Wikipedia search app Project using Django

Last Updated : 25 May, 2022

Django is a high-level Python based Web Framework that allows rapid development and clean, pragmatic design. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLlite3, etc. Today we will create joke app in django.

In this article we will make the wikipedia search app using django. For searching on wikipedia we will use “wikipedia” library in python

Django Views Model Template Forms Jinja Python SQLite Flask Json Postman Interview Ques

Creating Django Project –

First we have to install django

Ubuntu

```
pip install django
```

Then install wikipedia library

```
pip install wikipedia
```

Lets create new django project

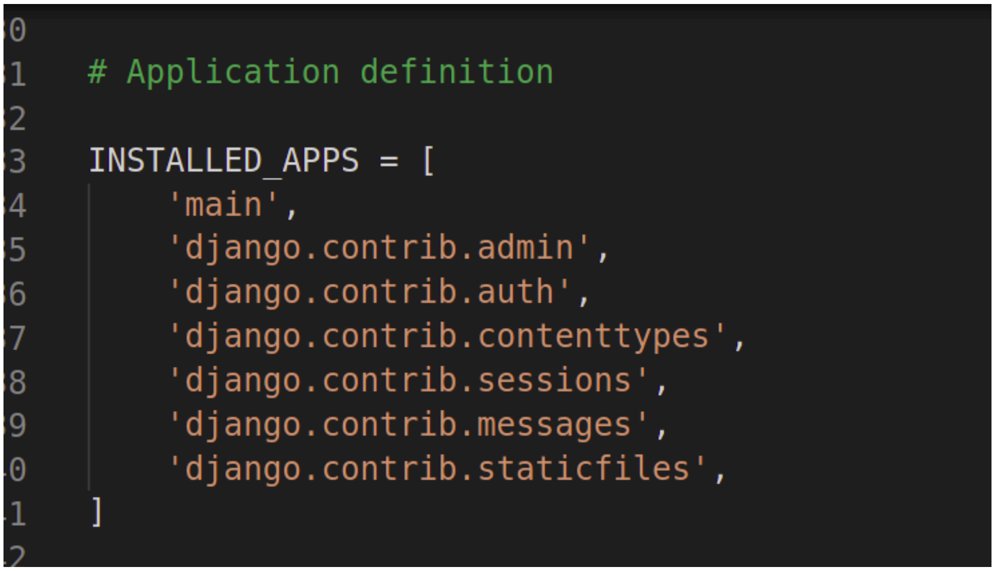
```
django-admin startproject wikipedia_app
```

```
cd wikipedia_app
```

Then create new app in django project

```
python3 manage.py startapp main
```

Then add app name inside the **settings.py** inside INSTALLED_APPS



```
# Application definition

INSTALLED_APPS = [
    'main',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

views.py

Python3

```
from django.shortcuts import render,HttpResponse
import wikipedia


# Create your views here.
def home(request):
    if request.method == "POST":
        search = request.POST['search']
        try:
            result = wikipedia.summary(search,sentences = 3) #No of sentences tha
        except:
            return HttpResponse("Wrong Input")
        return render(request,"main/index.html",{"result":result})
    return render(request,"main/index.html")
```

Create new directory **templates** inside that create new directory **main**

Inside that create new file **index.html**

index.html

HTML

```
<!DOCTYPE html>
```

```
<html>
<head>
    <title>GFG</title>
</head>
<body>
    <h1>Wikipedia Search</h1>
    <form method="post">
        {% csrf_token %}
        <input type="text" name="search">
        <button type="submit">Search</button>
    </form>
    {% if result %}
        {{result}}
    {% endif %}
</body>
</html>
```

Create new file **urls.py** inside the main app

Python3

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

urlpatterns = [
    path('', home,name="home"),
]
```

wikipedia_app/urls.py

Python3

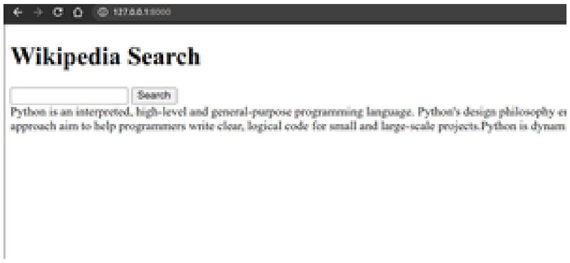
```
from django.contrib import admin
from django.urls import path,include

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

To run this app open cmd or terminal

```
python3 manage.py runserver
```

Output :-



Wikipedia Search

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy e
approach aim to help programmers write clear, logical code for small and large-scale projects.Python is dynam

Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our 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!

V vivek... 3

Next Article

Translator App Project using Django

Similar Reads

Wikipedia search app using Flask Framework - Python

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. Flask is a...

2 min read

Voice search Wikipedia using Python

Every day, we visit so many applications, be it messaging platforms like Messenger, Telegram or ordering products on Amazon, Flipkart, or knowing abo...

5 min read

Autocomplete Search using jQuery and Wikipedia OpenSearch API

In web design, the autocomplete feature is a common one. When the user types some value in the search text box, it automatically shows a relevant list of...

2 min read

Create a Wikipedia Search using HTML CSS and JavaScript

In this article, we're going to create an application, for searching Wikipedia. Using HTML, CSS, and JavaScript, users will be able to search for articles on Wikipedi...

3 min read

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our djang...

2 min read

Translator App Project using Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it...

2 min read

Web scraping from Wikipedia using Python - A Complete Guide

In this article, you will learn various concepts of web scraping and get comfortable with scraping various types of websites and their data. The goal is t...

9 min read

Wikipedia Summary Generator using Python Tkinter

Prerequisite: Tkinter Wikipedia Python offers multiple options for developing a GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is the most...

2 min read

Scraping Wikipedia table with Pandas using read_html()

In this article, we will discuss a particular function named read_html() which is used to read HTML tables directly from a webpage into a Pandas DataFrame...

3 min read

Wikipedia Summary Generator Card using Tailwind CSS & JavaScript

The Wikipedia Summary Generator is a web application that utilizes Tailwind CSS, enabling users to search for a topic and receive a summary from Wikipedia...

3 min read

Article Tags : Python Web Technologies Django-Projects Python Django +1 More

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

Languages

Python
Java
C++
PHP

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

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

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

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

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

GeeksforGeeks Videos

DSA

Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

Python
Java
C++
Web Development
Data Science
CS Subjects