

Start coding or [generate](#) with AI.

Create a registration form with the help of django/flask. And store the data in Google excel sheet.

```
pip install django
```

```
Collecting django
  Downloading Django-5.0.6-py3-none-any.whl (8.2 MB)
    _____ 8.2/8.2 MB 32.2 MB/s eta 0:00:00
Collecting asgiref<4,>=3.7.0 (from django)
  Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Requirement already satisfied: sqlparse>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from django) (0.5.0)
Requirement already satisfied: typing-extensions>=4 in /usr/local/lib/python3.10/dist-packages (from asgiref<4,>=3.7.0->django) (4.12.0)
Installing collected packages: asgiref, django
Successfully installed asgiref-3.8.1 django-5.0.6
```

Create a new Django project:

```
django-admin startproject registration_project
```

Create a new Django app within the project:

```
cd registration_project
python manage.py startapp registration
```

Now, let's create the registration form:

In the registration app, create a file named forms.py:

```
from django import forms

class RegistrationForm(forms.Form):
    first_name = forms.CharField(max_length=100)
    last_name = forms.CharField(max_length=100)
    email = forms.EmailField()
    # Add more fields as needed
```

In the registration app, create a file named views.py:

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

def register(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            # Process form data, save to Google Sheets
            # You'll need to implement this part using Google Sheets API
            # Example: https://developers.google.com/sheets/api/quickstart/python
            return render(request, 'registration/success.html')
    else:
        form = RegistrationForm()
    return render(request, 'registration/register.html', {'form': form})
```

Create HTML templates for the registration form and success page: Create a folder named templates in the registration app. Inside the templates folder, create another folder named registration. Create two HTML files: register.html and success.html inside the registration folder.

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration Form</title>
</head>
<body>
  <h2>Registration Form</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

File "["<ipython-input-4-c901826cbe46>"](#), line 1
 <!DOCTYPE html>
 ^
 SyntaxError: invalid syntax

Next steps: [Fix error](#)

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration Successful</title>
</head>
<body>
  <h2>Registration Successful</h2>
  <p>Thank you for registering!</p>
</body>
</html>
```

Add URLs for the registration form: In the registration app, create a file named urls.py if it doesn't exist. Add the following code: python

```
from django.urls import path
from . import views

urlpatterns = [
    path('register/', views.register, name='register'),
]
```

Include the registration URLs in the project's urls.py:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('registration/', include('registration.urls')),
]
```

T **B** **I** **<>** **↔** **🖼️** **”** **☰** **☷** **—** **ψ** **😊** **☰**

Now, you have a basic registration form set up with Django. Next, you'll need to integrate Google Sheets API to store the form data in a Google Excel sheet. You can follow the Google Sheets API documentation for Python to accomplish this. Once you have the code to store data in Google Sheets, you can integrate it into the view's register function.

Now, you have a basic registration form set up with Django. Next, you'll need to integrate Google Sheets API to store the form data in a Google Excel sheet. You can follow the Google Sheets API documentation for Python to accomplish this. Once you have the code to store data in Google Sheets, you can integrate it into the view's register function.

