Flask

Flask is a web framework written in python . Flask is a micro web application framework . Flask is a small and lightweight python web framework that provides useful tools and features that make creating web applications in python easier . Flask developed by Armin Ronacher, who lead team of international python group on python enthusiasts called POCCO projects.Flask does not support dynamic HTML pages .

Flask is used on WSGI toolkit and jinja2 template engine.

Flask is a micro framework that doesnot include an ORM.

* ORM- Object Relational Manager .

Flask uses the jinja2 template engine . 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 .

Web framework : a web application framework or a simply a web framework represents a collection of libraries and modules that allow developers to write web applications .

Modules : a python modules is a file containing python definitions and statements . A modules can define functions classes, and variables .

Libraries : a python library is simply a collection of codes or modules of codes that we can use in a program for specific operations .

Template engine : 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, and transforms the template into an HTML file sent to the client . This approach makes it easier to design an HTML page .

What is WSGI ?

WSGI – WEB SERVER GATEWAY INTERFACE

WSGI is an interface between web server and web apps for python.

Mod \_ wsgi is an Apache HTTP server module that enable Apache to server Flask application .

What is jinja2 ?

Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages .

COMPANIES USING FLASK :

# Netflix .

# MIT .

# Airbnb .

# Reddit .

# Lyft .

# Zillow .

# Mozilla .

# MailGui, etc .

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&#39;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 installation process :

first we should install python . Python 2.6 or higher is usually required for installation of flask.Recommended that flask should be installed on python 2.7.we use command prompt for flask installation.

install virtualenv for development environment :

virtualenv is a virtual python environment builder.it use to create multiple python environments side by side .

The following command use for to create a virtual environment .

Pip install virtualenv

once install virtual environment . Then create a folder for virtualenv

mkdir newfolder

then push created new folder move to drive

cd newfolder

virtual environment push to folder

virtualenv venv

now activate scripts for folder data move in to virtual environment

venv\ script\ activate

now ready to install flask in this environment

pip install flask

now the installation successfully done check that version for command

pip \_\_version

now we have to create file in command prompt

mkdir.flask

(here flask is a file name)

Now we check that file was save or not for use this command in command prompt

dir

Now pip list command

Cd flask

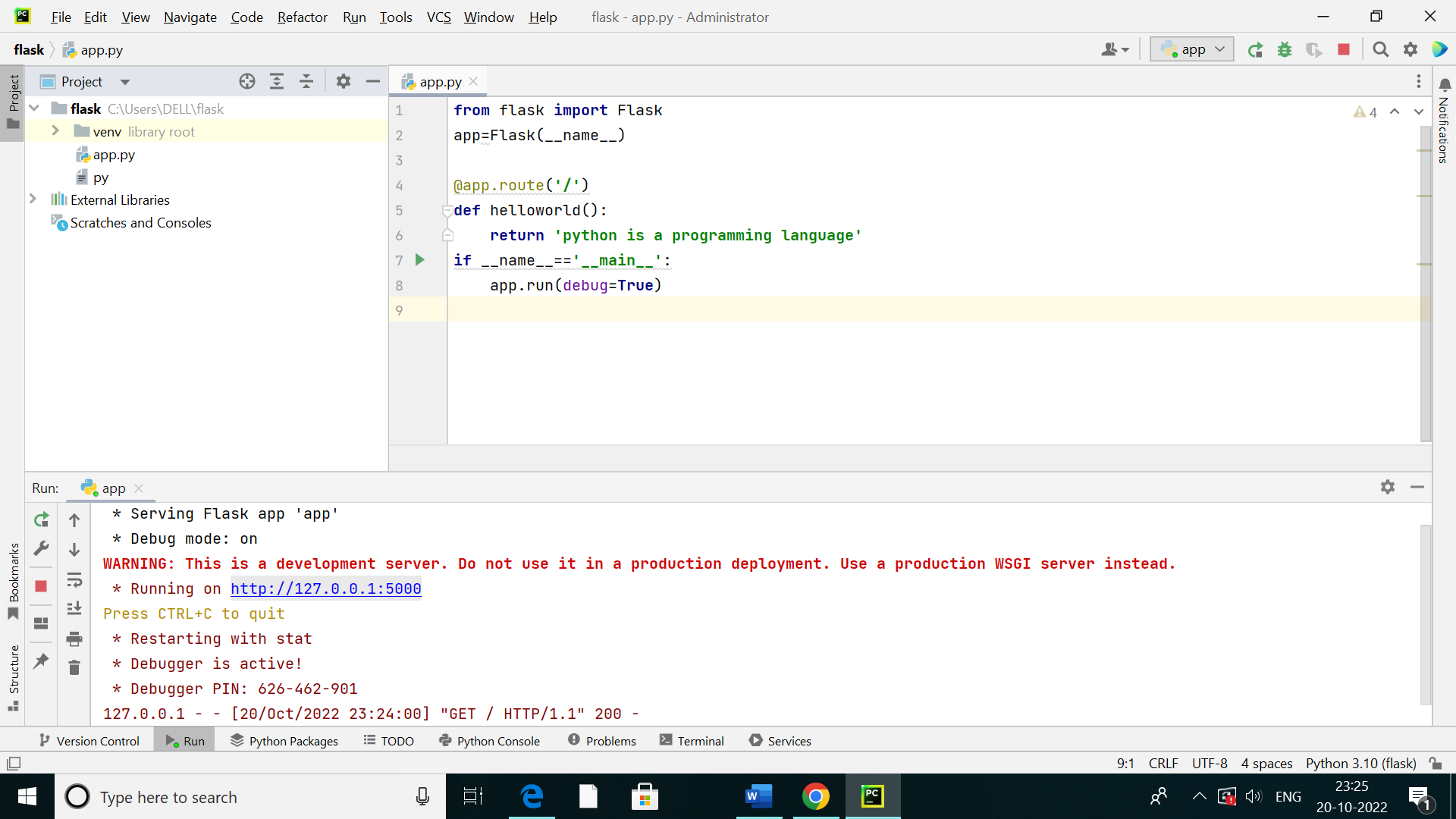
dir

(dir means checking tht file is exited or not)

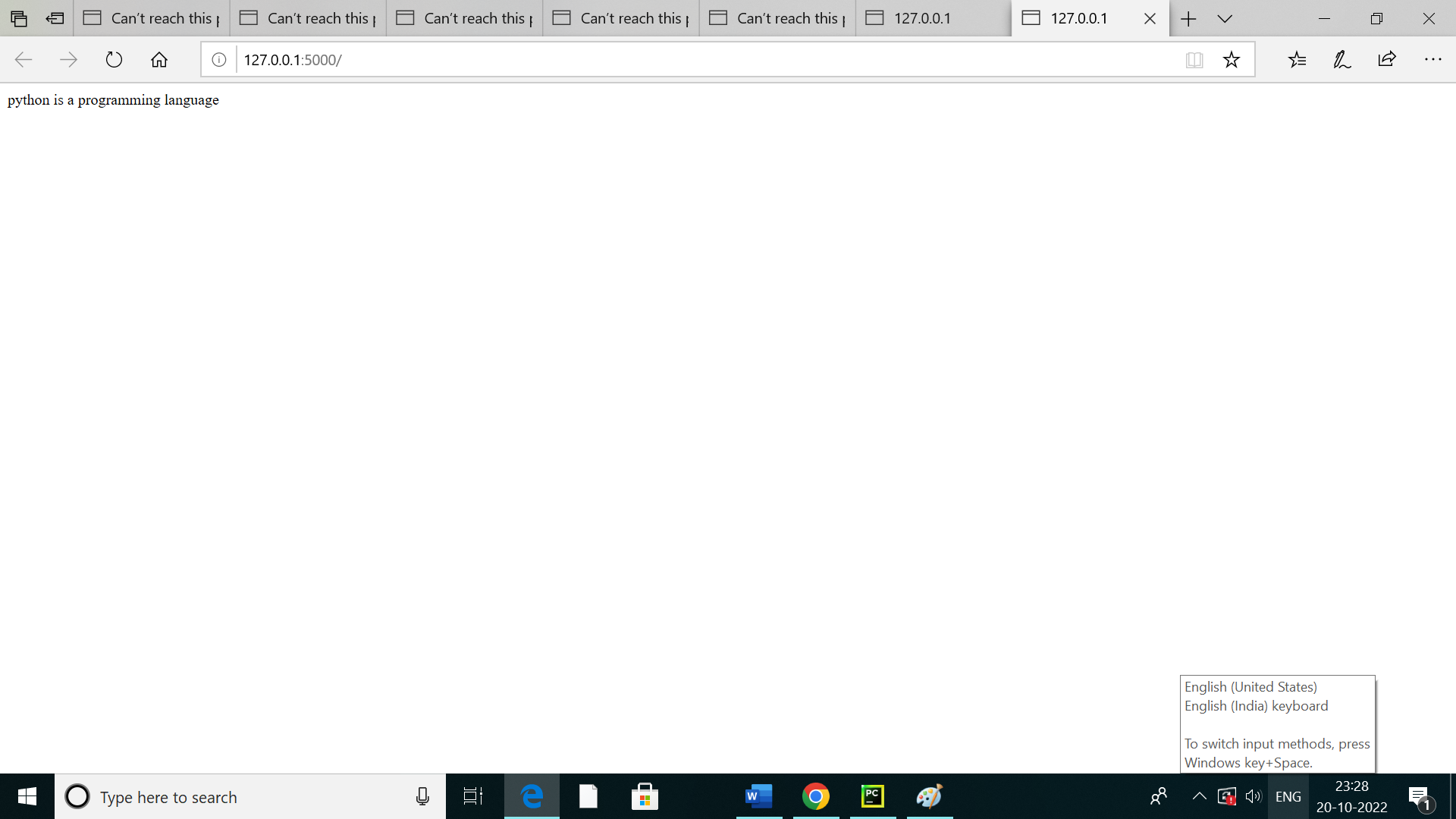
Sample programme :

**from** flask **import** Flask  
app=Flask(\_\_name\_\_)  
  
@app.route(**'/'**)  
**def** helloworld():  
 **return 'python is a programming language'  
if** \_\_name\_\_==**'\_\_main\_\_'**:  
 app.run(debug=**True**)

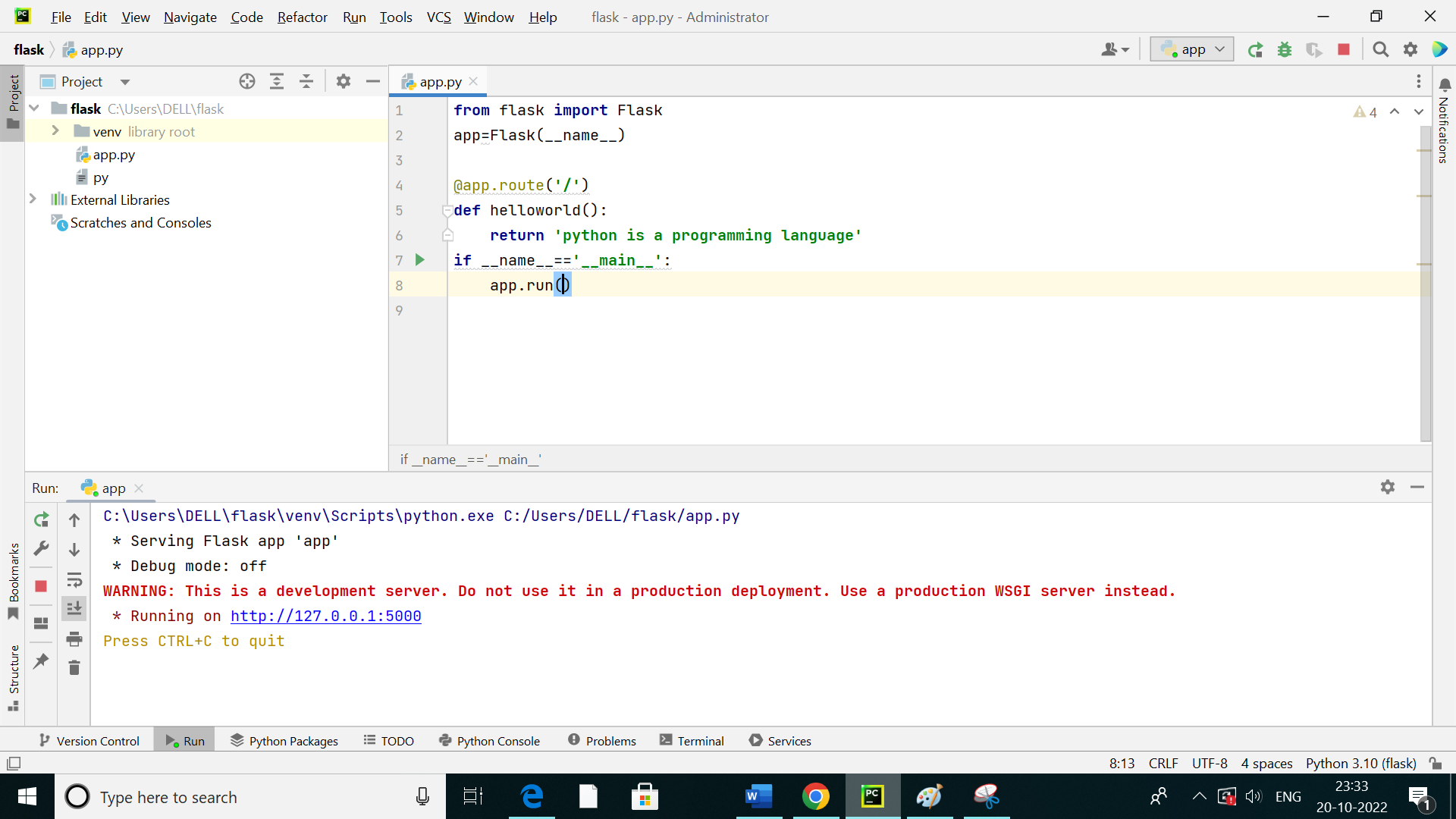
this programme when debug mode on , it can be refresh on Url code no need to run programme .



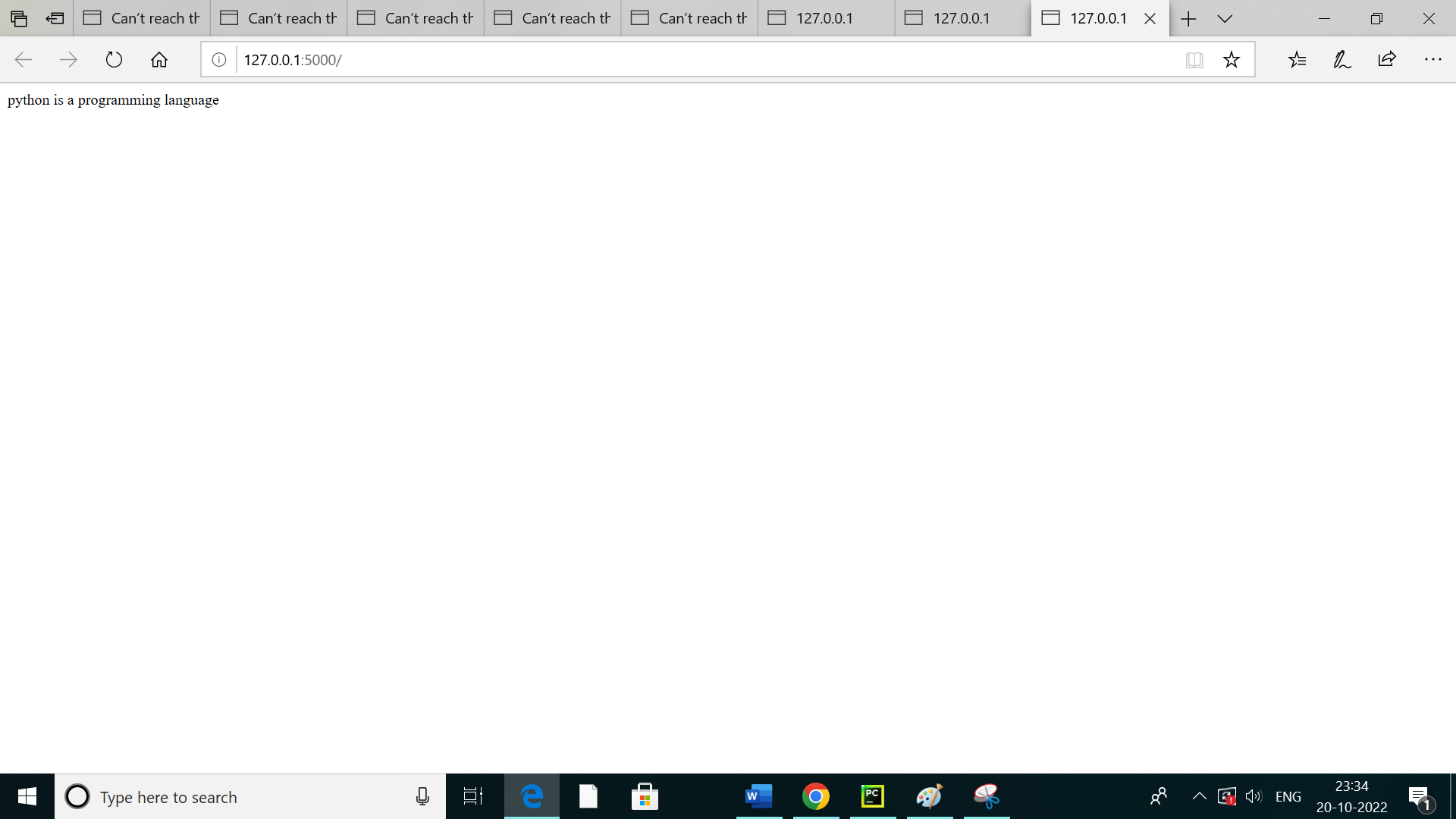
After run , this



Here debug mode off , this programme will run again



After run debug off mode



Flask debug mode allows developers to locate any possible error and as well the location of the error, by logon a traceback of the error

‘/’ 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

WHAT IS DJANGO?

Django is a free and open source web application framework written in python . Django is popular web framework . it is a high level web framework Django is a backend server side web framework . its like most modern framework .

Django is a python framework that makes it easier to create web sites using python . Django is especially helpful for database driven websites . Django follows the MVT design pattern (Model View Template) . Django offers dynamic HTML pages .

MVT :

Django , a Python framework to create web applications , is based on Model – View - Template (MVT) architecture .  is a software design pattern for developing a web application . It consists of the following three entities :

1.Model :

2.View :

3.Template:

1 . Model :

A  is an object that defines the structure of the data in the Django application .

It is responsible for maintaining the entire application’s data for which it provides various mechanisms to add , update , read and delete the data in the database .

2 . view :

A  view is a handler function that accepts HTTP requests, processes them , and returns the HTTP response .

it retrieves the necessary data to fulfill the request using Models and renders them on the user interface using Templates .

It can also create an HTML page using an HTML template dynamically, and populate it with data fetched from the model

3 . Template :

A Template  is a text file that defines the structure or layout of the user interface . The text file can be any type of file ; for example HTML, XML, etc .

It can accept data from the view and render it using  jinja syntax .

Companies using Django :

# Instagram .

# Coursera .

# Mozilla .

# Printerest .

# National Geographic .

# Spotify .

# Udemy .

# Zapier,etc .

Advantages of Django

Here are few advantages of using Django which can be listed out here −

**Object-Relational Mapping (ORM) \_\_**  Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

**Multilingual \_\_** Django supports multilingual websites through its built-in internationalization system . So you can develop your website, which would support multiple languages .

F**ramework** \_\_ Django has built-in support for Ajax, RSS, Caching and various other frameworks .

**Administration GUI** − Django provides a nice ready-to-use user interface for administrative activities .

**Development Environment** − Django comes with a lightweight web server to facilitate end-to-end application development and testing .

Django development environment consists of installing and setting up Python, Django, and a Database System . Since Django deals with web application, it's worth mentioning that you would need a web server setup as well .

Disadvantages of Django-based systems :

# # lower flexibility .

# # lower compatibility with latest technologies .

# # monolithism .

# # higher entry point for simple solutions .

# # larger size of the code base .

Django installation:

Installing Django is very easy ,but the steps required for its installation depends on your operating system.first we creat a virtual environment first,so then we go to command prompt .

In command prompt first go to drive use this command C: or D:

Use D drive are c drive .

And then create virtual environment .

Virtualenv venv

venv is virtual environment folder name ,

Then created folder move to the drive .

For use this command . . . . . .

cd venv

dir use this command for folder create or not .

now create scripts , this is for to activate virtual environment .

use this commad for activate script . .

scripts\activate

then back to device for use this command . . . . . .

cd..

And then install Django for use this command . . . . .

Pip install Django

Then create a project use this command . . . . .

django-admin startproject any name

now created a project than move to the drive .

cd any name

now create room for to send project to client send easily with this room use command . . . . .

py .manage.py startapp any name

now go to pycharm in go to file open,then opened created file

Django created new project display like this . . . . .

New project:

\_init\_.py

asgi.py

Settings.py

urls.py

wsgi.py

Manage.py

now go to settings add room name and then go to models

This is done in a file called urls.py

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

What is manage : A Manage is the interface through which database

query operations are provided to Django models.

What is settings : In settings we have to add the app name that what

we accessed in command prompt

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 .

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 .

Next open the setting files and add the file name with respected

strings and comma in the last now move to models.py file and first we write the class function,

And then go to terminal to create a shell, we are connecting the database to python. Before going to shell we move to d drive .

Py .\manage.py shell

Then import connection for use this command . . . . .

From Django.db import connection

Next command is

C=connection cursor()

Now exit command is

Exit()

Now do makemigration , here python file is changing into sql from app

Py .\manage.py makemigrations

we should do migrate we use command

py .\manage.py migrate

Now to store the data we have to create the superuser use this command

py .\manage.py createsuperuser

Now create any name

Email Address if u want give then skip

Create password, give any words or names

Re-enter the password, give same words or names

Now you created superuser successfully

After finishing it we have to run server ,we use to this commands . .

py .\manage.py runserver

after we get code ,click on that code open in url .

we add admin in url code after

then we go to pycharm open in 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 ,To see we have to register in that logic is:admin.site.register(class define name) after run server we get url code open that code in url

add admin now open the link we see the app name now add the objects and save here objects are created but I want to see the details so again move to models.py in pycharm.

Now define function it is called the dunder method

Logic is :def \_\_str\_\_(self):

Then return self.name (what the information given in class that

should be taken ) 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):

List \_display =[what we want to display that we can give]

Now register here again

Admin.site.register(empolye,employeAdmin)

Then run the server

Then we get all the details in the rows and columns.this is the Django.

DEFERENCE BETWEEN DJANGO AND FLASK :

DJANGO FLASK

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 . Django is a full – stack web framework 1 . Flask is a light weight frame work that  That enable ready to use solutions with gives abundant features without  Its batteries – included approach external libraries and minimalist . .features .   |  |  | | --- | --- | |  |  |   2 . Django is suitable for multiple page 2 . Flask is suitable for only single - page  applications . applications .   |  |  | | --- | --- | |  |  |   3 . The working style of Django is 3 . The working style of flask is  Monolithic . diversified style .   |  |  | | --- | --- | |  |  |      |  |  | | --- | --- | |  |  |   4 . Django does not support any virtual 4 . Flask has an in – built debugger that offers  Debugging . virtual debugging .   |  |  | | --- | --- | |  |  |   5 . Django framework supports the 5 .5 Flask web frame wrok allows  Mapping of URL to views through a URL to class – based view with werkzeug.  request .   |  |  | | --- | --- | |  |  |   6 . Django framework structure is more 6 . Flask web frame work structure is  conventional . random .   |  |  | | --- | --- | |  |  |   7 . Django supports dynamic HTML pages. 7 . Flask framework does not supports  Dynamic HTML .   |  |  | | --- | --- | |  |  |   8 . Best features 8 . Best features   1. Open – source . (i) Extensive documentation . 2. Great community . (ii) lightweight . 3. Fast development . (iii) minimal features . 4. Easy to learn (iv) Full control over the development .   process .   1. secure . (v) open – source .  |  |  | | --- | --- | |  |  |   9 . Django is suitable for high – end 9 . Flask is suitable for companies and projects  Technology companies like that want experimentation with the  Instagram , Udemy , Coursera , etc . module , and architecture of the  Framework like Netflix , Reddit ,Airbnb ,  Etc .  Conclusion Django vs flask :  Django vs Flask : one is an open - source framework for rapid development while the latter is a light-end framework for standard functionalities . Django and Flask are types of frameworks written in the Python programming language . These python-based frameworks are considered to be one of the popular frameworks for web development, according to the Developers Survey 2018 .  After reading and understanding the in-depth detail about both of the web frameworks, one must easily conclude that both have their functionalities . It means that there must be a reason why both are amongst the popular python-based frameworks in the domain of web development .  Flask renders full control and is highly suitable for small projects that necessitate experimentation .  Django is complicated and requires vast knowledge but it stands out as one of the best frameworks for building sophisticated applications . |  |