Interview challenges

Hiring people is the most important step to build a good organization. If they fail in building a good team or if they delay in hiring, the organization will also fail in project execution.

A normal hiring workflow process usually includes following high-level steps,

- Step-1 : Demand generation by project execution team
- Step-2 : Collect, screen & shortlist matching profile from different sources
- Step-3 : N-rounds of interview process as per organization policy
- Step-4 : Make offer

Here the hiring team is able to manage all steps independently on their own, but when it comes to step-3 i.e. [**Interview process**], they are dependent on a panel of experts.

Finding a slot from the [**Interview panel**] is becoming difficult day-by-day as the panel is already occupied so much in their day-to-day transactional work that it becomes very difficult for them to arrange an interview slot.

Another challenge is that if any expert has been voluntarily nominated, the hiring team takes them for granted and overloads them with piles of profiles. So many times, it looks like the hiring team is not fair and transparent with volunteer panelists by having evenly distributed profiles among all panelists. In most organizations taking an interview is like a support work to be done as charity work without any reward or [**incentivizing per interview**].

Ultimately the hiring team is not able to find the slot in time, which causes the delay in hiring and which further ***impacts the overall project time-line***. With the emergence of Generative AI, there are many real-life use-cases which can be performed comfortably i.e. without or lesser pain & frustration through the process.

I have shared articles in the past with the AI perspective into different areas e.g. health, education, legal, etc. In the same row, I wrote this article explaining how AI can help in the [**Interview process**]. I did the complete rehearsal of interviewing 3 candidates using ChatGPT & Gemini and feedback preparation, which is shared with the hiring team.

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

Objective :

[AI PanelPro] to solve the problem of human panel unavailability

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

Audience :

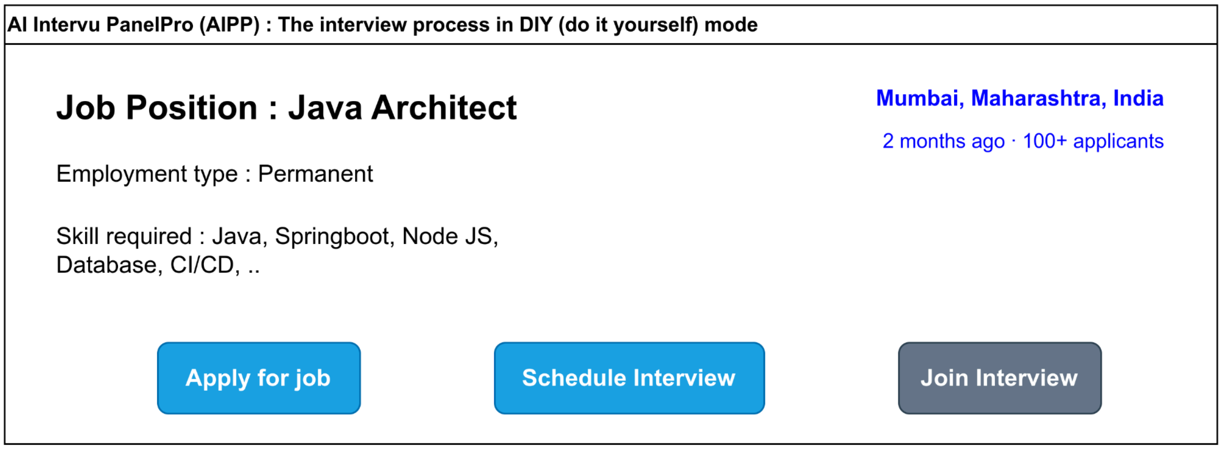
Since interview is an ongoing process in any organization, to support the interview process I foresee a growing need of [AI PanelPro] in future, which can streamline the interview process. Initially, it can be planned to handle smaller positions and early rounds (however it will have potential to support all rounds); it could be started with technical positions or lower-pyramid positions.

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

AI PanelPro : an AI assistant

In order to assist in [**Interview process**], I foresee an application/platform enabled with AI agent which manage only one process i.e. [**Interview process**] of complete hiring workflow. For the sake of focused discussion, I will name the AI agent as [**AI PanelPro (AIP)**] in the SaaS model. Existing hiring system will integrate [**AI PanelPro (AIP)**], so that the candidate's journey from [**Apply to Interview**] becomes [**DIY (do it yourself)**] like experience i.e.

a. Apply job [self assisted]
b. Schedule interview [self assisted]
c. Join interview [self assisted]



AI Intervu PanelPro (AIPP) : The interview process in DIY (do it yourself) mode

Job Position : Java Architect

Mumbai, Maharashtra, India

2 months ago · 100+ applicants

Employment type : Permanent

Skill required : Java, Springboot, Node JS, Database, CI/CD, ..

In this article, I will share details about my approach to how the application/platform [**AI PanelPro (AIP)**] is simulated using available LLM tools in the market.

>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<

AI PanelPro : Strategy & Approach

1. **Get the detailed [Job Description]**
   Project team who raises the demand should provide the detailed [**Job Description**] to the hiring team. Here [**Job Description**] plays an important role to filter the candidates profile e.g. [**Job Description**] should be covering
   a. Technology skill required
   b. Domain skill required
   c. Behavioral skills required
   d. Overall and skill wise experience required in months
   e. Minimum education qualification
   f. Certification & Rewards
   g. Must-have & Good-to-have skills
   h. Others (If anything important, optional)

2. **Prepare the [Question sets]**
   Usually questions vary based on role, technology group and domain. Sametime it can be felt that [**Intent to know**] from the question may be the same, but the style of asking the very same question will vary from one interviewer to another.

   Based on my personal experience and friend's experience who have been part of countless interview panels throughout their career, I feel that many questions can be standardized under the category of [**Roles, Technology & Domain**]. There will be some contextual questions based on the candidate profile. Based on need, the same standard questions can also be contextualized considering the candidate profile.

   Questions under each [**Question sets**] based on [**Roles, Technology & Domain**], will have a predefined average duration time, in which the candidate will be requested to articulate the best answer with maximum coverage to the intent of the question.

   [**AI PanelPro (AIP)**], will have sufficient knowledge in terms of content and FAQs applicable for [**Roles, Technology & Domain**]. However, LLMs available in the market are so knowledgeable that local level knowledge management may not be important for answer evaluation.

3. **Pre-Interview Process**
   [**AI PanelPro (AIP)**] will have the intelligence that it can send auto-reminders via call / email notifications. [**AI PanelPro (AIP)**] will parse the job description and candidate profile to align expectations.
   a. [**AI PanelPro (AIP)**] will identify core skills and nice-to-have skills.
   b. [**AI PanelPro (AIP)**] will fetch or prepare the [**Question set**] for required [**Roles, Technology & Domain**] salting along with difficulty & importance.

   c. Using the [**Question set**], [**AI PanelPro (AIP)**] will prepare the [**Interview conversation flow**] in advance, which shall be used for asking questions during the actual interview in a well-prepared sequence.

4. **During Interview Process**
   [**AI PanelPro (AIP)**] will start with an **icebreaker** to make the candidate comfortable and initiate the **introductory questions** to gauge familiarity with core skills. To avoid any confusion or mismanagement of time, [**AI PanelPro (AIP)**] should go through the [**Interview conversation flow**] prepared in [**Pre-Interview Phase**].

   [**AI PanelPro (AIP)**] will ask one question at a time and will also inform the candidate to respond to the answer in the proposed answer window.

5. **Post Interview Process**
   [**AI PanelPro (AIP)**] will aggregate and analyze responses. [**AI PanelPro (AIP)**] will send detailed feedback in a predefined template and additional key-points or remarks and **scorecards** to help the hiring team take next-step.

   Optionally, [**AI PanelPro (AIP)**] can suggest the **next steps**: (e.g. recommend for the next round, training suggestions, or rejection with feedback).

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

AI PanelPro : Key design consideration

1. **2-way uninterrupted communication**
    - Interviewer will finish the complete question first, then request the candidate to answer.
    - Candidate will reply to the complete answer to the very same question only, not any other question.
2. **Counter questions**
    - Based on the candidate's complete answer, there may be some counter question or clarification required.
    - Any counter question should also be processed in [**2-way uninterrupted communication**] mode only.
    - [**Counter questions**] are not considered in the simulation
3. **Question level feedback & assessment**
    - To make the hiring process transparent, feedback can be shared for each question with high-level [Right/Wrong] in the answer
    - [**Question level feedback & assessment**], can be shared as filler between the two questions i.e. after complete answer for last question asked and before next-question to be asked.
    - [**Question level feedback & assessment**] are not considered in the simulation
4. **Applicability in industry**
    - Organizations may decide which roles they would like to take help from AI.
    - In the IT world, to my view delivery roles like developer, tech-lead, technical architect, scrum-master, project-manager, others who do not play CXOs roles, should be the preferred roles to start with.
    - However, being from the IT industry there will be a shadow of IT industry unintentionally, [AI Interview Agent]
5. **Not a replacement of Human Panel**
    - The intent of [**AI PanelPro (AIP)**] is not to replace the human-panel, rather this is to solve the issue of slots availability by human-panel.
    - Just like peer team member, [**AI PanelPro (AIP)**] will be the co-panelist, who may ask question on-behalf of human
    - [**AI PanelPro (AIP)**] can work in both mode either lead-panel role or shadow-panel role.
6. **Ethical practices & potential fraud**
    - [**Interview process**] is one the high volume task, just like payment-transaction a high-volume activity. The main objective of [**AI PanelPro (AIP)**] is to enable humans to do more quality tasks of thinking, evaluating the candidate during the answer as co-panelist.
    - [**AI PanelPro (AIP)**] will also detect fraudulent activities at candidate’s premises with use of camera, microphone and technology.
7. **Hallucination**
    - To avoid the hallucination issue, the process is being kept very simple

- These [**Key design considerations**] are planned to overcome the hallucination issues.

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<

Mock of interview process using LLM

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

Candidate-1 (Xxxx Kumar) E2E Interview Process

Candidate-1 (Xxxx Kumar) Interview Scheduling

The hiring system sends confirmation to both the candidate and the employer.

Subject: "Interview Scheduled: Java Architect Role at NPCI"

```
Dear Xxxx Kumar,

Thank you for applying for the Java Architect position at National Payments
Corporation of India (NPCI). Your interview has been scheduled as per your preference.

**Date:** Sunday
**Time:** 15:00 PM IST
**Mode:** Virtual Interview Panel (Online)

Please ensure you are available at least 10 minutes before the scheduled time. Use the
link below to join the interview:
[Join the Interview](#)

Best regards,
NPCI Recruitment Team
```

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

Pre-interview process for candidate-1 (Xxxx Kumar)

The hiring system sends reminder emails to both the candidate and the employer. [**AI PanelPro (AIP)**] will analyze the candidate profile and job description. [**AI PanelPro (AIP)**] prepares the expected questions as follows for the candidate interview.

Question 1:
"You've mentioned experience in microservices. Can you describe a recent project where you

implemented microservices architecture? Specifically, what design principles and tools did you use, and what challenges did you encounter?"

Response Window: 4 minutes

Question 2:
"In large systems, performance is a critical factor. Can you explain a scenario where you optimized the performance of your application? What strategies and tools did you use to achieve this?"

Response Window: 3 minutes

Question 3:
"What is your understanding of API-first design, and how have you applied this principle in your previous projects?"

Response Window: 3 minutes

Question 4:
"How do you ensure security-first design in your applications? Can you provide an example of implementing security measures, such as authentication and data protection, in a project?"

Response Window: 4 minutes

Question 5:
"Could you explain how you handle reusable code in your projects? For instance, how do you design project directories or manage reusable components like database models or utility libraries?"

Response Window: 3 minutes

Question 6:
"You've worked with both SQL and NoSQL databases. Can you compare them in the context of one of your projects and explain why you chose one over the other?"

Response Window: 3 minutes

Question 7:
"As a technical leader, how do you ensure your team follows best practices for writing unit tests and maintaining code quality? Can you share a specific approach or strategy?"

Response Window: 4 minutes

Question 8:
"Explain a design pattern you used recently in your project. Why did you choose this pattern, and how did it benefit the system?"

Response Window: 4 minutes

Question 9:
"How would you design a user-management system for a platform supporting features like sign-up, sign-in (via OTP), reset password, and forgot password? Briefly discuss the microservices and database schema you would use."

Response Window: 5 minutes

Question 10:
"What is your approach to defining and implementing CI/CD pipelines in your projects? Can you describe your experience with tools like Jenkins or GIT?"

Response Window: 3 minutes

Optional Follow-Up Questions (if time allows)

1. *"Can you share your perspective on performance KPIs and how to achieve them in a high-traffic application?"*
2. *"What is your understanding of observability? How have you implemented observability features in your applications?"*
3. *"How do you approach handling SONAR issues like bugs, vulnerabilities, and code smells? Have you ever updated a SONAR policy?"*
4. *"What is your view on audit logs, and how do you implement them to ensure compliance?"*
5. *"Can you explain the difference between `StringBuilder` and `StringBuffer`, and when would you use each?"*

>>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

During interview process for candidate-1 (Xxxx Kumar)

AI Agent Context Setting (Opening Statement)

"Good afternoon, Xxxx Kumar. I hope you're doing well today. Thank you for joining this interview session. I am VirtuaHire, your virtual interview panelist."

"At NPCI, we aim to hire talented Java professionals to work on large and complex projects for various clients in domains such as telecom, retail, healthcare, airlines, and education. These projects demand a high level of expertise in Java, architecture, and scalable design principles."

"I will guide you through the session, and we'll evaluate your skills based on your responses. Please feel free to ask for clarification if needed. Shall we start with the first question?"

[**AI PanelPro (AIP)**] receives the candidate-1's candidate's confirmation response.

Question 1:

"You've mentioned experience in microservices. Can you describe a recent project where you implemented microservices architecture? Specifically, what design principles and tools did you use, and what challenges did you encounter?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"In my recent project, we migrated a monolithic e-commerce platform to a microservices architecture. We employed domain-driven design (DDD) to decompose the system into bounded contexts, each representing a specific business domain. We leveraged Spring Boot and Spring Cloud for building and orchestrating the microservices. To ensure scalability and fault tolerance, we implemented circuit breakers, rate limiting, and asynchronous communication patterns using technologies like Kafka and RabbitMQ. One of the significant challenges we faced was managing data consistency across microservices. We addressed this by using event sourcing and CQRS patterns. Additionally, we employed a robust testing strategy, including unit, integration, and end-to-end tests, to maintain high code quality."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 2:

"In large systems, performance is a critical factor. Can you explain a scenario where you optimized the performance of your application? What strategies and tools did you use to achieve this?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"In a recent e-commerce application, we faced performance bottlenecks during peak traffic hours. To address this, we implemented a multi-pronged approach. First, we profiled the application using tools like JProfiler to identify performance hotspots. We optimized database queries by creating indexes, using caching strategies, and minimizing round trips. We also employed asynchronous processing using message queues to offload heavy tasks. To further enhance performance, we implemented load balancing and horizontal scaling using techniques like Kubernetes. We continuously monitored the application's performance using tools like Prometheus and Grafana to proactively identify and address issues."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 3:

"What is your understanding of API-first design, and how have you applied this principle in your previous projects?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"API-first design is a development approach where APIs are designed and developed before the implementation of the actual services. This approach ensures that the API contracts are well-defined and consistent, leading to better integration and collaboration between teams. In a recent project, we adopted an API-first approach using OpenAPI Specification (OAS). We defined the API contracts early in the development cycle, which allowed for early testing, validation, and documentation. This approach enabled us to develop the frontend and backend teams concurrently, accelerating development and improving overall project efficiency."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 4:

"How do you ensure security-first design in your applications? Can you provide an example of implementing security measures, such as authentication and data protection, in a project?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"Security is a paramount concern in modern software development. I employ a layered security approach, incorporating multiple defense mechanisms. This includes robust authentication and authorization mechanisms, such as OAuth2 and OpenID Connect, to protect sensitive data. I also implement secure coding practices, regularly review code for vulnerabilities, and utilize tools like SonarQube for static code analysis. In a recent project, we secured a web application

using OAuth2 for authorization. We also implemented HTTPS to encrypt communication, protected sensitive data using encryption techniques, and regularly conducted security audits to identify and mitigate potential threats."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 5:

"Could you explain how you handle reusable code in your projects? For instance, how do you design project directories or manage reusable components like database models or utility libraries?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"I leverage a modular approach to organize my projects, separating concerns into distinct modules. This promotes code reusability and maintainability. I employ design patterns like the Factory and Singleton patterns to create reusable components. For database models, I use an ORM framework like Hibernate to map them to database tables, making them reusable across different projects. I also utilize version control systems like Git to manage reusable code effectively. I create separate repositories for shared libraries and use dependency management tools like Maven or Gradle to incorporate them into projects. This approach ensures consistency and facilitates easy updates and maintenance."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 6:

"You've worked with both SQL and NoSQL databases. Can you compare them in the context of one of your projects and explain why you chose one over the other?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"In a recent e-commerce project, we evaluated both SQL and NoSQL databases to determine the best fit for our data storage needs. While SQL databases provide strong consistency and ACID compliance, they can be less flexible for handling unstructured data and high-write workloads. NoSQL databases, on the other hand, offer excellent scalability and flexibility but may compromise on strong consistency. We chose a hybrid approach, using a relational database (MySQL) for structured data like customer information and order history, and a NoSQL database (MongoDB) for unstructured data such as product reviews and user preferences. This combination allowed us to optimize performance, scalability, and data flexibility."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 7:

"As a technical leader, how do you ensure your team follows best practices for writing unit tests and maintaining code quality? Can you share a specific approach or strategy?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"As a technical leader, I emphasize the importance of test-driven development (TDD) and continuous integration/continuous delivery (CI/CD) pipelines. We establish clear coding standards and guidelines, using tools like SonarQube to analyze code quality and identify potential issues. Regular code reviews are conducted to ensure adherence to best practices and to foster knowledge sharing. To promote a culture of quality, we set specific code coverage targets and track them over time. We also encourage the use of mocking frameworks to isolate units of code and write effective unit tests. By combining these strategies, we maintain high code quality and ensure the reliability of our applications."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 8:

"Explain a design pattern you used recently in your project. Why did you choose this pattern, and how did it benefit the system?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"In a recent microservices project, we employed the Observer pattern to implement a real-time notification system. This pattern allowed us to decouple the notification logic from the core business logic, making the system more flexible and scalable. When a specific event occurs, the subject (the event source) notifies all registered observers (the notification services), ensuring that they receive updates in real-time. This pattern improved the system's responsiveness and reduced coupling between components."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 9:

"How would you design a user-management system for a platform supporting features like sign-up, sign-in (via OTP), reset password, and forgot password? Briefly discuss the microservices and database schema you would use."

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"I would design a user-management microservice responsible for handling user-related operations. The microservice would interact with a dedicated user database, storing essential information like user ID, email, phone number, hashed password, and security tokens. Key Microservices: User Management Microservice: Handles user registration, authentication, authorization, password reset, and profile management. Database Schema: users table: user_id (PK), email, phone_number, hashed_password, salt, created_at, updated_at

password_reset_tokens table: token (PK), user_id (FK), expiration_time, created_at
otp_verification table: otp (PK), user_id (FK), expiration_time, created_at I would implement robust security measures, including strong password hashing, secure token generation, and rate limiting to protect against attacks like brute-force and DDoS."

[**AI PanelPro (AIP)**] receives the candidate-1's response, assesses internally and asks the next question.

Question 10:

"What is your approach to defining and implementing CI/CD pipelines in your projects? Can you describe your experience with tools like Jenkins or GIT?"

Candidate-1 said: *(Candidate-1's response provided in the next prompt.)*
"I advocate for a robust CI/CD pipeline that automates the build, test, and deployment processes. I utilize tools like Jenkins or GitLab CI/CD to orchestrate the pipeline. I begin by defining clear build stages, such as code compilation, unit testing, integration testing, and deployment. I configure the pipeline to trigger automatically on code commits or pull requests. To ensure code quality, I integrate static code analysis tools like SonarQube and security scanning tools into the pipeline. I also implement automated testing strategies, including unit, integration, and end-to-end tests. By automating these processes, we can accelerate delivery, reduce manual errors, and improve overall software quality."

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<

Post-interview process for candidate-1 (Xxxx Kumar)

Consolidated Feedback Summary for Candidate-1 (Xxxx Kumar)

Question	Score	Highlights	Improvement Areas
1	9/10	Strong knowledge of microservices, tools, and design principles.	Could expand on monitoring tools and measurable results.
2	8.5/10	Effective performance optimization with caching, profiling, and scaling.	Include performance metrics and challenges faced.
3	8/10	Clear understanding of API-first design and collaboration benefits.	Add challenges faced during API-first adoption.
4	8.5/10	Comprehensive security-first approach with real-world examples.	Discuss advanced security features like MFA.
5	8.5/10	Modular design and effective use of tools for reusability.	Provide real-world examples of reusable components.
6	9/10	Clear SQL vs. NoSQL comparison with a hybrid approach.	Expand on challenges and advanced database optimization techniques.
7	9/10	Strong leadership in quality practices like TDD, code reviews, and SonarQube.	Add real-world examples of resolving specific quality issues.
8	8.5/10	Practical use of the Observer pattern in microservices.	Address challenges and advanced use cases of the pattern.
9	9/10	Well-designed user-management system with security considerations.	Add advanced features like MFA or OAuth integration.
10	9/10	Comprehensive CI/CD pipeline with quality and security measures.	Include rollback strategies and real-world metrics.

>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<

Interview process for next candidate- (2 .. n)

Point to note that here the interview process is not required to be sequential like traditional human panel based interviews.

Some other n number of candidates have self scheduled interview, joining on [**AI PanelPro (AIP)**] platform, answering questions being asked by [**AI PanelPro (AIP)**], hiring team getting feedback for each candidate automatically.

In my rehearsal, I performed the simulation with 2 more candidates, the report based on 3 candidates was extrapolated for 30 more candidates.

>>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<<<

Consolidated Feedback Summary for Candidate-2 (Vxxx Raj)

Question	Score	Highlights	Improvement Areas
1	9/10	Clear understanding of microservices architecture with a focus on security.	Could expand on challenges and monitoring in microservices.
2	8.5/10	Practical performance optimization techniques, including database indexing and Redis.	Include real-world performance metrics and scalability considerations.
3	8/10	Solid understanding of API-first design with Swagger.	Could elaborate on challenges and provide advanced use cases.
4	8.5/10	Strong security-first approach with OAuth2, JWT, and encryption.	Could address challenges and discuss advanced security features.
5	8.5/10	Well-structured reusable code strategy, using SOLID principles and modular design.	More real-world examples and challenges in versioning reusable code.
6	9/10	Excellent understanding of choosing the right database (InfluxDB for time-series).	Expand on SQL limitations and challenges faced with NoSQL implementation.
7	9/10	Strong leadership skills in fostering code quality, with CI/CD integration.	Could discuss challenges and provide real-world metrics on quality improvements.
8	9/10	Effective use of the Strategy pattern for payment processing.	Could elaborate on challenges and other potential use cases of the pattern.
9	9/10	Well-designed user-management system with strong security practices.	Could discuss challenges in microservice integration or advanced security features.

10	9/10	Comprehensive CI/CD pipeline design with testing and security integration.	Include rollback strategies and provide measurable results.

Consolidated Feedback Summary for Candidate-3 (Pxxx Pha)

Question	Score	Highlights	Improvement Areas
1	6.5/10	Basic understanding of microservices and recognition of challenges (data consistency).	Needs more detail on design principles, tools, and resolving challenges.
2	6/10	Identified key performance issues and basic optimizations (caching, database queries).	Lack of depth on specific strategies, tools, and measurable results.
3	6/10	Basic understanding of API-first design.	Lacks details on tools, challenges, and real-world application.
4	5/10	Awareness of security practices but lacks specific details on implementation.	Needs concrete examples of security measures (e.g., JWT, OAuth2).
5	4/10	Hybrid database approach for structured vs unstructured data.	Did not address reusable code, project directory design, or patterns.
6	6/10	Good understanding of hybrid databases, with SQL vs NoSQL comparison.	Lacks details on the challenges and tools used for integration.
7	6.5/10	Leadership in promoting quality through CI/CD, code reviews, and automation.	Lacks specific practices, tools, and handling challenges in quality.
8	7/10	Basic use of the Singleton pattern with good application in managing resources.	Needs more detail on challenges and alternative patterns considered.
9	6.5/10	Basic design of user-management system with focus on security.	Lacks database schema details, microservices breakdown, and advanced features.
10	6.5/10	Solid understanding of CI/CD and automation using Jenkins/GitLab CI.	Lacks specific tools, failure handling, and metrics for improvement.

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<

Summary Table: Sample Feedback for 33 Candidates

Candidate	Gender	State	Q1: Microservices	Q2: Performance Optimization	Q3: API-First Design	Q4: Security Design	Q5: Reusable Code	Q6: SQL vs NoSQL	Q7: Code Quality	Q8: Design Pattern	Q9: User Management	Q10: CI/CD	Avg. Score
Amit Kumar (C1)	Male	Bihar	9/10	8.5/10	8/10	8.5/10	8.5/10	9/10	9/10	8.5/10	9/10	9/10	8.7/10
Shiv Raj (C2)	Male	Jharkhand	9/10	8.5/10	8/10	8.5/10	8.5/10	9/10	9/10	9/10	9/10	9/10	8.8/10
Arjit Da (C3)	Male	Uttar Pradesh	6.5/10	6/10	6/10	5/10	4/10	6/10	6.5/10	7/10	6.5/10	6.5/10	6.5/10
Priya Sharma	Female	Maharashtra	8.5/10	7.5/10	8/10	7.5/10	8/10	8.5/10	8/10	8/10	8.5/10	8.5/10	8.0/10
Saba Khan	Female	Delhi	8/10	8/10	7.5/10	8/10	7.5/10	8/10	8/10	8.5/10	8.5/10	8.5/10	8.1/10
Sameer Sheikh	Male	Gujarat	7.5/10	8/10	7/10	7.5/10	7/10	8/10	7.5/10	8/10	8/10	8/10	7.7/10
Lata Joshi	Female	Rajasthan	8.5/10	7.5/10	8/10	8/10	8/10	8/10	8/10	8.5/10	8.5/10	8.5/10	8.1/10
Amandeep Kaur	Female	Punjab	9/10	8.5/10	8/10	8.5/10	8.5/10	9/10	8.5/10	9/10	9/10	8.5/10	8.6/10
Arjun Iyer	Male	Tamil Nadu	9.5/10	9/10	8.5/10	9/10	9/10	9/10	9/10	9.5/10	9.5/10	9.5/10	9.2/10
Maria D'Souza	Female	Goa	8.5/10	8/10	8/10	8.5/10	8/10	8.5/10	8/10	8.5/10	8.5/10	8.5/10	8.2/10
Vikrant Das	Male	West Bengal	8/10	7.5/10	7.5/10	7.5/10	7.5/10	8/10	8/10	7.5/10	8/10	7.5/10	7.7/10
Sonia Pillai	Female	Kerala	8/10	8/10	7.5/10	8/10	8/10	8.5/10	8/10	8/10	8.5/10	8/10	8.0/10
Rashmi Verma	Female	Bihar	8/10	7.5/10	8/10	8/10	7.5/10	8/10	8/10	8/10	8.5/10	8/10	7.9/10
Zoya Ali	Female	Karnataka	8.5/10	8/10	8/10	8.5/10	8.5/10	9/10	8.5/10	8.5/10	8.5/10	8.5/10	8.4/10

>>>>>>>>>>>>>>>>>>>>>>>>> <<<<<<<<<<<<<<<<<<<<<<<<

Conclusion

Behind the scene, the architecture of [**AI PanelPro (AIP)**], I plan to share in the near future. I foresee a great potential in [**AI PanelPro (AIP)**] in SaaS model, which is not just limited to software or IT industry, rather it covers all the industry segments where assessment is based on [question-answer] method and answer to one question is more-less unified until its context is changed.

About author

Profile : Rajesh Verma - Brief profile

Source : link for this article here

Series : S2 (Artificial Intelligence)

Episode : WHITEPAPER : [S2-AIGAI] How I simulated AI for the interview panelist

Author's approach : Rajesh wants to share his learning & experience gained throughout his career from various sources. Author started the series on architectural topics including AI/ML & GAI topics and this article is one of the episodes in that attempt. Author feels that lots of information is available on various forums, but scattered here & there. Episodes in this series will be designed for most of the relevant topics in architecture-&-design, published gradually and organized in logical sequence. Principally episodes will have linkage with other episodes, so that readers can have proper connection among the topics and would be able to correlate with ongoing activities in their software life. Topics for example will be related to functional architecture, integration architecture, deployment architecture, microscopic view of mostly architecture-building-blocks (ABBs), security guidelines & approach to comply, performance KPIs & engineering, git branch & DevOps enabled automation strategy, NFR aspects (e.g. scalability, high-availability, stability, resiliency, etc.), commonly used architecture styles & design patterns, cloudification approaches, multi-tenancy approach, data migration, channel-cutover & rollout strategy, process standardization & simplification, greenfield rollout & brownfield transformation journeys, etc.

Thank you for reading the post, please stay connected.

Design Prompt : [AI PanelPro] is an AI agent for an interview panel who will perform an interview process on-behalf of a human panel for all industry segments. In the design please show the interviewer side with one human & AI agent and the other side of the candidate with one human.