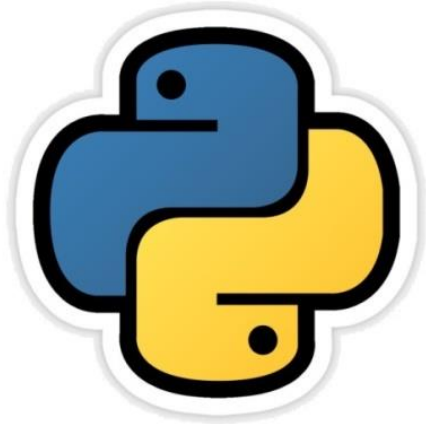


Django Based Web Application

Part - I

CBSE Syllabus Based

Class -12



CHAPTER -15 CS

CHAPTER-9 IP

By-

Neha Tyagi

PGT CS

KV 5 Jaipur II Shift

Blog- www.pythontrends.wordpress.com

YouTube Channel : “Python Trends”

E-mail: python.kvs@gmail.com

Introduction

- Now a days, Internet has invaded in all spheres of our lives whether directly or indirectly.
- Websites have own separate roll on internet.
- Many frameworks are available today for developing efficient websites.
- Django is one such framework that is used along with Python to develop dynamic websites.
- Django framework is used to develop Dynamic websites with python.
- In this chapter, we will learn about django framework.



What is Web framework?

- Web framework is a software tool which supports web development.
- In other words, a web Framework is a software tool that provides a way to build and run dynamic websites and web-enabled applications.

It provides four things-

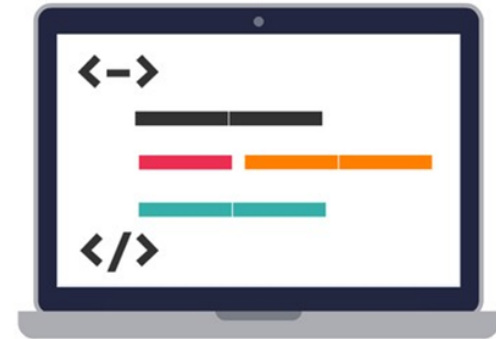
- URL Mapping
- Database Manipulation
- Templating
- Security Measures

• Some other web framework are-
ZEND, Flask, Ruby, Perl etc.

• A web framework will do majority of work and utilize your code in it and a library's function can be utilized in an application being developed by you.



django



What is Django?

- Django is pronounced as 'Jango', 'D' remains silent.
- This is a high level Python web framework which speeds up development of a website.
- This is a free and open source web application framework.
- For web development, it provides existing components.
- It prevents repetition of work.
- The most amazing idea of this is reusability.
- Django is a trademark of Django Software Foundation.
- It comes under BSD (Berkeley Source Distribution).

Advantages of Django

- Object Relational Mapping (ORM) Support
- Supports No-SQL Database.
- Support to various languages.
- Support of Framework.
- Administration GUI
- Development Environment
- Loosely Coupled
- Less Coding
- Don't Repeat Yourself (DRY)
- Scalable
- Secured

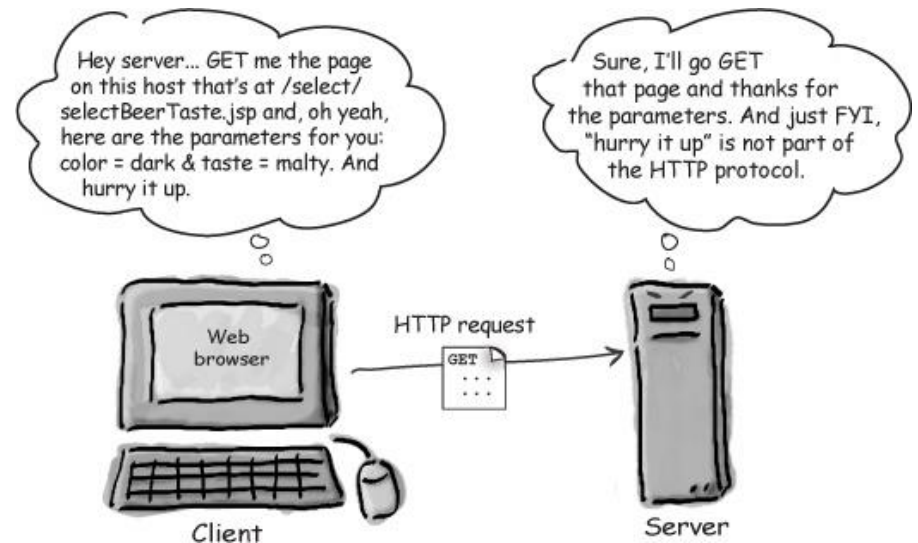


How a Website works?

- As we know that a website works on client-server architecture.
- Your browser acts as a client program , and the web server with which it interacts is the server.

HTTP Get Request: Whenever the web client has to display a webpage, it makes a GET request and sends the URL of the webpage. The server responds by sending the HTML of the URL, if available. If no such URL exists, it returns an error (404).

An HTTP GET request refers to a way of retrieving information from a web server using a given URL over web.



How a Website works?

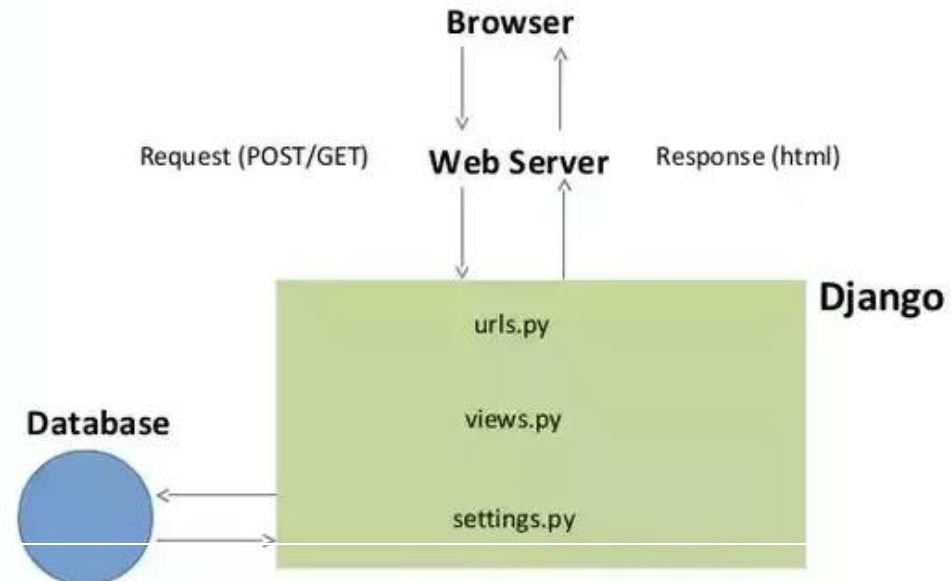
HTTP POST Request:

Whenever a web client has to send some data, this data will be sent to the web server for storing in a database through a POST request.

The HTTP response to a POST request is also either an error code, if not executed successfully.

How Django works?

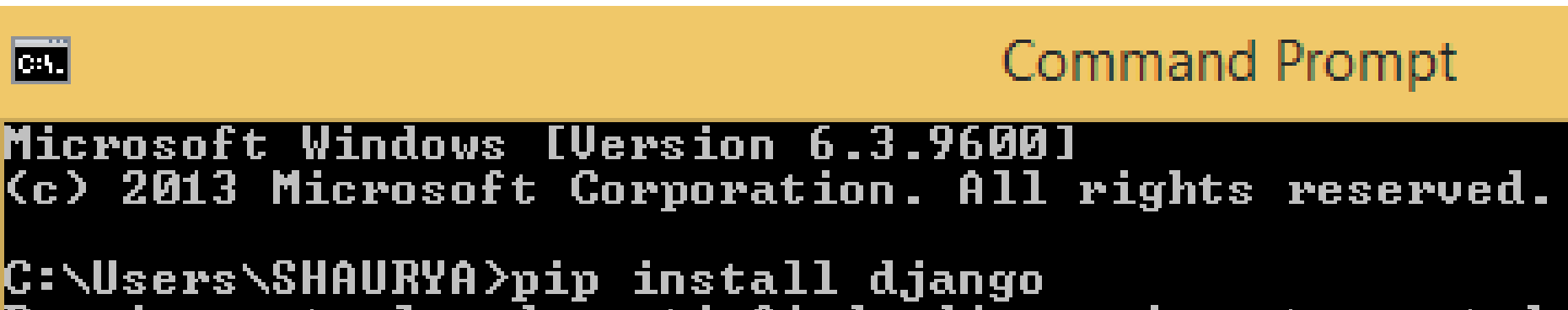
1. It supports MVT or MTV architecture (Model Template View)
2. Request/Response System: Django has the software components which facilitates receiving and responding of web requests.
3. Web Request enters in django applications via **URLs**.
4. Request are processed by **views**.
5. And then web response returns.



Installation of Django

Run the following command on DOS prompt to install Django -

```
C:\> pip install django
```

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text:

```
C:\>  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
C:\Users\SHAURYA>pip install django
```

*Internet is needed while installing Django.

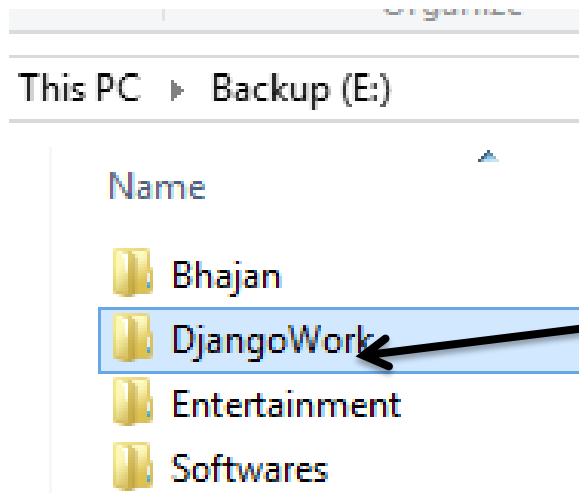
Django Project and App

Two words are frequently used in Django-Project and app

- A project refers to an entire application.
- An app is a sub module catering to one part of the project.
- For ex- if you want to develop a project on school then its submodule or app may be-
 - Student
 - Teachers
 - Exam
 - Fee etc

Creating Django Project and App

- Before starting, create a folder at the location of your choice to store all Django projects.



Now, all the django projects will be saved in this folder.

- After this, use the command to enter in folder using DOS command.

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

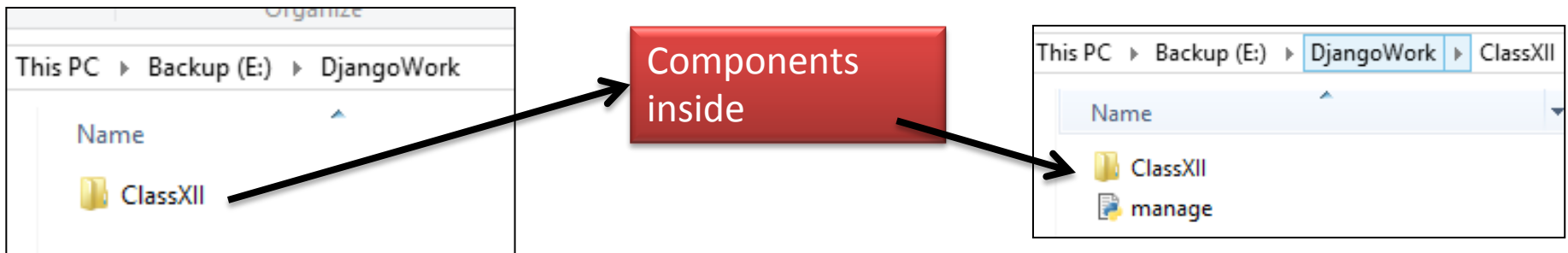
C:\Users\SHAURYA>e:
E:\>cd DjangoWork
E:\DjangoWork>
```

Creating Project in Django

- now use this command to create project.

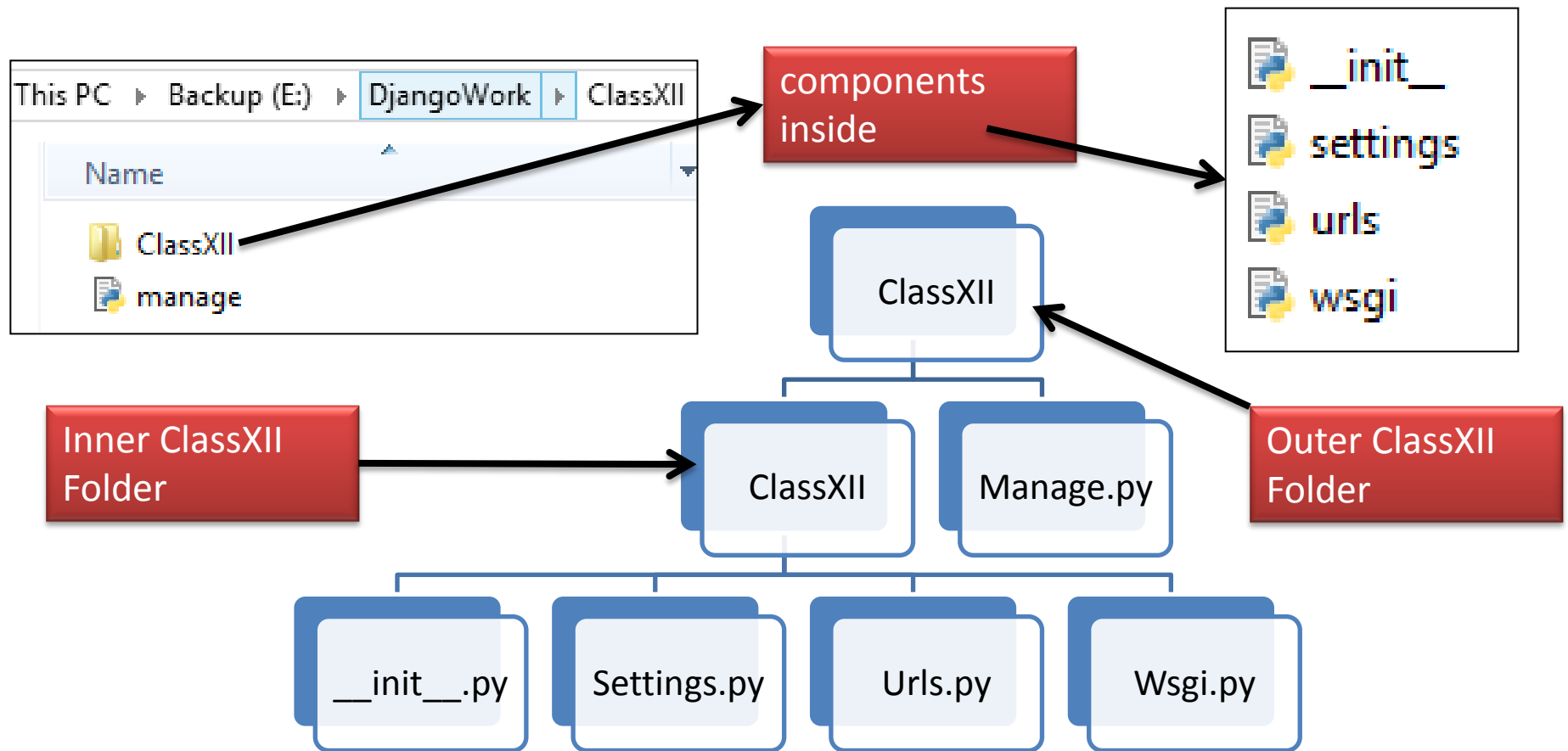
```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\SHAURYA>e:
E:\>cd DjangoWork
E:\DjangoWork>django-admin startproject ClassXII
E:\DjangoWork>
```

- A folder named ClassXII will be created inside DjangoWork which will have following components-



Creating Project in Django

- Now you can see an outer folder ClassXII and an inner folder ClassXII containing a file following components are inside the inner folder-



Running Django Server

- Now we will check whether Django server is working properly or not. For this, following command needs to run after entering project folder -

```
E:\DjangoWork>cd classXII
E:\DjangoWork\ClassXII>manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

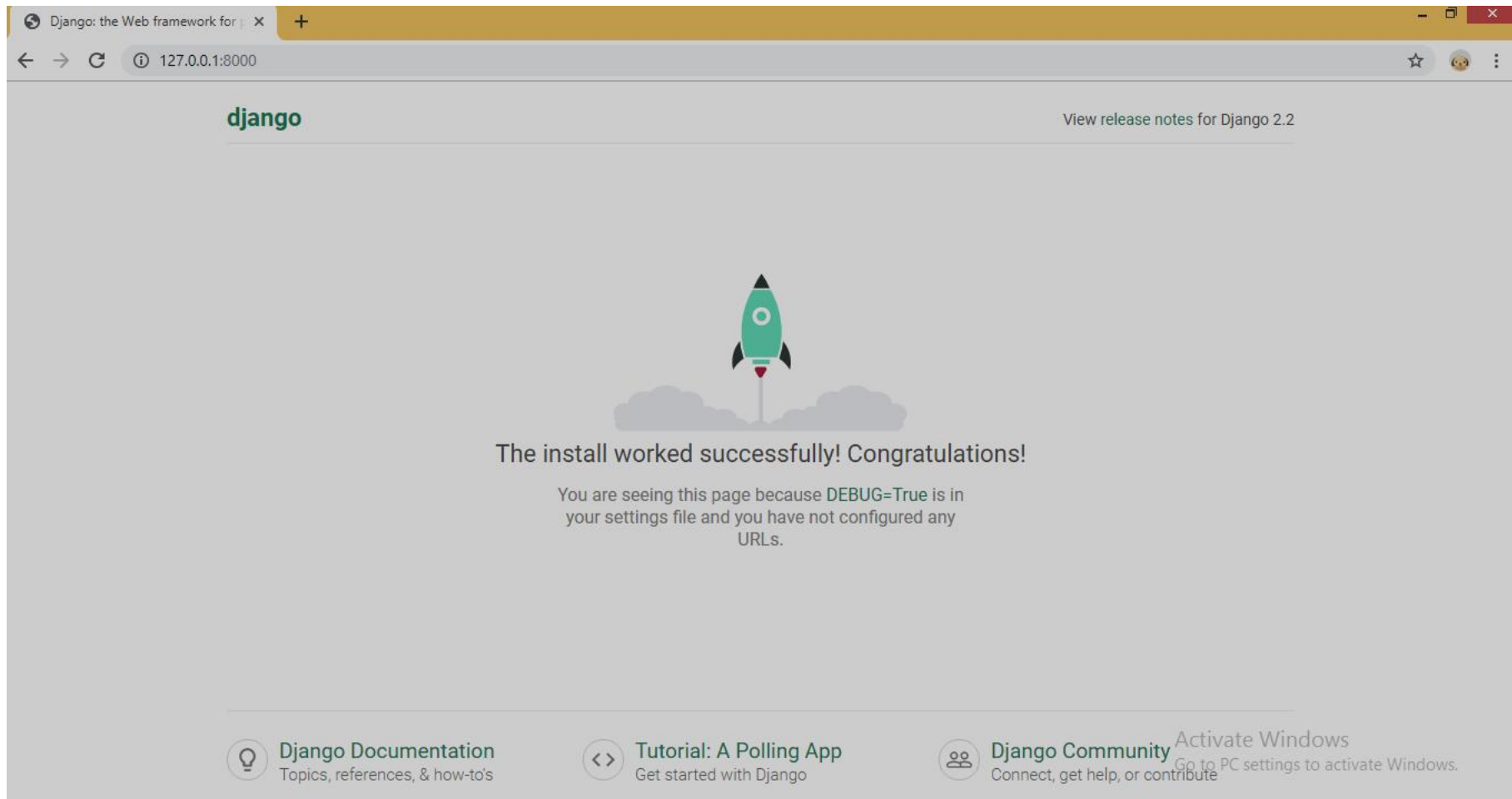
You have 17 unapplied migration(s). Your project may not
  apply the migrations for app(s): admin, auth, contentty
Run 'python manage.py migrate' to apply them.
July 29, 2019 - 00:50:05
Django version 2.2.3, using settings 'ClassXII settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

URL need to run in web browser

- Runserver needs to run from the same place as of manage.py.

Running Django Server

- Now when you open the URL which you got (127.0.0.1:8000) in web browser, if following screen will appear then it is working properly otherwise there is some problem.



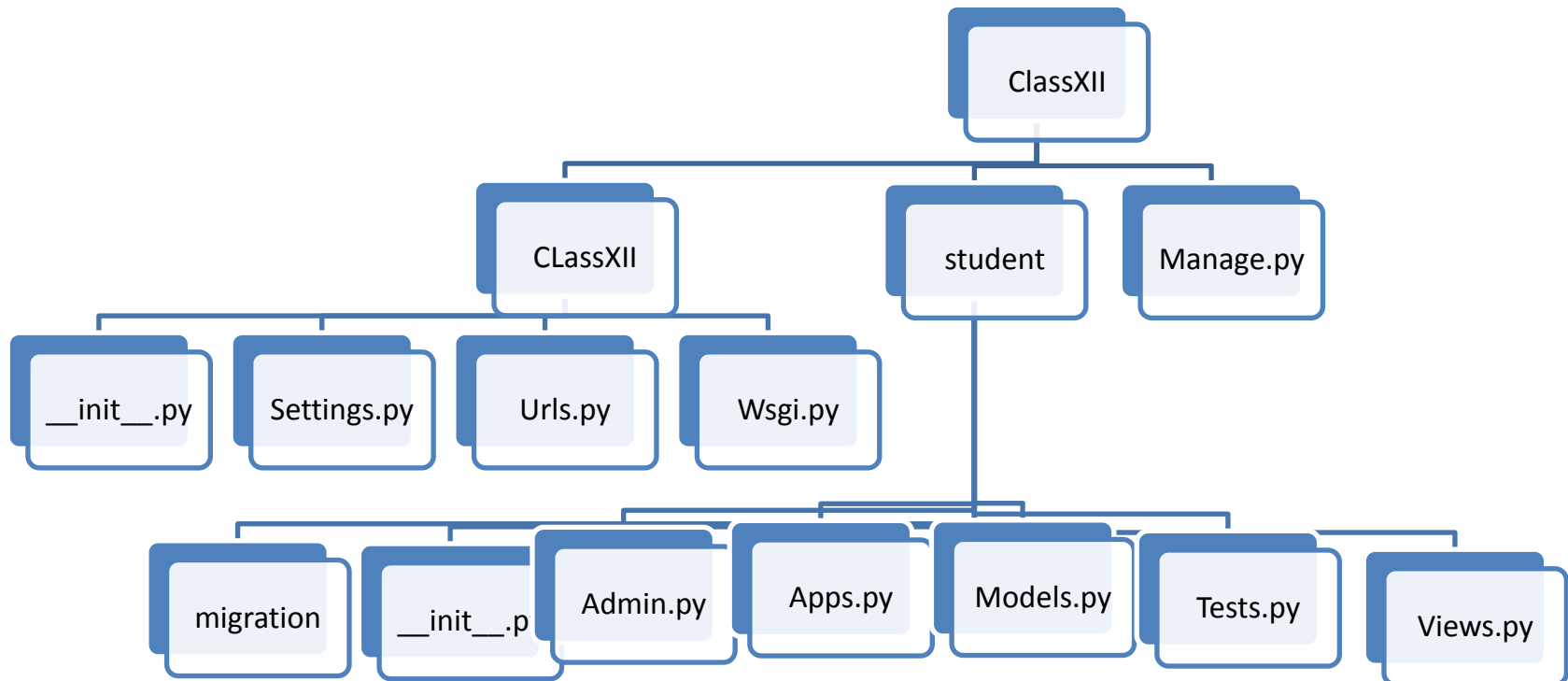
Making an App in Django

It is very important to create an app in a project. For this, manage.py will be used.

Lets create an app in ClassXII. Syntax will be-

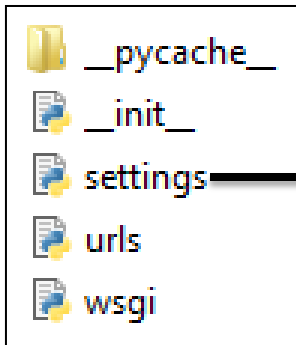
E:\DjangoWork/ClassXII\manage.py start app student

After this, a new folder with the name Student will be added to outer class and structure will be like as-

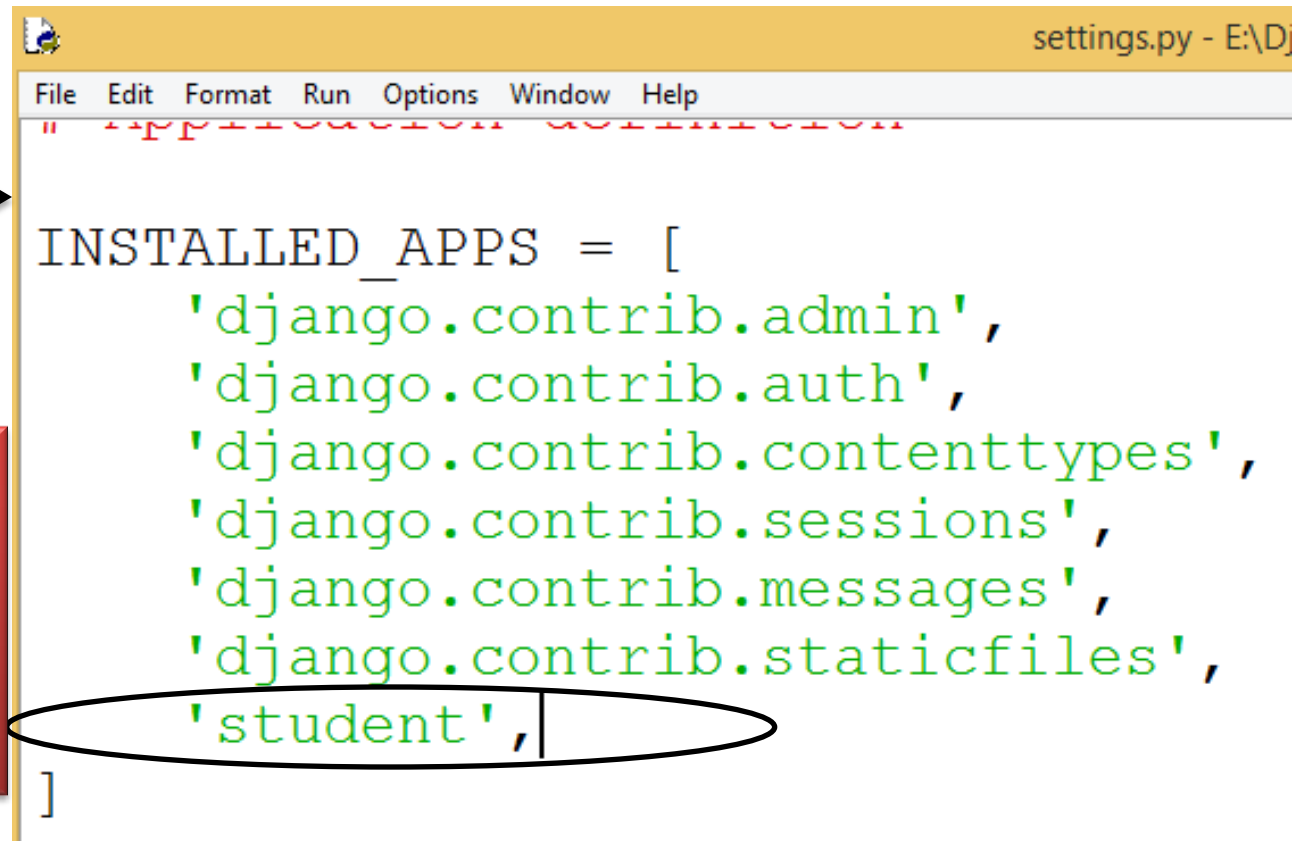


Registering the App in project

- By this time, app and project are separated. Now to register the app in project, we need to make changes in setting.py (by opening it in Python editor) from inner ClassXII folder.



We need to make an entry of student in `INSTALLED_APPS` in `setting.py`. By doing so, app will be registered in project.



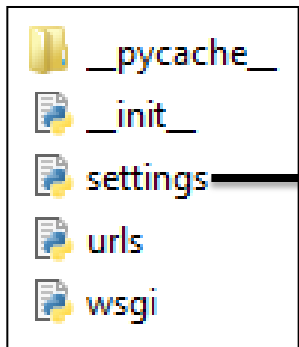
```
settings.py - E:\Dj
File Edit Format Run Options Window Help
"""APPLICATION CONFIGURATION"""

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'student',
]
```

The screenshot shows a Python IDE window titled 'settings.py - E:\Dj'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code content shows the `INSTALLED_APPS` list with several Django contrib apps and the 'student' app. The 'student' entry is circled in black.

Creating and saving of Template

- Because django works on MVT (Model Views Template) so, we need to create a folder in outer folder where we will keep our webpages. Folder can be of any name but we will be keeping only templates here. We will create html files inside it.
- We need to make an entry of this template in settings.py.



```
settings.py - E:\nFile Edit Format Run Options Window Help\n\nTEMPLATES = [\n    {\n        'BACKEND': 'django.template.backends.django.DjangoTemplates',\n        'DIRS': ['template'],\n        'APP_DIRS': True,\n        'OPTIONS': {\n            'context_processors': [\n                'django.template.context_processors.debug',\n                'django.template.context_processors.request',\n                'django.contrib.auth.context_processors.auth',\n                'django.contrib.messages.context_processors.messages',\n            ],\n        },\n    ],\n]
```

- Type name of 'templates' folder under DIRS key of TEMPLATES list.
- By doing so we are informing the project that there is an app with the name student whose all pages are inside template.

Creating and Saving Webpage

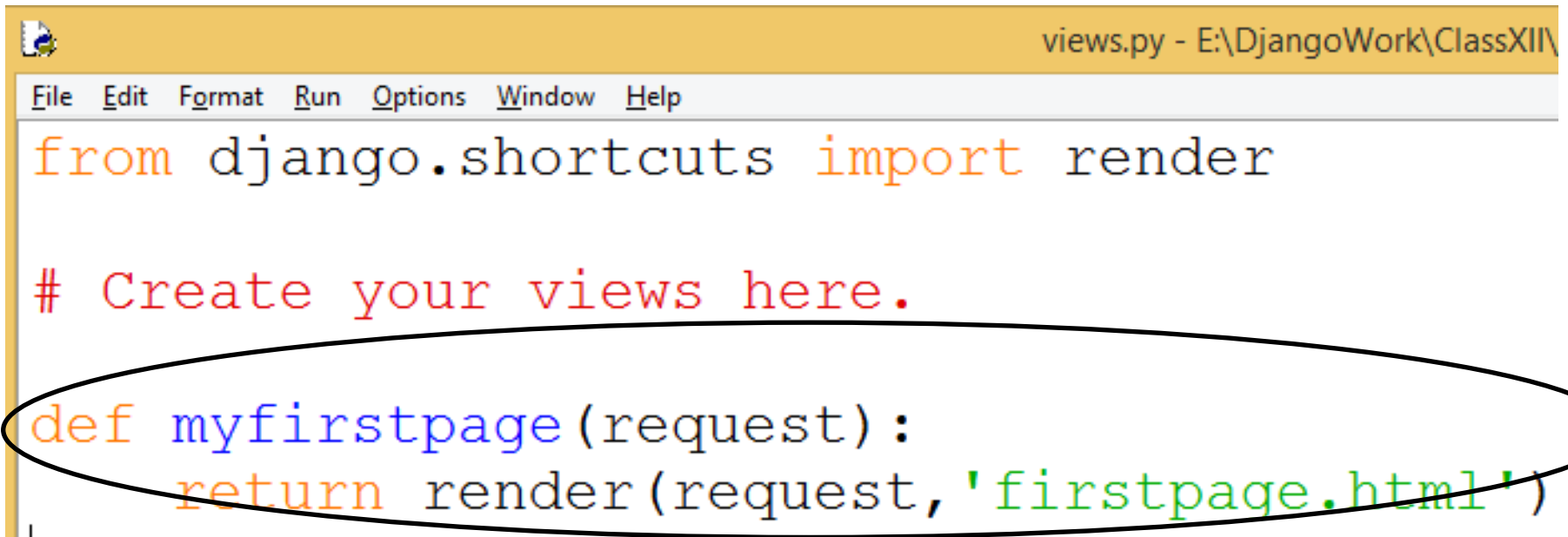
- A webpage is created with the name firstpage.html in Template folder.

```
firstpage - Notepad
File Edit Format View Help
<html>
<head><title>My First Page</title></head>
<body bgcolor=skyblue>
Hello All!<br>
This is my first Page.
This is designed using Django
</body>
</html>
```

- Now we will make an entry of this file in view's function.

Writing functions in Views.py

- Our next step is writing function for html pages in views.py.
- In this file, we will have all those functions which are to be executed on page.
- Views is used to render a page. For this, from student folder, views.py file is to be opened in IDLE and then create a view.



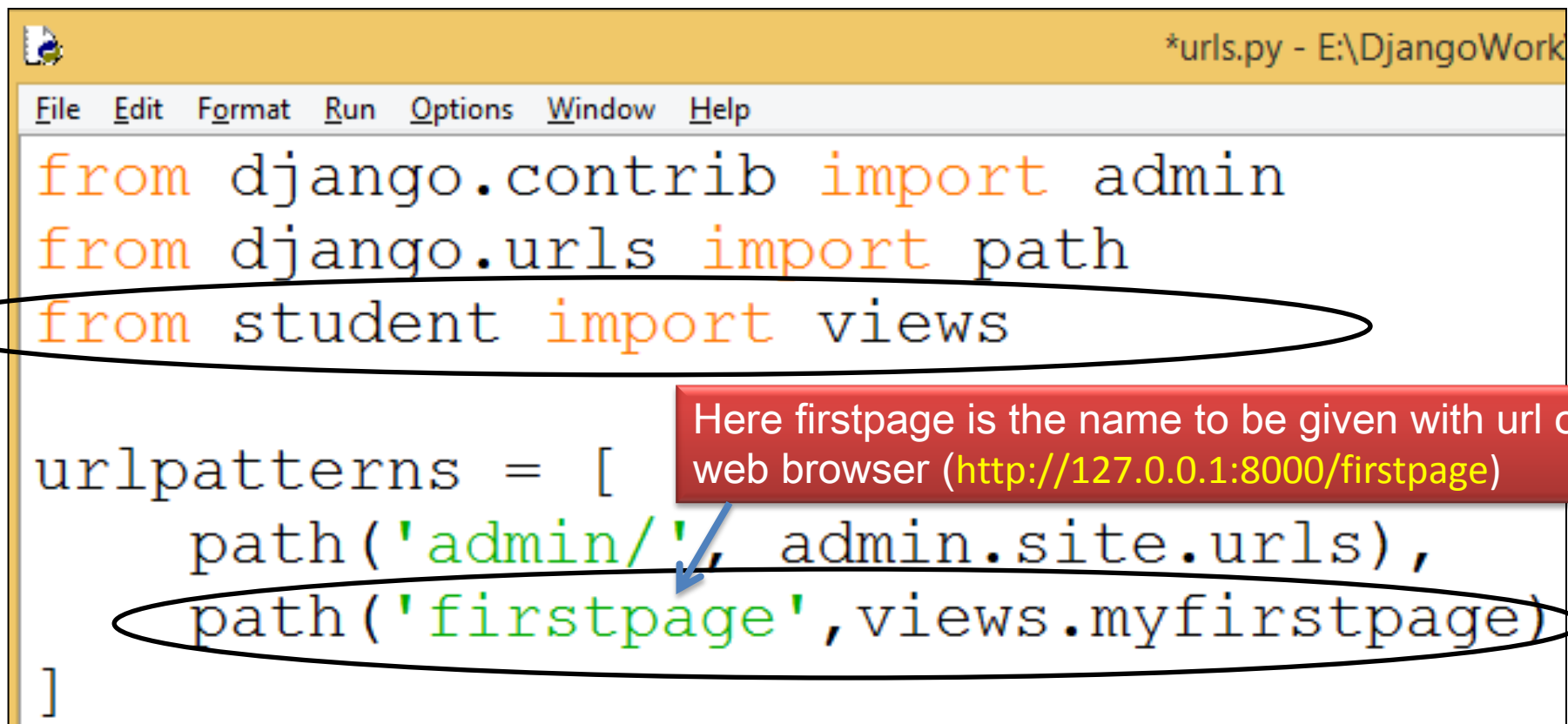
```
views.py - E:\DjangoWork\ClassXII\
File Edit Format Run Options Window Help
from django.shortcuts import render

# Create your views here.

def myfirstpage(request):
    return render(request, 'firstpage.html')
```

Setting views in Urls.py

- Now, we will add following code in inner project folder (ClassXII). Changes in urls.py ---



```
*urls.py - E:\DjangoWork
File Edit Format Run Options Window Help
from django.contrib import admin
from django.urls import path
from student import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('firstpage', views.myfirstpage)
]
```

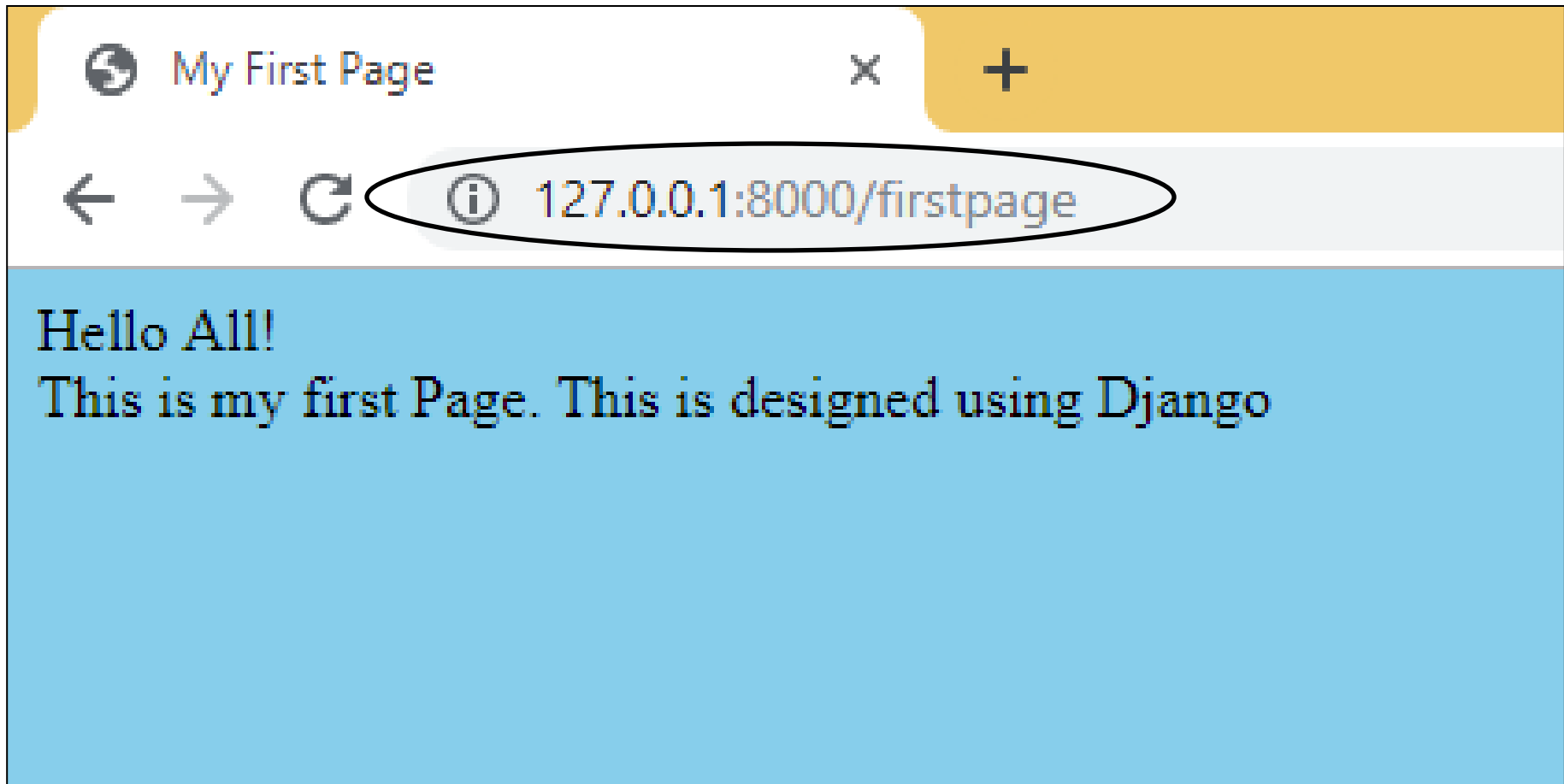
Here firstpage is the name to be given with url on web browser (<http://127.0.0.1:8000/firstpage>)

- Now save the file.
- Now run the django server from cmd by entering in project.

```
E:\DjangoWork\ClassXII>manage.py runserver
```

Displaying Webpage

- At last, by opening web browser, give the URL as per your created page then your html will be shown.



Django Based web Applications Part -2

- In this part, we have learned to display a webpage.
- We have learned project creation.
- We have learned app creation.
- We have learned creation of webpage through Template.

- In next part, we will learn adding various components in webpages. And will also learn sending data to server collected from a webpage.|
- Will also learn use of Get Method and POST Method.

Thank you

please follow us on our blog-

www.pythontrends.wordpress.com