Python is a high-level, interpreted programming language known for its readability and simplicity. It was created by Guido van Rossum and first released in 1991. Here are some key features of Python:

1. **Readability**: Python's syntax is clean and easy to understand, making it an excellent choice for beginners.
2. **Versatility**: Python is used in various domains, including web development, data analysis, artificial intelligence, scientific computing, and automation.
3. **Large Standard Library**: Python has a rich set of libraries and frameworks (like NumPy, Pandas, Flask, and Django) that extend its functionality and make development faster.
4. **Cross-Platform**: Python runs on various operating systems, including Windows, macOS, and Linux, which makes it versatile for different environments.
5. **Community Support**: Python has a large and active community, providing extensive resources, tutorials, and third-party modules.
6. **Dynamic Typing**: Python uses dynamic typing, which means you don't have to declare variable types explicitly.
7. **Interpreted Language**: Python code is executed line by line, which makes debugging easier but can impact performance compared to compiled languages.

Overall, Python is favored for its ease of use, making it popular among beginners and professionals

For project purpose

Library used

1 numpy : dimensional array create python have list

2 opencv: face recognition purpose

3. django : framework

4. pandas: large amount of data

What is NumPy?

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

## Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

## Why is NumPy Faster Than Lists?

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behavior is called locality of reference in computer science.

This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

Which Language is NumPy written in?

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

Where is the NumPy Codebase?

The source code for NumPy is located at this github repository <https://github.com/numpy/numpy>

Opencv: it works

* [**Introduction to OpenCV.js**](https://docs.opencv.org/4.x/df/df7/tutorial_js_table_of_contents_setup.html)

Learn how to use OpenCV.js inside your web pages!

* [**GUI Features**](https://docs.opencv.org/4.x/df/d04/tutorial_js_table_of_contents_gui.html)

Here you will learn how to read and display images and videos, and create trackbar.

* [**Core Operations**](https://docs.opencv.org/4.x/d1/d78/tutorial_js_table_of_contents_core.html)

In this section you will learn some basic operations on image, some mathematical tools and some data structures etc.

* [**Image Processing**](https://docs.opencv.org/4.x/d2/df0/tutorial_js_table_of_contents_imgproc.html)

In this section you will learn different image processing functions inside OpenCV.

* [**Video Analysis**](https://docs.opencv.org/4.x/de/db6/tutorial_js_table_of_contents_video.html)

In this section you will learn different techniques to work with videos like object tracking etc.

* [**Object Detection**](https://docs.opencv.org/4.x/dc/d73/tutorial_js_table_of_contents_objdetect.html)

In this section you will object detection techniques like face detection etc.

* [**Deep Neural Networks (dnn module)**](https://docs.opencv.org/4.x/d0/db7/tutorial_js_table_of_contents_dnn.html)

These tutorials show how to use dnn module in JavaScript

The architecture of Django is based on the Model-View-Template (MVT) software design pattern, which is an alternative to the Model-View-Controller (MVC) framework. The MVT pattern separates the application into three distinct layers:

* Models: Handle the data logic and database structure
* Views: Handle the application logic and functionality
* Templates: Handle the layout and structure of the user-facing application

Django is a Python framework for web development with many features, including:

* Rapid development

Django's structure provides a ready-to-use framework for common web development tasks, such as user authentication, content administration, and site maps.

* Security

Django's templates automatically escape common HTML characters in user-entered fields. It also has a built-in authentication system to restrict access to authorized users only.

* Scalability

Django apps can manage user sessions, so you can add more instances of your application without losing data.

* Database management

Django uses an object-relational mapper (ORM) to convert traditional database structure into Python classes.

* Versatility

Django can be used to build almost any type of website, including content management systems, wikis, social networks, and news sites.

* Support for most operating systems

Django supports most operating systems.

* Facet filters

Django 5.0 includes facet filters, which display facet counts for active filters in the admin change list.

* Database-computed default values

Django 5.0 also includes a Field.db\_default parameter that allows you to define default values using database functions.

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

## Why Use Pandas?

Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science.

What Can Pandas Do?

Pandas gives you answers about the data. Like:

* Is there a correlation between two or more columns?
* What is average value?
* Max value?
* Min value?

Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called *cleaning* the data.

Where is the Pandas Codebase?

The source code for Pandas is located at this github repository <https://github.com/pandas-dev/pandas>

<https://www.geeksforgeeks.org/djnago-installation-and-setup/>

install python in your system using python exe file

install pip using command prompt

**python -m pip install -U pip**

### **Step 2: Set Virtual environment:**

Setting up the virtual environment will allow you to edit the dependency which generally your system wouldn’t allow. Follow these steps to set up a virtual environment-

### **Step 3: Create virtual environment in Django**:

We should first go the directory where we want to create the [virtual environment](https://www.geeksforgeeks.org/python-virtual-environment/) then we type the following command to create virtual environment in django.

python -m venv env\_site

Step 4: Activate the virtual environment:

Run the activation script located in the bin directory within the virtual environment folder

For Windows:

.\env\_site\Scripts\activate.ps1

Step 5: Install Django:

Install django by giving following command

pip install django

Django Setup

Once Django is installed, we can start to create a new Django project.

Step 1: Start a new Django Project

Start a project by following command-

django-admin startproject geeks\_site

Step 2: Navigate to the Project Directory

Change directory to geeks\_site

cd geeks\_site

Step 3: Start the server

Start the server by typing following command in cmd-

python manage.py runserver

### Step 4: Verify Server Status

To check whether server is running or not go to web browser and enter **http://127.0.0.1:8000/**

Crud operation you can run on following github

<https://github.com/shiyunbo/django-crud-example>

go to folder and install pip and run following

python manage.py runserver

<http://127.0.0.1:8000/tasks/>

check crud functionality

Python folder structure

settings.py

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-5j2b%a2qy34jf3gsry0j3@akvnz(ikjm3v0w@m5ku7=icul+qw'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'geeks\_site.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'geeks\_site.wsgi.application'

# Database

# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

# Password validation

# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/5.1/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC\_URL = 'static/'

# Default primary key field type

# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

In this file all the configuration is done

1. Secret key
2. Define host which are allowed
3. Application definition
4. Middleware are defined
5. Templates are defined
6. Database is defined
7. Authentication password validators
8. Language code is defined
9. Time zone is defined
10. Static url is defined

Urls.py

from django.contrib import admin

from django.urls import path

urlpatterns = [

    path('admin/', admin.site.urls),

]

In this url paths are given to navigate the site

Asgi.py

It is used for asynchronous tasks and require to get into deep

https://www.geeksforgeeks.org/deploy-an-asgi-django-application/

manage.py

To execute the project manage.py is used by using below command

python manage.py runserver

db.sqllite is integrated generally by default.

Python works on MVT architecture

M->Model

v->View

T-> Template

M -> model layer is defined in it

V -> view works like controller business logic defined in it

T-> T is used for template html is defined in it.

In crud operation

<https://github.com/shiyunbo/django-crud-example>

tasks is considered as component in python

models.py

it is considered to define models

urls.py

it is used to define urls

views.py

to define the views

tasks/templates

html files are defined in it

In migrations

Models are defined when you run this the data is created into the database

Models.py : models are defined in it