

Appian Interview Experience

Role: Software Engineer

Online Test (2nd Aug 2025):

The online coding round consisted of 3 questions to be solved in 75 minutes.

Questions:

1. Debugging Problem:

We were given a problem statement along with a code containing errors. Our task was to debug the code so that all test cases passed. This problem had already been asked by Appian during our internship, but with a slightly simpler implementation at that time.

In the original problem, we were given a list of employees and the teams they work on. Then, we were provided with an array containing a sequence of employee names. If all employees from a particular team appeared consecutively in the array, we had to replace them with the team name. Otherwise, we would just add their names to the output.

For this test, the company had intentionally complicated the logic. They added extra conditions to decide which team an employee belongs to based on their first assigned team. Additionally, they used three HashMaps, six HashSets, and two global HashMaps to make the debugging more challenging.

2. Next Permutation Problem:

This question is a popular next permutation leetcode problem wrapped in a story that they are using this permutation for flight ticket number.

3. Treasure Hut:

This was a graph problem. We were given a matrix where we could navigate in four directions from a point. The task was to find the shortest path from the starting point to the treasure point (T). Some cells contained dead ends marked with X. The solution required using BFS.

I skipped 1st qn and moved to the permutation problem, Where i go near to solution but done a mistake on choosing the jth element to swap with so got failed testcases so i moved on to 3rd, solved using BFS and and returned to the 2nd problem, and completed it successfully.

By then, I had only 20 minutes remaining, I go to debugging question, First i read the entire code once a stretch they had clearly made it complex by using multiple HashSets and HashMaps in different places so I decided to debug from bottom to top, starting by printing the final output and tracing it backwards. The bugs are like they have wrongly placed a not operator in if, Missing logic to add an element to a set when a certain flag remained true like that. After debugging 3 bugs I got 8/9 testcases passed and I have only 1min remaining so I didn't risk to change any code further as it may make more test cases fail so I left it as it was.

Interview (5th Aug 2025):

A total of 33 students were shortlisted for the Software Engineer role from CEG and MIT. My interview slot was scheduled from 10:30 AM to 11:15 AM, but I reached CUIC, CEG by 9:00 AM. Candidates were called in according to their allocated slots.

The interview consisted of three rounds, one Technical Round and two Behavioural Rounds.

Take 3 copies of your resume

1st Round:

The round began with the interviewer reviewing my resume. His first question was: "**What do you think is the best thing about all the projects you have done so far?**". I replied that the main highlight across all my projects is that they are built as complete, ready-to-use products for the end user, not just as implementations of single features. I elaborated by explaining what each project does, how it works, and justified my point by showing that every project delivers a complete, usable solution.

Then he asked me to explain about one of the projects. It is named as DevAssist, a GenAI-based project to help visually impaired developers. I explained what it does. He asked questions about why you chose this project, what tech stack and how it differs from existing tools etc.

Next, he moved on to a DSA problem. He handed me a pen and paper and described the question:

We needed to build a **room booking system** with n rooms. Based on the requested time slots, we had to determine whether a booking request could be accepted or not. Initially, the problem sounded similar to the classic "**Minimum Number of Platforms Required**" greedy problem.

So I cross checked by asking about the type of input and expected output. but when I asked a question, can we allocate any of that n room available in that slot he realized I had identified the greedy solution pattern, so he modified the problem that each request was for a specific room number and we have to suggest the other possible options if the slot is not free.

The solution to this question was straightforward. I suggested using a **HashMap** to check for room availability. However, the interviewer then mentioned that the booking requests would not necessarily be in chronological order — for example, someone could request a slot from 4:30–5:30 at 3:30 PM or even at 4:15 PM. I then proposed modifying the approach to use a HashMap where the key is the room number. Store each room's bookings in a sorted ArrayList of time slots. When checking availability, scan this list to determine if the requested slot is free and, if not, suggest the next possible slot.

This part of the interview turned into more of a discussion with continuous follow-up questions rather than coding. I didn't write any Java code instead, I provided pseudocode and rough sketches of the data structures.

For optimization, I suggested merging contiguous slots to make availability checks faster and improve space efficiency. He then asked how I would clear past slot data. At this point, he shared his own idea — not merging overlapping slots but instead clearing them periodically through time-based checks.(yeah the DSA question has diverted to Devops like perspective on our discussion)

I explained that while both approaches were similar in terms of space complexity, his method could result in multiple calls for clearing past data if the slots were small. In contrast, my approach of merging slots would reduce the number of such calls,so I told him that I prefer this solution. He seemed impressed by this reasoning and decided to conclude the DSA question there.

The interviewer then returned to my resume and asked about my startup experience. Initially, he assumed I had interned at a startup, but I clarified that it was actually my own startup, which we later discontinued due to issues with our revenue model and the boom of quick commerce.

In explaining my learnings in startup I added my hackathon experience, where our team won by applying insights from our startup work. He asked a few follow-up questions about the hackathon, and we wrapped up the round.

This round lasted quite a while, but by the end, I felt confident that I had made a strong impression.

Behavioural Round (2nd Round):

After the first round, some students were shortlisted for the **Behavioural Round**, and only they moved forward.

This round was conducted **online**. A laptop was set up in a room, and the HR interviewer joined via video call. On the way to the room, one of the Appian organizers advised me to stay calm and suggested me to be strong in my selected SE role.

The round began with some basic questions about my school marks, my cutoff score, and why I chose MIT. The interviewer then asked, "What do you know about Appian?"

Appian's main working field is low code technology, I had built a project in my second semester called **CustCom** (built in MERN), which also aligns with no-code/low-code concepts. So I decided to push my project. I explained the project and expressed my interest in the low-code space. He then asked me about the **difference between low code and no code**, which I explained in detail. He appreciated my work, and then mentioned:

"If you are interested in low code, you could go for the Product Engineer role, where we work directly with low-code platforms. In the Software Engineer role, we only build low-code solutions."

I responded:

"Sir, I love to build low-code solutions rather than just working with low-code platforms, so I prefer to stay in the Software Engineer role."

He agreed with my reasoning. He then asked if I had any questions for him. I asked:

"What does a typical day look like for a Software Engineer at Appian, and what kind of projects would we be involved in?"

He gave me a detailed explanation about the kinds of tasks, responsibilities, and projects that Software Engineers handle at Appian. The round then concluded.

Behavioural Round (3rd Round):

We moved to this round without any further shortlisting. It was conducted in person.

The interviewer began by asking about my **native place** and **my father's occupation**. He then asked about my **CGPA** and **class rank**. I wasn't sure about my exact rank, so I said I wasn't sure. *(Tip: If you are within a good rank range, like the top 10, it's worth mentioning. Otherwise, it's better to say you're not sure)*

He then asked me about my **project experience** and the **name of the interviewer from my technical round**. This round was relatively short.

Towards the end, he asked if I had any questions. I paused for a moment to think, but before I could respond, he suddenly said, *"If you don't have any questions, that's fine, it's not necessary to ask."* This made me a bit panicked. However, I quickly managed to respond that I had already clarified my doubts in the previous rounds. He said that was good and then asked me to leave.

Important Note:

Please don't leave the campus until the PR officially announces that everyone can go.

After my 3rd round, I informed the Appian organizer, and she said, *"Okay, you can leave."* So I thought it was fine to go home since the results would only be posted in the placements group. I boarded a bus to return home, but suddenly a message popped up in the placements group saying that Candidates are requested to remain on campus until the decision-making is over, as they may be recalled. I completely panicked, got down at the very next stop, and rushed back to CUIC by Rapido (honestly, the most tense ride I've ever taken to CEG).

In the end, they didn't recall any of us. After about two hours, they finally told us we could leave. The results came out by 8.50 PM, and six of us were selected.

Final Tip:

- Be well-prepared to explain your projects, with short and clear descriptions for each.
- During DSA problems, don't just think silently, speak your thought process aloud so the interviewer will be engaged and can follow your reasoning and problem-solving skills.
- For a single question, try to answer in a way that showcases multiple skills like self learning, tech stack choosing, adaptability etc.
- Study about the company beforehand and if any project or your work align to it push it in the interview.
- Stay confident in yourself throughout the process.

All the best!!

Feel free to contact for any doubts,

Lokesh D,

(2022503519)

Department of Computer Technology, MIT Campus,

lokeshd0506@gmail.com [LinkedIn](#)