**WEEK 8**

**AIM:** A form validation using Django.

**DESCRIPTION:**

**Django Forms**

Forms are basically used for taking input from the user in some manner and using that information for logical operations on databases. For example, registering a user by taking input as his name, email, password, etc.

Django maps the fields defined in Django forms into HTML input fields. Django handles three distinct parts of the work involved in forms:

* preparing and restructuring data to make it ready for rendering
* creating HTML forms for the data
* receiving and processing submitted forms and data from the client

**Creating a Django Form**

Creating a form in Django is completely similar to creating a model, one needs to specify what fields would exist in the form and of what type. For example, to input, a registration form one might need First Name (CharField), Roll Number (IntegerField), and so on.

**Create Django Form from Models**

Django ModelForm is a class that is used to directly convert a model into a Django form.

**Django Form Validation**

Django provides built-in methods to validate form data automatically. Django forms submit only if it contains CSRF tokens. It uses uses a clean and easy approach to validate data.

The **is\_valid()** method is used to perform validation for each field of the form, it is defined in Django Form class. It returns True if data is valid and place all data into a cleaned\_data attribute.

**PROGRAM(s):**

**Models.py**

from django.db import models

# Create your models here.

class Feedback(models.Model):

Rollno=models.IntegerField()

Name=models.CharField(max\_length=40)

Marks=models.FloatField()

Feedback=models.CharField(max\_length=30)

**Views.py**

from django.shortcuts import render

from .import forms

# Create your views here.

def fD(request):

form = forms.Ff()

if request.method=='POST':

form = forms.Ff(request.POST)

if form.is\_valid():

form.save(commit=True)

return render(request,'input.html',{'form':form})

**admin.py**

from django.shortcuts import render

from .import forms

# Create your views here.

def fD(request):

form = forms.Ff()

if request.method=='POST':

form = forms.Ff(request.POST)

if form.is\_valid():

form.save(commit=True)

return render(request,'input.html',{'form':form})

**forms.py**

from django import forms

from modelapp.models import Feedback

class Ff(forms.ModelForm):

class Meta:

model=Feedback

fields='\_\_all\_\_'

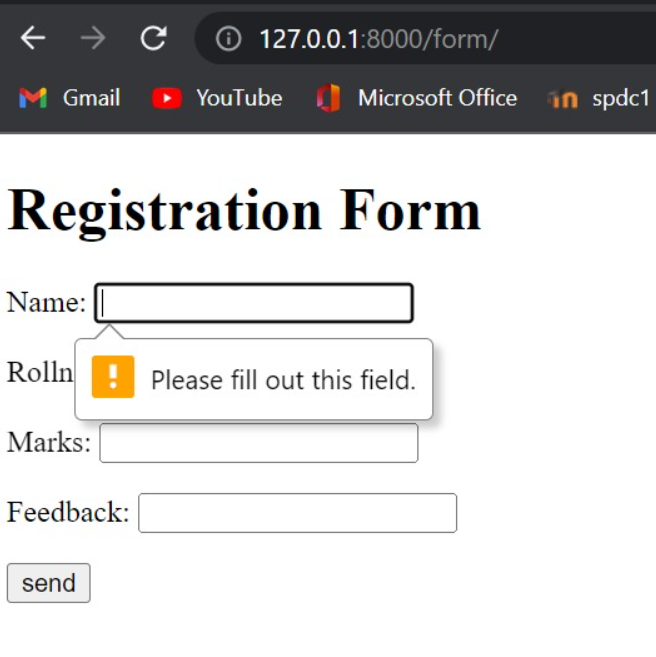
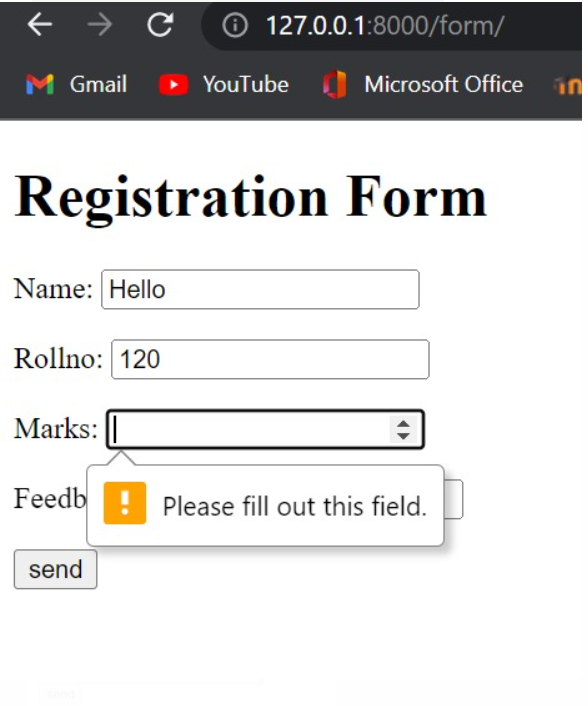
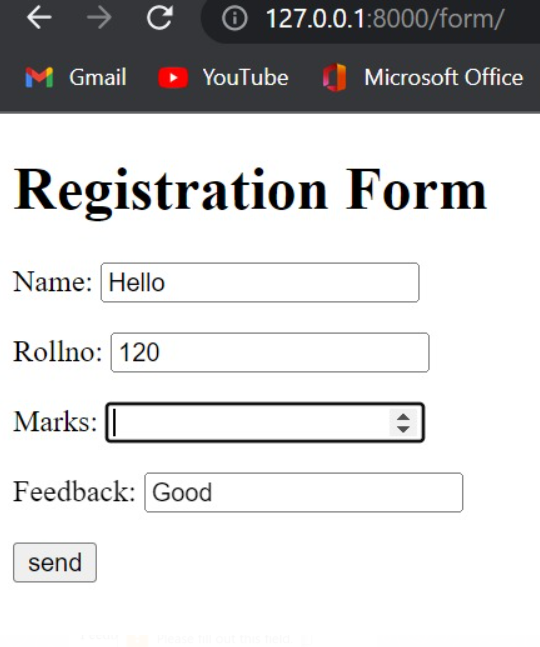
**OUTPUT:**

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, table

Description automatically generated

**  **

**CONCLUSION:**

The required form has been created using Django. Form validation has also been done successfully.