How to create a form using Django Forms ?

Last Updated : 16 Aug, 2021

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. This post revolves around how to create a basic form using various Form Fields and attributes. 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.

Creating a form using Django Forms

Illustration of **Django Forms** using an Example. Consider a project named geeksforgeeks having an app named geeks.

> *Refer to the following articles to check how to create a project and an app in Django.*

- *How to Create a Basic Project using MVT in Django?*
- *How to Create an App in Django ?*

In your geeks app make a new file called forms.py where you would be making all your forms. To create a Django form you need to use Django Form Class. Let's demonstrate it.

In your forms.py Enter the following,

Python3

```
from django import forms

# creating a form
class InputForm(forms.Form):
```



```
    last_name = forms.CharField(max_length = 200)
    roll_number = forms.IntegerField(
                    help_text = "Enter 6 digit roll number"
                    )
    password = forms.CharField(widget = forms.PasswordInput())
```

Let's explain what exactly is happening, left side denotes the name of the field and to the right of it, you define various functionalities of an input field correspondingly. A field's syntax is denoted as

Syntax :

```
Field_name = forms.FieldType(attributes)
```

Now to render this form into a view, move to views.py and create a home_view as below.

Python3

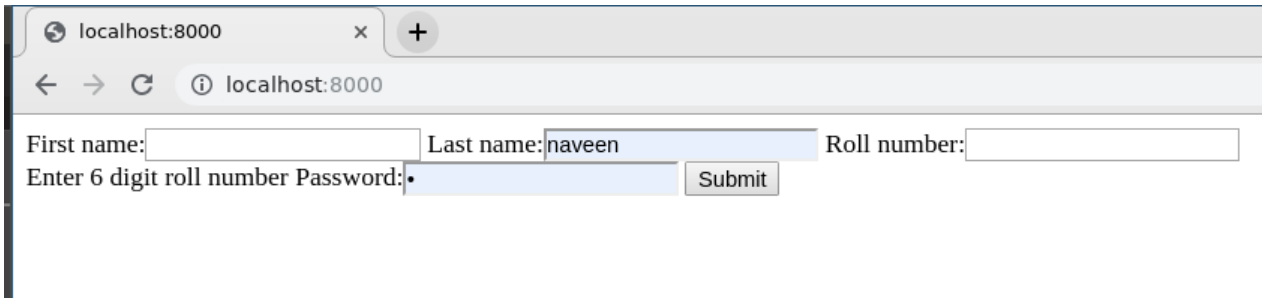
```
from django.shortcuts import render
from .forms import InputForm

# Create your views here.
def home_view(request):
    context ={}
    context['form']= InputForm()
    return render(request, "home.html", context)
```

In view, one needs to just create an instance of the form class created above in forms.py. Now let's edit templates > home.html

HTML

```
<form action = "" method = "post">
    {% csrf_token %}
    {{form }}
    <input type="submit" value=Submit">
</form>
```

All set to check if the form is working or not let's visit http://localhost:8000/



The form is working properly but visuals are disappointing, Django provides some predefined ways to show forms in a convenient manner. In templates, the following will modify the inputs as,

- {{ form.as_table }} will render them as table cells wrapped in <tr> tags
- {{ form.as_p }} will render them wrapped in <p> tags
- {{ form.as_ul }} will render them wrapped in <li> tags

One can modify these settings also and show fields as he wants using {{ form.field_name }} but this may alter the normal process of validation if some field is empty and hence needs extraordinary care. More – Django Forms

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our Mastering Django Framework - Beginner to Advanced Course on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you

to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N Nave... 23

Previous Article

Django Forms

Next Article

Django Form | Data Types and Fields

Similar Reads

Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

{{ form.as_p }} - Render Django Forms as paragraph

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

{{ form.as_table }} - Render Django Forms as table

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

{{ form.as_ul }} - Render Django Forms as list

Django forms are an advanced set of HTML forms that can be created using python and support all features of HTML forms in a pythonic way. Rendering...

2 min read

Initial form data - Django Forms

After creating a Django Form, if one requires some or all fields of the form be filled with some initial data, one can use functionality of Django forms to do so. I...

3 min read

Add Forms Dynamically In Django Using Formset And JavaScript

Formsets in Django allow us to manage a collection of forms as a single unit. They are particularly useful when we need to handle multiple forms on a single...

8 min read

Render HTML Forms (GET & POST) in Django

Django is often called "Batteries Included Framework" because it has a default setting for everything and has features that can help anyone develop a website...

5 min read

CharField - Django Forms

CharField in Django Forms is a string field, for small- to large-sized strings. It is used for taking text inputs from the user. The default widget for this input is...

5 min read

BooleanField - Django Forms

BooleanField in Django Forms is a checkbox field which stores either True or False. It is used for taking boolean inputs from the user. The default widget for...

5 min read

Article Tags : Python Django-forms Python Django

Practice Tags : python

GeeksforGeeks

Corporate & Communications Address:- A-143, 9th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305) | Registered Address:- K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

GET IT ON Google Play Download on the App Store

Company Languages

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

System Design Bootcamp
Interview Questions

School Subjects
Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos
DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved