DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies - Nuzvid,

Eluru, Andhra Pradesh – 521202.

STUDENT MANAGEMENT SYSTEM (SMS)
LOGIN BY
USING FACE RECOGNITION

A Project Progress Report

Submitted in partial fulfilment for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

J. PAVAN SANKAR	N180109
K. SRINIVASA RAO	N180083
CH. SAI	N180082
M. THANUJA	N180616
J. NANDINI	N180429

Under the Esteem Guidance of

Mrs. N. SWATHI

Assistant professor in Department of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid,

Eluru, Andhra Pradesh – 521202.

CERTIFICATE OF COMPLETION

This is to certify that the work entitled, **"STUDENT MANAGEMENT SYSTEM (SMS) LOGIN BY USING FACE RECOGNITION"** is the bonafied work of **J. PAVAN SANKAR (ID No: N180109), K. SRINIVASA RAO (ID No: N180083), CH. SAI (ID No: N180082), M. THANUJA (ID No: N180616), J. NANDINI (ID No: N180429)** carried out under my guidance and supervision for 3rd year mini project of Bachelor of Technology in the department of Computer Science and Engineering under RGUKT IIIT, Nuzvid. This work is done during the academic session February 2023 – July 2023, under our guidance.

……..

Mrs. N. Swathi
Assistant Professor,
Department of CSE,
RGUKT - Nuzvid.

……..

Dr. Sadu Chiranjeevi
Assistant Professor,
Head of the Department,
Department of CSE,
RGUKT - Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid,

Eluru, Andhra Pradesh – 521202.

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, **"STUDENT MANAGEMENT SYSTEM (SMS) LOGIN BY USING FACE RECOGNITION"** is the bonafied work of **J. PAVAN SANKAR (ID No: N180109), K. SRINIVASA RAO (ID No: N180083), CH. SAI (ID No: N180082), M. THANUJA (ID No: N180616), J. NANDINI (ID No: N180429)** and here by accord our approval of it as a study carried out and presented in a manner required for it's acceptance in 3rd year of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

…….……

Mrs. N. Swathi

Assistant Professor,
Department of CSE,
RGUKT - Nuzvid.

..……

Project Examiner

RGUKT - Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid,

Eluru, Andhra Pradesh – 521202.

DECLARATION

We "**J. PAVAN SANKAR (ID No: N180109), K. SRINIVASA RAO (ID No: N180083), CH. SAI (ID No: N180082), M. THANUJA (ID No: N180616), J. NANDINI (ID No: N180429)** here by declare that the project report entitled **"STUDENT MANAGEMENT SYSTEM(SMS) LOGIN BY USING FACE RECOGNITION"** done by us under the guidance of Mrs. N. Swathi, Assistant Professor is submitted for the fulfilment of mini project during the academic session February 2023 - July 2023 at RGUKT - Nuzvid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Date: 20-07-2023

Place: Nuzvid

J. PAVAN SANKAR	N180109
K. SRINIVASA RAO	N180083
CH. SAI	N180082
M. THANUJA	N180616
J. NANDINI	N180429

ACKNOWLEDGEMENT

We would like to express our profound gratitude and deep regards to our guide **Mrs. N. Swathi** mam for her exemplary guidance, monitoring and constant encouragement to us throughout the B. Tech course. We shall always cherish the time spent with her during the course of this work.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **"STUDENT MANAGEMENT SYSTEM LOGIN BY USING FACE RECOGNITION"**.

We express our gratitude to Dr. Sadu Chiranjeevi (HOD of CSE) and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

TABLE OF CONTENTS

Title page	Page no.
Abstract	7
Introduction	7
Existing Works	8
Proposed Method	9
Experimental Results	22
Conclusion & Future Scope	23
References	23

STUDENT MANAGEMENT SYSTEM LOGIN BY USING FACE RECOGNITION

ABSTRACT

The aim of this project is to automate SMS login through face recognition. We have a password-protected login system for SMS at our University. However, despite being password-protected, students were able to discover the password through the exam files and use it to log into other accounts, access their results and details, or change profiles to make fun of them. Here the student needs to manually enter a username and password to log into SMS. So, we are going to give a face recognition-based login which takes the user image input through a webcam and compares it with the details of the database and if a match is found he will be logged into the SMS automatically. Here we used the face recognition module for recognizing faces and the selenium library for browser automation. This model can apply to various websites for automatic login. Here we took images of all 2018 batch students (1191 students). With this we automated SMS login through face recognition by ensuring security.

Keywords: Student Management System, face recognition, face encodings, Fernet, MongoDB, selenium

1. INTRODUCTION

The combination of face recognition with automation offers a wide range of practical and efficient use cases. It enables automated access control, where authorized individuals can effortlessly gain entry to secured areas without the need for physical credentials. In our university, we have password-based login system for SMS. Even though it has password authentication, it was not secure as we can get passwords of all users from exam files. As security is a major concern while logging into our account, it was preferable to choose other modes of login. Somewhat, it provides authentication but not to the extent of any other like face recognition or voice recognition-based login. But face recognition-based login is better than voice-based login as our voice may change due to seasons or some reasons. So, we are

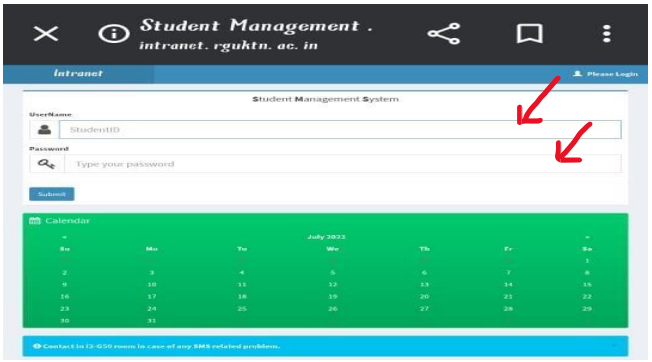
here with the face recognition-based login, as it authenticates through the user's face. We used the face recognition library which is the world's simplest one, for face recognition. The face recognition library of Python is a wrapper of Dlib which is a C++ library that performs tasks related to machine learning and computer vision. Here for face recognition, it uses a pre-trained CNN model to compute face encodings, a vector of 128 measurements. These encodings capture the essential facial characteristics and patterns that are crucial for accurate face recognition. At first, we stored the encodings of all student faces in MongoDB, which is a document-oriented database. We also stored SMS passwords of all users by encrypting them through the Fernet algorithm which is an AES (Advanced Encryption Standard) algorithm that works in CBC (Cipher Block Chaining) mode. So, whenever we perform real-time face recognition through a webcam, the encodings of the person in the webcam are compared with the encodings stored in the database. If a match is found then the person will be logged into the SMS with the aid of automation provided by the selenium module. For a match, it first fetches the details of the username and encrypted password from the database, and then the password it is decrypted using the Fernet. Then with the help of the selenium module, we open the browser to the respected URL and then send username and password from Python code resulted in logging into the respected person's SMS. If suppose multiple persons were identified in the webcam then it shows all the identified users as options. Then the selected user will be logged into the SMS after entering the OTP which was sent to their mail for reconfirmation. The overall process of recognition and browser automation results in enhancing security by reducing human effort in the manual entry of details on the login page. So, the combined version of face recognition and browser automation can be applicable in various works like automatically filling exam details[1] or forms (of the same structure) through face recognition. For our project, we collected photos of all 2018 batch students (total of 1191) using the requests module and took the passwords from exam files. In the end, we made a successful attempt to log users of RGUKT into SMS through face recognition.

2. EXISTING WORKS

For RGUKT SMS[2] we have a password-based login system which takes the username and password for login.

Limitations:

- It was not secure, as we can login to some other account, if we know their login details.
- Passwords were not securely stored as they can be accessible through exam files.

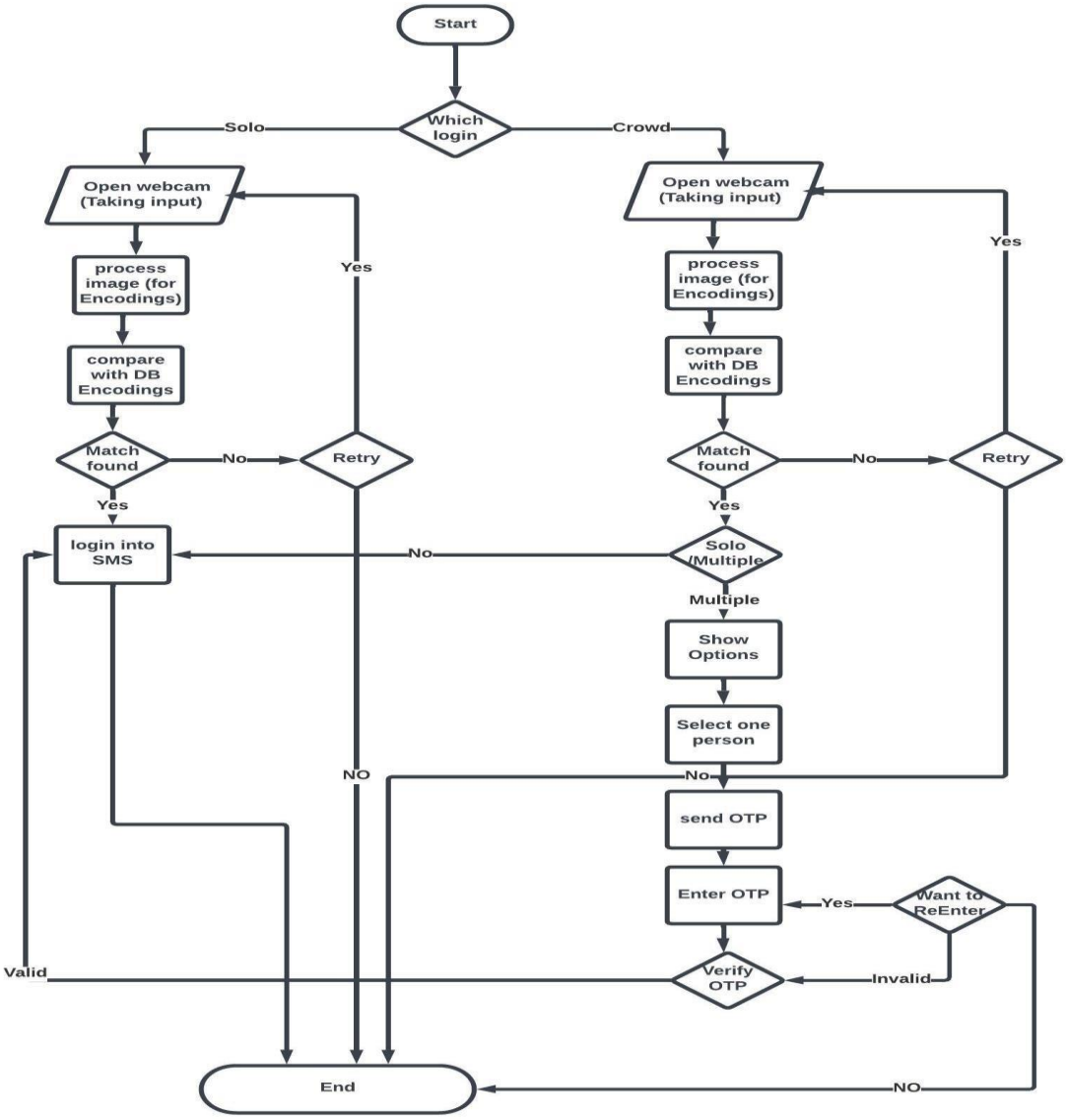


3. PROPOSED METHOD

We proposed a face recognition-based login system for our SMS to ensure security and to reduce human effort for manual entry of details at the login page. This system just tries to recognize the faces which it encounters through the webcam and logs into his/her SMS by fetching the details from the database. First, we encode images of all of the students, and store them with their respective IDs in the database. We also store the passwords of the students in encrypted format. Then at the time of recognition, we try to encode the faces of the webcam and compare them with the stored encodings. If a match is found we fetch his/her encrypted password and decrypt it using the fernet algorithm which runs in CBC mode of AES. Then we open the browser to the SMS site, followed by feeding the username and password to it and then logging into SMS through browser automation i.e., with the help of selenium.

3.1 Flowchart

The following is the flowchart of our project.



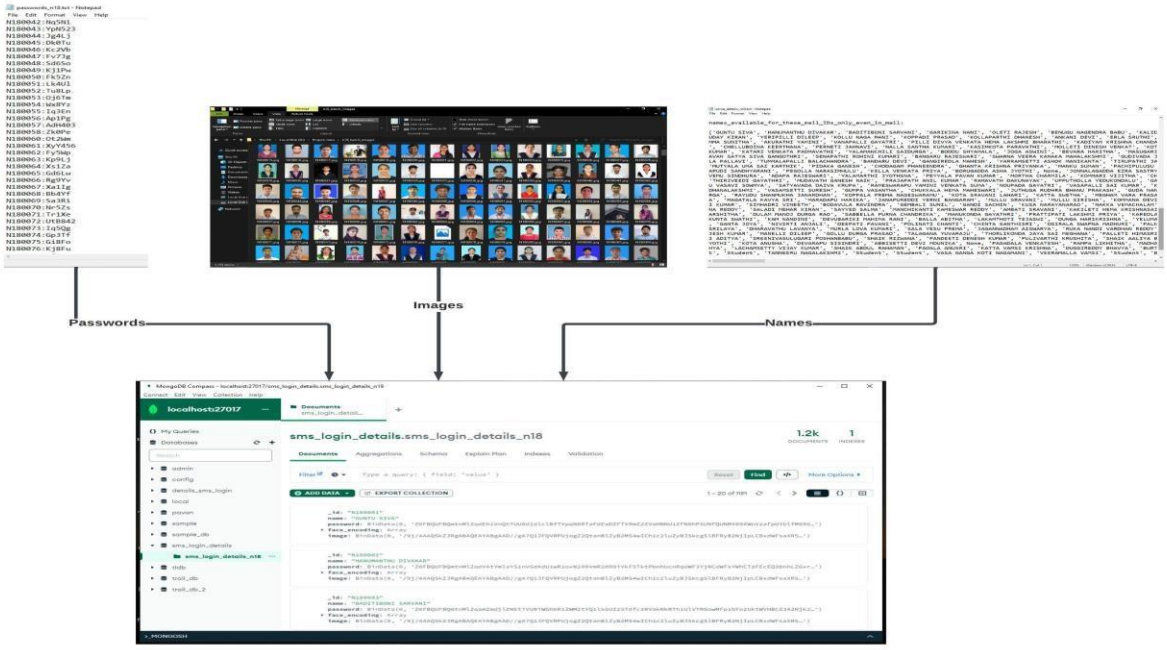
We store encoded images, encrypted passwords and names of 2018 batch students in MongoDB.

cryptography:

The cryptography library in python provides algorithms for encryption, decryption, hashing, and digital signatures. We used 'Fernet'(CBC mode of AES) class of cryptography.fernet module to encrypt passwords of all students. Then we stored them in the database.

pymongo:

One of the essential software components in our project is PyMongo, the official python driver for MongoDB. PyMongo acts as a bridge between our python code and the MongoDB database, facilitating seamless communication and data management. It enables us to establish connections, perform CRUD (create, read, update, delete) operations, and manage student data efficiently. With PyMongo, we can leverage the powerful features of MongoDB, such as flexible document storage, indexing, and data validation, to ensure secure and reliable storage of student information. This component plays a crucial role in the overall functionality and data management of our student management system.

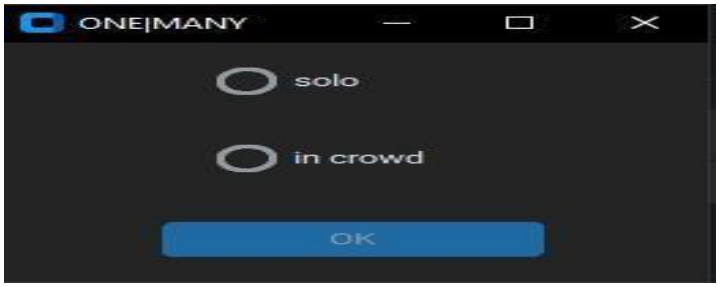


customtkinter:

In our project, we used CustomTkinter for the graphical user interface (GUI). CustomTkinter, a modified version of Tkinter offers enhanced flexibility and control in designing the visual elements and layout of our GUI. With CustomTkinter, we have created interactive components such as widgets, windows, and top levels. This library has enabled us to deliver a visually appealing and user-friendly interface, enhancing the overall user experience.

Start application:

➔ Select whether you are solo or in crowd



OpenCV:

OpenCV is widely used for computer vision and image processing tasks. It is a vital tool in our project, providing image processing capabilities. It enables us to capture input from students through webcam, perform real-time face detection, and annotate detected faces with text overlays and also uses rectangles around detected faces, providing a clear indication of the face detection process. Additionally, OpenCV facilitates image reading and writing, allowing us to process images for face recognition. By leveraging OpenCV's functionality, we enhance the user experience and enable essential features in our project.

face_recognition:

face_recognition library is a key component in our project, enabling the system to identify individuals based on their facial features. face_recognition module of face_recognition library detects and analyses facial landmarks, such as eyes, nose, chin and mouth to create unique facial encodings. These encodings are then compared against the stored encodings to determine the identity of a person. It provides an additional layer of security, ensuring only authorized individuals can access the student management system. It simplifies the process of face recognition, utilizing a pre-trained deep learning model. With this module, developers can easily perform face detection, face encoding, and face recognition operations. face recognition relies on the installation of Dlib and CMake. Dlib is a powerful C++ library that provides advanced tools, including facial recognition capabilities. CMake on the other hand, is a cross-platform build system that aids in the compilation and configuration of software projects. In our case CMake compiles Dlib. face_recognition need Dlib and Dlib need CMake.

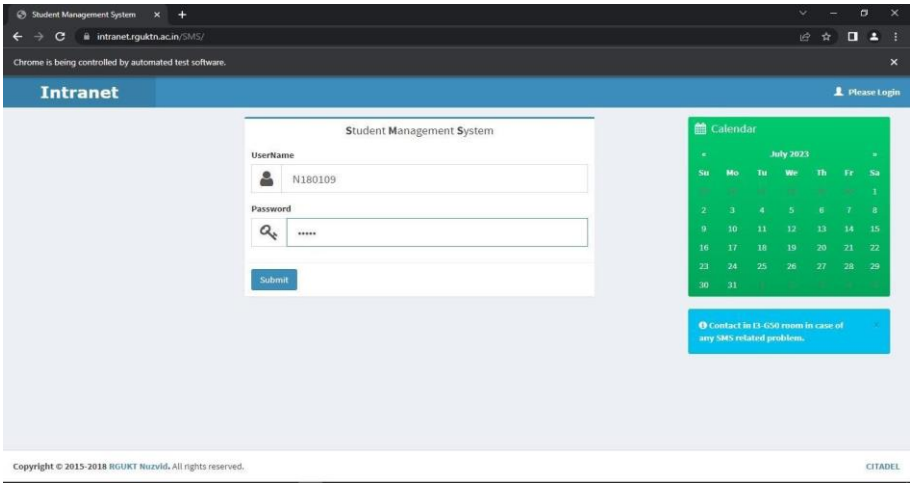
Case-1: Solo login

➔ Match found

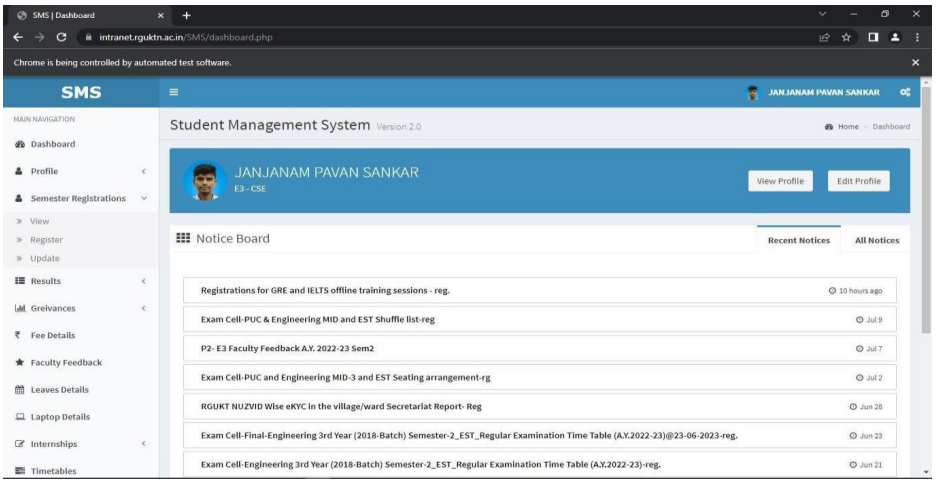
selenium:

One of the key software components in our project is Selenium, used for web browser interactions. By integrating selenium into our project, we enable automated login functionality on the website. This component allows us to input the student ID and password programmatically, eliminating the need for manual login. With Selenium, we can replicate human-like interactions with the web browser, ensuring efficient and accurate login for students.

➔ entering details into SMS automatically



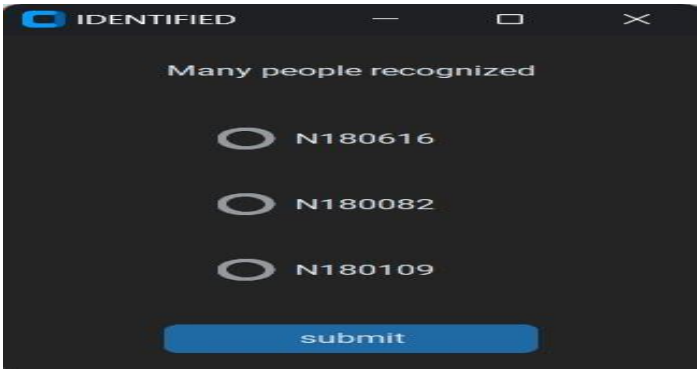
➔ Login successful



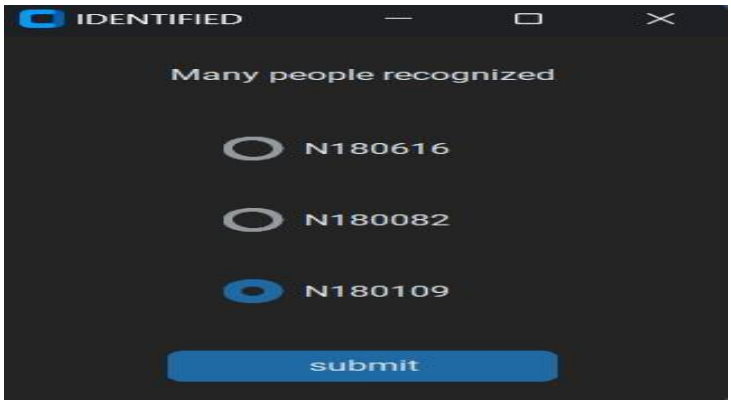
Case-2: In Crowd

➔ Many persons recognised

➔ Identified IDs shown in options



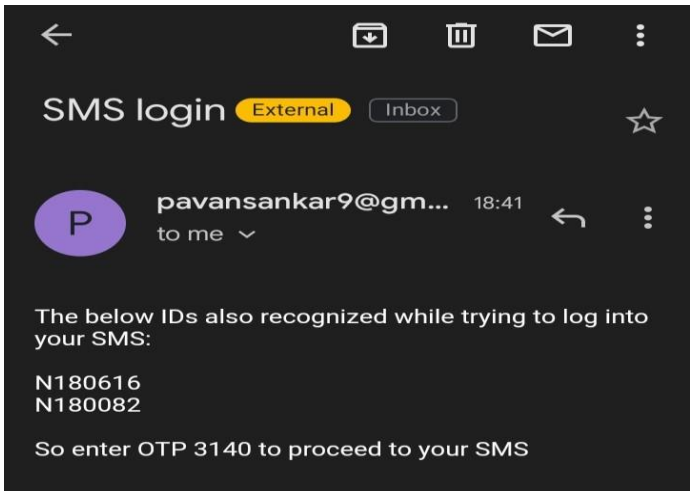
➔ Selecting one ID



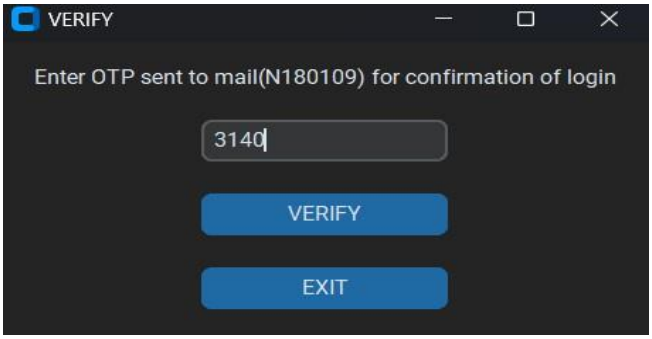
smtplib:

smtplib is a python library that provides a simple way to send an email using SMTP (simple mail transfer protocol). By incorporating Transport Layer Security (TLS) in our email communication, we ensure secure and encrypted transmission of emails to students. This component establishes a secure connection between our python program and the email server, encrypting the data transmission to safeguard the privacy and integrity of the information being sent. The utilization of TLS in conjunction with the SMTP library ensures that student emails are sent securely, providing an additional layer of protection for sensitive information shared through email communication within our student management system.

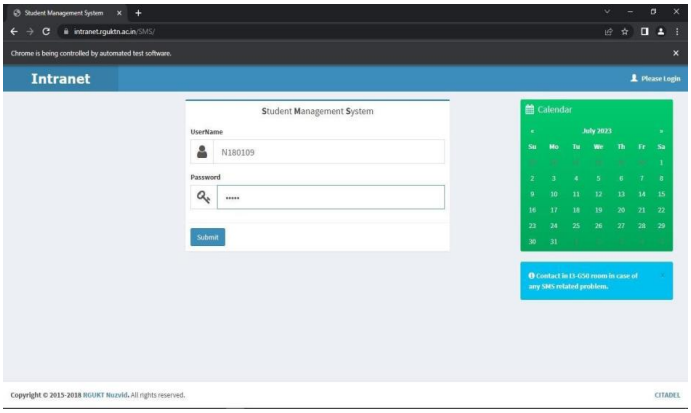
➔ Mail sent to the selected ID



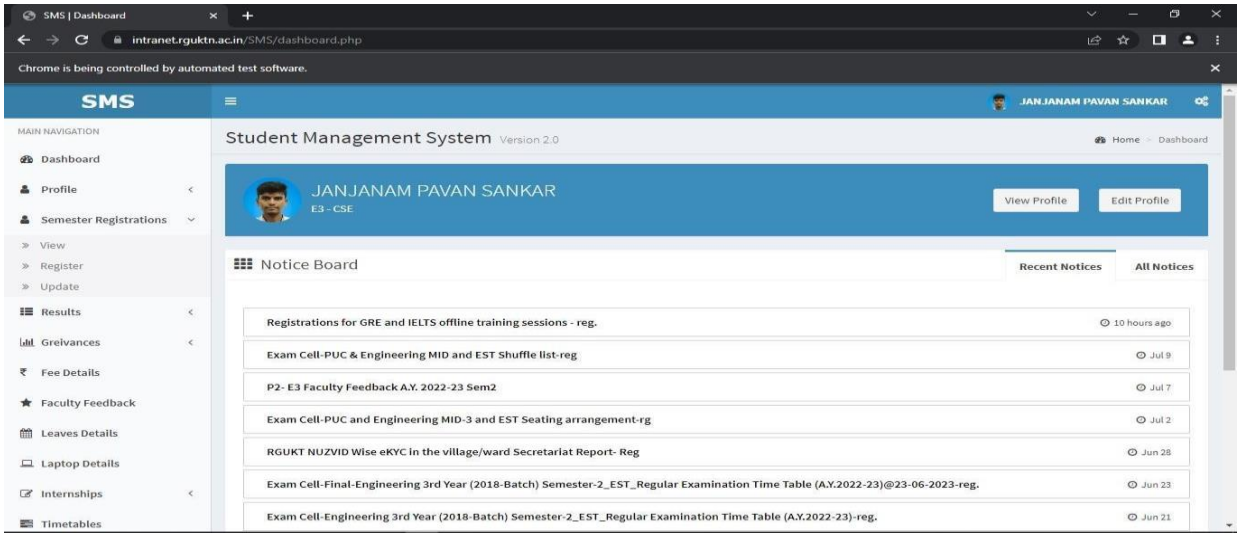
➔ Entering OTP for reconfirmation



➔ If entered OTP is valid, then it will directly log into the SMS



➔ Login Successful



3.2 ALGORITHMS

Let's discuss about the algorithms which were used by face_recognition module form Dlib to perform face recognition.

Face recognition: face detection → face encoding → face comparison

face_recognition module uses HOG algorithm of Dlib to do face detection

face_recognition module uses pre-trained CNN algorithm of Dlib to give face encodings

3.2.1 Histogram of Oriented Gradients (HOG): The HOG algorithm is used for face detection in the face_recognition module. It extracts features based on the distribution of gradients and edge orientations in an image. By analysing the gradients and orientations, the HOG algorithm can identify regions in an image that potentially contain faces.

Face detection using Histogram of Oriented Gradients:

- ➔ Preprocess the input image: The input image is typically pre-processed by resizing, converting to grayscale, and applying any necessary enhancements.
- ➔ Apply the HOG algorithm: The HOG algorithm analyses the gradients and orientations of image patches to identify regions that may contain faces.
- ➔ Sliding window approach: The HOG algorithm applies a sliding window technique, moving a fixed-sized window across the image at different scales and positions.
- ➔ Compute HOG features: For each window position, HOG features are extracted to represent the local gradient and edge orientations within the window.
- ➔ Detection using a classifier: A trained linear Support Vector Machine (SVM) classifier is used to classify each window as containing a face or not based on the extracted HOG features.
- ➔ Non-maximum suppression: To eliminate overlapping and redundant detections, non maximum suppression is typically applied to select the most confident face bounding boxes.
- ➔ Output: The algorithm provides the coordinates of the detected face bounding boxes in the input image.

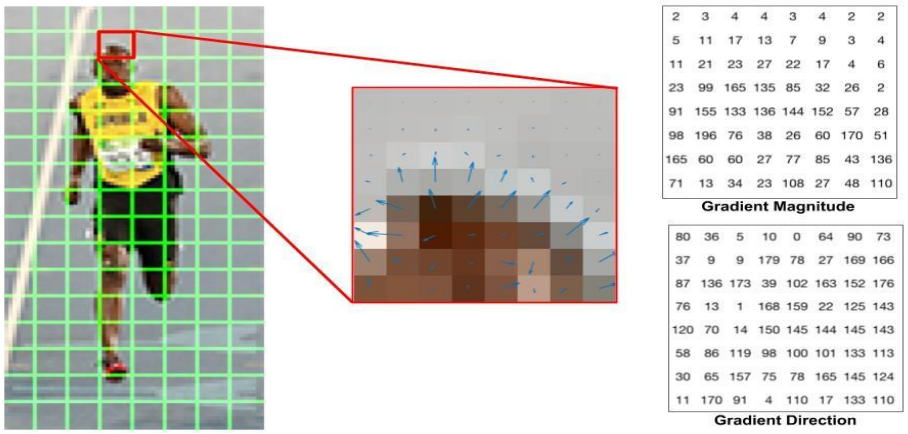


Fig-1: **Histogram of oriented gradients**

3.2.2 Face Encoding using Convolutional Neural Networks (CNN):

After detecting a face using the HOG algorithm, the face_recognition module employs a deep neural network based on the ResNet architecture to compute a face encoding for each detected face. The face encoding is a high-dimensional vector representation that captures unique facial features and characteristics. These encodings can be compared using distance metrics to determine the similarity between different faces for face recognition purposes.

Face Encoding using Convolutional Neural Networks (CNN):

- ➔ Preprocess the detected face region: The detected face region is pre-processed, ensuring it is aligned and suitable for further processing.
- ➔ Apply the CNN-based face encoding model: A pre-trained convolutional neural network(CNN), often based on the ResNet architecture, is utilized.
- ➔ Forward pass through the network: The preprocessed face region is passed through the CNN, resulting in a high-dimensional vector representation known as the face encoding.
- ➔ Compute the face encoding: The output of the CNN represents the unique facial features and characteristics of the detected face.
- ➔ Output: The algorithm provides the face encoding, which can be used for face recognition tasks.

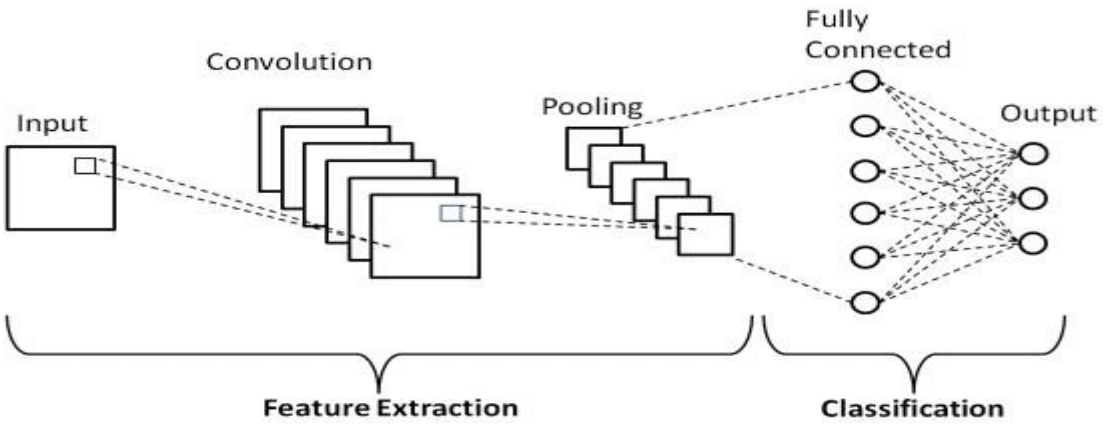


Fig-2: **Convolutional neural networks**

3.2.3 Fernet Algorithm:

In our project we used the fernet algorithm with 256-bit key for password encryption and decryption. Fernet is a symmetric AES encryption algorithm that works in CBC mode, used for secure communication and data storage. It provides confidentiality, integrity, and authenticity making it ideal for encrypting sensitive data. Fernet ensures that the same secret key is used for both encryption and decryption, enabling straightforward implementation and efficient encryption or decryption operations.

Step 1: Key Expansion

➔ The original secret key, which is of 256 bits, is expanded to generate a set of 14 round keys used in each round.

Step 2: Initial Round

➔ The input plaintext is divided into a 4x4 matrix(128-bit blocks) called the State.

➔ The initial round key is added to the State using bitwise XOR operation. This step is known as the "AddRoundKey".

Step 3: Main Rounds

➔ The main rounds consist of four operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.

a. SubBytes: Each byte in the State matrix is replaced by its corresponding value from an S-box, which is a predefined substitution table.

b. ShiftRows: The rows of the State matrix are cyclically shifted to the left. The first row remains unchanged, the second row is shifted one position to the left, the third row is shifted two positions to the left, and the fourth row is shifted three positions to the left.

c. MixColumns: Each column of the State matrix is multiplied using the predefined matrix which provides diffusion and confusion.

d. AddRoundKey: In each main round, a round key obtained from the key expansion step is added to the State using bitwise XOR operation.

➔ Each main round uses a different round key obtained from the key expansion step, and the State matrix is updated accordingly.

Step 4: Final Round

➔ In the final round, the SubBytes, ShiftRows and AddRoundKey operations are performed.

Step 5: Output

➔ The State matrix obtained after the final round represents the ciphertext, which is the encrypted version of the original plaintext.

To decrypt the ciphertext, the reverse operations are performed in the reverse order, using the same round keys in the key expansion step.

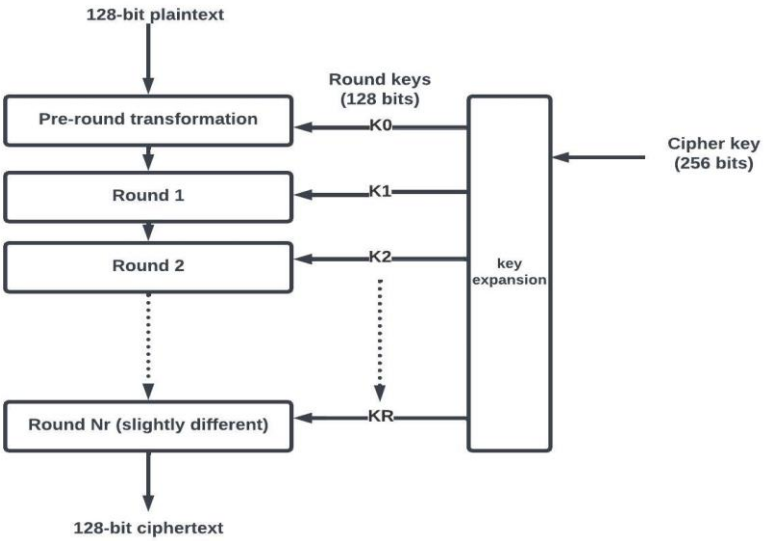


Fig-3: **Advanced Encryption Standard algorithm**

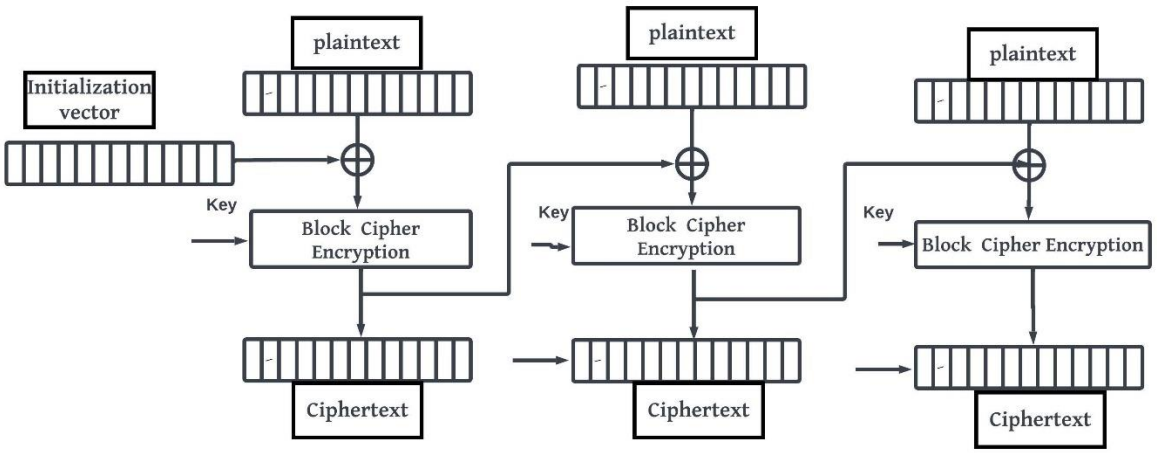
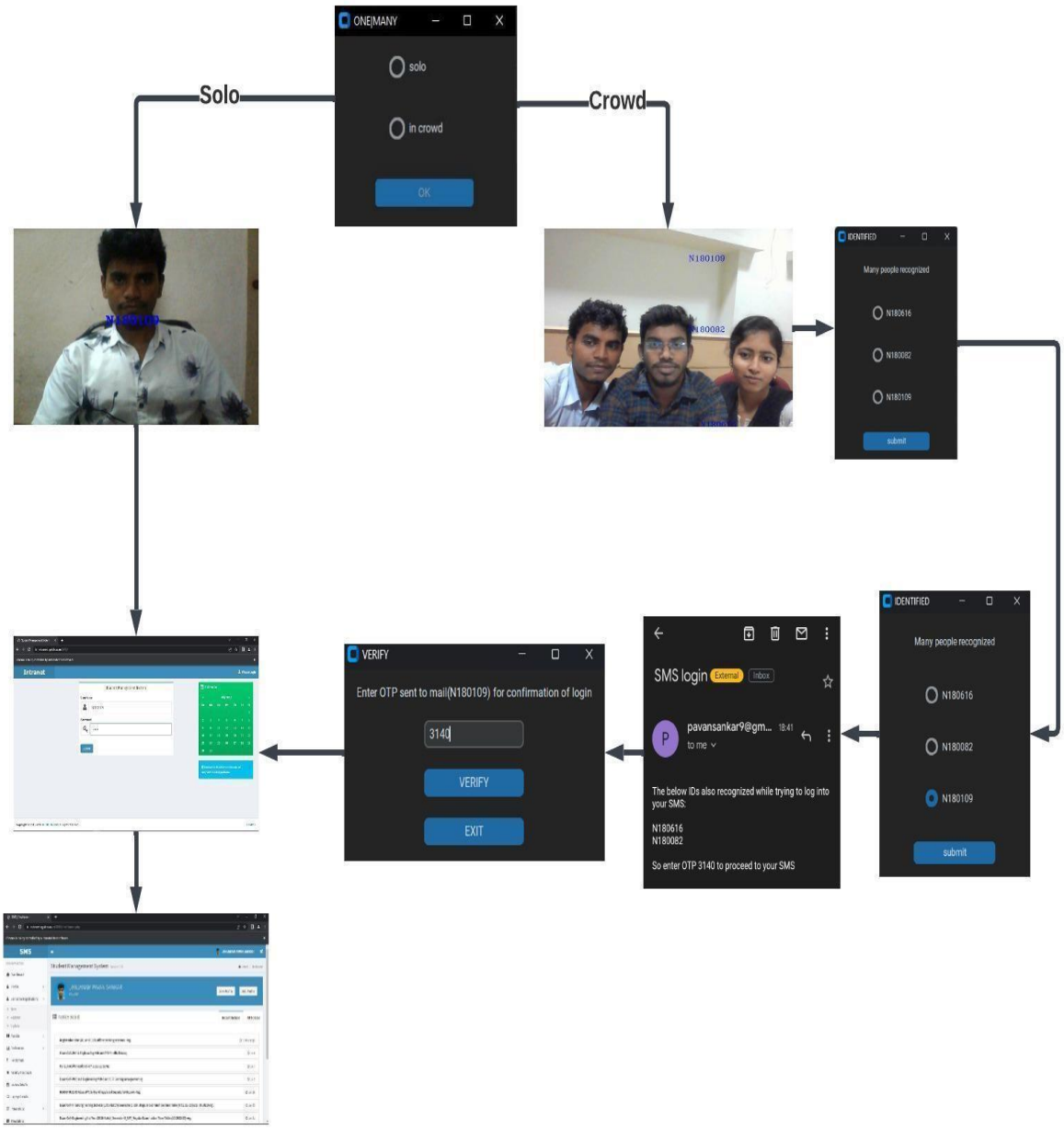


Fig-4: **Cipher block chaining (CBC) mode of encryption**

4. EXPERIMENTAL RESULTS:

The below one is the pictorial representation of flowchart of our project.



5. CONCLUSION AND FUTURE SCOPE

In conclusion, our project has successfully implemented a comprehensive student management system that leverages advanced technologies to enhance security and user experience. We have established a secure login process based on student facial features. The utilization of MongoDB as our database solution enables the storage of encrypted passwords, student details, ensuring privacy and data integrity. The incorporation of web scraping facilitates retrieval of student images and browser automation. The integration of email functionality enables automated notifications and updates. The custom GUI developed using customtkinter offers visually appealing interface for students. Collectively, these components contribute to a robust and user friendly student management system that provides a seamless experience for students.

Future Scope:

- As our future work, we would like to develop a web interface and deploy it in the cloud.
- We further want to make it as an open source contribution.
- We are going to perform spoof detection.

6. REFFERENCES

[1] Kumar, Bishnu Deo, Hilal Ahmad Mir, Mohd Kaif Ahmed, and Mohd Tabish Siddiqui. "Exam form automation using facial recognition." *Materials Today: Proceedings* 80 (2023): 2236-2240.

[2] https://intranet.rguktn.ac.in/SMS/