

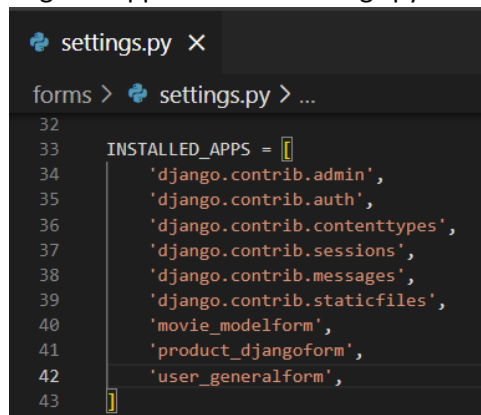
1. Create project

```
\DJANGO T417 203>django-admin startproject forms  
  
\DJANGO T417 203>cd forms  
  
\DJANGO T417 203\forms>code .
```

2. Create 3 applications

```
\forms>python manage.py startapp product_djangoform  
  
\forms>python manage.py startapp movie_modelform  
  
\forms>python manage.py startapp user_generalform
```

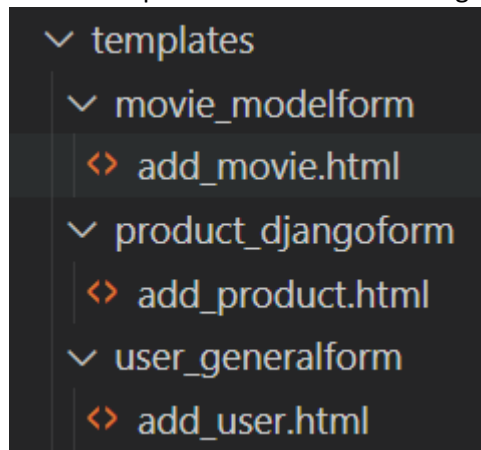
3. Register applications in settings.py



The screenshot shows a code editor window titled 'settings.py'. The file content shows the 'INSTALLED\_APPS' list with the following entries: 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles', 'movie\_modelform', 'product\_djangoform', and 'user\_generalform'.

```
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     'movie_modelform',  
41     'product_djangoform',  
42     'user_generalform',  
43 ]
```

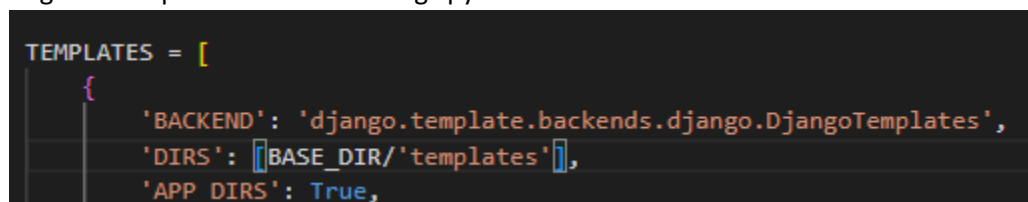
4. Create templates folder with following structure



The screenshot shows a file explorer view of the 'templates' folder. It contains three subfolders: 'movie\_modelform', 'product\_djangoform', and 'user\_generalform'. Each subfolder contains a single HTML file: 'add\_movie.html', 'add\_product.html', and 'add\_user.html' respectively.

```
▼ templates  
  ▼ movie_modelform  
    <> add_movie.html  
  ▼ product_djangoform  
    <> add_product.html  
  ▼ user_generalform  
    <> add_user.html
```

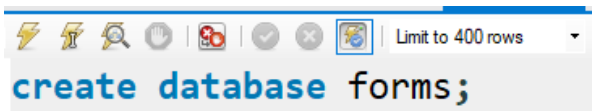
5. Register templates folder in settings.py



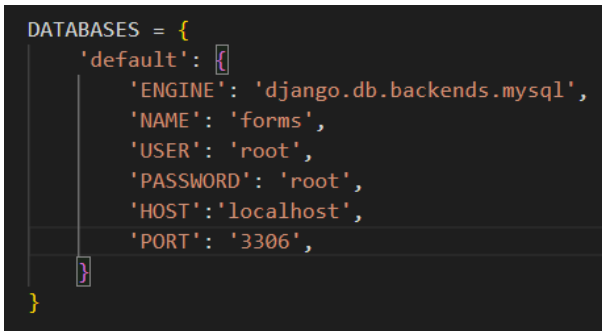
The screenshot shows a code editor window titled 'settings.py'. The file content shows the 'TEMPLATES' list with the following entry: 'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': [BASE\_DIR/'templates'], and 'APP\_DIRS': True.

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [BASE_DIR/'templates'],  
        'APP_DIRS': True,
```

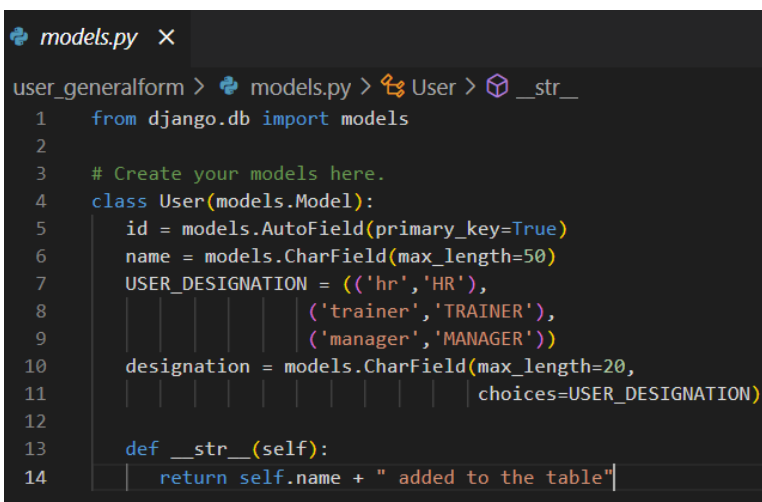
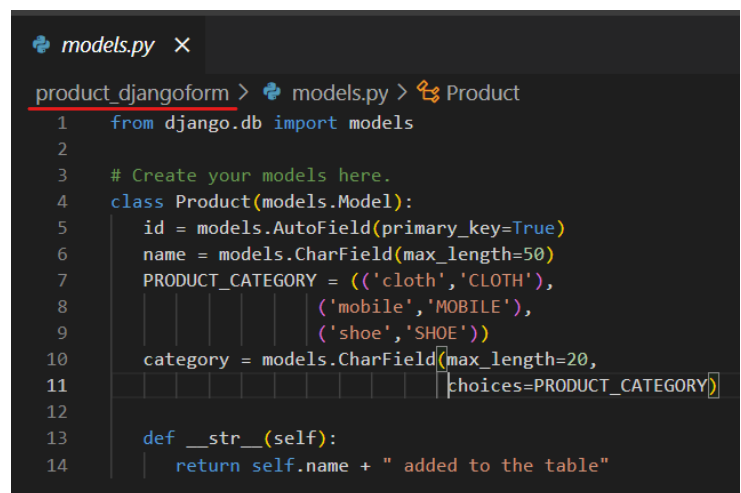
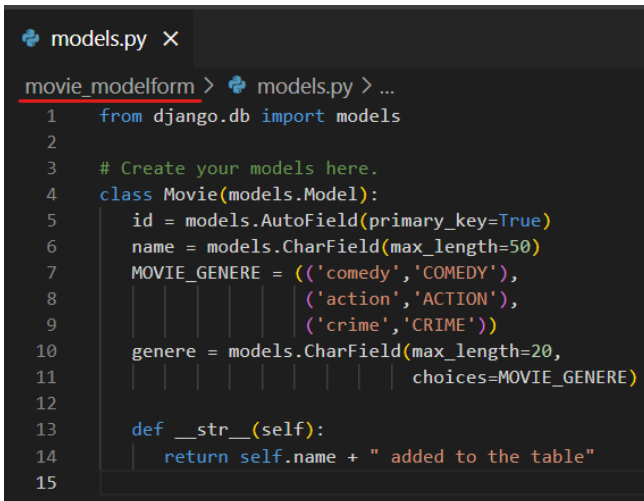
6. Create database forms



7. Register database in settings.py



8. Create model in each application



## 9. Register model of each application in admin.py

```
admin.py > admin.py > MovieAdmin
1 from django.contrib import admin
2 from movie_modelform.models import Movie
3
4 # Register your models here.
5 class MovieAdmin(admin.ModelAdmin):
6     list_display = ['id','name','genre']
7     admin.site.register(Movie,MovieAdmin)
8

admin.py > admin.py > ProductAdmin
1 from django.contrib import admin
2 from product_djangoform.models import Product
3
4 # Register your models here.
5 class ProductAdmin(admin.ModelAdmin):
6     list_display = ['id','name','category']
7     admin.site.register(Product,ProductAdmin)
8
9

admin.py > admin.py > UserAdmin
1 from django.contrib import admin
2 from user_generalform.models import User
3
4 # Register your models here.
5 class UserAdmin(admin.ModelAdmin):
6     list_display = ['id','name','designation']
7     admin.site.register(User,UserAdmin)
8
```

## 10. Makemigrations and migrate

```
C:\TEJAS KASARE (Very imp folder)\my notes\Batch Wise\DJANGO T417 203\forms>python manage.py makemigrations
Migrations for 'movie_modelform':
  movie_modelform\migrations\0001_initial.py
    - Create model Movie
Migrations for 'product_djangoform':
  product_djangoform\migrations\0001_initial.py
    - Create model Product
Migrations for 'user_generalform':
  user_generalform\migrations\0001_initial.py
    - Create model User

C:\TEJAS KASARE (Very imp folder)\my notes\Batch Wise\DJANGO T417 203\forms>python manage.py migrate
Operations to perform:
```

## 11. Check tables

- 1 • create database forms;
- 2 • use forms;
- 3 • show tables;

Result Grid	Filter Rows:	Export: Wri
Tables_in_forms		
auth_permission		
auth_user		
auth_user_groups		
auth_user_user_permissions		
django_admin_log		
django_content_type		
django_migrations		
django_session		
movie_modelform_movie		
product_djangoform_product		
user_generalform_user		

## 12. Working with django form

Since we are creating django form in Product application, bellow step are for application - **product\_djangoform**

- a. Create forms.py in product\_djangoform application

```
forms.py  X
product_djangoform > forms.py > ProductForm
1  from django import forms
2
3  class ProductForm(forms.Form):
4      name = forms.CharField(max_length=50)
5      PRODUCT_CATEGORY = (('cloth','CLOTH'),
6                          ('mobile','MOBILE'),
7                          ('shoe','SHOE'))
8      category = forms.ChoiceField(choices=PRODUCT_CATEGORY)
```

- b. Create view to display above form

```
views.py  X
product_djangoform > views.py > ...
1  from django.shortcuts import render
2  from product_djangoform.forms import ProductForm
3
4  # Create your views here.
5  def add_product(request):
6      data = {}
7      form=ProductForm()
8      data['product_form']=form
9      return render(request,'product_djangoform/add_product.html',context=data)
10
```

- c. Display form in templates > product\_djangoform > add\_product.html

```
add_product.html  X
templates > product_djangoform > add_product.html
8  <body>
9      <h1>Add Your Product</h1>
10     <form action="post">
11         {{product_form.as_p}}
12         <br>
13         <input type="submit">
14     </form>
15 </body>
16 </html>
```

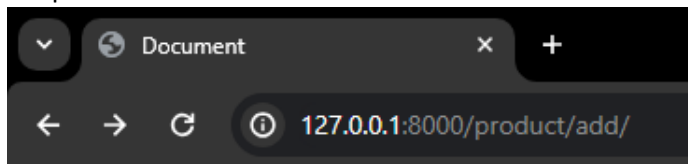
- d. Create application level url
  - i. Create product\_urls.py in product\_djangoform

```
product_urls.py ×  
product_djangoform > product_urls.py > ...  
1 from django.urls import path  
2 from product_djangoform import views  
3 urlpatterns = []  
4     path('add/', views.add_product),  
5 ]
```

- e. Register above application level url in urls.py

```
urls.py ×  
forms > urls.py > ...  
17 from django.contrib import admin  
18 from django.urls import path,include  
19  
20 urlpatterns = [  
21     path('admin/', admin.site.urls),  
22     path('product/',include('product_djangoform.product_urls'))  
23 ]  
24
```

- f. Output :



## Add Your Product

Name:

Category: