# 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

ϖ ## 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 ϖ## 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 Why flask is used: Flask gives the developer varieties of choice when developing web applications, it provides you with tools, libraries, and mechanics that allow you to build a web application •## 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 • Integrated support for unit testing • RESTful request dispatching • Uses Jinja templating • Support for secure cookies (client side sessions) • 100% WSGI 1.0 compliant • Unicode-based • Complete documentation • 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) ϖ INSTALLATION PROCESS NOW WE START WITH COMMAND PROMPT
9. We have to move to our local drive d command: d:
10. Then we have create a new folder into a drive-d command: mkdir folder name(school)
11. Now create a virtual environment command: pip install virtualenv
12. Next command: virtualenv venv
13. Cd venv
14. Scripts\activate
15. Now again we should move back to local drive d
16. Here have to install Django to that command is command: pip install Django
17. Now we have to create a project to that command is command: Django-admin startproject projectname(student)
18. Again comeback to cd student
19. Now we have to create a app to that command is command: python manage.py startapp appname(marks)
20. 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
21. 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
22. 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
23. Django Working: Django works on the MVT design pattern (Model View Template). • Now let us discuss about the MVT in details ϖ 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. ϖ 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
24. 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
25. Next open the setting files and add the file name with respected strings and comma in the last
26. 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
27. Then start any class program while writing a class function we use the model logic which is model .Model
28. Next open terminal in p y charm and now move to shell because we are connecting the database to python
29. Before going to shell we move to d drive
30. Then now move to shell command: py.\manage.py shell
31. Now we moved to shell next command: from django.db import connection
32. Next command: c= connection .cursor()
33. Now we have to exit from shell command: exit()
34. Now make migration command: py .\manage.py makemigrations here python file is changing into sql from app
35. Now we should create table so we should do migrate
36. Command: py .\manage.py migrate
37. Now to store the data we have to create the superuser command: py .\manage.py createsuperuser
38. Now create any name
39. Email Address if u want give then skip
40. Create password
41. Re-enter the password
42. Now you created superuser successfully
43. After finishing it we have to run server command: py .\manage.py runserver
44. 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
45. 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
46. To see we have to register in that logic is :admin.site.register(class define name)
47. Run the server ctrl S
48. Now open the link we see the app name
49. Now add the objects and save
50. Here objects are created
51. But I want to see the details so again move to models.py in p y charm
52. Now define function it is called the dunder method
53. Logic is :def –str—(self):
54. Then return self.name (what the information given in class that should be taken )
55. 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):
56. List \_display =[what we want to display that we can give]
57. Now register here again
58. Admin.site.register(empolye,employeAdmin)
59. Then run the server
60. Then we get all the details in the rows and columns  
    ϖ 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. ϖ PANDAS

1. What is pandas
2. What is panel Data
3. What is Data analysis
4. Why we use pandas
5. What is data frames
6. What panda can do
7. Installation
8. Sample program writing
9. Explaining about program
10. Series in pandas
11. What is series
12. Sample program
13. Series with label
14. Sample program
15. Pandas in series of keys/values
16. Sample program
17. Features of pandas
18. Advantages of pandas
19. Dis advantages of pandas
20. Conclusion of pandas

What is pandas: Pandas is a Python library. Pandas is used to analyse data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. What is panel Data: Panel data, sometimes referred to as longitudinal data, is data that contains observations about different cross sections across time. Examples of groups that may make up panel data series include countries, firms, individuals, or demographic groups. What is Data analysis: Data Analysis is the technique to collect, transform, and organize data to make future predictions, and make informed data-driven decisions Why we use pandas: Pandas is mainly used for data analysis and associated manipulation of tabular data in Data Frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. What is data frames: A Pandas Data Frame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns. What pandas can do: Pandas gives you answers about the data. Like: • Is there a correlation between two or more columns? • What is average value? • Max value? • Min value? Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called cleaning the data. ϖ Installation Now first we have to create a folder Command: mkdir folder name Then now cd folder name Next create a virtual environment Command: pip install virtualenv Next command: virtualenv venv Next command : scripts\activate Lets install pandas To that command :pip install pandas Now move to p y charm and open files and open the folder that what we created in command prompt And execute a programme as shown in below SAMPLE PROGRAMME WRITTING

import pandas as pd

data = { "calories": [420, 380, 390], "duration": [50, 40, 45] }

df = pd.DataFrame(data, index = ["day1", "day2", "day3"])

print(df) HERE the output is below Output: calories duration day1 420 50 day2 380 40 day3 390 45 EXPLAINING ABOUT PROGRAM: Pandas is usually imported under the pd alias. Create an alias with the as keyword while importing: Here alias means Refer to same thing Series in pandas What is series: A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any type. Sample program: import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a)

print(myvar) output:  
0 1 1 7 2 2

Series with labels: The values are label with their index number and First value has index 0, second value has index 1 etc. This label can be used to access a specified value. With the index argument, you can name your own labels. Sample program: import pandas as pd

a = [1, 7, 2]

my-var = pd.Series(a, index = ["x", "y", "z"])

print(my-var) output: x 11 1 y 7 z 2 Pandas in Series of key/values: Now key/value object, like a dictionary, when creating a Series. Sample program: import pandas as pd

calories = {"day1": 420, "day2": 380, "day3": 390}

my-var = pd.Series(calories)

print(my-var) output: day1 420 day2 380 day3 390 FEATURES OF PANDAS:

1. Handling of data.
2. Alignment and indexing.
3. Handling missing data.
4. Cleaning up data.
5. Input and output tools.
6. Multiple file formats supported.
7. Merging and joining of datasets.
8. A lot of time series.

ADVANTAGES OF PANDAS:

• It was Data representation. Pandas provide extremely streamlined forms of data representation. • It is Less writing and more work done. • An extensive set of features. • It is Efficiently handles large data. • And Makes data flexible and customizable

DISADVANTAGES OF PANDAS:

• Steep learning curve. Pandas initially have a mild learning slope. • It is Difficult syntax. While, being a part of Python, Pandas can become really tedious with respect to syntax. • Poor compatibility for 3D matrices. • Bad documentation.

CONCLUSION OF PANDAS:

The Pandas library is really an amazing tool to have in Python. This article just goes over the tip of the iceberg as to what you can accomplish with the Pandas You can begin to see the true capabilities that Pandas has to offer when starting to work with data in Python. Pandas introduced two new types of objects for storing data that make It was analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and Data Frames, which have a tabular structure. Pandas serves as one of the pillar libraries of any data science workflow It allows you to perform processing, wrangling and munging of data. Pandas provide a huge feature set to apply on the data you have so that you can customize, edit and pivot it according to your own will and desire. This helps to bring the most out of your data. We can use Pandas to perform various tasks like filtering your data according to certain conditions, or segmenting and segregating the data according to preference, etc. Using Pandas helps to shorten the procedure of handling data. With the time saved, we can focus more on data analysis algorithms.