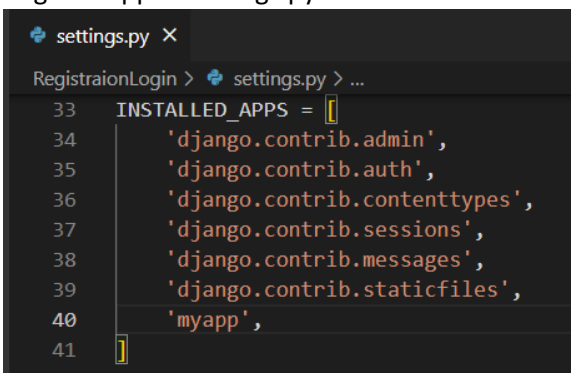
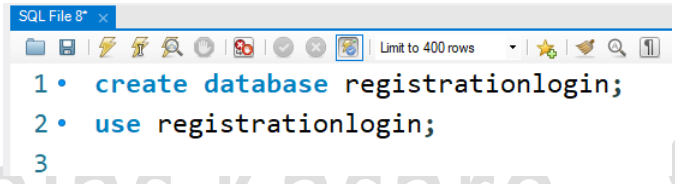
1. Create project - RegistraionLogin and Create app – myapp in it

```
\my notes\Django>django-admin startproject RegistraionLogin

\my notes\Django>cd RegistraionLogin

\my notes\Django\RegistraionLogin>python manage.py startapp myapp

\my notes\Django\RegistraionLogin>
```

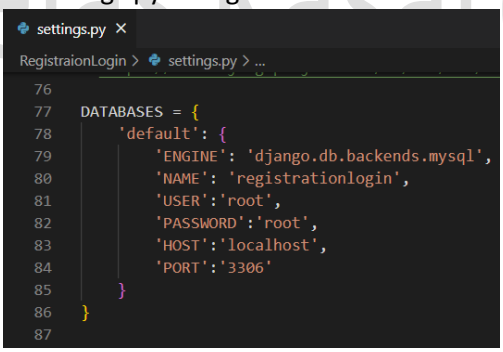
2. Register app in settings.py



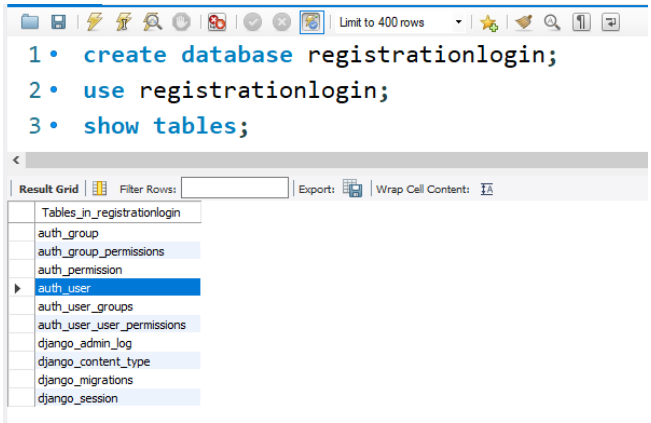
3. Create RegistraionLogin database in workbench

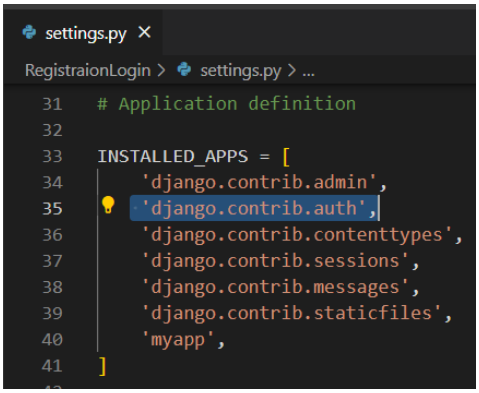


4. In settings.py change database details into mysql



5. makemirations
6. migrate
7. check tables in workbench
   Where auth is name of application and user is name of model present inside auth application





8. let's describe auth_user table to get its stricture

```
4 •  desc auth_user;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
password	varchar(128)	NO		NULL	
last_login	datetime(6)	YES		NULL	
is_superuser	tinyint(1)	NO		NULL	
username	varchar(150)	NO	UNI	NULL	
first_name	varchar(150)	NO		NULL	
last_name	varchar(150)	NO		NULL	
email	varchar(254)	NO		NULL	
is_staff	tinyint(1)	NO		NULL	
is_active	tinyint(1)	NO		NULL	
date_joined	datetime(6)	NO		NULL	

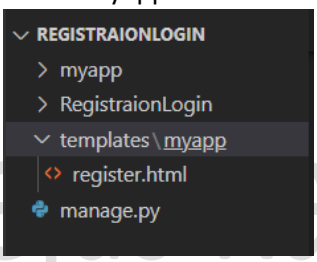
9. create templates folder, register it in settings.py

settings.py ×

RegistraionLogin > settings.py > ...

```
55  TEMPLATES = [
56      {
57          'BACKEND': 'django.template.backend
58          'DIRS': [BASE_DIR/'templates'],
59          'APP_DIRS': True,
```

10. create myapp folder in templates and create register.html file



11. code for register.html

```
<body>
    <table align="center" border="1" cellpadding="5" cellspacing="5">
        <form method="POST">
            {% csrf_token %}
        <thead>
            <tr>
                <th colspan="2">Registration Form</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td><label for="username">UserName</label></td>
                <td><input type="text" id="username" name="username" value=""></td>
            </tr>
            <tr>
                <td><label for="password">Password</label></td>
                <td><input type="password" id="password" name="password" value=""></td>
            </tr>
            <tr>
                <td><label for="password2">Confirm Password</label></td>
                <td><input type="password" id="password2" name="password2" value=""></td>
            </tr>
            <tr>
                <td><input type="reset"></td>
                <td><input type="submit"></td>
            </tr>
        </tbody>
    </form>
    </table>
</body>
```

12. create view to show registration form

views.py

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

# Create your views here.
def register_user(request):
    return render(request,'myapp/register.html')
```

13. create url pattern for above view at application level (create myapp_urls.py in myapp folder)

myapp_urls.py

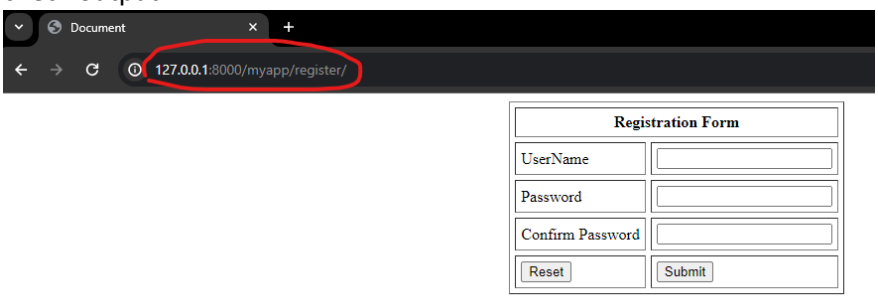
```python
from django.urls import path
from myapp import views

urlpatterns = [
    path('register/', views.register_user),
]
```

14. create url for myapp -> myapp_url.py in urls.py (at project level)

urls.py

```python
"""
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('myapp/', include('myapp.myapp_urls')),
]
```


15. runserver
16. check output



17. create superuser

```
C:\TEJAS KASARE (Very imp folder)\my notes\Django\RegistraionLogin>python manage.py createsuperuser
Username (leave blank to use 'admin'): tejas
Email address: tejas@gmail.com
Password:
Password (again):
Superuser created successfully.
```

18. check superuser details in auth_user table

```
5 •  select * from auth_user;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined
1	pbkdf2_sha256$600000$OLLuzQsPY0ntVDVqQ...	NULL	1	tejas			tejas@gmail.com	1	1	2024-01-22 09:31:40.159219
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

19. collect user data from registration form and insert into table – auth_user (update view logic)

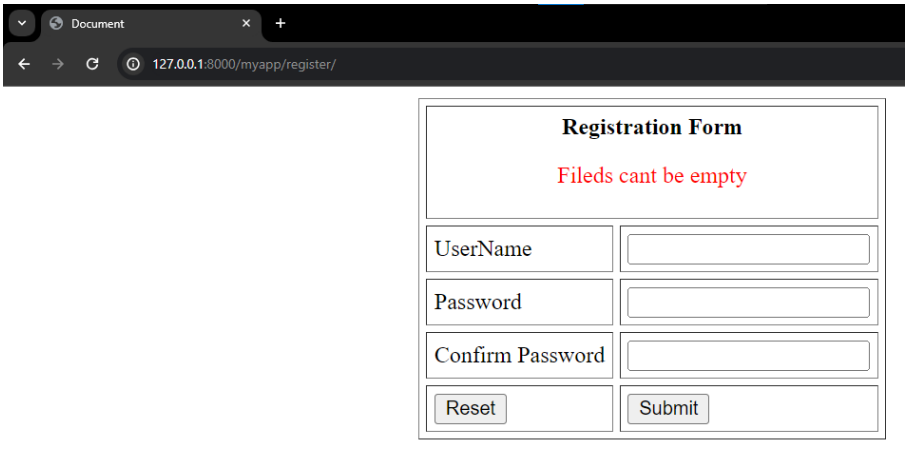
views.py

```python
from django.contrib.auth.models import User
# Create your views here.
def register_user(request):
    data={}
    if request.method=="POST":
        uname=request.POST['username']
        upass=request.POST['password']
        uconf_pass=request.POST['password2']
        #implementing validation
        if (uname=='' or upass =='' or uconf_pass ==''):
            data['error_msg']='Fileds cant be empty'
            return render(request,'myapp/register.html',context=data)
        elif(upass!=uconf_pass):
            data['error_msg']='Password and confirm password does not matched'
            return render(request,'myapp/register.html',context=data)
        elif(User.objects.filter(username=uname).exists()):
            data['error_msg']=uname + ' alredy exist'
            return render(request,'myapp/register.html',context=data)
        else:
            user=User.objects.create(username=uname)
            #here username and password aee column names present inside auth_user table
            user.set_password(upass) #encrypting passowrd
            user.save() #saving data into table
            return HttpResponse("Registraion done")
    return render(request,'myapp/register.html')
```


20. update register.html to show error

```html
<body>
    <table align="center" border="1" cellpadding="5" cellspacing="5">
        <form method="POST">
            {% csrf_token %}
        <thead>
            <tr>
                <th colspan="2">
                    Registration Form
                    {% if error_msg %}
                    <p style="color: red; font-weight: lighter;">{{error_msg}}</p>
                    {% endif %}
                </th>

            </tr>
        </thead>
```

21. output



22. login functionality
    a. create login.html in templates > myapp folder

login.html

templates > myapp > login.html

```
<body>
    <table align="center" border="1" cellpadding="5" cellspacing="5">
        <form method="POST">
            {% csrf_token %}
        <thead>
            <tr>
                <th colspan="2">
                    Login Form
                    {% if error_msg %}
                    <p style="color: red; font-weight: lighter;">{{error_msg}}</p>
                    {% endif %}
                </th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td><label for="username">UserName</label></td>
                <td><input type="text" id="username" name="username" value=""></td>
            </tr>
            <tr>
                <td><label for="password">Password</label></td>
                <td><input type="password" id="password" name="password" value=""></td>
            </tr>
            <tr>
                <td><input type="reset"></td>
                <td><input type="submit"></td>
            </tr>
        </tbody>
    </form>
    </table>
</body>
```

    b. create view to view login form

```
def login_user(request):
    return render(request,'myapp/login.html')

```

    c. create url for above view (in application level – myapp_urls.py)

myapp_urls.py

myapp > myapp_urls.py > ...

```
from django.urls import path
from myapp import views

urlpatterns = [
    path('register/', views.register_user),
    path('login/', views.login_user),
]

```

d. logic to perform login operation

```
33  def login_user(request):
34      data={}
35      if request.method=="POST":
36          uname=request.POST['username']
37          upass=request.POST['password']
38          #implementing validation
39          if (uname=='' or upass ==''):
40              data['error_msg']='Fileds cant be empty'
41              return render(request,'myapp/login.html',context=data)
42          elif(not User.objects.filter(username=uname).exists()):
43              data['error_msg']=uname + ' user is not registered'
44              return render(request,'myapp/login.html',context=data)
45          else:
46              #from django.contrib.auth import authenticate
47              user=authenticate(username=uname,password=upass)
48              print(user)
49              if user is not None:
50                  return redirect('/myapp/home')
51              else:
52                  data['error_msg']='Wrong Password'
53                  return render(request,'myapp/login.html',context=data)
54      return render(request,'myapp/login.html')
```

There are 2 important line

1 – authenticate() function – accepts username and password and return User object if user exist with given credentials otherwise None(line 47 above)

2- on successful login, we are redirecting to home page. So we need to create home.html file, view to display home.html and finally url for view.

e. Creating home.html in templates > myapp folder

home.html

templates > myapp > home.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Document</title>
5  </head>
6  <body>
7      <h1>Welcome to home</h1>
8  </body>
9  </html>
```

f. Creating view to display home.html

```
55
56  def home(request):
57      return render(request,'myapp/home.html')
58
```

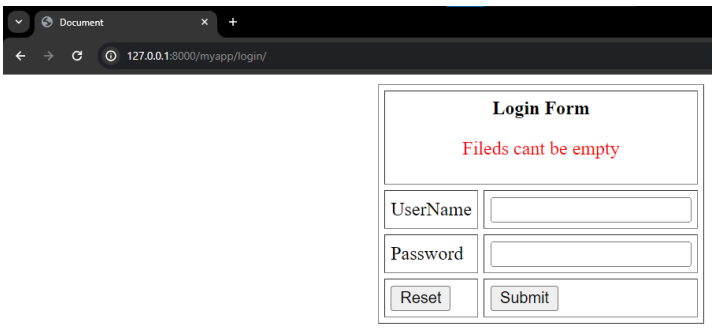
g. url for above view

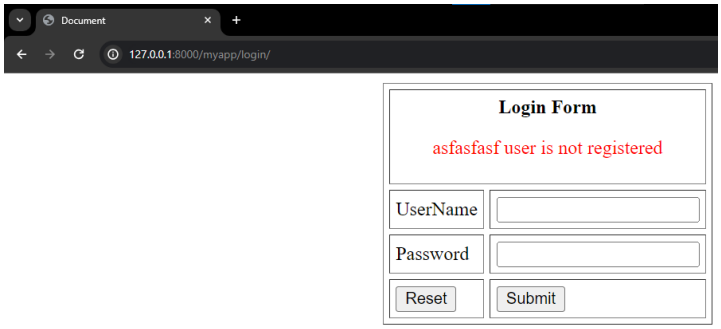
myapp_urls.py

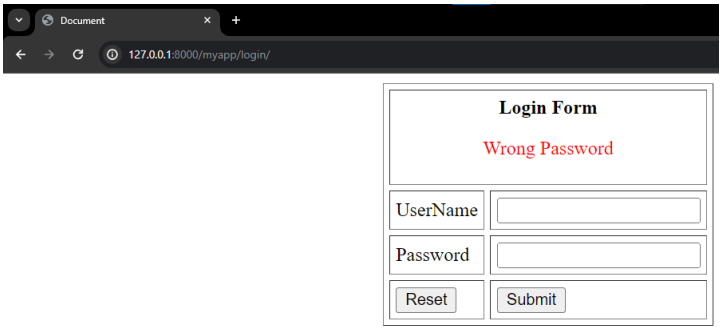
myapp > myapp_urls.py > ...

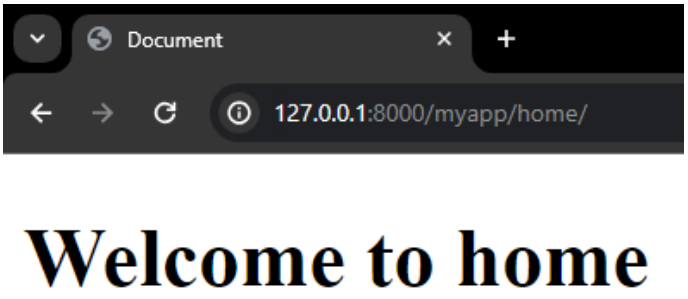
```
1  from django.urls import path
2  from myapp import views
3
4  urlpatterns = [
5      path('register/', views.register_user),
6      path('login/', views.login_user),
7      path('home/', views.home),
8  ]
9
```

h. rnserver and check for output









23. Session