Apr 02

Insert data into model class by using faker module

D:\Django_20MAR_7PM>django-admin startproject modelproject3
D:\Django_20MAR_7PM>cd modelproject3
D:\Django_20MAR_7PM\modelproject3>py manage.py startapp testapp

-->Add app in settings.py

-->Database configuration(Use Mysql)
-->Create a database in Mysql(studentdb_7pm)

-->Check database connections.

- **models.py**

```
class Student(models.Model):
   rollno = models.IntegerField()
   name = models.CharField(max_length=30)
   dob = models.DateField()
   marks = models.IntegerField()
   email = models.EmailField()
   phonenumber = models.BigIntegerField()
   address = models.TextField()
```

-->Makemigrations and migrate

- **admin.py**

```
from testapp.models import Student
class StudentAdmin(admin.ModelAdmin):
   list_display = ['rollno','name','dob','marks','email','phonenumber','address']
admin.site.register(Student,StudentAdmin)
```

-->Create super user, then for table in admin interface.

-->Create a file with the name populate.py under modelproject3 folder.

- **populate.py**

```
import os
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'modelproject3.settings')
```

```
import django
django.setup()

from testapp.models import Student
from faker import Faker
from random import *
def phonenumbergen():
        d1 = randint(6,9)
        num = '' + str(d1)
        for i in range(9):
                num += str(randint(0,9))
        return int(num)
def populate(n):
   for i in range(n):
      fake = Faker()
      frollno = fake.random_int(min=1,max=999)
      fname = fake.name()
      fdob = fake.date()
      fmarks = fake.random_int(min=1,max=100)
      femail = fake.email()
      fphonenumber = phonenumbergen()
      faddress = fake.address()

Student.objects.get_or_create(rollno=frollno,name=fname,dob=fdob,marks=f
marks,email=femail,phonenumber=fphonenumber,address=faddress)
n = int(input('Enter number of records:'))
populate(n)
print(f'{n} Records Inserted Successfully.....')
```

- **views.py**

```
from testapp.models import Student
def student_view(request):
   student_list = Student.objects.all()
   return render(request,'testapp/std.html',{'student_list':student_list})
```

- **urls.py**

```
path('std/',views.student_view),
```

- **std.html**

```
<body>
   <h1>Student Information</h1>
   {% if student_list %}
   {% for student in student_list%}
   <h2>{{student.name}} Information</h2>
   <ul>
    <li>Student Rollno:{{student.rollno}}</li>
    <li>Student BOD:{{student.dob}}</li>
    <li>Student Marks:{{student.marks}}</li>
    <li>Student Email:{{student.email}}</li>
    <li>Student Phone Number:{{student.phonenumber}}</li>
    <li>Student Address:{{student.address}}</li>
   </ul>
   <br>
   {% endfor %}
   {% else %}
   <p>No records found in the database</p>
   {% endif %}
  </body>
```

- **views.py**

```
student_list = Student.objects.filter(marks__lt=35)
student_list = Student.objects.filter(name__startswith='S')
student_list = Student.objects.all().order_by('marks')#Ascending order of
marks
student_list = Student.objects.all().order_by('-marks')#Descending order
```