What is Django?

Django is a high level python web – framework that supports rapid development of secure and maintainable web applications. it comes with many built-in tools and features so developers can focus on writing the actual application instead of reinventing the wheel.

🔹 Originally developed in 2003 for a newsroom application  
🔹 Open-sourced in 2005  
🔹 Currently maintained by the **Django Software Foundation**

Key Features of Django

|  |  |
| --- | --- |
| ✅ **MVT Architecture** | - Clean separation of concerns using Model-View-Template |

|  |  |
| --- | --- |
| ✅ **ORM (Object Relational Mapping)** | -Maps database tables to Python classes |

|  |  |
| --- | --- |
| ✅ **Admin Interface** | * Auto-generated admin dashboard for managing models |

|  |  |
| --- | --- |
| ✅ **Security** | * Prevents common vulnerabilities like XSS, CSRF, SQL injection |

|  |  |
| --- | --- |
| ✅ **Authentication System** | * Comes with built-in user authentication and permissions |

|  |  |
| --- | --- |
| ✅ **Scalability** | * Used by large-scale applications like Instagram, Disqus |

|  |  |
| --- | --- |
| ✅ **URL Routing** | * Elegant URL dispatcher using regex/path-based routes |

|  |  |
| --- | --- |
| ✅ **Middleware Support** | * Hooks to process requests and responses globally |

|  |  |
| --- | --- |
| ✅ **Form Handling** | * Built-in form classes for validation and rendering |

|  |  |
| --- | --- |
| ✅ **Template Engine** | * Separates presentation logic using Django’s templating language |

|  |  |
| --- | --- |
| ✅ **Built-in Development Server** | * Lightweight server for testing during development |

|  |  |
| --- | --- |
| ✅ **Internationalization (i18n)** | * Supports multiple languages and time zones |

|  |  |
| --- | --- |
| ✅ **Extensible and Modular** | Use third-party packages via pip (like Django REST framework, Celery, etc.) |

What are models in django?

In Django, models are defined as Python classes that represent the structure of database tables and the relationships between them. Models define the schema of your application's data and provide a high-level, Pythonic way to interact with the database.

What are migrations?

Migrations are used to manage and apply changes to the database schema based on your model definitions, ensuring that the database reflects the current state of your application's data structure. This approach simplifies database management and makes it easier to maintain and evolve your application's data model over time.