**VMS Instruction Guide**

For Front-end Angular installation

**Step :- 1**

Install Angular CLI:-

In terminal run:- npm install -g @angular/cli

**Step:- 2**

Open the project folder in VScode

In terminal run:- npm install

**Step:- 3**

Now add some modules one by one in the terminal

* npm install chart.js
* npm install --save-dev @types/chart.js
* npm install html2pdf.js
* npm install --save-dev @types/html2pdf.js
* npm install jspdf html2canvas
* ng add @angular/material
* -Select prebuilt theme name : Purple/Green
* -Write yes to set up global Angular Material typography style
* -Select include and enable animations

**Step:- 4**

In terminal run:- npm install

**Step:- 5 (Final step)**

In terminal run:- ng serve

For backend Python installation

**Step:- 1**

Downloading Python 3.12. For windows :-

* <https://www.python.org/downloads/release/python-3122/>
* Install the downloaded exe/installer file
* In cmd :- “python -V”

Downloading Python 3.12. For Linux ubuntu :-

* sudo apt update
* sudo apt install python3.12
* python3 --version

And for required modules :-

* We have created a freeze file named **requirements.txt** containing all the required modules to be installed

**Step:- 2**

Then open VS code editor and create a folder named VMS\_backend and add the following VMS files in the folder from the git repository .

Then install the extension for python by microsoft in VS code extensions if needed

**Step :- 3**

For database connection in our VMS\_backend folder we have **connection.py** file with all details

To connect with postgresql database following are the required credentials:-

* Database name = ”VMS”
* Database username = “postgres”
* Database password =”root”
* Database host=”localhost”
* Database Port = 1024
* **Step:- 4 (Final step)**

Now run python file name “app.py” by clicking run button on top right or by writing the command in VS code’s terminal as:-

“& "C:/Program Files/Python312/python.exe" c:/Users/user1/Desktop/VMS/app.py”

Replace the file path according to your installed directory for app.py to run app.py.

**Now the list of which routes allocated inside which files :-**

1. app.py :- This file is the main file to run for python running on localhost with port=8000.
2. register.py :-This file contains the “/register/” route for user registration into the database.
3. login.py :- This file contains “/login/” route for users registration into system as anonymous role.

* Also it contains the “/logout/” route for user login into the system.

1. reset\_password.py :-This file contains “/reset-password/” route for users password resetting.

* Also it contains the “/forgotpassword/” route for users if they forgot the password.

1. add\_camera.py :- This file contains “/add\_camera/” route for users integrating camera into system.

* Also it contains the “/add\_camera\_options/” route for users to add more dynamic camera options in the system such as location,camera\_type,etc.
* Also it contains the “/get\_camera\_options/” route for users to select/fetch the added options .

1. camera\_list.py :- This file contains the “/camera\_list/” route for users to fetch the list of cameras added in the system.

* Also it contains “/update\_camera/” route for user to update/edit the integrated camera options .
* Also it contains the “/delete\_camera/” route for user to delete the added camera from the system .

1. camera\_ip\_feed.py :- This file contains the “/camera\_ip\_details/” route for users to send the Ip camera’s credentials to the backend.

* Also it contains the “/camera\_ip\_feed/” route to generate video feeds for the desired ip camera and send it to the frontend for displaying the live-view.
* Also it contains “/start\_recording/”, “/stop\_recording/”, “/start\_recording/”, “/schedule-recording/”and “/capture\_image/” routes for recording and storing the video feed in video and image formats

1. users\_list.py :- This file contains “/users\_list/” where list of all the users is fetched

* Also it contains”/update\_users/” to update the activeness and change the role assigned for the users.
* Also it contains the “/users\_profile/” route for a particular user to display his own profile details.

1. roles\_list.py :- This file contains “/roles\_list/” and “/get\_the\_roles/”where list of all the roles is fetched

* Also it contains”/update\_roles/” to update the activeness and edit the role names.
* Also it contains “/delete\_roles/” route for a role to be deleted in case role is assigned to the user then it will be updated to anonymous user.
* Also it contains”/add\_roles/” to add another role and its activeness.

1. roles\_permissions.py :-This file contains“/store\_permissions/” and “/get\_permissions/” where different permissions is assigned for different roles and fetched to the frontend.
2. permisisons.py :- This file contains“/add\_permissions/” and “/fetch\_permissions/” where new permissions is added by its title and fetched to the frontend.

* Also it contains “/delete\_permissions/”and “/edit\_permissions/” route for a permissions to be deleted and if needed to edit the title name only

1. admin\_reports.py :- This file contains“/admin\_reports/” where generated reports regarding to all users cameras and other details are fetched and generated as a report.
2. non\_admin\_reports.py :- This file contains“/non\_admin\_reports/” where generated reports regarding to the session user for cameras and other details are fetched and generated as a report