**What is Django?**

**D**jango is a high-level python web framework that encourages (does it) Rapid Development and clean, Pragmatic design.

[Free & open Source, Large & Active community]

What is Web Framework.

It allows you to focus and developing business logic of a websites, without starting from scratch and that was Django does.

Provide many codes from common functionality to help you make your website or web services faster.

Ex: Authentication We don’t need to write down from scratch, Django provides many functions & methods we can use.

Who use?

*YouTube*, *Spotify*, Instagram, Mozilla.

* **Ex: Web Service, Web resources, web API’s**

**What are the Technologies using here…...?**

* **Front-end: (HTML, CSS, JAVASCRIPT, BOOTSTRAP ………)**
* **Back-end: (PYTHON, DATABASE[MYSQL], DJANGO)**

**Django:**

* **Free and open-source web frame.**
* **Written by named after famous guitarist Django created in 2003, but open source in 2005**
* **Backend web development framework**
* **Completely developed by using python**
* **DFS (Django software foundations)**
* **Developed by [👦 Adrian Holovaty and 🧑 Simon Willison]**
* **MVC (Model, View, Control): Most popular frame work to develop web application**

**Almost application follows MVC pattern.**

* **MVT (Model, View, Template) Design Pattern, Django follows MVT**

**Top 5 Features of Django:**

1. **Fast**
2. **Fully Loaded: Means Already predefined**

* **Admin, Security, Authentication, Session Management (these activities already predefined)**

1. **Security: Csrf token (Cross Site Request Forgery)**
2. **Scalability: The Ability to handle increasing number of users, transactions, or data, with out loss in performance or quality.**

**Django Project = Several Applications + Configure Info**

**HOW TO CREATE PROJECT**

* **SY: django-admin startproject (Your project name).**
* **EX: django-admin *startproject*  *project1***

**Sub command projectname**

**After Creating the project, the files in the project are:   
 A). project1:**

1. **\_\_init\_\_.py**
2. **Settings.py**
3. **Urls.py**
4. **Wsgi.py**

**B). Mange.py**

**Structure of Project:**

The project folder is made up of two elements

* Python manage.py help
* Python manage.py runserver

**Purpose of files:**

1. **\_\_init\_\_.py:** An Empty file that tells python that this directory should be considered a python package.
2. **Settings.py:** This is the setting for your project configuration (INSTALLED\_APP, MIDDLEWARE, TEMPLATES, DATABASE LIKE …….)
3. **Urls.py:** This is the link to your project as well as the functions to call
4. **Wsgi.py:** This is needed if you want to perform a deployment of your project via wsgi (Web server gateway interface).
5. **Manage.py:** To give commands related to our Django project. (if you want to run server using this manage.py)

**Create an application:**

* Basically, a project is made up of many files and applications to run the website.
* Inside the project folder we will be having subfolders those folders have some functionality, those are apps.
* All of these applications have an objective that can be homes page and pages etc.

**Django AppFolder:**

1. **\_\_init\_\_.py:** This is for python and it should be treated as package
2. **Admin.py:** This file helps you to make the app modifiable in the admin interface.
3. **Models.py:** Data models loaded here (models always connected database)
4. **Test.py:** Unit test cases.
5. **Views.py:** View functions loaded here.

**Manually created Files:**

Forms.py

┌────────────────────┐

│ Web Browser │

│ (HTML, CSS, JS) │

└────────────────────┘

│

▼ ① Request (HTTP)

┌────────────────────┐

│ Django Views │ ← [Python Logic]

└────────────────────┘

│

▼

┌────────────────────┐

│ Django Models │ ← [ORM handles DB]

└────────────────────┘

│

▼

┌────────────────────┐

│ MySQL DB │

└────────────────────┘

▲

│

┌────────────────────┐

│ Django Templates │ ← [HTML + Dynamic Content]

└────────────────────┘

▲

│

② Response (HTML)

│

▼

┌────────────────────┐

│ Web Browser │

└────────────────────┘

**Step-by-Step Flow:**

1. **User sends a request** (like submitting a form) from the browser.
2. Django **View** receives the request and handles it using Python logic.
3. If needed, it accesses **Model** to interact with the **MySQL Database**.
4. Django **Template** prepares the HTML response dynamically.
5. The browser displays the final response page to the user.

**How it all connects**

1. User fills a form on the HTML page.
2. JavaScript may validate it before submitting.
3. Data is sent to Django server (Python backend).
4. Django processes it and stores/retrieves data from MySQL.
5. Django sends response (new page or data).
6. Browser shows result to user.