Full Stack Python Developer - 200+ Interview Questions

Python (Core + Advanced)

- What are Python's key features?
- Difference between Python 2 and Python 3.
- Explain Python memory management.
- What are Python data types?
- Difference between mutable and immutable objects.
- Explain Python's garbage collection mechanism.
- What are Python namespaces?
- Difference between shallow copy and deep copy.
- What are Python iterators and generators?
- What are Python comprehensions?
- Explain Python decorators.
- Explain context managers in Python.
- What is monkey patching in Python?
- What is Python's GIL (Global Interpreter Lock)?
- Explain Python's multithreading vs multiprocessing.
- What are Python metaclasses?
- Difference between `@staticmethod`, `@classmethod`, and instance methods.
- Explain Python exception hierarchy.
- What are Python descriptors?
- What are Python slots and their benefits?
- Explain duck typing in Python.
- What is the difference between `is` and `==`?

- How is memory allocated to variables in Python?
- What is Python's `with` statement?
- Explain Python's import system.
- What are Python packages and modules?
- Explain Python virtual environments.
- What is pip and pipenv?
- What are Python type hints and annotations?
- Explain data classes in Python.
- What is difference between abstract class and interface in Python?
- How does Python achieve polymorphism?
- What are Python mixins?
- What is MRO (Method Resolution Order) in Python?
- What is `__new__` and `__init__` in Python?
- Explain Python's `__slots__`.
- What are Python dunder methods?
- Explain Python's `__str__` vs `__repr__`.
- What are Python property decorators?
- What are Python weak references?
- What is memoization in Python?
- Scenario: API response is slow. How do you optimize Python code?
- Scenario: You must handle millions of rows in Python. How to optimize memory usage?
- Write a Python function to check if a string is a palindrome.
- Write Python code to reverse a linked list.
- Write a Python program to implement LRU cache.
- Write Python code to merge two sorted lists.
- Write Python program to find duplicates in a list.

- Write a Python program to detect a cycle in a linked list.
- Write Python program to implement binary search.
- Write Python program to generate Fibonacci numbers.
- Write Python program to check for prime numbers.
- Write Python program to count character frequency in a string.
- Write Python program to flatten a nested list.
- Write Python program to implement stack using queues.
- Write Python program to implement queue using stacks.

MySQL

- Explain relational databases.
- Difference between SQL and NoSQL databases.
- Explain primary key vs unique key.
- Explain foreign key with example.
- What are indexes in MySQL?
- When should indexes be avoided?
- Explain clustered vs non-clustered indexes.
- What is a composite index?
- Difference between DELETE, TRUNCATE, and DROP.
- What is normalization? Explain all normal forms.
- What is denormalization?
- Explain MySQL constraints.
- What are transactions in MySQL?
- Explain ACID properties.
- What are MySQL storage engines?
- What is difference between InnoDB and MyISAM?
- Explain MySQL joins (INNER, LEFT, RIGHT, FULL).

- What is a self-join?
- What is a cross join?
- Explain MySQL subqueries.
- What is difference between correlated and non-correlated subqueries?
- Explain MySQL views.
- What are stored procedures?
- What are triggers in MySQL?
- What are functions in MySQL?
- Difference between UNION and UNION ALL.
- How do you optimize slow queries?
- Scenario: Table with millions of rows, SELECT is slow. How to optimize?
- What are covering indexes?
- What is query execution plan?
- Write SQL to fetch nth highest salary.
- Write SQL to count employees in each department.
- Write SQL to find duplicate records.
- Write SQL to find employees who don't have managers.
- Write SQL to update salary of employees by 10%.
- Write SQL to delete duplicates keeping one.
- Explain database sharding.
- Explain partitioning in MySQL.
- Explain replication in MySQL.
- What is MySQL clustering?

Django

- Explain Django's MVT architecture.
- Difference between Django and Flask.

- Explain Django ORM.
- Difference between ORM and raw SQL.
- Explain Django models and migrations.
- What are Django forms?
- Difference between ModelForm and Form.
- What are Django signals?
- Explain middleware in Django.
- What are Django sessions?
- What is Django caching framework?
- Explain select_related and prefetch_related.
- What are Django template tags and filters?
- How are static files handled in Django?
- Difference between static and media files.
- What is CSRF in Django?
- How do you secure a Django application?
- What are Django class-based views?
- Difference between FBV and CBV.
- What are mixins in Django?
- Explain authentication and authorization in Django.
- How does Django handle passwords securely?
- What is the difference between OneToOne, ForeignKey, and ManyToMany?
- Scenario: Add login/logout functionality in Django.
- Scenario: Optimize a slow Django query.
- Scenario: Handle file upload in Django.
- Scenario: Deploy Django on AWS.
- Scenario: Debug Django app not working on production.

- Explain Django REST Framework (DRF).
- What are serializers in DRF?
- Explain ViewSets and Routers in DRF.
- What are throttling and rate-limiting in DRF?
- How do you handle pagination in DRF?
- Scenario: Build a REST API for Employee Management in Django.
- Scenario: Add JWT authentication in Django REST API.
- What is Django Channels?
- Explain asynchronous support in Django.
- Explain signals in Django with example.
- What is difference between caching at view level and template level?

HTML/CSS

- What is semantic HTML?
- Explain HTML5 new features.
- Difference between block and inline elements.
- What are HTML attributes?
- What are HTML5 forms?
- What are meta tags in HTML?
- What is the difference between relative, absolute, and fixed positioning?
- Explain CSS specificity rules.
- Difference between inline, internal, and external CSS.
- What are CSS pseudo-classes?
- What are CSS pseudo-elements?
- Explain CSS box model.
- What is the difference between Flexbox and Grid?
- Scenario: Debug CSS not applying in a webpage.

- Scenario: Page looks different in Chrome and Firefox. Debugging steps?
- How do you make a website responsive?
- What are media queries in CSS?
- What are CSS animations?
- Explain CSS transitions.
- What is difference between em, rem, px, and % units?
- What is z-index in CSS?
- What are CSS variables?
- What are shadow DOM and web components?
- What is difference between absolute and relative paths in HTML?

JavaScript

- What are JavaScript data types?
- Difference between == and ===.
- What are closures in JavaScript?
- Explain promises in JavaScript.
- What is async/await?
- What is event loop in JavaScript?
- What are callbacks in JavaScript?
- What are arrow functions?
- Explain var, let, and const.
- What is hoisting in JavaScript?
- What is scope in JavaScript?
- What is difference between local and global scope?
- What is prototypal inheritance?
- Explain ES6 modules.
- What are JavaScript classes?

- What is difference between function declaration and expression?
- What is event bubbling and capturing?
- Explain event delegation.
- What are JavaScript timers?
- What is setTimeout and setInterval?
- Explain JSON and its usage in JavaScript.
- What is AJAX in JavaScript?
- Scenario: Make sequential API calls in JavaScript.
- Scenario: Handle multiple API calls in parallel.
- What are fetch API and XMLHttpRequest?
- What is difference between synchronous and asynchronous JS?
- What are JavaScript promises states?
- What are generators in JavaScript?
- What is difference between call, apply, and bind?
- What is this keyword in JavaScript?
- Scenario: Debug undefined errors in JS.
- Scenario: Optimize JS-heavy webpage.
- Explain CORS in JavaScript.
- What is localStorage, sessionStorage, and cookies?
- Explain debouncing and throttling.
- What is difference between DOM and Virtual DOM?
- What are JS frameworks and libraries?
- Explain React vs Vanilla JS difference (if asked in full stack role).

Full Stack Scenario-Based

- Scenario: Build a Passport Management System with Django and frontend.
- Scenario: Optimize Django app for high traffic.

- Scenario: Debug Django app not starting on server.
- Scenario: API returns 500 error randomly. Debugging steps?
- Scenario: Database connection pool exhausted. How to fix?
- Scenario: Static files not loading in Django. Debugging steps?
- Scenario: Website loads slowly. How to optimize backend?
- Scenario: Website loads slowly. How to optimize frontend?
- Scenario: User login taking too long. Debugging steps?
- Scenario: Build and secure REST APIs in Django.
- Scenario: Deploy Django + MySQL on AWS.
- Scenario: Integrate third-party payment gateway in Django.
- Scenario: Implement caching for API responses.
- Scenario: How do you design scalable architecture for full-stack app?
- Scenario: Fix CORS issue in frontend-backend integration.
- Scenario: Debug Django queries taking too long.
- Scenario: Implement file upload feature securely.
- Scenario: Add JWT authentication in Django REST API.
- Scenario: Handle frontend form validation with JS.
- Scenario: Integrate React/JS frontend with Django REST API.