■ Techno MR Round – Technical Q&A; (Python & FastAPI)

1. **What are Python’s key features that make it suitable for backend development?**

■ Easy syntax, large ecosystem, strong libraries (FastAPI, Flask, Django), async support, and excellent community.

2. **Difference between List, Tuple, and Set in Python?**

■ List: ordered, mutable. Tuple: ordered, immutable. Set: unordered, unique elements.

3. **What are Python decorators? Give an example.**

■ A function that modifies another function without changing its code. Example: @app.get("/") in FastAPI.

4. **Explain Generators in Python.**

■ Functions that use yield to produce items lazily instead of all at once → memory efficient.

5. **Difference between @classmethod and @staticmethod?**

■ classmethod: takes cls (class reference). staticmethod: no self or cls; just a normal function in a class.

6. **What are Iterators in Python?**

■ Objects that can be looped with __iter__() and __next__().

7. **What is FastAPI? Why is it popular?**

■ A modern async Python web framework with auto validation (Pydantic), async support, and Swagger docs.

8. **How do you validate data in FastAPI?**

■ Using Pydantic models → enforce types, email format, regex, etc.

9. **What is Dependency Injection in FastAPI?**

■ A way to share common logic (DB session, auth, configs) across routes using Depends.

10. **Explain JWT Authentication in FastAPI.**

■ Client sends credentials → server returns JWT → client uses token in Authorization header → server verifies token.

11. **How do you connect FastAPI with a database?**

■ Use SQLAlchemy + SessionLocal + dependency get_db().

12. **What is Middleware in FastAPI? Give an example.**

■ Functions that run before/after every request (e.g., logging, CORS).

13. **Explain async vs sync in FastAPI.**

■ Sync blocks thread (CPU tasks), async is non-blocking (I/O tasks).

14. **How do you handle errors in FastAPI?**

■ Using HTTPException(status_code=..., detail=...) or custom handlers.

15. **How do you test FastAPI applications?**

■ Using TestClient with pytest.

16. **What are common status codes in APIs?**

■ 200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Server Error.

17. **What is ORM and why use it?**

■ Object Relational Mapper → maps Python classes to DB tables → avoids raw SQL.

18. **Difference between PUT and PATCH in APIs?**

■ PUT replaces whole resource, PATCH does partial update.

19. **What are advantages of FastAPI over Flask?**

■ Async support, faster, Pydantic validation, auto docs.

20. **How do you secure passwords in FastAPI?**

■ Use hashing libraries like passlib[bcrypt], never store plain text.