

User\_project all API ENDPOINTS  
register new user 🖱️ http://127.0.0.1:8000  
login user 🖱️ http://127.0.0.1:8000/login  
logout user 🖱️ http://127.0.0.1:8000/logout  
create city record 🖱️ http://127.0.0.1:8000/create  
update city record 🖱️ http://127.0.0.1:8000/update/record\_number/  
delete city record 🖱️ http://127.0.0.1:8000/delete/record\_number/

## Register New User

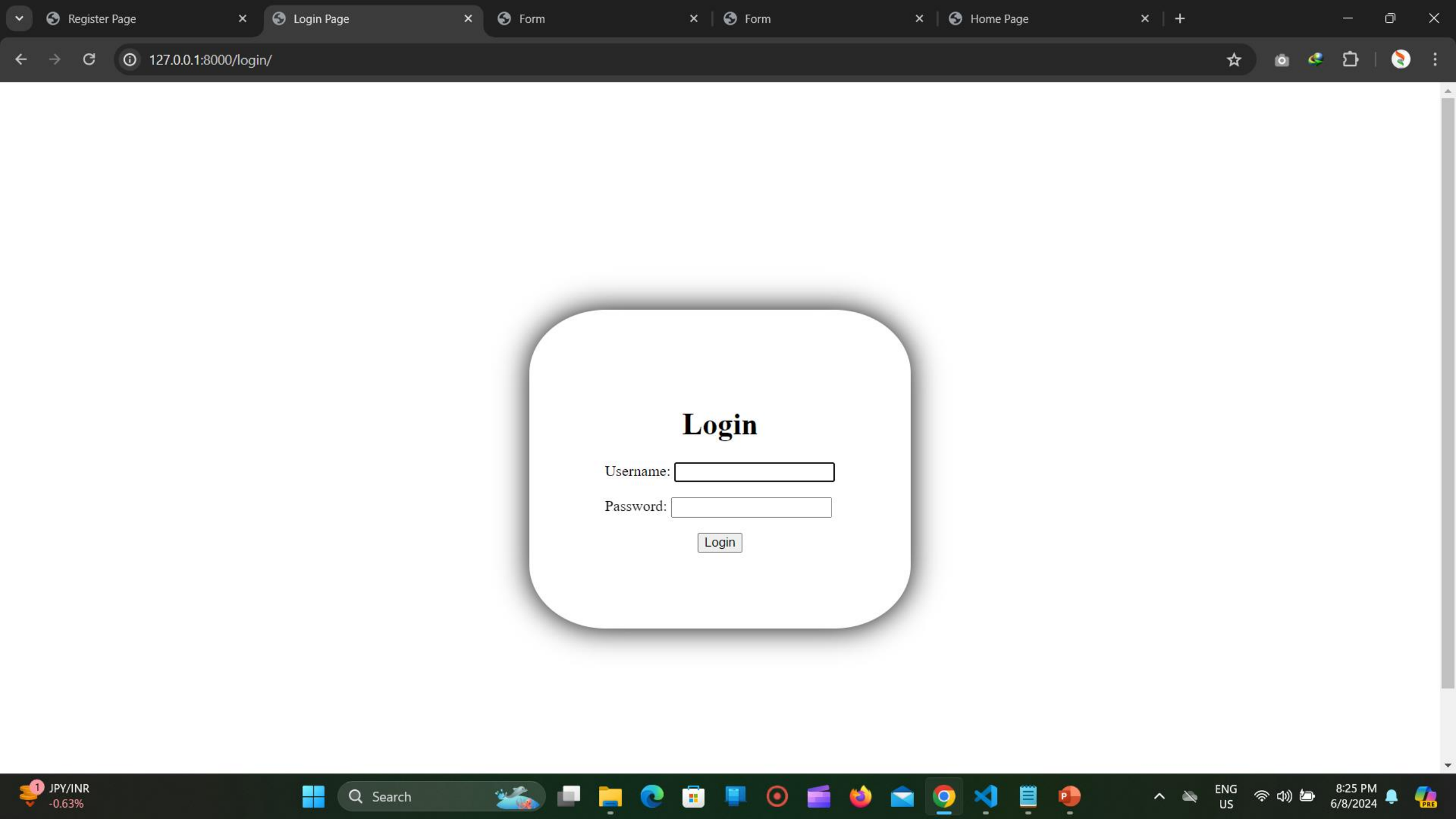
Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

submit

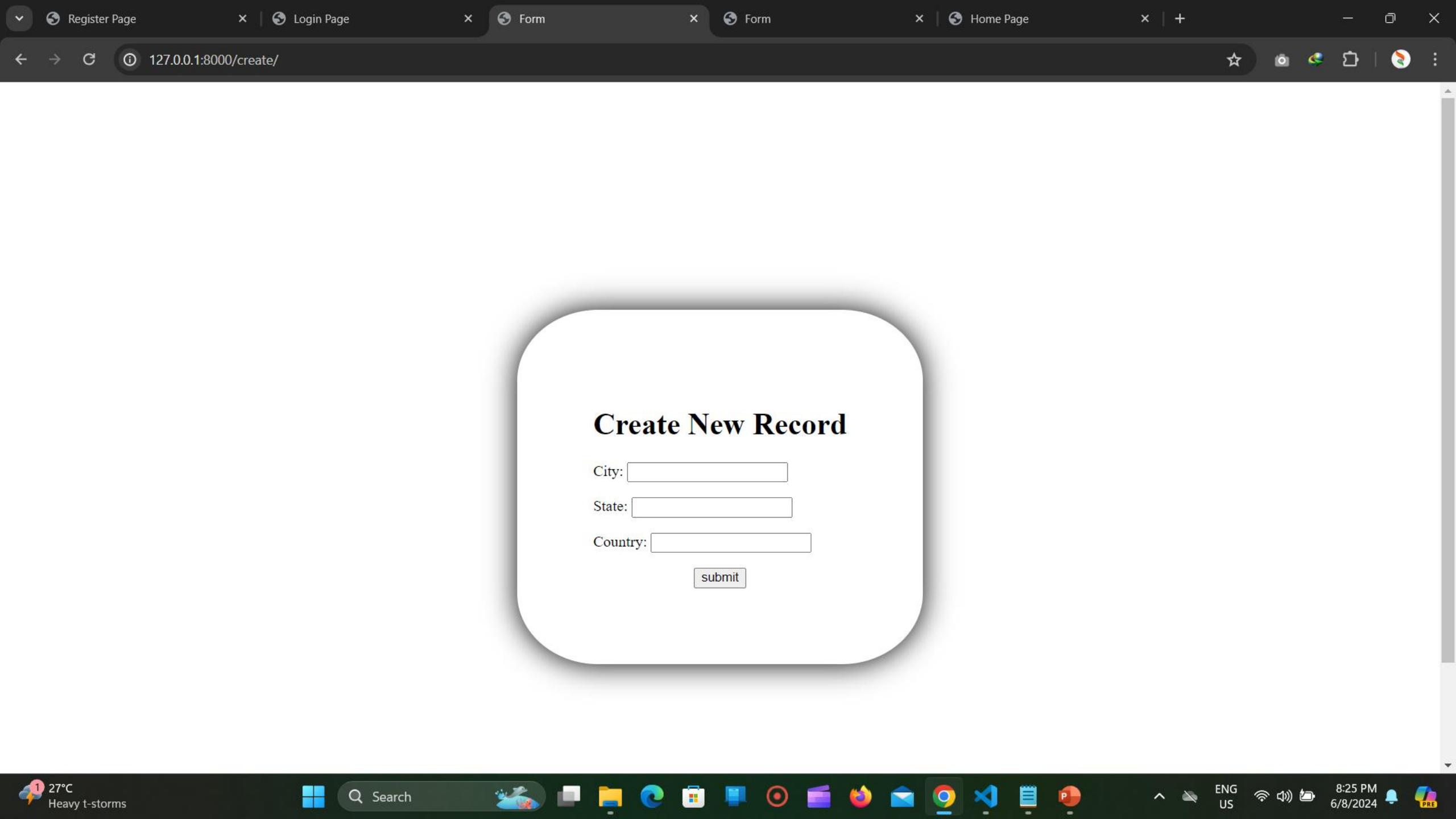


# Login

Username:

Password:

Login



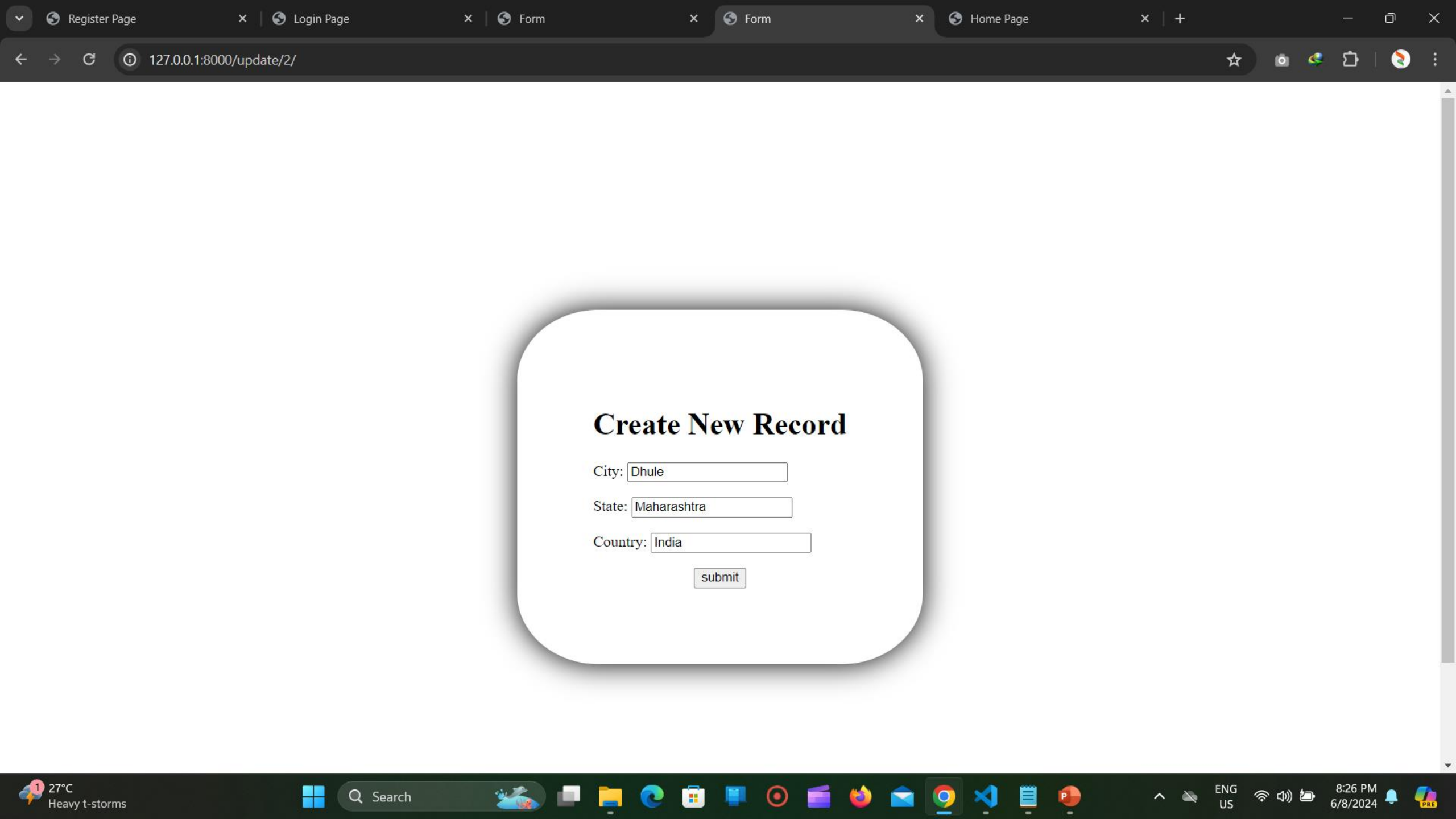
## Create New Record

City:

State:

Country:

submit

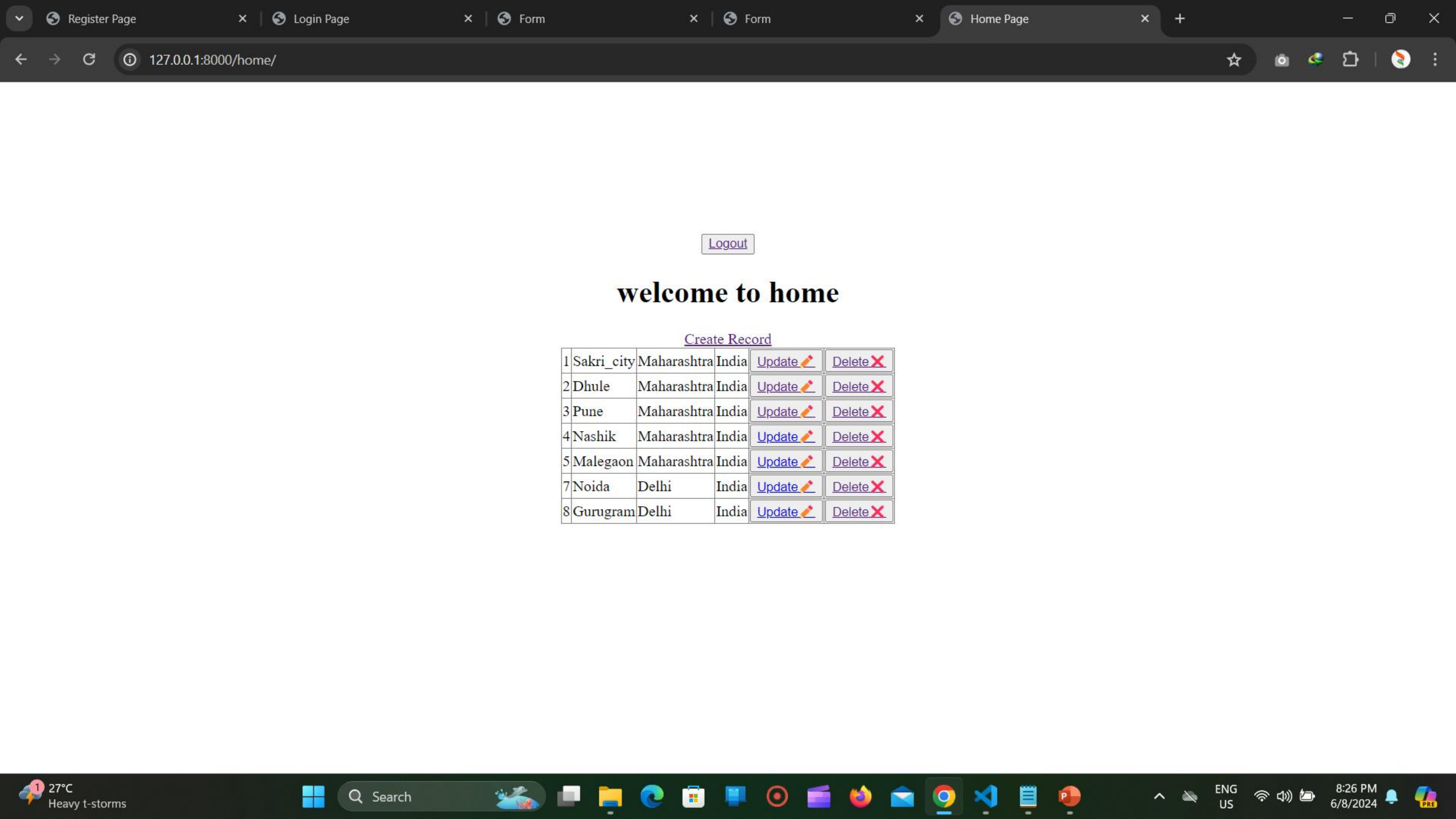


## Create New Record

City:

State:

Country:



[Logout](#)

welcome to home

[Create Record](#)

1	Sakri_city	Maharashtra	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
2	Dhule	Maharashtra	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
3	Pune	Maharashtra	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
4	Nashik	Maharashtra	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
5	Malegaon	Maharashtra	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
7	Noida	Delhi	India	<a href="#">Update</a> 	<a href="#">Delete</a> 
8	Gurugram	Delhi	India	<a href="#">Update</a> 	<a href="#">Delete</a> 

EXPLORER

THURSDAY

user\_project

- media
- templates
- user\_app
  - \_\_pycache\_\_
  - migrations
  - templates
    - form.html
    - home.html
    - login.html
    - register.html

- \_\_init\_\_.py
- admin.py
- apps.py
- forms.py
- models.py
- tests.py
- urls.py
- views.py

user\_project

- \_\_pycache\_\_
- \_\_init\_\_.py
- asgi.py
- settings.py
- urls.py
- wsgi.py
- db.sqlite3
- manage.py

settings.py

urls.py ...\user\_project

urls.py ...\user\_app

views.py

register.html X

user\_project > user\_app > templates > register.html > html > head > style > body

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Register Page</title>
7   <style>
8     form{
9       margin: 5rem;
10      padding: 5rem;
11      align-items: center;
12      box-shadow: 0 0 30px;
13      border-radius: 20%;
14    }
15    body{
16      display: flex;
17      flex-direction: column;
18      align-items: center;
19      margin: 1.2rem;
20      padding: 1.2rem;
21    }
22  </style>
23 </head>
24 <body>
25   <code>
26   <center>User_project all API ENDPOINTS</center>
27   register new user 🚀 http://127.0.0.1:8000<br/>
28   login user 🚀 http://127.0.0.1:8000/login<br/>
29   logout user 🚀 http://127.0.0.1:8000/logout<br/>
30   create city record 🚀 http://127.0.0.1:8000/create<br/>
31   update city record 🚀 http://127.0.0.1:8000/update/record\_number/ <br/>
32   delete city record 🚀 http://127.0.0.1:8000/delete/record\_number/ <br/>
33   </code>
34   <form method="POST" enctype="multipart/form-data">
35     <center><h1>Register New User</h1></center>
36     {% csrf_token %}
37     {{form.as_p}}
38     <center><button type="submit">submit</button></center>
39   </form>
40 </body>
41 </html>
```

EXPLORER

THURSDAY

user\_project

media

templates

user\_app

> \_\_pycache\_\_

> migrations

templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

user\_project

> \_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

settings.py

urls.py ...\user\_project

urls.py ...\user\_app

views.py

register.html

user\_project > user\_app > templates > login.html > ...

1

<!DOCTYPE html>

2

<html lang="en">

3

<head>

4

<meta charset="UTF-8">

5

<meta name="viewport" content="width=device-width, initial-scale=1.0">

6

<title>Login Page</title>

7

<style>

8

form{

9

margin: 5rem;

10

padding: 5rem;

11

align-items: center;

12

box-shadow: 0 0 30px;

13

border-radius: 20%;

14

}

15

body{

16

display: flex;

17

flex-direction: column;

18

align-items: center;

19

margin: 5rem;

20

padding: 5rem;

21

}

22

</style>

23

</head>

24

<body>

25

<form method="POST" enctype="multipart/form-data">

26

<center><h1>Login</h1></center>

27

{% csrf\_token %}

28

{{ form.as\_p }}

29

<center><button type="submit">Login</button></center>

30

</form>

31

</body>

32

</html>

EXPLORER... settings.py urls.py ...\user\_project urls.py ...\user\_app views.py register.html home.html X

THURSDAY

user\_project

media

templates

user\_app

> \_\_pycache\_\_

> migrations

templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

user\_project

> \_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

user\_project > user\_app > templates > home.html > html

```
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Home Page</title>
7     <style>
8         table{
9             border-collapse: collapse;
10        }
11        body{
12            display: flex;
13            flex-direction: column;
14            align-items: center;
15            margin: 5rem;
16            padding: 5rem;
17        }
18    </style>
19 </head>
20 <body>
21     <button><a href="{% url 'logout' %}">Logout</a></button>
22     <h1>welcome to home </h1>
23     <center><a href="{% url 'create_city' %}">Create Record</a></center>
24     <table border="1">
25         {% for x in data %}
26         <tr>
27             <td>{{x.id}}</td><td>{{x.city}}</td><td>{{x.state}}</td><td>{{x.country}}</td>
28             <td><button><a href="{% url 'update' x.id %}">Update</a></button></td>
29             <td><button><a href="{% url 'delete' x.id %}">Delete</a></button></td>
30         </tr>
31         {% endfor %}
32     </table>
33 </body>
34 </html>
```



EXPLORER

THURSDAY

- user\_project
  - media
  - templates
  - user\_app
    - \_\_pycache\_\_
    - migrations
    - templates
      - form.html
      - home.html
      - login.html
      - register.html
- \_\_init\_\_.py
- admin.py
- apps.py
- forms.py
- models.py
- tests.py
- urls.py
- views.py
- user\_project
  - \_\_pycache\_\_
  - \_\_init\_\_.py
  - asgi.py
  - settings.py
  - urls.py
  - wsgi.py
- db.sqlite3
- manage.py

settings.py | urls.py ...\user\_project | urls.py ...\user\_app | views.py | register.html

user\_project > user\_app > templates > form.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Form </title>
7     <style>
8         form{
9             margin: 5rem;
10            padding: 5rem;
11            align-items: center;
12            box-shadow: 0 0 30px;
13            border-radius: 20%;
14        }
15        body{
16            display: flex;
17            flex-direction: column;
18            align-items: center;
19            margin: 5rem;
20            padding: 5rem;
21        }
22    </style>
23 </head>
24 <body>
25     <form method="POST" enctype="multipart/form-data">
26         <center><h1>Create New Record</h1></center>
27         {% csrf_token %}
28         {{form.as_p}}
29         <center><button type="submit">submit</button></center>
30     </form>
31 </body>
32 </html>
```

EXPLORER

THURSDAY

user\_project

media

templates

user\_app

\_\_pycache\_\_

migrations

templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

user\_project

\_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

settings.py

urls.py ...\user\_project

urls.py ...\user\_app

user\_project > user\_app > models.py > cityModel > \_\_str\_\_

1

2

3

4

5

6

7

8

9

10

from django.db import models

class cityModel(models.Model):

id = models.AutoField(primary\_key=True)

city = models.CharField(max\_length=500)

state = models.CharField(max\_length=500)

country = models.CharField(max\_length=500)

def \_\_str\_\_(self) -> str:

return super().\_\_str\_\_()

EXPLORER

THURSDAY

user\_project

media

templates

user\_app

\_\_pycache\_\_

migrations

templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

user\_project

\_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

settings.py

urls.py ...\user\_project

urls.py ...\user\_app

user\_project > user\_app > forms.py > cityForm > Meta

1

2

3

4

5

6

7

from django import forms

from .models import cityModel

class cityForm(forms.ModelForm):

class Meta:

model = cityModel

fields = "\_\_all\_\_"

## EXPLORER



settings.py

urls.py ...\user\_project

urls.py ...\user\_app X

views.py

## THURSDAY

## user\_project

## media

## templates

## user\_app

&gt; \_\_pycache\_\_

&gt; migrations

## templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

## user\_project

&gt; \_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

user\_project &gt; user\_app &gt; urls.py &gt; ...

```
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path("", views.register_user, name="register"),
6      path("home/", views.home, name="home"),
7      path("login/", views.login_user, name="login"),
8      path("logout/", views.logout_user, name="logout"),
9      path("create/", views.create_city, name="create_city"),
10     path("update/<int:id>/", views.update_city, name="update"),
11     path("delete/<int:id>/", views.delete_city, name="delete")
12 ]
```

EXPLORER

THURSDAY

user\_project

media

templates

user\_app

> \_\_pycache\_\_

migrations

templates

form.html

home.html

login.html

register.html

\_\_init\_\_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

user\_project

> \_\_pycache\_\_

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

db.sqlite3

manage.py

settings.py

urls.py

user\_project

urls.py

user\_app

views.py

register.html

home.html

form.html

forms.py

apps.py

models.py

```
user_project > user_app > views.py > delete_city
1  from django.shortcuts import get_object_or_404, redirect, render
2  from django.contrib.auth.forms import UserCreationForm, AuthenticationForm
3  from django.contrib.auth import login, logout
4  from django.contrib.auth.models import User
5  from django.contrib.auth.decorators import login_required
6  from .models import cityModel
7  from .forms import cityForm
8  # Create your views here.
9
10 # register user
11 def register_user(request):
12     context = {} # empty dictionary
13     if request.method == "POST": # request if POST
14         form = UserCreationForm(request.POST) # create UserCreation form using request.POST data
15         if form.is_valid(): # check if form is valid
16             form.save() # save the form data into user model
17             return redirect("/login") # redirect to login page
18     else:
19         context["form"] = UserCreationForm() # if request is "GET" then create empty UserCreation form
20     return render(request, "register.html", context) # render register.html with empty form data
21
22 @login_required # for forcing login before view the home page or @login_required(login_url='/custom/login/', redirect_field_name='redirect_to')
23 def home(request):
24     obj = cityModel.objects.all()
25     return render(request, "home.html", {"data": obj})
26
27 # create model
28 def create_city(request):
29     context = {}
30     if request.method == "POST":
31         form = cityForm(request.POST or None)
32         if form.is_valid():
33             form.save()
34             return redirect("/home")
35     else:
36         context["form"] = cityForm()
37     return render(request, "form.html", context)
38
39 def update_city(request, id):
40     context = {}
41     instance = get_object_or_404(cityModel, id=id)
42     form = cityForm(instance=instance, data=request.POST)
43     if request.method == "POST":
44         if form.is_valid():
45             form.save()
46             return redirect("/home")
47     else:
48         context["form"] = cityForm(instance=instance)
49     return render(request, "form.html", context)
50
51 def delete_city(request, id):
52     instance = get_object_or_404(cityModel, id=id)
53     if instance is not None:
54         instance.delete()
55         return redirect("/home")
56
57
58
59 # login user
60 def login_user(request):
61     context = {} # create context empty dictionary
62     if request.method == "POST": # if request of "POST" type
63         form = AuthenticationForm(request, data=request.POST) # create AuthenticationForm with request data
64         if form.is_valid(): # check if form is valid
65             user = form.get_user() # recall user data from form
66             login(request, user) # login the user
67             return redirect("/home") # redirect to home page after successfully login
68     else:
69         # if request.method == "GET" then create new empty form
70         context["form"] = AuthenticationForm() # create empty AuthenticationForm()
71     return render(request, "login.html", context) # render empty login.html page with context data
72
73 # logout user
74 def logout_user(request):
75     logout(request) # logout with request data
76     return redirect("/login") # redirect to login page
77
```

## THURSDAY

## user\_project

## media

## templates

## user\_app

## &gt; \_\_pycache\_\_

## &gt; migrations

## templates

## &lt;&gt; form.html

## &lt;&gt; home.html

## &lt;&gt; login.html

## &lt;&gt; register.html

## \_\_init\_\_.py

## admin.py

## apps.py

## forms.py

## models.py

## tests.py

## urls.py

## views.py

## user\_project

## &gt; \_\_pycache\_\_

## \_\_init\_\_.py

## asgi.py

## settings.py

## urls.py

## wsgi.py

## db.sqlite3

## manage.py

user\_project &gt; user\_project &gt; urls.py &gt; ...

```
1
2 from django.contrib import admin
3 from django.urls import path,include
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path("",include("user_app.urls"))
8 ]
9 |
```

settings.py

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

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'user_project.urls'

LOGIN_REDIRECT_URL = "/"
LOGOUT_REDIRECT_URL = "/"
LOGIN_URL = "/login/" # to use @login_required decorator on any view

STATIC_URL = '/static/'
MEDIA_URL = "/media/"
MEDIA_ROOT = BASE_DIR / "media"
```