

Django Part -2

Understanding GET and POST Methods

Introduction

- ▶ In this presentation we are going to create a simple app that allows an HTML form to send data to a page using GET and POST Methods.
- ▶ We are not going to make things complicated for the learner and if you follow the simple steps that are in continuation with our first presentation (Django Part -1), you will easily be able to build and test this app.
- ▶ In this presentation we are going to first use the GET method to send data to another page and display it.
- ▶ In the second part we are going to achieve the same using POST method

GET Method explained

- ▶ GET is used to request data from a specified resource
- ▶ <http://localhost:8000/display?rollno=100&name=ashok>
- ▶ In the above example the data is sent through the HTTP header and is visible in the address bar
 - ▶ GET request can be cached
 - ▶ GET request remain in browser history
 - ▶ GET request can be bookmarked
 - ▶ GET request should never be used for sending sensitive information
 - ▶ GET request have length restrictions
 - ▶ GET request is generally used to request data and no to modify it.

POST Method explained

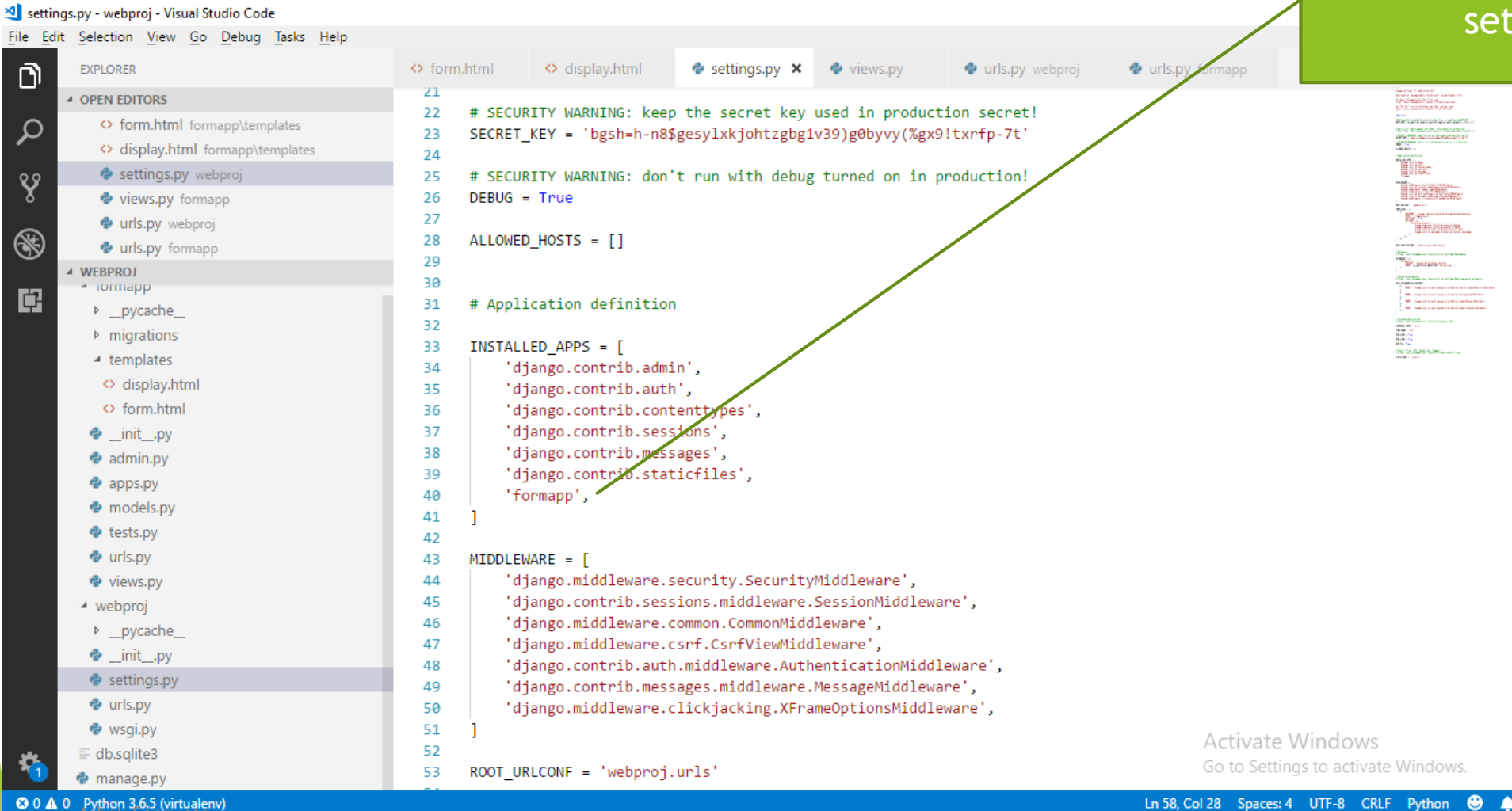
- ▶ POST method is used to send data to a server to create or modify a resource
 - ▶ POST requests are never cached
 - ▶ POST request do not remain in browser history
 - ▶ POST requests cannot be bookmarked
 - ▶ POST request have no restrictions on length
 - ▶ POST request are not visible in the HTTP header and are send as an attachment to the server.

Creating the formapp

- ▶ Step 1: Create a Django project inside your selected virtual environment and name it as webproj `(env1) D:\projects\django-admin startproject webproj`
- ▶ Step 2: Create a Django app inside the webproj and name it as formapp
`(env1) D:\projects>django-admin startapp formapp`
or
`(env1) D:\projects>python manage.py startapp formapp`
- ▶ Step 3: Now open your project in Visual Studio code (IDE used for python)
`(env1) D:\projects>code .`

Register your app in settings.py

Registering formapp in the settings.py



The screenshot shows the Visual Studio Code interface with the settings.py file open. The Explorer sidebar on the left shows the project structure, including the 'formapp' app directory. The main editor area displays the following Python code:

```
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'bgsh=h-n8$gesylxkjohtzgbg1v39)g0byvy(%gx9!txrfp-7t'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'formapp',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
52
53 ROOT_URLCONF = 'webproj.urls'
```

Activate Windows
Go to Settings to activate Windows.

Create the templates folder in formapp folder and create the form.html

Observe that we are using GET method

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** The file explorer on the left shows the project structure. The 'templates' folder is expanded, and 'form.html' is highlighted.
- Open Editors:** The 'form.html' file is open in the editor.
- Code Editor:** The code in 'form.html' is as follows:

```
1 <html>
2   <body>
3     <form action='display' method='GET'>
4       RollNo:<input type='text' name='rollno'><br>
5       Name1:<input type='text' name='name'><br>
6       <input type='submit' value='submit'>
7     </form>
8   </body>
9 </html>
```
- Taskbar:** The bottom status bar shows 'Python 3.6.5 (virtualenv)', 'Ln 6, Col 46', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

Activate Windows
Go to Settings to activate Windows.

Register the templates folder in settings.py

templates folder is registered in 'DIRS'

```
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
52
53 ROOT_URLCONF = 'webproj.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': ['templates'],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     },
69 ]
70
71 WSGI_APPLICATION = 'webproj.wsgi.application'
72
73
74 # Database
75 # https://docs.djangoproject.com/en/2.2/ref/settings/#databases
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
81     }
82 }
```

Activate Windows
Go to Settings to activate Windows.

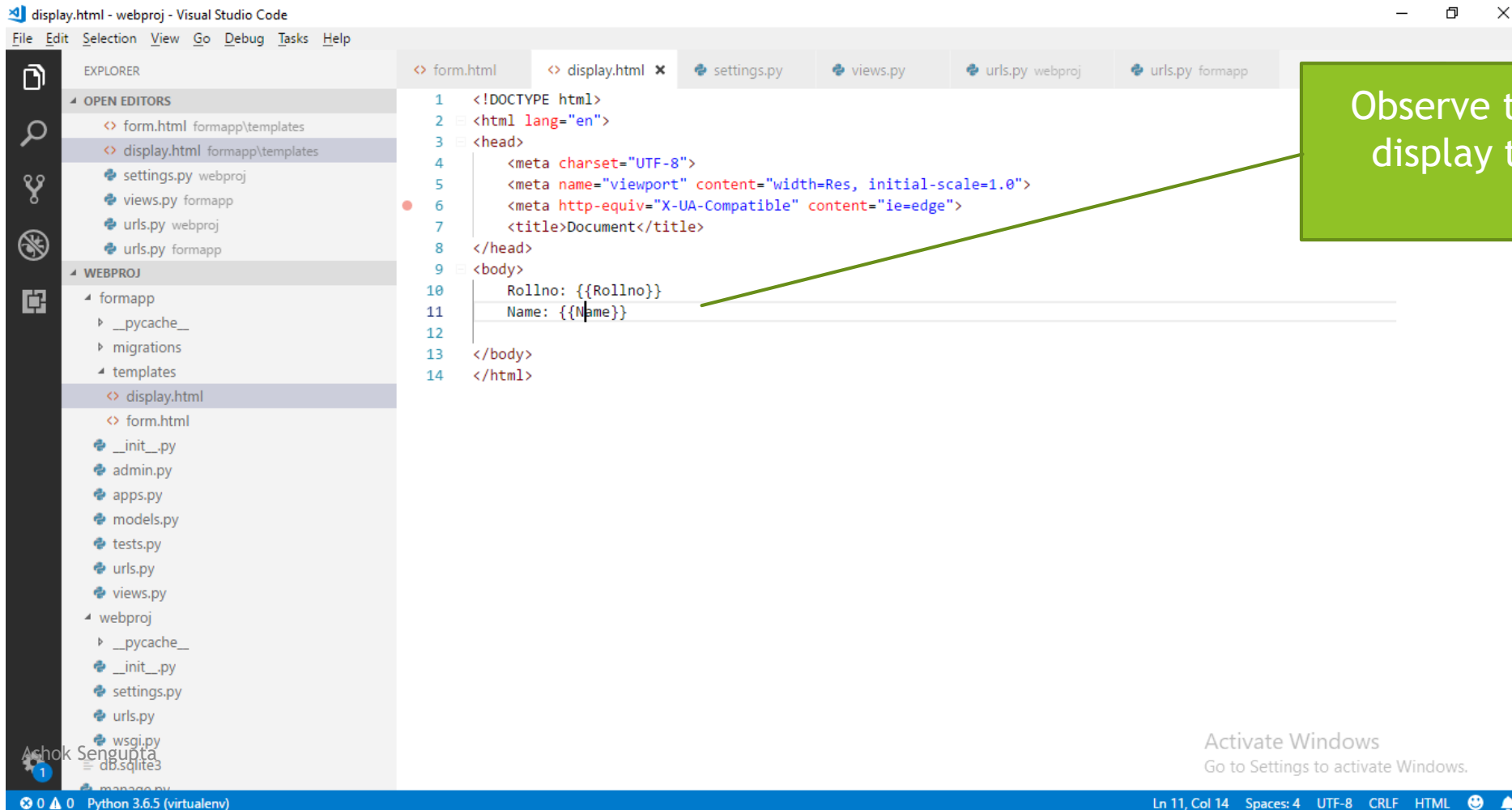
Edit the views.py to create two views. One for the formapp and other for the display

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def formapp(request):
5     return render(request, 'form.html', {})
6
7 def display(request):
8     rollno = request.GET['rollno']
9     name = request.GET['name']
10    return render(request, 'display.html', {'Rollno':rollno, 'Name':name})
```

Method to handle the form

Display method is to handle the GET request to the display.html page with information as Rollno and Name

Create the display.html page in the templates folder



The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** The file explorer on the left shows the project structure. The 'templates' folder is expanded, and 'display.html' is selected.
- Editor:** The main editor area displays the content of 'display.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=Res, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Document</title>
8 </head>
9 <body>
10   Rollno: {{Rollno}}
11   Name: {{Name}}
12
13 </body>
14 </html>
```
- Callout Box:** A green callout box on the right contains the text: "Observe the DTL used to display the Rollno and Name". A green line points from this box to the DTL tags in the code.
- Status Bar:** The bottom status bar shows "Ln 11, Col 14 Spaces: 4 UTF-8 CRLF HTML".

Observe the DTL used to display the Rollno and Name

Activate Windows
Go to Settings to activate Windows.

Ashok Sengupta
db.sqlite3

Python 3.6.5 (virtualenv)

Ln 11, Col 14 Spaces: 4 UTF-8 CRLF HTML

Edit the urls.py files in the webproj folder

```
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('formapp.urls')),
23 ]
24
```

File Edit Selection View Go Debug Tasks Help

EXPLORER

OPEN EDITORS

- form.html formapp\templates
- display.html formapp\templates
- settings.py webproj
- views.py formapp
- urls.py webproj
- urls.py formapp

WEBPROJ

- formapp
 - __pycache__
 - migrations
 - templates
 - display.html
 - form.html
- __init__.py
- admin.py
- apps.py
- models.py
- tests.py
- urls.py
- views.py
- webproj
 - __pycache__
 - __init__.py
 - settings.py
 - urls.py
 - wsqj.py
 - db.sqlite3
 - manage.py

Ashok Sengupta

Python 3.6.5 (virtualenv)

Ln 22, Col 38 Spaces: 4 UTF-8 CRLF Python

Don't forget to import the include

URL mapping for the formapp

Activate Windows
Go to Settings to activate Windows.

Create urls.py in formapp folder and edit it.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** Shows the project structure with folders like `formapp` and `webproj`. The file `urls.py` is highlighted in the `formapp` folder.
- Open Editors:** Lists several files, including `urls.py formapp` which is the active editor.
- Code Editor:** Contains the following Python code:

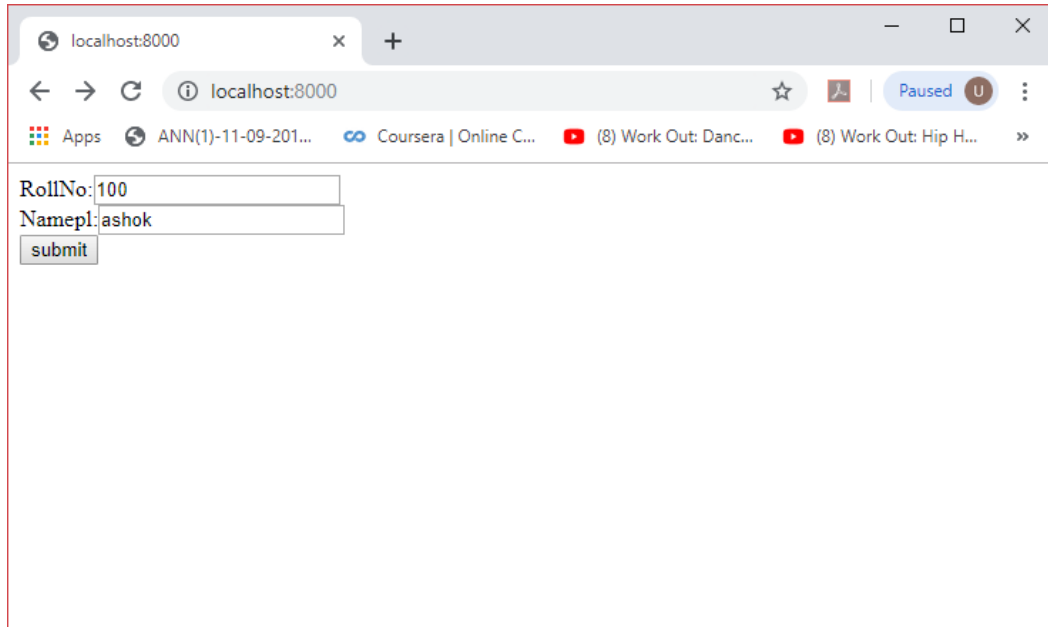
```
1 from django.urls import path
2 from . import views
3
4
5
6 urlpatterns = [
7     path('', views.formapp, name='formapp'),
8     path('display', views.display, name='display'),
9 ]
10
```
- Taskbar:** Shows the system tray with the text "Python 3.6.5 (virtualenv)", "Ln 10, Col 1", "Spaces: 4", "UTF-8", "CRLF", and "Python".

Need to add path entries for formapp and display in urlpatterns

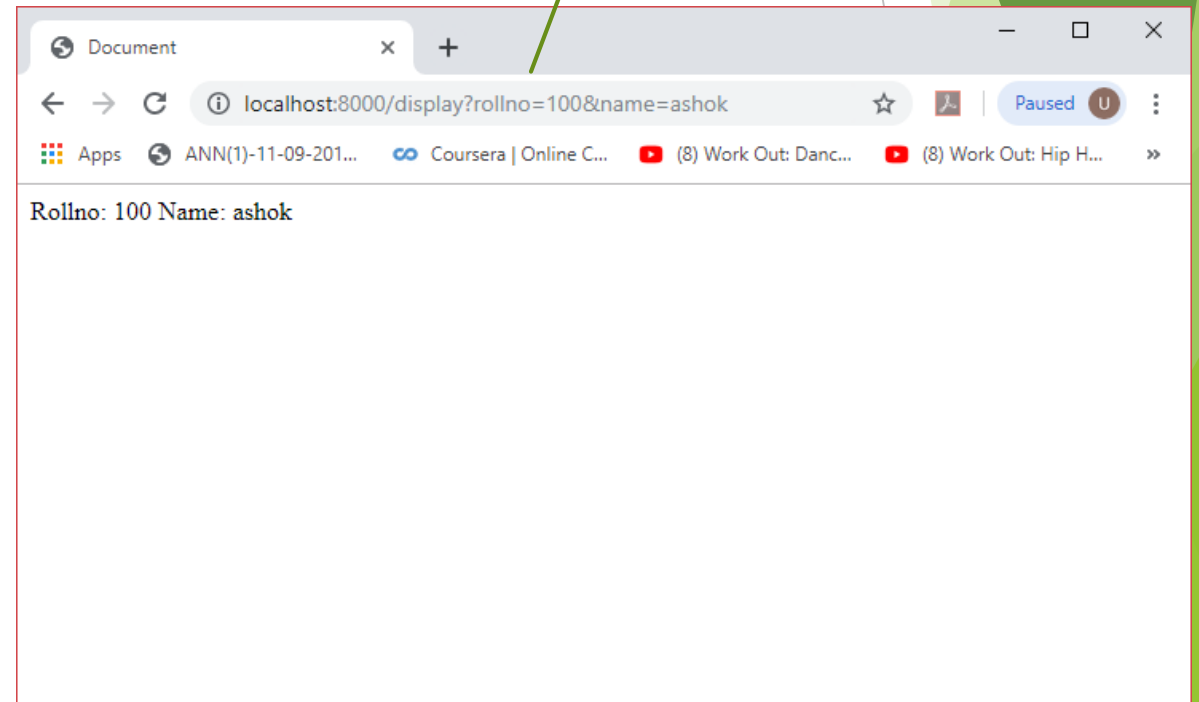
Activate Windows
Go to Settings to activate Windows.

That's it friends. You can test the app by running the server

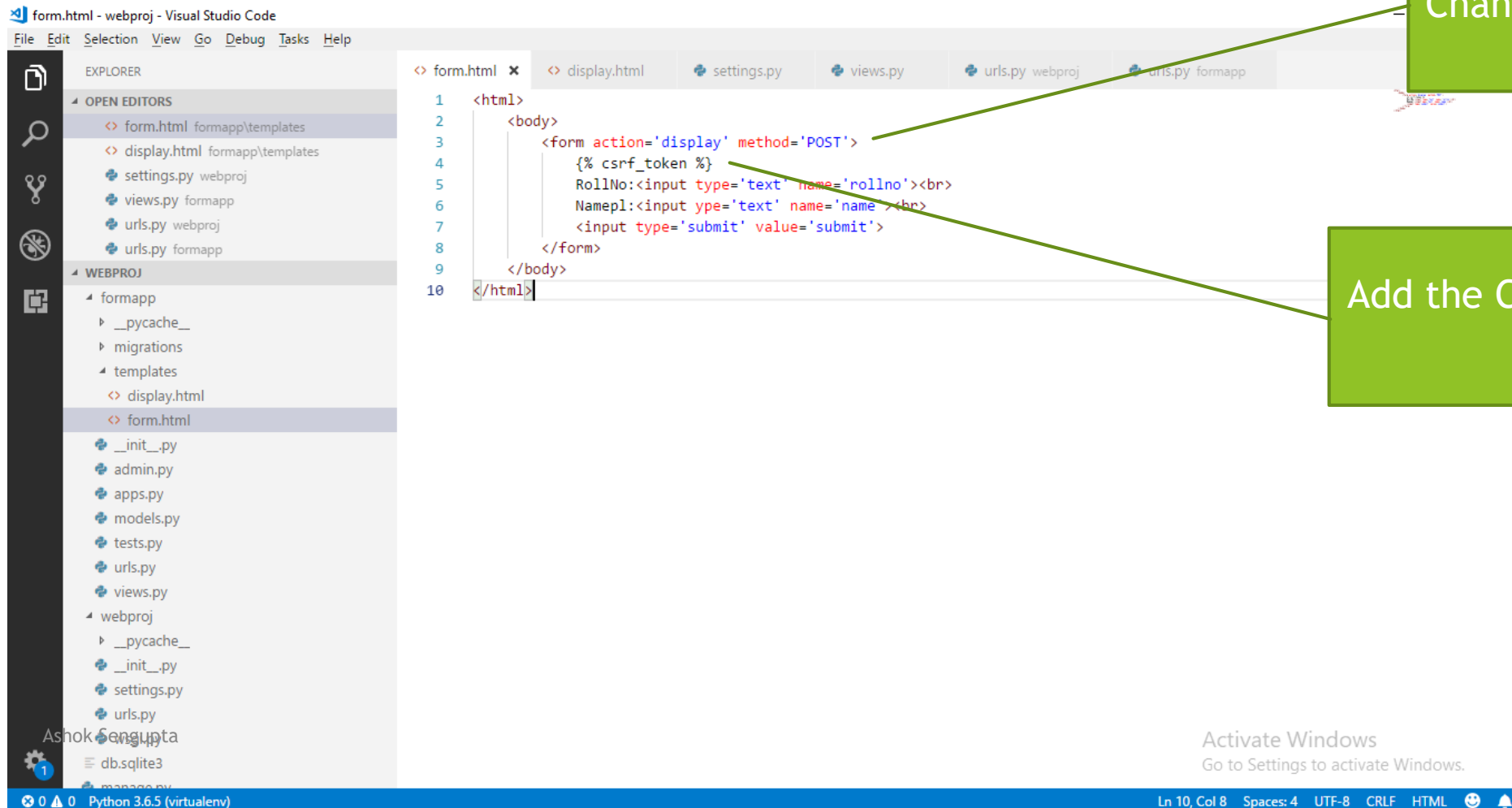
See in GET request the HTTP header contains the data



A screenshot of a web browser window at localhost:8000. The page contains a form with two input fields: "RollNo:" with the value "100" and "Namepl:" with the value "ashok". Below the fields is a "submit" button.



Only a few changes to achieve the same using POST method- first edit the form.html



Change the method to POST

Add the CSRF token in zinza code

Now edit the views.html

views.py - webproj - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXPLORER

OPEN EDITORS

- form.html formapp\templates
- display.html formapp\templates
- settings.py webproj
- views.py formapp
- urls.py webproj
- urls.py formapp

WEBPROJ

- formapp
 - __pycache__
 - migrations
 - templates
 - display.html
 - form.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- webproj
 - __pycache__
 - __init__.py
 - settings.py
 - urls.py
 - wsgi.py
 - manage.py

```
1 from django.shortcuts import render
2
3 # Create your views here.
4 def formapp(request):
5     return render(request, 'form.html', {})
6
7 def display(request):
8     rollno = request.GET['rollno']
9     name = request.GET['name']
10    return render(request, 'display.html', {'Rollno':rollno, 'Name':name})
```

Change the request to POST

Ashok Sengupta

Activate Windows
Go to Settings to activate Windows.

Python 3.6.5 (virtualenv) Ln 10, Col 72 Spaces: 4 UTF-8 CRLF Python

That's all friends the app will work the same way.

Thank You