

A PROJECT REPORT ON ATTENDANCE VIA QR CODE & FACE RECOGNITION

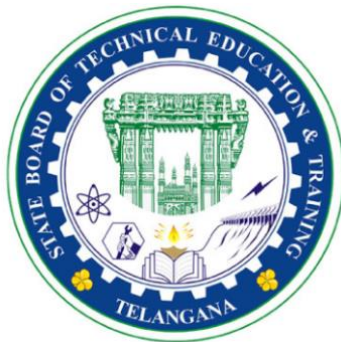
DIPLOMA IN COMPUTER ENGINEERING BY

S. PRAVEEN	21047-CS-012
P. NANDA KISHORE	21047-CS-016
D. VARSHINI SNIGDHA	21047-CS-002
P. SATHWIK	21047-CS-006
K. RAM CHARAN	21047-CS-014
G. CHIRU THEJ	21047-CS-054

Under the Guidance of

G. SREE MADHURI, M. Tech.,

LECTURER IN COMPUTER ENGINEERING



DEPARTMENT OF COMPUTER ENGINEERING
SANJAY GANDHI MEMORIAL GOVERNMENT POLYTECHNIC
Near Ramoji Film City, Abdullapurment, R. R. Dist.- 501505

(Accredited by National Board of Accreditation)

Department of Computer Engineering



CERTIFICATE

This is to certify that the project work entitled “**ATTENDANCE VIA QR CODE & FACE RECOGNITION**” carried out by **S. PRAVEEN, P. NANDA KISHORE, D. VARSHINI SNIGDHA, P. SATHWIK, K. RAM CHARAN, G. CHIRU THEJ** bearing Roll No. **21047-CS-012, 21047-CS-016, 21047-CS-002, 21047-CS-006, 21047-CS-014, 21047-CS-054** in partial fulfilment of the requirements for the award of the degree of Diploma in Computer Engineering is a record of bonafide work carried out by him/her under my guidance.

The results of investigation enclosed in this report have been verified and found satisfactory. The results embodied in this project report have been submitted to any other University or Institute for the award of any other degree of diploma.

Signature of the Guide
G. SREE MADHURI, M. Tech.,

Head of the Department
B. VAJRAIAH, M. Tech.,

External

Principal
CH. V. KRISHNA RAO
M.E., MBA, DIS, FIE FIE(PH.D)

DECLARATION BY CANDIDATE

We **S. PRAVEEN, P. NANDA KISHORE, D. VARSHINI SNIGDHA, P. SATHWIK, K. RAM CHARAN, G. CHIRUTEJ** bearing Roll No **21047-CS-012, 21047-CS-016, 21047-CS-002, 21047-CS-006, 21047-CS-014, 21047-CS-054** in “**ATTENDANCE VIA QR CODE & FACE RECOGNITION**” is done under the guidance of **Mrs G. SREE MADHURI**, Department of Computer Engineering, S.G.M. Government Polytechnic, is submitted in partial fulfilment of the requirements for the award of the Diploma in Computer Engineering.

This is a record of bonafide work carried out by me in S.G.M. GOVERNMENT POLYTECHNIC and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

BY

S. PRAVEEN	21047-CS-012
P. NANDA KISHORE	21047-CS-016
D. VARSHINI SNIGDHA	21047-CS-002
P. SATHWIK	21047-CS-006
K. RAM CHARAN	21047-CS-014
G. CHIRU THEJ	21047-CS-054

CONTEXT

Section	Page no.
Introduction	5-5
Applications of the project to the real world	6-6
Hardware Requirements	7-7
Software Requirements	7-7
Existing System	8-8
Proposed System	8-8
Python	9-10
Machine Learning	11-12
K-Nearest Neighbors (KNN)	13-16
Installed Libraries	17-20
System Diagram	21-21
Source Code	22-25
File Explanation	25-27
Output Screens	27-35
Conclusion	36-36

Abstract

The "Attendance via QR & Face Recognition" system presents an innovative solution for efficient and secure attendance tracking in educational institutions. Leveraging the power of QR codes and advanced facial recognition technology, this system aims to streamline the attendance process for students. Each student is assigned a unique QR code containing essential information such as their name, pin, and branch.

To mark attendance, students are required to scan their individual QR codes using a designated scanner. Subsequently, they must present their faces to the system's facial recognition module. The system then undergoes a verification process, ensuring that the facial features match the information encoded in the QR code. Upon successful verification, the attendance record is automatically logged into an Excel sheet.

This integrated approach not only enhances the speed and accuracy of attendance tracking but also ensures the security and authenticity of the recorded data. The system's reliance on QR codes minimizes the likelihood of errors during the scanning process, while facial recognition adds an additional layer of identity validation. The automated logging of attendance in an Excel sheet facilitates easy data management and analysis for educational institutions.

In summary, the "Attendance via QR & Face Recognition" system offers a robust and reliable solution for modernizing attendance procedures, promoting efficiency, and ensuring the integrity of attendance records in educational settings.

Introduction

In the evolving landscape of attendance management systems, the integration of advanced technologies such as QR code recognition and face detection has become paramount. This project, titled "**Attendance via QR-Code and Face Recognition**," represents a pioneering effort to streamline and enhance the attendance tracking process. By combining the efficiency of QR code data extraction with the precision of face recognition, this system offers a comprehensive solution for attendance management in various domains.

Features of the Project

- **QR Code Integration:** The system initiates attendance tracking by requesting a QR code from the user. QR code data extraction enables seamless identification and retrieval of user information.
- **Face Recognition:** Users can authenticate attendance by presenting their faces to the system. Live face recognition ensures the system only acknowledges and classifies attendance for genuine, in-person participants.
- **Dynamic QR Code Generation:** The system is equipped to generate QR codes for new users, facilitating the on boarding process. For existing users, the system displays their unique QR codes for quick and convenient attendance recording.
- **Error Handling:** In cases where a static image or non-live face is presented, the system promptly detects and flags the attempt as a face recognition error. This ensures the system's robustness against fraudulent attendance practices.
- **CSV File Marking:** Successfully authenticated attendance is logged and marked in a CSV file for convenient record-keeping.

Application of the Project to the Real World

- **Educational Institutions:**
 - Streamlines attendance tracking in schools, colleges, and universities.
 - Integrates with existing processes, reducing manual effort and enhancing accuracy.
- **Corporate Settings:**
 - Facilitates efficient employee attendance management.
 - Seamless integration with corporate systems enhances organizational efficiency.
- **Events:**
 - Provides a reliable solution for tracking attendance at conferences, seminars, and workshops.
 - Enhances event management by automating the attendance process.

Hardware Requirements

- Camera for face recognition
- QR code scanner
- Intel i5 or equivalent processor
- Minimum 8GB RAM
- 500GB SSD storage
- Ethernet or Wi-Fi connectivity
- GPU with CUDA support (optional)

Software Requirements

- Programming language environment (e.g., Python)
- OpenCV library for face recognition
- QR code decoding library
- CSV file handling libraries

Existing System:

Traditional attendance systems often rely on manual methods, such as paper registers or card swiping, leading to inefficiencies, errors, and time-consuming processes. These systems are prone to inaccuracies and lack the sophistication needed to adapt to the dynamic requirements of modern attendance tracking.

Proposed System:

The proposed system leverages QR code technology and face recognition to overcome the limitations of traditional methods. By automating the attendance process, reducing the risk of proxy attendance, and providing a user-friendly interface, it offers an advanced and reliable solution for efficient attendance management in diverse environments. The system's adaptability, accuracy, and real-time functionality make it a significant improvement over existing systems.



Python, a versatile and high-level programming language, has become a cornerstone in the realm of software development, data science, and automation. Created by **Guido van Rossum** in the late 1980s, Python is renowned for its readability and clean syntax, making it an ideal choice for both beginners and seasoned developers.



GUIDO VAN ROSSUM

One of Python's key strengths lies in its extensive standard library, offering a wide array of modules and packages that simplify complex tasks. Its object-oriented programming paradigm promotes code reusability and modularity, fostering efficient and maintainable software development practices.

Python's popularity in data science stems from libraries like NumPy, Pandas, and Matplotlib, empowering scientists and analysts to handle large datasets and visualize results seamlessly. Additionally, frameworks such as Django and Flask facilitate web development, enabling the creation of robust and scalable applications.

FEATURES

- Free and Open Source
- Easy to code
- Easy to Read
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible to Debug
- Python is a Portable language
- Interpreted Language
- Large Standard Library
- Dynamically Typed Language
- Frontend and backend development

ADVANTAGES AND DISADVANTAGES OF PYTHON

Advantages	Disadvantages
Python is easy to learn, making it accessible for beginners.	Python may face speed limitations compared to lower-level languages like C or C++.
Python boasts a large standard library, simplifying complex tasks for developers.	Python is not a top choice for mobile app development due to performance and support issues.
Its versatility extends to various applications, from web development to machine learning.	Its dynamic typing can lead to design constraints and potential runtime errors.
The language offers integration capabilities with other languages and technologies.	Some databases may have better support for languages like Java than Python.
As an open-source language, Python encourages collaboration and continuous improvement.	Python's performance may be limited in CPU-intensive tasks.

Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data. The primary goal of machine learning is to enable computers to automatically learn and improve from experience without being explicitly programmed.

Types of Machine Learning

1. Supervised Learning:

- The algorithm is trained on a labelled dataset, where the input data and corresponding are provided.
- It learns to map the input to the output, making predictions on new, unseen data.

2. Unsupervised Learning:

- The algorithm is given unlabelled data and must find patterns or structures on its own.
- Common tasks include clustering and dimensionality reduction.

3. Reinforcement Learning:

- The algorithm learns by interacting with an environment and receiving feedback in the form of rewards or penalties.
- It aims to learn a sequence of actions that maximizes the cumulative reward.

Advantages and Disadvantages of Machine Learning

Advantages	Disadvantages
Automation: ML can automate complex tasks and make systems more efficient.	Data Dependency: ML models heavily rely on the quality and quantity of training data.
Data-Driven Insights: ML can discover patterns and trends in large datasets that may not be apparent through traditional analysis.	Lack of Interpretability: Some ML models, especially complex ones, can be difficult to interpret, making it challenging to understand their decision-making process.
Adaptability: ML models can adapt to new data, making them suitable for dynamic environments.	Over fitting: Models may perform well on training data but fail to generalize to new, unseen data.
Improved Decision Making: ML algorithms can make predictions and decisions based on data, leading to more informed choices.	Bias: ML models can inherit biases present in the training data, leading to biased predictions.

Model Selection

Choosing the right machine learning model depends on various factors such as the nature of the problem, the type of data, and the desired outcome. Common models include:

- Linear Regression
- Logistic Regression
- Decision Trees
- Random Forest
- Support Vector Machines (SVM)
- Neural Networks

K-Nearest Neighbors (KNN)

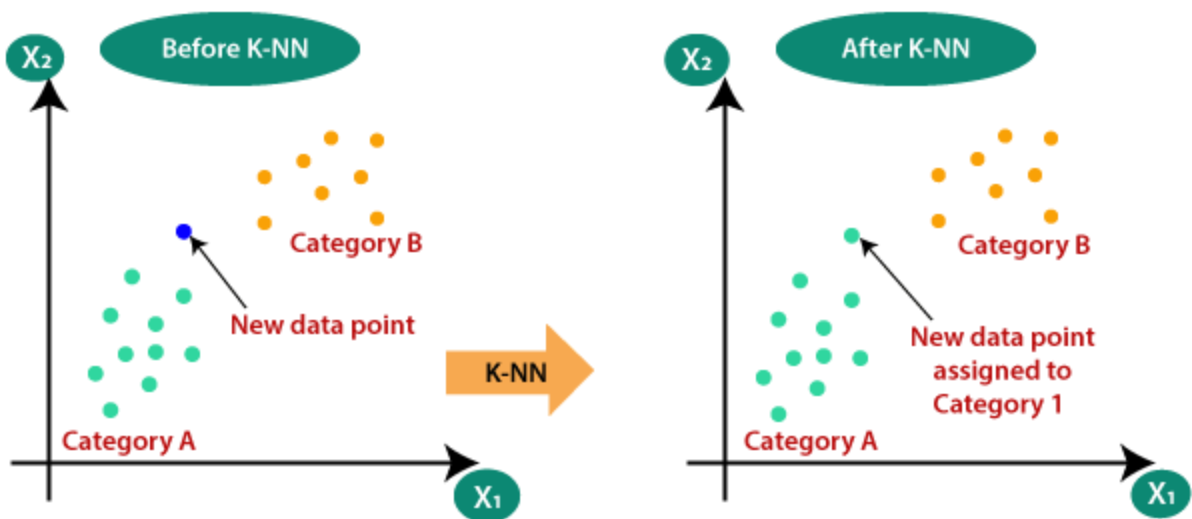
Overview

K-Nearest Neighbors (KNN) is a simple and intuitive algorithm used for both classification and regression tasks. It operates on the principle that data points close to each other in feature space are more likely to belong to the same class or have similar outputs.

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



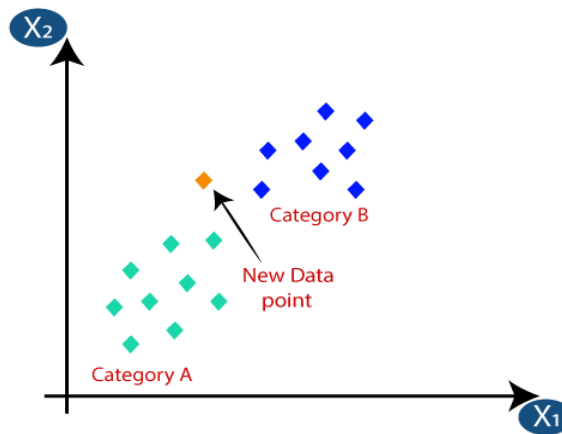
How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

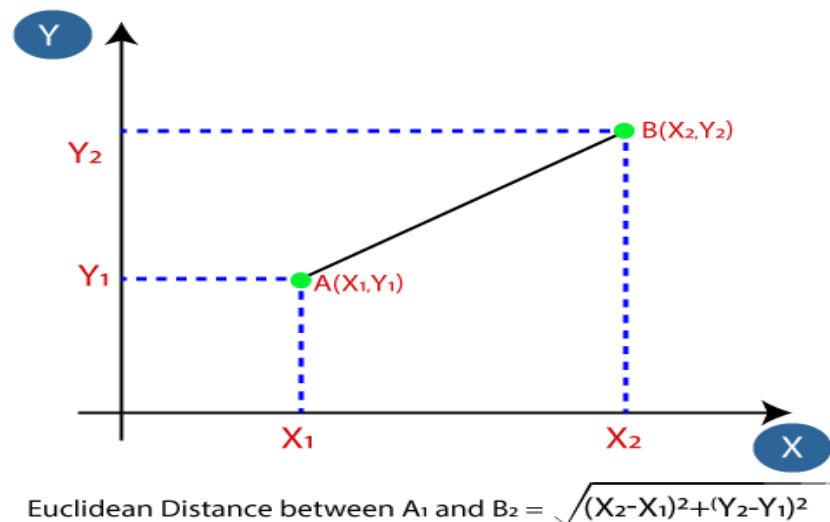
- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.

- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

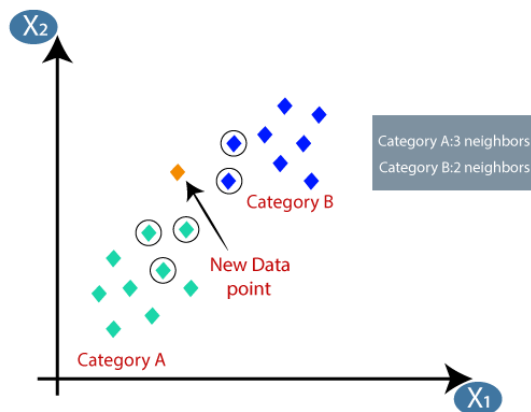
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbors, so we will choose the $k=5$.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

How to select the value of K in the K-NN Algorithm ?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

INSTALLED LIBRARIES

1. QR Code:

Purpose: Generates QR codes, which are two-dimensional barcodes.

Usage: Useful for encoding information like URLs, text, or other data in a visually scan able format.

Example: Creating QR codes for mobile payments, event tickets, or contactless information sharing.

2. OS:

Purpose: Provides a way to interact with the operating system, facilitating file and directory manipulation.

Usage: Used for tasks like file handling, directory navigation, and running system commands.

Example: Listing files in a directory, checking file existence, or executing shell commands.

3. Flask:

Purpose: A web framework for building web applications in Python.

Usage: Enables the creation of web endpoints, handling HTTP requests, and rendering HTML templates.

Example: Building a simple web application for data visualization or providing a RESTful API.

4. CV2 (OpenCV):

Purpose: Open Source Computer Vision library for image and video processing.

Usage: Used in computer vision tasks, image manipulation, object detection, and video analysis.

Example: Face recognition, image filtering, or extracting features from images.

5. Pandas:

Purpose: Provides data structures and functions for efficiently manipulating large datasets.

Usage: Ideal for data cleaning, exploration, and analysis.

Example: Loading and manipulating CSV data, performing statistical analysis on datasets.

6. Numpy:

Purpose: A powerful library for numerical operations and array manipulations.

Usage: Essential for scientific computing and data analysis.

Example: Performing mathematical operations on arrays, linear algebra operations, or statistical analysis.

7. Sklearn (scikit-learn):

Purpose: Machine learning library offering simple and efficient tools for data analysis and modelling.

Usage: Building and training machine learning models, data pre-processing, and model evaluation.

Example: Implementing classification algorithms, regression models, or clustering.

8. Joblib:

Purpose: Library for lightweight pipelining in Python, particularly for parallel processing.

Usage: Useful for efficiently executing tasks in parallel and caching results.

Example: Parallelizing computations, caching expensive function calls, or handling large-scale data processing.

9. Datetime:

Purpose: Provides classes for working with dates and times.

Usage: Used for date and time manipulation, formatting, and calculations.

Example: Recording timestamps, calculating time differences, or formatting dates for display.

Requirements.txt

The requirements.txt file is a commonly used convention in the python programming ecosystem to specify the dependencies of a python project. This file typically contains a list of python packages along with their versions that are required for the project to run successfully.

The use of a requirements.txt file in a python project offers several benefits:

- **Dependency Management**
- **Reproducibility**
- **Collaboration**

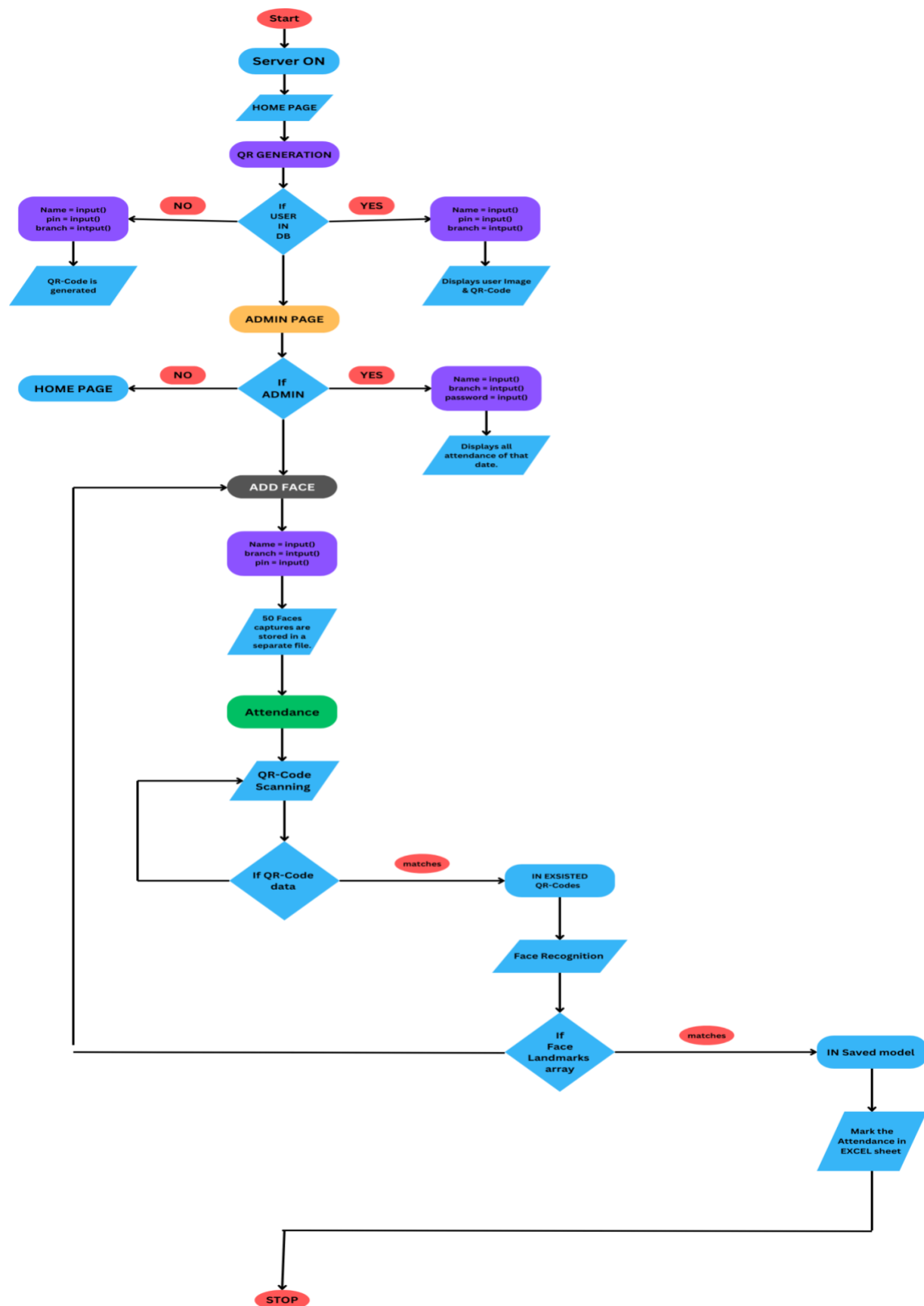
Installations of libraries

To download and install the libraries specified in the requirements.txt file, you can use the pip package manager. Use this command to download all libraries.

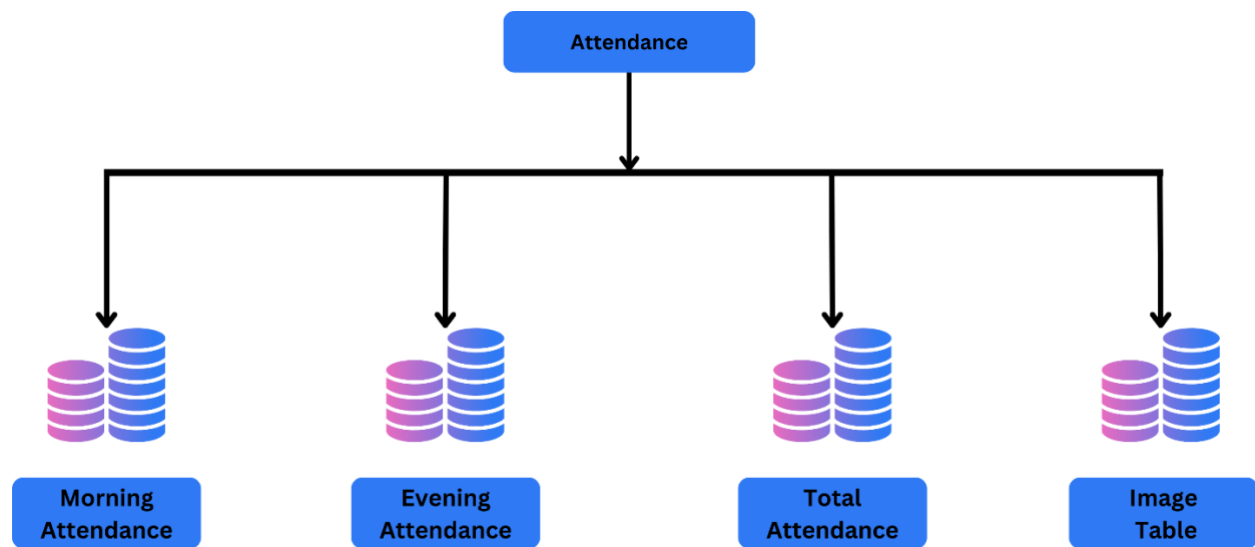
“pip install -r requirements.txt”

To know which libraries are downloaded uses this command **“pip freeze”**

System Diagram



DATABASE SCHEMA



In the Attendance database, there are 4 tables named [Morning Attendance, Evening Attendance, Total Attendance, Image Table].

Morning Attendance Table: [name, pin, branch, initial_time] these are the columns, and the use case is whenever the user takes the attendance before 10. Am, then the attendance will be inserted into this table so that know one can modify it except admin.

Evening Attendance Table: [name, pin, branch, final_time] these are the columns, and the use case is whenever the user takes the attendance before 3.30 pm, then the attendance will be inserted into this table so that know one can modify it except admin.

Final Attendance Table: [name, pin, branch, initial_time, final_time, time difference, status] these are the columns, and the use case is the students who are present in the both morning table and evening table are inserted into the this table and this is the one considered for overall.


Image Table: [name_pin, images_list] these are the columns, and the use case is whenever the student add his face in the interface then the images are the directly inserted into this table in the binary format.


SOURCE CODE


File Structure


 app.py


 Attendance


 evng_attendance


 Attendance-08_12_23.csv


 Attendance-10_12_23.csv


 Attendance-11_12_23.csv


 Attendance-12_12_23.csv


 Attendance-19_05_23.csv


 final_attendance


 Attendance-08_12_23.csv


 Attendance-10_12_23.csv


 Attendance-11_12_23.csv


 Attendance-12_12_23.csv


 Attendance-19_05_23.csv


 mrng_attendance

 Attendance-08_12_23.csv

 Attendance-10_12_23.csv


 Attendance-11_12_23.csv


 Attendance-12_12_23.csv

 Attendance-19_05_23.csv

 attendance.py


 demo.py


 main.py

 models.py

 QR_DATA.txt


 README.md


 scanings.py

 static

 css


 header.css


 face_recognition_model.pkl


 images


 back2.jpg


 camera 1.png


 card_back.jpg


 clg.jpeg

 faces

 ESHWAR_21047-CS-003


 ESHWAR_21047-CS-003_0.jpg


 ESHWAR_21047-CS-003_1.jpg


 ESHWAR_21047-CS-003_29.jpg

.....


 PRAVEEN_21047-CS-012

 PRAVEEN_21047-CS-012_0.jpg

 PRAVEEN_21047-CS-012_1.jpg


 PRAVEEN_21047-CS-012_10.jpg


.....


 form_background.png

 LOGO.jpg


 logo.png


 mainpage_back.png


 QR_CODES

 ESHWAR_21047-CS-003.png


 NANDA KISHORE_21047-CS-016.png


 NAVYA_21047-CS-011.png


 PRAVEEN_21047-CS-012.png

 SAMEER_21047-CS-043.png

 qr_scanner.png


 students


 no_user.png


 PRAVEEN_21047-CS-012.png


 js


 common.js


 templates

 add_face.html

 data.html


 evening.html

 faces.html










 face_att.html

 haarcascade_frontalface_default.xml

 home.html

 morning.html

 qr.html

-  qr_scanning.html
-  records.html
-  signin_page.html
-  signup_page.html
-  smile.xml
-  statistics.html
-  streaming.html
-  total_attendance.html
-  validation.html

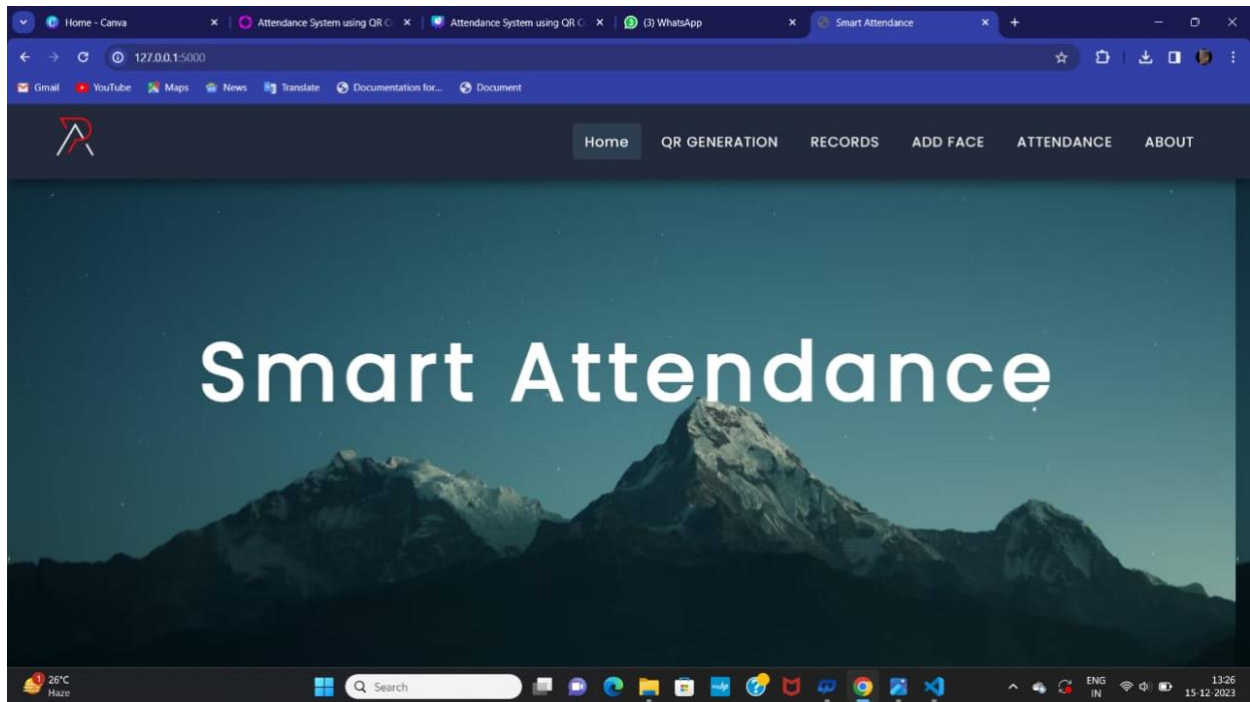
File Explanation

- 1. app.py:** Main application file, likely containing the core logic and routing for the attendance management system.
- 2. Attendance/evng_attendance/.csv:** Evening attendance records for various dates.
- 3. Attendance/final_attendance/.csv:** Finalized attendance records for various dates.
- 4. Attendance/mrng_attendance/.csv:** Morning attendance records for various dates.
- 5. attendance.py:** Module for handling attendance-related functionalities.
- 6. demo.py:** Demo module, possibly showcasing features or providing usage examples.
- 7. main.py:** Main execution file for the application.
- 8. models.py:** File likely containing data models or structures used in the application.
- 9. QR_DATA.txt:** Text file containing QR code data.
- 10. README.md:** Documentation file providing information and instructions for using the application.
- 11. scanings.py:** Module likely related to scanning functionality.
- 12. static/css/header.css:** Stylesheet for the application's header.
- 13. static/face_recognition_model.pkl:** Face recognition model file.
- 14. static/images/:** Directory containing various images used in the application, such as logos and background images.

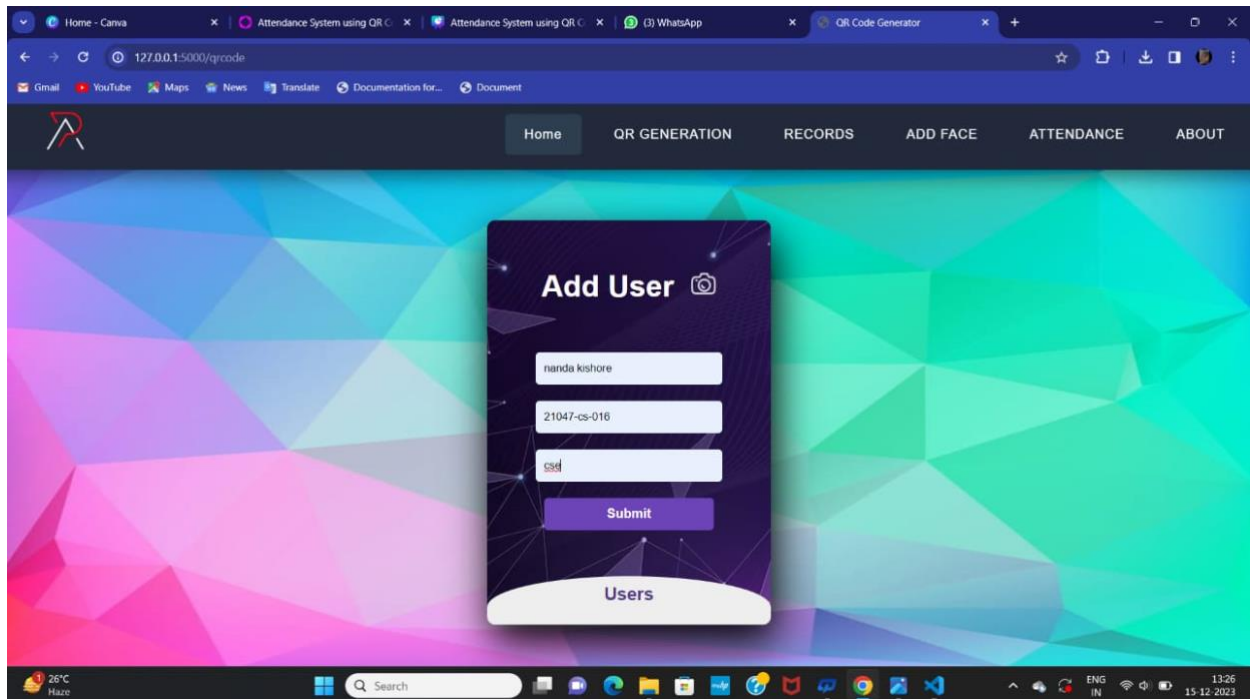
15. **static/images/faces/:** Directory containing individual images of students' faces.
16. **static/js/common.js:** JavaScript file with common functionalities for the application.
17. **templates/:** Directory containing HTML templates for different pages of the application.
18. **templates/add_face.html:** HTML template for adding a new face to the system.
19. **templates/data.html:** HTML template for displaying data.
20. **templates/evening.html:** HTML template for evening attendance.
21. **templates/faces.html:** HTML template for managing faces.
22. **templates/face_att.html:** HTML template for face attendance.
23. **templates/home.html:** HTML template for the home page.
24. **templates/morning.html:** HTML template for morning attendance.
25. **templates/qr.html:** HTML template for QR code functionality.
26. **templates/qr_scanning.html:** HTML template for QR code scanning.
27. **templates/records.html:** HTML template for displaying records.
28. **templates/signin_page.html:** HTML template for the sign-in page.
29. **templates/signup_page.html:** HTML template for the sign-up page.
30. **templates/statistics.html:** HTML template for displaying statistics.
31. **templates/streaming.html:** HTML template for streaming functionality.
32. **templates/total_attendance.html:** HTML template for total attendance.
33. **templates/validation.html:** HTML template for validation.
34. **static/js/common.js:** Common JavaScript functions used across the application.

Output Screens

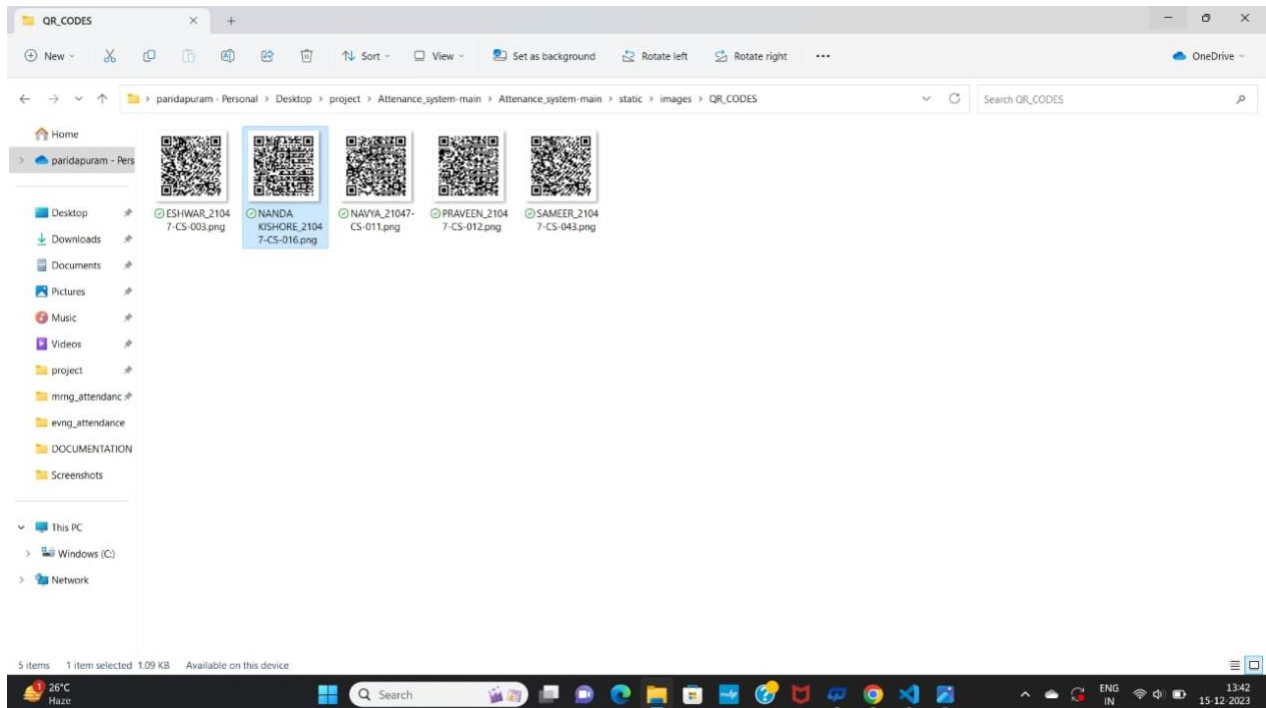
Home Page



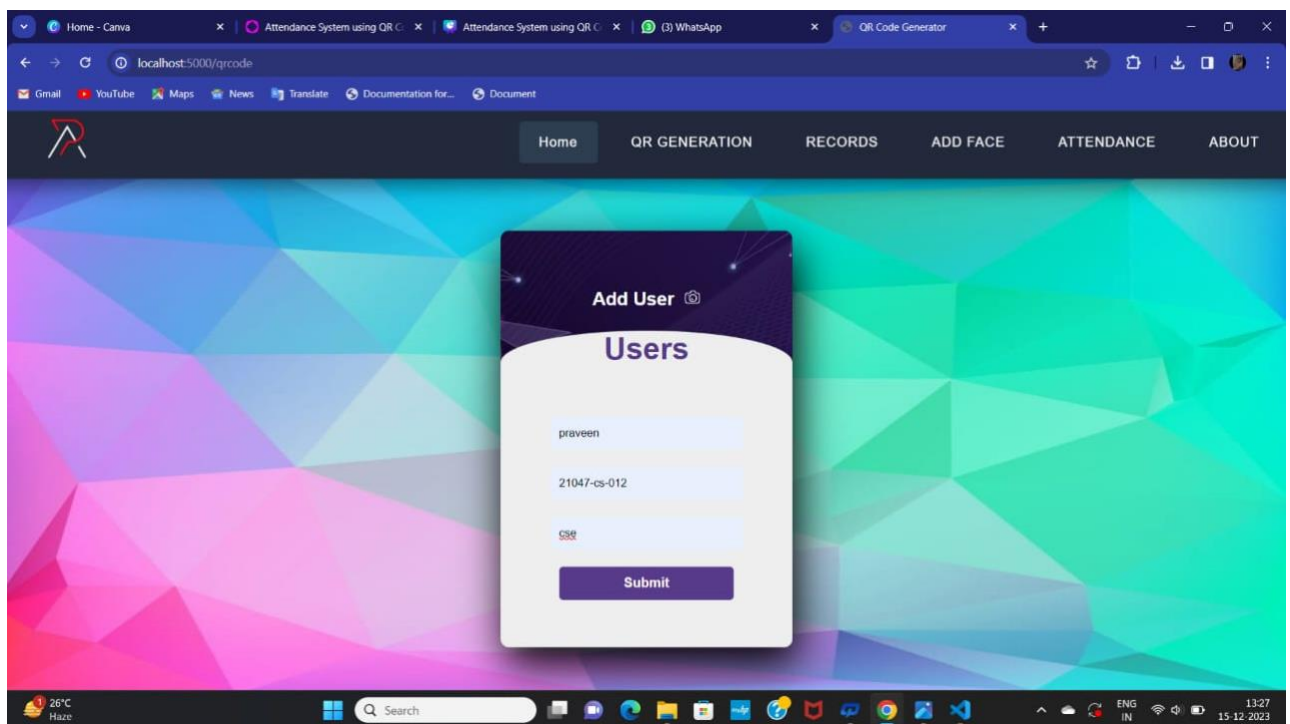
QR Code Generation

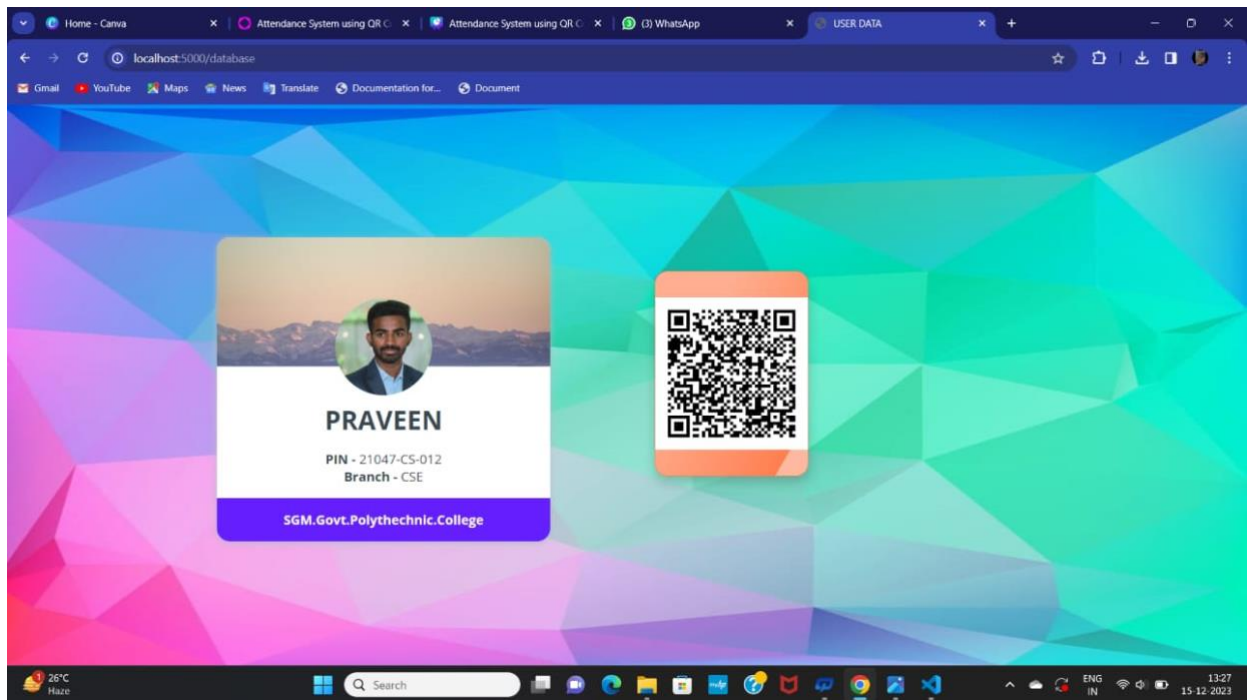


Generated QR Codes

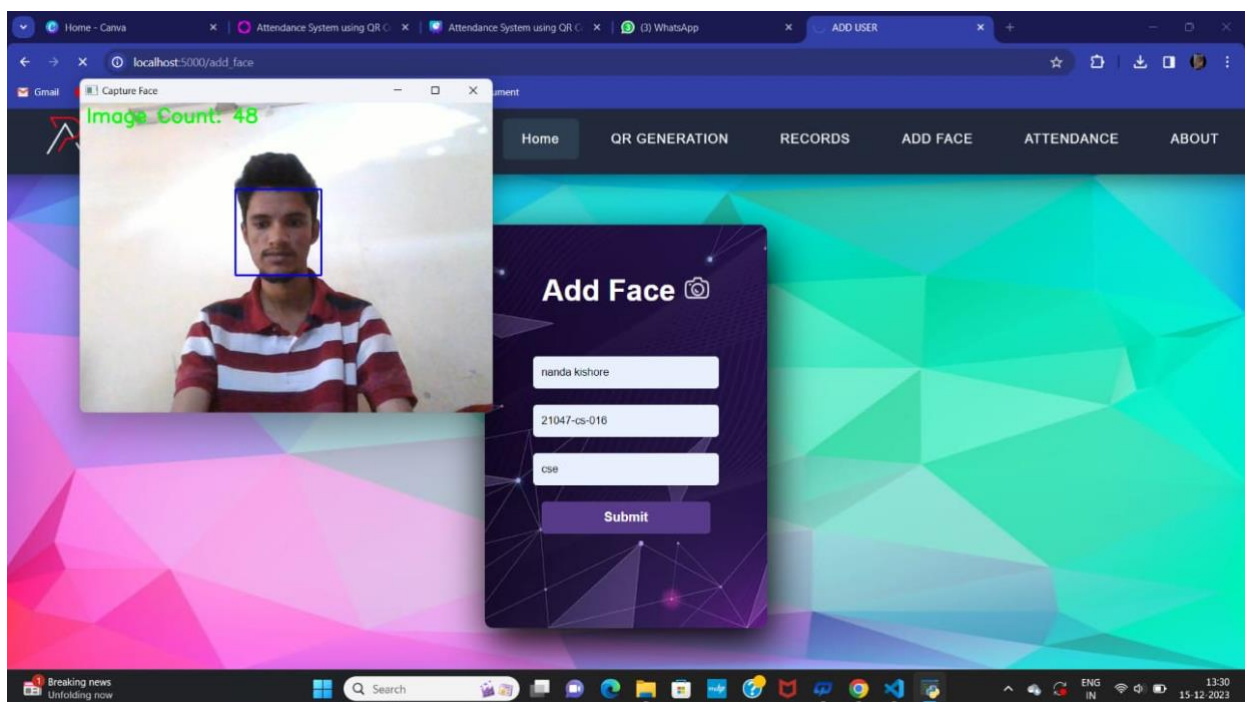


Checking QR Code in STORED DATA

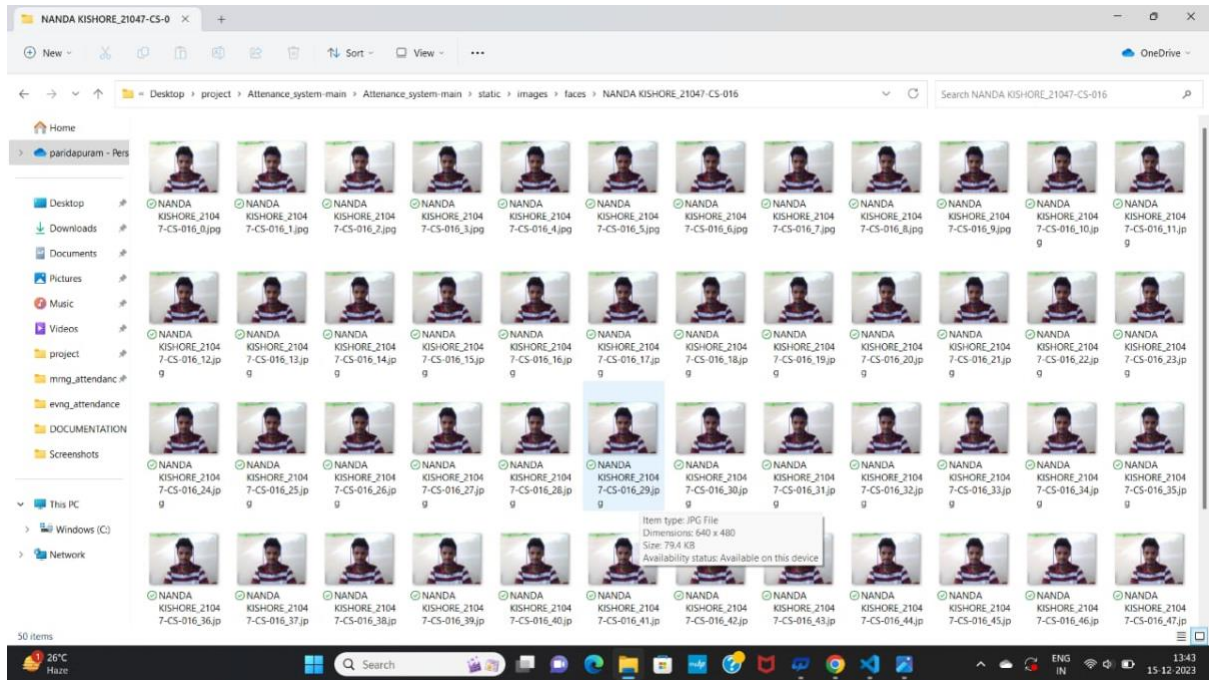




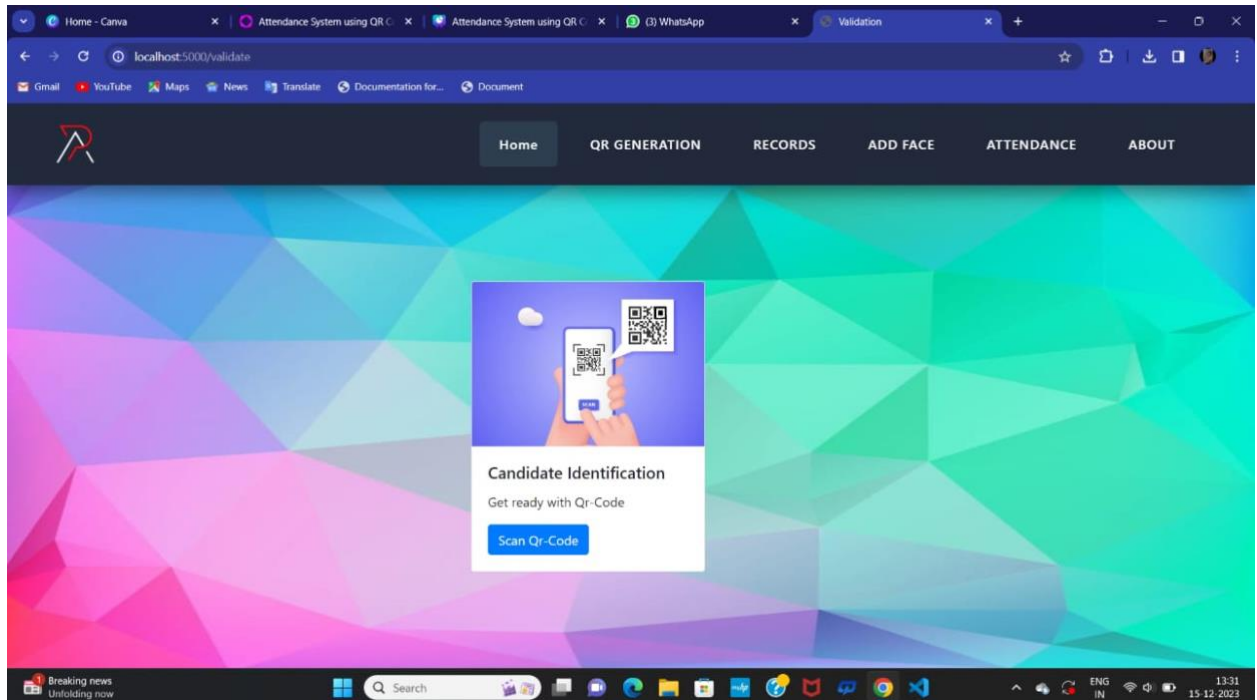
Adding Face



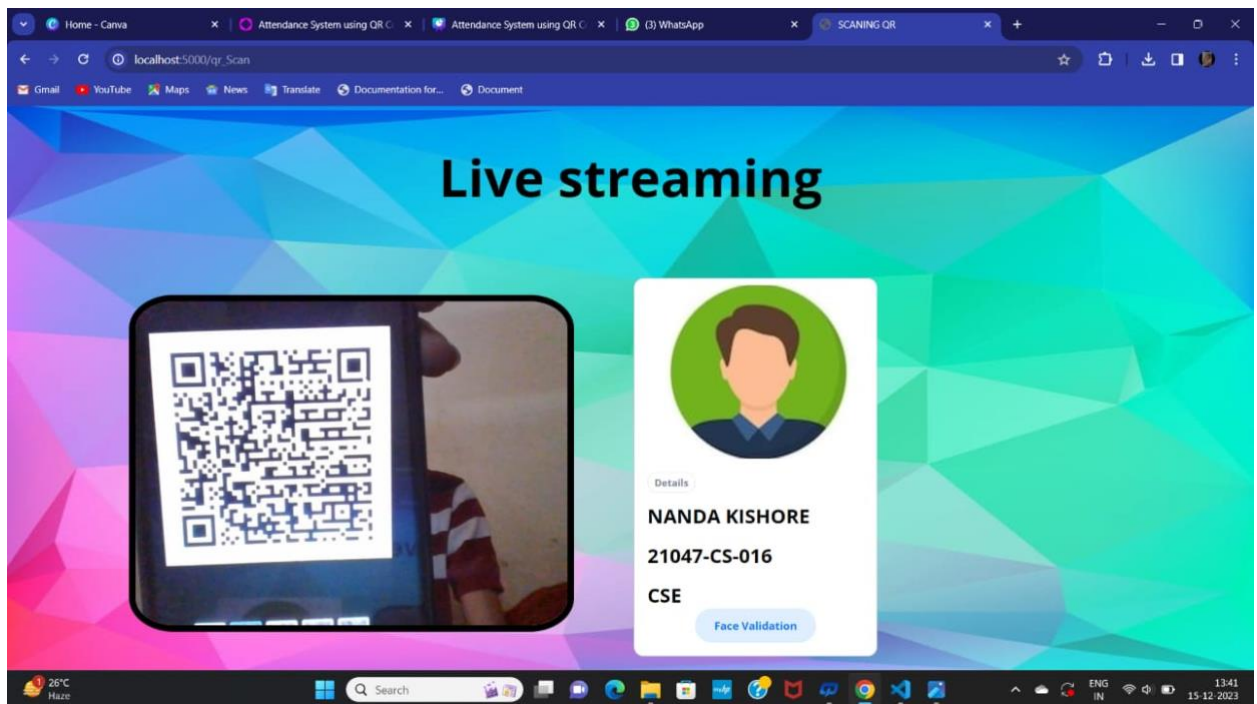
Faces Captured



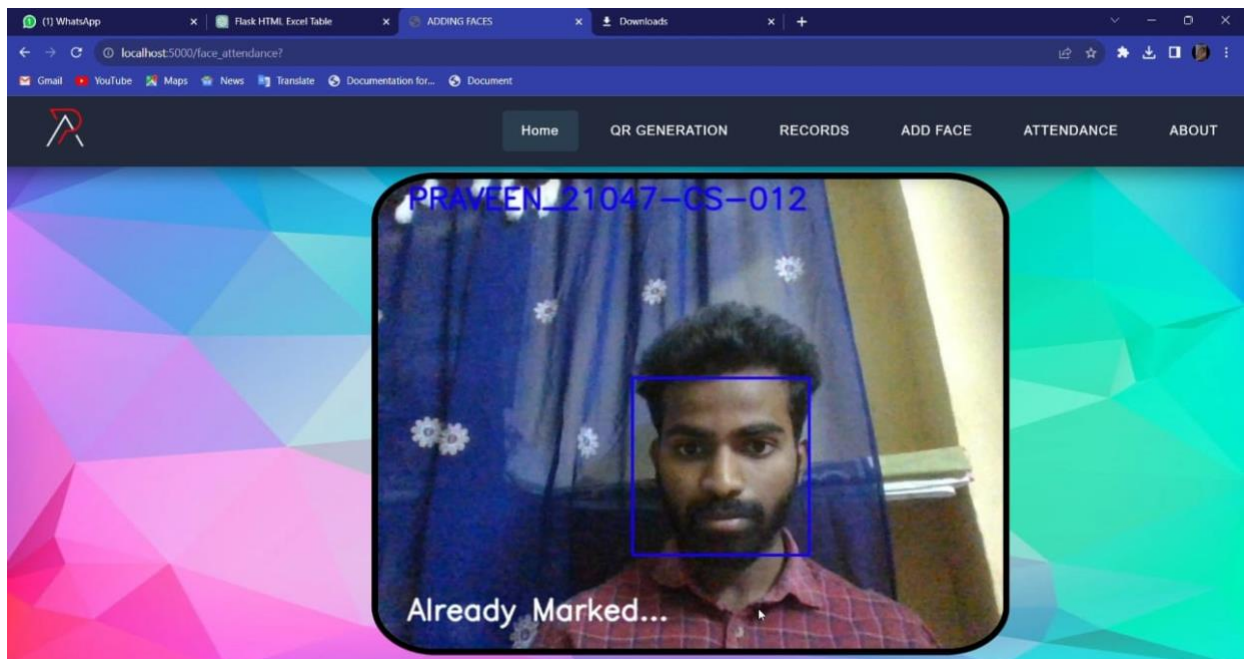
Attendance



Scanning QR Code



Face Validation



Records

Morning Attendance

The screenshot shows a web browser window with the URL `127.0.0.1:5000/morning`. The page has a navigation bar with links: **Back**, **Morning** (active), **Evening**, **Total**, and **Statistics**. Below the navigation bar, there is a header section with the Sanjay Gandhi Memorial Government Polytechnic logo, the name "Sanjay Gandhi Memorial Government Polytechnic Abdullapurmet", the department "Department : computer science engineering", the word "Attendance" in large blue font, and the date "Date : 15/12/2023". A table displays the morning attendance records:

Name	Pin	Branch	Initial_time
PRAVEEN	21047-CS-012	CSE	14:29:07
ESHWAR	21047-CS-003	CSE	14:29:56

The background of the page features a photograph of the college building. The Windows taskbar at the bottom shows the date as 15-12-2023 and the time as 15:14.

Evening Attendance

The screenshot shows the same web application but for the evening session, with the URL `127.0.0.1:5000/evening`. The navigation bar now has **Evening** as the active link. The header section remains the same, but the word "Attendance" is still present. A table displays the evening attendance records:

Name	Pin	Branch	Final_time
NANDA KISHORE	21047-CS-016	CSE	14:30:40
ESHWAR	21047-CS-003	CSE	14:32:13
PRAVEEN	21047-CS-012	CSE	14:32:45

The background image and Windows taskbar are identical to the morning attendance screenshot.

Final Attendance

Sanjay Gandhi Memorial Government Polytechnic Abdullapurmet

Department : COMPUTER SCIENCE ENGINEERING **Attendance** Date : 15/12/2023

Name	Pin	Branch	Initial_time	Final_time	Time_difference	Status
PRAVEEN	21047-CS-012	CSE	14:29:07	14:32:45	4.0	ERROR
ESHWAR	21047-CS-003	CSE	14:29:56	14:32:13	2.0	ERROR
NANDA KISHORE	21047-CS-016	CSE	NaN	14:30:40	NaN	NaN

Excel Records

Morning Attendance

Attendance-15_12_23

Name	Pin	Branch	Initial_time
PRAVEEN	21047-CS-CSE		14:29:07
ESHWAR	21047-CS-CSE		14:29:56

Evening Attendance

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Name	Pin	Branch	Final_time																	
1	NANDA KI	21047-CS-CSE		14:30:40																	
2	ESHWAR	21047-CS-CSE		14:32:13																	
3	PRAVEEN	21047-CS-CSE		14:32:45																	
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					

Total Attendance

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Name	Pin	Branch	Initial_time	Final_time	Time_diff	Status														
1	PRAVEEN	21047-CS-CSE		14:29:07	14:32:45	4	ERROR														
2	ESHWAR	21047-CS-CSE		14:29:56	14:32:13	2	ERROR														
3	NANDA KI	21047-CS-CSE		14:30:40																	
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					

Conclusion

The "Attendance via QR-Code and Face Recognition" project represents a significant leap in attendance management, seamlessly merging QR code technology and live face recognition. Its dynamic QR code generation, robust error handling, and applicability across education, corporate, and event sectors make it a cutting-edge solution for efficient and accurate attendance tracking. This project marks a step towards a technologically advanced and user-friendly future in attendance management systems.

Reference

Python (<https://python.org>)

Scikit-learn (<https://scikit-learn.org/stable/>)

Chatgpt (<https://chat.openai.com/>)

Pandas (<https://pandas.pydata.org/>)

Opencv (<https://opencv.org/>)

Joblib (<https://joblib.readthedocs.io/en/stable/>)

QR code (<https://pypi.org/project/PyQRCode/>)

Numpy (<https://numpy.org/>)