

Django-Assignment

1. What is Django?

Ans. Django is a free and open-source, Python -based web framework that follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established in the US as a 501 (c) (3) non-profit.

2. What is crud in Django?

Ans. CRUD can be best explained as an approach to building a Django web application. In general, CRUD means performing Create, Retrieve, Update, and Delete operations on a table in a database.

- **CREATE procedures:** Performs the INSERT statement to create a new record.
- **READ procedures:** Reads the table records based on the primary key noted within the input parameter.
- **UPDATE procedures:** Executes an UPDATE statement on the table based on the specified primary key for a record within the WHERE clause of the statement.
- **DELETE procedures:** Deletes a specified row in the WHERE clause.

3. What is CSRF ?

Ans. CSRF stands for Cross-Site Request Forgery. It is an attack that forces an end user to execute unwanted actions on a web application in which they are currently authenticated.

4. What are the features of Django ?

Ans.

- **Excellent Documentation**

This is one of the main reasons to start learning Django. If we compare Django with other open source technologies, it offers the best documentation in the market. Better documentation of any technology is like a very well-established library for any developer. There, he can search for any function desired with ease with the time involving in the searching purpose only. The documentation of any technology is also one of the categories to grade a technology, as it lets other developers other than its own creators to efficiently utilize the technology.

- **Python Web-framework**

Python is also one of the main reasons people started learning Django. It is that one tool which can solve all your problems and in any kind of operation out there, we can use it. It's very simple

and easy to use. All these features are inside Python. In fact, Python is currently the most popular language in the market.

- SEO Optimized

This is a special feature of Django due to which it has edge over others. SEO is Search Engine Optimization as from the name it means that adding your website to the search engine such that it appears in the top results. As we know that the search engines do use some algorithms which sometimes doesn't cooperate much with the web-developer. Since we are creating our website in the human understandable form and they have to add it in the URL form on the server so that its best recognized by the search engine.

- High Scalability

A lot of MNCs on a worldwide scale uses Django and it gets implement there without any defects or errors. It is the best example of Django being scalable.

Scalability means that at what scope or level, our technology gets to implement. For bigger websites like Instagram, there are lots of active users (millions of them) which generate data in huge amounts (terabytes of data/day). This kind of level requires our system or application to be very precise and error-free. It is, of course, difficult even for programmers and web developers that have years of experience.

- Versatile in Nature

Django is very versatile in its own Django way. The logical project structure and MVT architecture of Django sometimes seem very limiting. But, that's just opposite because by giving us the files it is providing us with a solid foundation which can then be used to make whichever application we want to create.

- Offers High Security

Django is super secure. To prove the feature, you can always take examples of lots of websites which are worldwide and posses huge traffic.

Django is secure because it covers the loopholes by default which were once left open for the backend developer to complete. Although while using Django you may not feel it but those expert backend developers can tell the quality and security of the work done by Django.

5. Architecture of Django

Ans. The architecture of Django is MVT architecture.

MVT- Model View Templates architecture.

The Template handles the UI and architecture part of an application. The view does the logical part of the application and interacts with the Model to get the data and in turn modifies the template accordingly. Django works as a controller and gets a URL that is linked to the view part of the application and thus transports the user responses to the application. This complete interaction is dealt with this Django MVT architecture. When we create a project, there would be some default files that would be created.

Models: The section which handles the database of the project.

Views : The section which contains the python functions that receive the web request and web response.

Templates: It is a collection of html page/user interface part of the project.

5. What is models.py and forms.py?

Ans. Models.py - A model is a class that represents a table or collection in our DB, and where every attribute of the class is a field of the table or collection.

forms.py is where the Django documentation recommends you to place all your forms code, to keep your code easily maintainable. Also, since it's a convention mentioned in the documentation, it helps when you are collaborating with others because that is where others will expect to look for your code dealing with forms.

7. What are templates in Django?

Ans. In Django templates are nothing but the collection of HTML pages, A Django template is a text document or a python string marked-up using recognized and interpreted by the template engine.

8. What are views in Django?

Ans: views are the python files which receive the web request and return the web responses.

9. What are static files?

Ans: Static files are the external files such as image, pdf, video, CSS, font etc...

10. What is DRF?

Ans: Django Rest Framework, which is a powerful and flexible toolkit for building web APIs

11. What is Django URL?

Ans: In Django URL stands for Uniform Resource Locator, which connects views and the templates.

12. What is migrate command?

Ans. Migrations are Django's way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema. They're designed to be mostly automatic, but you'll need to know when to make migrations, when to run them, and the common problems you might run into.

13. What is the use of makemigration?

Ans. makemigrations, which is responsible for creating new migrations based on the changes you have made to your models.

14. What are the difference between project and app?

Ans. A project in Django is a python package that represents the whole web application. Project in Django basically contains the configuration and setting related to the entire website. A single project can also have multiple apps in it that can be used to implement some functionality.

An app in Django is a sub-module of a project, and it is used to implement some functionality. Now, you can refer to an app as a standalone python module that is used to provide some functionality to your project. We can create multiple apps within a single Django project. And these apps can be independent of one another. Theoretically, we can use an app from one Django project to another without making any changes to it.

15. What is request in Django?

Ans. Django uses request and response objects to pass state through the system. When a page is requested, Django creates an HTTP Request object that contains metadata about the request. Then Django loads the appropriate view, passing the HTTP Request as the first argument to the view function. Each view is responsible for returning an HTTP Response object.

16. What is middle ware in Django?

Ans. Middleware is a framework of hooks into Django's request/response processing. It's a light, low-level "plugin" system for globally altering Django's input or output.

Each middleware component is responsible for doing some specific function. For example, Django includes a middleware component, AuthenticationMiddleware, that associates users with requests using sessions

17. What is Django context?

Ans. Django-context is an BSD licensed library written in Python . It could be helpful for you if you want to set some custom context in your Django app. The one of the examples is add an extra info into your logging records.

18. Find the latest version of Django?

Ans: 4.1 is the latest version of Django

19. What is difference Charfield and textfield in Django?

Ans. CharField has max_length of 255 characters while TextField can hold more than 255 characters. Use TextField when you have a large string as input. It is good to know that when the max_length parameter is passed into a TextField it passes the length validation to the TextArea widget.

20. Common exceptions occurs in Django

Ans:

- ObjectDoesNotExist: The base class for DoesNotExist exceptions.
- FieldDoesNotExist: It raises when the requested field does not exist.
- MultipleObjectsReturned: This exception is raised by a query if only one object is expected, but multiple objects are returned.
- ViewDoesNotExist : It is raised by django.urls when a requested view does not exist.
- FieldError: It is raised when there is a problem with a model field.
- ValidationError: It is raised when data validation fails form or model field validation.

21. What are cookies?

Ans: 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.

22. How to check the version of the Django installed in your device?

Ans. `import django print django.get_version`

23. What is `django.shortcuts.render`?

Ans: Combines a given template with a given context dictionary and returns an `HttpResponse` object with that rendered text. Django does not provide a shortcut function which returns a `TemplateResponse` because the constructor of `TemplateResponse` offers the same level of convenience as `render()` .

24. What is the command in Django to create project and app?

Ans:

- `django-admin startproject project_name`
- `python manage.py startapp app_name`

25. What is query set in Django?

Ans: A Query Set is a collection of data from a database. A Query Set is built up as a list of objects. Query Sets makes it easier to get the data you actually need, by allowing you to filter and order the data.

26. Explain the type of files that generate automatically when create a project

Ans:

- models.py
- admin.py
- WSGI
- settings.py
- urls.py

27. What is difference between filter and & set?

Ans: The Difference between Django's filter() and get() methods are: get throws an error if there's no object matching the query. Basically use get() when you want to get a single unique object, and filter() when you want to get all objects that match your lookup parameters.

28. What is Django session?

Ans: It is mechanism of storing information in server side during interact with application