**Explain Django?**

Django is a free and open source web application framework, written in python. It is a server-side web framework that provides high secure websites.

**Explain architectural pattern does Django follow?**

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



Here user request for a resource to the Django, Django works as a controller and check whether it is available in URL.

If URL maps, a view is called based on that view login interaction with model will be done and it renders to the template.

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

**Explain Django architecture?**

Django follows MVT (Model View Template) pattern. It is slightly different from MVC.

Model: It is the data access layer. It contains everything about the data, i.e. how to access it, how to validate it.

View: It is a business logic layer. This layer contains logic that is used to interact with the database and renders to the appropriate template.

Template: It is a presentation layer. It is responsible for displaying the response to the end user.

**Is Django a high-level web framework or low-level framework?**

Django is a high-level python’s web framework.

**Is Django stable?**

Yes, Django is stable. Many companies are using Django like Instagram, Pinterest, YouTube, Bitbucket, Mozilla Firefox, NASA, etc.

**What are the features available in Django web framework?**

Features available in Django web framework are:

1. Admin interface (CRUD)
2. Templating
3. Form handling
4. Object-relational mapping (ORM)
5. Fantastic documentation
6. Secure
7. Fully loaded
8. Ridiculously fast

**Explain the advantages of Django?**

Advantages of Django are:

1. Easy to learn
2. Clear and readable
3. Versatile (companies, organizations and governments are using Django)
4. Fast to write
5. Secure
6. Scalable

**What are the disadvantages of Django?**

Following is the list of disadvantages of Django:

1. Django’s modules are bulky
2. It is completely based on Django ORM
3. Components are deployed together

**What are the inheritance styles in Django?**

There are three inheritance styles in Django:

1.Abstract base classes: If we have common fields in two model classes, then it is not recommended to write two times. Instead we create a base model class and extend that to the child model classes.

For base model class, we have to define abstract=True inside class Meta.

2.Multi-table inheritance: Here we will extend the existing model class to have the existing fields in the child model class.

3.Proxy models: It is used when we want to change the behaviour of the model, without changing the model’s fields, by making proxy=True inside class Meta.

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

Some usage of middlewares in Django is:

1. Session management
2. User authentication
3. Cross-site request forgery protection
4. Content Gzipping

**What does Django field class types (field types) do?**

The Django field class types specify:

1. The database column type
2. The default HTML widget, while rendering a form field.
3. The minimal validation requirements used in Django admin
4. Automatic generated forms

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

admin.py: It is Django’s command line utility for administrative tasks.

manage.py: It is automatically created file in each project. It has the following usage:

1. It puts our project’s package on sys.path.
2. It sets the DJANGO\_SETTING\_MODULE environment variable to points to our project’s settings.py file.

**What are the signals in Django?**

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

**What are the two important parameters in signals?**

Two important parameters in signals are:

Receiver: It specifies the callback function which connected to the signal.

sender: It specifies a particular sender from where a signal is received.