
DJANGO TASK 5

1.Create a new django project

```
C:\Windows\System32\cmd.e  +  v
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

D:\Beinex\Python _task\15.03.2024>python -m venv env

D:\Beinex\Python _task\15.03.2024>.\env\scripts\activate

(env) D:\Beinex\Python _task\15.03.2024>pip install django
Collecting django
  Using cached Django-5.0.3-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.7.0 (from django)
  Using cached asgiref-3.7.2-py3-none-any.whl.metadata (9.2 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.4.4-py3-none-any.whl.metadata (4.0 kB)
Collecting tzdata (from django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.0.3-py3-none-any.whl (8.2 MB)
Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-5.0.3 sqlparse-0.4.4 tzdata-2024.1

(env) D:\Beinex\Python _task\15.03.2024>django-admin startproject book_stall

(env) D:\Beinex\Python _task\15.03.2024>cd book_stall

(env) D:\Beinex\Python _task\15.03.2024\book_stall>python manage.py startapp book_app

(env) D:\Beinex\Python _task\15.03.2024\book_stall>|
```

2.Create required apps and models

```
from django.db import models

# Create your models here.

class BookModel1(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    publiced_year = models.IntegerField()

    class Meta:
        ordering = ['id'] |

class BookModel2(models.Model):
    title = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    price = models.IntegerField()
    image = models.FileField(upload_to='image')

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

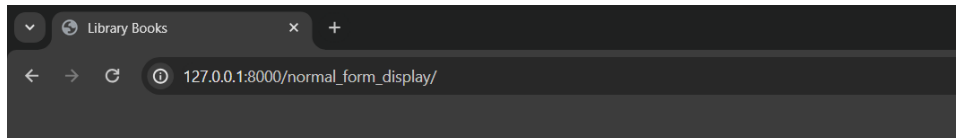
3.4. Create a view for adding details to first model (Use normal forms) and Create views for listing created records of each model in a table

```
from django.shortcuts import render, redirect
from .models import BookModel1, BookModel2
from .forms import NormalForm, BookForm

# Create a view for adding details to first model (Use normal forms)
def add_normal_form(request):
    if request.method == 'POST':
        form = NormalForm(request.POST)
        if form.is_valid():
            title = form.cleaned_data['title']
            author = form.cleaned_data['author']
            publiced_year = form.cleaned_data['publiced_year']
            BookModel1.objects.create(title=title, author=author, publiced_year=publiced_year)
            return redirect(normal_form_display)
    else:
        form = NormalForm()
    return render(request, 'add_book.html', {'form': form})

# Create views for listing created records of each model in a table
def normal_form_display(request):
    book_data=BookModel1.objects.all()
    return render(request, 'normal_form_display.html', {'book_data':book_data})
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Library Books</title>
    <style>...
</head>
<body>
    <div>
        <h1>Library Books Normal Form</h1>
        <table>
            <thead>
                <tr>
                    <th>Title</th>
                    <th>Author</th>
                    <th>Publiced Year</th>
                </tr>
            </thead>
            <tbody>
                {% for book in book_data %}
                    <tr>
                        <td>{{ book.title }}</td>
                        <td>{{ book.author }}</td>
                        <td>{{ book.publiced_year }}</td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</body>
</html>
```



Library Books Normal Form

Title	Author	Publiced Year
Half Girlfriend	Chetan Bhagat	2017
Beach Read	Emily Henry	2020
Outlander	Diana Gabaldon	2014
New moon	Stephenie Meyer	2006

5,6.Create views for listing created records of each model in a table and Implement file upload using ModelForm and list the uploaded files

```
from django.shortcuts import render,redirect
from .models import BookModel1, BookModel2
from .forms import NormalForm,BookForm

# Create a view for adding details to second model (Use model forms)

def add_model_form(request):
    form_book = BookForm
    if request.method == 'POST':
        form_book = BookForm(request.POST, request.FILES)
        if form_book.is_valid():
            form_book.save()
            return redirect(model_form_display)
    return render(request, 'book.html', {'form_book': form_book})

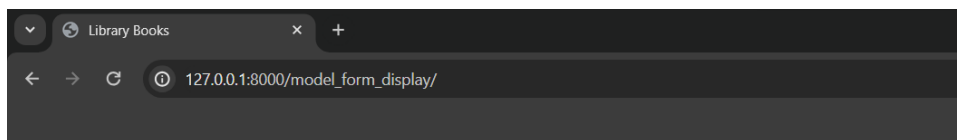
# Implement file upload using ModelForm and list the uploaded files

def model_form_display(request):
    book_data=BookModel2.objects.all()
    return render(request,'display.html',{'book_data':book_data})
```





```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Library Books</title>
  <style>...
</style>
</head>
<body>
  <div>
    <h1>Library Books Model Form</h1>
    <table>
      <thead>
        <tr>
          <th>Title</th>
          <th>Author</th>
          <th>Price</th>
          <th>Image</th>
        </tr>
      </thead>
      <tbody>
        {% for book in book_data %}
          <tr>
            <td>{{ book.title }}</td>
            <td>{{ book.author }}</td>
            <td>{{ book.price }}</td>
            <td></td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
  </div>
</body>
</html>

```



Library Books Model Form

Title	Author	Price	Image
Harry Potter	J. K. Rowling	2000	
A Game of Thrones	George R. R. Martin	2000	
Circe	Madeline Miller	1500	
The Lord of the Rings	J.R.R. Tolkien	3000	

Form.py

```
from django import forms
from .models import BookModel2

class NormalForm(forms.Form):
    title = forms.CharField(max_length=100)
    author = forms.CharField(max_length=100)
    publiced_year = forms.IntegerField()

class BookForm(forms.ModelForm):
    class Meta:
        model = BookModel2
        fields = '__all__'
```