* FLASK

1.What is flask

2.Web frame work

3.What is wsgi

4.What is jinja2

5.Templet engine

6.Futures of Flask

7.Advantages of Flask

8.Flask is a Frontend or Backend ?

9.Installation process

10.Flask virtual environment

11.Sample program writing

12.Explaining about program

* What is FLASK: Flask is web application framework written in python. Flask is very easy to learn and also its implementation is straight forward.in just a few lines of code .Flask is used in top tech companies also like :Netflix, Reddit
* Web Framework: Python Web framework is **a collection of packages or modules that allow developers to write Web applications or services** Creating a website and writing a code is called web frame work
* Modules: A module can define functions, classes, and variables and also include runnable code.
* Flask is based on WSGI and JINJA2

# WHAT IS WSGI?

WSGI defines Web Server Gateway Interface. It is interface between the web server and the web applications.

# WHAT IS JINJA2?

Jjinja2 is a web template engine which combines a templet with a certain data source to render the dynamic web page

# TEMPLATE ENGINE:

A template engine **enables you to use static template files in your application**. At runtime, the template engine replaces variables in a template file with actual values.

* FEATURES OF FLASK:
* Development server and [debugger](https://en.wikipedia.org/wiki/Debugger)
* Integrated support for [unit testing](https://en.wikipedia.org/wiki/Unit_testing)
* [RESTful](https://en.wikipedia.org/wiki/Representational_state_transfer) request dispatching
* Uses Jinja templating
* Support for secure cookies (client side sessions)
* 100% [WSGI](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) 1.0 compliant
* [Unicode](https://en.wikipedia.org/wiki/Unicode)-based
* Complete documentation
* [Google App Engine](https://en.wikipedia.org/wiki/Google_App_Engine) compatibility
* Extensions available to extend functionality
* ADVANTAGES OF FLASK:
* Scalable. Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. ...
* Flexible
* Easy
* Lightweight.
* Documentation.
* Not a lot of tools.
* Difficult to get familiar with a larger Flask app.
* Maintenance costs.
* Flask a frontend or backend?

Flask is a **back-end** framework, which means that it provides the technologies, tools, and modules that can be used to build the actual functionalities of the web app rather than the design or look of it.

### INSTALLATION PROCESS

* First we have to check that python is their in our device. If not we have to go URL and python download
* Now we should download pip latest version from command prompt

COMMAND: pip install

* We have install pip flask now that mean from URL copy the command and install in command prompt

COMMAND: PIP install flask

Now move to command prompt check it out that the installation was successfully done or not by using some commands

COMMAND: python –version

COMMAND: pip –version

Now we have to create a file in command prompt

COMMAND: mkdir.flask (here flask is a file name )

Now we should check that the file was saved into drive

COMMAND: dir

Now check pip list command

Cd flask

dir (dir means checking that file is exited or not)

* Flask – Virtual Environment(venv)

A virtual environment is a Python environment that the Python interpreter, libraries and scripts installed .Move to command prompt and command

COMMAND: pip install virtualenv

Now we have to dir in command prompt

Now we are moving that venv into cd

COMMAND: virtualenv venv

COMMAND: cd venv

COMMAND: dir

COMMAND: cd script

COMMAND: dir

COMMAND: Activate.bat

Now move to p y charm and write a programme and run

The output should be in URL (That means the code will come that code can search in URL the output can visible to users )

* SAMPLE PROGRAMME WRITING

**from** flask **import** Flask  
app=Flask(\_\_name\_\_)  
  
  
@app.route(**'/'**)  
**def** hello world():  
 **return 'this is python programme'  
if** \_\_name\_\_==**'\_\_main\_\_'**:  
 app. run()

If we want to debug the programme use the command given below

COMMAND: app. run(debug = True)

* EXPLAINING ABOUT PROGRAMME:

‘/’ URL is bound with hello() function. When the home page of web server is opened in browser, the output of this function will be rendered accordingly.

The Flask application is started by calling the run() function. The method should be restarted manually for any change in the code. To overcome this, the debug support is enabled so as to track any error

* DJANGO

1. What is Django
2. Installation process
3. Django working
4. MVT
5. Explaining about files
6. **PROCESS OF EXECUTING A PROGRAMME IN PY CHARM**
7. Difference between flask and Django
8. Conclusion of flask and django

What is Django: It is a back-end side web framework, and it is free, open source are written in Python. Django makes easy to create the web pages from Python.

Latest version of Django is 4.0.3 (March 2022)

* INSTALLATION PROCESS

NOW WE START WITH COMMAND PROMPT

1. We have to move to our local drive d command: d:
2. Then we have create a new folder into a drive-d command: mkdir folder name(school)
3. Now create a virtual environment command: pip install virtualenv
4. Next command: virtualenv venv
5. Cd venv
6. Scripts\activate
7. Now again we should move back to local drive d
8. Here have to install Django to that command is command: pip install Django
9. Now we have to create a project to that command is command: Django-admin startproject projectname(student)
10. Again comeback to cd student
11. Now we have to create a app to that command is command: python manage.py startapp appname(marks)
12. After finishing this process in command prompt let us move to the p y charm project and open the project in p y charm files then open the given project what we created in command prompt
13. In that project we can see the app that what we created in command prompt and in that app we can see migrations, --init--py, admin.py, apps.py, models.py, tests.py, views.py
14. Similarly we can see that project file that what we created in command prompt and in that we see inti.py ,asgi.py ,setting.py ,url.py, wsgi.py, manage.py
15. Django Working: Django works on the MVT design pattern (Model View Template).

* Now let us discuss about the MVT in details
* MVT

What is Model: It provides data from the database.

In Django, the data is delivered as an Object Relational Mapping (ORM), It is a technique designed to make it easy to work with databases.

The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.

Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

The models are usually located in a file called models.py

What is views: A view is a function or method that takes http requests as arguments, imports the relevant model(s)

It finds out what data to send to the template, and returns to the final result.

The views are usually located in a file called views.py

# What is Templates: Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on .html files.

The templates of an application is located in a folder named templates.

* EXPLANING ABOUT FILES

What is URLS: When a user requests a URL, Django decides which view it will send it to.

This is done in a file called urls.py

# What is manage: A Manage is **the interface through which database query operations are provided to Django models**.

# At least one Manage exists for every model in a Django application.

# The Manage classes work is document

# This document specifically touches on model options that depend on Manage

This is done in p y charm terminal command: python \manage.py

What is admin : The Django admin is an automatically-generated user interface for Django models.

The register function is **used to add models to the Django admin so that data for those models can be created, deleted, updated and queried through the user interface**.

What is settings: In settings we have to add the app name that what we accessed in command prompt

* **PROCESS OF EXECUTING A PROGRAMME IN PY CHARM**

1. We have to open p y charm after creating a new project and new app then now click on files and open the file which you created and here we see the both app and project and in that we can see the files
2. Next open the setting files and add the file name with respected strings and comma in the last
3. Now move to models.py file and first we have to check that the python version and check that the draft one should be clear
4. Then start any class program while writing a class function we use the model logic which is model .Model
5. Next open terminal in p y charm and now move to shell because we are connecting the database to python
6. Before going to shell we move to d drive
7. Then now move to shell command: py.\manage.py shell
8. Now we moved to shell next command: from django.db import connection
9. Next command: c= connection .cursor()
10. Now we have to exit from shell command: exit()
11. Now make migration command: py .\manage.py makemigrations here python file is changing into sql from app
12. Now we should create table so we should do migrate
13. Command: py .\manage.py migrate
14. Now to store the data we have to create the superuser command: py .\manage.py createsuperuser
15. Now create any name
16. Email Address if u want give then skip
17. Create password
18. Re-enter the password
19. Now you created superuser successfully
20. After finishing it we have to run server command: py .\manage.py runserver
21. Finally we get the url code in output click on that url and type admin and enter username, password then one page will open in that we cant see any app or projectname
22. Now to show app and project move to p y charm and move to admin.py, we have to do from app. app name import class beside what we represent that we take here logic is : from app. app name import name
23. To see we have to register in that logic is :admin.site.register(class define name)
24. Run the server ctrl S
25. Now open the link we see the app name
26. Now add the objects and save
27. Here objects are created
28. But I want to see the details so again move to models.py in p y charm
29. Now define function it is called the dunder method
30. Logic is :def –str—(self):
31. Then return self.name (what the information given in class that should be taken )
32. We con not get all full details if we need full details move to admin.py and use class function that is classempolyerAdmin(admin.ModelAdmin):
33. List \_display =[what we want to display that we can give]
34. Now register here again
35. Admin.site.register(empolye,employeAdmin)
36. Then run the server
37. Then we get all the details in the rows and columns

* DIFFERENCES BETWEEN FLASK AND DJANGO

|  |  |
| --- | --- |
| FLASK | DJANGO |
| It is a lightweight framework that is commonly referred to as a micro framework. | Django is a web application framework that handles many standard features needed to create secure and maintained websites. |
| Flask comes with a small collection of easy-to-learn comprehensive documentation. | It’s a flexible framework that can be used to build any website (social network, news site, content management system, and so on) with the content in HTML, XML, JSON, and other formats. It can be used in tandem with any client-side framework. |
| Routing URLs is simple. | End-to-end application testing is possible with Django. |
| The code base size is relatively smaller. | Allow you to set patterns for your application’s URLs. |
| It is easy to use for the simple case | Framework for quick web development at a high level. |

CONCULUSION OF FASK AND DJANGO:

Although Django and Flask share many fundamental concepts, Django is more complex and large , requiring a steep learning curve.

Django requires more than twice as many lines of code compared to Flask.

Django is a production-ready framework.

Each project can be a single application with numerous models and views, while the single application in flask.