

Here are **short, technical, clear answers** for all questions in **Sheet 3**:

1. How do you set up a database connection in Django?

In settings.py, configure the DATABASES dictionary with engine, name, user, password, host, and port.

Example:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

2. How can you serve static files during development in Django?

Enable STATIC_URL and STATICFILES_DIRS in settings.py and use:

```
urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

3. What is Django REST Framework (DRF)?

DRF is a powerful framework for building RESTful APIs using Django with serializers, viewsets, authentication, and browsable API support.

4. What is get_object_or_404 in Django?

A shortcut that returns an object if found; otherwise raises Http404.

Example:

```
post = get_object_or_404(Post, id=1)
```

5. What is the purpose of settings.py in a Django project?

It stores global configurations: database, installed apps, static files, middleware, templates, authentication, etc.

6. How does Django's authentication system work?

It handles login, logout, passwords, user sessions, permissions, and user models via middleware and auth views.

7. What is Django Admin, how is it used?

A built-in admin panel to manage database models. Enable by registering models in admin.py.

8. What is the use of the urls.py file in Django?

It maps URL patterns to views using routing configurations.

9. What is Django ORM? What operations can be performed?

ORM translates Python code into SQL.

Operations: Create, Retrieve, Update, Delete.

Example:

```
User.objects.create(username="mahfuz")
```

10. How does Django's ORM manage relationships?

Using ForeignKey, ManyToManyField, and OneToOneField to link models and allow query joins automatically.

11. How do you retrieve all records of a model?

```
records = ModelName.objects.all()
```

12. How do you create a new record in Django ORM?

```
obj = ModelName.objects.create(field1="data")
```

13. How can you delete multiple records in Django ORM?

Use filter() then delete():

```
ModelName.objects.filter(status="inactive").delete()
```

14. What is a view in Django?

A function or class that processes an HTTP request and returns a response.

15. What is a model in Django?

A Python class that defines a database table structure.

16. How do you handle forms in Django?

Use forms.Form or forms.ModelForm, process form data in views, validate, and save.

17. What is the use of manage.py in Django?

A command-line utility to runserver, migrate, create apps, run tests, etc.

18. What is the purpose of Meta class in a Django model?

Defines model options (ordering, table name, verbose name, unique_together).

Example:

```
class Meta:
    ordering = ['-created_at']
```

19. How do you create a custom model manager in Django?

```
class ActiveManager(models.Manager):
    def active(self):
        return self.filter(status=True)
```

```
class MyModel(models.Model):
    objects = ActiveManager()
```

20. How do you create a function-based view in Django?

```
def my_view(request):
    return HttpResponse("Hello")
```

21. What is render() in Django views, and what does it do?

Renders an HTML template with a context and returns an HttpResponse.

```
return render(request, 'home.html', {'data': info})
```

22. What is the purpose of redirect() in Django views?

Redirects the user to another URL or view.

```
return redirect('home')
```

23. How do you handle form submission in Django views?

Check request method, validate form, save data.

```
if request.method == "POST":
    form = MyForm(request.POST)
    if form.is_valid():
        form.save()
```

24. What is get_context_data() in a CBV?

Used to add extra context data to a class-based view template.

25. How do you create a view that only logged-in users can access?

Use decorator:

```
@login_required
```

```
def dashboard(request):
```

```
    ...
```

26. How do you use a for loop in Django templates?

```
{% for item in items %}
```

```
    {{ item.name }}
```

```
{% endfor %}
```

27. What is template inheritance in Django?

Allows child templates to extend a base template using:

```
{% extends 'base.html' %}
```

28. How do you delete a record in Django ORM?

```
obj = ModelName.objects.get(id=1)
```

```
obj.delete()
```
