1. Which architectural pattern does Django follow?

Django follows Model-View-Template (MVT) architectural pattern.

See the following graph that shows the MVT based control flow.



Here, a user **requests** for a resource to the 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 a template.

Django responds back to the user and sends a template as a **response**.

1. What are the features available in Django web framework?

Features available in Django web framework are:

* Admin Interface (CRUD)
* Templating
* Form handling
* Internationalization
* A Session, user management, role-based permissions
* Object-relational mapping (ORM)
* Testing Framework
* Fantastic Documentation

### What are the inheritance styles in Django?

There are three possible inheritance styles in Django:

**1) Abstract base classes:** This style is used when you only want parent's class to hold information that you don't want to type out for each child model.

**2) Multi-table Inheritance:** This style is used if you are sub-classing an existing model and need each model to have its database table.

**3) Proxy models:** This style is used, if you only want to modify the Python level behavior of the model, without changing the model's fields.

### What is some typical usage of middlewares in Django?

Some usage of middlewares in Django is:

* Session management,
* Use authentication
* Cross-site request forgery protection
* Content Gzipping

1. What is the usage of Django-admin.py and manage.py?

**Django-admin.py:** It is a Django's command line utility for administrative tasks.

**Manage.py:** It is an automatically created file in each Django project. It is a thin wrapper around the Django-admin.py. It has the following usage:

* It puts your project's package on sys.path.
* It sets the DJANGO\_SETTING\_MODULE environment variable to points to your project's setting.py file.

1. How to handle URLs in Django?

To handle URL, **django.urls** module is used by the Django framework.

Let's open the file **urls.py** of the project and see the what it looks like:

**// urls.py**

1. from django.contrib **import** admin
2. from django.urls **import** path
4. urlpatterns = [
5. path('admin/', admin.site.urls),
6. ]

See, Django already has mentioned a URL here for the admin. The path function takes the first argument as a route of string or regex type.

The view argument is a view function which is used to return a response (template) to the user.

The **django.urls** module contains various functions, **path(route,view,kwargs,name)** is one of those which is used to map the URL and call the specified view.

### What is Django Session?

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

### What are the signals in Django?

Signals are pieces of code which contain information about what is happening. A dispatcher is used to sending the signals and listen for those signals.

### How to handle URLs in Django?

To handle URL, **django.urls** module is used by the Django framework.

Let's open the file **urls.py** of the project and see the what it looks like:

**// urls.py**

1. from django.contrib **import** admin
2. from django.urls **import** path
4. urlpatterns = [
5. path('admin/', admin.site.urls),
6. ]

See, Django already has mentioned a URL here for the admin. The path function takes the first argument as a route of string or regex type.

The view argument is a view function which is used to return a response (template) to the user.

The **django.urls** module contains various functions, **path(route,view,kwargs,name)** is one of those which is used to map the URL and call the specified view.

### What is Django Exception?

An exception is an abnormal event that leads to program failure. To deal with this situation, Django uses its exception classes and supports all core Python exceptions as well. Django core exceptions classes are defined in django.core.exceptions module.