FACE RECOGNITION
BASED
SMART ATTENDANCE TRACKER

INTRODUCTION

- Based on Face Recognition Technology
- Recognize the face of each staff and automatically mark their attendance as present
- Automatically figures out the absent staffs
- Automatically sends day to day attendance report to the manager through an email notification.

REQUIREMENTS

install python 3.6
install flask
install git
opencv-python==4.2.0.34
tensorflow==2.4.0
sklearn==0.0
Pillow==8.1.0
PyMySQL==1.0.2
pandas==1.1.5
gTTS==2.2.1
APScheduler==3.7.0

HOW TO RUN THIS PROJECT

(The code is tested and implemented in 3.6 so install python 3.6)

- **Install Anaconda Prompt**
  https://www.anaconda.com/products/distribution

- **Install git (if not installed)**
  https://git-scm.com/download/win
  click on **64-bit Git for Windows Setup**.

- **Download GitHub repository on your local computer**

- After downloading, you have to open Anaconda prompt to run this project.
  Before running any files, you have to set up virtual environment in the directory where the project is located and install all the dependencies required for this project.

  Creating virtual environment enable us to install the dependencies virtually for this project only without affecting the python dependencies on your computer.

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.

- First of all, you have to install virtual environment tool to create one.

  Recommended (on Anaconda Prompt)
  For installing virtual environment on Anaconda Prompt (Windows):
  conda install -c anaconda virtualenv

- After installing virtual environment, you have to install all the dependencies required to run this project in your virtual environment. For doing so you have to follow the following steps:
  First of all, you have to change your working directory to the location of this repository in your computer by using the following command:
  cd /*location to the repository */

- After changing the working directory to the current repository/project create a virtual environment by using the following commands:

  Recommended
  On Anaconda Prompt (Windows)
  conda create -n "your virtual environment name" python=3.6
  (The code is tested and implemented in 3.6 so install python 3.6)
  e.g.
  conda create -n “sams” python=3.6

- After creating a virtual environment in a working directory, you need to activate the virtual environment:

  Recommended
  On Anaconda Prompt (Windows):
  conda activate "your virtual environment name"
  e.g
  conda activate sams

- Now you need to install all the requirements and dependencies for running this project.
  pip install -r requirements.txt
  Install the dependencies by seeing the requirements.txt file.

- Write command to install flask
  pip install flask

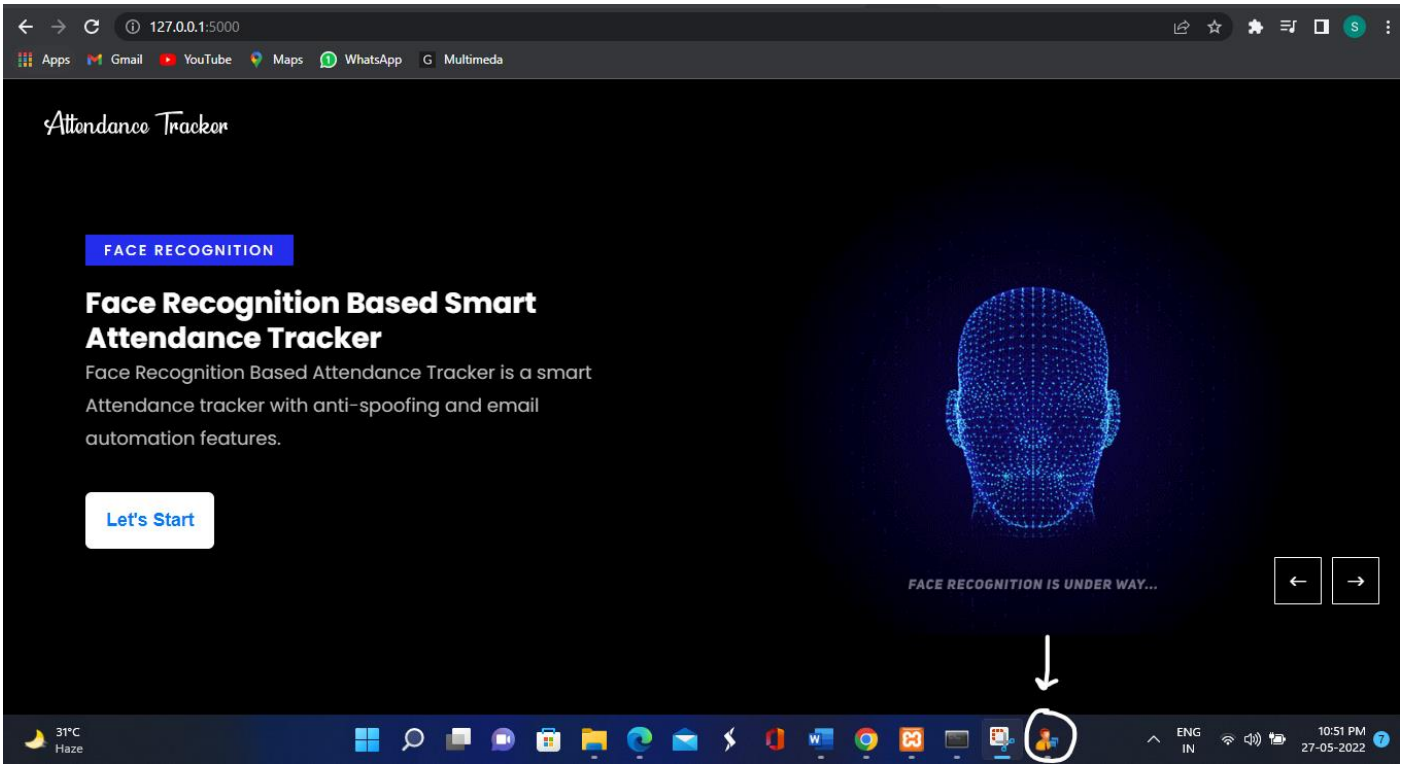
- Run project with command
  python attendance_with_antispoofing.py

- After that

```
(sams) C:\Users\sunai\Downloads\demo_face>python attendance_with_antispoofing.py
2022-05-27 21:48:14.069822: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll
 not found
2022-05-27 21:48:14.070045: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
 * Serving Flask app 'attendance_with_antispoofing' (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 stat
2022-05-27 21:48:20.376081: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll
 not found
2022-05-27 21:48:20.376218: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
 * Debugger is active!
 * Debugger PIN: 101-105-973
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Run http://127.0.0.1:5000/ this command on your system

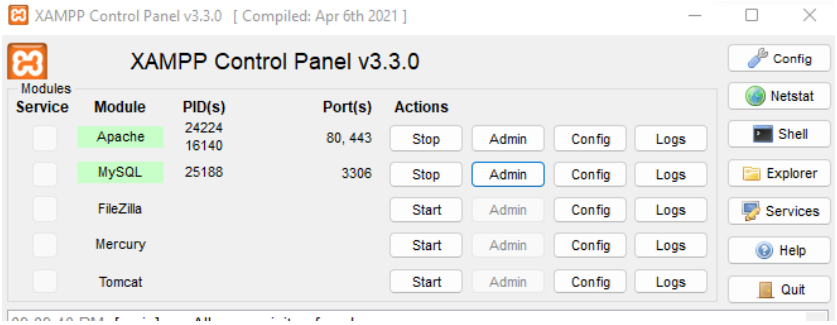
- Click on Let's Start Button



Indicated icon will pop out on clicking **Let's Start, click on that and start the project**

!! IMP SETUP DATABASE BEFORE RUNNING PROGRAM

- Install Xampp Control Panel
  https://www.apachefriends.org/download.html



Click on Apache, MySQL then Click on admin of MYSQL, it will open new page (localhost PhpMyAdmin)

Follow Steps to create database:

- Click on new , enter the ‘recognition’ database name and enter create
- After that go to import and chose **recognition.sql** file from project folder and click go
- Hence database created successfully.

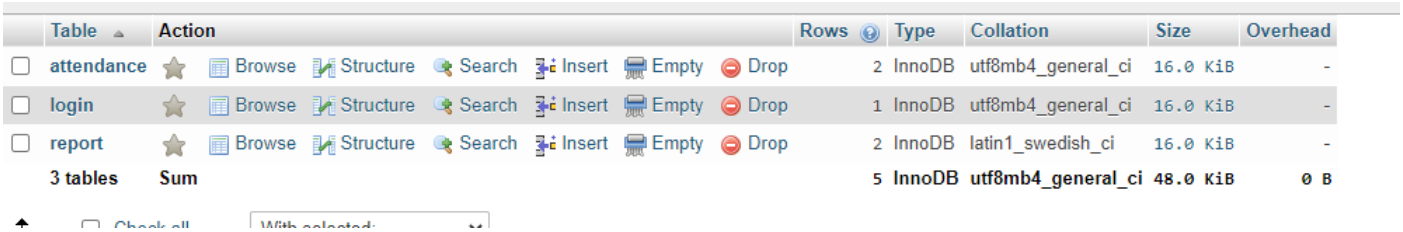


	Table	Action							Rows	Type	Collation	Size	Overhead
☐	attendance	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	login	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	report	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
	3 tables	Sum							5	InnoDB	utf8mb4_general_ci	48.0 KiB	0 B

NOW RUN THE PROJECT WITH

USERNAME: Admin
PASSWORD: 123

EMAIL AUTOMATION:

- Open event_scheduler.py from folder
- Enter your email id and password at required places and set your time for email automation.

MAIN FEATURES OF THE PROJECT

- Email Automation
  - Automatically sends day to day attendance report to the manager through an email notification.
  - Also sends emails to absent employees regarding their absence.

- Liveness detection method for checking the frauds
  - This project has anti spoofing features means Facial anti-spoofing is the task of preventing false facial verification by using a photo, video, mask or a different substitute for an authorized person’s face.

- Do not mark attendance multiple times for the same person in the same day
  - It means suppose a staff has marked his attendance at a time then again at the same time he tries to mark attendance then system will show warning that attendance has already been marked. Staff can mark attendance again on next day only.

- Face recognizer module recognizes the staff with showing name and id on the camera.

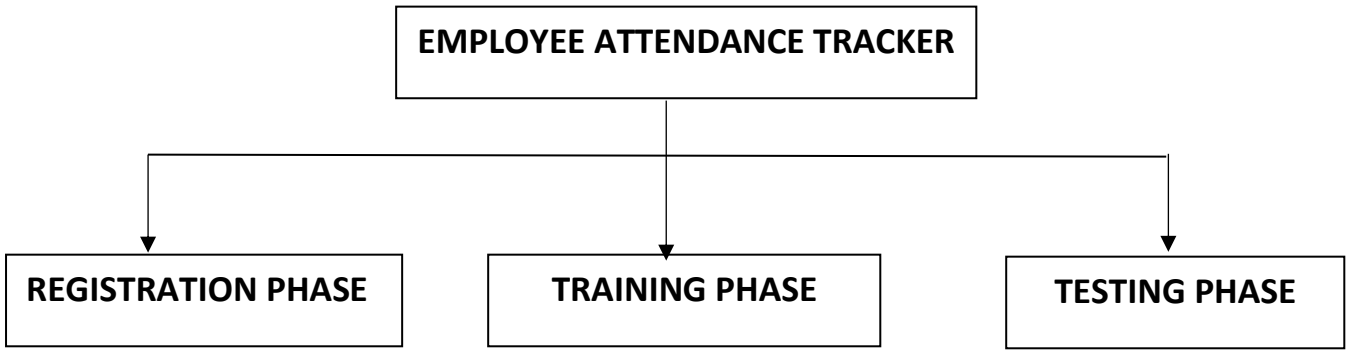
- Feature of changing password and username at any time.
  - Through admin account, admin can change password and username.

- Feature of adding, clearing, updating and deleting the employee details, and also contains the features of searching the data.

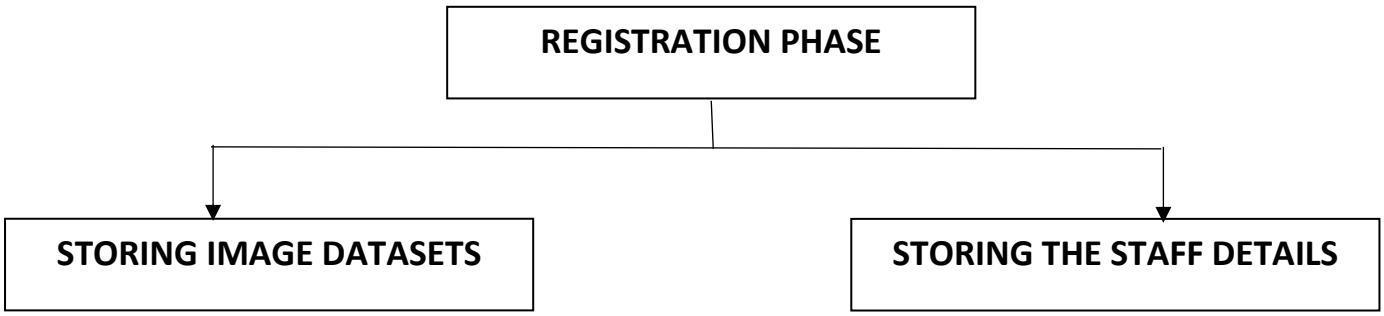
- Attendance report shows details of employees with his present or absent status and time of attendance marked and also contains the features of searching the data.

- Can update the photos of the staff as well.

METHODOLOGY



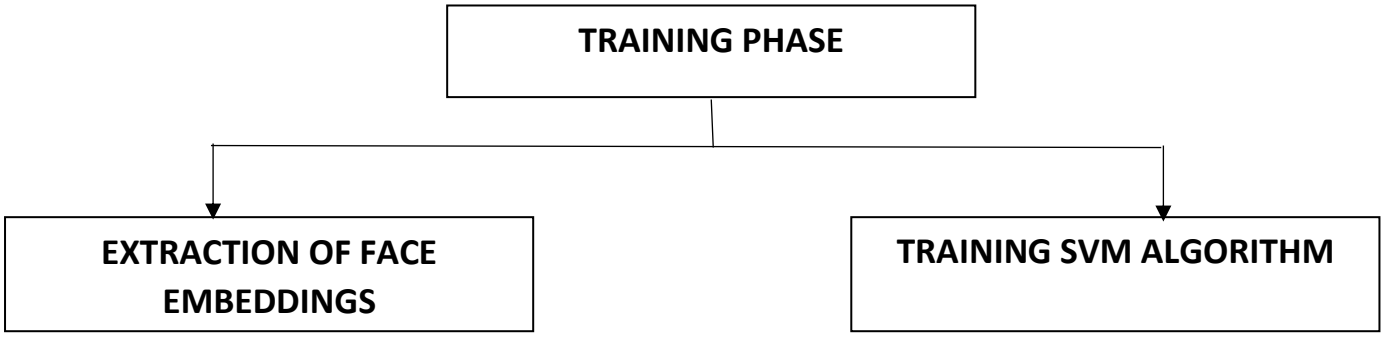
1. REGISTRATION PHASE



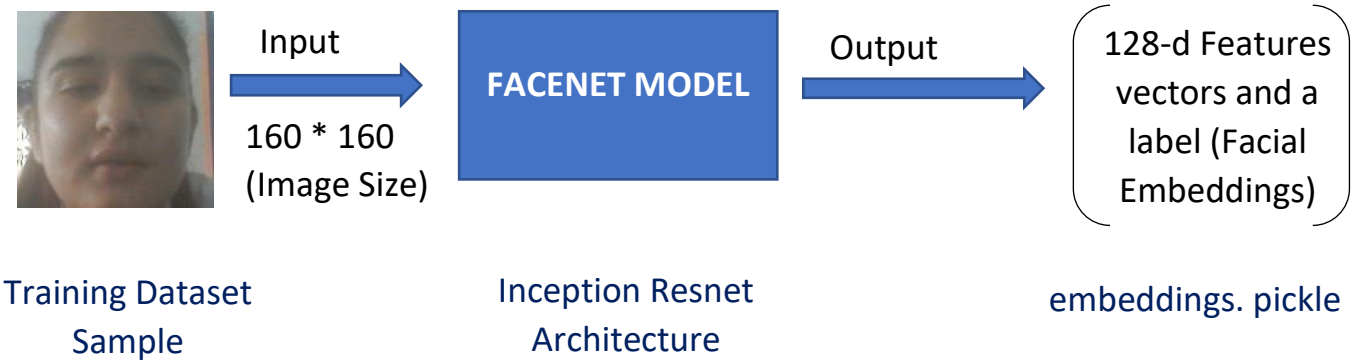
Steps involved in registration phase:

- Fill up the staff details
- Initialize the web camera
- Capture 50 images of each staff from the webcam.
- Face detection
- Image Resizing
- Destroy the web camera after capturing the 50 images.
- Storing images in local directory
- Inserting the staff details into the database.

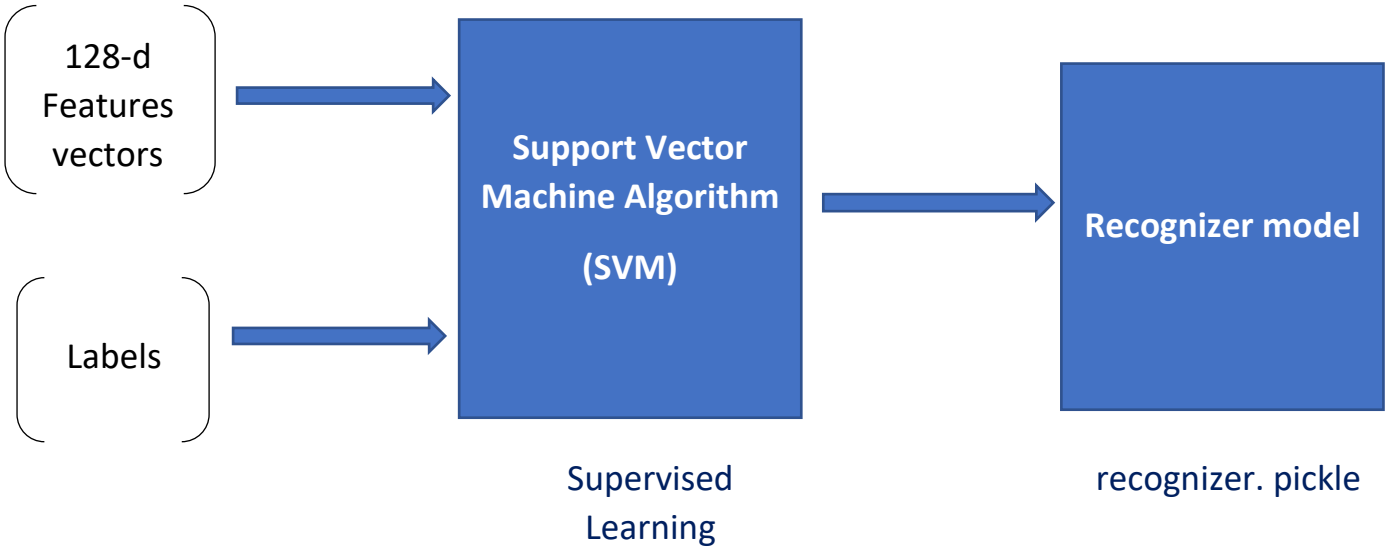
2. TRAINING PHASE



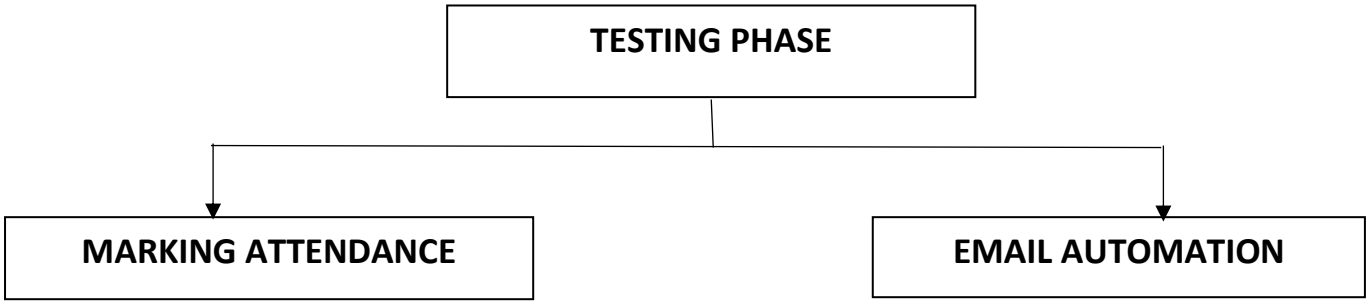
2.1 Extraction of face embeddings



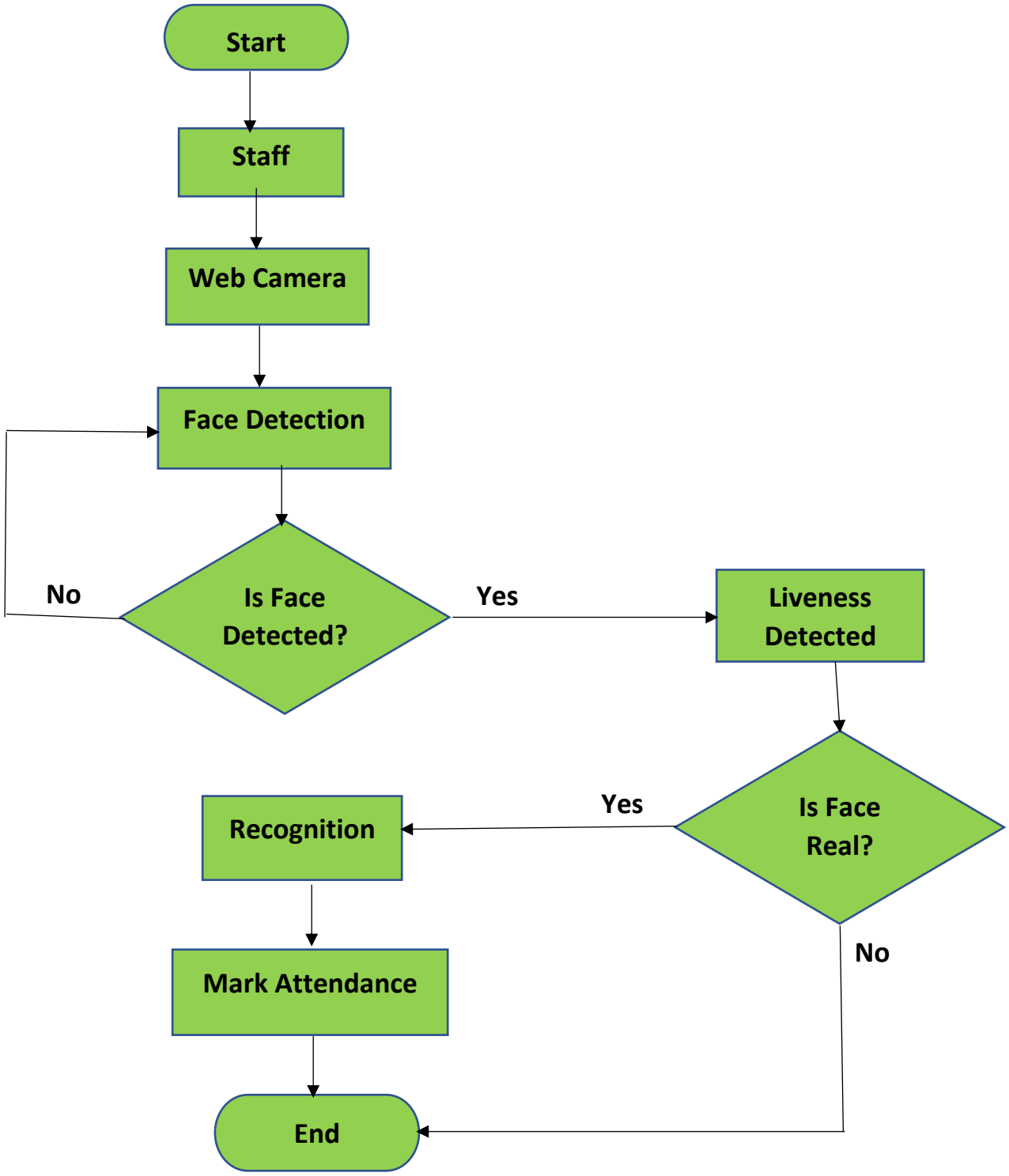
2.2. Training



3. TESTING PHASE



3.1 Marking Attendance



3.2 Email Automation

- Assumptions of timing boundaries
- If staff not present in given time interval:
  - Attendance marked as absent
  - Email sent to each absent employee
  - Attendance report of that day is sent to the Manager through email.