Django and API Concepts – Simplified Notes

1. Types of Applications

Applications can be built in different ways depending on how they interact with the user and system:

- Static Application – Only uses HTML, CSS, and Bootstrap. The content does not change dynamically.
- Dynamic Application – Involves Frontend, Backend, and Database. Content changes based on user interaction.
- Smart Application – Similar to dynamic apps but uses Machine Learning models integrated into the backend.

2. Django Overview

Django is a Python framework used to create backend servers. It helps developers handle user requests,
process data, and send proper responses. With Django, we can build APIs that connect frontend and backend easily.

3. Types of Communication (APIs)

When two systems talk to each other, they use APIs. Django is often used to create these APIs.

- REST API – Used when the user or application wants to control the data format and method.
- SOAP – Used when the data format and rules are decided by the service provider.
- WebSockets – Used for real-time communication, like chatting apps or instant updates.
- Webhooks – Used to get live updates or notifications automatically from another system.

4. CRUD Operations using REST API

Using Django, we can perform the main CRUD operations through HTTP methods:

- POST → Used to create a new record or request.
- GET → Used to read or retrieve data from the server.
- PUT/PATCH → Used to update or modify existing data.
- DELETE → Used to remove or cancel something, like an order.

5. HTTP Methods

HTTP methods are commands that tell the server what action to perform on a resource.
They are: POST, GET, PUT, PATCH, DELETE.

6. Django Project Versions and Virtual Environments

Different projects can use different Django versions. Each project should have its own virtual environment
so that the versions and packages do not conflict with each other.

Examples:

- Project 1 – Django v2
- Project 2 – Django v2.7
- Project 3 – Django v3.3
- Project 4 – Django v4

7. Creating a Virtual Environment

Step 1: Create a virtual environment

```
python -m venv venv
```

Step 2: If you get an error about execution policy:

- Open PowerShell as Administrator.
- Run the same command again.

This creates a safe environment to install Django and other required packages.

8. Important Project Files

- .gitignore – Lists the files and folders that should not be uploaded to GitHub.
- site-packages – Contains all installed Python libraries and frameworks.
- requirements.txt – Stores the list of all installed packages for easy setup on other systems.

9. Fun Code Names (Just for Reference)

KALKI → KARNA
RAJASAABH → PSY
BB → KING

SPIRIT → COP

These are just placeholders or internal code names used during project development for easier identification.