**Django Form and Field Level Validations**

Validations are applying business rules.

These rules are defined at form level or field level.

**Form level validations**

The **is\_vaild()** function is used when we need to validate complete form data. It is defined in Django’s form class. In this validation, Python-datatypes will be checked. This function will return boolean results, i.e. it returns True if data is valid and otherwise False.

**Models.py**

from django.db import models

# Create your models here.

class Student(models.Model):

    rollno=models.IntegerField(primary\_key=True)

    name=models.CharField(max\_length=20)

    course=models.CharField(max\_length=15)

    email=models.EmailField()

    class Meta:

        db\_table="student"

**forms.py**

from django.forms import ModelForm

from app1.models import Student

class StudentForm(ModelForm):

    class Meta:

        model=Student

        fields="\_\_all\_\_"

**views.py**

from django.shortcuts import render,redirect

from app1.forms import StudentForm

# Create your views here.

def student\_view(request):

    if request.method=="POST":

        stform=StudentForm(request.POST)

        if stform.is\_valid():

            stform.save()

            return redirect("/studview")

    else:

        stform=StudentForm()

        return render(request,"home.html",context={"form":stform})

**urls.py**

from django.contrib import admin

from django.urls import path

from app1 import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('studview',views.student\_view)

]

Template

Home.html

<form  action="studview" method="post">

    {% csrf\_token %}

    {{form.as\_p}}

<input type="submit" value="save">

</form>

Settings.py

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'app1',

]

import os.path

template\_path=os.path.join(BASE\_DIR,"templates")

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [template\_path],

        'APP\_DIRS': True,

**Form validations using cleaned\_data**

The **cleaned\_data** is used to validate certain fields. Basically, any information entered into a form will be sent to the server as strings. It makes no difference which type of form field was utilized when the form was created. Everything would ultimately be converted to strings by the browser.

So, we can say that Django automatically changes data to the appropriate type when it cleans it. This cleansed and validated data is referred to as cleaned data in Django. And, the cleaned data dictionary provides access to cleaned data.

**Syntax**

cleaned\_data[‘field-name’]