**Smart Attendance Management System Using**

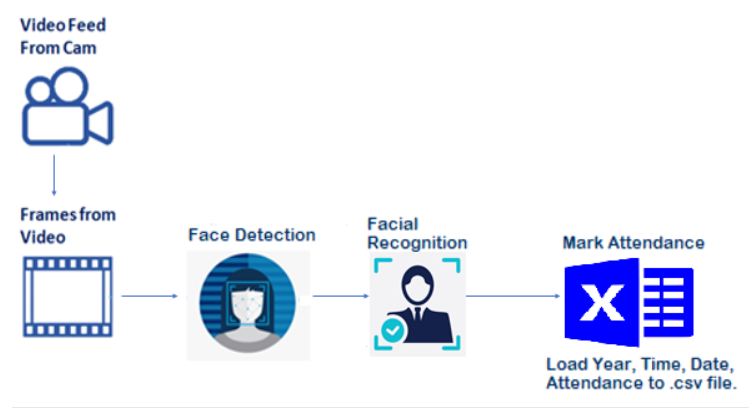
**Face Recognition Technology**

**Project Description:**

Maintaining attendance is very important in all organizations for checking the performance of an Employee. Every organization has its own method in this regard. The traditional approach was marking attendance manually using the old paper or file-based approach and some have adopted methods of automatic attendance using biometric techniques. But in these methods employees have to wait for a long time in making a queue when they enter the office.

We are going to build this project using dlib which uses 128-point face detectors which output these 128 points from all the faces and compare them with existing faces. This model uses the integrated webcam to capture the video frame. The image of the person captured in the video frame is compared with the encodings of the faces of the pre-trained model. If there is a match, then the image of the employee is recognized and attendance is marked along with the timestamp in a log file(excel file).

**Architecture:**



**Prerequisites:**

**In order to develop this project we need to install the following software/packages:**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform,  package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder

1. **To build a smart attendance system you must require the following packages**

* Install dlib
  + dlib is an open-source library used for face detection. We recommend installing dlib updated version for better compatibility with OpenCV. Given a face, dlib which is a pre-trained model can extract features from the face like eyes, nose, lips, and jaw using facial landmarks.
  + For installation of dlib refer to the below link(in anaconda prompt) <https://tinyurl.com/Requiredinstallations>
* Install face-recognition
  + Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.This also provides a simple face\_recognition command line tool that lets you do face recognition on a folder of images from the command line
  + For installation of face-recognition refer to the below link(in anaconda prompt) <https://pypi.org/project/face-recognition/>
* Install OpenCV
  + [OpenCV](https://en.wikipedia.org/wiki/OpenCV) is a library of programming functions mainly aimed at real-time computer vision. Here, OpenCV is used to capture frames by accessing the webcam in real-time.
  + Open anaconda prompt and type command “pip install opencv-python”.

**Prior knowledge**

One should have knowledge of the following concepts

* Dlib face recognition: <https://youtu.be/7kjxxP2N3lU>
* Open CV: <https://youtu.be/-ZrDjwXZGxI>
* Flask Basics: <https://youtu.be/lj4I_CvBnt0>

**Project Objectives**

By the end of this project you will:

* know fundamental concepts of Python, Deep learning  Computer vision, loading log file to excel.
* gain a broad understanding of face detection and recognition.
* know how to install necessary packages and setting up the environment for Facial Recognition.
* know how to build a web application using the Flask framework.

### **Project Flow**

We can divide the process of developing  smart attendance system into the following steps:

1.Open the integrated webcam when the user clicks mark attendance button.

2. Detecting the face in the video stream.

3. Recognising the face in the video stream.

4. Displaying bounding boxes around the detected face with the name.

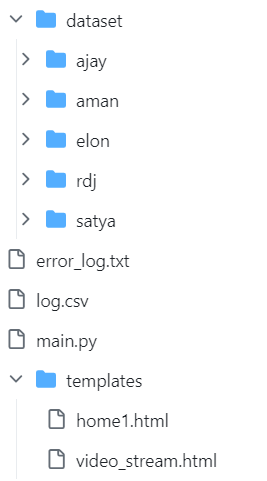
5. Mark the attendance along with name and time in a log file(excel file).

To accomplish this, we have to complete all the activities and tasks listed below

* Create dataset
* Face detection using dlib
* Import Necessary Libraries
* Face Encoding
* Video Capture
* Capturing the input frames
* Face Recognition
* Convert the image from BGR to RGB
* Encoding the input image
* Compare input image with known faces
* Mark Attendance
* Load the data to CSV file
* Displaying the result
* Application Building
* Build an HTML Page
* Build the python flask app
* Run the application

**Project Structure**

### Create the Project folder which contains files as shown below



### Activity-1:

### **Import Necessary Libraries**

**face\_recognition:**

dlib is an open-source library used for face detection. We recommend installing dlib updated version for better compatibility with OpenCV. Given a face, dlib which is a pre-trained model can extract features from the face like eyes, nose, lips, and jaw using facial landmarks.

**OpenCV:**

[OpenCV](https://en.wikipedia.org/wiki/OpenCV) is a library of programming functions mainly aimed at real-time computer vision. Here, OpenCV is used to capture frames by accessing the webcam in real-time.

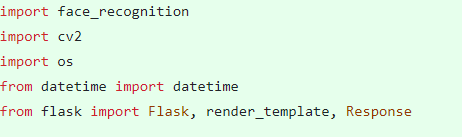
**os:**

This library provides numerous tools to deal with filenames,paths, directories.

**datetime:**

datetime function supplies classes for manipulating dates and times.

**All the above modules can be imported into our program using the below code**



### **Create Dataset**

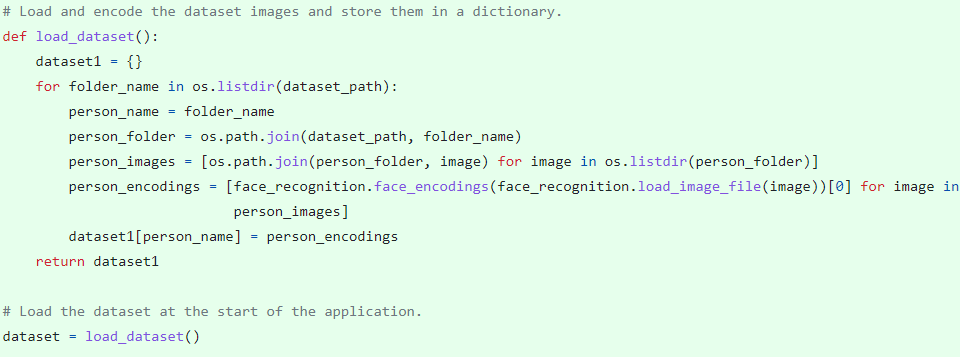
Collect face images of employee’s organized into subdirectories based on their respective names as shown in the project structure.Create folders of people who need to be recognized.

In this project, we have collected images of 4 persons ajay, aman, elon, satya and they are saved in the respective subdirectories with their respective names.



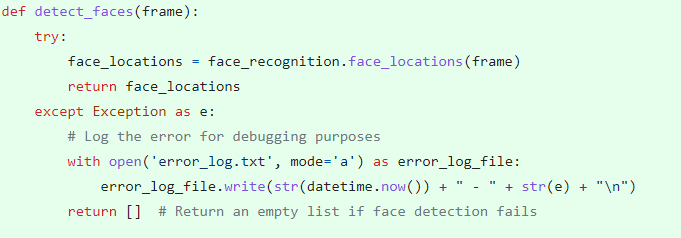
**Activity 2: Face Encoding**

We are now going to use dlib’s face recognition to detect all the faces in the image. This face detection model is based on HoG and SVM.

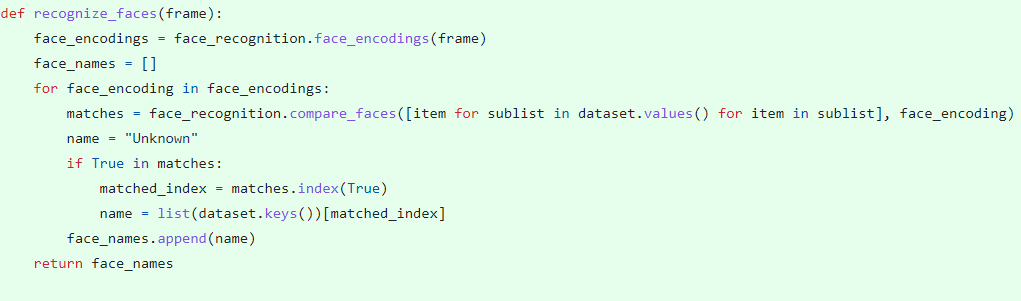


First, we load the image file of the characters ajay,aman,elon and satya using load\_image\_file function which is a predefined function in face\_recognition library. The loaded images are encoded into a list of 128 numbers. This is known as encoding the face into a vector and face\_recognition.face\_encodings  method handles it.

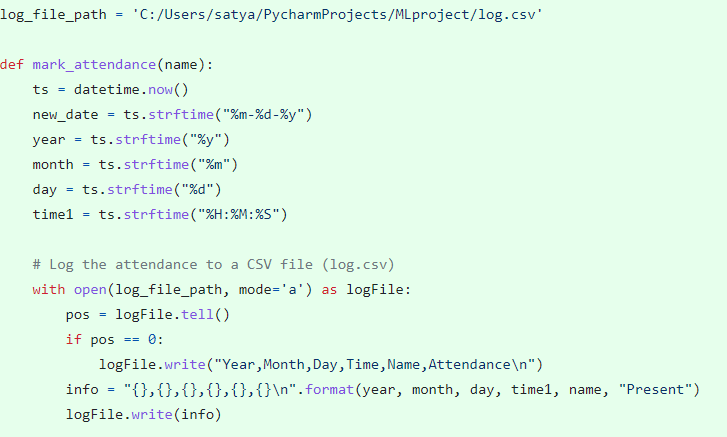
**Face detection using dlib**



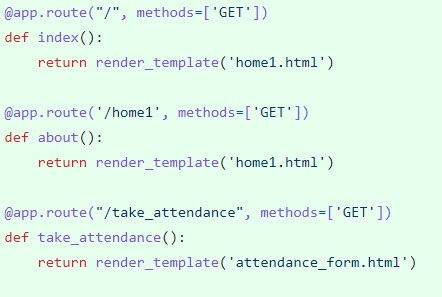
**Activity 3: Face Recognition**



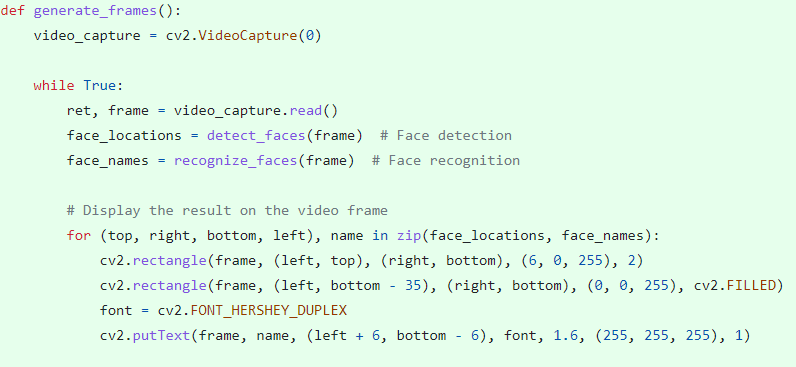
**Activity 5: Mark Attendance**



**Activity 6: Build the flask app**



**Activity 7: Video capture and Processing**



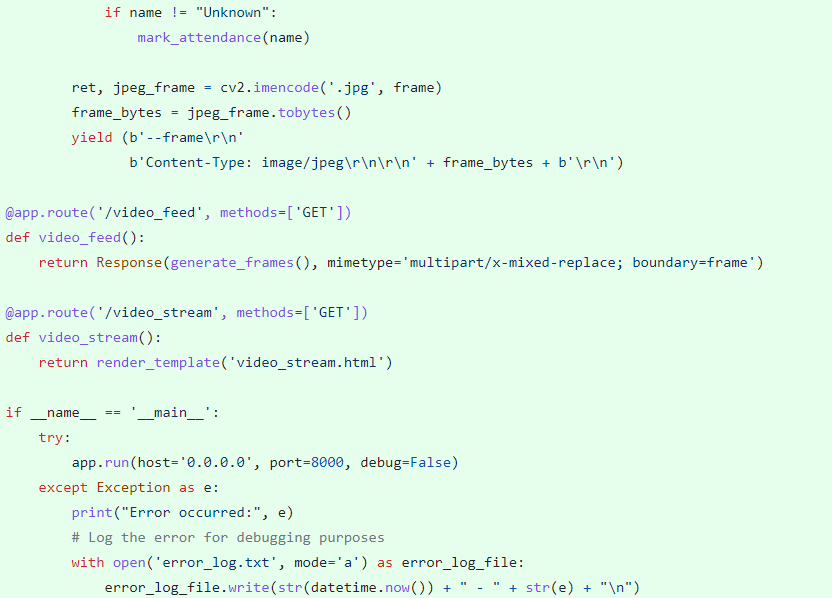
**Activity 8: Display result and log attendance**

After finding the match we have stored the name of the person recognised in the name variable.

Then, we have to mark the attendance of the person along with timestamp and date in the CSV file.

After we recognise the name of the detected image, we have to display the same on the live video stream with bounding boxes for visualization.

Append the names of the person identified to the list face\_names.



**Conclusion:**

**Activity 9: Application Building**

Now that, we have understood the concept and coding of how to detect and recognize a face and mark the attendance by writing the log to .csv file. Let us build an application for smart attendance system.

We will be using python for server-side scripting. Let’s see step by step process for writing backend code.

**Build An HTML Page**

We Build an HTML page to take the input live video stream upon clicking the button for mark attendance, it has to redirect to url for “predict” which returns opens the webcam. The output is to be then displayed on the live video stream. The HTML pages are put under templates folder and any style sheets if present is put in the static folder.

* We use HTML to create the front end part of the web page.
* Here, we created 2 HTML pages- home1.html, video\_stream.html.
* Home1.html displays the home page
* Video\_stream.html displays that attendance is successfully marked.

### Full-Stack Web Development Internship Program: http://bit.ly/fs-int..

<https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Build The Python Flask App**

In the flask application, the input parameters are taken from the HTML page These factors are then given to the model to know if the Visa is approved or not and is sent back to the HTML page to notify the user.

In the flask application, whenever the user interacts with UI and press “Mark Attendance” button,  integrated web cam is opened to take the live video stream. The face is detected and recognized and attendance is marked in the log.csv file with timestamp.

**Task 1: Importing Libraries**

Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument Pickle library to load the model file.

**Task 2: Routing to the html Page**

Here, declared constructor is used to route to the HTML page created earlier.

In the above example, ‘/’ URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page is rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Here, “home1.html” is rendered when home button is clicked on the UI

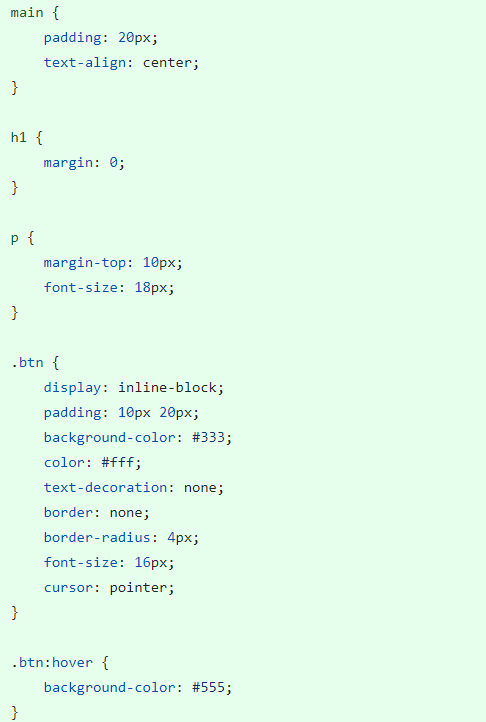
**When “Mark Attendance “ is clicked on the UI, predict function is executed.**

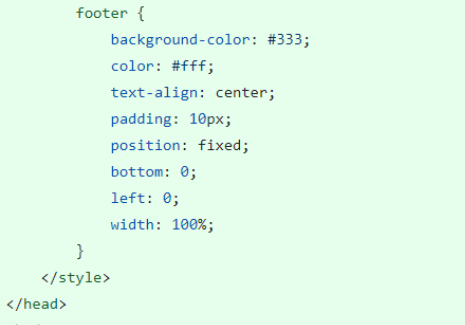
**Task 3: Main Function**

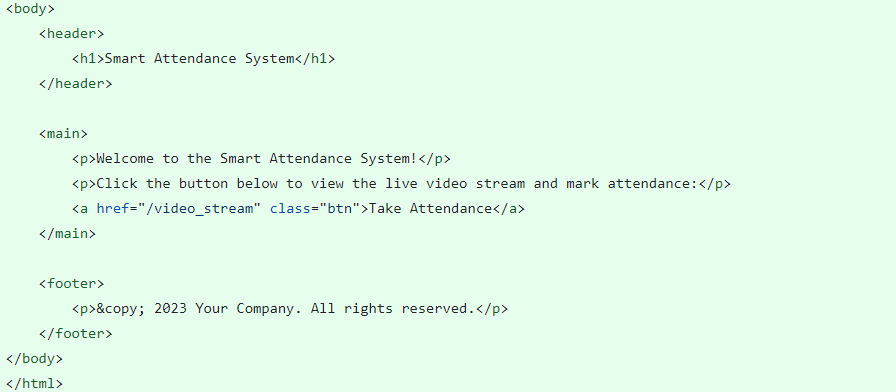
This is used to run the application in a localhost.

**Home1.html**

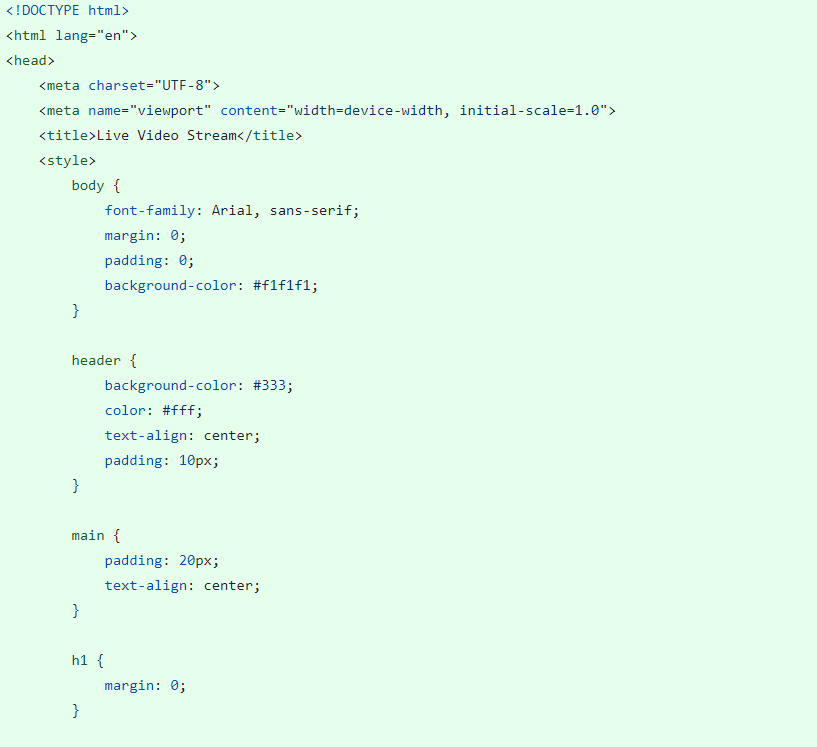


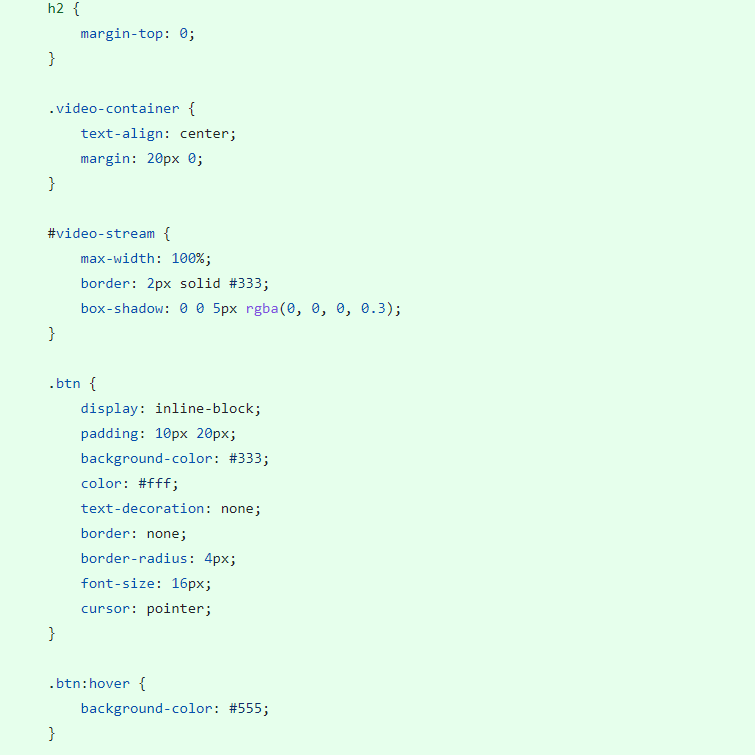


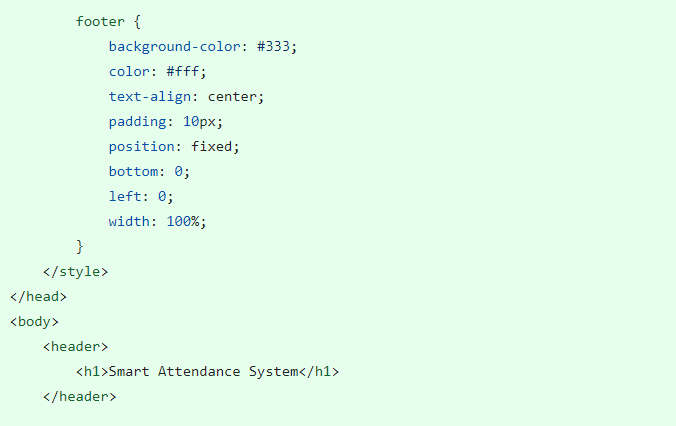


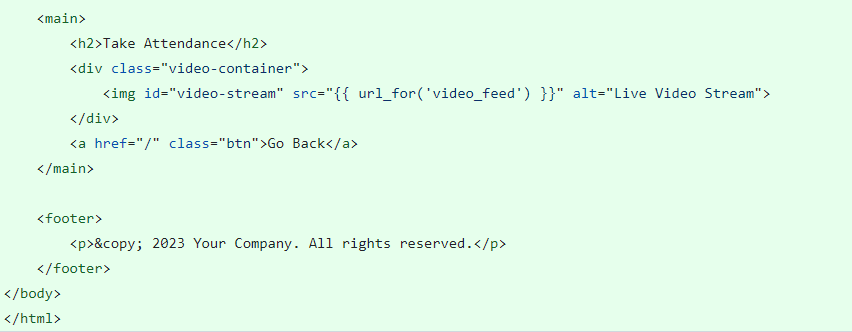


**Video\_stream.html**









### **Run The Application**

* Open the anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on http://127.0.0.1.8000/
* Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.
* Enter the values, click on the predict button and see the result/prediction on the web page.
* Click on mark attendance
* Click on another spyder window which is opened to view the opened web cam.
* The attendance of the person with name and time stamp is saved in the log.csv file.

