Build a Django web application where users:

* Upload a **CSV** file
* Perform **Pandas** analysis (summary, filters)
* Visualize data using **Seaborn/Matplotlib**
* See results in browser

**🔧 Components Overview:**

| **Component** | **Description** |
| --- | --- |
| **Django Views** | Handle requests: file upload, processing, rendering |
| **Pandas** | Used in views or helpers to load and analyze data |
| **Matplotlib/Seaborn** | Generate plots from CSV, save as images or base64 |
| **Templates (HTML)** | Render upload form, show tables and images |
| **Static Files** | Host images, CSS, JS |
| **Media Folder** | Temporarily store uploaded CSV files |

**📁 Project Structure**

csharp

CopyEdit

csvanalyzer/

├── analyzer/ # Django app

│ ├── views.py

│ ├── urls.py

│ ├── forms.py

│ ├── templates/

│ │ └── index.html

├── media/ # Uploaded CSVs

├── static/ # Output plots

├── csvanalyzer/ # Django project settings

├── manage.py

**⚙️ Step-by-Step Setup**

**1. ✅ Create Django project and app**

bash

CopyEdit

django-admin startproject csvanalyzer

cd csvanalyzer

python manage.py startapp analyzer

**2. ✅ Add to settings.py**

python

CopyEdit

# csvanalyzer/settings.py

INSTALLED\_APPS = [

...

'analyzer',

]

MEDIA\_URL = '/media/'

MEDIA\_ROOT = BASE\_DIR / 'media'

STATIC\_URL = '/static/'

STATICFILES\_DIRS = [ BASE\_DIR / 'static' ]

**3. ✅ Create forms.py for CSV upload**

python

CopyEdit

# analyzer/forms.py

from django import forms

class CSVUploadForm(forms.Form):

file = forms.FileField(label='Upload your CSV file')

**4. ✅ Create views.py to handle logic**

python

CopyEdit

# analyzer/views.py

from django.shortcuts import render

from .forms import CSVUploadForm

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import os

import uuid

def index(request):

table\_html = None

plot\_path = None

if request.method == 'POST':

form = CSVUploadForm(request.POST, request.FILES)

if form.is\_valid():

csv\_file = request.FILES['file']

df = pd.read\_csv(csv\_file)

# Save preview table

table\_html = df.head().to\_html(classes="table table-striped")

# Plot and save

plt.figure(figsize=(10, 6))

sns.histplot(df.select\_dtypes(include='number').iloc[:, 0])

plot\_id = uuid.uuid4().hex

plot\_path = f'static/plot\_{plot\_id}.png'

plt.savefig(plot\_path)

plt.close()

else:

form = CSVUploadForm()

return render(request, 'index.html', {

'form': form,

'table\_html': table\_html,

'plot\_path': plot\_path

})

**5. ✅ Create urls.py in app and main**

python

CopyEdit

# analyzer/urls.py

from django.urls import path

from .views import index

urlpatterns = [ path('', index, name='home') ]

python

CopyEdit

# csvanalyzer/urls.py

from django.contrib import admin

from django.urls import path, include

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('analyzer.urls')),

] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

**6. ✅ Create templates/index.html**

html

CopyEdit

<!-- analyzer/templates/index.html -->

<!DOCTYPE html>

<html>

<head>

<title>CSV Analyzer</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

</head>

<body class="container mt-5">

<h2>Upload CSV and Analyze</h2>

<form method="POST" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="btn btn-primary">Analyze</button>

</form>

{% if table\_html %}

<hr>

<h3>Data Preview</h3>

{{ table\_html|safe }}

{% endif %}

{% if plot\_path %}

<hr>

<h3>Plot</h3>

<img src="/{{ plot\_path }}" class="img-fluid" />

{% endif %}

</body>

</html>

**7. ✅ Run server and test**

bash

CopyEdit

python manage.py migrate

python manage.py runserver

Open <http://localhost:8000>

**🚀 Possible Enhancements**

* Select column for plotting
* Generate multiple plot types (scatter, boxplot, heatmap)
* Export results as Excel
* Interactive charts with **Plotly**
* Add login to save reports