Experience - 158

Oracle
SMTS | IC3
Interview Experience

Summary

📌 **Job Role**: Software Development Engineer
🔢 **Number of Rounds**: 6
📜 **Offer Status**: Offer
📍 **Location**: Hyderabad
👤 **Candidate Name**: Not disclosing due to signed NDA

Interview Process:

- The interview process at Oracle OCI for the SMTS (IC3) position consisted of one technical phone screen followed by an onsite interview with five rounds.
- Each onsite round was 1 hour long.
- The process included a mix of algorithmic coding challenges, low-level design, behavioral assessments, and system design style questions.

Preparation Guide

- While there was no specific preparation source mentioned, based on the nature of the questions, a strong preparation plan would include:
- Revisiting data structures and algorithms fundamentals (especially arrays, matrices, and mathematical problem-solving without built-in functions).
- Practicing low-level design (LLD) problems, focusing on defining classes, methods, and handling feature requirements.
- Reviewing system design principles for small to medium-scale systems.
- Preparing for standard behavioral interview questions, with clear examples from past work experience.
- Revisiting all resume points in depth, as multiple rounds heavily referred back to previous work and academic projects.

Round 1: Technical

- **Duration**: 60 minutes
- **Difficulty Level**: Medium
- **Experience**:
- The round began with a short introduction, followed by questions on my previous work experience and Master's Thesis. After this, the interviewer presented a Hackerrank-style problem:
- Problem Statement:
- Given an m × n matrix of 0s and 1s where each row is sorted, find the lowest index of 1 across all rows.
- Example:
- [0 0 0 1 1] [0 0 0 0 1] [0 1 1 1 1] [0 0 1 1 1] Index of 1 in each row: Row0 - 3 Row1 - 4 Row2 - 1 Row3 - 2 Answer = 1 (lowest index)
- Follow-up Questions:
- Optimize the solution.
- Modify the approach if each row has a different number of elements.
- Example for follow-up:
- [0 0 1] [0 0 0 1 1] [0 1 1 1] [0]

Round 1: Technical

- **Key Learnings:**
- Be prepared to explain your academic and project experience in depth.
- Practice matrix manipulation problems and approaches for optimization.
- Think about edge cases such as variable row lengths.
-

Round 2 : LLD

- **Duration**: 60 minutes
- **Difficulty Level:** Medium
- **Experience:**
- Started with a brief introduction followed by behavioral questions:
- What was your most challenging work?
- How do you decide between two approaches to solve a task?
- Main Task:
- Design a Music Player application with the following requirements:
- User should be able to select a song.
- Play a song.
- Add a song.
- Remove a song.
- Create a playlist – add/remove songs, play playlist.
- Shuffle playlist.
- Expectations:
- Define the classes, member variables, and methods for the above requirements.
- Implement the shuffle playlist functionality.
- **Key Learnings:**
- LLD questions often require a balance between completeness and clarity.
- Understand object-oriented design concepts well.
- Be ready to implement utility functions (e.g., shuffle logic).

Round 3 : Problem Solving

- **Duration**: 60 minutes
- **Difficulty Level**: Hard

- **Experience**:
- This round began with around 15 minutes discussing previous work experience, followed by a challenging problem:
- Problem Statement:
- Given an array of integers, find the longest perfect square chain.
- Definition: A perfect square chain is a sequence where each number is the perfect square of the previous number.
- Example:
- A = [2, 3, 4, 5, 9, 16, 25, 81, 256] Chains: 4 -> 16 -> 256 (length = 3) 9 -> 81 (length = 2) 25 (length = 1) Answer = [4, 16, 256]
- Constraints:
- Do not use Math.sqrt() or any direct square root function.

Round 3 : Problem Solving

- **Key Learnings:**
- Some problems may intentionally restrict built-in functions to test mathematical thinking.
- Be comfortable deriving mathematical properties manually.
- Practice writing clean and modular code for complex array manipulations.

Round 4 : Behavioural

- **Duration**: 60 minutes
- **Difficulty Level**: Medium

- **Experience:**
- Purely behavioral. The interviewer went through almost every point in my resume, asking in-depth follow-up questions. Questions were standard but expected detailed, specific answers.

- **Key Learnings:**
- Review your resume thoroughly.
- Prepare real, concrete examples to support every point you've listed.

Round 5 : Coding

- **Duration**: 60 minutes
- **Difficulty Level**: Medium

- **Experience:**
- Problem: Design Browser History (LeetCode).
- The task required designing classes and methods to manage browser navigation history with features like visiting URLs, moving backward, and moving forward in history.

- **Key Learnings:**
- LLD-type coding problems often combine data structure usage with object-oriented design.
- Practice implementing small system features cleanly.

Round 6 : Hiring Manager

- **Duration**: 60 minutes
- **Difficulty Level**: Medium

- **Experience:**
- Standard behavioral questions, including:
- What challenges have you faced at work?
- How do you deal with a difficult person?
- Describe a time when you took a risk.
- Critical feedback you received from peers and how you handled it.
- A time when you disagreed with a teammate and how it was resolved.

- **Key Learnings:**
- Hiring Manager rounds focus on alignment with team values and leadership principles.
- Be ready with STAR (Situation, Task, Action, Result) format responses.

Final Thoughts:

- Oracle OCI’s SMTS interview process was thorough and tested both technical and behavioral competencies.
- A balanced preparation across coding, design, and behavioral questions is essential.
- Resume depth matters — be prepared for repeated references to your past work.
- Avoid over-relying on built-in functions; they may be restricted in some problems.
- LLD practice should be part of your preparation strategy, especially for IC3 and above roles.

Experience Link

Thank You

Roundz.ai

Follow us

in **roundz-ai**

X **roundz-ai**