DJANGO INTERVIEW QUESTIONS & ANSWER

By - AMIT MAHTO

Write down a command to create Virtual Envirnoment?

python -m venv venvname

Write down a command to activate Virtual Envirnoment?

venvname\Scripts\activate

Write down a command to install Django ?

pip install django

Write down a command to check version of django?

django-admin --version

Write down available command in django?

django-admin

Write down a command to create new project?

django-admin startproject projectname

Write down a command to run a Django project?

python manage.py runserver

by default this command starts the server on internal IP at port 8000

Write down a command to change the server port?

python manage.py runserver8080

Write down a command to change server IP?

python manage.py runserver0.0.0.0:8000

Write down a command to create a app?

py manage.py startapp appname

Write down a command to create migratiion?

py manage.py makemigrations

Write down a command to apply the changes in database?

py manage.py migrate

Write down a command to see raw SQL QUERY executing behind applied migration?

python manage.py sqlmigrate appname migrationname

Write down a command to see all migrations, execute ?

python manage.py showmigrations

Write down a command to see specific migration by specific app?

python manage.py showmigrations appname

Write down a command to create superuser?

py manage.py createsuperuser

Write down a command to change a password?

py manage.py changepassword username

Write down a command to check all the version of installed modules?

pip freeze

Write down a command to for admin interface?

pip install django-admin-interface

Write down a command to view a database schema of an existing database?

python manage.py inspectdb

Write down a Django command to view all the items in Model?

Users.Objects.all()

where Users will be your model name

Write down a Django command to view single id in Model?

Users.Objects.get(id=7)

Write down a Django command to filter items in Model?

Users.Objects.filter(name="AMIT")

How to delete an object using Queryset in Django?

Users.Objects.get(id=7).delete()

How to update an object using Queryset in Django?

user_to_be_modify = User.Objects.get(id=7)

user_to_be_modify.city ="PUNE"

user_to_be_modify.save()

How to insert/add an object using Queryset in Django?

new_user = User(name="AMIT MAHTO", city="PUNE")

new_user.save()

What is Django?

- Django is a web development framework that was developed in a fast-paced newsroom.
- It is a free and open-source framework that was named after Django Reinhardt who was a jazz guitarist from the 1930s.
- Django is maintained by a non-profit organization called the Django Software Foundation.
- The main goal of Django is to enable Web Development quickly and with ease.

Name some companies that make use of Django?

- Instagram
- DISCUS
- Mozilla Firefox
- YouTube
- Pinterest

What are the features of Django?

- SEO Optimized
- Extremely fast
- Fully loaded framework that comes along with authentications, content administrations, RSS feeds, etc
- Very secure thereby helping developers avoid common security mistakes such as cross-site request forgery (csrf), clickjacking, cross-site scripting, etc
- It is exceptionally scalable which in turn helps meet the heaviest traffic demands
- Immensely versatile which allows you to develop any kind of websites

What are the advantages of using Django?

- Django's stack is loosely coupled with tight cohesion
- The Django apps make use of very less code
- Allows quick development of websites
- Follows the DRY or the Don't Repeat Yourself Principle which means, one concept or a piece of data should live in just one place
- Django offer Rapid Development
- Django is highly Scalable

Can you explain me the flow of MVT in Django?

User Request →Django server →urls → views→now it can go in either in Model or in Template.

- here user requests for a resource to Django, Django works as a controller and check to the available resource in URL.
- If URL maps a view is called that interact with model and template. it renders template
- Django responds back to the user and sends a template response.

Explain Django architecture.

- Django follows the MVT or Model View Template architecture whcih is based on the MVC or Model View Controller architecture.
- The main difference between these two is that Django itself takes care of the controller part.

What is Django Model?

- In Django, the model does the linking to the database and each model gets mapped to a single table in the database.
- These fields and methods are declared under the file models.py

What is Django Views?

- This is the part where actually we would be mentioning our logic.
- This coding is done through the python file views.py
- Views in Django either run an HttpResponse or raise an exception such as Http404.
- Django views serve the purpose of encapsulation. They encapsulate the logic liable for processing a user's request and for returning the response back to the user.

What is Django Templates?

- Using templates, you can generate HTML dynamic.
- The HTML consists of both static as well as dynamic parts of the content.
- You can have any number of templates depending on the requirement of your project.

Explain how a request is processed in Django?

- here user requests for a resource to Django, Django works as a controller and check to the available resource in URL.
- When django server is started the manage.py file searches for setting.py file which contains information of all the application installed in the project, middleware used, database connections and path of the main urls config.
- Manage.py >> Setting.py >> urls.py >> views.py >> models.py >> templates
- Django first determines which root URL config or URL configuration module to be used
- Then, that particular python module urls is loaded and then django looks for the variable urlpatterns.
- Then it check each URL patterns in urls.py file, and it stops at the first match of requested url
- Once that is done, the Django then imports and calls the given views.\In case none of URL match the required URL django invokes an error – handling views
- If URL maps a view is called that interact with modeland template, it renders a template
- Django now responds back to the user and sends a template as a response.

What are the various files that are created when you create a Django Project and app? Explain briefly.

File name	Description
manage.py	A command-line utility that allows you to interact with your Django project
__init__.py	An empty file that tells Python that the current directory should be considered as a Python package
settings.py	Consists of the settings for the current project
urls.py	Contains the URL's for the current project
wsgi.py	This is an entry-point for the web servers to serve the project you have created.
test.py	App unit test automation classes are included in this

What is the difference between a Project and an App?

- A Project is the entire django application and an APP is a module inside the project that deals with one specific use case.
- For example : payment system(app) in the eCommerce app(Project)
- An APP is basically a web application that is created to perform a specific task
- A Project on the other hand is a collection of these apps
- Therefore a single project and contain 'n' numbers of apps and a single app can be in multiple projects.

Which is the default database in the settings file in Django?

Database Name : SQLite

Why is Django called a loosely coupled framework?

- It is called loosely coupled framework because of its MVT architecture, which is a variant of MVC
- MVT helps in separating the server code from client-related code
- Django models and views are present on the client machine and only templates return to the client which are essentially HTML, CSS code and contains the required data from the models.
- These components are totally independent of each other and therefore front-end developer and back-end developers can work simultaneously on the project as these two parts changing will have little to no effect on each other when changed .

Explain Migration in django?

- Migration in django is to make changes o our models.py file like deleting a model, adding a field etc into your database schema.
- A migration in django is a python file that contains changes we make to our models so that they can be converted into database schema in or DBMS
- So, instead of manually making changes to our database schema by writing queries in our DBMS shell we can just make a changes in our models
- Then, we can use django to generate migration from those model changes and run those migration to make changes to our database schema.

What is Django ORM?

- ORM stands for Object Relational Mapper.
- This ORM enables us to interact with databases in a more pythonic way like we can avoid writing raw queries.
- It is possible to retrive , save, delete and perform other operations over the database without ever writing any SQL query
- It helps us with working with data in a more object oriented way.

EXAMPLE

SQL query where Student table to retrieve s student name

Select * from Student where name="AMIT";

The Equivalent django ORM Query will be:

std=student.objects.filter(name='AMIT')

Explain how you can set up the Database in Django?

- To set a database in Django , you can find its configuration in setting.py file that represent Django settings
- By default , Django uses SQLite database.
- It is easy for Django users because it doesn't require any other type of installation

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

if you want to change your database then make this changes shown below:

'ENGINE': 'django.db.backends.mysql',

What do you mean by the CSRF Token?

- CSRF stands for Cross Site Request Forgery.
- The csrf_token is used for protection against Cross-Site Request Forgeries.
- This kind of attack takes place when a malicious website consists of a link, some JavaScript or a form whose aim is to perform some action on your website by using the login credentials of a genuine user.

What is QuerySet in Django?

- It is a collection of SQL Queries
- A QuerySet in django is basically a collection of objects from our databases
- Queryset are used by the Django ORM.
- It is comparable to a database select operation

EXAMPLE

users.objects.all()

users.objects.filter(name="AMIT")

users.objects.get(id=7)

Difference between select_related and prefetch_related in Django?

- select_related: it returns a QuerySet that will "follow" foreign-key relationship
- prefetch_related: we use this when we are going to get a setoff things. that means forward ManyToMany and backward ManyToMany , Foreign keys

How do you connect your Django project to the database?

- Django comes with a default database which is SQLite.
- To connect your project to this database, use the following commands:

1. python manage.py migrate (migrate command looks at the INSTALLED_APPS settings and creates database tables accordingly)
2. python manage.py makemigrations (tells Django you have created/ changed your models)
3. python manage.py sqlmigrate <name of the app followed by the generated id> (sqlmigrate takes the migration names and returns their SQL)

What are static files?

- Static files in Django are those files that serve the purpose of additional files such as the CSS, images or JavaScript files.
- These files are managed by django.contrib.staticfiles.
- These files are created within the project app directory by creating a subdirectory named as static.

What are 'signals'?

- Django consists of a signal dispatcher that helps allow decoupled applications to get notified when actions occur elsewhere in the framework.
- Django provides a set of built-in signals that basically allow senders to notify a set of receivers when some action is executed.

Briefly explain Django Field Class.

- 'Field' is basically an abstract class that actually represents a column in the database table.
- The Field class, is in turn, a subclass of RegisterLookupMixin.
- In Django, these fields are used to create database tables (db_type()) which are used to map Python types to the database using get_prep_value() and vice versa using from_db_value() method.
- Therefore, fields are fundamental pieces in different Django APIs such as models and querysets.

What is mixin?

- Mixin is a type of multiple inheritance wherein you can combine behaviors and attributes of more than one parent class.
- Mixins provide an excellent way to reuse code from multiple classes.
- For example, generic class-based views consist of a mixin called TemplateResponseMixin whose purpose is to define render_to_response() method.
- When this is combined with a class present in the View, the result will be a TemplateView class.

Explain the use of Middlewares in Django.

- Middleware is a framework that is light and low-level plugin system for altering Django's input and output globally.
- It is basically a framework of hooks into the request/ response processing of Django.
- Each component in middleware has some particular task.
- For example, the AuthenticationMiddleware is used to associate users with requests using sessions.
- Django provides many other middlewares such as cache middleware to enable site-wide cache, common middleware that performs many tasks such as forbidding access to user agents, URL rewriting, etc, GZip middleware which is used to compress the content for browsers, etc.

What is the significance of manage.py file in Django?

- The manage.py file is automatically generated whenever you create a project.
- This is basically a command-line utility that helps you to interact with your Django project in various ways.
- It does the same things as django-admin but along with that, it also sets the DJANGO_SETTINGS_MODULE environment variable in order to point to your project's settings.
- Usually, it is better to make use of manage.py rather than the django-admin in case you are working on a single project.

Explain the use of 'migrate' command in Django?

- In Django, migrations are used to propagate changes made to the models.
- The migrate command is basically used to apply or unapply migrations changes made to the models.
- This command basically synchronizes the current set of models and migrations with the database state.
- You can use this command with or without parameters.
- In case you do not specify any parameter, all apps will have all their migrations running.

Explain how a request is processed in Django?

- In case some user requests a page from some Django powered site, the system follows an algorithm that determines which Python code needs to be executed.
- Here are the steps that sum up the algorithm:

1. Django first determines which root URLconf or URL configuration module is to be used
2. Then, that particular Python module is loaded and then Django looks for the variable urlpatterns
3. These URL patterns are then run by Django, and it stops at the first match of the requested URL
4. Once that is done, the Django then imports and calls the given view
5. In case none of the URLs match the requested URL, Django invokes an error-handling view

How to use file-based sessions?

- In order to make use of file-based sessions, you will need to set the SESSION_ENGINE setting to "django.contrib.sessions.backends.file".

Explain the Django URLs in brief?

- Django allows you to design your own URLs however you like.
- The aim is to maintain a clean URL scheme without any framework limitations.
- In order to create URLs for your app, you will need to create a Python module informally called the URLconf or URL configuration which is pure Python code and is also a mapping between the URL path expressions to the Python methods.
- The length of this mapping can be as long or short as required and can also reference other mappings.
- When processing a request, the requested URL is matched with the URLs present in the urls.py file and the corresponding view is retrieved.

What is Jinja templating?

- Jinja Templating is a very popular templating engine for Python, the latest version is Jinja2.

Some of its features are:

1. Sandbox Execution - This is a sandbox (or a protected) framework for automating the testing process
2. HTML Escaping - It provides automatic HTML Escaping as <, >, & characters have special values in templates and if using a regular text, these symbols can lead to XSS Attacks which Jinja deals with automatically.
3. Template Inheritance
4. Generates HTML templates much faster than default engine
5. Easier to debug as compared to the default engine.

What are different model inheritance styles in the Django?

1. **Abstract Base Class Inheritance:** Used when you only need the parent class to hold information that you don't want to write for each child model.
2. **Multi-Table Model Inheritance**: Used when you are subclassing an existing model and need each model to have its own table in the database.
3. **Proxy Model Inheritance:** Used when you want to retain the model's field while altering the python level functioning of the model.

Explain user authentication in Django?

- Django comes with a built-in user authentication system, which handles objects like users, groups, user-permissions, and few cookie-based user sessions.
- Django User authentication not only authenticates the user but also authorizes him.
- The system consists and operates on these objects:
  1. Users
  2. Permissions
  3. Groups
  4. Password Hashing System
  5. Forms Validation
  6. A pluggable backend system

How to configure static files?

- Ensure that django.contrib.staticfiles is added to your INSTALLED_APPS
- In your settings file. define STATIC_URL for ex.

```
STATIC_URL = '/static/'
```

- In your Django templates, use the static template tag to create the URL for the given relative path using the configured STATICFILES_STORAGE.
- {% load static %}

<img src="{% static 'my_sample/abcxy.jpg' %}" alt="ABC image">

- Store your static files in a folder called static in your app.
- For example my_sample/static/my_sample/abcxy.jpg

What's the significance of the settings.py file?

- As the name suggests this file stores the configurations or settings of our Django project, like database configuration, backend engines, middlewares, installed applications, main URL configurations, static file addresses, templating engines, main URL configurations, security keys, allowed hosts, and much more.

How to view all items in the Model?

ModelName.objects.all()

How to filter items in the Model?

ModelName.objects.filter(field_name="term")

Explain Q objects in Django ORM?

- Q objects are used to write complex queries, as in filter() functions just `AND` the conditions while if you want to `OR` the conditions you can use Q objects

What are Django exceptions?

In addition to the standard Python exceptions, Django raises of its own exceptions.List of the exceptions by Django

What is the Django Admin interface?

- Django comes equipped with a fully customizable, built-in admin interface.
- This portal lets developers see and make changes to all the data residing in the database that contains registered apps and models.

- The model must be registered in the admin.py file to use a database table with the admin interface.

What is Function Based Views in Django?

- Function-based views are good for beginners.
- It is very easy to understand in comparison to class-based views.
- Initially when you want to focus on core fundamentals, using the function-based views gives the advantage to understand it.

EXAMPLE:

```
def example_create_view(request, pk):
template_name = 'form.html'
form_class = FormExample

form = form_class

if request.method == 'POST':
        form = form_class(request.POST)
        if form.is_valid():
        form.save()
        return HttpResponseRedirect(reverse('list-view'))

return render(request, template_name, {'form': form})
```

What is Class Based Views in Django?

- Class-based views are the alternatives of function-based views.
- It is implemented in the projects as Python objects instead of functions.
- Class-based views don't replace function-based views, but they do have certain advantages over function-based views.
- Class-based views take care of basic functionalities such as deleting an item or add an item.
- Using the class-based view is not easy if you're a beginner.

How to give a path to FBV and CBV?

FBV:

path('home',views.home,name='home),

CBV:

path('home',views.Home.as_view(), name='home'),

Difference between FBV and CBV?

FBV	CBV
It is not easy to re-use code	When you are using class-based views you can re-use code
Easy to read, understand and implement.	Complex to implement and harder to read
Explicit code flow	Implicit code flow.
Straightforward usage of decorators	Use of view decorators require extra import, or method override
Condition branching will be used handle HTTP request.	We can use the different class instance method (instead of conditional branching statement inside function-based-views) to generate the HTTP requests.

What is the job of middleware?

ANSWER:

--> In Django, middleware is a lightweight plugin that processes during request and response execution. Middleware is used to perform a function in the application.

--> The functions can be a security, session, csrf protection, authentication etc.

--> Django provides various built-in middleware and also allows us to write our own middleware.

--> In settings.py file of Django project that contains various middleware, that is used to provides functionalities to the application.

--> For example, Security Middleware is used to maintain the security of the application

setting.py file

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

--> There are two types of Middleware in Django:

1. Built-in Middleware - Built-in Middleware are provided by default in Django when you create your project. You can check the default Middleware in settings.py file of your project.

2. Custom Middleware - You can write your own middleware which can be used throughout your project.

What is Session ?

- A session is a mechanism to store information on the server side during the interaction with the web application.
- In Django, by default session stores in the database and also allows file-based and cache based sessions.

How to set and get a Sessions?

urls.py file

```
 path('setsession', views.setsession,name="setsession"),
 path('getsession', views.getsession,name="getsession"),
```

views.py file

```
def setsession(request):
    request.session['name']='AMIT'
    return HttpResponse('seesion are saved')

def getsession(request):
   sname=request.session['name']
   return HttpResponse(sname)
```

How to Implement Session ?

Put **django.contrib.sessions.middleware.SessionMiddleware** in MIDDLEWARE and **django.contrib.sessions** in INSTALLED_APPS of settings.py file.

What is Cookies?

- A cookie is a small piece of information which is stored in the client browser.
- It is used to store user's data in a file permanently (or for the specified time).
- Cookie has its expiry date and time and removes automatically when gets expire. Django provides built-in methods to set and fetch cookie.
- The set_cookie() method is used to set a cookie and get() method is used to get the cookie.
- The request.COOKIES['key'] array can also be used to get cookie values.

How to set and get a Cookies ?

urls.py file

```
 path('setcookie', views.setcookie,name="setcookie"),
 path('getcookie', views.getcookie,name="getcookie"),
```

views.py file

```
def setcookie(request):
    response=HttpResponse("cookies set")
    response.set_cookie('user','AMIT')
    return response

def getcookie(request):
    name=request.COOKIES['user']
    return HttpResponse(name)
```

How to import user class in django?

from django.contrib.auth.models import User

How to create new user in django?

```
from django.contrib.auth.models import User
user = User.objects.create_user(username=amit,
                                email=amit@gmail.com',
                                password='amit23456')
```

How to register model in admin file?

```
admin.site.register(Book)
```

How to get acesss in admin panel ?

By creating superuser

How to check authentication?

```
from django.contrib.auth import login as auth_login,logout as auth_logout,authenticate
def login(request):
 user=authenticate(username=username,password=password)
```

How to import authenticate?

```
from django.contrib.auth import login as auth_login,logout as auth_logout,authenticate
```

How to check user is login or not?

using auth_login function

Difference HttpResponse and render.

render is basically a simple wrapper around a HttpResponse which renders a template.

Difference between render and redirect in Django?

Render is a rendering variable to the template, and redirect is a jump function in HTTP, which typically generates a 302 status code

What is HttpResponse in Django?

- HttpResponse (source code) provides an inbound HTTP request to a Django web application with a text response.
- This class is most frequently used as a return object from a Django view.

How to change title, header and index title in django admin?

```
admin.site.site_header="Django Classes"
admin.site.site_title="Django"
admin.site.index_title="Welcome to my project"
```

What is use of cleaned_data in django?

- **form.cleaned_data** returns a dictionary of validated form input fields and their values, where string primary keys are returned as objects.
- **form.data** returns a dictionary of un-validated form input fields and their values in string format (i.e. not objects).

Difference between Emp.object.filter(), Emp.object.get() and Emp.objects.all() in Django Queryset?

Emp.object.filter() and **Emp.object.get()** - To filter out some element from database you either use the get() method or the filter() method as follows:

Users.objects.filter(name="AMIT")

Users.objects.get(name="AMIT")

get() we use when we want to get a single unique object. It will give error if there's no object matching query

filter() we use when we want to get all objects that match your lookup parameters. It will return an empty queryset

Emp.objects.all() - In order to view all the items from your database you can make use of the all() function as mention below:

Users.objects.all()

where Users is some class that you have created in your models

What is the difference between Flask and Django?

FLASK	DJANGO
Created in 2010	Created in 2005
Python web framework built for rapid development.	Python web framework built for easy and simple projects.
Flask is WSGI framework.	Django is a Full Stack Web Framework.
Templates, Admin and ORM it require installation	Templates, Admin and ORM are built-in
Bootstrapping-tool are not available	Bootstrapping-tool are built-in
Support Visual Debug	Does not support Visual Debug

What is the difference between Authentication and Authorization in Django?

Authentication means who Rare you ? whereas Authorization means what permission you have?

Authentication is a process of verifying who someone is whereas Authorization means it is a process of verifying what specific application, files and data a users have access.

Authentication is done before the Authorization whereas Autherization is done after the Authentication

Authentication usually needs login details whereas Authorization needs user's privilege or security level

Disadvantage of Django?

- Django is a Monolithic. You must Know full system to work with it.
- Components are develop together
- Django modules are bulky
- Django is completely based on Django ORM

What is the Django shortcut method to more easily render an HTML response?

Django shortcut methods is render_to_response

What does {% include %} do in Django?

- It loads a template and renders it with the current context.
- This is a way of including other templates within a templates.
- Syntax: {%include "template_name.html" %}

Give the exception classes present in Django.

- Django uses its own exceptions as well as those present in Python.
- Django core exceptions are present in django.core.exceptions class some of which are mentioned

ObjectDoesNotExist - This is the base class for DoesNotExist exceptions

MultipleObjectsReturned - This is raised by a query if multiple objects are returned and only one object was expected

EmptyResultSet - This exception may be raised if a query won't return any result

What is serialization in Django?

It is a process of coverting django models into other format like XML, JSON,etc

What is WSGI?

Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.

What is Werkzeug ?

It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.