53R_54R_TEST_DJANGO

Duration: 2 hrs

Max Marks: 100

Section A – MCQs (10 × 2 = 20 marks)

Choose the correct option and tick/circle the answer.

1. In Django, which file is mainly responsible for mapping URLs to views?
   a) settings.py
   b) urls.py
   c) models.py
   d) wsgi.py

2. Which class do we generally inherit from to create a model in Django?
   a) django.db.Model
   b) django.db.models.Model
   c) django.core.Model
   d) django.models.BaseModel

3. JsonResponse in Django is mainly used to return:
   a) HTML pages
   b) Plain text
   c) JSON data (like dict, list)
   d) XML data

4. HTTP GET method is usually used to:
   a) Create data
   b) Update data
   c) Delete data
   d) Read / fetch data

5. Which decorator is used to skip CSRF validation on a view?
   a) @csrf_required
   b) @csrf_off
   c) @csrf_exempt
   d) @no_csrf

6. Middleware in Django runs:
   a) Only before the view
   b) Only after the view

c) Around the view (before and after)
d) Only when server starts

7. Which of the following is true about hashing passwords?
a) Hashed passwords can be decrypted easily
b) Hashing is a one-way process
c) Hashing and encryption are the same
d) Hashing is only for images

8. In Django, make_password() is used to:
a) Compare passwords
b) Encrypt and decrypt passwords
c) Hash a password for storing in DB
d) Send password via email

9. In Django, check_password(raw, hashed) returns:
a) The hashed password
b) True or False
c) The decrypted password
d) The username

10. In a Django view, request.body when we send JSON from Postman usually contains:
a) Already parsed Python dict
b) A bytes/string JSON that we need to parse
c) HTML data
d) Only query parameters

Section B – Short Answer Questions

Each question carries 5 marks. (8 × 5 = 40 marks)

1. What is the difference between HttpResponse and JsonResponse? Give a small example use case for each.

2. In simple English, what is a model in Django? Write 3–5 bullet points explaining its purpose.

3. Explain what @csrf_exempt does and when we typically use it in APIs.

4. Define encryption and hashing. Write one key difference between them. Which one is used for passwords and why?

5. Explain the role of __init__ and __call__ methods in a Django middleware class like this:

```
class RequestLoggerMiddleware:
    def __init__(self, get_response):
```

```
        self.get_response = get_response

    def __call__(self, request):
        # some logic
        response = self.get_response(request)
        return response
```

6. What is the purpose of self.get_response in middleware? Explain in simple terms.

7. You have a single URL /student/ and a view that handles POST to create a student and GET to list all students.
- Which variable in Django tells you the HTTP method?
- How do you check it in the view?

8. Explain the signup + login concept with hashed passwords in 4–6 lines (just the logic, not code).

Section C – Long Answer / Coding Questions

Each question carries 8 marks. (4 × 10 = 40 marks)

1. Q1. Correct the Login Logic (Password Hashing)

You have the following signup and login code (pseudo):

```
# Signup
def signUp(request):
    if request.method == "POST":
        data = json.loads(request.body)
        user = Users.objects.create(
            username=data.get('username'),
            email=data.get('email'),
            password=data.get('password')   # directly storing
        )
        return JsonResponse({"status": "success"}, status=200)

# Login
def login(request):
    if request.method == "POST":
        data = json.loads(request.body)
        username = data.get('username')
        password = data.get('password')

        user = Users.objects.get(username=username, password=password)
        return JsonResponse({"status": "login successful"}, status=200)
```

a) Explain what is wrong in this code from a security point of view.
b) Rewrite the signup and login parts using make_password() and check_password() correctly (only main logic needed).

2. Q2. Write a View Handling CRUD with Same URL

You have a StudentNew model with fields: name, age, email.

Write a Django view function student_api(request) that uses the same URL and handles:
- POST → Create a new student (data coming as JSON in request.body)
- GET → Return all students as JSON (list of dicts)
- PUT → Update a student's email based on id (given in JSON)
- DELETE → Delete a student based on id (given in JSON)

You can ignore try/except if it's too long, but show the basic logic for each method.

3. Q3. Middleware Understanding (Theory + Small Code)

a) In 4–5 lines, explain what middleware is in Django and why we use it.

b) Write a simple middleware that prints:
- request method
- request path
before calling the view, and prints:
- response status code
after calling the view.

(Just the class, no need to register it in settings.py.)

4. Q4. Short Notes on Password Security (Explain Like a Teacher)

Write short notes (in simple English) on the following (2–3 lines each):
a) Why storing "harish123" directly in DB is dangerous.
b) What make_password() does internally (high-level explanation).
c) How check_password(raw, hashed) decides True or False.
d) What is a salt in password hashing and why it is used.
e) Why we cannot decrypt hashed passwords.