INTERVIEW HOURS

1. Introduction

- **Tell me a little about yourself and your background.**
    - Look for a concise summary of their academic background and any practical experience with Django, Python, or web development.
- **Why are you interested in a backend development role, particularly with Django and DRF?**
    - The candidate should express interest in backend development and possibly mention the scalability and performance of Django/DRF or the ease of building APIs with DRF.
- **Can you describe any projects you've worked on related to Django or Python?**
    - Look for real-world projects, open-source contributions, or academic projects using Django, DRF, or Python.

2. General Django Knowledge

Basic Django Concepts

- **What is Django/Python, and what is it typically used for?**
    - Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It is typically used for building web applications with database-backed systems.
    - For more, go to the pdf for created Interview Questions (for Python/Django).

3. Django Rest Framework (DRF) Knowledge

- **What is Django Rest Framework, and why would you use it over Django's built-in views for APIs?**
  - DRF is a powerful toolkit for building Web APIs in Django. It simplifies API creation with features like serializers, authentication, permissions, and more.
- **Can you explain what a serializer is in DRF and why it's used?**
  - A serializer in DRF is used to convert complex data types (like Django models) into JSON and vice versa. It's important for transforming data to be returned as an API response and for validating incoming request data.
- **What is the difference between ModelSerializer and Serializer in DRF?**
  - ModelSerializer is a shortcut for creating serializers based on Django models. Serializer is a more generic class that you can use to define custom fields and behavior.
- **How do you handle pagination in DRF? Can you explain the different types of pagination available?**
  - DRF provides built-in pagination classes such as PageNumberPagination, LimitOffsetPagination, and CursorPagination. These allow for controlling how API responses are split into pages.
- **What is the difference between a APIView and a ViewSet in DRF? When would you use one over the other?**
  - APIView is a class-based view that allows you to define methods like get(), post(), etc. ViewSet is used for standard CRUD operations and works with Router for automatic URL routing. APIView is used for custom behavior, and ViewSet is used when the operations are standard.
- **How would you authenticate an API in DRF? Explain how Token-based authentication works.**
  - Authentication can be implemented via classes like TokenAuthentication or SessionAuthentication. Token-based authentication involves generating a unique token for each user that is sent with each API request to verify the user's identity.

5. Testing

- **How do you test Django applications? What testing frameworks have you worked with?**
    - Django provides a built-in test framework using Python's unittest. The candidate may also mention using pytest or factory_boy for fixtures.
- **How would you write unit tests for a Django model?**
    - Create a test class in tests.py, use Django’s test client to simulate requests, and check model behavior using assertions like self.assertEqual().
- **How would you write tests for DRF APIs? What testing tools or strategies do you use for API testing?**
    - Use Django’s test client or DRF’s APIClient for testing endpoints. Tests can simulate API calls (get(), post(), etc.) and check for expected responses.
- **What is test coverage, and why is it important? How would you ensure sufficient test coverage for a Django project?**
    - Test coverage refers to the percentage of the codebase that is tested by unit tests. It’s important to ensure that all critical parts of the code are tested to avoid bugs in production.

6. Problem-Solving Questions

- **Imagine you have an API that takes too long to respond due to a complex database query. How would you approach optimizing it?**
    - Use database indexing, optimize queries with select_related or prefetch_related, paginate large result sets, and ensure database schema is optimized.
- **You have a Django project with a lot of users, and your database is becoming slow. What steps would you take to improve the performance and scalability of your application?**
    - Use caching (e.g., Redis), implement indexing on frequently queried fields, optimize database queries, and use database sharding if necessary.
- **How would you implement a feature that allows users to change their email address while keeping their previous email history?**
    - Create a EmailHistory model that stores the user's old email addresses and their change timestamps. Link this model to the user model using a ForeignKey.
- **If you were building a RESTful API for a blogging platform, how would you handle comments and replies in terms of API design?**
    - Use nested serializers to handle the hierarchical relationship between posts and comments. Each comment can have a parent comment for replies.
- **What would you do if an API endpoint you created starts throwing an error during production, but it works fine during testing?**
    - Check the logs, ensure the environment variables and configurations are correct, verify database migrations, and check for any differences between test and production environments.

7. Behavioural Questions

- **Describe a situation where you faced a challenging bug. How did you go about debugging and fixing it?**
    - Look for their ability to explain how they identify issues, use debugging tools, and systematically narrow down the root cause.
- **How do you approach learning new technologies or frameworks?**
    - Look for a growth mindset, mentioning practices like reading documentation, following tutorials, experimenting with sample projects, etc.
- **How do you prioritize tasks when working on multiple projects?**
    - Look for answers that show they can assess urgency, importance, and complexity to prioritize tasks effectively.
- **Have you worked in a team environment before? How do you handle conflicts or disagreements in a team?**
    - Look for answers that show teamwork, conflict resolution, and collaboration.
- **How do you manage your time when there is a tight deadline for a project?**
    - They should talk about task prioritization, breaking down tasks into smaller chunks, and managing time efficiently.

8. Hands-on Task (if time allows)

Go to PDF.

9. Closing

- **Do you have any questions for me about the role, the company, or the team?**
    - This will show if they are truly interested in the position and if they have done their homework about your company.
- **What excites you the most about working with Django and DRF?**
    - They should express a genuine interest in the technology stack.
- **Where do you see yourself in the next 2–3 years as a backend developer?**
    - Look for growth-oriented answers, ideally showing that they want to deepen their knowledge and take on more responsibility.