

MMEC	M. M. ENGINEERING COLLEGE – MULLANA		LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET		
	EXPERIMENT TITLE: NOT APPLICABLE		
LABORATORY MANUAL NO.: MMEC-WI-BCSE-533L		ISSUE NO.: 01	ISSUE DATE: 04/07/2022
LABORATORY: Python & its framework Lab		SEMESTER: V	PAGE: 01 OF 01

MASTER LIST OF EXPERIMENT

S. No.	EXPERIMENT NO MMEC-WI-	EXPERIMENT DESCRIPTION	REV. NO.	REV. DATE	PAGE NO.
1.	BCSE-533L-01	Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.			
2.	BCSE-533L-02	Demonstrate how to create a basic project using MVT in Django.			
3.	BCSE-533L-03	Demonstrate how to activate & use Django's Admin module (interface) and to perform admin related tasks.			
4.	BCSE-533L-04	Develop a program to add Site Header, Site Title, Index Title in a Django Project.			
5.	BCSE-533L-05	Develop program to demonstrate adding tags using Django-Taggit.			
6.	BCSE-533L-06	Develop program to create a simple view for a web browser.			
7.	BCSE-533L-07	Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.			
8.	BCSE-533L-08	Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.			
9.	BCSE-533L-09	Develop program to send simple emails to the registered users of your Django application.			
10.	BCSE-533L-10	Demonstrate how to integrate Django with Reactjs using Django REST Framework.			
List of Projects					
11.	BCSE-533L-11 (Project-01)	Virtual Community			
12.	BCSE-533L-12 (Project-02)	Blog Application			

PREPARED BY: DR. RAJEEV GUPTA	APPROVED BY: HOD	Page 1
--------------------------------------	-------------------------	-----------------

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-01		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 01 OF 04

Aim: Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.

PYTHON INSTALLATION:

Step 1: Select Version of Python to download Full Installer and install.

Python >>> Downloads >>> Windows

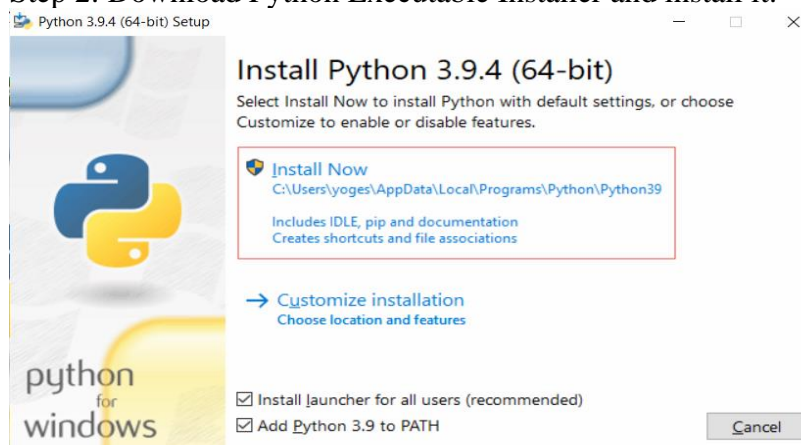
Python Releases for Windows

- Latest Python 3 Release - Python 3.9.4
- Latest Python 2 Release - Python 2.7.18

Files

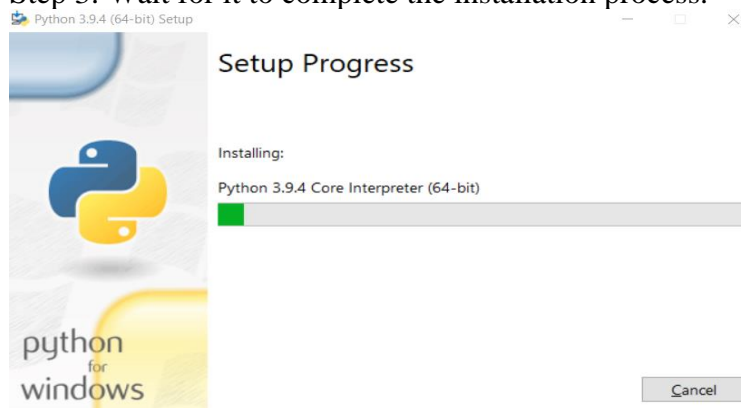
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		cc8507b3799ed4d8baa753acd8d5b35f	25411523	SIG
XZ compressed source tarball	Source release		2a3d8a5c75b695c45cf1806156e1a97	18900304	SIG
macOS 64-bit Intel installer	Mac OS X	for macOS 10.9 and later	2b974bd787f941fb8f80b5b8084e569	29866341	SIG
macOS 64-bit universal2 installer	Mac OS X	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	9aa68872b9582c6c71151d5d4d5ebca	37648771	SIG
Windows embeddable package (32-bit)	Windows		b4bd8ec0891891158000c6844222014d	7580762	SIG
Windows embeddable package (64-bit)	Windows		5c34eb7e73cfe8a92bf56b5168a459f4	8419530	SIG
Windows help file	Windows		aaacfe224768b5e4aa7583c12af68fb0	8859759	SIG
Windows installer (32-bit)	Windows		b790dda8f648f757bf0f23e4d05c053	27222976	SIG
Windows installer (64-bit)	Windows	Recommended	ebc65aaa142b1d6de450ce241c50e61c	28323440	SIG

Step 2: Download Python Executable Installer and install it.

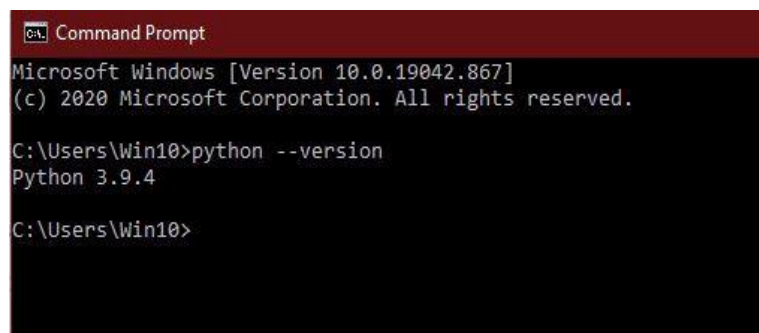


MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-01		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 02 OF 04

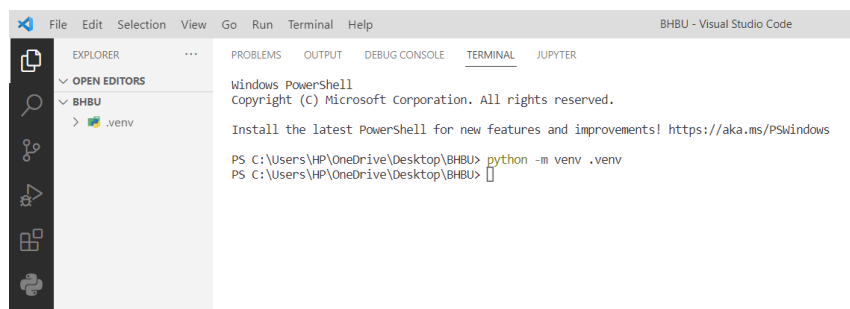
Step 3: Wait for it to complete the installation process.



Step 4: Verification of installation of python in Windows.



SETTING VIRTUAL ENVIRONMENT:



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-01		ISSUE NO.: 01
REV. DATE: NA		REV. NO.: 00
LABORATORY: Python & its Framework Lab		DEPTT.: CSE
		PAGE: 03 OF 04

```

File Edit Selection View Go Run Terminal Help
BHBH - Visual Studio Code

EXPLORER
  OPEN EDITORS
  BHBH
    .venv
      Include
      Lib
      Scripts
      pyvenv.cfg

TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP\OneDrive\Desktop\BHBH> .\.venv\Scripts\activate
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHBH>

```

DJANGO INSTALLATION:

```

File Edit Selection View Go Run Terminal Help
BHBH - Visual Studio Code

EXPLORER
  OPEN EDITORS
  BHBH
    .venv
      Include
      Lib
      Scripts
      pyvenv.cfg

TERMINAL
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHBH> pip install django
Collecting django
  Using cached django-4.1.1-py3-none-any.whl (8.1 MB)
Collecting asgiref<4,>=3.5.2
  Using cached asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting tzdata
  Using cached tzdata-2022.2-py2.py3-none-any.whl (336 kB)
Collecting sqlparse<0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1.1 sqlparse-0.4.2 tzdata-2022.2
WARNING: You are using pip version 21.2.3; however, version 22.2.2 is available.
You should consider upgrading via the 'C:\Users\HP\OneDrive\Desktop\BHBH\.venv\Scripts\python.exe -m pip install --upgrade pip' command.
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHBH>

```

VERIFICATION OF DJANGO:

```

File Edit Selection View Go Run Terminal Help
BHBH - Visual Studio Code

EXPLORER
  OPEN EDITORS
  BHBH
    .venv
      Include
      Lib
      Scripts
      pyvenv.cfg
      BlogProject

TERMINAL
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHBH\BlogProject> django-admin

Type 'django-admin help <subcommand>' for help on a specific subcommand.

Available subcommands:

[django]
check
compilemessages
createcachetable
dbshell
diffsettings
dumpdata
flush
inspectdb
loaddata
makemessages

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Tool Installation (Python, Virtual Environment, Django) and building a simple app from scratch.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-01		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 04 OF 04

CREATE A SIMPLE PROJECT:

```

File Edit Selection View Go Run Terminal Help
BHB - Visual Studio Code

EXPLORER
OPEN EDITORS
BHB
.venv
  Include
  Lib
  Scripts
  pyvenv.cfg
BlogProject
  BlogApp
  migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  views.py
BlogProject
  __pycache__
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
  manage.py

TERMINAL
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHB\> django-admin startproject BlogProject
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHB\> cd BlogProject
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHB\BlogProject> python manage.py startapp BlogApp
(.venv) PS C:\Users\HP\OneDrive\Desktop\BHB\BlogProject>

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to create a basic project using MVT in Django.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-02		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 01 OF 03

Aim: Demonstrate how to create a basic project using MVT in Django.

Step 1: Create a basic HTML file under template folder.

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the project structure with 'BlogApp' selected. The 'templates' folder is expanded, and 'home.html' is highlighted. The main editor shows the content of 'home.html' with the following code:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Home</title>
9 </head>
10
11 <body>
12     <h1>Welcome!</h1>
13 </body>
14
15 </html>

```

Step 2: Navigate to the 'views.py' file in the 'BlogApp' directory. Define a function 'homePage', it will render an HTML file called 'home.html'.

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the project structure with 'BlogApp' selected. The 'views.py' file is highlighted. The main editor shows the content of 'views.py' with the following code:

```

1 from django.shortcuts import render
2 from . import templates
3
4
5 def homePage(request):
6     return render(request, "home.html")
7

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to create a basic project using MVT in Django.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-02		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 02 OF 03

Step 3: Inside BlogApp/urls.py, include a URL pattern for view function.

```

BlogApp > urls.py > ...
1 from django.urls import path
2 from . import views
3 urlpatterns = [
4     path('home/', views.homePage),
5 ]
6

```

Step 4: Inside BlogProject/urls.py, include a URL configuration for the BlogApp.

```

BlogProject > urls.py > ...
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('BlogApp.urls')),
7 ]
8

```

Step 5: Register App 'BlogApp' in module BlogProject/settings.py under 'INSTALLED_APPS'.

```

BlogProject > settings.py > INSTALLED_APPS
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     'BlogApp',
41 ]
42

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to create a basic project using MVT in Django.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-02		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 03 OF 03

Step 6: In terminal, run server using 'python manage.py runserver' command.

```

File Edit Selection View Go Run Terminal Help BlogProject - Visual Studio Code
EXPLORER
OPEN EDITORS
BLOGPROJECT
  BlogApp
    __pycache__
    migrations
    templates
    admin
      base_site.html
      home.html
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  BlogProject

TERMINAL
System check identified no issues (0 silenced).
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

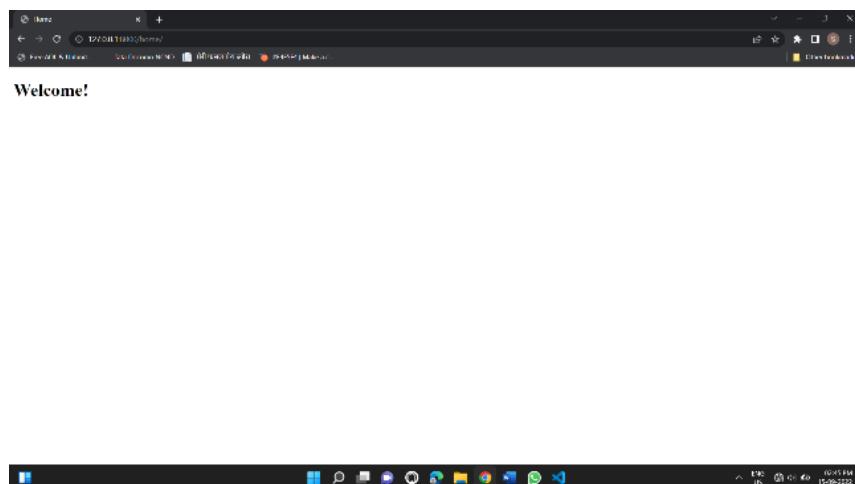
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\python\django\BlogProject> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 15, 2022 - 21:33:34
Django version 4.1.1, using settings 'BlogProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Not Found: /
[15/Sep/2022 21:33:39] "GET / HTTP/1.1" 404 2216
Not Found: /favicon.ico
[15/Sep/2022 21:33:39] "GET /favicon.ico HTTP/1.1" 404 2267
[15/Sep/2022 21:33:44] "GET /home/ HTTP/1.1" 200 279

```

OUTPUT: Open link in browser – 'http://127



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to activate & use Django's Admin module (interface) and to perform admin related tasks.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-03		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 01 OF 02

Aim: Demonstrate how to activate & use Django's Admin module (interface) and to perform admin related tasks.

Step 1: Create superuser using 'python manage.py createsuperuser'.

```

File Edit Selection View Go Run Terminal Help BlogProje
EXPLORER
OPEN EDITORS
BLOGPROJECT
  BlogApp
    _pycache_
    migrations
    templates
    admin
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS D:\python\django\BlogProject> python manage.py createsuperuser
Username (leave blank to use 'hp'): Saloni_Goyal
Email address: salonigoyal300@gmail.com
Password:
Password (again):
Superuser created successfully.
PS D:\python\django\BlogProject>

```

Step 2: In terminal, run server using 'python manage.py runserver' command.

```

File Edit Selection View Go Run Terminal Help BlogProject - Visual Studio Code
EXPLORER
OPEN EDITORS
BLOGPROJECT
  BlogApp
    _pycache_
    migrations
    templates
    admin
    base_site.html
    home.html
    _init_.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  BlogProject
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
System check identified no issues (0 silenced).
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

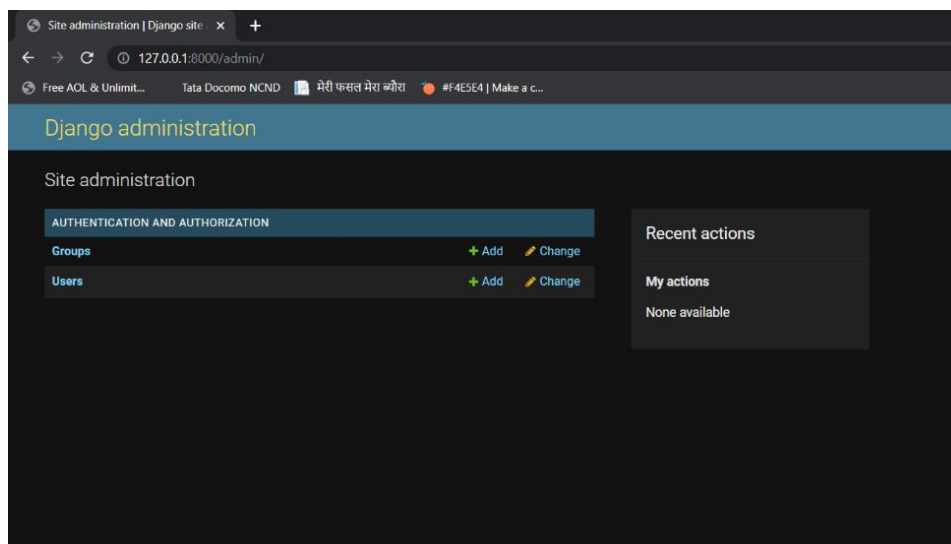
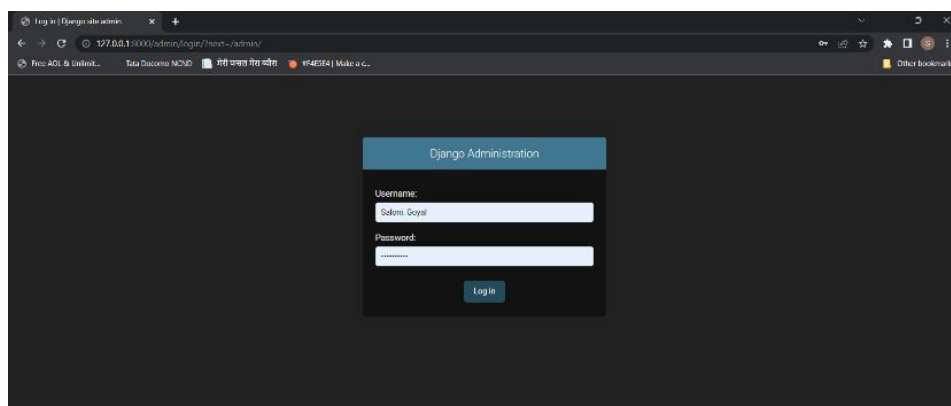
PS D:\python\django\BlogProject> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 15, 2022 - 21:33:34
Django version 4.1.1, using settings 'BlogProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Not Found: /
[15/Sep/2022 21:33:39] "GET / HTTP/1.1" 404 2216
Not Found: /favicon.ico
[15/Sep/2022 21:33:39] "GET /favicon.ico HTTP/1.1" 404 2267
[15/Sep/2022 21:33:44] "GET /home/ HTTP/1.1" 200 279

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to activate & use Django's Admin module (interface) and to perform admin related tasks.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-03	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 02 OF 02

Step 3: Login using username and password.



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop a program to add Site Header, Site Title, Index Title in a Django Project.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-04		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 01 OF 01

Aim: Develop a program to add Site Header, Site Title, Index Title in a Django Project.

Follow the steps-1 to 3 of Exp.-3 and adding Site Header, Site Title and Index Title to Django Admin Portal.

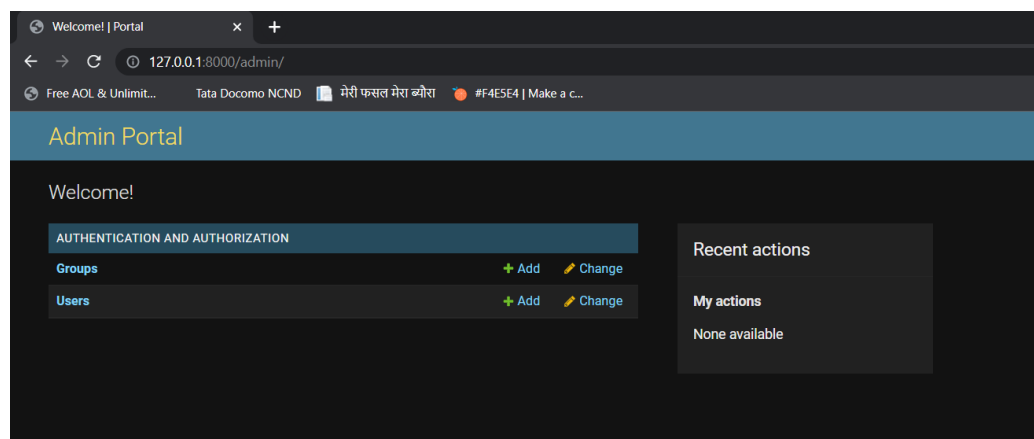
Step 4: Inside BlogProject/urls.py, add the following code.

```

1  from django.contrib import admin
2  from django.urls import path, include
3
4  admin.site.site_header = 'Admin Portal'
5  admin.site.index_title = 'Welcome!'
6  admin.site.site_title = 'Portal'
7
8  urlpatterns = [
9      path('admin/', admin.site.urls),
10     path('', include('BlogApp.urls')),
11 ]
12

```

OUTPUT: Open link in browser – ‘http://127.0.0.1:8000/admin/’



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to demonstrate adding tags using Django-Taggit.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-05		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 01 OF 02

Aim: Develop program to demonstrate adding tags using Django-Taggit.

Step 1: Installing django-taggit .

```
PS C:\Users\HP\OneDrive\Desktop\djjjjjj\URLconf> pip install django-taggit
WARNING: Ignoring invalid distribution -jango (c:\users\hp\appdata\local\programs\python\python39\lib\site-packages)
WARNING: Ignoring invalid distribution -jango (c:\users\hp\appdata\local\programs\python\python39\lib\site-packages)
Collecting django-taggit
  Downloading django_taggit-3.0.0-py3-none-any.whl (59 kB)
    59.9/59.9 kB 1.6 MB/s eta 0:00:00
Requirement already satisfied: Django>=3.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from django-taggit) (3.2.12)
Requirement already satisfied: pytz in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from Django>=3.2->django-taggit) (2022.1)
Requirement already satisfied: sqlparse<=0.2.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from Django>=3.2->django-taggit) (0.2.2)
Requirement already satisfied: asgiref<4, >=3.3.2 in c:\users\hp\appdata\local\programs\python\python39\lib\site-packages (from Django>=3.2->django-taggit) (3.3.2)
WARNING: Ignoring invalid distribution -jango (c:\users\hp\appdata\local\programs\python\python39\lib\site-packages)
Successfully installed django-taggit-3.0.0
```

Step 2: ADD it to Main Project's settings

```
# Application definition

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

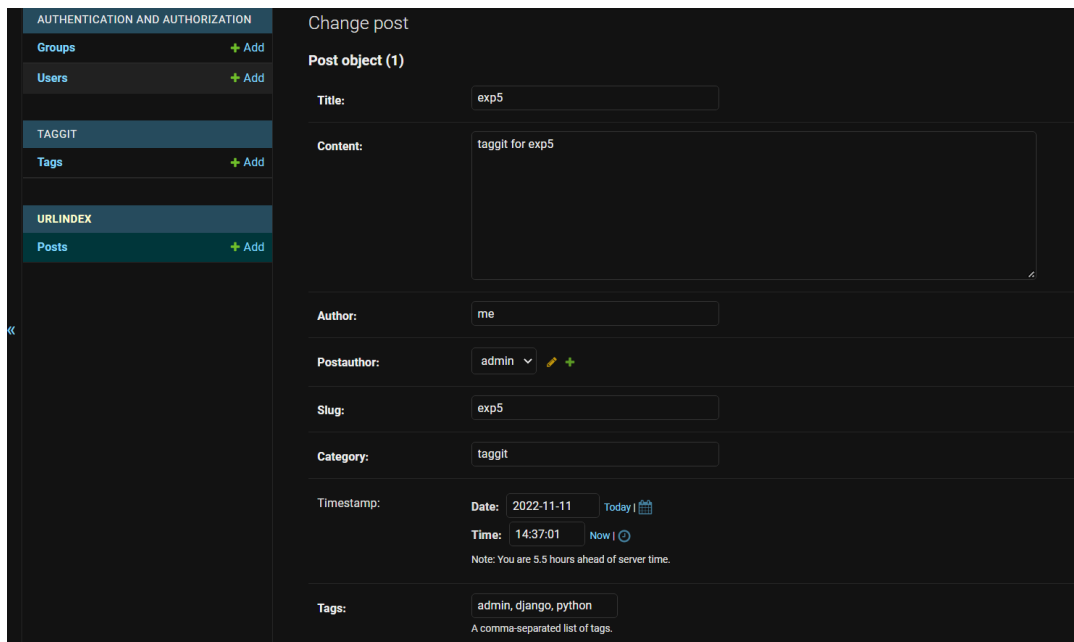
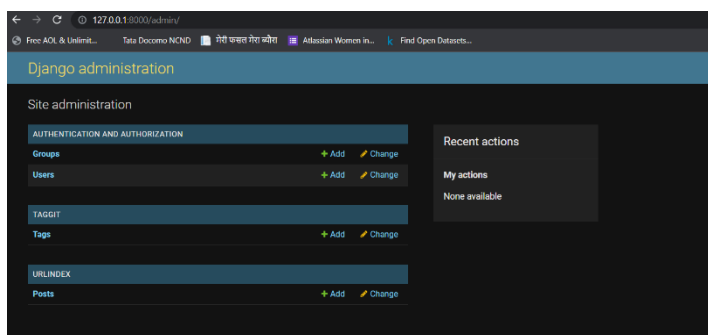
Step 3: Changes To models.py file.

```
from django.db import models
from django.utils.timezone import now
from taggit.managers import TaggableManager
from django.contrib.auth.models import User

class Post(models.Model):
    postid = models.AutoField(primary_key=True)
    title = models.CharField(max_length=255)
    content = models.TextField()
    author = models.CharField(max_length=100)
    postauthor = models.ForeignKey(User, on_delete=models.CASCADE)
    slug = models.SlugField(unique=True)
    category = models.CharField(max_length=200)
    timestamp = models.DateTimeField(default=now, blank=True)
    tags = TaggableManager()
```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to demonstrate adding tags using Django-Taggit.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-05		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 02 OF 02

Step 4: Run the following command ‘python manage.py runserver’.



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to create a simple view for a web browser.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-06		ISSUE NO.: 01
REV. DATE: NA		DEPTT.: CSE
LABORATORY: Python & its Framework Lab		SEMESTER: V
		PAGE: 01 OF 02

Aim: Develop program to create a simple view for a web browser.

Step 1: Create a HTML CSS web view.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta content="width=device-width, initial-scale=1.0" name="viewport">
6     <title>Saloni</title>
7     <meta content="" name="description">
8     <meta content="" name="keywords">
9
10    <!-- Favicon -->
11    <link href="{% static 'assets/img/favicon.png' %}" rel="icon">
12    <link href="{% static 'assets/img/apple-touch-icon.png' %}" rel="apple-touch-icon">
13
14    <!-- Google Fonts -->
15    <link
16        href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Baloo:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i"
17        rel="stylesheet">
18
19    <!-- Vendor CSS Files -->
20    <link href="{% static 'assets/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
21    <link href="{% static 'assets/vendor/bootstrap-icons/bootstrap-icons.css' %}" rel="stylesheet">
22    <link href="{% static 'assets/vendor/font-awesome/css/font-awesome.min.css' %}" rel="stylesheet">
23    <link href="{% static 'assets/vendor/glightbox/css/glightbox.min.css' %}" rel="stylesheet">
24    <link href="{% static 'assets/vendor/remixicon/remixicon.css' %}" rel="stylesheet">
25    <link href="{% static 'assets/vendor/swiper/swiper-bundle.min.css' %}" rel="stylesheet">
26
27    <!-- Template Main CSS File -->
28    <link href="{% static 'assets/css/style.css' %}" rel="stylesheet">
29
30    <!-- =====
31    * Template Name: Personal - v1.0.0
32    * Template URL: https://bootstrapmade.com/personal-free-resume-bootstrap-template/
33    * Author: Bootstrapmade.com
34    * License: https://bootstrapmade.com/license/
35    =====>
36
37 </head>

```

Step 2: Navigate to the ‘views.py’ file in the ‘BlogApp’ directory. Define a function ‘indexPage’, it will render an HTML file called ‘index.html’.

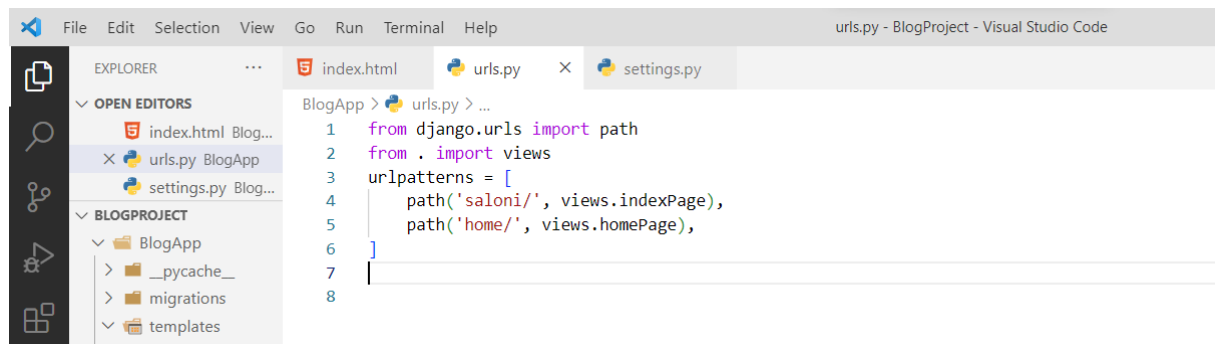
```

1 from django.shortcuts import render
2 from . import templates
3
4
5 def homePage(request):
6     return render(request, "home.html")
7
8
9 def indexPage(request):
10     return render(request, "index.html")
11
12

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to create a simple view for a web browser.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-06	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 02 OF 02

Step 3: Inside BlogApp/urls.py, include a URL pattern for view function.

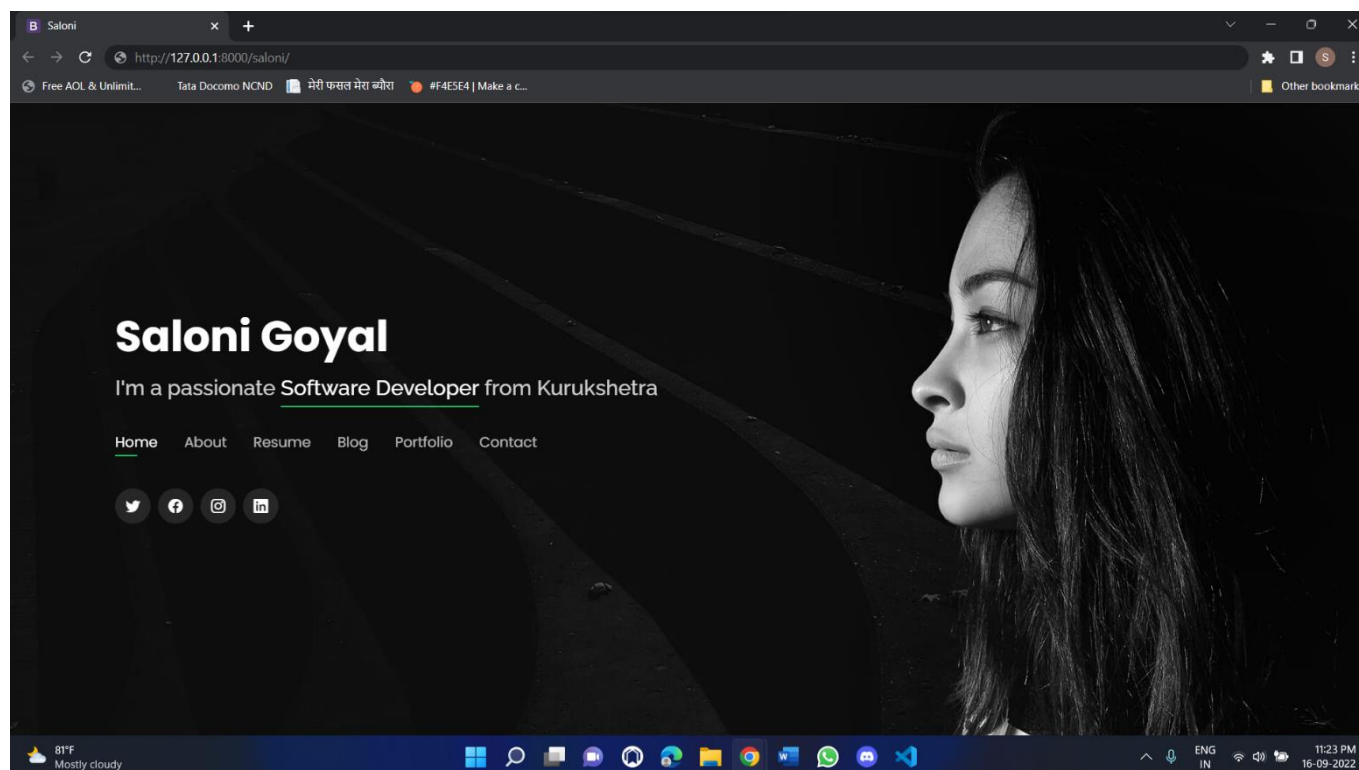


```

BlogApp > urls.py > ...
1  from django.urls import path
2  from . import views
3  urlpatterns = [
4      path('saloni/', views.indexPage),
5      path('home/', views.homePage),
6  ]
7
8

```

OUTPUT: Open link in browser – ‘http://127.0.0.1:8000/saloni/’



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL	
	PRACTICAL EXPERIMENT INSTRUCTION SHEET		
	EXPERIMENT TITLE: Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.		
EXPERIMENT NO.: MMEC-WI-BCSE-533L-07		ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA		REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab		SEMESTER: V	PAGE: 01 OF 04

Aim: Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.

Step 1: Register the app in settings.py file.

```

28
29 ALLOWED_HOSTS = []
30
31
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'BlogApp',
42
43 ]

```

Step 2: Create a basic HTML file under template folder.

```

BlogApp > templates > abouthtml > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>About page</title>
9  </head>
10
11 <body>
12     <h1> This is about page </h1>
13
14 </body>
15
16 </html>

```


MMEC	M. M. ENGINEERING COLLEGE – MULLANA		LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET		
	EXPERIMENT TITLE: Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.		
EXPERIMENT NO.: MMEC-WI-BCSE-533L-07		ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA		REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab		SEMESTER: V	PAGE: 02 OF 04

```

BlogApp > templates > index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Index Page</title>
9  </head>
10
11 <body>
12     <h1>This is index page</h1>
13 </body>
14
15 </html>

```

```

BlogApp > templates > home.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Home</title>
9  </head>
10
11 <body>
12     <h1>Welcome!</h1>
13 </body>
14
15 </html>

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA		LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET		
	EXPERIMENT TITLE: Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.		
EXPERIMENT NO.: MMEC-WI-BCSE-533L-07		ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA		REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab		SEMESTER: V	PAGE: 03 OF 04

Step 3: Navigate to the 'views.py' file in the 'URLIndex' directory. Define a function 'homePage', it will render an HTML file called 'home.html'; 'indexPage' it will render an HTML file called 'index.html' and 'aboutPage' it will render an HTML file called 'about.html'.

```

1  from django.shortcuts import render
2  from . import templates
3
4
5  def homePage(request):
6      |   return render(request, "home.html")
7
8
9  def indexPage(request):
10     |   return render(request, "index.html")
11
12
13  def aboutPage(request):
14     |   return render(request, "about.html")
15
16

```

Step 4: Inside URLIndex/urls.py, include a URL pattern for view function.

```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.indexPage),
    path('home/', views.homePage),
    path('about/', views.aboutPage),
    path('index/', views.indexPage),
]

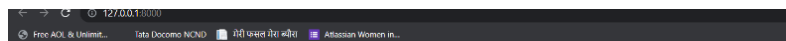
```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA		LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET		
	EXPERIMENT TITLE: Create a Python module called a URLconf and design URLs for an application. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.		
EXPERIMENT NO.: MMEC-WI-BCSE-533L-07		ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA		REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab		SEMESTER: V	PAGE: 04 OF 04

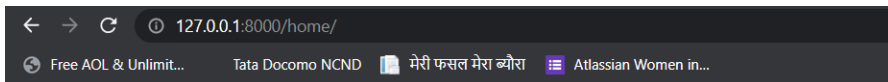
Step 5: In terminal, run server using ‘python manage.py runserver’ command.

```
performing system checks...
System check identified some issues:
WARNINGS:
: (staticfiles.W004) The directory 'D:\python\django\BlogProject\BlogProject\staticfiles' does not exist.
System check identified 1 issue (0 silenced).
October 19, 2022 - 22:19:06
django version 4.1.1, using settings 'BlogProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

OUTPUT: Open link in browser – ‘http://127.0.0.1:8000/’



This is index page



Welcome!

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-08		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 01 OF 04

Aim: Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.

Step 1: Create model in models.py in App.

```

1  from django.db import models
2
3
4  class ShashankModel(models.Model):
5
6      # fields of the model
7      name = models.CharField(max_length=200,default='name')
8      title = models.CharField(max_length=200)
9      description = models.TextField()
10
11     # renames the instances of the model
12     # with their title name
13     def __str__(self):
14         return self.title
15

```

Step 2: Create forms in forms.py in App.

```

from django import forms

from .models import ShashankModel

class ShashankForm(forms.ModelForm):

    class Meta:
        model = ShashankModel
        fields = '__all__'

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-08		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 02 OF 04

Step 3: Create views in views.py in App.

```

1  from .forms import ShashankForm
2  from django.shortcuts import get_object_or_404,render,HttpResponseRedirect
3  from .models import ShashankModel
4  from . import templates
5
6  def homePage(request):
7      return render(request, "home.html")
8
9  def indexPage(request):
10     return render(request, "index.html")
11
12 def aboutPage(request):
13     return render(request, "about.html")
14
15 def create_view(request):
16     context = {}
17     form = ShashankForm(request.POST or None, request.FILES or None)
18     if form.is_valid():
19         form.save()
20         context['form'] = form
21     return render(request, "create_view.html", context)
22
23 def update_view(request, id):
24     context = {}
25     obj = get_object_or_404(ShashankModel, id=id)
26     form = ShashankForm(request.POST or None, instance=obj)
27     if form.is_valid():
28         form.save()
29         return HttpResponseRedirect('/list')
30     context["form"] = form
31     return render(request, "update_view.html", context)
32
33 def list_view(request):
34     context = {}
35     context["dataset"] = ShashankModel.objects.all()
36     return render(request, "list_view.html", context)
37
38 def delete_view(request, id):
39     context = {}
40     obj = get_object_or_404(ShashankModel, id = id)
41     if request.method == "POST":
42         obj.delete()
43     return HttpResponseRedirect("/")
44     return render(request, "delete_view.html", context)
45

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-08	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 03 OF 04

Step 4: Create html files in templates in App.

```

1 <form method="POST" enctype="multipart/form-data">
2
3     <!-- Security token -->
4     {% csrf_token %}
5
6     <!-- Using the formset -->
7     {{ form.as_p }}
8
9     <input type="submit" value="Submit">
0 </form>
1 |

```

```

<div class="main">
  <!-- Create a Form -->
  <form method="POST">
    <!-- Security token by Django -->
    {% csrf_token %}

    <!-- form as paragraph -->
    {{ form.as_p }}

    <input type="submit" value="Update">
  </form>
</div>

```

```

<div class="main">

  {% for data in dataset %}.

  {{ data.name }}<br />
  {{ data.title }}<br />
  {{ data.description }}<br />
  <hr />

  {% endfor %}

</div>

```

```

<div class="main">
  <!-- Create a Form -->
  <form method="POST">
    <!-- Security token by Django -->
    {% csrf_token %}
    Are you want to delete this item ?
    <input type="submit" value="Yes" />
    <a href="/">Cancel </a>
  </form>
</div>

```

Step 5: Add path in urls.py files in App.

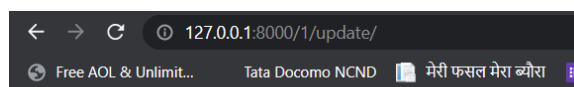
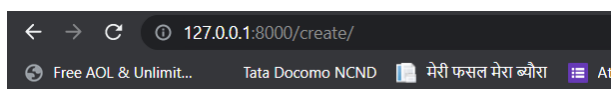
```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.indexPage),
    path('home/', views.homePage),
    path('about/', views.aboutPage),
    path('index/', views.indexPage),
    path('create/', views.create_view),
    path('<id>/update/', views.update_view),
    path('list/', views.list_view),
    path('<id>/delete/', views.delete_view),
]

```

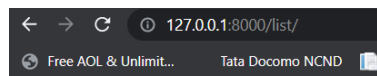
MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to generate forms from existing models and use those forms to perform CRUD operations on data.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-08		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 04 OF 04

OUTPUT:-

Name: Name: Title: Title:

Description:

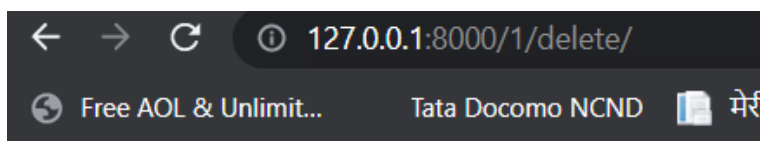
Description:



. name
create view
test

. name
create view2
test2

. name
title 1
test1



Are you want to delete this item ? [Cancel](#)

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to send simple emails to the registered users of your Django application.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-09	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 01 OF 02

Aim: Develop program to send simple emails to the registered users of your Django application.

Step 1: In settings.py, add the following code.

```
# Email-setup
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.mail.yahoo.com'
EMAIL_USE_SSL = True
EMAIL_PORT = 465
EMAIL_HOST_USER = 'satyamverma@yahoo.com'
EMAIL_HOST_PASSWORD = '*****'

# Application definition
```

Step 2: In views.py, add the following code.

```
1 from django.shortcuts import render
2 from URLconf.settings import EMAIL_HOST_USER
3 from . import forms
4 from django.core.mail import send_mail
5
6 def feedback(request):
7     sub = forms.Feedback()
8     if request.method == 'POST':
9         sub = forms.Feedback(request.POST)
10        subject = 'Welcome to Rucler Technologies'
11        message = 'Good Efforts and Hope you are enjoying your Work'
12        receipient = str(sub['Email'].value())
13        send_mail(subject,
14                message, EMAIL_HOST_USER, [receipient], fail_silently = False)
15        return render(request, 'success.html', {'receipient': receipient})
16    return render(request, 'index.html', {'form':sub})
17
```


MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Develop program to send simple emails to the registered users of your Django application.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-09		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		SEMESTER: V
		PAGE: 02 OF 02

Step 3: Send the email.

Feedback Form

Customer Feedback Details

Email:

Submit

OUTPUT:

email sent successfully to: {{ receipient }}

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to integrate Django with Reactjs using Django REST Framework.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-10		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 01 OF 05

Aim: Demonstrate how to integrate Django with Reactjs using Django REST Framework.

Step 1:- Install pipenv using pip command.

```

C:\Users\yashp>pip install pipenv
Requirement already satisfied: pipenv in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (2021.5.29)
Requirement already satisfied: virtualenv-clone>=0.2.5 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from pipenv) (0.5.4)
Requirement already satisfied: virtualenv in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from pipenv) (20.4.7)
Requirement already satisfied: pip>=18.0 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from pipenv) (21.1.2)
Requirement already satisfied: certifi in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from pipenv) (2021.5.30)
Requirement already satisfied: setuptools>=36.2.1 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from pipenv) (56.0.0)
Requirement already satisfied: appdirs<2,>=1.4.3 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from virtualenv->pipenv) (1.4.4)
Requirement already satisfied: six<2,>=1.9.0 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from virtualenv->pipenv) (1.16.0)
Requirement already satisfied: distlib<1,>=0.3.1 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from virtualenv->pipenv) (0.3.2)
Requirement already satisfied: filelock<4,>=3.0.0 in c:\users\yashp\appdata\local\programs\python\python39\lib\site-packages (from virtualenv->pipenv) (3.0.12)
C:\Users\yashp>

```

Step 2:- Create model(in models.py) to connect with database and insert table.

```

DRProject > todoApp > models.py > {} models
1  from django.db import models
2
3  # Create your models here.
4  class Todo(models.Model):
5      title = models.CharField(max_length=100)
6      description = models.TextField()
7      completed = models.BooleanField(default=False)
8
9      def __str__(self):
10         return self.title

```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to integrate Django with Reactjs using Django REST Framework.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-10	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 02 OF 05

Step 3:- Putting in the APIs

```

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     'todoApp',
41     'corsheaders',
42     'rest_framework',
43 ]
44
45 MIDDLEWARE = [
46     'django.middleware.security.SecurityMiddleware',
47     'django.contrib.sessions.middleware.SessionMiddleware',
48     'django.middleware.common.CommonMiddleware',
49     'django.middleware.csrf.CsrfViewMiddleware',
50     'django.contrib.auth.middleware.AuthenticationMiddleware',
51     'django.contrib.messages.middleware.MessageMiddleware',
52     'django.middleware.clickjacking.XFrameOptionsMiddleware',
53     'corsheaders.middleware.corsMiddleware',
54 ]
55

```

Step-4:- Create class (in views.py) which perform the CRUD operations

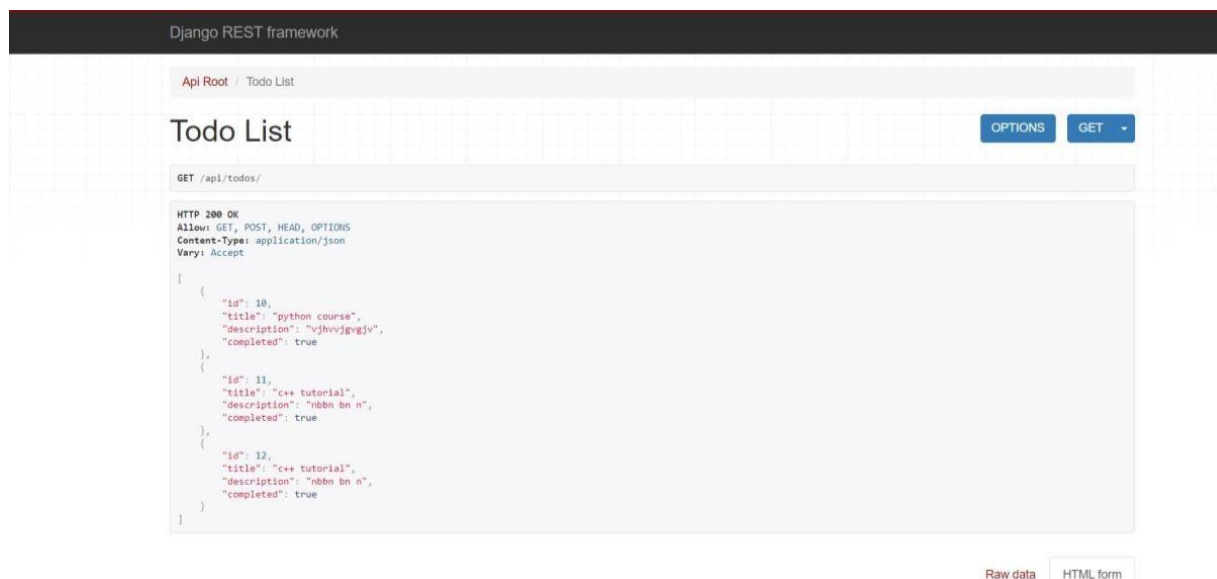
```

1 from django.shortcuts import render
2 from .serializers import TodoSerializer
3 from rest_framework import viewsets
4 from .models import Todo
5
6 class TodoView(viewsets.ModelViewSet):
7     serializer_class = TodoSerializer
8     queryset = Todo.objects.all()

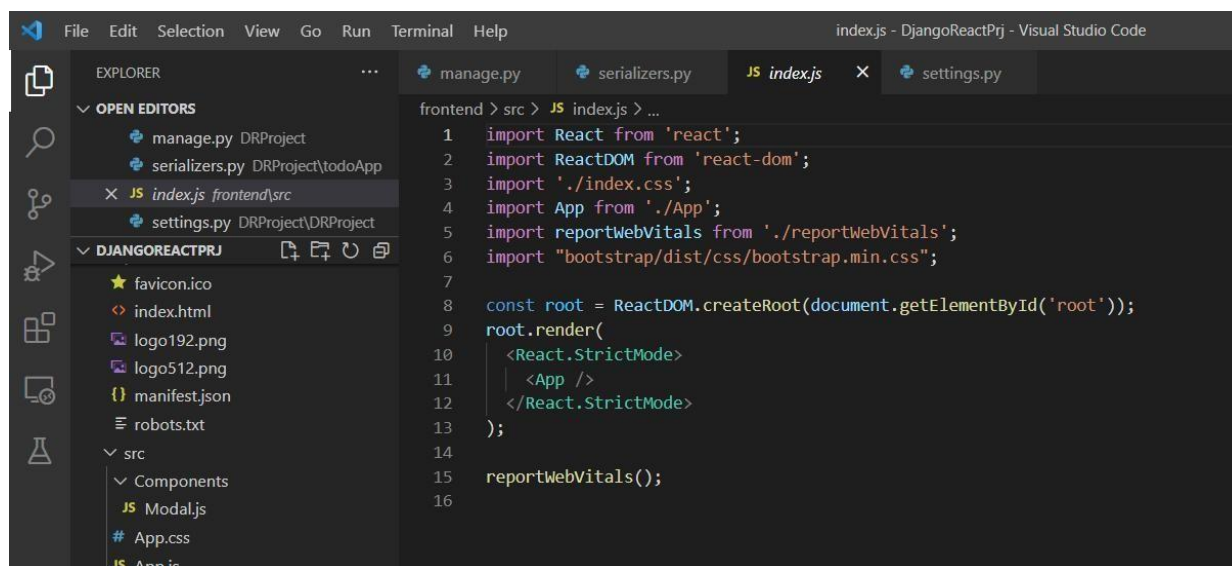
```

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to integrate Django with Reactjs using Django REST Framework.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-10	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 03 OF 05

Step-5:-Performing CRUD operations to our items is enabled by the router.



Step-6:- Frontend using React.



MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to integrate Django with Reactjs using Django REST Framework.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-10	ISSUE NO.: 01	ISSUE DATE: 04/07/2022
REV. DATE: NA	REV. NO.: 00	DEPTT.: CSE
LABORATORY: Python & its Framework Lab	SEMESTER: V	PAGE: 04 OF 05

Step-7:-Substitute the below code in src/App.js.

MMEC	M. M. ENGINEERING COLLEGE – MULLANA	LABORATORY MANUAL
	PRACTICAL EXPERIMENT INSTRUCTION SHEET	
	EXPERIMENT TITLE: Demonstrate how to integrate Django with Reactjs using Django REST Framework.	
EXPERIMENT NO.: MMEC-WI-BCSE-533L-10		ISSUE NO.: 01
REV. DATE: NA		ISSUE DATE: 04/07/2022
LABORATORY: Python & its Framework Lab		REV. NO.: 00
		DEPTT.: CSE
		PAGE: 05 OF 05

Step-8:-Your application should be like the one below at this point:

The screenshot shows a web interface for a REST API. At the top right, there are two tabs: 'Raw data' (highlighted in red) and 'HTML form'. Below the tabs is a form with three main sections: 'Title' with a text input field, 'Description' with a larger text area, and 'Completed' with a checkbox. At the bottom right of the form is a blue button labeled 'POST'.