2. Django - это веб- фреймворк для разработки на Python. Он используется для создания веб-приложений, сайтов и сервисов, которые могут обрабатывать большие объемы данных и работать с базами данных

4. Паттерн – это шаблон, который может быть использован для решения конкретных задач. DRY является общим принципом, который призывает избегать дублирования кода

6. Шаблоны в Django - это файлы, которые содержат в себе HTML,данные которые будут отображаться на экране.

<!DOCTYPE html>  
<html>  
 <head>  
 <title>{{ title }}</title>  
 </head>  
 <body>  
 <h1>{{ heading }}</h1>  
 <ul>  
 {% for item in items %} #цикл  
 <li>{{ item }}</li>  
 {% endfor %}  
 </ul>  
 </body>  
</html>

8.Можно использовать Django rest framework и через апишки взаимодействовать с фронтендом.

10. Маршруты в Django - это пути, по которым обрабатываются запросы приложения.Пример:

from django.urls import path  
import views  
  
urlpatterns = [  
 path(r'^posts/(?P<post\_id>\d+)/$', views.post\_view, name='post\_view'),  
]

Регулярные выражения – это шаблоны, которые проверяют(?) адреса ссылок.

12. Параметр on\_delete в поле ForeignKey определяет, что должно произойти с объектом модели, когда объект удален.

14. CRUD – (Create, Read, Update, Delete). основные операции, которые могут выполняться с данными в бд и приложениях. Create – создание новых записей или объектов; Read –получение данных из бд; Update – обновление существующих объектов в бд; Delete – удаление объектов из бд

16. Django сигналы - это механизм, который позволяет отправлять сигналы из разных частей приложения при наступлении определенных событий. С помощью сигналов можно управлять работой приложения на основе определенных событий, которые происходят внутри Django.

18. Авторизация и аутентификация – предназначены для защиты данных пользователя.

1)В Django есть готовая системя аутентификации и авторизации - username и password.

2)в Django есть встроенная библиотека Django Rest Framework.

3) В Django также можно использовать разные уровни доступа к представлениям, используя декораторы **@login\_required**

20.Django-admin - это утилита командной строки, которая поставляется с Django и используется для управления проектом: создание приложений, запуск сервера разработки, управление миграциями базы данных и т.д.

22. Django поддерживает несколько баз данных, среди которых: SQLite,PostgreSQL,MySQL,Oracle,Microsoft SQL Server.

24. Функция render в Django - это способ генерации HTTP-ответа на основе шаблона и контекста данных.

Контекст в Django - это словарь Python, который содержит переменные и их значения для использования в шаблоне. Он передает контекст в виде словаря с переменными и их значениями.

26. Промежуточное ПО (middleware) в Django - это слой обработки запросов и ответов между веб-сервером и представлениями (views).Стандартные:

**jango.middleware.security.SecurityMiddleware**: обеспечивает базовую безопасность приложения

**django.middleware.csrf.CsrfViewMiddleware**: обеспечивает защиту от атаки CSRF ,проверяя токены в каждом POST-запросе.

**django.contrib.auth.middleware.AuthenticationMiddleware**: обеспечивает аутентификацию пользователей, сохраняет информацию о пользователе в объекте **request.user** и предоставляет доступ к методам авторизации и аутентификации.

**django.middleware.locale.LocaleMiddleware**: обеспечивает выбор локали для пользователя на основе его предпочтений.

**django.middleware.common.CommonMiddleware**: добавляет дополнительные методы для работы с запросами и ответами

28. В Python существует два основных стиля наследования - классическое и наследование нового типа. Django использует наследование нового типа, которое позволяет создавать более сложные иерархии наследования

30. Миксин - это способ повторного использования кода в ООП, который представляет собой класс, содержащий некоторые методы или свойства, которые можно добавить в другие классы через множественное наследование.

from django.views.generic import ListView  
from django.contrib.auth.mixins import LoginRequiredMixin  
  
class View(LoginRequiredMixin, ListView):  
 model = Model  
 template\_name = 'template.html'

34.django.shortcuts.render - это функция, которая используется в Django для быстрого создания HTTP-ответа при помощи переданных данных

36.Объект Q позволяет создавать сложны запросы используя логические операторы и поиска объектов в бд.

from django.db.models import Q  
from datetime import date  
from myapp.models import Model  
  
example = Model.objects.filter(Q(created\_\_lt=date(2022, 1, 1)))