AI - POWERED HOURLY ATTENDANCE CAPTURING SYSTEM

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,

HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

TALLAPALLY SAGAR	**19UK1A05M1**
BAIRI SANJAY KUMAR	**19UK1A05H2**
PINNINTI PRATHYUSHA	**19UK1A05J5**

Under the esteemed guidance of

Dr.KALYANAPU SRINIVAS

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005
2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL – 506005

2019 – 2023

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-1

This is to certify that the UG Phase-1 Project entitled **“AI-POWERED HOURLY ATTENDENCE CAPTURING SYSTEM ”** is being submitted by **TALLAPALLY SAGAR (19UK1A05M1),BAIRI SANJAYKUMAR(19UK1A05H2),PINNINTI PRATHYUSHA(19UK1A05J5)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University,Hyderabad** during the academic year **2022-23**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Dr. Kalyanapu Srinivas
(Associate Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making me available all the required assistance and for his support and inspiration to carry out this UG Phase-1 Project in the institute.

We extend my heartfelt thanks to **Dr. R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Phase-1 Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the Industry Oriented Mini Project and for their support in completing the UG Phase-1 Project.

We express heartfelt thanks to the guide, **Dr. KALYANAPU SRINIVAS** Associate professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Phase-1 Project.

Finally, We express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

TALLAPALLY SAGAR	**19UK1A05M1**
BAIRI SANJAY KUMAR	**19UK1A05H2**
PINNINTI PRATHYUSHA	**19UK1A05J5**

ABSTRACT

This project focuses on creating an automated system that takes the attendance of students on hourly basis using preinstalled cameras in the classes. We make use of AWS services to marks the attendance, store the attendance in DB.S3 Bucket It is the storage service of AWS, where image dataset will be stored. AWS Recognition It provides the deep learning service, which is used to detect the faces in images stored in S3 API Gateway : It provides the REST API service, which would be used with Lambda function to connect to Dynamo DB, so that attendance is inserted to DB or attendance count can be read from DB. API Gateway creates a special token so that client can send.

TABLE OF CONTENTS:-

1. **INTRODUCTION.... ....1**

 1.1 OVERVIEW…………………………………………………...1

 1.2 PURPOSE…………………………………………………..1

2. **LITERATURE SURVEY..2-4**

 2.1 EXISTING SYSTEM……………………………………..2

 2.2 DISADVANTAGES OF EXISTING SYSTEM……………….3

 2.3 PROPOSED SYSTEM…………………………………………4

3. **THEORETICAL ANALYSIS……………………………...5-8**

 3.1 BLOCK DIAGRAM……………………………………..5

 3.2 HARDWARE/SOFTWARE DESIGN………………………..6

 3.3 PROJECT FLOW………………………………………..7

 3.4 SOFTWARE AND HARDWARE REQUIREMENTS……8

4. **EXPERIMENTAL   INVESTIGATIONS……………………..9-12**

5. **FLOW CHART…………………………………………………13**

6. **CONCLUSION…………………………………………………14**

7. **FUTURE SCOPE………………………………………………15**

LIST OF FIGURES PAGE NO

Fig 1 : Project architecture...**7**

Fig 2: The services provided by Dynamo DB..**9**

Fig 3: AWS S3 bucket use case diagram..**10**

Fig 4: AWS API Gateway interaction with lambda function and end user................**11**

Fig 5: In above diagram Lambda is triggered...**12**

Fig 6: Flow of project...**13**

1. INTRODUCTION

1.1 OVERVIEW:

The Attendance Monitoring System is crucial for tracking student performance in organizations, but manually checking attendance is a difficult task. Usually, attendance is taken by calling out students' names or register numbers and recording their presence in attendance registers provided by department heads. Some organizations also require students to sign these sheets for future reference. However, this method is repetitive, prone to errors, and may lead to proxy attendance. Additionally, it is challenging to monitor attendance in large classrooms. To solve these problems, we propose using face detection and recognition technology to automatically take the student's attendence. The preinstalled cameras capture's photo of the student and compares it to the image stored at the time of enrollment, marking attendance if there is a match. We could also use artificial intelligence to analyze motion pictures of students in class to gather data on how much time they spend there.

1.2 PURPOSE:

The objective of this project is to automate the attendance system of the institutions to reduce problems that occur when attendance is taken manually. By using cameras to monitor classrooms and evaluate students' interest, attendance can be marked accurately and students are encouraged to pay attention. The developed system is highly beneficial as it saves valuable time for students and lecturers, reduces paper usage, and generates reports promptly. Furthermore, it minimizes administrative tasks, eliminates human errors, prevents proxy attendance, resolves time-related disputes, and simplifies the process of updating and maintaining attendance records.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM :

Maintaining attendance is a critical aspect of evaluating students' performance in all institutes, and various methods are used for this purpose. Some institutes take attendance manually by registering attendance for every hour and later upload the data to the server or file-based approach, while others have adopted biometric techniques for automatic attendance. However, these methods are inefficient and time-consuming. Artificial intelligence can provide a solution to this problem by streamlining attendance tracking and reducing the time and effort required..

2.1.1 Fingerprint-based attendance System :

This attendance system utilizes a portable fingerprint device that can be passed to all students during class time to scan their thumb impression and mark attendance without the need for faculty involvement. This method ensures a reliable attendance record. However, One challenge associated with this approach is that it may be disruptive to students' concentration if they are required to pass around the fingerprint device during class. This could ultimately be counterproductive and pose difficulties in managing the classroom without causing interruptions.

2.1.2 Radio Frequency Identification-based Attendance System :

A radio frequency identification (RFID) system is utilized in which each student is required to carry an ID card containing a radio frequency tag. When the ID card is placed on a card reader, the student's attendance is recorded and stored in a database. However, this system is vulnerable to unauthorized access, as individuals may use another person's ID card to enter the organization. This process lacks security, and may result in individuals marking attendance for their friends by scanning the approved ID card on the scanner.

2.1.3 Iris-Recognition Based Attendance System :

The biometric system that utilizes iris recognition technology. This system captures images of the iris and extracts unique features that are then matched with those in the database. However, one major challenge with this system is ensuring that the transmission lines of the scanner are in optimal condition to capture the entire iris passing through the system. The system is highly secure, reliable, and efficient due to its real-time face detection feature. However, there is still a need for further development to ensure its accuracy in different lighting conditions.

2.2 DISADVANTAGES OF EXISTING SYSTEM:

Rate of Error:

When it comes to biometric attendance machines, there are two types of errors: False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR occurs when the device accepts an unauthorized person, while FRR arises when an authorized person is rejected. The error rate of these machines is typically around 1%, which means that for an organization with 20,000 employees, there could be an error attendance rate of approximately 200 employees.

Delay:

The process of using biometric attendance devices to mark attendance is often time-consuming, leading to long queues of workers during morning and evening rush hours.

Infection Carrier:

Due to the highly contagious nature of the coronavirus, it can easily spread through human touch. If an infected employee touches a biometric attendance system and another person touches the same surface, they may be susceptible to contracting the virus. This is because everyone who uses the biometric attendance management system can share germs and potentially spread the virus..

Difficulty in Scanning:

Scanning biometric identifiers such as iris and fingerprints can be challenging due to various factors. For instance, the presence of eyelashes, eyelids, lens, and reflections from the cornea can make it difficult to accurately scan the iris. Similarly, if a finger is injured, a fingerprint scanner may not be able to accurately scan the finger.

Physical Challenges:

Many employees face physical challenges due to lost or damaged body parts, making it difficult to enroll them in the biometric attendance system. As a result, it can be a challenge to get these employees involved in the enrollment process.

Environmental Challenges:

Extreme temperatures, both cold and heat, can also lead to a high error rate in biometric attendance systems. This poses a challenge for the system's use in such conditions.

2.3 PROPOSED SOLUTION:

The proposed solution/application aims to capture hourly attendance without the need for manual intervention. This will be achieved through the development of a smart device integrated with a camera that captures the images of Student every hour in the classroom. These images are then sent to a model that utilizes AWS Rekognition Service to recognize the faces of students and store the images in S3 for storage,then every hour the captured images are compared with registered images.Attendance will be updated automatically into the database, and a web-based dashboard will be built to visualize the student attendance information. This innovative solution eliminates the need for physical attendance devices and reduces the potential for errors in the attendance-taking process.

3. THEORITICAL ANALYSIS

Milestone 1: Data Collection

Machine learning is a field that relies heavily on data to enable machines to learn and improve their performance. The availability of a high-quality training dataset is essential for training machine learning algorithms to perform various tasks effectively. Without sufficient training data, the algorithm will not be able to learn how to perform the desired tasks.The training dataset serves as the foundation on which the model will be built, and the quality of the model will depend largely on the quality of the training dataset. The training dataset can be collected from various sources, including the web, public datasets, or by creating your own dataset..

Milestone 2: Configuring the AWS Cloud

2.1 Create an AWS Free Tier Account:

To work on this project, we need a AWS cloud account which provides services which is required by our project.

2.2 Create an AWS IAM user:

AWS Identity and Access Management (IAM) is a crucial web service that ensures secure and controlled access to AWS resources. IAM enables centralized management of permissions that determine which AWS resources can be accessed by users. IAM provides an authentication mechanism that verifies the identity of users and authorizes their access to resources based on their permissions. Through IAM, administrators can create and manage user accounts, groups, roles, and policies that help to protect the confidentiality, integrity, and availability of AWS resources..

Milestone 3: Explore the AWS S3 Service

Amazon Simple Storage Service (Amazon S3) which is mainly used for storage purpose we can store data like images,audio files,documents etc. Amazon S3 is a fully managed service that allows you to store and manage large amounts of data as objects in a highly secure and scalable manner.

Milestone 4: Explore an AWS Rekognition Service.

4.1 Create Collection Ids:

Creating a face collection is the first step to store facial information, for every registered image we have an collection by that collection id we can check the similarity of the images.

4.2 Create face indexes:

The IndexFaces operation in AWS Rekognition service allows filtering of the faces that are indexed from an image. It offers the flexibility to specify the large number of faces to index only the faces detected with high quality. This functionality enables accurate and efficient storage information of facial feature for all faces in a collection. This stored facial information can be used to detect the collection for face matches, providing reliable and secure facial recognition capabilities..

Milestone 5: Explore an AWS Dynamo DB Service

Amazon DynamoDB is a highly efficient NoSQL database service that is fully managed and designed to offer fast and predictable performance while also being easily scalable. With DynamoDB, you can create database tables to store and retrieve any amount of data, and efficiently serve any level of request traffic. DynamoDB also provides the ability to scale up or scale down the throughput capacity of your tables without any performance degradation or downtime. This makes it an ideal choice for applications that require seamless scalability and high performance.

Milestone 6: Building a Model

6.1 Write a Python Code for Video Streaming.

6.2 Compare Captured Image with Stored Image.

6.3 Insert the Attendance into DynamoDB through API Gateway.

MilesStone 7: Building an Application

7.1 Create a Lamda function to get data from DB.

7.2 Create an HTML Application.(Flask).

7.3 Display the student's attendance on Web Application

3.1 BLOCK DIAGRAM:

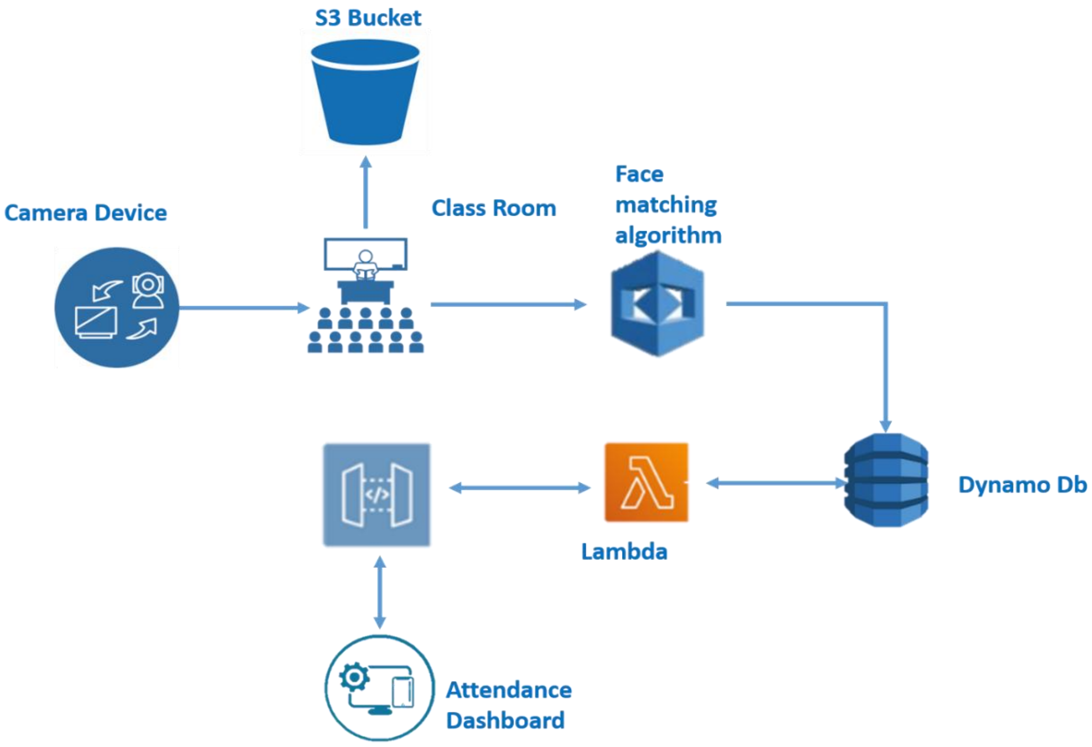


Fig 1 : project architecture.

3.2 Steps invovled in project:

Step1. Storing the Image of students in S3.

Step2. Capturing the image of the students on an hourly basis

Step3. By Using the lambda funtionwe compare the captured images with registered images by using Face comparison algorithm in AWS.

Step4. Marking the Student's attendance and storing the data into DynamoDB.

Step5. Creating a Lambda function and rest API using API gateway to connect to dynamoDB through a web application.

Step6. Creating a web-based dashboard to visualize the attendance using HTML.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements:

1. Anaconda Environment
2. Flask
3. Python 3.9
4. And other python libraries like NumPy,pandas.

Hardware Requirements:

1. Operating system
2. Processing
3. RAM
4. Operating system specifications
5. Disk space

4. EXPERIMENTAL INVESTIGATION

Python Web Frameworks:

It is web Framework that provides the developers tools and required stuff for building an application withoud satisfying the lower level components. The core components of a web application may include an HTTP application server, a database, a template engine, a request dispatcher, an authentication module, and an AJAX toolkit. These components can be separate or provided together as part of a high-level framework. High-level frameworks such as Django, TurboGears, and web2py are among the most popular options for web development..

AWS DynamoDB:

In AWS DynamoDB the developers are not required to write queries to perform operations. It is faster compare to other DataBase it is used in serverless applications.It can handle millions of data,it can read and write queries per second. Compared to traditional relational databases, DynamoDB is specifically designed to handle high-performance, internet-scale applications. It is fully managed and can operate across multiple regions and master nodes, ensuring durability and security while delivering single-digit millisecond performance. DynamoDB also offers features such as backup and restore, as well as in-memory caching for optimized performance of internet-scale applications.

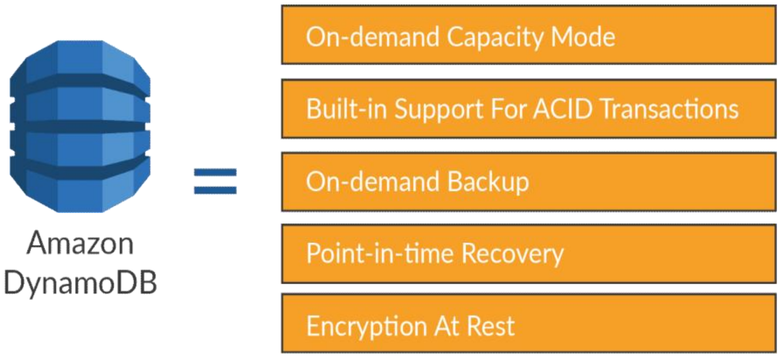


Fig 2: The services provided by Dynomo DB.

AWS Rekognition:

Amazon Rekognition is a powerful tool that provides effortless image and video analysing tools for our application through cutting-edge and deep learning technology. Unlike other ML applications, Amazon Rekognition requires no machine learning experience, making it an accessible solution for developers. With Amazon Rekognition,we can identify people,text,images,celebrites photo with this technology in an inappropriate content. Additionally,it has accurate analysing tools and seaching capabilities, which are used to verify users, count people, and improve public safety through face comparisons.

Amazon S3:

Amazon S3 stands for " Amazon Simple Storage Service " which is mainly used for storage purpose we can store data like images,audio files,documents etc. Amazon S3 is a fully managed service that allows you to store and manage large amounts of data as objects in a highly secure and scalable manner. Amazon S3 provides a range of storage solutions for various purposes, such as data storage for websites, mobile apps, backup and restore, archiving, enterprise applications, IoT devices, and big data analytics. Additionally, it offers user-friendly management features that enable efficient organization of data and customization of access controls to meet specific business, organizational, and compliance needs.

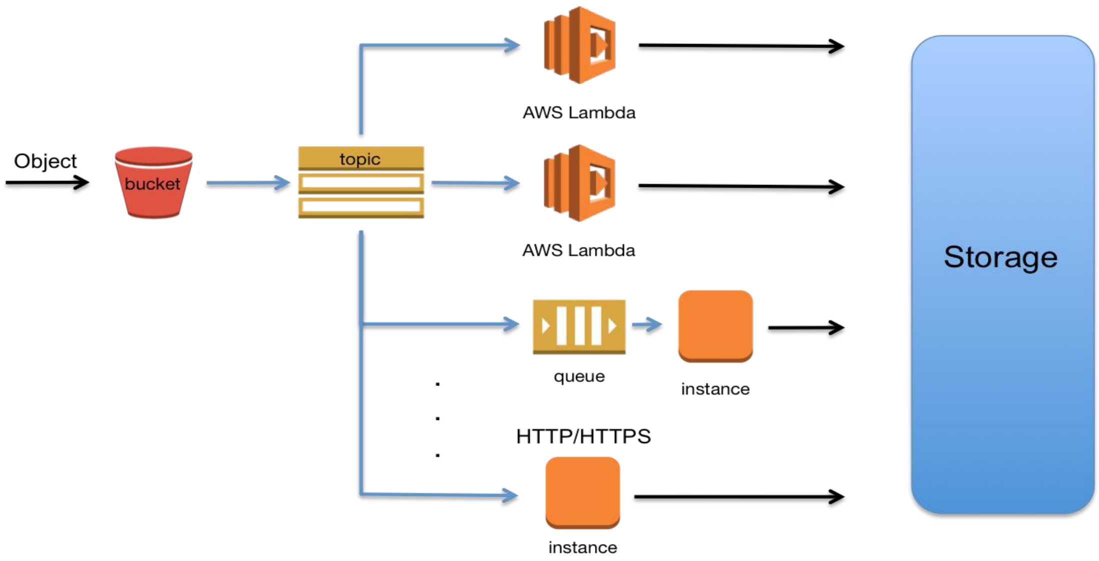


Fig 3: AWS S3 bucket usecase diagram.

AWS API Gateway:

Amazon API Gateway service is used for creating API's for Application. With APIs acting as the gateway to access back-end services, API Gateway provides an easy and secure way to access the data and Funtionality. RESTful and WebSocket APIs can be created to provide real-time two-way communication applications. API Gateway offers support for serverless and containerized workloads and also supports web applications. With features such as traffic management,authorization and access control, monitoring, and API version management, with other features, it handles all tasks related to accepting and processing huge API calls. The best part is, there are no startup costs. Based upon the API calls and service we used we need to pay. The tiered pricing model of API Gateway allows you to reduce costs as your API usage scales.

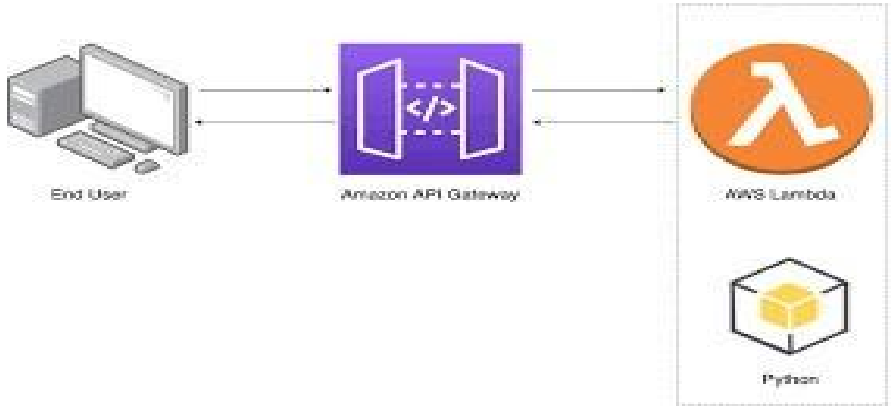


Fig 4: AWS API Gateway interaction with lambda funtion and end user.

AWS Lambda:

It is an Computing service that doesn’t required any servers for running codes. It means we only write codes remaining stuff will be handled by lambda Funtion. You can use Lambda funtion to run code for any kind of application , without the need for administration. Once you upload your code, Lambda automatically takes care of everything required to run and scale it with high availability.

One of the great things about Lambda is that after setting the code we can call it from web application or Monile apps. This allows you to create event-driven applications and micro services without having to worry about the underlying infrastructure.

With Lambda, you only pay for the compute time that you consume, and there is no charge when your code is not running. This makes it an extremely cost-effective solution for running your code. Lambda supports different programming languages, including Node.js, Python, Java, Go, and C#. You can also use the AWS Lambda Layers feature to manage your in-development function code and resource dependencies separately.

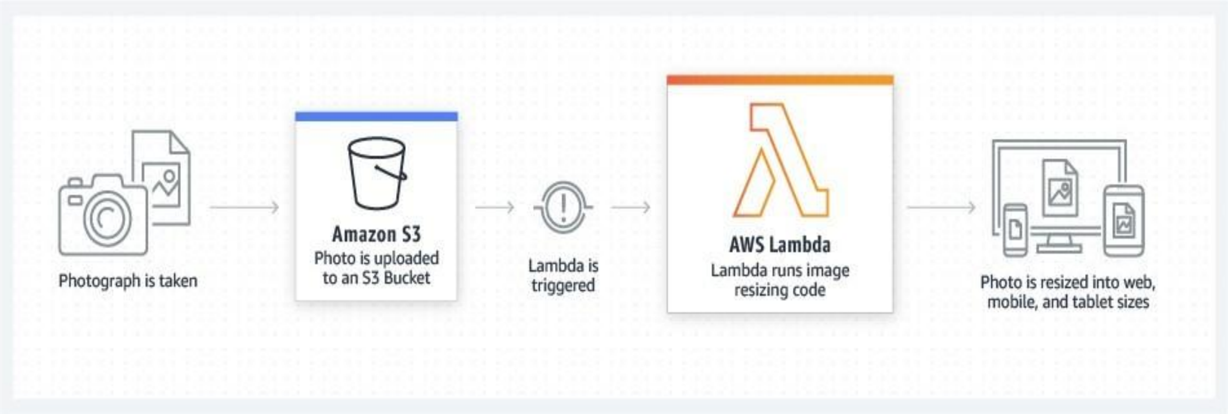

Photograph is taken
Amazon S3
Photo is uploaded to an S3 Bucket
Lambda is triggered
AWS Lambda
Lambda runs image resizing code
Photo is resized into web, mobile, and tablet sizes

Fig 5: In above diagram Lambda is triggered.

AWS Lambda is a Computing service which helps in running the code without managing the physical servers. By simply uploading your code, Lambda handles all the necessary resources to run and scale it with high availability, making it an ideal solution for applications that require quick and frequent updates, such as image or object uploads to Amazon S3 or updates to DynamoDB tables. Lambda can also be triggered by other AWS services, such as website clicks or IoT connected device readings, and can be called directly from any device such as web Applications or Mobile application.

5. FLOWCHART

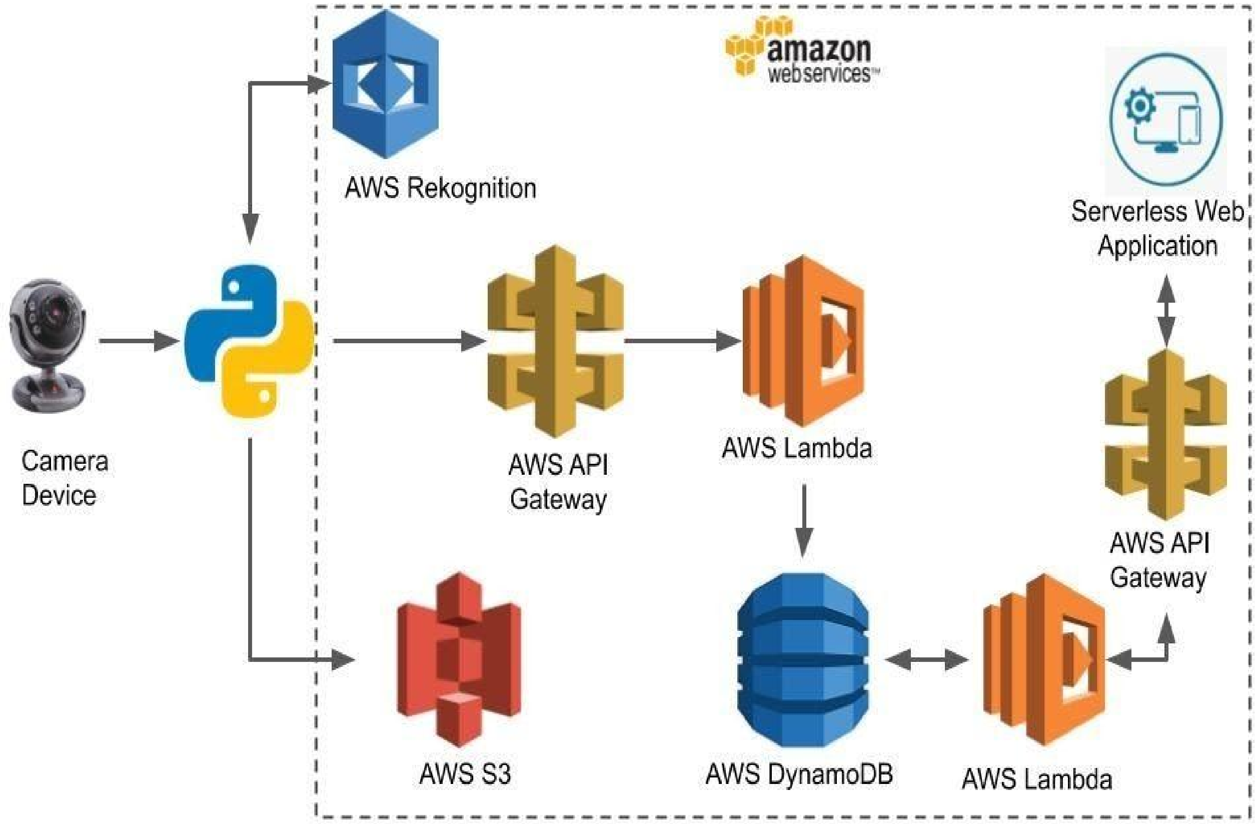


Fig 6: Flow of project.

The above flowchart shows about the project that we are working .Initially the images are registered in S3 bucket whenever the image captured by the preinstalled cameras then the lambda function get triggered and the captured images are analyzed using the face print (recognition ID) which are present in dynamo DB.If the face print of the images are matched then the attendence is Marked.For every hour the attendence is marked .

6.CONCLUSION

Our team is implementing an attendance system that automates the process to reduce errors that result from manually taking attendance. By installing cameras in classrooms, the system can monitor student interest and mark attendance automatically, potentially increasing student attendance. An Artificial Intelligence-based solution can help monitor and mark attendance accurately, encouraging students to attend school or college regularly. The developed system is efficient and saves valuable time for students and lecturers, while also reducing paper usage and providing timely reports. The system eliminates most administrative tasks and minimizes human errors, avoiding issues such as proxy punching and time-related disputes.

7. FUTURE SCOPE

Facial recognition technology has a wide range of potential applications that can bring significant benefits. Some of these applications include:

1. Improving ATM security and preventing fraud in India: By creating a database of all ATM card users in the country and installing facial recognition systems at ATMs, the technology can be used to match the user's photo with the stored photo in the database for access permission.

2. Identifying and reporting duplicate voters in India during elections.

3. Streamlining passport and visa verification for efficient processing.

4. Enabling faster and more accurate driving license verification.

5. Enhancing surveillance and security at important locations such as airports and government buildings.

6. Verifying the identity of candidates during exams such as Civil Services, SSC, IIT, and MBBS.

7. Deploying the technology for verification and attendance tracking at government offices and corporate.

8. Improving access control verification and identification of authorized users at bank lockers and vaults.

9. Identifying criminals and enhancing law enforcement efforts by the police force.

These applications can help save time, reduce human errors, increase security, and improve the overall efficiency of various systems..

AI - POWERED HOURLY ATTENDANCE CAPTURING SYSTEM

A UG PROJECT PHASE-2 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

TALLAPALLY SAGAR	19UK1A05M1
BAIRI SANJAY KUMAR	19UK1A05H2
PINNINTI PRATHYUSHA	19UK1A05J5

Under the esteemed guidance of

Dr.KALYANAPU SRINIVAS

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL – 506005

2019 – 2023

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-2

This is to certify that the UG Phase-2 Project entitled **“AI-POWERED HOURLY ATTENDENCE CAPTURING SYSTEM ”** is being submitted by **TALLAPALLY SAGAR (19UK1A05M1),BAIRI SANJAY KUMAR(19UK1A05H2),PINNINTI PRATHYUSHA(19UK1A05J5)** in partial fulfillment of the requirements for the award of the degree of**Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University,Hyderabad** during the academic year **2022-23**, is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
Dr. Kalyanapu Srinivas	**Dr. R. Naveen Kumar**
(Associate Professor)	(Professor)

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making me available all the required assistance and for his support and inspiration to carry out this UG Phase-2 Project in the institute.

We extend my heartfelt thanks to **Dr. R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Phase-2 Project.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the Industry Oriented Mini Project and for their support in completing the UG Phase-2 Project.

We express heartfelt thanks to the guide, **Dr. KALYANAPU SRINIVAS** Associate professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Phase-2 Project.

Finally, We express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

TALLAPALLY SAGAR	**19UK1A05M1**
BAIRI SANJAY KUMAR	**19UK1A05H2**
PINNINTI PRATHYUSHA	**19UK1A05J5**

ABSTRACT

This project focuses on creating an automated system that takes the attendance of students on hourly basis using preinstalled cameras in the classes. We make use of AWS services to marks the attendance, store the attendance in DB.S3 Bucket It is the storage service of AWS, where image dataset will be stored. AWS Recognition It provides the deep learning service, which is used to detect the faces in images stored in S3 API Gateway : It provides the REST API service, which would be used with Lambda function to connect to Dynamo DB, so that attendance is inserted to DB or attendance count can be read from DB. API Gateway creates a special token so that client can send.

TABLE OF CONTENTS:-

1. INTRODUCTION……..………………………………………………….22

1.1 OVERVIEW………………………………………………………...22

1.2 PURPOSE…………………………………………………………...22

2. LITERATURE SURVEY…………………………………………….…..23

2.1 EXIXTING SYSTEM……………………………………………….23

2.2 DISADVANTAGES OF EXISTING SYSTEM…………………………….24

2.3 PROPOSED SYSTEM……………………………………………………25

3. CODE SNIPPETS……………………………………………………….26-34

3.1 MODEL CODE…………………………………………………………26-31

3.2 HTML AND PYTHON CODE……………………………………………32-34

4. RESULT………………………………………………….………..….35

5. ADVANTAGES AND DISADVANTAGES……………………….……..36

6. APPLICATIONS……………………………………………………………37

7. CONCLUTION……………………………………………….………..38

8. FUTURESCOPE……………………………………….……….……...39

LIST OF FIGURES PAGE NO

Fig 1 : Importing boto3 and csv library………………………………………………….………**26**

Fig 2: Defining a function for creating collection ID……………………………………**26**

Fig 3: Defining main funtion for collection ID…………………………………………..…**26**

Fig 4: Adding faces to the S3 bucket……………………………………………………**27**

Fig 5: Getting the outputs of the facecount,face id and external id……………………...**28**

Fig6: Excuting the Flask code to get Output……………………………………….………**34**

Fig7: Final output of the model……………………………………………………….……**35**

1. INTRODUCTION

1.3 OVERVIEW:

The Attendance Monitoring System is crucial for tracking student performance in organizations, but manually checking attendance is a difficult task. Usually, attendance is taken by calling out students' names or register numbers and recording their presence in attendance registers provided by department heads. Some organizations also require students to sign these sheets for future reference. However, this method is repetitive, prone to errors, and may lead to proxy attendance. Additionally, it is challenging to monitor attendance in large classrooms. To solve these problems, we propose using face detection and recognition technology to automatically take the student’s attendence. The preinstalled cameras capture’s photo of the student and compares it to the image stored at the time of enrollment, marking attendance if there is a match. We could also use artificial intelligence to analyze motion pictures of students in class to gather data on how much time they spend there.

1.4 PURPOSE:

The objective of this project is to automate the attendance system of the institutions to reduce problems that occur when attendance is taken manually. By using cameras to monitor classrooms and evaluate students' interest, attendance can be marked accurately and students are encouraged to pay attention. The developed system is highly beneficial as it saves valuable time for students and lecturers, reduces paper usage, and generates reports promptly. Furthermore, it minimizes administrative tasks, eliminates human errors, prevents proxy attendance, resolves time-related disputes, and simplifies the process of updating and maintaining attendance records.

2. LITERATURE SURVEY

2.4 EXISTING SYSTEM :

Maintaining attendance is a critical aspect of evaluating students' performance in all institutes, and various methods are used for this purpose. Some institutes take attendance manually by registering attendance for every hour and later upload the data to the server or file-based approach, while others have adopted biometric techniques for automatic attendance. However, these methods are inefficient and time-consuming. Artificial intelligence can provide a solution to this problem by streamlining attendance tracking and reducing the time and effort required..

2.4.1 Fingerprint-based attendance System :

This attendance system utilizes a portable fingerprint device that can be passed to all students during class time to scan their thumb impression and mark attendance without the need for faculty involvement. This method ensures a reliable attendance record. However, One challenge associated with this approach is that it may be disruptive to students' concentration if they are required to pass around the fingerprint device during class. This could ultimately be counterproductive and pose difficulties in managing the classroom without causing interruptions.

2.4.2 Radio Frequency Identification-based Attendance System :

A radio frequency identification (RFID) system is utilized in which each student is required to carry an ID card containing a radio frequency tag. When the ID card is placed on a card reader, the student's attendance is recorded and stored in a database. However, this system is vulnerable to unauthorized access, as individuals may use another person's ID card to enter the organization. This process lacks security, and may result in individuals marking attendance for their friends by scanning the approved ID card on the scanner.

2.4.3 Iris-Recognition Based Attendance System :

The biometric system that utilizes iris recognition technology. This system captures images of the iris and extracts unique features that are then matched with those in the database. However, one major challenge with this system is ensuring that the transmission lines of the scanner are in optimal condition to capture the entire iris passing through the system. The system is highly secure, reliable, and efficient due to its real-time face detection feature. However, there is still a need for further development to ensure its accuracy in different lighting conditions.

2.2 DISADVANTAGES OF EXISTING SYSTEM:

Rate of Error:

When it comes to biometric attendance machines, there are two types of errors: False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR occurs when the device accepts an unauthorized person, while FRR arises when an authorized person is rejected. The error rate of these machines is typically around 1%, which means that for an organization with 20,000 employees, there could be an error attendance rate of approximately 200 employees.

Delay:

The process of using biometric attendance devices to mark attendance is often time-consuming, leading to long queues of workers during morning and evening rush hours.

Infection Carrier:

Due to the highly contagious nature of the coronavirus, it can easily spread through human touch. If an infected employee touches a biometric attendance system and another person touches the same surface, they may be susceptible to contracting the virus. This is because everyone who uses the biometric attendance management system can share germs and potentially spread the virus..

Difficulty in Scanning:

Scanning biometric identifiers such as iris and fingerprints can be challenging due to various factors. For instance, the presence of eyelashes, eyelids, lens, and reflections from the cornea can make it difficult to accurately scan the iris. Similarly, if a finger is injured, a fingerprint scanner may not be able to accurately scan the finger.

Physical Challenges:

Many employees face physical challenges due to lost or damaged body parts, making it difficult to enroll them in the biometric attendance system. As a result, it can be a challenge to get these employees involved in the enrollment process.

Environmental Challenges:

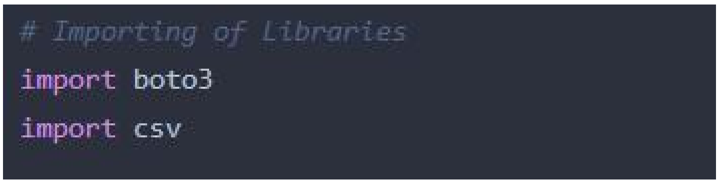
Extreme temperatures, both cold and heat, can also lead to a high error rate in biometric attendance systems. This poses a challenge for the system's use in such conditions.

2.3 PROPOSED SOLUTION:

The proposed solution/application aims to capture hourly attendance without the need for manual intervention. This will be achieved through the development of a smart device integrated with a camera that captures the images of Student every hour in the classroom. These images are then sent to a model that utilizes AWS Rekognition Service to recognize the faces of students and store the images in S3 for storage,then every hour the captured images are compared with registered images.Attendance will be updated automatically into the database, and a web-based dashboard will be built to visualize the student attendance information. This innovative solution eliminates the need for physical attendance devices and reduces the potential for errors in the attendance-taking process.

3.CODE SNIPPETS

3.1 MODEL CODE:

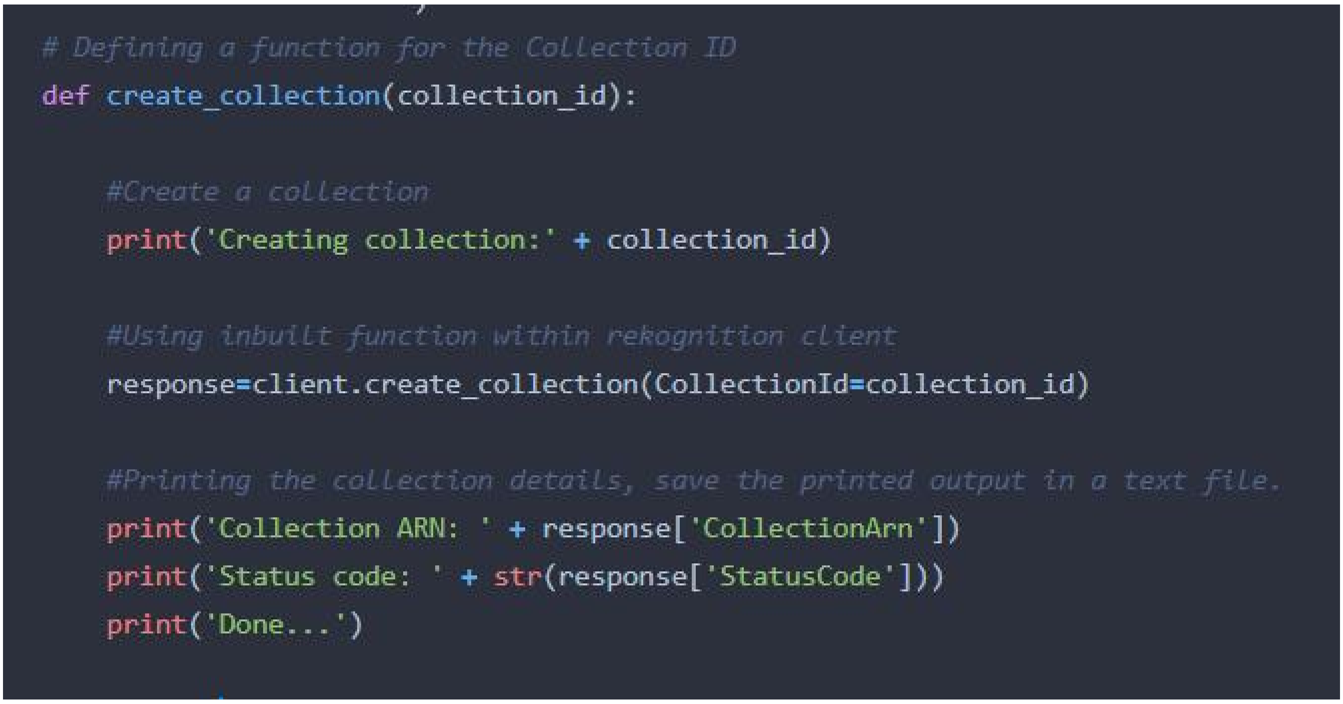


```
# Importing of Libraries
import boto3
import csv
```

Fig1: importing boto3 and csv library .

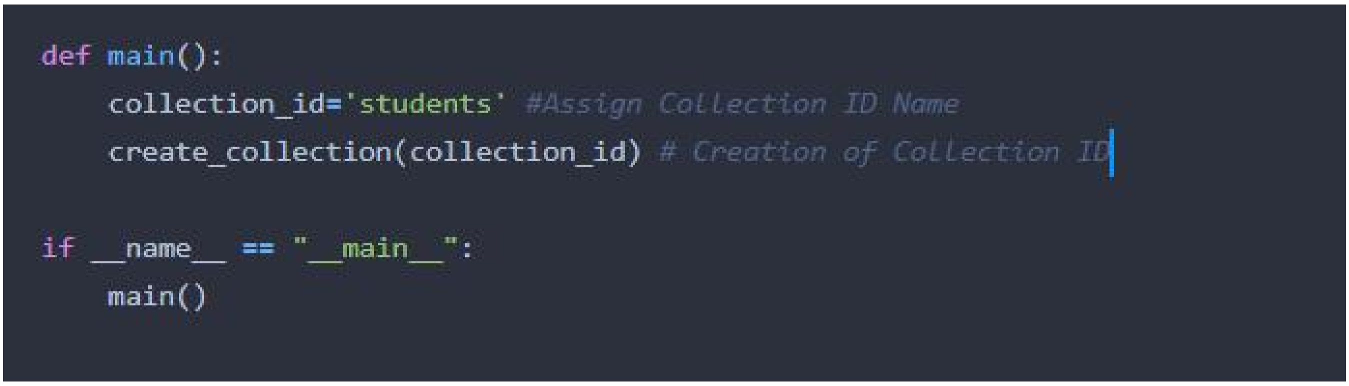
With the help of boto3 library we can communicate with the AWS Console directly with python code.

Here, we are creating a collection ID to store the images,retrive and manipulate the data.



```
# Defining a function for the Collection ID
def create_collection(collection_id):

    #Create a collection
    print('Creating collection:' + collection_id)

    #Using inbuilt function within rekognition client
    response=client.create_collection(CollectionId=collection_id)

    #Printing the collection details, save the printed output in a text file.
    print('Collection ARN: ' + response['CollectionArn'])
    print('Status code: ' + str(response['StatusCode']))
    print('Done...')
```

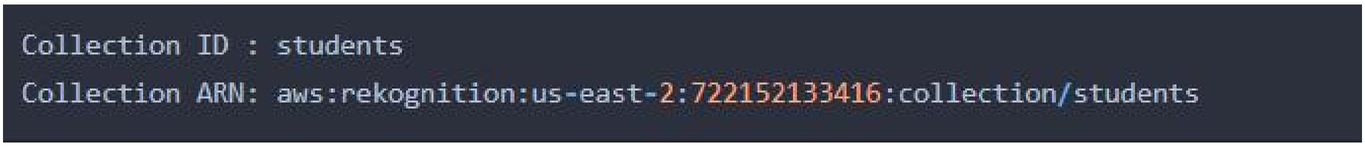
Fig 2: Defining a function for creating collection ID.



```
def main():
    collection_id='students' #Assign Collection ID Name
    create_collection(collection_id) # Creation of Collection ID

if __name__ == "__main__":
    main()
```

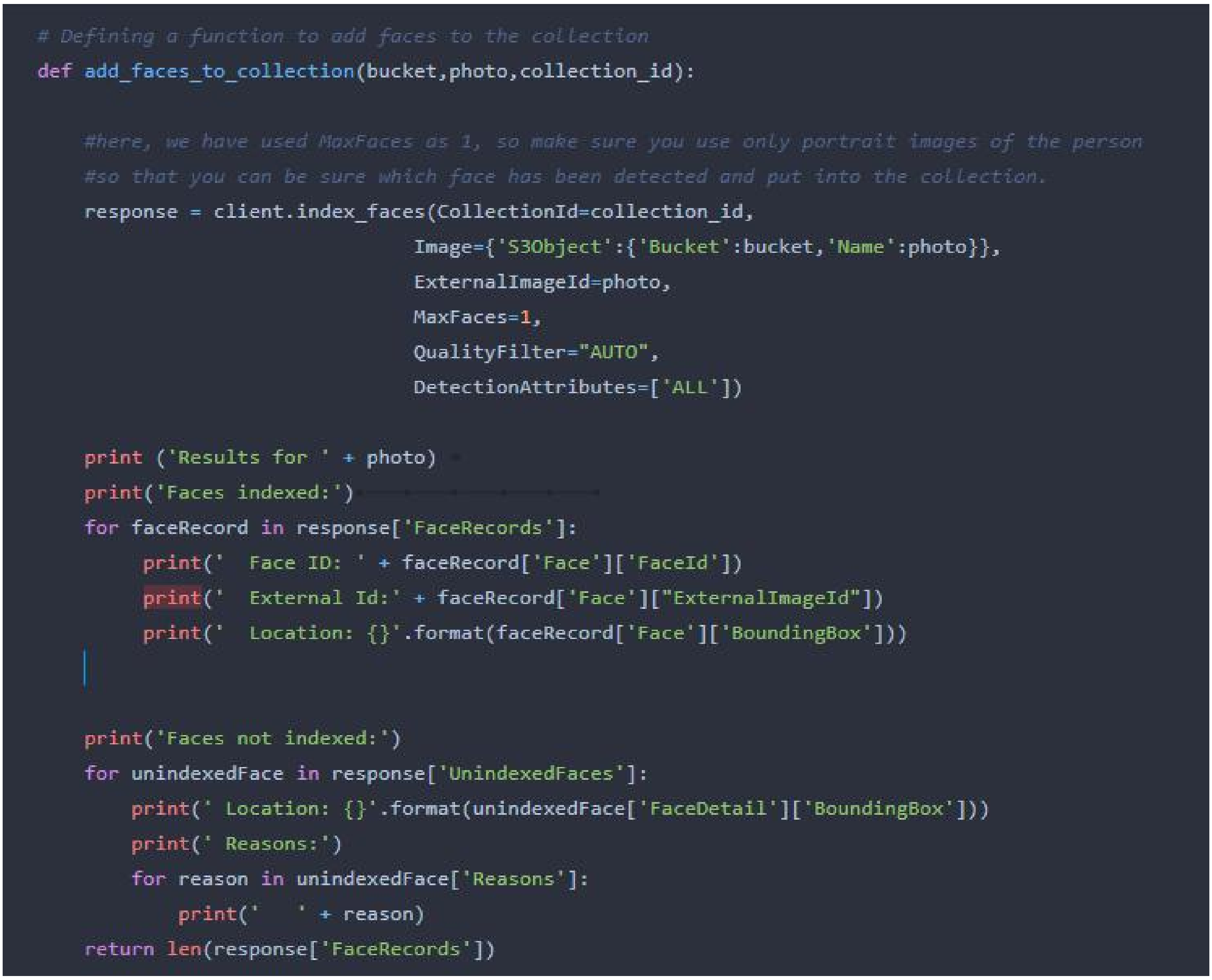
Fig3: Defining main funtion for collection ID.

We named the collection ID as students by running the entire code the collection ID is generated.



```
Collection ID : students
Collection ARN: aws:rekognition:us-east-2:722152133416:collection/students
```

Defining a function to add the faces to the collection :

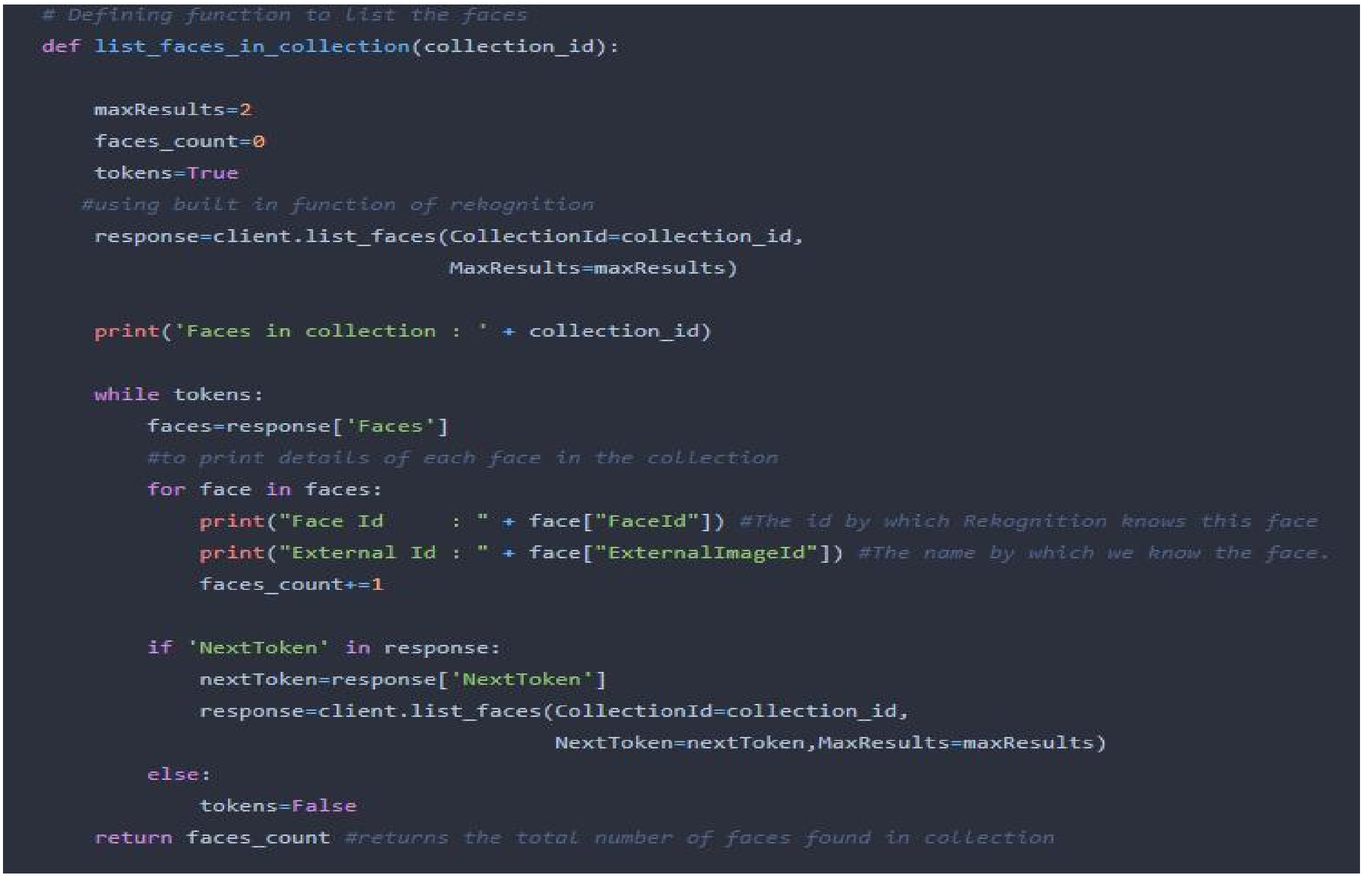


```python
# Defining a function to add faces to the collection
def add_faces_to_collection(bucket,photo,collection_id):

    #here, we have used MaxFaces as 1, so make sure you use only portrait images of the person
    #so that you can be sure which face has been detected and put into the collection.
    response = client.index_faces(CollectionId=collection_id,
                                Image={'S3Object':{'Bucket':bucket,'Name':photo}},
                                ExternalImageId=photo,
                                MaxFaces=1,
                                QualityFilter="AUTO",
                                DetectionAttributes=['ALL'])

    print ('Results for ' + photo)
    print('Faces indexed:')
    for faceRecord in response['FaceRecords']:
         print('  Face ID: ' + faceRecord['Face']['FaceId'])
         print('  External Id:' + faceRecord['Face']["ExternalImageId"])
         print('  Location: {}'.format(faceRecord['Face']['BoundingBox']))


    print('Faces not indexed:')
    for unindexedFace in response['UnindexedFaces']:
        print(' Location: {}'.format(unindexedFace['FaceDetail']['BoundingBox']))
        print(' Reasons:')
        for reason in unindexedFace['Reasons']:
            print('   ' + reason)
    return len(response['FaceRecords'])
```

Fig4: Adding faces to the S3 bucket.

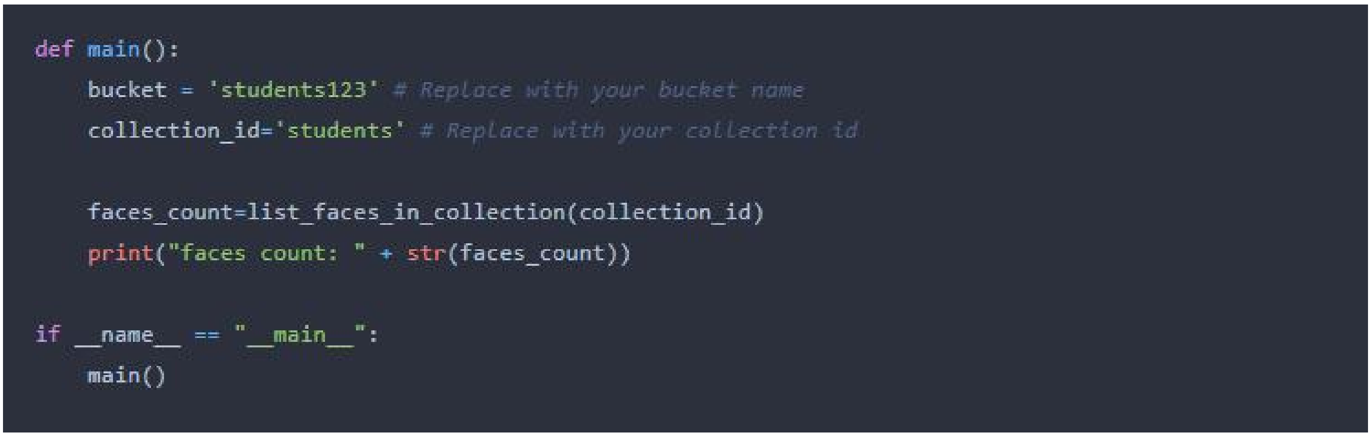
Defining a function to list the faces:

The below function will list all the faces stored in the server container in the form of collection id .



```
# Defining function to list the faces
def list_faces_in_collection(collection_id):

    maxResults=2
    faces_count=0
    tokens=True
   #using built in function of rekognition
    response=client.list_faces(CollectionId=collection_id,
                               MaxResults=maxResults)

    print('Faces in collection : ' + collection_id)

    while tokens:
        faces=response['Faces']
        #to print details of each face in the collection
        for face in faces:
            print("Face Id     : " + face["FaceId"]) #The id by which Rekognition knows this face
            print("External Id : " + face["ExternalImageId"]) #The name by which we know the face.
            faces_count+=1

        if 'NextToken' in response:
            nextToken=response['NextToken']
            response=client.list_faces(CollectionId=collection_id,
                                       NextToken=nextToken,MaxResults=maxResults)
        else:
            tokens=False
    return faces_count #returns the total number of faces found in collection
```

Defining the main function to list the faces:

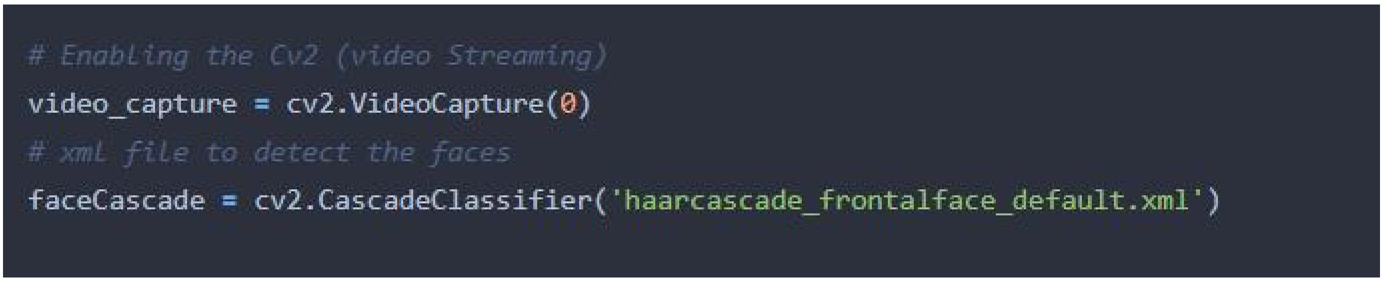


```
def main():
    bucket = 'students123' # Replace with your bucket name
    collection_id='students' # Replace with your collection id

    faces_count=list_faces_in_collection(collection_id)
    print("faces count: " + str(faces_count))

if __name__ == "__main__":
    main()
```

Fig5: Getting the outputs of the facecount,face id and external id.

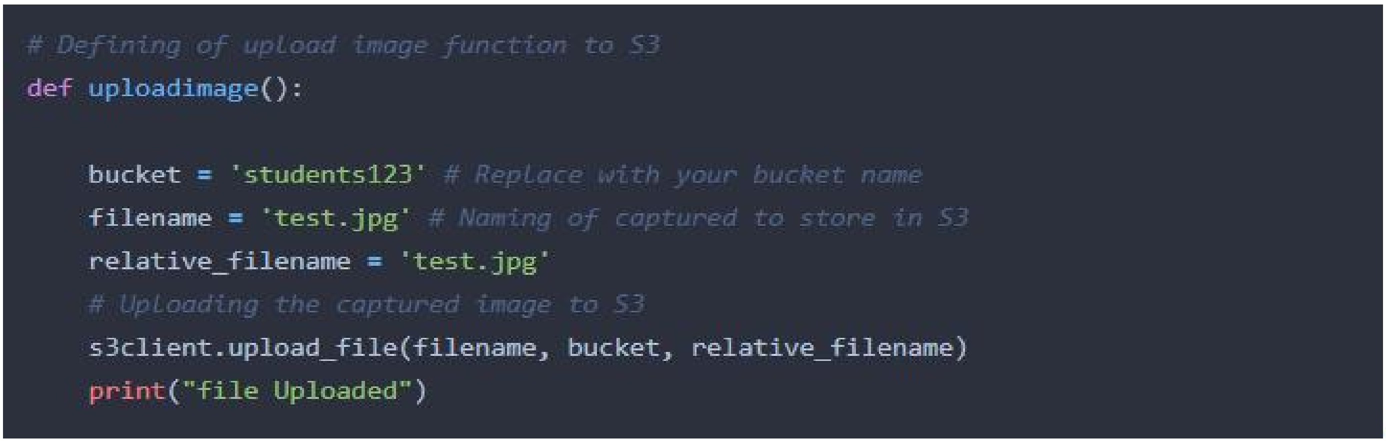
Code for video streaming.

Enable the video streaming with cv2 & initialize the face detection file.



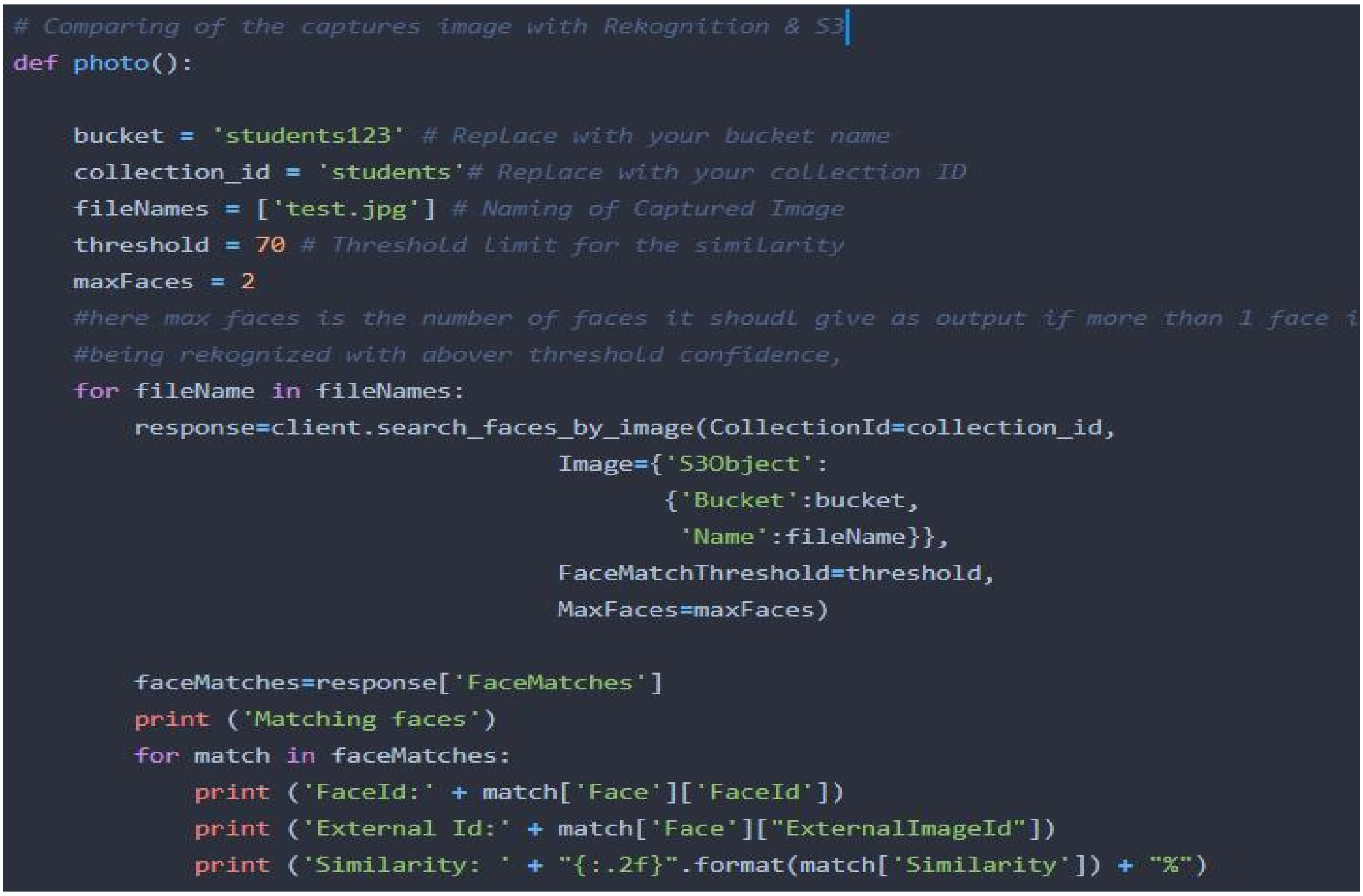
```
# Enabling the Cv2 (video Streaming)
video_capture = cv2.VideoCapture(0)
# xml file to detect the faces
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

Defining of upload image from video stream function to S3 service.



```
# Defining of upload image function to S3
def uploadimage():

    bucket = 'students123' # Replace with your bucket name
    filename = 'test.jpg' # Naming of captured to store in S3
    relative_filename = 'test.jpg'
    # Uploading the captured image to S3
    s3client.upload_file(filename, bucket, relative_filename)
    print("file Uploaded")
```

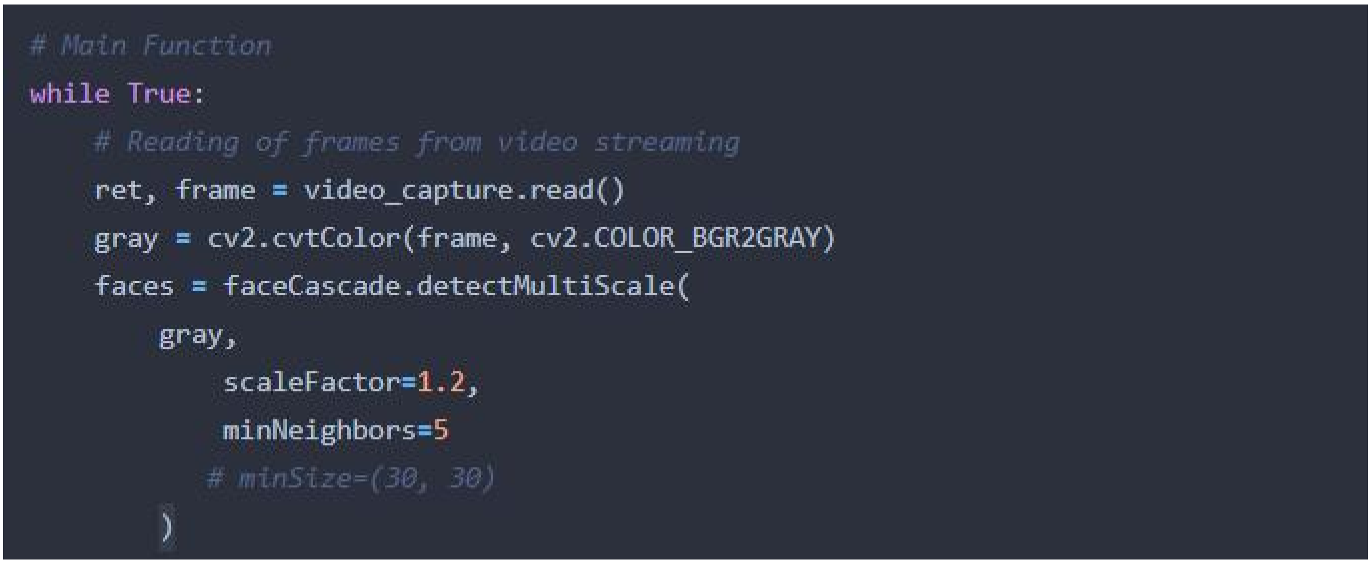
Now we are writing a function to compare the captured images with rekognition & S3



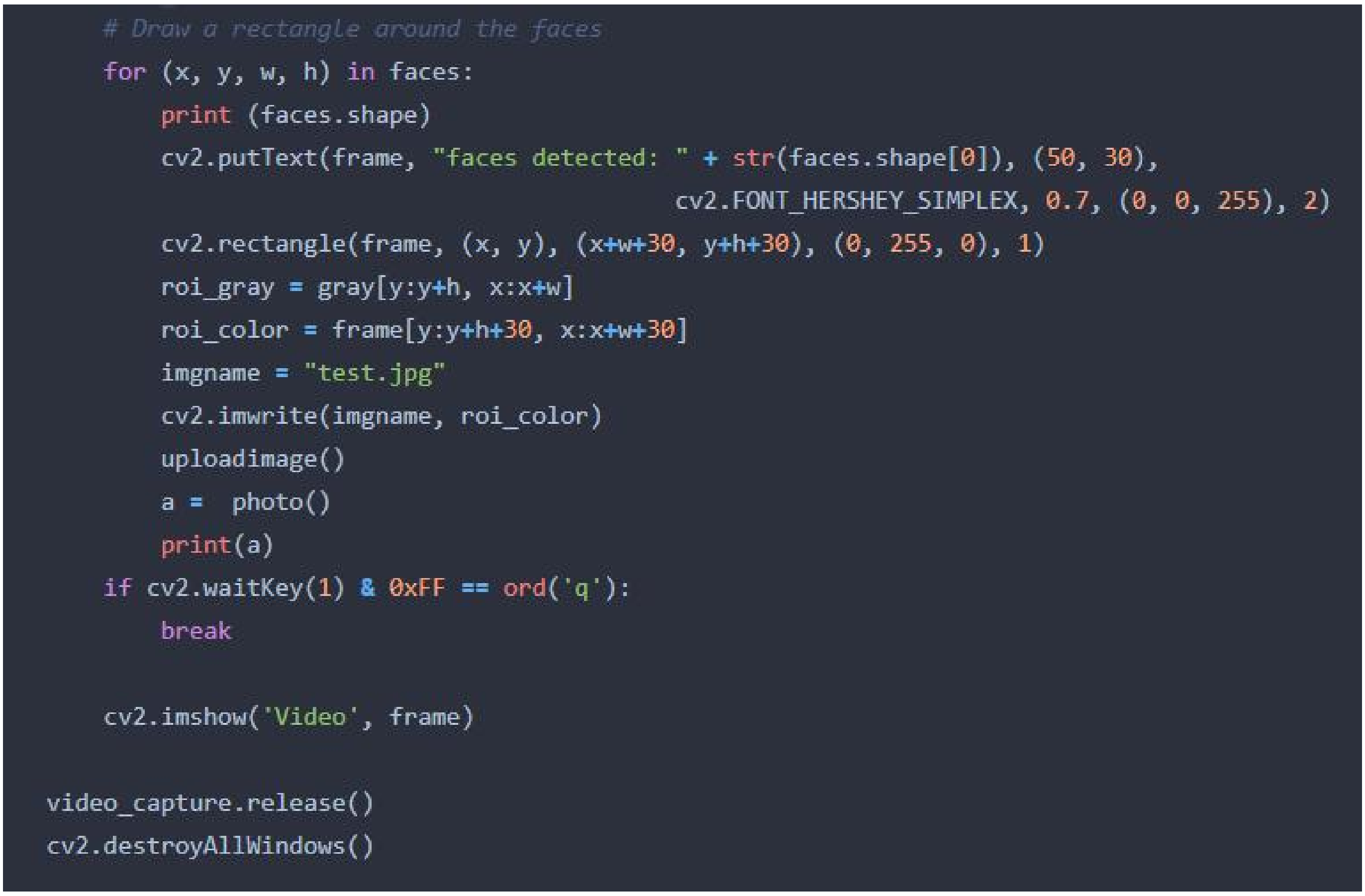
```
# Comparing of the captures image with Rekognition & S3
def photo():

    bucket = 'students123' # Replace with your bucket name
    collection_id = 'students'# Replace with your collection ID
    fileNames = ['test.jpg'] # Naming of Captured Image
    threshold = 70 # Threshold Limit for the similarity
    maxFaces = 2
    #here max faces is the number of faces it shoudl give as output if more than 1 face i
    #being rekognized with abover threshold confidence,
    for fileName in fileNames:
        response=client.search_faces_by_image(CollectionId=collection_id,
                                    Image={'S3Object':
                                           {'Bucket':bucket,
                                            'Name':fileName}},
                                    FaceMatchThreshold=threshold,
                                    MaxFaces=maxFaces)

        faceMatches=response['FaceMatches']
        print ('Matching faces')
        for match in faceMatches:
            print ('FaceId:' + match['Face']['FaceId'])
            print ('External Id:' + match['Face']["ExternalImageId"])
            print ('Similarity: ' + "{:.2f}".format(match['Similarity']) + "%")
```

Here, we are printing the FaceId, External Id & Similarity for the known captured images.

Now we will write the code for main funtion in which it reads the frames from the video and it compares with the trained one.



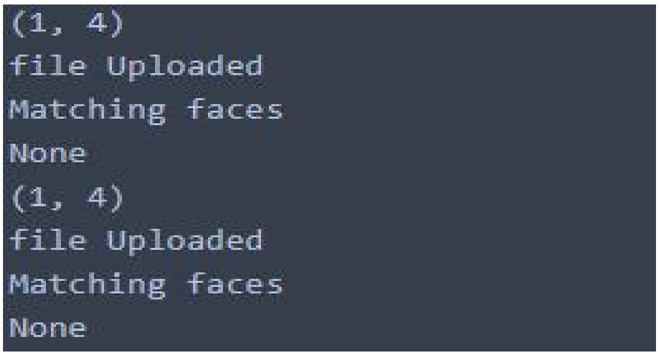
```
# Main Function
while True:
    # Reading of frames from video streaming
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
            scaleFactor=1.2,
            minNeighbors=5
           # minSize=(30, 30)
        )
```



```
    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        print (faces.shape)
        cv2.putText(frame, "faces detected: " + str(faces.shape[0]), (50, 30),
                                            cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.rectangle(frame, (x, y), (x+w+30, y+h+30), (0, 255, 0), 1)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h+30, x:x+w+30]
        imgname = "test.jpg"
        cv2.imwrite(imgname, roi_color)
        uploadimage()
        a =  photo()
        print(a)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

    cv2.imshow('Video', frame)

video_capture.release()
cv2.destroyAllWindows()
```

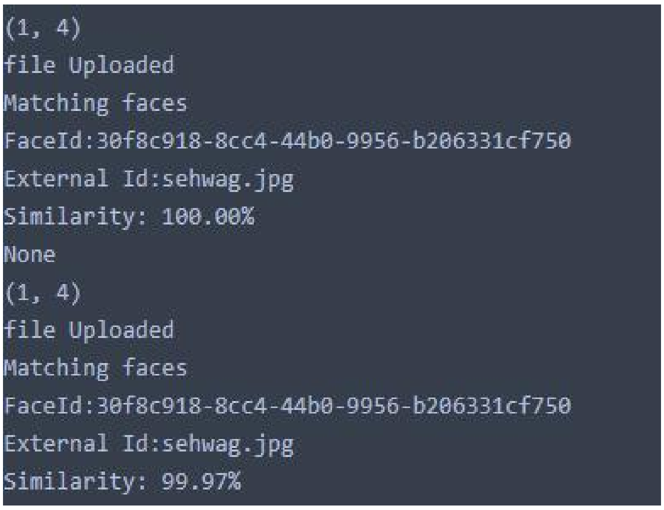
Now, execute the entire to compare the captured image with the trained one.

The outputs of the code for unknown image will be.

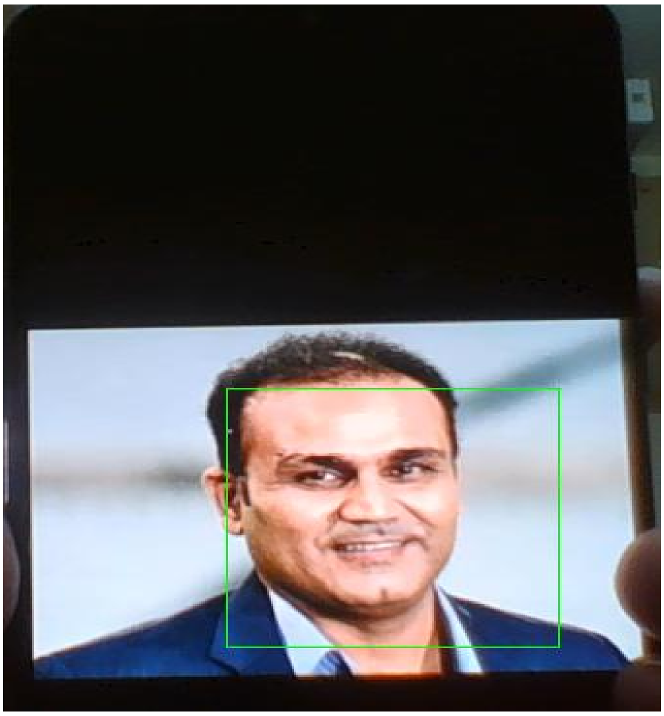


```
(1, 4)
file Uploaded
Matching faces
None
(1, 4)
file Uploaded
Matching faces
None
```

The outputs of the code for known image will be.



```
(1, 4)
file Uploaded
Matching faces
FaceId:30f8c918-8cc4-44b0-9956-b206331cf750
External Id:sehwag.jpg
Similarity: 100.00%
None
(1, 4)
file Uploaded
Matching faces
FaceId:30f8c918-8cc4-44b0-9956-b206331cf750
External Id:sehwag.jpg
Similarity: 99.97%
```



The output includes the allocated FaceId, External Id & Similarity (Accuracy) of the known person image.

3.2 HTML CODE AND PYTHON CODE

Html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Hourly Students Presence Calculation in Class using AWS</title>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
   <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
   <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
   <!link rel="stylesheet" href="style.css">
</head>
<style>
                .big{
                top:70;
                background-color:white;
                margin-top:80px;
                margin-left:550px;
                margin-right:550px;
                height:200px;
                border-radius: 25px;
                border: 3px solid #4a77d4;
                box-shadow: 6px 8px 4px grey;
                text-align:center;
                }
                .row{

                height:150px;
                }
                .col{
                        margin:10px;
                        margin-left:50px;
                        margin-right:50px;
                        border-radius: 25px;
                        border: 1px solid #4a77d4;
                        box-shadow: 0px 8px 4px grey;
                        text-align:center;
                }
                .ext{
                margin-top:25px;
                line-height:40px;
                }
                .ext1{
                margin-top:40px;
                line-height:50px;
                font-size:25px;
                color:#f95450;
                }

</style>
<body>

<div class="container-fluid">
<div class="header">
<div><b>Hourly Students Presence Calculation in Class using AWS</b></div>
</div>
   <br>
   <div class="big">
      <div class="box">
                <div      class="ext1"><font      size="20px">{{a}}</font><br><b>Total
Detections 3</b></div>
        </div>
   </div>
   <br>
```

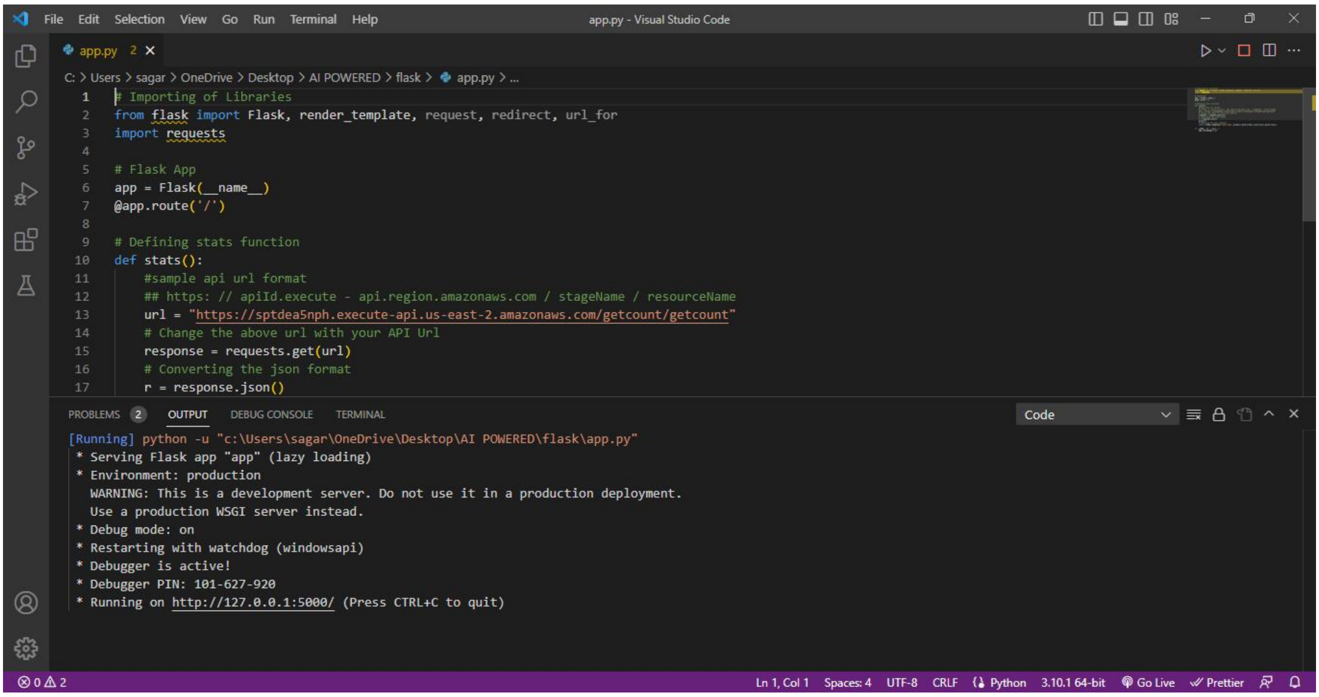
```
    <div class="row">
        <div class="col" >
                    <div class="ext">{{b}}<br><b>Ganguly</b></div>
          </div>
        <div class="col" >
                    <div class="ext">{{c}}<br><b>Sehwag</b></div>
          </div>
        <div class="col" >
                    <div class="ext">{{d}}<br><b>Sagar</b></div>
    </div>
    </div>
</div>
</body>
</html>
```

Python code:

Now we are writing the Flask code by which we can obtain the result of our model. we save the Flask code as App.py by excuting the code ouput will be generated in form of link.

It will show the local host where your app is running on http://127.0.0.1.5000/

Copy that localhost URL and open that URL in the browser. It does navigate you to where you can view your web page.



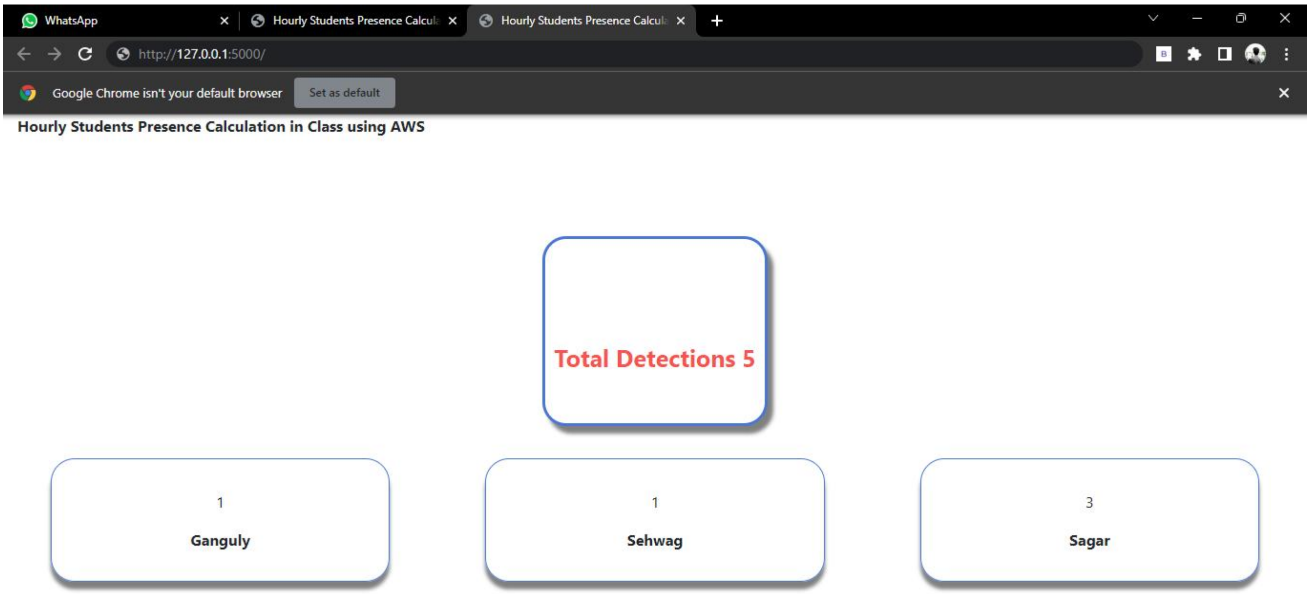
```
# Importing of Libraries
from flask import Flask, render_template, request, redirect, url_for
import requests

# Flask App
app = Flask(__name__)
@app.route('/')

# Defining stats function
def stats():
    #sample api url format
    ## https: // apiId.execute - api.region.amazonaws.com / stageName / resourceName
    url = "https://sptdea5nph.execute-api.us-east-2.amazonaws.com/getcount/getcount"
    # Change the above url with your API Url
    response = requests.get(url)
    # Converting the json format
    r = response.json()
```

```
[Running] python -u "c:\Users\sagar\OneDrive\Desktop\AI POWERED\flask\app.py"
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 101-627-920
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig6: Excuting the Flask code to get Output.

4.RESULT

Enter the url in the browser to check the output of the model.
the output will be look like as below:


WhatsApp
Hourly Students Presence Calcul
Hourly Students Presence Calcul
http://127.0.0.1:5000/
Google Chrome isn't your default browser
Set as default

Hourly Students Presence Calculation in Class using AWS

Total Detections 5

1
Ganguly

1
Sehwag

3
Sagar

Fig7: Final output of the model.

5. ADVANTAGES AND DISADVANTAGES

Advantages :

1. Cost-Effective
2. Increses security
3. Time Saving
4. Easy to Manage
5. High Accuracy Rate

Disadvantages :

1. Data privacy breach
2. Network issues

6.APPLICATIONS

1. Contact less biometric attendance system in educational institutes and offices.
2. Airport security increases.
3. In Warehouse, control process to provision entry and exit of vehicles
4. Identifying the criminals in Department.

7. CONCLUSION

Our team is implementing an attendance system that automates the process to reduce errors that result from manually taking attendance. By installing cameras in classrooms, the system can monitor student interest and mark attendance automatically, potentially increasing student attendance. An Artificial Intelligence-based solution can help monitor and mark attendance accurately, encouraging students to attend school or college regularly. The developed system is efficient and saves valuable time for students and lecturers, while also reducing paper usage and providing timely reports. The system eliminates most administrative tasks and minimizes human errors, avoiding issues such as proxy punching and time-related disputes.

8. FUTURE SCOPE

Facial recognition technology has a wide range of potential applications that can bring significant benefits. Some of these applications include:

10. Improving ATM security and preventing fraud in India: By creating a database of all ATM card users in the country and installing facial recognition systems at ATMs, the technology can be used to match the user's photo with the stored photo in the database for access permission.

11. Identifying and reporting duplicate voters in India during elections.

12. Streamlining passport and visa verification for efficient processing.

13. Enabling faster and more accurate driving license verification.

14. Enhancing surveillance and security at important locations such as airports and government buildings.

15. Verifying the identity of candidates during exams such as Civil Services, SSC, IIT, and MBBS.

16. Deploying the technology for verification and attendance tracking at government offices and corporate.

17. Improving access control verification and identification of authorized users at bank lockers and vaults.

18. Identifying criminals and enhancing law enforcement efforts by the police force.

These applications can help save time, reduce human errors, increase security, and improve the overall efficiency of various systems.