Create a contact form with Django and SQLite

Pratik Sah

Aug 21, 2020 · 31 min read

Welcome back. In this post, we'll be using **HTML** template and **Django** to create a contact form with Django and SQLite. We will also try to see all the listings in the admin panel.

This is a basic Django web application focusing on some basic crud and we will be saving the data in an SQLite database.

Why Django?

> *Django is a free and open-source python-based web framework for creating dynamic web apps. Some popular sites that use Django are Instagram, Mozilla, The Washington Post, etc.*

The Web framework for perfectionists with deadlines | Django!

We will use bootstrap for styling the form. For using the bootstrap, we can use either **CDN** or we can download the CSS files from the bootstrap site.

Below is the code for a simple Form made using bootstrap.

COPY

```
<!doctype html>
<html lang="en">
```

```
    <meta name="viewport" content="width=device-width, initial-

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn
      integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9Hu

    <title>Contact</title>
  </head>

  <body>

    <h1 class="text-center my-5">Contact Form</h1>
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-lg-6">
          <form action="" method="POST">
            <div class="form-group">
              <label for="name">Full Name</label>
              <input type="text" class="form-control" name="nam
            </div>
            <div class="form-group">
              <label for="email">Email address</label>
              <input type="email" name="email" class="form-cont
            </div>
            <div class="form-group">
              <label for="message">Your message</label>
              <textarea class="form-control" name="desc" id="me
            </div>
            <input type="submit" class="btn btn-primary" value=
          </form>
        </div>
      </div>
```

code.

To submit our form, I have used *input type submit* but you can also use the button tag in place of input type submit.

In place of form action, we will try to submit the form in the same path that will be the root URL and we will handle form submission in our code.

This code will help us in creating a form which we will be using it in the templates of our **Django** project.

Now we are done with the HTML, and now, we can start with the Django project setup.

Note: *Before starting with Django, we will create a virtual environment where we will have all our files and packages stored.*

Creating a Virtual Environment

Before starting with creating a virtual environment, we need to install a package that will help us in creating the environment.

> *A virtual environment is a tool to create a private workspace for a Python application.*



We have many python packages for creating the virtual environment. Here I'll be using the `pipenv`.


Like

The main reason for using `pipenv` is that its commands are the same on all the platforms.

To install this package on your system, copy the below command and paste it in your terminal.

COPY

```
# for mac and Linux users
pip3 install pipenv

# for windows users
pip install pipenv
```

terminal This package automatically creates and manages virtual env for your projects, as well as adds/remove packages from your pipfile as you install/remove packages.

Now this package is installed on your system. The next step is to

terminal `mkdir myenv` will help you in creating a new directory, and `cd myenv` will help you to change the working directory to `myenv`.

Like

Now we are inside our folder where we want to create the env, run the following command to create a new environment inside `myenv` directory.

COPY

```
# command to create a new virtual env
pipenv shell
```

terminal This command will **create** and **activate** a new virtual environment in your present directory, and now you are ready to install new packages in your env.

But wait. You can't install any packages simply by writing `pip install`.

```
# this will install Django in your virtual env
pipenv install Django
```

terminal To check the installed packages, type the following command, and it will list all the installed packages in your terminal.

COPY

```
pip freeze
```

terminal You'll see the list of all the installed packages in your terminal, and it will also have Django in it.

Next is to change the directory to `contactform` and start our server.

Like

COPY

```
# to change the working directory
cd contactform

# to see the files and folders
ls
```

Since we are done with the project setup and our server is started now, we can move on to creating our `app` for the contact form.

Like

You might be thinking that we have already created a project above using the `django-admin` command, then why are we going to create a new app.

But before moving forward, let's revise the points as they are very confusing at the beginning.

Like

Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like




Like

MORE ARTICLES

Pratik Sah

Like

Handle file uploads in Node JS with Formidable

While handling file uploads via the form in Express, you might have tried Multer. It is good but For...

Pratik Sah

Log your Node Js error in a log file like a PRO!

We all have used console.log() at some point in time but these are just for debugging purpose only. ...

Pratik Sah

Like

Let’s create a dictionary with vanilla javascript

Hi, welcome back. Last night I was making a video for my Javascript playlist and it was the last vid...

Comments

+ Write a comment

  

 Like