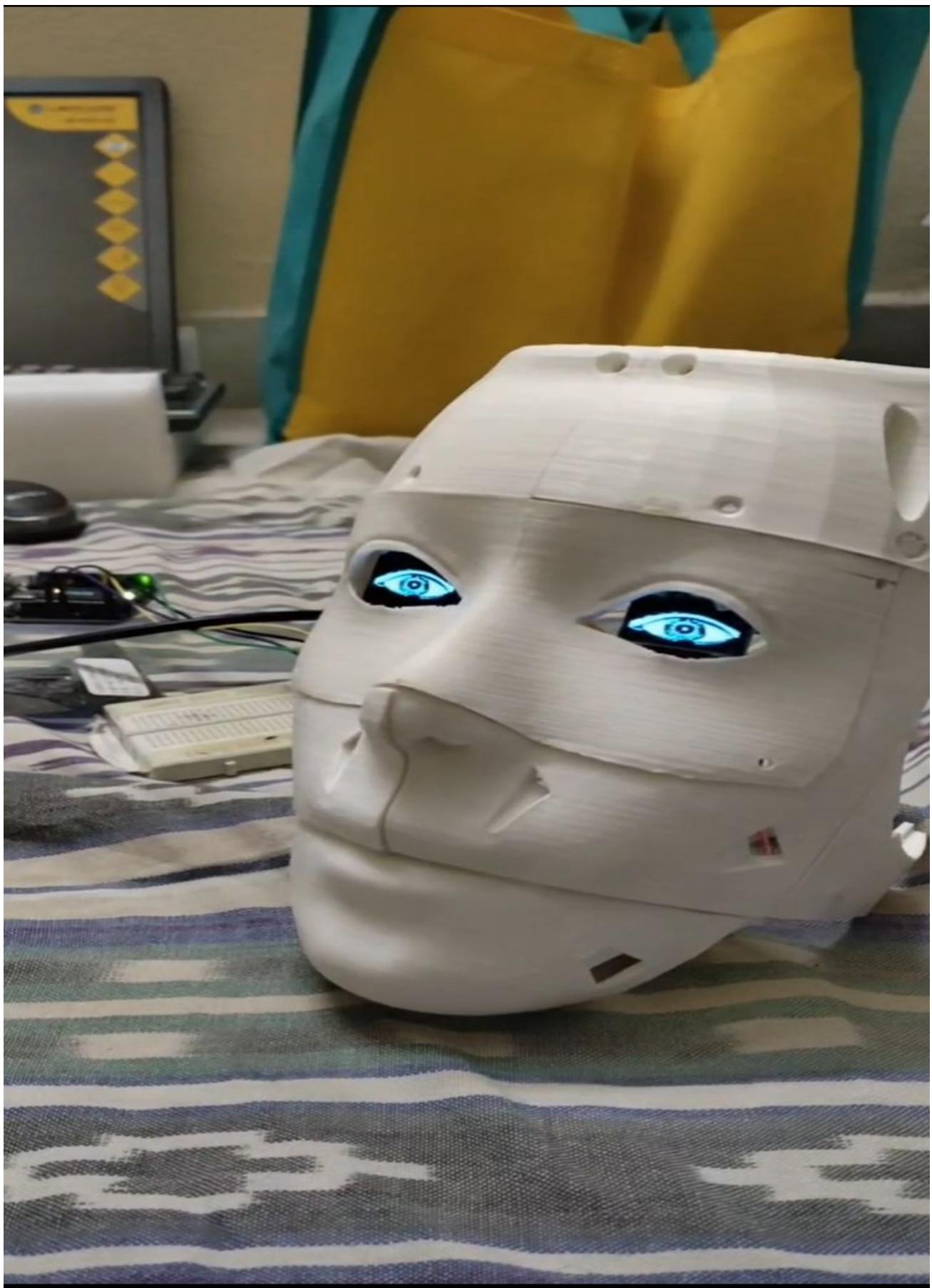
Showing 1 to 6 of 6 entries

Previous 1 Next

Export as CSV

Search:





7. CONCLUSION

In conclusion, the facial recognition-based attendance system is a highly efficient and accurate solution for managing attendance in various settings. With the use of advanced technology, such as computer vision and machine learning, this project can identify and verify individuals' identities with a high degree of precision. By integrating it with hardware devices such as Raspberry Pi, Arduino Uno, and webcam, it can be implemented in a real-time environment.

The use of various libraries such as OpenCV, NumPy, pytsx3, and datetime, along with web development technologies such as HTML, CSS, JavaScript, JSON, and jQuery, has made it possible to create a comprehensive and user-friendly system that is easily customizable and scalable. The project's web-based interface allows for easy monitoring and management of attendance data from any location.

The project's accuracy, ease of use, and flexibility make it an ideal solution for managing attendance in various institutions and organizations, including schools, colleges, universities, and businesses. Additionally, the system's ability to generate reports, including the total number of attendees, latecomers, and absentees, provides valuable insights that can help improve the attendance management system's overall efficiency.

Overall, the project demonstrates the potential of facial recognition technology in revolutionizing attendance management systems, and it can be further improved with the integration of more advanced features such as real-time monitoring and analysis of attendance data.

8. REFERENCES

- [1] A. Ahmed and S. Nandyal, “An Automatic Attendance System Using Image processing,” pp. 1–8, 2015.
- [2] V. Shehu and A. Dika, “Using real time computer vision algorithms in automatic attendance management systems,” Inf. Technol. Interfaces (ITI), 2010 32nd Int. Conf., pp. 397–402, 2010.
- [3] J. Joseph and K. P. Zacharia, “Automatic Attendance Management System Using Face Recognition,” Int. J. Sci. Res., vol. 2, no. 11, pp. 327–330, 2013.
- [4] J. Kanti and A. Papola, “Smart Attendance using Face Recognition with Percentage Analyzer,” vol. 3, no. 6, pp. 7321–7324, 2014.
- [5] T. H. Le, “Applying Artificial Neural Networks for Face Recognition,” Adv. Artif. Neural Syst., 2011.
- [6] P. Mehta, “An Efficient Attendance Management System based on Face Recognition using Matlab and Raspberry Pi 2,” Int. J. Eng. Technol. Sci. Res. IJETSR, vol. 3, no. 5, pp. 71–78, 2016.
- [7] S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, and Y. T. Liu, “Facial feature detection using geometrical face model: An efficient approach,” Pattern Recognit., vol. 31, no. 3, pp. 273–282, 1998.
- [8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” IEEE Trans. Pattern Anal. Mach. Intell., 1997.
- [9] T. Ojala, M. Pietikainen, and D. Harwood, “Performance evaluation of texture measures with classification based on Kullback discrimination of distributions,” in

Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on, 1994.

- [10] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” Pattern Recognit., 1996.
- [11] G. J. Edwards, C. J. Taylor, and T. F. Cootes, “Interpreting face images using active appearance models,” Proc. Third IEEE Int. Conf. Autom. Face Gesture Recognit., 1998.
- [12] T. F. Cootes, G. J. Edwards, and C. J. Taylor, Active appearance models. 1998.
- [13] M. Turk and A. Pentland, “Eigenfaces for Recognition,” J. Cogn. Neurosci., vol. 3, no. 1, pp. 71–86, 1991.
- [14] W. Haider, H. Bashir, A. Sharif, I. Sharif, and A. Wahab, “A Survey on Face Detection and Recognition Techniques,” Res. J. Recent Sci., 2014.
- [15] D. L. Swets, “Using discriminant eigenfeatures for image retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 8, pp. 831–836, 1996.
- [16] L. C. Paul and A. Al Sumam, “Face Recognition Using Principal Component Analysis Method,” vol. 1, no. 2012, pp. 135–139, 2012.
- [17] I. Dagher, “Incremental PCA-LDA algorithm,” in CIMSA 2010 - IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Proceedings, 2010.
- [18] Y. Kawaguchi, “Face Recognition-based Lecture Attendance System,” 3rd AEARU ..., no. October, 2005.
- [19] N. Kar, M. K. Debbarma, A. Saha, and D. R. Pal, “Study of Implementing Automated Attendance System Using Face Recognition Technique,” Int. J. Comput. Commun. Eng., vol. 1, no. 2, pp. 100–10

9. APPENDIX I

DETAILS OF PROJECT AND RELEVANCE TO ENVIRONMENT, SAFETY,COST AND MAPPING WITH Pos AND PSO WITH JUSTIFICATION

| Title of the project | Roll no. of students | Project supervisor | Relevance to environment | Relevance to safety | Type |
|---|---|--------------------|--------------------------|---------------------|---------------------------|
| COMPUTER VISION-BASED ATTENDANCE TRACKING WITH VOICE FEEDBACK | 1604-19-735-015 1604-19-735-019 1604-19-735-030 | Dr. Salma Fauzia | 3 | 3 | Application based Project |

Mapping with PO and PSO with justification

3 - Highly Relevant

2 - Moderately Relevant

1 - Less Relevant

10. APPENDIX II

GANTT CHART

