



Django project to create a Comments System

Last Updated : 02 Nov, 2023

Commenting on a post is the most common feature a post have and implementing in Django is way more easy than in other frameworks. To implement this feature there are some number of steps which are to be followed but first lets start by creating a new project.

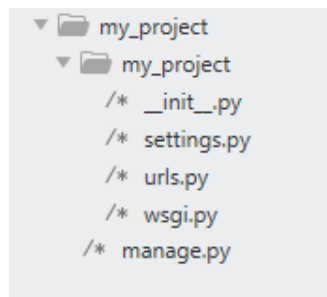
How to create Comment Feature in Django?

- Open command prompt and run the following commands.

Start a new django project as:

```
django-admin startproject my_project
```

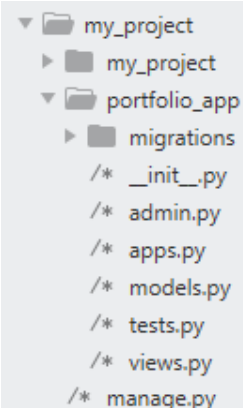
After executing this command you will see following files and folder that django creates for you. To see this folder you can make use of tree command on command prompt or open the folder in an ide, in my case I have used *sublime text editor*.



Create a new app as:

```
python manage.py startapp portfolio_app
```

Django automatically creates files so that you can focus on your task without bothering of other things like making a file and structuring them.



```
▼ my_project
  ► my_project
  ▼ portfolio_app
    ► migrations
      /* __init__.py
      /* admin.py
      /* apps.py
      /* models.py
      /* tests.py
      /* views.py
      /* manage.py
```

In this article we will make a post model to post a picture:

The post consist of following fields:

1. Image
2. Caption or description
3. Date posted
4. Author of the image.

First, importing the required modules in **models.py**:

```
from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User
```

Now, continuing with models.py module, append the following snippet.

python3

```
class Post(models.Model):
    image = models.ImageField(
        default = "default_foo.png", upload_to = "post_picture")
    caption = models.TextField()
    date_posted = models.DateTimeField(default = timezone.now)
    author = models.ForeignKey(User, on_delete = models.CASCADE)

    def __str__(self):
        return f'{self.author.username}\''s Post- {self.title}'

    def save(self, *args, **kwargs):
        super().save(*args, **kwargs)
        img = Image.open(self.image.path)
        if img.height > 400 or img.width > 400:
```

```
output_size = (300, 300)
img.thumbnail(output_size)
img.save(self.image.path)
```

Name of the model is post model (it could be any of your choice). Let us understand the details of each field used.

Image field: This field allows to store image in the database. The default value includes a default image which I have named as default_foo.png. It is automatically saved when no picture is available or posted by the user. This Image is located in a folder named Media in your my_project directory. If a user upload a picture, it is uploaded in the directory named post_picture, set by the attribute upload_to.

Caption: This field depicts the description of the image named as caption for the image field.

Date posted: This field takes account of the time and date the image is uploaded. Default value timezone.now saves the time at which this post is created. The time/date is not changed if the post is updated later after the post.

Author: This attribute takes account of the user posted this post. It takes django user as the parameter to do so. The on_delete field takes care for abnormalities which might happen to the post if a user gets deleted. Models, cascade field deletes all the user related post data simultaneously when a user gets deleted.

__str__ method: These methods are also known as magic methods in python. It takes self as its attributes. This method shows the title information to the django admin at the backend.

save method: This method is override to save the post with user customized values. If we want to save images of a particular size in our database, save method is used to do so. For this you have to import PIL in your models.py module to read the image. Open the image by open method into a variable and store it in a particular output size in your database to avoid storage of large images in your database.

After successfully creating the post model run the following commands on the terminal the make and save the changes in database.

```
python manage.py makemigrations
python manage.py migrate
```

This will make the post table in your database. I'm using django default database.

*For visualizing your tables and database I recommend you to download **DB Browser for sqlite** which will provide you an interface to view, edit all your schemas.*

After implementing your post model now it's time to implement comment feature on your post.

1. Make a form in forms.py module to get the data.
2. Make a function to save or delete the comment in views.py module.
3. Get the comments from the html file.

Make a form to collect the content of comment in **forms.py module**. If You don't have form.py in your django app, then create one .

Django has inbuilt forms to create it and access it directly without writing a new one from scratch.

python3

```
from django import forms
from .models import Comment

class CommentForm(forms.ModelForm):
    content = forms.CharField(label="", widget = forms.Textarea(
        attrs = {
            'class': 'form-control',
            'placeholder': 'Comment here !',
            'rows': 4,
            'cols': 50
        })
    ))
    class Meta:
        model = Comment
        fields = ['content']
```

Moving to the **views.py** module, to process and save the data coming from html form the following code goes on. I am implementing the comment feature in post detail view, you can use it wherever you want to, the method would be same.

python3

```
from .forms import CommentForm

def post_detailview(request, id):

    if request.method == 'POST':
        cf = CommentForm(request.POST or None)
        if cf.is_valid():
            content = request.POST.get('content')
            comment = Comment.objects.create(post = post, user = request.user, content
            comment.save()
            return redirect(post.get_absolute_url())
        else:
            cf = CommentForm()

    context = {
        'comment_form': cf,
    }
    return render(request, 'socio / post_detail.html', context)
```

Importing *CommentForm* from forms.py module, make an object of commentform and check if it is valid. Django provides built-in feature to check the form credentials and format by *is_valid* method.

Scraping the content of post by get method and creating a new comment by create method followed by Save method saves the data of object created while creating new post in the database.

The cf object is passed into the html by context dictionary to access all the comments in HTML.

Finally, for getting the data from html form, a form tag is used with method used as post, since we have to send the data to the server. The following

snippet shows how the value is taken from the form in the html.

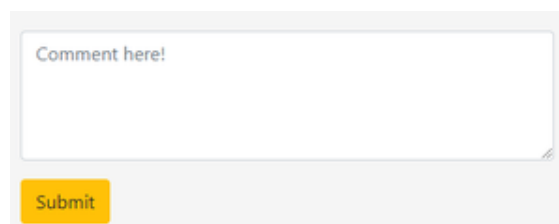
html

```
{% load crispy_forms_tags %}
<html>
  <head>
    <title></title>
  </head>
  <body>
    <form method="POST">
      {% csrf_token %}
      {{ comment_form.as_p }}
    </form>
  </body>
```

Here, *Csrf_token* is used for the security purpose. Django handles the encryption of form through this token.

as_p method is used to show the form in paragraph manner for better visualization.

Output –



Comment box

You can show all the comments through the for loop in HTML. Django provides new HTML language to access data and variables passed in the context in views.py module which contains for loops, if-else conditions etc.

You can view this features in my social media site project in my github.

<https://github.com/Shikharm16/social-media-site-using-django>

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!



shikh...



2

Previous Article

[How to Send Email with Django](#)

Next Article

[Voting System Project Using Django Framework](#)

Similar Reads

Integrating Facebook Comments Plugin in Django Project

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because...

2 min read

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our django...

2 min read

Integrating Facebook Like, Comments and Share Plugin in Flask Project

Flask is a Framework of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than...

2 min read

Voting System Project Using Django Framework

Project Title: Pollster (Voting System) web application using Django framework
Type of Application (Category): Web application. Introduction: We wil...

14 min read

College Management System using Django - Python Project

In this article, we are going to build College Management System using Django and will be using sqlite database. In the times of covid, when education has...

15+ min read

How do we create multiline comments in Python?

Comments are pieces of information present in the middle of code that allows a developer to explain his work to other developers. They make the code more...

3 min read

Project Idea | (Project Approval System)

Academic Project management is a major issue which is faced by many educational institutes, the main reason for this is there is no automated system...

2 min read

How to Create a Basic Project using MVT in Django ?

Prerequisite - Django Project MVT Structure Assuming you have gone through the previous article. This article focuses on creating a basic project to render a...

2 min read

How to create a new project in Django using Firebase Database?

Django is a Python-based web framework that allows you to quickly create efficient web applications. If you are new to Django then you can refer to Django...

3 min read

Create a new Django project in Pycharm using Pycharm Terminal

PyCharm is one of the most popular Python-IDE developed by JetBrains used for performing scripting in Python language. PyCharm provides many useful featur...

2 min read

Article Tags :

Articles

How To

Project

Python

+4 More

Practice Tags :

python



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



Company

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

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

Web Technologies

Languages

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

Data Science & ML

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

Python Tutorial

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

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

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

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

GeeksforGeeks Videos

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved