4. Django Introduction

## Django is a **high-level, open-source Python web development framework** that enables developers to build robust, secure, and scalable web applications rapidly. It follows the **Model-View-Template (MVT)** architectural pattern and provides built-in tools for authentication, database management, form handling, and templating.

1. **Overview of Django: Web development framework.**

Django was developed to help programmers create complex, data-driven websites efficiently.  
It focuses on **reusability, rapid development, and clean design**.  
Key features include:

* **MVT Architecture** – separates business logic from user interface.
* **ORM (Object Relational Mapper)** – interacts with the database using Python instead of SQL.
* **Automatic Admin Interface** – built-in admin panel for database management.
* **Security** – protects against common attacks such as SQL injection, XSS, and CSRF.

1. **Advantages of Django (e.g., scalability, security).**

 **Scalability:** Django supports large-scale applications, making it suitable for high-traffic websites.

 **Security:** Comes with built-in protection mechanisms against various web vulnerabilities.

 **Versatility:** Works well for any kind of application—e-commerce, social media, healthcare, etc.

 **Rapid Development:** Provides reusable components that reduce development time.

 **Community Support:** Has a vast and active global developer community.

1. **Django vs. Flask comparison:**

| **Feature** | **Django** | **Flask** |
| --- | --- | --- |
| **Type** | Full-stack framework | Lightweight micro-framework |
| **Architecture** | MVT (Model-View-Template) | Flexible structure |
| **Database Support** | Built-in ORM | Manual setup |
| **Admin Panel** | Built-in | Requires third-party tools |
| **Ideal For** | Large and complex web apps | Small to medium projects |
| **Learning Curve** | Moderate | Easier for beginners |

1. **Which to choose and why.**

 Choose **Django** when you need a **complete, enterprise-level** solution with built-in tools and faster development.

 Choose **Flask** for **smaller, lightweight** projects requiring flexibility and more control over components.