

Experiment No. 08

| | |
|-----------------------------|---|
| Semester | S.E. Semester IV – Computer Engineering |
| Subject | Skill Base Lab Course: Python Programming(CSL405) |
| Subject Professor In-charge | Prof. Swapnil S. Sonawane |
| Assisting Teachers | Prof. Swapnil S. Sonawane |
| Student Name | HERAMBA LIMAYE |
| Roll Number | 19102A0033 |

Title:

Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression).

Objective:

To explore Django web framework for developing python-based web application

Explanation:

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience.

Advantages of Django

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

- **Object-Relational Mapping (ORM) Support** – 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 Support** – Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.
- **Framework Support** – 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.

Create a Project

Whether you are on Windows or Linux, just get a terminal or a **cmd** prompt and navigate to the place you want your project to be created, then use this code –

```
django-admin startproject myproject
```

This will create a "myproject" folder with the following structure –

```
myproject/
manage.py
myproject/
__init__.py
settings.py
urls.py
wsgi.py
```

```
python manage.py help
```

- **The “myproject” subfolder** – This folder is the actual python package of your project. It contains four files –
 - **__init__.py** – Just for python, treat this folder as package.
 - **settings.py** – As the name indicates, your project settings.
 - **urls.py** – All links of your project and the function to call. A kind of ToC of your project.
 - **wsgi.py** – If you need to deploy your project over WSGI.

Program Code:

aa.html :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body bgcolor="purple">
```

```

<form method="get" action="login">
    <b>Username</b>
    <input type="text" name="a" value="" size="10">
    <br>
    <b>Password</b>
    <input type="password" name="b" value="" size="10">
    <br>
    <b>Confirm Password</b>
    <input type="text" name="c" value="" size="10">
    <br>
    <input type="submit" value="LOGIN">
</form>
</body>
</html>

```

a1\urls.py :-

```

from django.contrib import admin
from django.urls import path
from . import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("",views.f1,name='homepage'),
    path('profile',views.f2,name='profile page'),
    path('login',views.f3,name="Login page")
]

```

VIT2\urls.py :-

```

from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("",include('a1.urls')),
    path('profile',include('a1.urls'))
]

```

Views.py :-

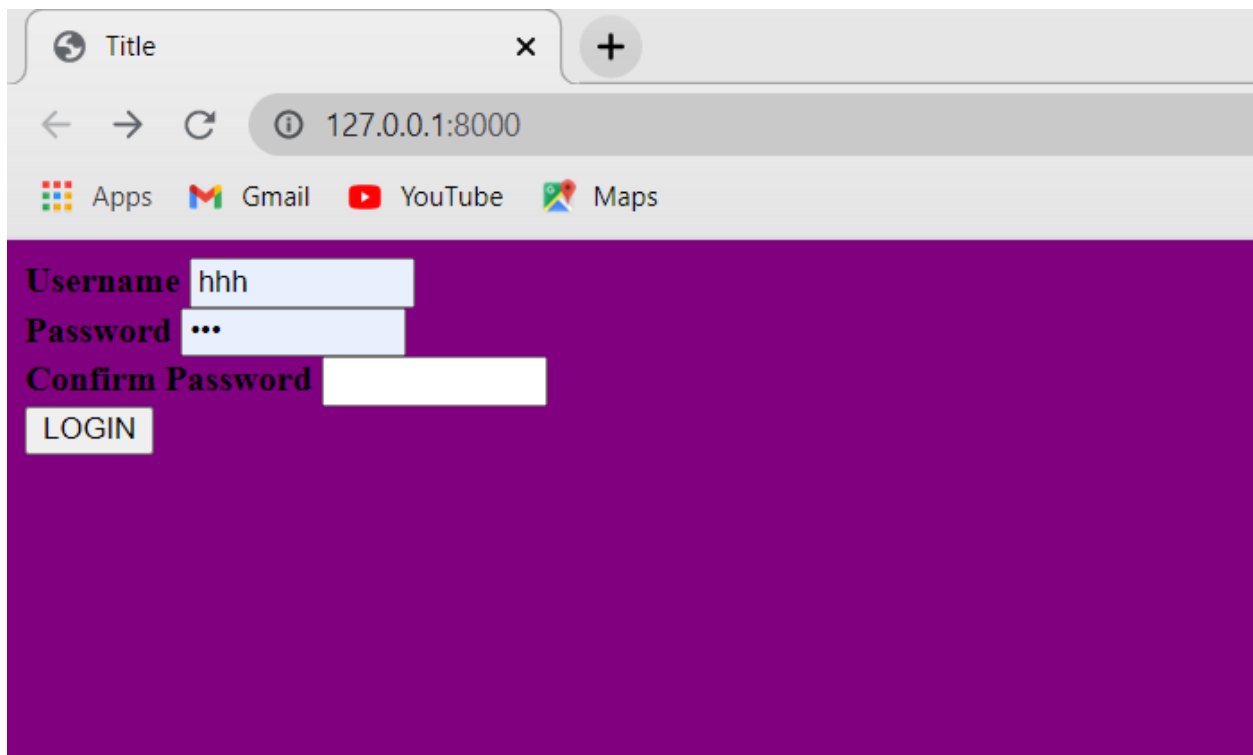
```

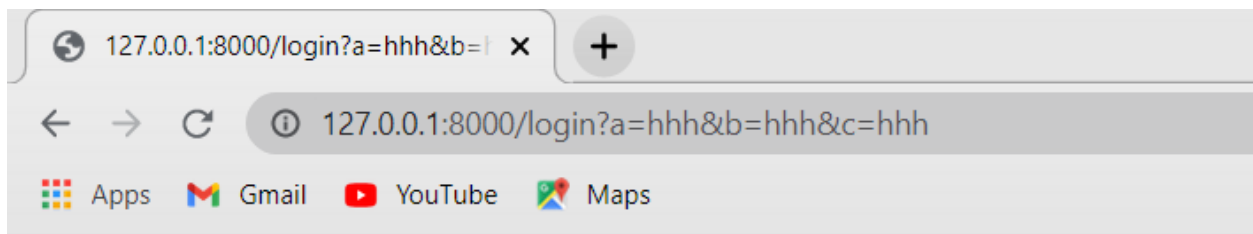
from django.shortcuts import render
from django.http import HttpResponse

```

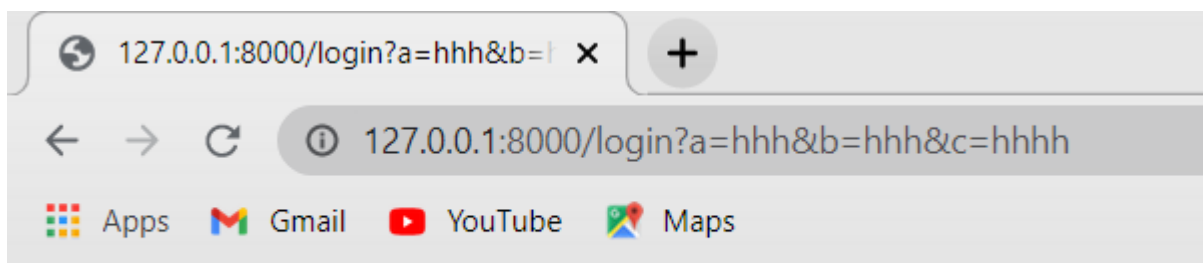
```
def f1(request):  
    return render(request,'aa.html')  
def f2(request):  
    return HttpResponse('This is VIT profile page...')  
def f3(request):  
    s1=request.GET['b']  
    s2=request.GET['c']  
    if s1==s2:  
        return HttpResponse('Valid User')  
    else:  
        return HttpResponse('Invalid User')
```

Output:





Valid User



Invalid User

Conclusion:

Through this experiment we learnt to create web application using Django web framework