
DJANGO TASK 4

1. Create a new Django project

Library Project crated

2. Create an app

Librart_app crated

3. Create two models and then add items (using admin panel or shell)

Model.py

```
from django.db import models

# Create your models here.

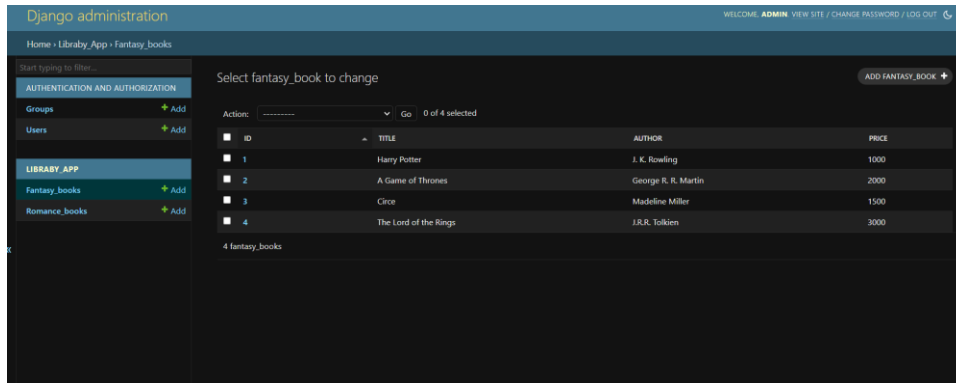
class fantasy_book(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    price = models.IntegerField()

    class Meta:
        ordering = ['id']

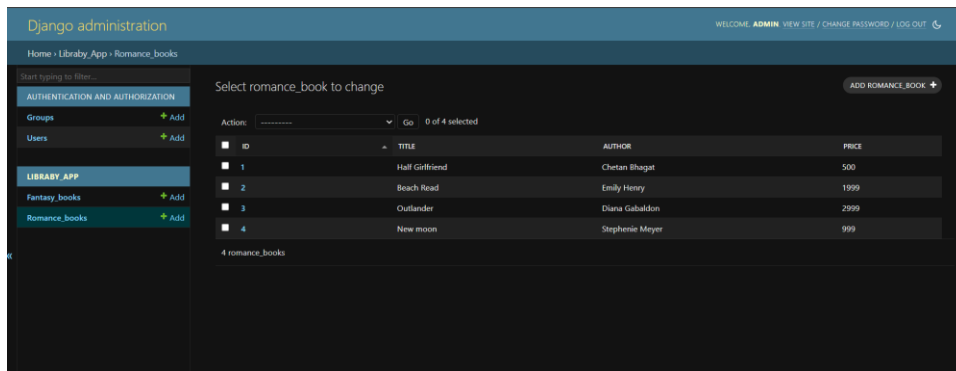
class romance_book(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    price = models.IntegerField()

    class Meta:
        ordering = ['id']
```

Model 1 with data



Model 2 with data



4. Create list view using function-based view to show list of items from the first model

View.py

```
from django.views.generic import ListView,DetailView
from django.shortcuts import render
from .models import fantasy_book,romance_book
# Create your views here.

# list view using function based view

def function_list(request):
    fun_list=fantasy_book.objects.all()
    return render(request,'index.html',{'data':fun_list})
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h3>List view using Function based view</h3>
  {% for i in data %}
  <ul>
    <li>Book Name : {{i.title}}</li>
    <li>Author : {{i.author}}</li>
    <li>Price: {{i.price}}</li>
  </ul>
  {% endfor %}
</body>
</html>
```

Output



List view using Function

- Book Name : Harry Potter
- Author : J. K. Rowling
- Price: 1000
- Book Name : A Game of Thrones
- Author : George R. R. Martin
- Price: 2000
- Book Name : Circe
- Author : Madeline Miller
- Price: 1500
- Book Name : The Lord of the Rings
- Author : J.R.R. Tolkien
- Price: 3000

5. Create list view using class-based view to show list of items from the second model

View.py

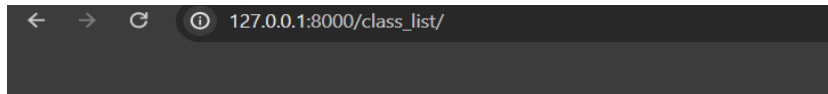
```
from django.views.generic import ListView,DetailView
from django.shortcuts import render
from .models import fantasy_book,romance_book
# Create your views here.

# list view using class based view
class class_list(ListView):
    model = romance_book
    template_name = 'home.html'
    context_object_name = 'book_list'
```

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3>List view using Class based view </h3>
    {% for i in book_list %}
    <ul>
        <li>Book Name : {{i.title}}</li>
        <li>Author : {{i.author}}</li>
        <li>Price: {{i.price}}</li>
    </ul>
    {% endfor %}
</body>
</html>
```

Output



List view using Class based view

- Book Name : Half Girlfriend
- Author : Chetan Bhagat
- Price: 500

- Book Name : Beach Read
- Author : Emily Henry
- Price: 1999

- Book Name : Outlander
- Author : Diana Gabaldon
- Price: 2999

- Book Name : New moon
- Author : Stephenie Meyer
- Price: 999

6. Create detail view using function-based view to show detail view of an item
View.py

```
from django.views.generic import ListView,DetailView
from django.shortcuts import render
from .models import fantasy_book,romance_book
# Create your views here.

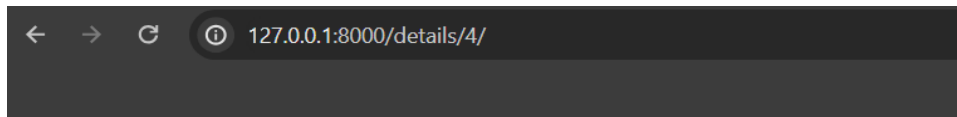
|
# detail view using function based view
def function_details(request,pk):
    fun_list=fantasy_book.objects.get(pk=pk)

    return render(request,'details.html',{'data':fun_list})
```

Details.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h3>Detail view using function based view</h3>
  <ul>
    <li>Book Name : {{data.title}}</li>
    <li>Author : {{data.author}}</li>
    <li>Price: {{data.price}}</li>
  </ul>
</body>
</html>
```

Output



Detail view using function based view

- Book Name : The Lord of the Rings
- Author : J.R.R. Tolkien
- Price: 3000

7. Create detail view using class-based view to show detail view of an item

View.py

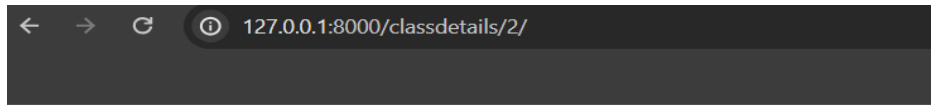
```
from django.views.generic import ListView, DetailView
from django.shortcuts import render
from .models import fantasy_book, romance_book
# Create your views here.

class Class_details(DetailView):
    model = romance_book
    template_name = 'class_details.html'
    context_object_name = 'book_list'
```

Class_details.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h3>Detail view using class based view</h3>
    <ul>
        <li>Book Name : {{book_list.title}}</li>
        <li>Author : {{book_list.author}}</li>
        <li>Price: {{book_list.price}}</li>
    </ul>
</body>
</html>
```


Output



Detail view using class based view

- Book Name : Beach Read
- Author : Emily Henry
- Price: 1999

View.py

```
from django.views.generic import ListView,DetailView
from django.shortcuts import render
from .models import fantasy_book,romance_book
# Create your views here.

# list view using function based view

def function_list(request):
    fun_list=fantasy_book.objects.all()
    return render(request,'index.html',{'data':fun_list})

# list view using class based view
class class_list(ListView):
    model = romance_book
    template_name = 'home.html'
    context_object_name = 'book_list'

# detail view using function based view
def function_details(request,pk):
    fun_list=fantasy_book.objects.get(pk=pk)

    return render(request,'details.html',{'data':fun_list})

# detail view using class based view
class Class_details(DetailView):
    model = romance_book
    template_name = 'class_details.html'
    context_object_name = 'book_list'
```

App.urls.py

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

urlpatterns = [
    path('', function_list, name="function_list"),
    path('details/<int:pk>/', function_details, name='function_details'),

    path('class_list/', class_list.as_view()),
    path('classdetails/<int:pk>/', Class_details.as_view(), name='Class_details'),

]
```