Tops Technology

Module 16) Python DB and Framework

Presented By : Nandni Vala

Django Introduction

1.Overview of Django: Web development framework.

- Django is a high-level, open-source Python web framework designed for rapid development and clean, pragmatic design. It encourages reusability, scalability, and secure web application development.

Key Features:

- **Full-Stack Framework**: Includes tools for both frontend and backend development.
- **Batteries Included**: Comes with built-in features like ORM (Object-Relational Mapping), authentication, and admin interface.
- **Scalability**: Suitable for handling both small projects and large-scale applications.
- **Secure by Design**: Provides protection against common security threats like CSRF, XSS, and SQL injection.
- **Versatile**: Can be used for various types of projects, from content management systems to e-commerce platforms.

Core Components:

- **Models**: Define database schema and act as an interface to the database.
- **Views**: Handle business logic and request processing.
- **Templates**: Render HTML content with dynamic data.
- **URLs**: Map specific URLs to corresponding views.
- **Admin Interface**: Auto-generate a web-based interface for managing app data.

Advantages:

- **Rapid Development**: Reduces development time with built-in components.
- **DRY Principle**: Promotes "Don't Repeat Yourself" for efficient code reuse.
- **Wide Ecosystem**: Large community support and rich plugin ecosystem.

Use Cases:

- E-commerce sites
- Social media platforms
- API backends

2.Advantages of Django (e.g.,scalability, security).

- **Advantages of Django**
- **Scalability :**
  - Django is designed to handle both small-scale and large-scale projects.
  - Features like database optimization, caching, and load balancing make it suitable for high-traffic applications (e.g., Instagram).
- **Security**
  - Django provides built-in protection against common security threats, including:
    - **SQL Injection**
    - **Cross-Site Scripting (XSS)**
    - **Cross-Site Request Forgery (CSRF)**
    - **Clickjacking**
  - It also ensures safe password storage with hashing and salting mechanisms.

- **Rapid Development**
- Django’s "batteries-included" philosophy provides built-in tools (e.g., ORM, admin panel, forms) to accelerate development.
- **Versatility**
- Supports diverse applications such as e-commerce platforms, CMS, APIs, and social networks.
- **Built-in Admin Interface**
- Auto-generates a web-based admin panel to manage application data without additional coding.

3.Django vs. Flask comparison: Which to choose and why.

Feature	Django	Flask
Type	Full-stack web framework.	Lightweight, micro-framework.
Flexibility	Less flexible; follows a set structure.	Highly flexible; allows more customization.
Built-in Features	Admin panel, ORM, authentication, forms, etc.	Requires third-party libraries for most features.
Use Cases	Complex, data-driven applications (e.g., e-commerce, CMS).	Simple, smaller applications or APIs.
Setup Time	Longer initial setup due to built-in tools.	Lightweight and faster for small apps.