## **Activating Virtual Environment:**

Virtual Environments are very useful because they will help in creating a separate space for your app in the system and you can copy the dependencies information from this space using the *pip freeze* command.

pip install virtualenv [To install virtual environment || First time on the system]

virtualenv env [to create the environment named env]

Set-executionpolicy Unrestricted [Only for people whose powershell command prompt is restricted in **Powershell Admin Mode**]

.\env\Scripts\activate.ps1 [You should be in the new virtual environment after running this command].

## **Installing Flask Module:**

pip install flask

First Code:

**A Minimal Application**

A minimal Flask application looks something like this:

**from** flask **import** Flask

app = Flask**(**\_\_name\_\_**)**

@app.route**(**"/"**)**

**def** hello\_world**():**

**return** "<p>Hello, World!</p>"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

run the above code in **app.py** file

You will get a URL something like this <http://127.0.0.1:5000> in the console. Put it in the web browser it will show Hello World.

## static and templates folder

static folder: Whatever we put in the folder can be accessed with the url path eg: [http://127.0.0.1:5000/static/<filename](http://127.0.0.1:5000/static/%3cfilename)>

eg: <http://127.0.0.1:5000/static/app.txt>

template folder: We can put templates such as index.html in this folder and render them using render\_template

*from* flask *import* Flask, render\_template

app = Flask(\_\_name\_\_)

@app.route("/")

def hello\_world():

*return* render\_template('index.html')

Here in the above code render\_template is used to render the html files. The HTML file shall be inside the template folder

## Connecting the database:

pip install SQLAlchemy