



[Django Basics](#) [Django Projects](#) [Django Interview Question](#) [Flask](#) [Flask Projects](#) [Python API](#) [Torando](#) [Che](#)

Last Updated : 24 Sep, 2024

Django, a powerful web framework for Python, makes it relatively easy to create a robust and user-friendly Image Gallery Graphical User Interface (GUI) for your web application. In this article, we'll guide you through the process of building an Image Gallery GUI using Django.

## Image Gallery GUI using Django

An Image Gallery GUI using Django is a web application that allows users to view and manage a collection of images stored in a database through a user-friendly graphical interface.

### Setting Up the Project Folder

```
django-admin startproject <project_name>
cd <project_name>
python manage.py startapp gallery
```

**model.py:** Here we have created a Product table with name, description, price, image, created\_at, and updated\_at fields in the table.

#### Python

```
1 from django.db import models
2
3 class Product(models.Model):
4     name = models.CharField(max_length=255)
5     description = models.TextField()
6     image = models.ImageField(upload_to='products/')
7     created_at = models.DateTimeField(auto_now_add=True)
8     updated_at = models.DateTimeField(auto_now=True)
9
10     def __str__(self):
```

```
return self.name
```

**admin.py:** Here we are registering our table in the admin.

```
from django.contrib import admin
from .models import Product
# Register your models here.
admin.site.register(Product)
```

**views.py:** Here, the `product_list` render all the product images from the database and pass to the [HTML](#) file for user interface.

### Python



```
1 from django.shortcuts import render
2 from .models import Product
3 from django.http import HttpResponse
4
5
6 def product_list(request):
7     products = Product.objects.all()
8     return render(request, 'index.html', {'products':
9         products})
10
11 def home(request):
12     return HttpResponse('Hello, World!')
```

**urls.py:** Define the URL patterns in the `urls.py` file of the catalog app to map views to URLs.

### Python



```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('/home', views.home, name='home'),
6     path('', views.product_list, name='product_list'),
```

]

Creating an **image gallery** with Django helps in learning about handling media files. To master these features and more advanced functionalities, the [Django Web Development Course](#) will give you the confidence to tackle more complex projects.

## Image Gallery GUI

These HTML templates use Django template language (`{% ... %}`) to render dynamic data from your Django models. In the product list template, it loops through the products queryset and displays each product's name, description, price, and an image. The product detail template shows detailed information about a single product and provides a link to go back to the product list.

Create two folders, one is templates and second is static Folder. create the index.html file in templates folder and paste the below code.

### index.html

#### Python

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Product Catalog</title>
6     <style>
7         /* Add CSS styles for flex container and items */
8         .product-list {
9             display: flex;
10            flex-wrap: wrap; /* Allow items to wrap to the
next row if necessary */
11            justify-content: space-between; /* Space items
evenly along the main axis */
12            list-style: none; /* Remove list styles */
13            padding: 0;
14        }
15
16        .product-item {
```

```

        flex: 1; /* Grow to fill available space evenly
    */
18         max-width: 30%; /* Limit item width to avoid
    overcrowding */
19         margin: 10px; /* Add spacing between items */
20         border: 1px solid #ccc; /* Add a border for
    visual separation */
21         padding: 10px;
22         text-align: center;
23     }
24
25     </style>
26 </head>
27 <body>
28     <h1>Product Catalog</h1>
29
30     <ul class="product-list">
31         {% for product in products %}
32         <li class="product-item">
33             <a href="#">
34                 
35             </a>
36             <h2>{{ product.name }}</h2>
37             <p>{{ product.description }}</p>
38         </li>
39         {% endfor %}
40     </ul>
41 </body>
42 </html>

```

## Setting Up Static and Media Files

Configure Django to serve static and media files during development. In your project's settings.py file, add the following:

```

# Static files (CSS, JavaScript, images)
STATIC_URL = '/static/'
STATICFILES_DIRS = [BASE_DIR / "static"]
# Media files (uploaded user files)
MEDIA_URL = '/media/'

```

```
MEDIA_ROOT = BASE_DIR / "media"
# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

**urls.py:** Add the necessary URL patterns for serving media files in your project's urls.py.

### Python

```
1 from django.contrib import admin
2 from django.urls import path, include
3 from django.conf.urls.static import static
4 from django.conf import settings
5
6
7
8 urlpatterns = [
9     path('admin/', admin.site.urls),
10    path('', include('gallery.urls')),
11 ] + static(settings.MEDIA_URL,
            document_root=settings.MEDIA_ROOT)
```

## Deploy the Project

Create a super user to add data into the data base by the following command:

```
python manage.py createsuperuser
```

```
Username (leave blank to use 'gfg19473@gfg.geeksforgeeks.org'): admin
Email address: admin@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is too short. It must contain at least 11 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Now, Go to the <http://127.0.0.1:8000/admin/> and add the Images, name and its description.

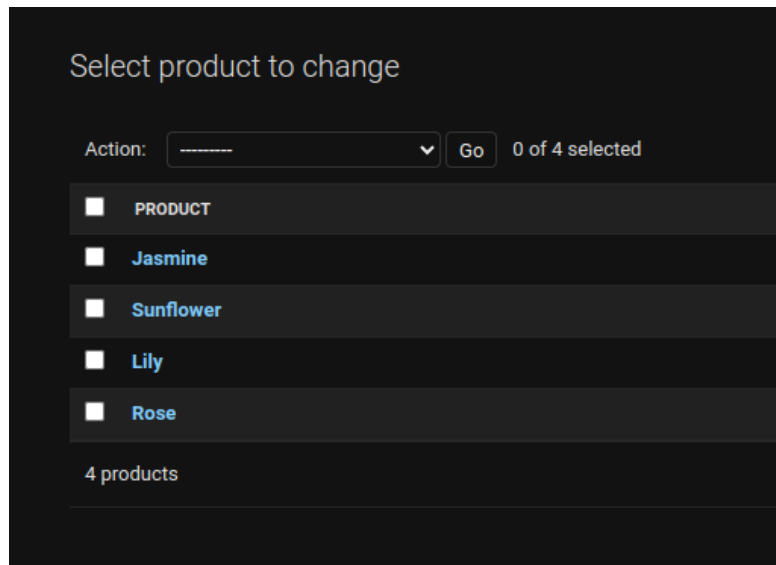
Migrate the data into the database.

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

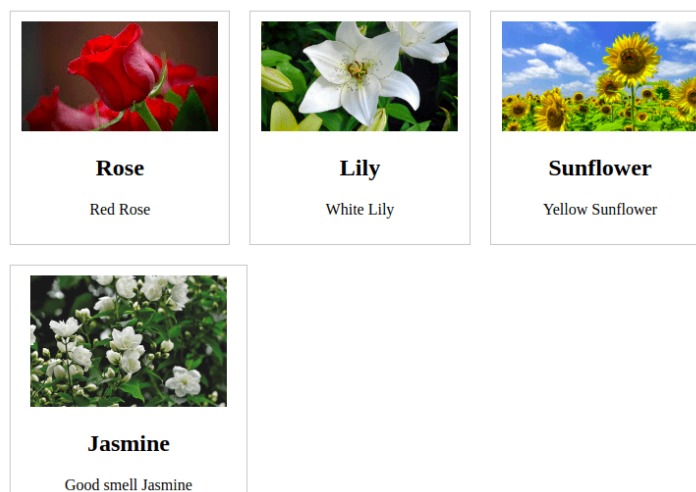
Deploy the project

```
python manage.py runserver
```

Output:



### Product Catalog



Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](https://www.geeksforgeeks.org/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!

A amit...



1

### Previous Article

What does GTE mean in Django Range Query

### Next Article

## Similar Reads

### Create an Image/Video Gallery using React-Native

An Image/Video Gallery is a common feature in mobile applications. This article will guide you for creating an Image/Video Gallery using React Native. We will...

5 min read

### Receipt Print with GUI using Django

In this article, we will guide you through creating a Recipe Print GUI. To achieve this, we first establish a login system, allowing sellers to create accounts through...

11 min read

### Create an Image Gallery App with Vue.js

In this tutorial, we'll walk through the process of building an image gallery application using Vue.js. We'll explore different approaches to displaying image...

3 min read

### Design a Rotating Image Gallery App in HTML CSS & JavaScript

We'll gather some images and build a gallery that can be rotated with straightforward buttons. To rotate the images to the right, we'll use the right...

3 min read

## Create a Portfolio Gallery using Bootstrap

The portfolio gallery is useful when your website contains different types of content or so much content. With the help of a portfolio gallery, you can easily...

3 min read

## Python | Simple GUI calculator using Tkinter

Prerequisite: Tkinter Introduction, lambda function Python offers multiple options for developing a GUI (Graphical User Interface). Out of all the GUI methods,...

6 min read

## Python | ToDo GUI Application using Tkinter

Prerequisites : Introduction to tkinter Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is...

5 min read

## GUI chat application using Tkinter in Python

Chatbots are computer program that allows user to interact using input methods. The Application of chatbots is to interact with customers in big enterprises or...

7 min read

## Build a Basic Form GUI using CustomTkinter module in Python

In this article, we are going to learn how to build a simple form GUI using CustomTkinter in Python. CustomTkinter: It is an extension of the Tkinter modul...

7 min read

## Count number of Faces from Image using Django

This article outlines the process of creating a face-counting feature in images through a Simple Django Web App. The project allows users to count the numb...

6 min read

Article Tags :

[Django](#)[Geeks Premier League](#)[Project](#)[Geeks Premier League 2023](#)





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

[HTML](#)  
[CSS](#)  
[JavaScript](#)  
[TypeScript](#)  
[ReactJS](#)

## 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

[Python Programming Examples](#)  
[Python Projects](#)  
[Python Tkinter](#)  
[Web Scraping](#)  
[OpenCV Tutorial](#)

NextJS

Bootstrap

Web Design

Python Interview Question

Django

Computer Science

DevOps

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

Interview Preparation

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

GeeksforGeeks Videos

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved