Django Question for Re-Assessment

1. **What is a Django model in the context of web development?**
- A Django model is a class that represents database tables and structures data for Django applications.

2. **How can you perform database migrations in Django?**
- Use python manage.py makemigrations to create migrations and python manage.py migrate to apply them.

3. **Which Django command is used to generate the necessary database schema based on the model?**
- python manage.py migrate

4. **What is the purpose of the** auto_now_add **option in a Django DateTimeField?**
- It automatically sets the field to the current timestamp when the record is first created.

5. **What is the purpose of the** {% block %} **tag in Django templates?**
- {% block %} defines a section in a template that child templates can override or fill **Which template tag is used to display the URL for a named URL pattern in Django templates?**lates?**
- {% url 'name_of_the_url_pattern' %}

7. **Which template tag is used to perform if-else logic in Django templates?**
- {% if %} ... {% else %} ... {% endif %}

8. **How do you start the development server in Django?**
- python manage.py runserver

9. **Which template tag is used for escaping HTML content in Django templates?**
- {{ variable|escape }}

10. **How can you handle form submissions in Django views?**
- Use request.POST in views to access submitted form data and handle it accordingly.

1. What is the purpose and use of the save () function in Django?
2. How are cards utilized in Bootstrap, and what are their key features?
3. How can large static files be effectively managed in a Django project?
4. What are the advantages of using URL patterns in Django?
5. How are template views implemented in Django's Class-Based Views (CBVs)?
6. How can conditional logic be used to display "Yes" or "No" in a Django template?

1. **What is a Django model?**
   a) A Python module for defining forms
   b) A class that represents database tables
   c) A controller for handling HTTP requests
   d) A template for defining HTML pages
   Answer: b

2. **Which of the following options ensures a unique value for a field in a model?**
   a) primary_key=True
   b) blank=True
   c) unique=True
   d) null=True
   Answer: c

3. **What does the ForeignKey field represent in Django?**
   a) A many-to-many relationship
   b) A one-to-one relationship
   c) A many-to-one relationship
   d) A self-referential relationship
   Answer: c

4. **What does the choices parameter in a Django model field do?**
   a) Limits the query results
   b) Defines pre-defined options for the field
   c) Ensures field values are unique
   d) Validates the field data
   Answer: b

5. **How do you query all objects from a model?**
   a) Model.fetch_all()
   b) Model.objects.all()
   c) Model.objects.retrieve()
   d) Model.all()
   Answer: b

6. **What is the default return type of a Django view?**
   a) JSON Response
   b) HTML Response
   c) HttpResponse object
   d) Plain text
   Answer: c

7. **What does @login_required in Django do?**
   a) Blocks non-authenticated users from accessing a view
   b) Logs in a user
   c) Redirects unauthenticated users to the home page
   d) Enables session management
   Answer: a

8. **Which function is used to render templates in Django?**
   a) render_template
   b) render
   c) HttpResponse
   d) render_view
   Answer: b

9. **How can you display a variable in a Django template?**
   a) {variable}
   b) {{ variable }}
   c) {% variable %}
   d) <variable>
   Answer: b

10. **Which tag is used to loop through a list in a template?**
    a) {% for %}
    b) {% loop %}
    c) {% iterate %}
    d) {% while %}
    Answer: a

11. **How do you include another template inside a Django template?**
    a) {% import 'template_name.html' %}
    b) {% use 'template_name.html' %}
    c) {% include 'template_name.html' %}
    d) {% render 'template_name.html' %}
    Answer: c

12. **What is the purpose of the {% csrf_token %} tag?**
    a) Adds a security token to forms
    b) Renders a CSRF error message
    c) Verifies user authentication
    d) Encrypts the form data
    Answer: a

13. **How can you pass a context dictionary to a template?**
a) Through the render() function
b) Using HttpResponse()
c) Using JsonResponse()
d) Adding it directly to the template
Answer: a

14. **Which command creates a new migration file?**
a) python manage.py migrate
b) python manage.py makemigrations
c) python manage.py create_migration
d) python manage.py generate_migration
Answer: b

15. **Which database is the default for Django projects?**
a) PostgreSQL
b) MySQL
c) SQLite
d) MongoDB
Answer: c

16. **How can you rollback the last migration?**
a) python manage.py rollback
b) python manage.py migrate <app_name>
c) python manage.py migrate <app_name> zero
d) python manage.py migrate <app_name> <previous_migration>
Answer: d

17. **Which file contains the list of middleware classes in a Django project?**
a) urls.py
b) views.py
c) settings.py
d) middleware.py
Answer: c

18. **How can you disable middleware in a Django project?**
a) Remove it from INSTALLED_APPS
b) Comment it out in settings.py
c) Remove it from MIDDLEWARE in settings.py
d) Add disable_middleware=True in settings.py
Answer: c

19. **Which method is used to authenticate a user in Django?**
a) authenticate_user()
b) login_user()
c) authenticate()
d) verify_user()
Answer: c

20. **What is the default model for user authentication in Django?**
a) AuthUser
b) User
c) AbstractUser
d) BaseUser
Answer: b

21. **How do you log out a user in Django?**
a) request.logout()
b) logout_user()
c) logout(request)
d) deactivate_user()
Answer: c

22. **What is the purpose of Django's is_authenticated property?**
a) To log in a user
b) To check if a user is active
c) To check if a user is logged in
d) To validate user permissions
Answer: c

23. **Where should you define the path for static files in a Django project?**
a) urls.py
b) settings.py
c) views.py
d) models.py
Answer: b

24. **Which command collects all static files into a single directory?**
a) python manage.py gather_static
b) python manage.py collectstatic
c) python manage.py getstatic
d) python manage.py staticfiles
Answer: b

25. **What is the default static files directory in Django?**
a) /static
b) /static_root
c) /assets
d) /media
Answer: a

26. **Which class is used to create a form in Django?**
a) BaseForm
b) Form
c) FormView
d) ModelForm
Answer: b

27. **What does {{ form.as_p }} do in a template?**
a) Renders a form as plain text
b) Renders a form with <p> tags
c) Renders a form with table tags
d) Renders a form in JSON format
Answer: b

28. **Which method validates a form in Django?**
a) is_valid()
b) validate_form()
c) check()
d) validate()
Answer: a

41. **Which command is used to run the development server?**
a) python manage.py runserver
b) django startserver
c) manage.py server
d) runserver.py
Answer: a

42. **Which WSGI server is commonly used for Django in production?**
a) Apache
b) Nginx
c) Gunicorn
d) Flask
Answer: c

43. **What file contains the WSGI configuration for a Django project?**
a) settings.py
b) urls.py
c) wsgi.py
d) config.py
Answer: c

44. **What is the default port for the Django development server?**
a) 8000
b) 8080
c) 3000
d) 5000
Answer: a

45. **What is the purpose of the Meta class in Django models?**
a) Define database table settings
b) Define the model's methods
c) Handle migrations
d) Define custom signals
Answer: a

46. **Which model field would you use to store large text?**
a) TextField
b) CharField
c) SlugField
d) BigTextField
Answer: a

47. **Which option is used to make a field optional in a Django model?**
a) optional=True
b) null=True, blank=True
c) required=False
d) nullable=True
Answer: b

48. What does the on_delete argument in a ForeignKey do?
a) Defines what happens to related records when a record is deleted
b) Automatically deletes all related objects
c) Prevents deletion of the parent record
d) Triggers a pre-delete signal
Answer: a

49. **How do you register a model in the Django admin?**
a) admin.register(model_name)
b) admin.site.add_model(model_name)
c) admin.site.register(model_name)
d) admin.model_register(model_name)
Answer: c

50. **What is the purpose of the list_display attribute in an admin class?**
a) To display detailed data for each model
b) To control the columns displayed in the list view
c) To add custom forms
d) To paginate the list view
Answer: b

51. □ **Which method is used to customize the form for a model in the admin?**
a) form
b) model_form
c) get_form
d) customize_form
Answer: c

52. What is the purpose of the search_fields attribute in admin.py?
a) Adds a search box in the admin interface
b) Filters data by field
c) Limits fields displayed in the form
d) Exports data in search results
Answer: a

53. **What does render() do in a Django view?**
a) Handles raw HTTP requests
b) Renders a template with a context dictionary
c) Executes SQL queries
d) Redirects to another URL
Answer: b

54. **Which type of view is TemplateView?**
a) Function-based
b) Class-based
c) Middleware-based
d) Database-based
Answer: b

55. □ **How do you redirect users to another URL in a view?**
a) return redirect('url_name')
b) HttpResponse('url')
c) HttpResponseRedirect('url')
d) return redirect_to('url_name')
Answer: a

56. □ **What is the default HTTP method for Django views?**
a) POST
b) GET
c) PUT
d) PATCH
Answer: b

57. □ **Which decorator restricts a view to logged-in users only?**
a) @csrf_exempt
b) @login_required
c) @require_http_methods
d) @restricted
Answer: b

58. Django urls.py

59. **Which function is used to define URL patterns in Django?**
a) url_patterns
b) path()
c) define_urls()
d) set_urls()
Answer: b

60. **What is the purpose of include() in urls.py?**
a) Allows nested URL configurations
b) Imports additional templates
c) Registers views dynamically
d) Adds middleware
Answer: a

61. **How do you pass a dynamic value in a URL pattern?**
a) <int:parameter_name>
b) {parameter_name}
c) *parameter_name*
d) ?parameter_name
Answer: a

62. **What does the name argument in path() do?**
a) Specifies the view function to call
b) Defines a human-readable name for the URL pattern
c) Allows reverse URL resolution
d) Binds middleware to the URL
Answer: c

63. **Which of the following resolves a URL by name in templates?**
a) {% get_url 'url_name' %}
b) {% reverse_url 'url_name' %}
c) {% url 'url_name' %}
d) {% path 'url_name' %}
Answer: c

64. **Which file contains the project's settings and configurations?**
a) settings.py
b) config.py
c) manage.py
d) env.py
Answer: a

65. **What is the purpose of the .env file in Django?**
a) Store environment-specific variables securely
b) Define URL configurations
c) Handle database migrations
d) Register middleware
Answer: a

66. **Which command initializes a new Django project?**
a) django-admin startapp
b) django-admin startproject
c) django-admin createproject
d) python manage.py newproject
Answer: b

67. **How can you switch between development and production settings in Django?**
a) Use different settings.py files
b) Modify the database URL
c) Set DEBUG=True for production
d) Use Django's internal environment switcher
Answer: a

68. **What is the purpose of serializers in DRF?**
a) Validate database models
b) Convert complex data to JSON and vice versa
c) Manage migrations
d) Handle URL routing
Answer: b

69. **Which class is used to create API views in DRF?**
a) APIView
b) APITemplate
c) APIRenderer
d) RESTView
Answer: a

70. **What does the @api_view decorator do?**
a) Converts a class-based view to a REST API view
b) Defines allowed HTTP methods for a function-based view
c) Handles authentication in APIs
d) Renders HTML templates in API responses
Answer: b

71. **What is the purpose of HyperlinkedModelSerializer in DRF?**
a) Adds hyperlinks to model relations
b) Formats data in plain text
c) Generates models dynamically
d) Handles raw SQL queries
Answer: a

72. **Which HTTP status code indicates a successful DELETE request?**
a) 204
b) 200
c) 201
d) 304
Answer: a

73. **How can you enable CORS for Django REST APIs?**
a) Add django.middleware.cors.CorsMiddleware
b) Use the cors-headers package
c) Set ALLOW_CORS=True in settings
d) Add middleware for CORS in settings.py
Answer: b

General Django Basics

1. What is Django?
2. What are the main features of Django?
3. What is the default database supported by Django?
4. How do you create a Django project?
5. What is the use of manage.py?
6. How do you start a new app in a Django project?
7. What is the difference between a project and an app in Django?
8. What is the role of settings.py in a Django project?

Models and Database

9. What is a model in Django?
10. How do you define a model in Django?
11. What command is used to apply changes to the database after modifying models?
12. What are makemigrations and migrate commands in Django?
13. What is the purpose of Meta class in a Django model?
14. How can you retrieve all records from a model?
15. What is the difference between get() and filter() in Django QuerySet?
16. What is the purpose of the models.py file in a Django app?
17. How do you define a model in Django?
18. What is the significance of the id field in Django models?
19. What is the default primary key field in Django models?
20. How do you define a field as a primary key in Django?
21. Name some commonly used field types in Django models.
22. What is the purpose of CharField and TextField in Django models?
23. How does ForeignKey work in Django models?
24. What is a ManyToManyField in Django, and when should you use it?
25. What is the difference between DateField and DateTimeField?
26. What is the purpose of the Meta class in a Django model?
27. How do you define a table name for a model using Meta?
28. How can you make a model field unique?
29. What is the use of db_index=True in a model field?
30. How can you order querysets by default for a model?
31. What is the difference between OneToOneField, ForeignKey, and ManyToManyField?
32. How do you set up a self-referential ForeignKey in a model?
33. How do you define a related name for a ForeignKey field?

34. What is on_delete in a ForeignKey, and what options are available?
35. How do you retrieve related objects for a ManyToManyField?
36. What is the purpose of makemigrations and migrate commands?
37. How do you add a new field to an existing model?
38. What is a migration file in Django?
39. How do you rename a field in a Django model?
40. What happens if you delete a model without applying migrations?
41. How do you define a custom manager for a model?
42. What is the purpose of the default argument in a model field?
43. How do you set a field to allow null values in the database?
44. How do you set a field to be optional in forms but not null in the database?
45. How do you set a default value for a field in Django models?
46. How do you retrieve all objects of a model in Django?
47. How do you filter objects in a queryset?
48. How do you retrieve the first or last object from a queryset?
49. What is the difference between all() and values() in a queryset?
50. How do you perform aggregation in a Django queryset?
51. How do you delete a record in Django models?
52. What is the purpose of the `choices` attribute in a model field?
53. What is the difference between `blank=True` and `null=True` in Django models?
54. How do you handle image uploads in a Django model?
55. What is the use of `verbose_name` and `verbose_name_plural` in Django models

Views and URLs

16. What are views in Django?
17. How do you map a view to a URL in Django?
18. What is the difference between Function-Based Views (FBV) and Class-Based Views (CBV) in Django?
19. What is the purpose of the urlpatterns list in Django?
20. How do you pass parameters in a URL in Django?

Templates

21. What is the role of templates in Django?
22. How do you render a template in a Django view?
23. What is the syntax for using a variable in a Django template?
24. How can you include one template into another?
25. What is the use of {% block %} and {% endblock %} in templates?

Forms

26. What is the purpose of Django forms?
27. How do you create a form in Django?
28. What is the difference between ModelForm and forms.Form?
29. How do you validate form data in Django?
30. What method is used to handle POST data in a Django view?

Admin

31. How do you register a model with the Django admin site?
32. What is the command to create a superuser in Django?
33. How can you customize the Django admin interface for a model?
34. What is the purpose of the admin.py file in a Django app?
35. How do you filter data in the Django admin interface?

Authentication and Security

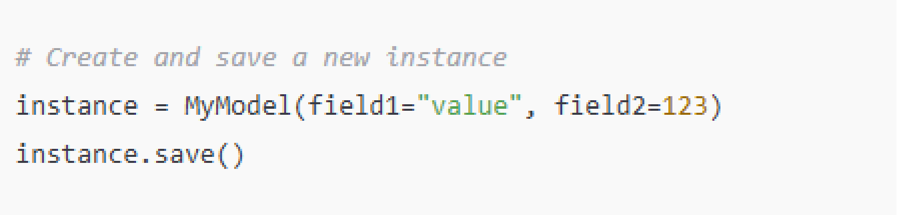
36. What is the default User model in Django?
37. How can you create a custom user model in Django?
38. How does Django handle user authentication?
39. What is CSRF and how is it handled in Django?
40. What is the purpose of LOGIN_URL and LOGIN_REDIRECT_URL?

General DRF Basics

1. What is Django REST Framework (DRF)?
2. Why is DRF used in Django projects?
3. What are serializers in DRF?
4. What is the purpose of a serializer in DRF?
5. What is the difference between Serializer and ModelSerializer in DRF?

1. What is the purpose and use of the save() function in Django?

The save() function in Django is used to persist model instances to the database. When you call save() on a model instance:



```
# Create and save a new instance
instance = MyModel(field1="value", field2=123)
instance.save()
```

2. How are cards utilized in Bootstrap, and what are their key features?

Cards in Bootstrap are flexible and extensible content containers designed to display information like text, images, or links in an organized manner. They are often used for creating UI elements such as user profiles, dashboards, or product listings.

Key Features:

- **Customizable Content**
- **Headers and Footers**
- **Responsive Layouts**
- **Shadow and Border Styling**
- **Grid and Deck Support**

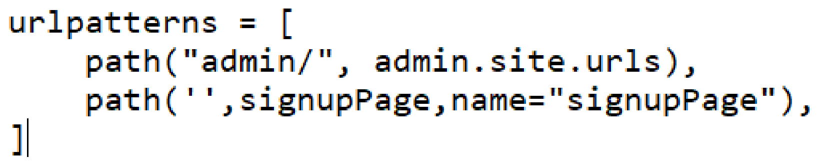
3. How can large static files be effectively managed in a Django project?

Large static files can be managed in Django by using specialized storage backends and optimizing the way files are served.

I. Use a Content Delivery Network (CDN)
II. Configure STATICFILES_STORAGE
III. Compression and Minification:

4. **What are the advantages of using URL patterns in Django?**

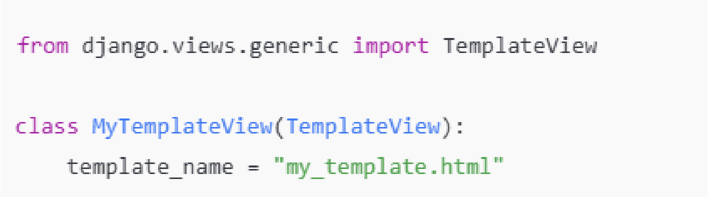
- Separation of Concerns:
- Dynamic URL Routing:
- Reverse URL Resolution:
- Scalability
- Error Handling:



```
urlpatterns = [
    path("admin/", admin.site.urls),
    path('',signupPage,name="signupPage"),
]
```

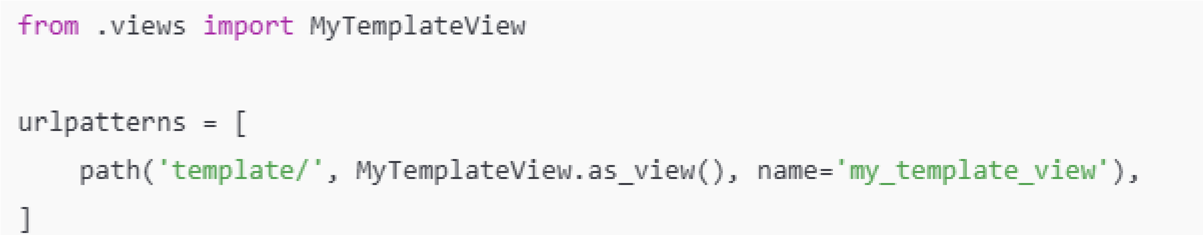
5. **How are template views implemented in Django's Class-Based Views (CBVs)?**

Template views in Django CBVs are implemented using the TemplateView class, which renders a template without requiring additional logic.



```
from django.views.generic import TemplateView

class MyTemplateView(TemplateView):
    template_name = "my_template.html"
```

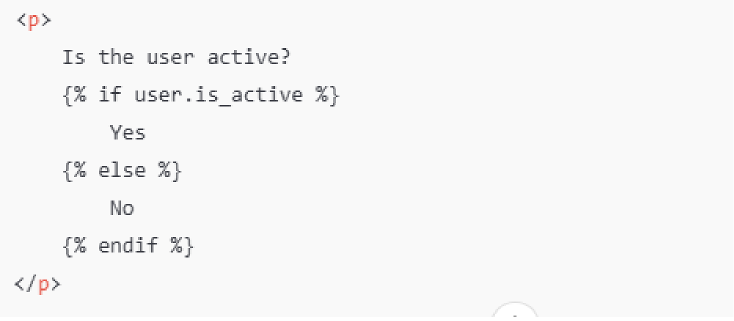


```
from .views import MyTemplateView

urlpatterns = [
    path('template/', MyTemplateView.as_view(), name='my_template_view'),
]
```

6. **How can conditional logic be used to display "Yes" or "No" in a Django template?**

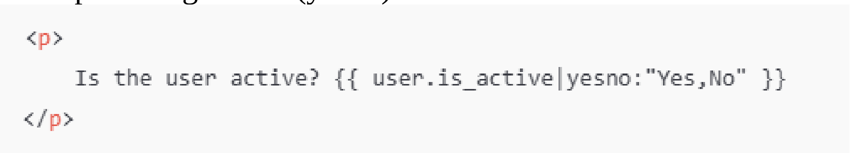
Example Using {% if %}:



```
<p>
    Is the user active?
    {% if user.is_active %}
        Yes
    {% else %}
        No
    {% endif %}
</p>
```

Or

Example Using Filters (yesno):



```
<p>
    Is the user active? {{ user.is_active|yesno:"Yes,No" }}
</p>
```