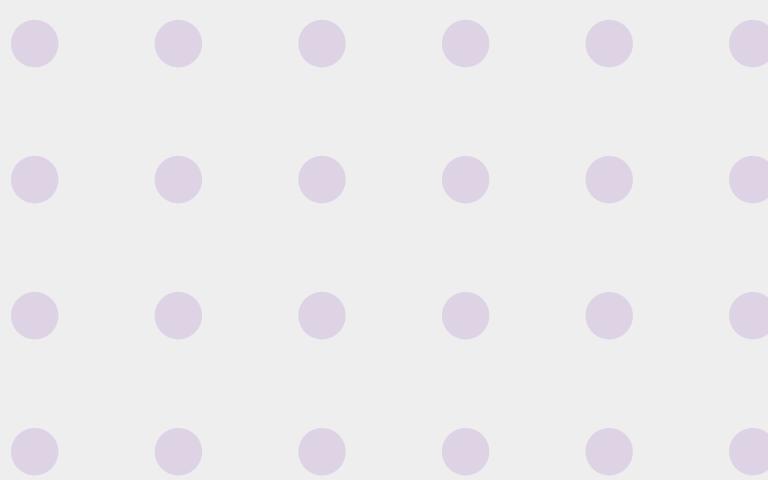


**Experience - 200**



**PhonePe**  
**SDE 2**

**Interview Experience**



**PhonePe**



# Summary

-  **Job Role:** Software Development Engineer
-  **Number of Rounds:** 4
-  **Offer Status:** Offer
-  **Location:** Bangalore
-  **Candidate Name:** Not disclosing due to signed NDA



# Interview Process:

- The interview was conducted virtually for the Bangalore location. It consisted of 4 rounds in total – Machine Coding Round, Problem Solving/Data Structures Round, System Design Round, and Hiring Manager Round.
- After completing all the rounds, I was informed within 1 working day that I had successfully cleared the interviews and they would extend me an offer.



# Preparation Guide

- The interviews at PhonePe were a mix of hands-on coding, data structures and algorithms, system design (both HLD and LLD), and behavioral discussions.
- Key areas to prepare before such an interview include:
- Object-Oriented Programming and Machine Coding Practices
  - focus on code reusability, extensibility, and clean design principles.
- Problem Solving & Data Structures – interval scheduling, greedy algorithms, graph problems, and common LeetCode-style questions.
- System Design – microservices architecture, database design (SQL vs NoSQL), sharding, replication, and scaling solutions.
- Behavioral Questions – clear articulation of past project experiences, decision-making in critical situations, and teamwork.



# Round 1: Machine Coding

- Duration: 120 minutes (plus code review)
- Difficulty Level: Medium
- **Experience:**
  - This was an offline round lasting 2 hours, followed by a code review session. I was asked to design a Pendency System with the following requirements:
  - Start the tracking of an Id with a hierarchy of [instrument, state, city].
  - Stop tracking of Id.
  - Get the count of active Ids in any valid hierarchy, e.g. active counts in [UPI, Karnataka] or [UPI].
  - The expectation was that the latency of the Count API should be minimal. The review session focused on how I structured my classes using OOP principles, whether the code was extensible, and how well I applied concepts like code reusability and separation of concerns.
- **Key Learnings:**
  - Structure your classes properly and think about extensibility from the beginning.
  - Follow OOP principles strictly—clean abstractions and encapsulation matter.
  - Keep APIs optimized for latency.



# Round 2: Coding

- Duration: 60 minutes
- Difficulty Level: Medium
- **Experience:**
  - This was a pure problem-solving round with two coding questions:
  - Gas Station Problem – a classic greedy algorithm problem.
  - Interval Scheduling with Profits – given a list of intervals with profits, find the maximum profit by scheduling work such that only one task is active at a time.
  - Both problems tested my ability to quickly analyze the constraints, choose the right algorithmic strategy, and write optimized code.
- **Key Learnings:**
  - Practice greedy problems and interval scheduling algorithms thoroughly.
  - Focus on clarity and correctness over premature optimization.
  - Get comfortable solving problems within tight time limits.



# Round 3: System Design

- Duration: 60 minutes
- Difficulty Level: Difficult
- **Experience:**
  - This round focused on designing a social network-like system with the following features:
  - Create Posts (with text, image, or video).
  - Add connections (friends/contacts).
  - View Feed on the timeline, which includes posts from connections.
  - I started with the high-level design (HLD), then went deeper into the low-level design (LLD). The interviewer tested me on:
  - Microservices architecture and communication.
  - Database design (SQL vs NoSQL, federated DBs).
  - Handling scalability using DB sharding, replication, and caching.
  - Load balancers, API gateways, and async processing for feeds.
  - The discussion was highly interactive, with back-and-forth tradeoff questions around scalability, fault tolerance, and performance.
- **Key Learnings:**
  - Be thorough with system design fundamentals.
  - Always discuss tradeoffs—why you are choosing one approach over another.
  - Know scaling strategies (sharding, replication, cache, load balancing).



# Round 4: Hiring Manager

- Duration: 60 minutes
- Difficulty Level: Medium
- **Experience:**
  - This was more of a behavioral and managerial round. I was asked about my past projects, challenges I faced, and how I solved them. There were situational and behavioral questions to assess my problem-solving style and communication.
  - I was done with formal questions in about 40 minutes, after which the discussion shifted to PhonePe's teams, organizational structure, and work culture.
- **Key Learnings:**
  - Prepare to talk in-depth about your projects—both technical and decision-making aspects.
  - Reflect on challenges you've faced and how you handled them.
  - Be curious and ask about the company's culture, it shows genuine interest.



# Final Thoughts:

- The overall interview process at PhonePe was balanced and rigorous.
- The Machine Coding round set the tone by testing clean coding practices.
- The Problem Solving round ensured strong algorithmic skills.
- The System Design round was challenging and tested the ability to think at scale.
- The Hiring Manager round ensured cultural and behavioral fit.
- For candidates preparing for PhonePe:
  - Strengthen OOP and coding practices.
  - Regularly solve LeetCode Medium/Hard problems.
  - Practice system design discussions with peers.
  - Reflect deeply on your past projects and experiences.
- Overall, it was a well-structured process, and I was happy to receive the offer.



Experience Link

# Thank You

Roundz.ai

---

Follow us

