



[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

Python | Django Admin Interface

Last Updated : 21 Aug, 2023

In this article, we delve into the capabilities and advantages of the Django Admin Interface, exploring how its customizable features and streamlined workflows empower developers to effortlessly administer their projects, from data management to user interactions.

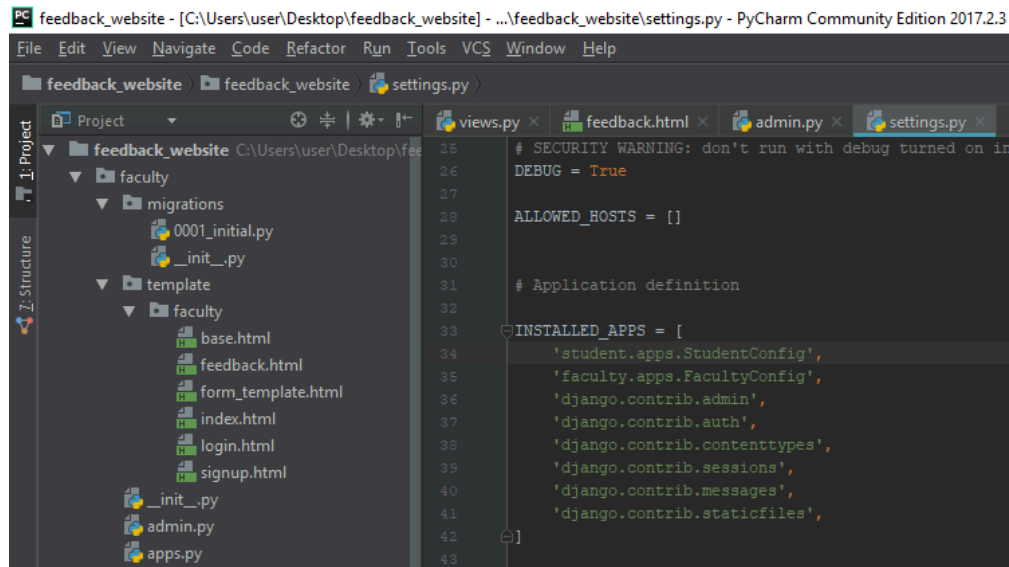
Prerequisites: [django-introduction-and-installation](#) | [django-introduction-set-2-creating-a-project](#)

Django Admin Interface

When you [start a project with Django](#) it also provides a default admin interface that can be used to perform create, read, update, and delete operations on the model directly. It reads a set of data that explains and gives information about data from the model, to provide an instant interface where the user can adjust the contents of the application. This is an in-built module designed to execute admin-related work for the user.

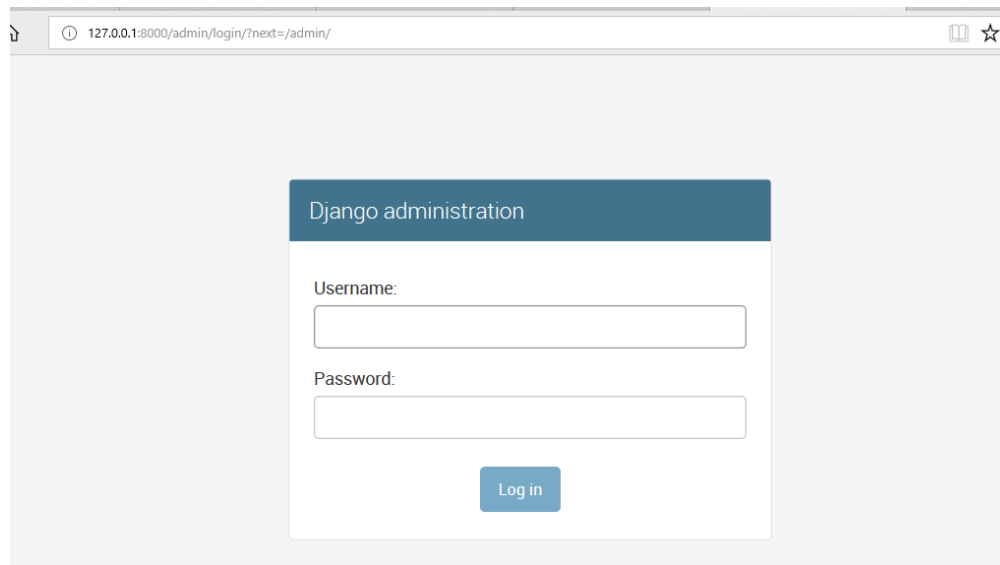
Creating a Superuser in Django Project

The Superuser is the one who can access the Django administration panel and can perform CRUD operation on the created models which are shown in the admin panel. The admin app(django.contrib.admin) is enabled by default and already added to the INSTALLED_APPS list present in the settings.py file.



To access this admin interface on browser write ‘/admin/’ at ‘localhost:8000/admin/’ and it shows the output as given below:

Note: we are assuming that you have deployed your project on local server.



It prompts for login details, if no login id is created before, then a new superuser can be created by using the command given below:

```
python manage.py createsuperuser
```

```

Windows PowerShell
PS C:\Users\user\Desktop\feedback_website> python manage.py createsuperuser
[0x7FFBD68BE350] ANOMALY: meaningless REX prefix used
[0x7FFBD68D5D90] ANOMALY: meaningless REX prefix used
[0x7FFBD99917D0] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBD99B7C60] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBDA1B5AF0] ANOMALY: meaningless REX prefix used
Username: sinny
Email address: sinny@gmail.com
Password:
Password (again):
Superuser created successfully.
PS C:\Users\user\Desktop\feedback_website>

```

Now, access the admin login page after starting the Development Server which can be done by using the command given below.

`python manage.py runserver`

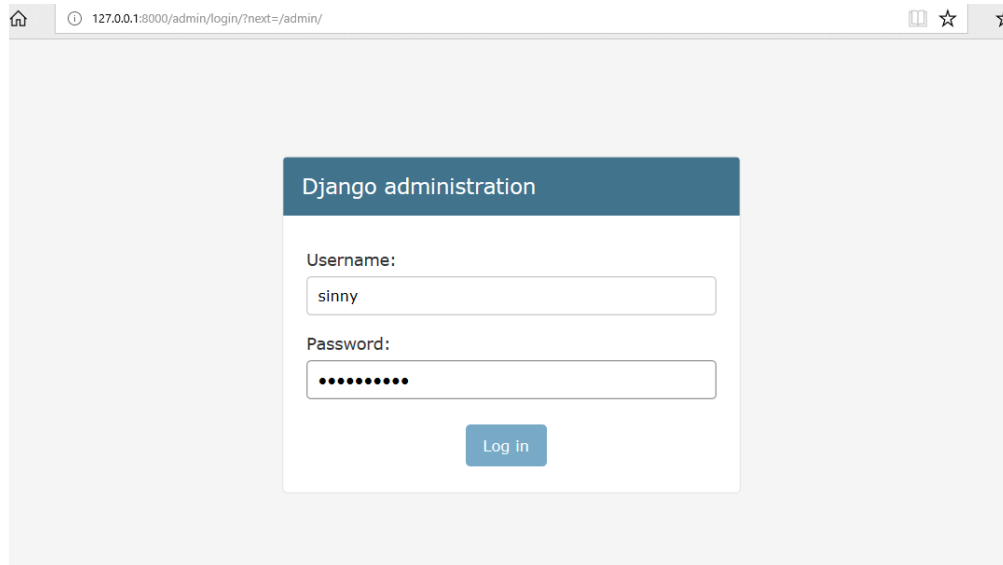
```

Windows PowerShell
PS C:\Users\user\Desktop\feedback_website> python manage.py runserver
[0x7FFBD68BE350] ANOMALY: meaningless REX prefix used
[0x7FFBD68D5D90] ANOMALY: meaningless REX prefix used
[0x7FFBD99917D0] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBD99B7C60] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBDA1B5AF0] ANOMALY: meaningless REX prefix used
[0x7FFBD68BE350] ANOMALY: meaningless REX prefix used
[0x7FFBD68D5D90] ANOMALY: meaningless REX prefix used
[0x7FFBD99917D0] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBD99B7C60] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBDA1B5AF0] ANOMALY: meaningless REX prefix used
Performing system checks...

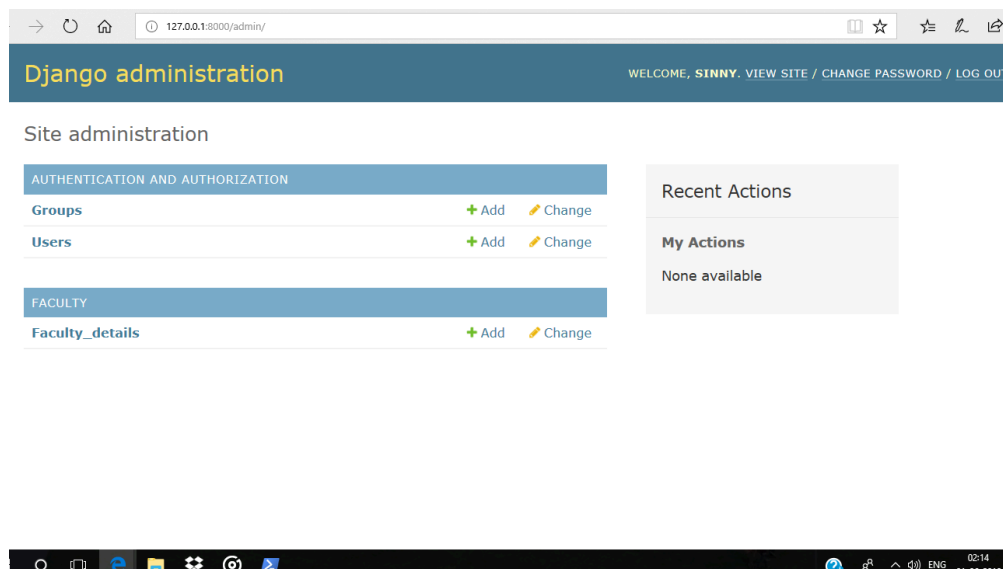
System check identified no issues (0 silenced).
June 01, 2018 - 03:26:56
Django version 1.9, using settings 'feedback_website.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Enter username and password, then hit login .



After logging in successfully, it shows the interface as shown below: .



This is what is called a Django Admin Dashboard where one can add, delete and update data belonging to any registered model.

Reset Django Admin Password

In Django we can reset django admin password but for this this must make sure you already created a superuser already for which we uses ``createsuperuser`` command . This command allows you to create a new superuser account or update an existing one if you already have one, and after creating the superuser as show in the above article then we can use the ``changepassword`` command provided with name of the super user with it and

it will change the superuserpassword this will reset the django admin password.

Before writing the command make sure that current directory must have manage.py file within it and If not Open the terminal and `cd` (change directory) into the project folder (the directory with the `manage.py` file).

Then run the command:

```
python manage.py changepassword <username>
```

Replace `<username>` with the username of the user for whom you want to change the password.

Here's an example of how you would use the `changepassword` command in my case the superuser is `Abhishek_Shakya` so command would be like

```
python manage.py changepassword Abhishek_Shakya
```

```
C:\Users\GFG0263\Desktop\Django-Projects>myprojects\Scripts\activate
(myprojects) C:\Users\GFG0263\Desktop\Django-Projects>cd my_project
(myprojects) C:\Users\GFG0263\Desktop\Django-Projects\my_project>python manage.py changepassword Abhishek_Shakya
Changing password for user 'Abhishek_Shakya'
Password:
Password (again):
Password changed successfully for user 'Abhishek_Shakya'
(myprojects) C:\Users\GFG0263\Desktop\Django-Projects\my_project>
```

Changepassword of the superuser

and by this we can reset or update the superuser password

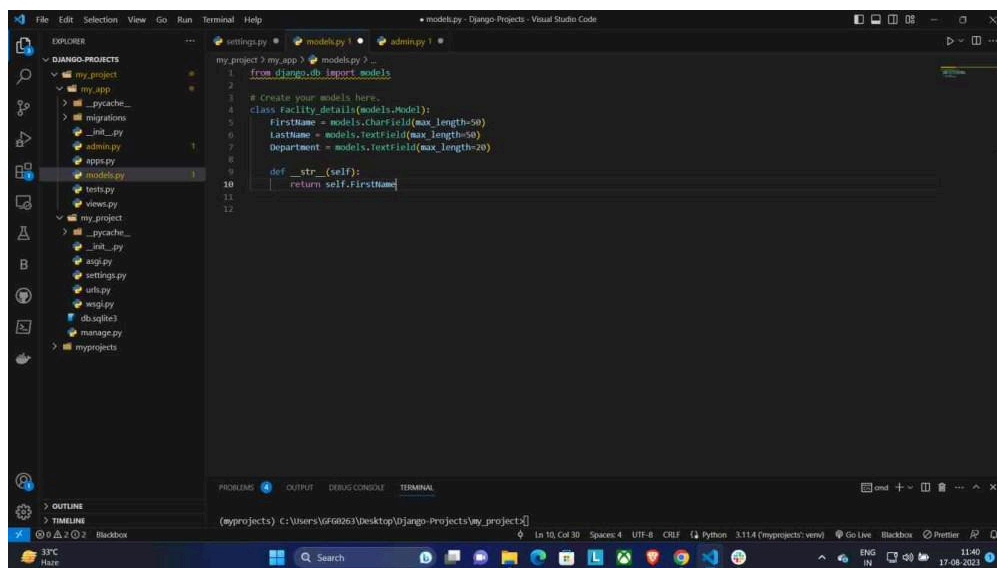
Register Model in Django Admin

After starting your project in django and creating or resetting your django superuser you can now access the django administrator and through django admin panel you can perform CRUD operations on your models but the model you created in your app only shows in the admin panel when you register them in your `'admin.py'` of the app you created.

after stating the project create an app

```
python manage.py createapp myapp
```

Now in the myapp folder we have the file models.py in which we create our model and after creating those model we need to register them in the admin.py file of the myapp folder after that we makemigrations of the model created.



register model in django-admin

and the after model creation you have to register the model in django by registering them into admin.py file of `myapp` folder

first import your model at the top of the admin.py file if the model you are importing is in the same file you can import by

```
from .models import <model.name>
```

In this case the model is Facility_details, so

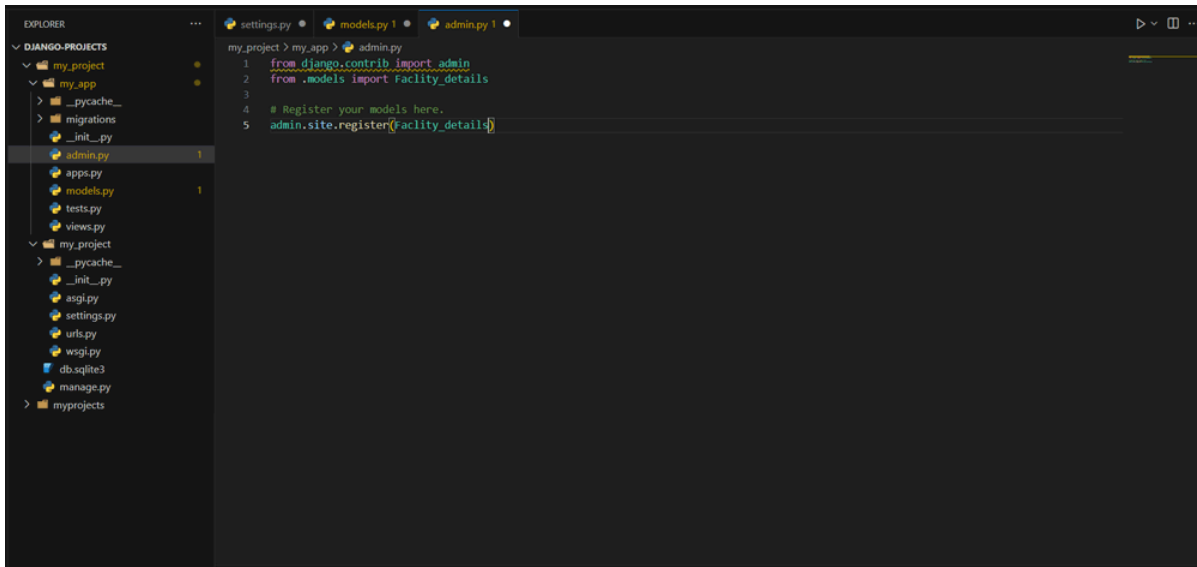
```
from .models import Facility_details
```

Now we have to register the model in the admin.py file so that it can be visible on the admin panel

```
admin.site.register(<Model.name>)
```

In this case the model is Facility_details, so

```
admin.site.register(Facility_details)
```



```

1 from django.contrib import admin
2 from .models import Faculty_details
3
4 # Register your models here.
5 admin.site.register(Faculty_details)

```

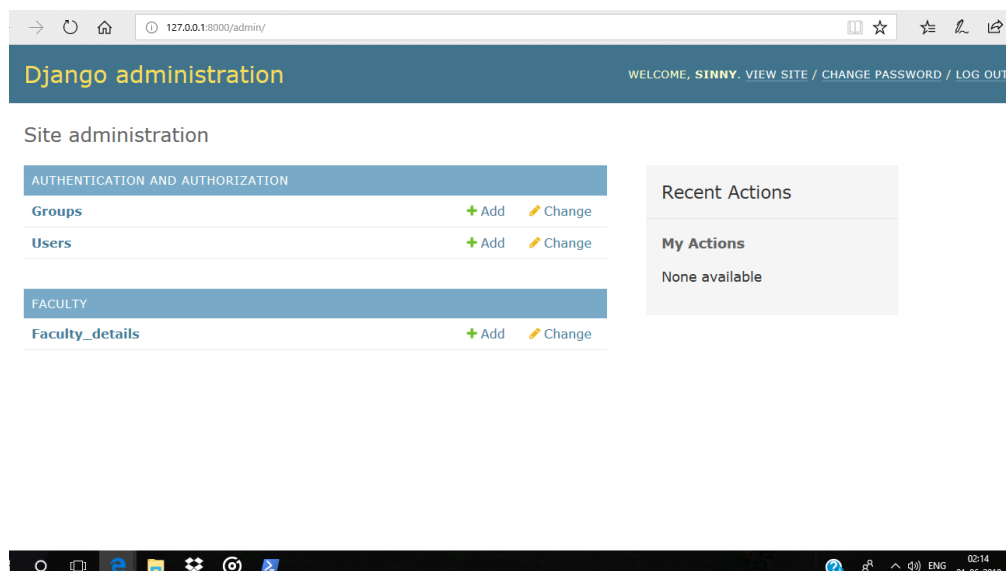
Register model in admin

python manage.py makemigrations

after that migrate all the migrations

python manage.py migrate

Now the admin page will have the model Faculty_details but make sure the server is up and running.



Now you are all set to go.



ankth...



14

Previous Article

URL fields in serializers - Django REST Framework

Next Article

Handling Ajax request in Django

Similar Reads

Render Model in Django Admin Interface

Rendering model in admin refers to adding the model to the admin interface so that data can be manipulated easily using admin interface. Django's ORM...

3 min read

Customize Django Admin Interface

Django admin by default is highly responsive GUI, which offers various features and an overall CRUD application to help developers and users. Moreover, Djang...

3 min read

Disable Admin-StyleBrowsable Interface of Django REST Framework

Django REST Framework (DRF) is a powerful and flexible toolkit for building Web APIs in Django. One of its features is the browsable API, which provides a user-...

3 min read

Django Admin - Redesign and Customization

Often the default Django admin page is great and it is all we need for our projects but sometimes we may want to expand what default admin interface can do. W...

4 min read

Securing Django Admin login with OTP (2 Factor Authentication)

Multi factor authentication is one of the most basic principle when adding security for our applications. In this tutorial, we will be adding multi factor authentication...

2 min read

How to Show a Many-to-Many Field with "list_display" in Django Admin

In Django, the list_display attribute in the Django Admin interface allows developers to specify which fields to display in the list view of the admin panel....

3 min read

How to Override CSS in Django Admin?

Django's admin interface is a powerful tool for managing application data. It comes with a default styling that may not always fit the visual identity of your...

3 min read

Making a Django Admin Field Non-Required Without Custom Form Creation

The Django admin interface is a great tool for managing data in your Django application. Sometimes, you may need to make some fields optional in the admi...

2 min read

Django Admin - Disable the 'Add' Action for a Specific Model

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It's designed to help developers take...

3 min read

How to Override and Extend Basic Django Admin Templates?

The Django admin interface provides a robust way to manage our application's data, offering a user-friendly and efficient design. By default, it gives a clean and...

10 min read

Article Tags : [Python](#) [Python Django](#)

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

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

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
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