Tops Technology

Module 16)
Python DB and Framework

Presented By : Nandni Vala

Virtual Environment

1.Understanding the importance of a virtual environment in Python projects.

➢ A virtual environment in Python is a tool that allows you to create an isolated workspace for your Python projects.

➢ **Dependency Management**

➢ **Isolated Dependencies:** Each project can have its own dependencies and versions without affecting other projects.

➢ **Reproducibility**

➢ **Consistent Setup:** The virtual environment allows you to recreate the same environment on another system using the requirements.txt file.

➢ **Clean Global Environment**

➢ **No Global Pollution:** Virtual environments prevent cluttering the global Python environment with project-specific packages.

- **Compatibility**

- **Multiple Python Versions:** A virtual environment can use different Python versions for different projects, making it easier to work on legacy projects or experiment with newer features.

- **Best Practices**

- **Industry Standard:** Using a virtual environment is a standard practice in Python development, demonstrating professionalism and good habits.

  - **Creating and Using a Virtual Environment**

- **Create:**

- python -m venv myenv

- **Activate:**

- On Windows:

- Myenv\Scripts\activate

- **Install Dependencies:**

- pip install <package-name>

2.Using venv or virtualenv to create isolated environments.

➢ ou can use **venv** or **virtualenv** to create isolated Python environments. Here's how:

➢ **Using venv**

➢ **Create the Environment**:

➢ python -m venv myenv

➢ **Activate the Environment**:

➢ myenv\Scripts\activate

➢ **Using virtualenv**

➢ **Install virtualenv**:

➢ pip install virtualenv

➢ **Create the Environment**:

➢ virtualenv myenv

➢ **Activate the Environment**:

➢ myenv\Scripts\activate

➢ **Key Differences**

➢ **venv**: Built into Python 3.3+; lightweight and simple.

➢ **virtualenv**: Requires installation; works with older Python versions and offers additional features.