Python Developer JD - Interview Questions & Answers (Hinglish)

Q1: Python mein reusable, testable aur efficient code likhne ke best practices kya hain?

A: Reusable aur efficient code likhne ke liye modular programming ka use karte hain jahan functions aur classes ko reuse kiya ja sake. Testable code ke liye unit tests likhte hain using `unittest` ya `pytest`. Yeh ensure karta hai ki har module independently test ho sake. Example: Ek database connection function likhna jo sab jagah reuse ho sake:

```
def get_db_connection():
    import sqlite3
    return sqlite3.connect('example.db')
```

Q2: Low-latency aur high-performance applications Python mein kaise design karte ho?

A: Aise applications ke liye hum asynchronous programming (`asyncio`, `aiohttp`) use karte hain, database queries ko optimize karte hain, aur caching jaise techniques (e.g. Redis) ka use karte hain. Load balancing aur multithreading/multiprocessing se bhi performance improve hoti hai. Flask/FastAPI with Gunicorn aur Nginx bhi isme madadgar hote hain.

Q3: Front-end elements ko server-side logic ke sath kaise integrate karte ho?

A: Django ya Flask jaise frameworks mein front-end elements ko backend se integrate karne ke liye templating engines (Jinja2) use karte hain. Agar front-end React/Vue hai, to REST APIs ya GraphQL ke zariye data exchange hota hai. Backend JSON response deta hai jo front-end render karta hai.

Q4: Backend development mein security aur data protection kaise ensure karte ho?

A: Security ke liye input validation, CSRF protection, HTTPS, JWT for authentication, rate limiting, aur secure headers use karte hain. Sensitive data (like passwords) ko hash karte hain using bcrypt or Argon2. Django jaise frameworks in-built protection provide karte hain.

Q5: Multiple data sources ko ek system mein kaise integrate karte ho?

A: Python mein SQLAlchemy ya Django ORM ka use karke multiple databases se interact kiya ja sakta hai. Alag-alag connection configurations banakar, ek unified logic likha ja sakta hai jo data sources ko combine kare. ETL process bhi isme help karta hai.

Q6: ORM kya hota hai aur uska use Python web development mein kaise hota hai?

A: ORM (Object Relational Mapper) Python classes ko database tables ke sath map karta hai. Isse SQL likhne ki zarurat nahi padti. Django ORM ya SQLAlchemy iske examples hain. Example:

```
class User(Base):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    name = Column(String)
```

Q7: Python mein threading aur multiprocessing ka kya use hai?

A: Threading se hum lightweight concurrent tasks perform kar sakte hain, lekin Python ka GIL isme limitation hai. CPU-bound tasks ke liye multiprocessing better hota hai kyunki yeh alag processes run karta hai. I/O-bound tasks ke liye threading use karte hain.

Q8: Server-side templating languages jaise Jinja2 ka kya role hota hai?

A: Jinja2 jaise templating engines Python ke backend se HTML generate karne ke liye use hote hain. Yeh variables aur control flow ko HTML mein embed karne dete hain. Example: {% for user in users %} {{ user.name }} {% endfor %} se dynamic content show hota hai.

Q9: Python backend ke liye unit testing aur debugging techniques kya hain?

A: Unit testing ke liye `unittest`, `pytest`, ya `nose` frameworks use karte hain. Debugging ke liye `pdb`, `logging`, aur IDE ka debugger tool use hota hai. Testing se bugs pehle hi pakad mein aate hain aur deployment reliable hota hai.

Q10: Git jaise version control tools ka kya role hai development mein?

A: Git se hum code changes track kar sakte hain, alag features ke liye branches bana sakte hain aur safely code merge kar sakte hain. Command examples: git commit, git push, git branch, git merge. Team collaboration ke liye yeh essential hai.