**What is Django?**

Django is a high-level Python web framework that enables rapid development of websites. It is free and open source and has a thriving and active community support and great documentation.

**Why Django?**

Django helps you write software that is:

**Complete**

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product",

**Versatile**

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc).

**Secure**

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see [Website security](https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security) for more details of such attacks).

**Scalable**

Django uses a component-based “[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)” architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

**Maintainable**

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the [Model View Controller (MVC)](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture) pattern).

**Portable**

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

**History  
  
Where did it come from?**

Django was initially developed between 2003 and 2005 by a web team who were responsible for creating and maintaining newspaper websites. After creating a number of sites, the team began to factor out and reuse lot of common code and design patterns. This common code evolved into a generic web development framework, which was open-sourced as the "Django" project in July 2005.   
  
Django web applications typically group the code that handles each of these steps into separate files:



* **URLs:** A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL.
* **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
* **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
* **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. Sending the request to the right view (urls.py)

#### What is the Django development environment?

The development environment is an installation of Django on your local computer that you can use for developing and testing Django apps prior to deploying them to a production environment.

The main tools that Django itself provides are a set of Python scripts for creating and working with Django projects, along with a simple development webserver that you can use to test local (i.e. on your computer, not on an external web server) Django web applications on your computer's web browser.

#### What operating systems are supported?

Django web applications can be run on almost any machine that can run the Python 3 programming language: Windows, macOS X, Linux/Unix, Solaris, to name just a few.

#### What version of Python should you use?

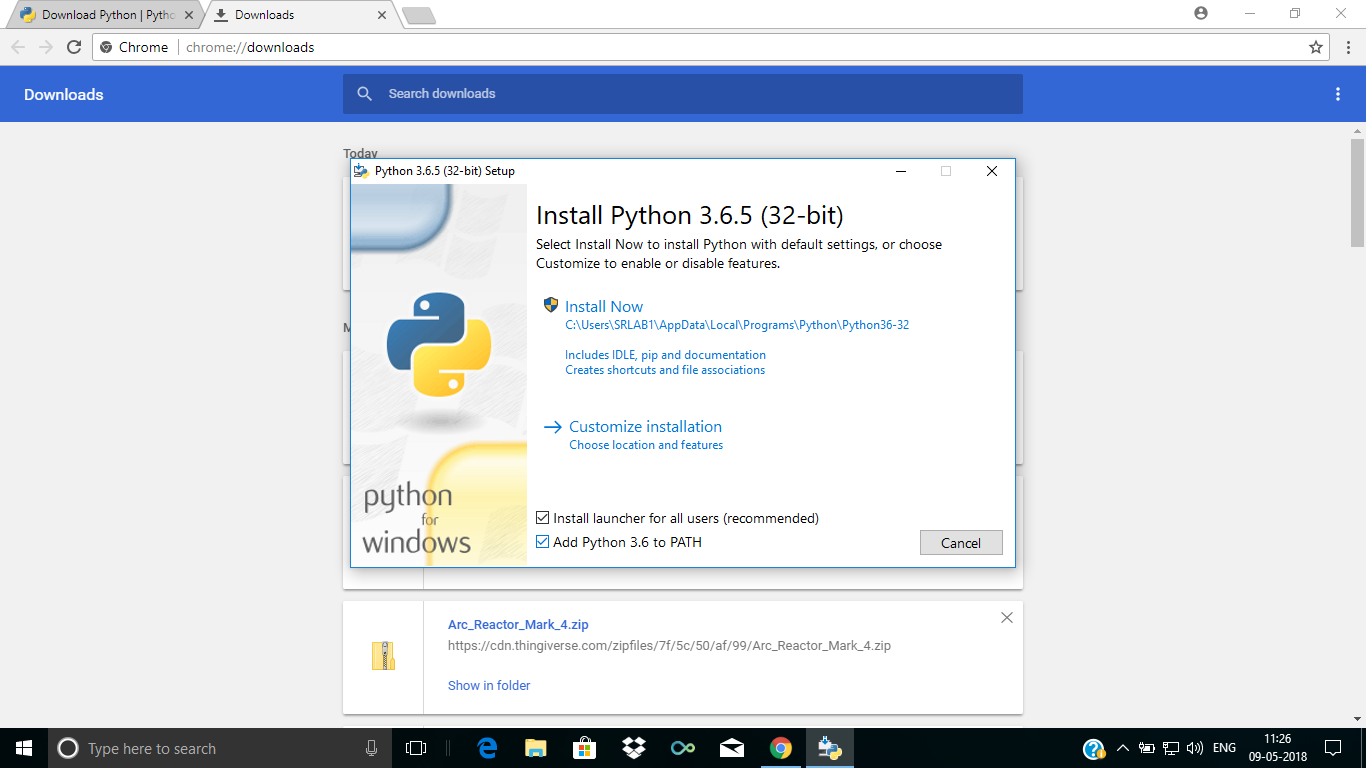
We recommend that you use the most recent version available — at time of writing this is Python 3.6.

#### Installation Instruction and creating the django project?

**1. Install python (only for the first time)**

a) go to **https://www.python.org/downloads/** and **click the big yellow button** that says “Download Python 3.6.5

b) Once you have downloaded the Python installer, **go to your Downloads folder and double click the file “python-3.x.x.msi”** to run the installer.



Make sure you have checked the **Add Python 3.6 to PATH**

**2. Install the Python Package Repository (PyPi) the *pip* tool (only for the first time)**

go to the command prompt and get back to root   
 c:\users\raj>**cd \** c:\> **python -m pip install -U pip**

**3. Install the Python Virtual Environment (only for the first time)**

c:\> **pip install virtualenv**

**5. Creating and enter a root directory for all your projects (only for the first time)**

c:\>**md workshop**

c:\>**cd workshop**

**6. Create a virtual environment (only for the first time)**

c:\workshop>**virtualenv django**

If you get inside the django folder, you will see the following:

\Include

\Lib

\Scripts

pip-selfcheck.json

**7. Activate the environment**

(django) c:\workshop>**cd scripts**

(django) c:\workshop\django>**cd scripts**

(django) c:\workshop\django\scripts>**activate**

**8. Install Django (only for the first time)**

(django) c:\workshop\django\scripts>**cd c:\workshop**

(django) c:\workshop>**pip install django**

**9. Create your Django project**

(django) c:\workshop>**django-admin startproject firstpj**

The following directory structure must have been created

firstpj/

manage.py

mysite/

\_\_init\_\_.py

settings.py

urls.py

wsgi.py

**10. Run the server locally**

(django) c:\workshop>**cd firstpj**

(django) c:\workshop\firstpj>**python manage.py runserver**

If you find the output as given below, then it is a success otherwise you need to recheck the steps

*Performing system checks...*

*System check identified no issues (0 silenced).*

*May 16, 2017 - 16:48:29*

*Django version 1.11, using settings 'mysite.settings'*

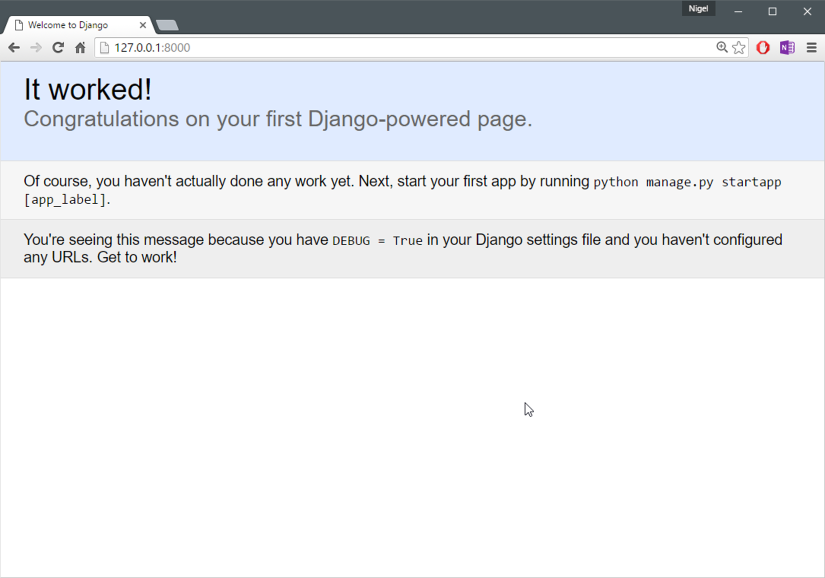
*Starting development server at http://127.0.0.1:8000/*

*Quit the server with CTRL-BREAK.*

Minimize the command prompt window

**11. Run the server locally**

Type **http://127.0.0.1:8000/** with your Web browser. You’ll see a “Welcome to Django” page . It worked!



Explanation of the files created:

* manage.py. A command-line utility that lets you interact with your Django project in various ways.
* The inner firstpj/ directory. It’s the Python package for your project. It’s the name you’ll use to import anything inside it (e.g. firstpj.urls).
* firstpj/\_\_init\_\_.py. An empty file that tells Python that this directory should be considered a Python package.
* firstpj/settings.py. Settings/configuration for this Django project.
* firstpj/urls.py. The URL declarations for this Django project; a “table of contents” of your Django-powered site.
* firstpj/wsgi.py. An entry-point for WSGI-compatible web servers to serve your project.

Our first django project pj1

Steps

1. (django) c:\workshop>**django-admin startproject pj1**
2. (django) c:\workshop\pj1>**python manage.py runserver**
3. (django) c:\workshop\pj1\pj1>

Edit the file url.py

**from django.conf.urls import url**

**from . import views**

**urlpatterns = [**

**url(r'^$', views.home, name='homepage'), ]**

1. (django) c:\workshop\pj1\pj1>

Edit views.py

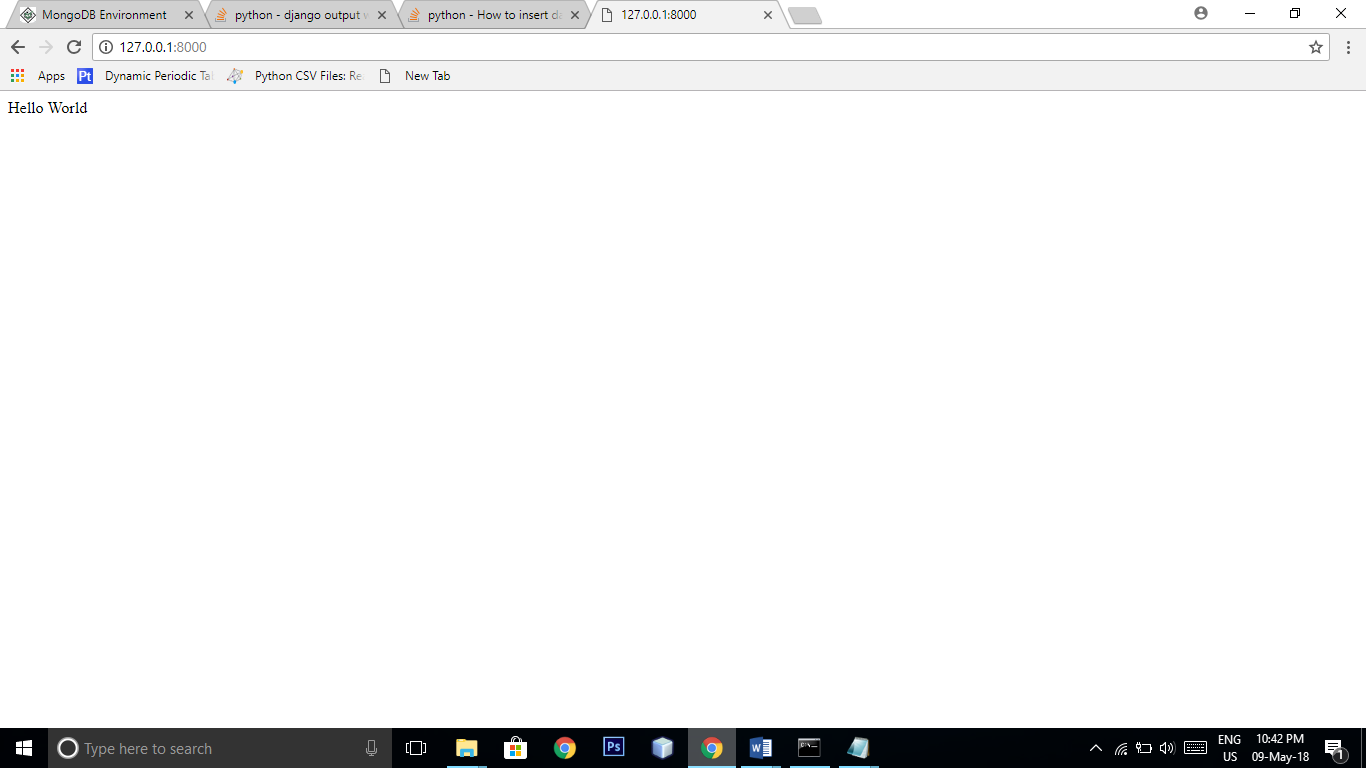
**from django.http import HttpResponse**

**def home(request):**

**return HttpResponse(‘Hello World’)**

1. Type **http://127.0.0.1:8000/** with your Web browser.

Output:



**-----------------------------------------------------------------------------------------------------------------**

Our second django project pj2

Steps

1. (django) c:\workshop>**django-admin startproject pj2**
2. (django) c:\workshop\pj2>**python manage.py runserver**
3. (django) c:\workshop\pj2\pj2>

Edit the file url.py

**from django.conf.urls import url**

**from . import views**

**urlpatterns = [**

**url(r'^$', views.home, name='homepage'), ]**

1. (django) c:\workshop\pj2\pj2>

Edit views.py

from django.http import HttpResponse

from django.template import engines

from django.template.loader import render\_to\_string

def home(request):

title='welcome page in HTML'

author='raj kumar pal'

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<center><h1>About ''' + title + '''</h1></center>

<center><p>Welcome to the <font color=red>First Page</font> in HTML</p></center>

<center><table border=2 bgcolor=yellow>

<tr><th>Roll</th><th>Name</th><th>Age</th></tr>

<tr><td>1</td><td>Raman</td><td>15</td></tr>

<tr><td>2</td><td>Sohan</td><td>17</td></tr>

</table>

<center><p><h1>Thank You</h1></center>

</body>

</html>'''

django\_engine = engines['django']

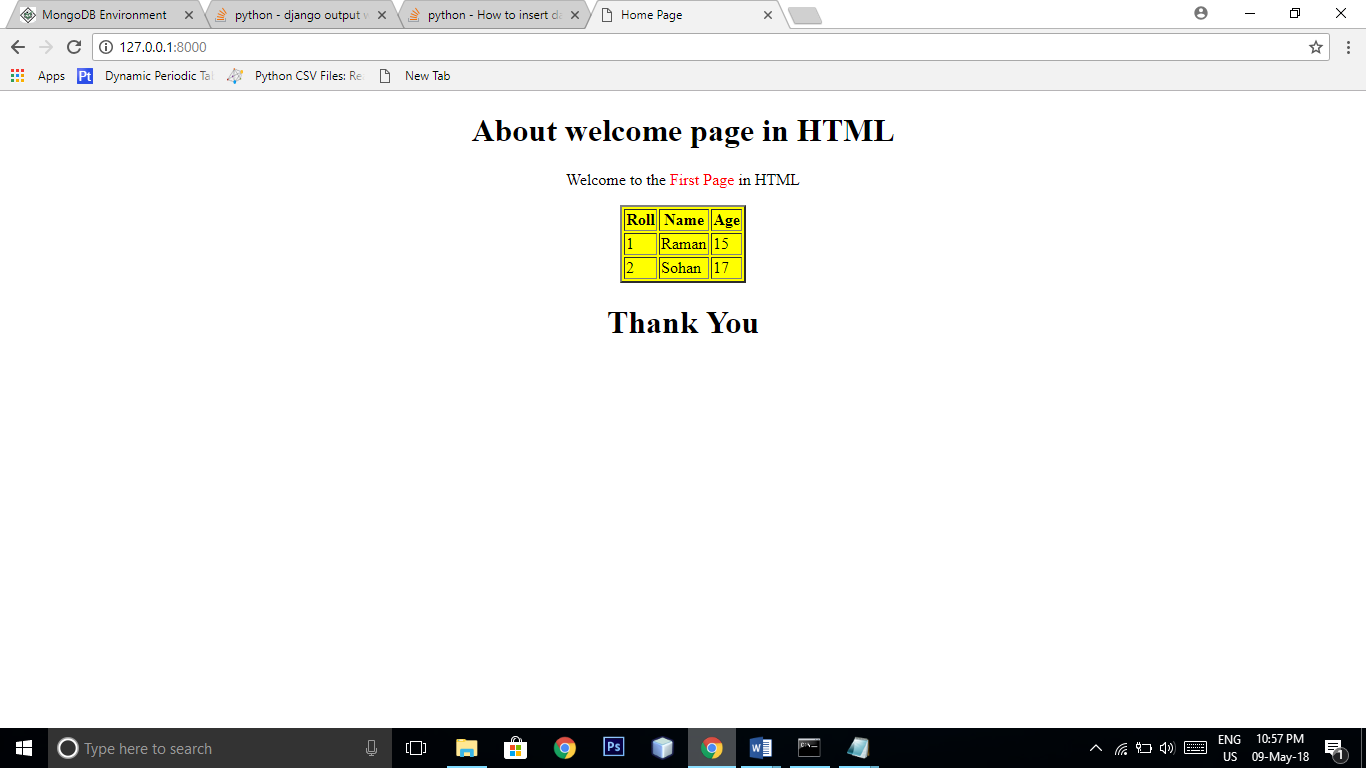
template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

return HttpResponse(html)

1. Type **http://127.0.0.1:8000/** with your Web browser.

Output:



----------------------------------------------------------------------------------------------------------------

Our third django project pj3

Steps

1. (django) c:\workshop>**django-admin startproject pj3**
2. (django) c:\workshop\pj3>**python manage.py runserver**
3. (django) c:\workshop\pj3\pj3>

Edit the file url.py

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$', views.home, name='homepage'),

url(r'^page1/$', views.page1, name='pageone'),

]

1. (django) c:\workshop\pj3\pj3>

Edit views.py

from django.http import HttpResponse

from django.template import engines

from django.template.loader import render\_to\_string

def home(request):

title='welcome page in HTML'

author='raj kumar pal'

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<center><h1>About ''' + title + '''</h1></center>

<center><p>Welcome to the <font color=red>First Page</font> in HTML</p></center>

<center><table border=2 bgcolor=yellow>

<tr><th>Roll</th><th>Name</th><th>Age</th></tr>

<tr><td>1</td><td>Raman</td><td>15</td></tr>

<tr><td>2</td><td>Sohan</td><td>17</td></tr>

</table>

<center>Click for <center><p><a href="{% url 'pageone' %}">next page</a>.</p></center>

</body>

</html>'''

django\_engine = engines['django']

template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

return HttpResponse(html)

def page1(request):

title = 'cal v1.0'

author ='raj kumar pal'

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>''' + title + '''</title>

</head>

<body>

<center><h1>About ''' + title + '''</h1></center>

<center><p>Thank You ''' + author + '''</p></center>

<center><p>We are on the next page</p></center>

<center><p><a href="{% url 'homepage' %}">Return to the homepage</a>.</p></center>

</body>

</html>'''

django\_engine = engines['django']

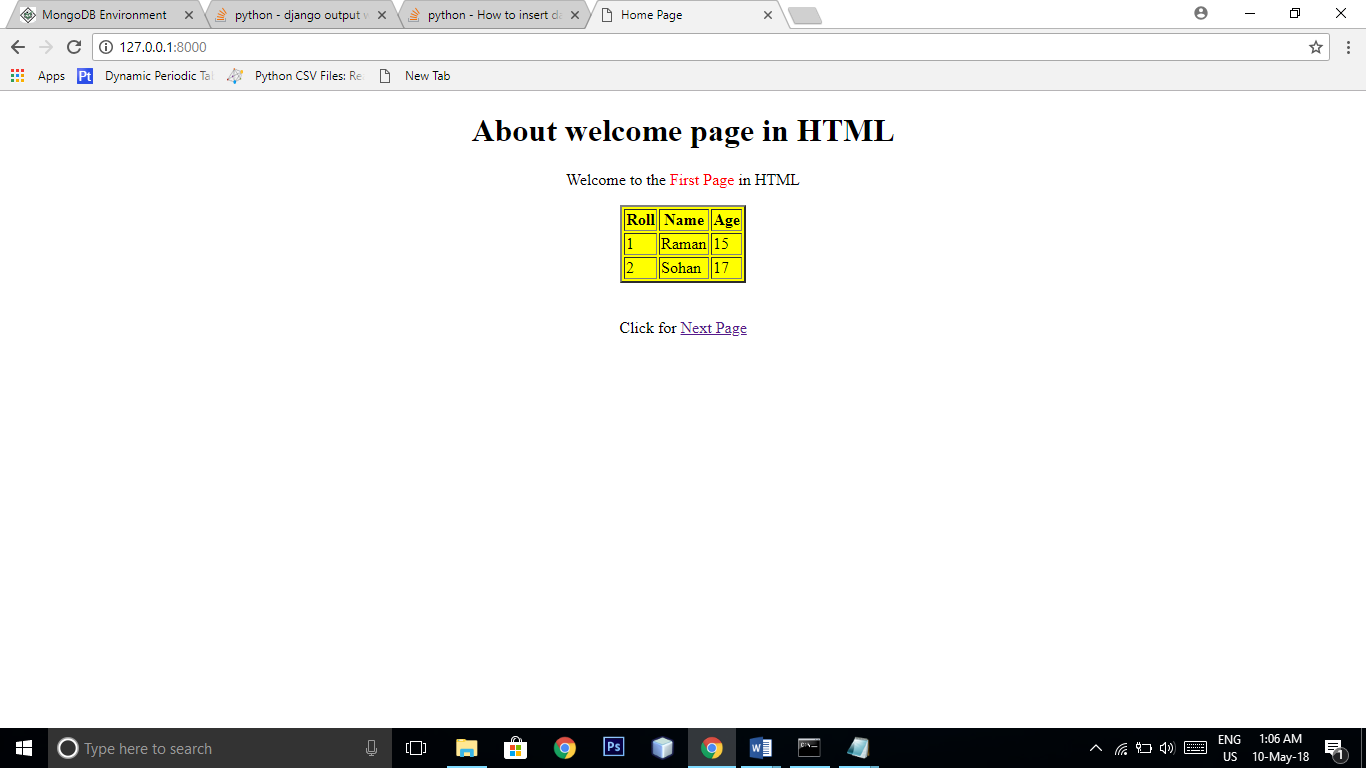
template = django\_engine.from\_string(about\_template)

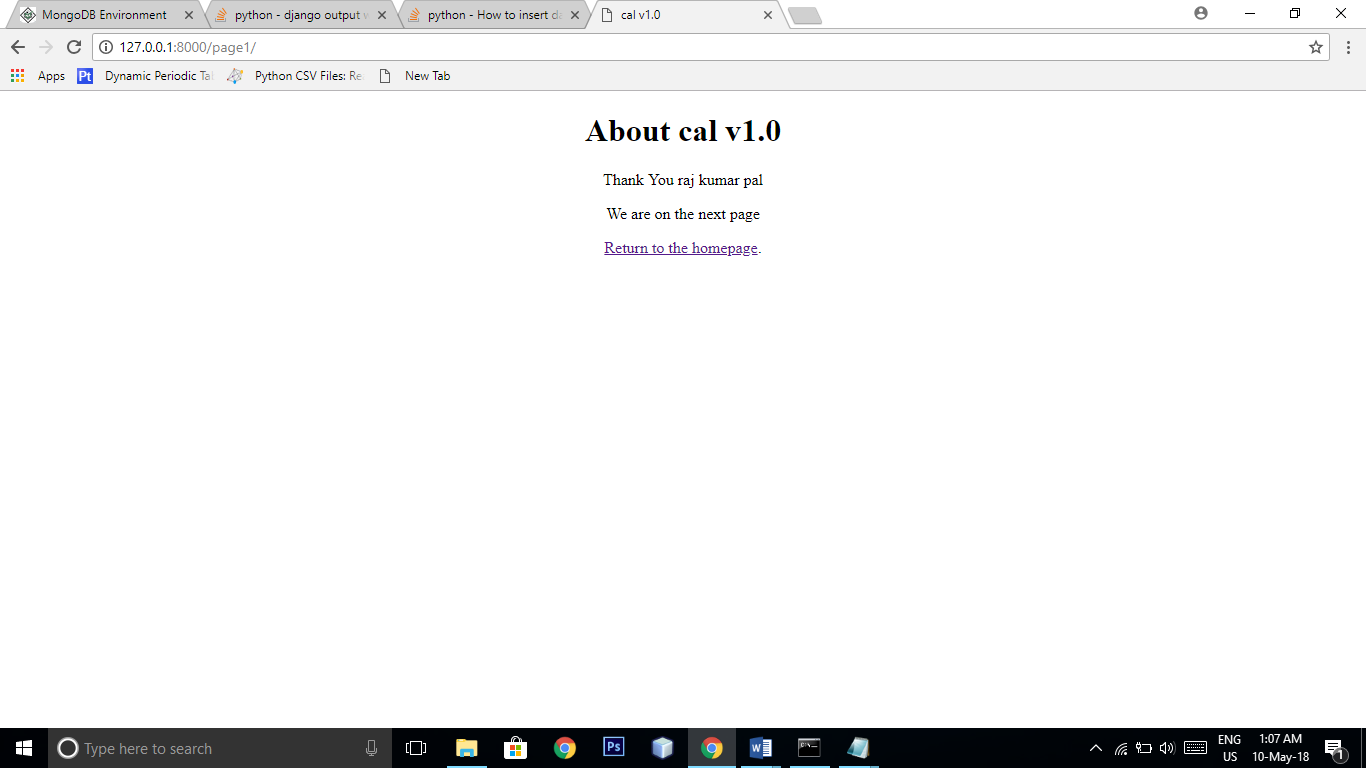
html = template.render({'title': title, 'author': author})

return HttpResponse(html)

1. Type **http://127.0.0.1:8000/** with your Web browser.

Output:





-------------------------------------------------------------------------------------------------------------------

Our fourth django project pj4

Steps

1. (django) c:\workshop>**django-admin startproject pj4**
2. (django) c:\workshop\pj4>**python manage.py runserver**
3. (django) c:\workshop\pj4\pj4>

Edit the file url.py

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$', views.home, name='homepage'),

url(r'^page1/$', views.page1, name='authpage'),

]

1. (django) c:\workshop\pj4\pj4>

Edit views.py

from django.http import HttpResponse

from django.template import engines

from django.template.loader import render\_to\_string

def home(request):

title='cal v1.0'

author='raj kumar pal'

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<center><h1>User Authentication</h1></center>

<center><p>Please verify your userid and password</p></center>

<br><br><br>

<center>

<form name="auth" action="page1" method="get">

Enter your userid

<input name="uid" type="text" value=""><br><br>

Enter your password

<input name="pwd" type="password" value=""><br><br>

<input name="sbmt" type="submit" value="submit">

<input name="sst" type="reset" value="reset">

</center>

</form>

</body>

</html>'''

django\_engine = engines['django']

template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

return HttpResponse(html)

def page1(request):

title='cal v1.0'

author='raj kumar pal'

uid = request.GET.get('uid')

pwd = request.GET.get('pwd')

if pwd=="rana9999":

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>Home Page</title>

</head>

<body>

<center><h1>About Calculator</h1></center>

<center><p>This Website has an application on a mathematical tool</p></center>

<br><br><br>

<center><h1>UId: = <font color=red>'''+uid+'''</font> and PWD: = <font color=blue>'''+pwd+'''</h1></center>

<center><h1><font color=green><a href="{% url 'homepage' %}">Home page</a></font><h1></center>

</body>

</html>'''

django\_engine = engines['django']

template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

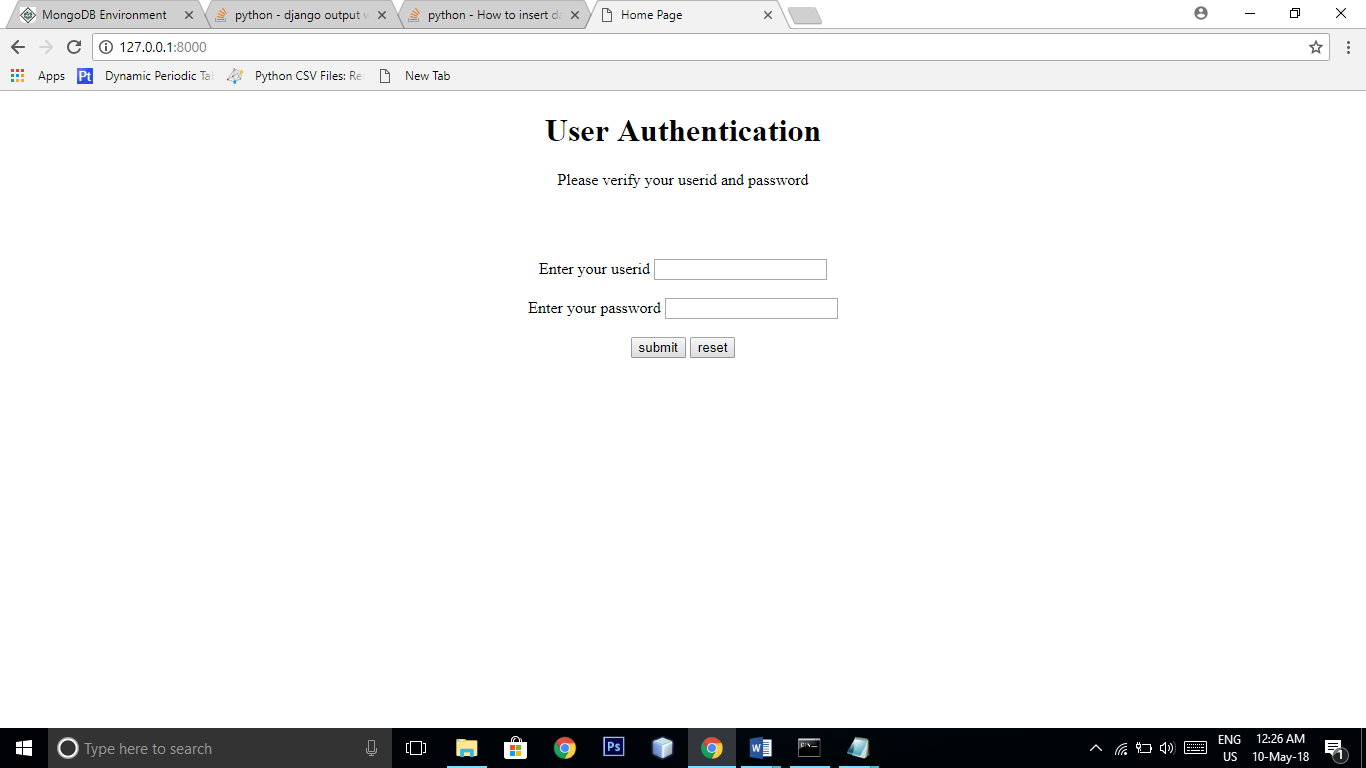
return HttpResponse(html)

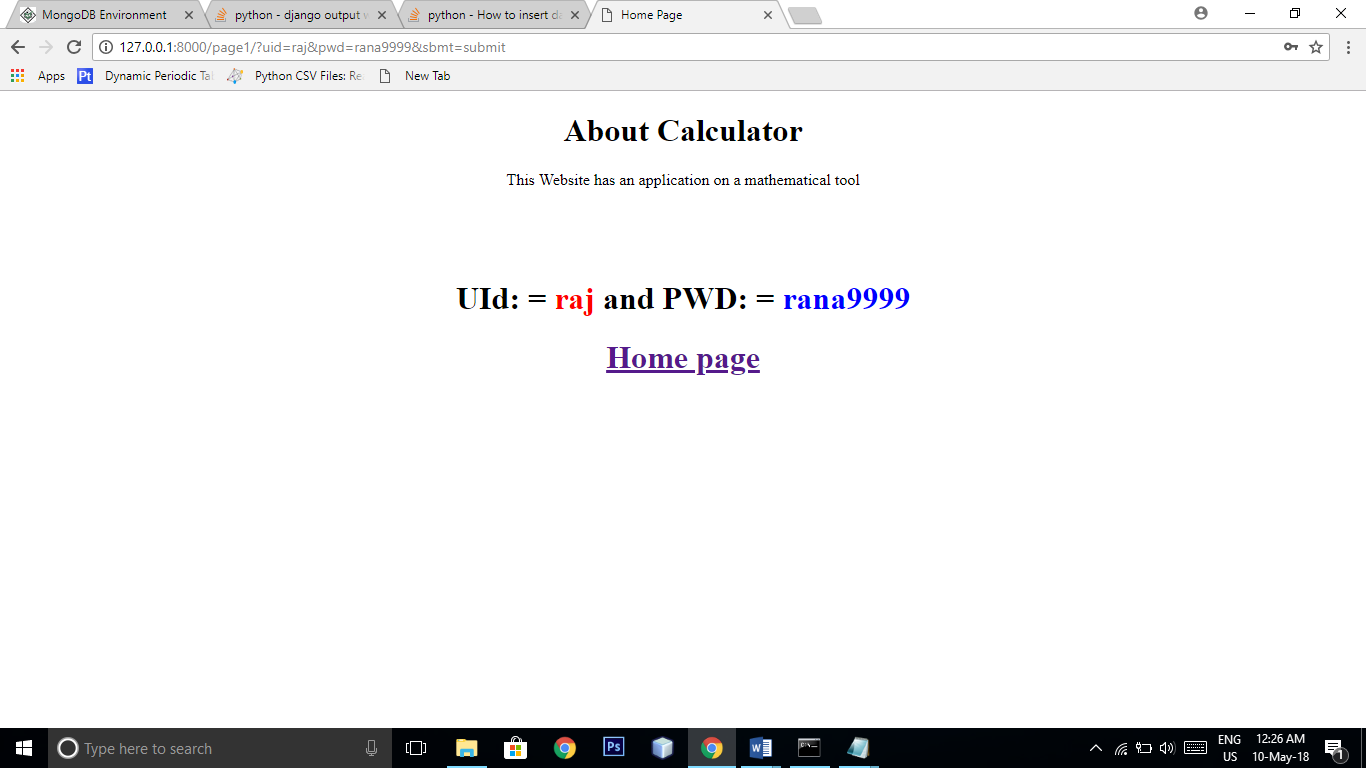
else:

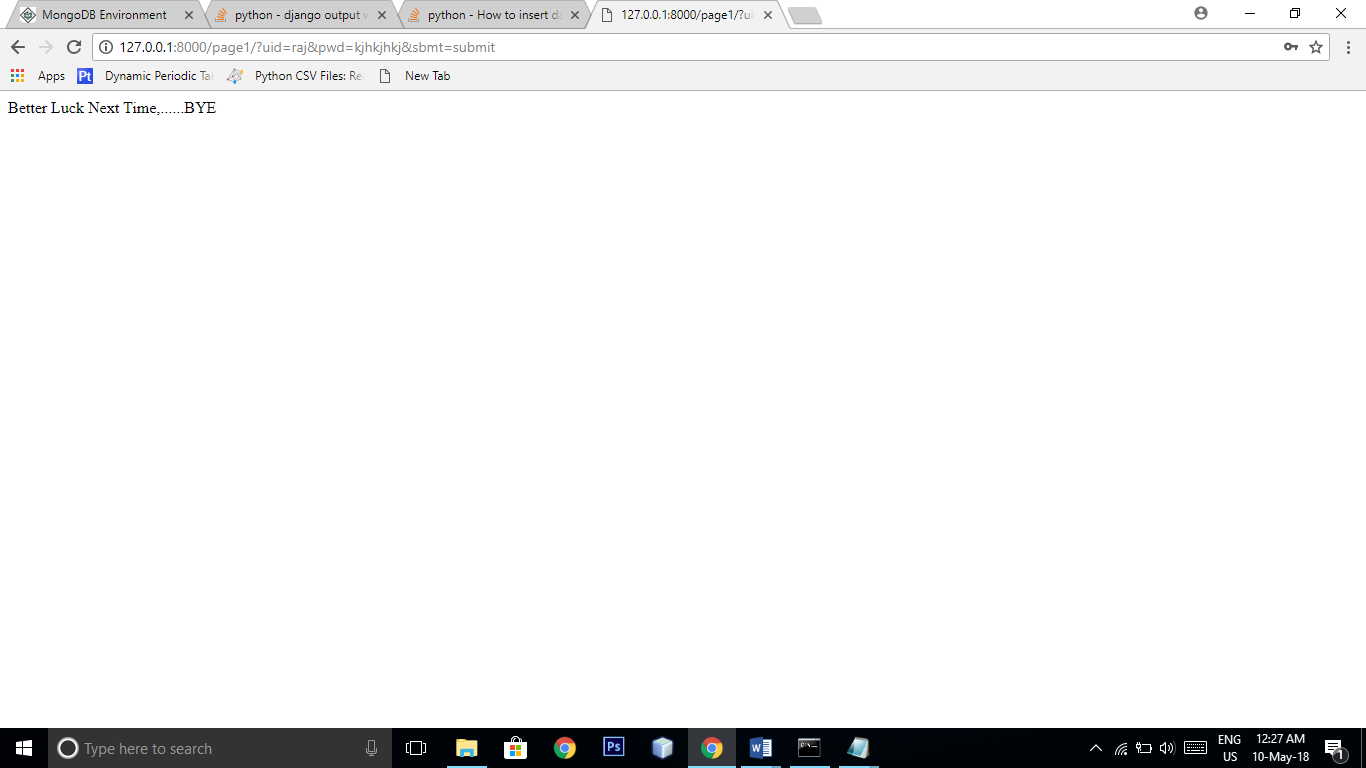
return HttpResponse("Better Luck Next Time,......BYE")

1. Type **http://127.0.0.1:8000/** with your Web browser.

Output:







Our fifth django project pj5

Steps

1. (django) c:\workshop>**django-admin startproject pj5**
2. (django) c:\workshop\pj5>**python manage.py runserver**
3. (django) c:\workshop\pj5\pj5>

Edit the file url.py

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$', views.home, name='homepage'),

url(r'^page1/$', views.page1, name='authpage'),

]

1. (django) c:\workshop\pj4\pj4>

Edit views.py

from django.http import HttpResponse

from django.template import engines

from django.template.loader import render\_to\_string

def home(request):

title = 'cal v1.0'

author ='raj kumar pal'

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>''' + title + '''</title>

</head>

<body>

<center><h1>About ''' + title + '''</h1></center>

<center><p>This Website was developed by ''' + author + '''.</p></center>

<center>

<form name="cal" action="\page1" method="get">

Enter the first value

<input name="val1" type="text" value=""><br>

Enter the second value

<input name="val2" type="text" value=""><br>

<input name="sbmt" type="submit" value="submit">

<input name="sst" type="reset" value="reset">

</center>

</form>

<center><p><a href="{% url 'homepage' %}">Return to the homepage</a>.</p></center>

</body>

</html>'''

django\_engine = engines['django']

template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

return HttpResponse(html)

def page1(request):

title = 'cal v1.0'

author ='raj kumar pal'

val1 = int(request.GET.get('val1'))

val2 = int(request.GET.get('val2'))

sum=val1+val2

about\_template = '''<!DOCTYPE html>

<html>

<head>

<title>''' + title + '''</title>

</head>

<body>

<center><h1>About ''' + title + '''</h1></center>

<center><p>The sum is ''' + str(sum) + '''</p></center>

<center><p><a href="{% url 'homepage' %}">Return to the homepage</a>.</p></center>

</body>

</html>'''

django\_engine = engines['django']

template = django\_engine.from\_string(about\_template)

html = template.render({'title': title, 'author': author})

return HttpResponse(html)

1. Type **http://127.0.0.1:8000/** with your Web browser.

Output:

