Python Virtual Environment Setup (Windows)

FOR WINDOWS

Introduction

A Python Virtual Environment (venv) is an isolated environment used to manage project-specific dependencies without affecting the global Python installation.

Requirements

- Windows OS

- Python 3.x installed (**New Version**)

- Command Prompt or PowerShell

Check Python installation:

python --version

Steps to Create Virtual Environment

1. **Open Command Prompt**

2. **Navigate to project directory**

   cd C:\Users\Admin\Desktop\my_project

3. Create virtual environment

   python -m venv venv

Note- (Can give any name to your virtual Environment

For Example: - My Env……)

4. Activate environment

   venv\Scripts\activate

5. **Upgrade pip**

   python -m pip install --upgrade pip

6. **Install packages**

pip install flask pymongo

7. Freeze dependencies

pip freeze > requirements.txt

8. Deactivate environment

deactivate

Folder Structure

my_project/

├─ venv/

├─ app.py

├─ requirements.txt

├─ Readme

FOR LINUX

1. **Check Python Installation (Linux)**

python3 --version

Note:- **If not installed:**

sudo apt update

sudo apt install python3 python3-venv python3-pip

2. **Navigate to Project Directory**

cd /home/user/my_project

3. **Create Virtual Environment**

python3 -m venv venv

4. **Activate Virtual Environment**

source venv/bin/activate

5. **Upgrade pip**

   pip install --upgrade pip

6. **Install Dependencies**

   pip install flask pymongo

7. **Freeze Dependencies**

   pip freeze > requirements.txt

8. **Deactivate Environment**

   deactivate

9. **Recommended Project Structure**

   my_project/

   ├── venv/

   ├── app.py

   ├── requirements.txt

   └── .env

10. **Common Errors & Solutions**

• Python not found:

Install Python and add it to PATH

• Permission denied (Linux):

Use sudo or check directory permissions

• Virtual environment already exists:

Delete existing venv or create with new name