1. Create a folder for flask project and open that folder in the vs code.

2. Open Terminal in vscode and run following commands:

Python -m venv venv

Creates the virtual environment venv.

3. Venv\Scripts\activate.bat

4. Venv\Scripts\Activate.ps1

Allows us to run scripts.

5. pip install flask

Install flask in the created environment.

6. Create a file named app.py

7. Go to <https://flask.palletsprojects.com/en/1.1.x/quickstart/>

8. Copy the minimal app and paste it in app.py

9. Add the following code to the minimal application

if \_\_name\_\_== "\_\_main\_\_":

app.run(debug=True)

To build the python web application, we need to import the Flask module. An object of the Flask class is considered as the WSGI application. Web Server Gateway Interface (WSGI) a universal interface between the web server and the web applications. Here, it is app.

The route() function of the Flask class defines the URL mapping of the associated function.

10. Always keep the name of the created app and file name same.

app1 = Flask(\_\_name\_\_)

will be saved in app1.py

11. Flask App routing: App routing is used to map the specific URL with the associated function that is intended to perform some task. Create a new file app1.py and add following code to the file:

app1.py

run command: python app1.py

It will take number provided in the address bar and assign it to variable age.

11. We can add add\_url\_rule() function for routing.

from flask import Flask

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

def about():

return "This is about page";

app.add\_url\_rule("/about","about",about)

if \_\_name\_\_ =="\_\_main\_\_":

app.run(debug = True)

12. If we want to change the port number on the server, we can change it in the app.run(debug=True, port=8000)

13. The url\_for() function is used to build a URL to the specific function dynamically. This function is useful in the sense that we can avoid hard-coding the URLs into the templates by dynamically building them using this function.

app2.py

run command: python app2.py

14. Flask HTTP methods: POST and GET

Created folders: post\_flask and get\_flask.

Open index.html in browser and see difference.

15. Flask facilitates us to return the response in the form of HTML templates.

Store the template base. html in templates folder. And run app3.py.

16. The jinja2 template engine uses the following delimiters for escaping from HTML.

{% ... %} for Statements

{{ ... }} for Expressions to print to the template output

{# ... #} for Comments not included in the template output

Save table.html in template. Run app4.py.

17. Flask Request Object: In the client-server architecture, the request object contains all the data that is sent from the client to the server.

Templates: customer.html, result\_data.html

Main file: app5.py

18. Flask can make use of the SQLite3 module of the python to create the database web applications. We will create a CRUD (create - read - update - delete) application.

Create database: EmployeeDB.py

Main logic: crud.py

Templates: add.html, index.html, view.html, delete\_record.html, delete.html, success.html

Run command: python EmployeeDB.py

This will create the database.

Run command: python crud.py

This will load index.html and you can perform operations in sqlite.