For this week, we have created the authority and user registration pages with backend database connection.

Steps followed to do:

Created the authority.html, authority\_reg.html, user.html, user\_reg.html, index.html, header.html files under template folder in Pycharm.

Created the app.py and DBconnection.py file under CIS folder

In the DBconnection.py file mention password to connect our python code to database.

To do this we need have the module named mysql-connector-python version 8.0.26

Command -> **pip install mysql-connector-python==8.0.26**

Run the DBconnection.py file. It shows output pointing to an object which means connection is successful.

Go to app.py file and run. Open the link in the browser.

Click on Authority button to register as Authority.

In this first we will go to login page if already registered we can give login details if not registered we can click register fill the required details and click submit.

**Code:** to register – authority\_reg.html

To login – authority.html

When registration is successful, check in authority table in database whether details are filled.

Click on Users. Click on REGISTER HERE to register as a user.

It is also same as authority, first we will go to login page if already registered we can give login details if not registered we can click register

Fill the required details and click submit.

**Code:** to register – user\_reg.html

To login – user.html

When registration is successful, check in users table in database whether details are filled.

**Path:** MySQL Workbench -> criminal identification -> authority

-----------------------------------------------------------------------------------------------

When we click on the link, the navigation is going to be in such a way that when we click on Authority, in the web address, /authority will be indicating the redirection of page. this is a url pattern or mapping. With the help of /authority, lets check the code in app.py file.

**All functions which help to redirect to other html pages will be in app.py file.**

**Here is how we navigate:**

if we search for /authority in app.py configuration file, it will directly point here.now as both addresses match, this function is executed.here in again we have called for authority.html file,so lets go into authority.html file.

Here if we click on register button,in the web address we see "authority\_reg" . if we search for /authority\_reg in app.py configuration file,when both the address match authority\_reg ,here this function will be exectuted,here we cal authority\_reg.html. Once we give all details nd at submitting while clicking on the button ,we have to look at the form action to know how they are related.now authority \_reg2 will be invoked.as both addresses match,this function def...will be executed.here we see two methods get and post and we dont use get becoz if we use get all d values passed will be visible in the browser address but for post they won't that's why.

Now if it is a post method,every value written in the form will be retrieved based on the key.here database connection will be established.thedbconnect.get connection will immediately search for dbconnection in this page,here dbconnection class is imported frm dbconnection file.here this function in .py file will be executed.now to execute queries we need to have a object named cursor.If you give the deatils of existing one u will get this msg if new one u will go into else condition and insert query will all the values given by u into the authority table.commit will save the new row recorded.finally this will go to authority.html, message will be returned.

@app.route("/authority") has a function named authority in which we return the authority.html file. This file can be found in templates which has code to display that page.

**Code**: app.py 15-17 line -> authority.html

In authority.html file, we have given reference for registration page 'authority\_reg' which is invoked in @app.route("/authority\_reg"). This calls "authority\_reg.html".

**Code**: app.py 25-28 line -> authority\_reg.html

While filling the form, the form action is on "authority\_reg2" and method is "POST". This is invoked in @app.route("/authority\_reg2") in app.py file. This is used to retrieve values from form and store in database.

**Code**: app.py 30-53 line

POST method is used to hide the details that we type in form to not display in web address.

CURSOR object is used to execute the query.

cursor.execute is used insert values to table.

To save we use **commit()**.

Javascript is used to dispaly messages in HTML pages. We used this to display when user registration is successful, it shows in green color and for invalid it shown in red.

**Code:** For green ( authority.html page 42-46 line)

For red (50-54)