Django Interview Guide

1. Why Choose Django?

What is Django?

Django is a high-level Python web framework that simplifies web development by providing built-in features for security, database management, and scalability. It follows the Model-View-Template (MVT)

architecture, making it structured and efficient.

Why Choose Django Over Other Tech Stacks?

Django provides built-in authentication, an admin panel, and a powerful ORM, making development faster and more secure. Unlike Flask, which is minimalistic, Django is full-featured, reducing the need

for third-party libraries.

Key Django Advantages:

- Security Protection against SQL injection, CSRF, and XSS.
- Scalability Handles high traffic efficiently (used by Instagram, Pinterest).
- Built-in Admin Panel Reduces development time.
- ORM Interact with the database using Python instead of raw SQL.
- Rapid Development Follows the DRY (Don't Repeat Yourself) principle.

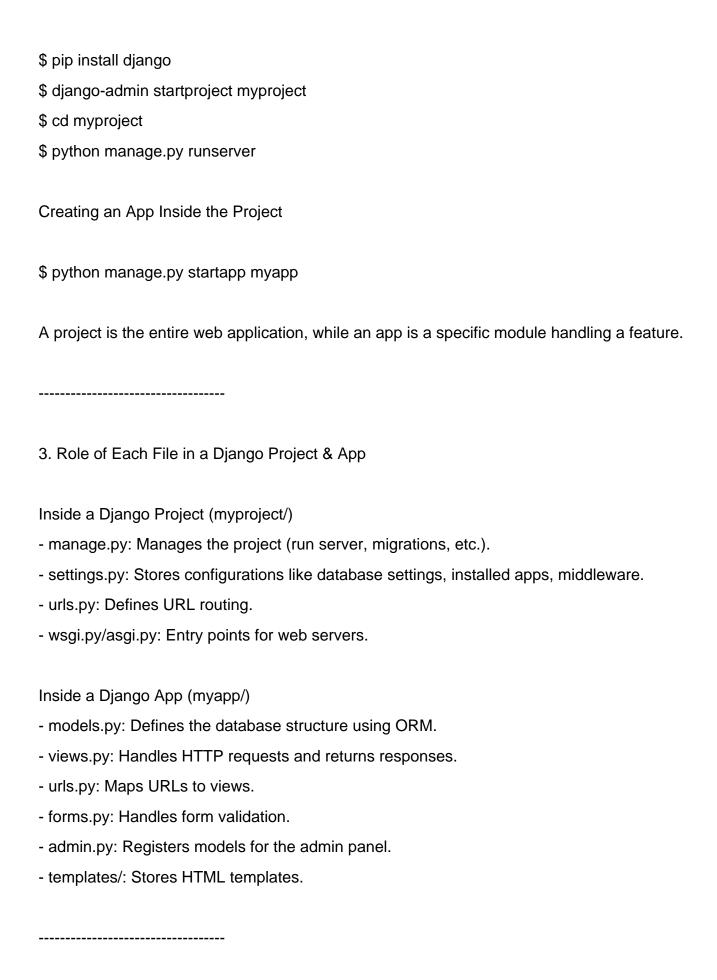
Interview Question: Why did you choose Django over Flask or Node.js?

Answer: Django provides built-in authentication, an admin panel, and a powerful ORM, making development faster and more secure. Flask is minimalistic, while Django is full-featured.

2. Django Project Flow & App Creation

Setting Up a Django Project

Django Interview Guide



Django Interview Guide

4. Most-Used Django Commands & Their Roles
- python manage.py runserver # Start development server
- python manage.py makemigrations # Detect model changes
- python manage.py migrate # Apply migrations to the database
- python manage.py createsuperuser # Create an admin user
- python manage.py collectstatic # Gather static files for production
5. Popular Django Interview Questions & Answers
How does Djangos ORM work?
Djangos ORM allows interaction with the database using Python instead of raw SQL.
What is Django Middleware?
Middleware is a series of components that process requests before reaching views and responses before
reaching the user.
What is the difference between function-based views and class-based views?
- Function-Based Views (FBVs) Simple, procedural.
- Class-Based Views (CBVs) More reusable, uses OOP concepts.
How do you handle static files in Django?
Django serves static files using the STATICFILES_DIRS setting. In production, collectstatic gathers
all static files into one directory for efficient serving.