Apr 12

Working with Model Forms:

Model Form-->Model based form

```
name = form.cleaned_data['name']
marks = form.cleaned_data['marks']

records = Student.get_or_create(name=name,marks=marks)
```

10-->fields are there
100 fields

Model based form
only one line
form.save()

-->Sometimes we can create form based on Model, such type of forms are called as Model Based Forms or Model Forms.

-->The main advantage of model forms is we can grab the end user input data and we can save that data very easily in the database.

```
form.save()
form.save(commit=True)
```

How to develop model based forms:

1). While developing form class, we have to inherit from forms.ModelForm class instead of forms.Form class

```
class RegistrationForm(forms.Form)--->Normal Form
class RegistrationForm(forms.ModelForm)-->Model Form
```

2).

```
class RegistrationForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = '__all__'
```

Case-1:- Instead of all fields, if we required particular fields.

```
class Meta:
    model = Student
    fields = ('field1','field2','field3')
```

Case-2:- Instead of all fields, if we want to exclude certain fields.

```
class Meta:
    model = Student
    exclude = ['field1','field2']
```

Ex:ModelFormProjct

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

-->Add app in settings.py

- **models.py**

```
class Student(models.Model):
   name = models.CharField(max_length=30)
   marks = models.IntegerField()
```

-->Makemigrations and migrate.

- **admin.py**

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

- **forms.py**

```
from django import forms
from testapp.models import Student
class StudentForm(forms.ModelForm):
   name = forms.CharField()
   marks = forms.IntegerField()
   class Meta:
      model = Student
      field = '__all__'
```

- **views.py**

```
from testapp.forms import StudentForm
def student_view(request):
   if request.method == 'POST':
      form = StudentForm(request.POST)
      if form.is_valid():
         form.save(commit = True)
         print('Record inserted into DB successfully....')
   form = StudentForm()
   return render(request,'testapp/studentform.html',{'form':form})
```

- **studentform.html**

```
<body>
   <div class="container" align='center'>
    <h1>Student Registration Form</h1>
    <form  method="post">
     {{form.as_p}}
     {% csrf_token %}
     <input type="submit" class="btn btn-primary" name="" value="Register">
    </form>
   </div>
</body>
```

- **urls.py**

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

- **std1.css**

```
body{
  background: yellow;
  color:red;
}
h1{
  text-align: center;
}
```

Ex:MovieProject

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

-->Add app in settings.py

- **models.py**

```
class Movie(models.Model):
   rdate = models.DateField()
   moviename = models.CharField(max_length=20)
   hero = models.CharField(max_length=20)
   heroine = models.CharField(max_length=20)
   rating = models.FloatField()
```

-->makemigrations and migrate

- **admin.py**

```
from testapp.models import Movie
class MovieAdmin(admin.ModelAdmin):
   list_display = ['rdate','moviename','hero','heroine','rating']
admin.site.register(Movie,MovieAdmin)
```

- **views.py**

```
def index_view(request):
   return render(request,'testapp/index.html')
```

- **urls.py**

```
path('',views.index_view)
```

- **index.html**

```
<body>
   <div class="container" align='center'>
     <h1>Sunny Movies.....</h1>
     <p>Upto date movie information</p>
     <a href="#" class="btn btn-primary">ADD</a>
     <a href="#" class="btn btn-primary">List Movies</a>
   </div>
  </body>
```