Face Attendance System

MySQL installation in VM (same as in local machine):

Run the following commands in the terminal to install mysql server in Ubuntu 20.04 :

- sudo apt update
- sudo apt install mysql-server

Check if mysql is running or not:
- sudo systemctl status mysql

Now install mysql securely

- sudo mysql_secure_installation

Set a password and follow the instructions in the terminal.

Video Link: https://www.youtube.com/watch?v=7VJiUJaF784
Weblink:https://support.rackspace.com/how-to/install-mysql-server-on-the-ubuntu-operating-system/

MySQL Initial Setup:

Login to mysql as root user and then create a New User using the command

- CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';

At this point newuser has no permissions to do anything with the databases. In fact, even if newuser tries to login (with the password, password), they will not be able to reach the MySQL shell.
Therefore, the first thing to do is to provide the user with access to the information they will need.

- GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
- FLUSH PRIVILEGES;

This will make the changes into effect.

Link:https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql

Note: The MySQL installation should be done in the VM only.

After MySQL Installation is successful and the user is created with all privileges.
Login to MySQL using the user credentials created by you.

Create a database named ‘**MyDatabase**’ using the command:

- `CREATE DATABASE MyDatabase;`

Creating a database does not select it for use; you must do that explicitly. To make ‘MyDatabase’ the current database, use this statement:

- `USE MyDatabase;`

Now for our app create Three Tables in ‘MyDatabase’

1. StudentRecord : Stores the information about each Student who registers in the app.

```
CREATE TABLE `MyDatabase`.`StudentRecord` (
  `Id` VARCHAR(45) NOT NULL,
  `Name` VARCHAR(45) NULL,
  `Image` LONGBLOB NULL,
  `Email` VARCHAR(45) NULL,
  `DOB` VARCHAR(45) NULL,
  PRIMARY KEY (`Id`));
```

2. TeacherRecord : Stores the information about registered teachers.

```
CREATE TABLE `MyDatabase`.`TeacherRecord` (
  `TeacherName` VARCHAR(45) NULL,
  `Email` VARCHAR(45) NULL,
  `password` VARCHAR(45) NULL,
  PRIMARY KEY (`TeacherName`));
```

3. TeacherClasses: Stores the TeacherName along with the ClassName associated with the teacher.

```
CREATE TABLE `MyDatabase`.`TeacherClasses` (
  `TeacherName` VARCHAR(45) NULL,
  `ClassName` VARCHAR(45) NULL);
```

<u>Note: These three tables need to be created in the database before running the web application. Make sure to change the credentials for connecting to the database via mysql-connector in the</u>

main.py file with the credentials of the newuser created by you for the database. Change the code given below to your requirements in the main.py file.

```
def CONNECTION():
    mydb =  mysql.connector.connect(

        host="localhost",
        user="<user_created>",
        passwd="<your_password>",
        database="MyDatabase",
        use_pure="True"
    )
    return mydb
```

Setup for Web Application:

Step 1 : Clone the repository on your Virtual Machine:

https://github.com/Abhishek9934/FACE_ATTENDENCE_SYSTEM.git

Step 2: Create a virtual environment on your VM

- Using "**python -m virtualenv <venv_name>**"
- Activate the virtual environment using "**source <venv_name>/bin/activate**" command.
- Enter the project directory and install all dependencies of your virtual environment. Use "**pip install -r requirements.txt**" command to do this. This will install all the dependencies.

Step 3: Download the pretrained facenet model in your Virtual Machine. Visit this link to download the file.
https://www.kaggle.com/yhuan95/face-recognition-with-facenet/data?select=facenet_keras.h5

facenet_keras.h5(88.12 MB)

- Once you have downloaded the "facenet_keras.h5' file in your local machine upload it to your VM machine an place the file in the same directory as that of the "main.py" file.

Step 4: Run the flask app using Gunicorn

- Just type the following command in the terminal of your VM to make the website live (You should activate the virtual env and must be in the directory containing the 'main.py' file to execute this command successfully) “**gunicorn -w 2 -b :<port_number> main:app --threads 3 --timeout 120 --daemon** “.

For example, for deploying on port 9000(say) use the command:

- gunicorn -w 2 -b :9000 main:app --threads 3 --timeout 120 --daemon

This makes the app run in the background and sets it free from the terminal. To know more about gunicorn visit this link https://docs.gunicorn.org/en/latest/settings.html

You can stop the site by typing the following command in VM terminal:

- pkill gunicorn .

- The site can be accessed using the url “**http://<external_ip>:<port_number>**” where external_ip is public ip.