🔷 Basic Django Questions

Q1. What is Django?

👉 Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the **MTV (Model-Template-View)** pattern.

Q2. Explain MTV architecture in Django.

- **Model**: Defines the data structure (database tables).
- **Template**: Handles presentation (HTML, CSS, frontend).
- **View**: Contains business logic (handles requests & responses).

Q3. Difference between Django project and Django app?

- **Project**: Whole application setup (can contain multiple apps).
- **App**: A module/component inside the project (e.g., blog, jobs, accounts).

Q4. How do you create a new Django project?

django-admin startproject projectname

Q5. How do you create a new app in Django?

python manage.py startapp appname

Q6. What are migrations in Django?

👉 Migrations are used to apply changes in models (database schema) to the database.

Q7. What are Django middlewares?

👉 Middleware is a framework of hooks into Django's request/response processing. Examples: AuthenticationMiddleware, SessionMiddleware.

Q8. What is the use of manage.py file?

👉 It is a command-line utility that lets you interact with your Django project (runserver, makemigrations, migrate, createsuperuser).

🔷 settings.py Questions

Q1. Role of settings.py?

👉 Stores project configuration (database, apps, middleware, templates, etc.).

Q2. What is INSTALLED_APPS?

👉 A list of all apps activated in the project.

Q3. What is MIDDLEWARE?

👉 A list of middleware classes that process request/response.

Q4. What is DATABASES configuration?

```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'mydb',
        'USER': 'myuser',
        'PASSWORD': 'mypassword',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Q1. What are Django models?

👉 Python classes that represent database tables.

```python
from django.db import models

class Student(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
```

Q3. Use of primary_key, unique, and null?

- primary_key=True: Makes field a unique identifier.
- unique=True: No duplicate values allowed.
- null=True: Allows storing NULL in database.

Q4. Difference between ForeignKey, OneToOneField, and ManyToManyField?

- ForeignKey: Many-to-one relationship.
- OneToOneField: One-to-one relationship.
- ManyToManyField: Many-to-many relationship.

Q5. Purpose of Meta class?

👉 To define metadata like table name, ordering, verbose_name.

🔷 admin.py Questions

Q1. Purpose of admin.py?

👉 To register models for Django Admin interface.

Q2. Register a model?

```
from django.contrib import admin
from .models import Student

admin.site.register(Student)
```

🔷 views.py Questions

Q1. What is a view?

👉 A function/class that handles requests and returns responses.

Q2. FBV vs CBV?

- **FBV**: Simple functions.
- **CBV**: Class-based, reusable, more structured.

Q3. HttpResponse vs JsonResponse?

- HttpResponse: Returns plain HTML/text.
- JsonResponse: Returns JSON data.

🔷 Superuser Questions

Q1. What is a superuser?

👉 A user with all permissions in Django Admin.

Q2. How to create a superuser?

python manage.py createsuperuser

Q3. Can multiple superusers exist?

👉 Yes.

🔷 Django Project Questions

Q1. Structure of a project?

```
project/
 ├– manage.py
 ├– project/
 |   ├– settings.py
 |   ├– urls.py
 |   ├– wsgi.py
 |   ├– asgi.py
 ├– app/
 |   ├– models.py
 |   ├– views.py
 |   ├– urls.py
 |   ├– admin.py
```

Q2. urls.py project vs app?

- Project urls.py → Root URLs.
- App urls.py → App-specific routes.

Q4. What is WSGI and ASGI?

- WSGI → Web Server Gateway Interface (for sync).
- ASGI → Asynchronous Server Gateway Interface (for async).

Q5. Run Django project on another port?

python manage.py runserver 8080

🔷 **Django App Questions**

Q1. Create new app?

python manage.py startapp appname

Q2. Can a project have multiple apps?

👉 **Yes, recommended for modularity.**

Q3. Purpose of apps.py?

👉 **Stores app config (AppConfig class).**