

AI Interviewer Agent – Technical Report

1. System Architecture

The AI Interviewer is a modular, state-driven interview simulation platform designed to conduct realistic multi-round interviews with adaptive difficulty, personality-based behavior, multimodal inputs, and structured evaluation.

1.1 High-Level Components

Frontend (Streamlit UI)

- Collects candidate details and resume
- Displays interview chat interface.
- Provides whiteboard interface for technical rounds
- Shows post-interview analytics, feedback, and learning plan
- Allows downloadable DOCX report generation

Interview Orchestration Layer (LangGraph)

- Manages interview flow using a finite-state graph
- Controls transitions between HR → Technical → Behavioral → Closing
- Enforces hard gates (e.g., technical round failure stops interview)
- Maintains strict single-pass finalization to prevent duplicate summaries

LLM Layer (Groq + Vision Models)

- Text reasoning via Groq LLaMA-3.1 models
- Vision reasoning for whiteboard analysis using OpenRouter / LLaMA vision
- Separate prompts for intent detection, answer judgment, reactions, hints, and final evaluation

Evaluation & Analytics Engine

- Multi-dimension scoring (technical depth, clarity, confidence, communication, correctness)
- Hint penalties and whiteboard bonuses
- Topic-level performance breakdown
- Round-wise and final verdict computation

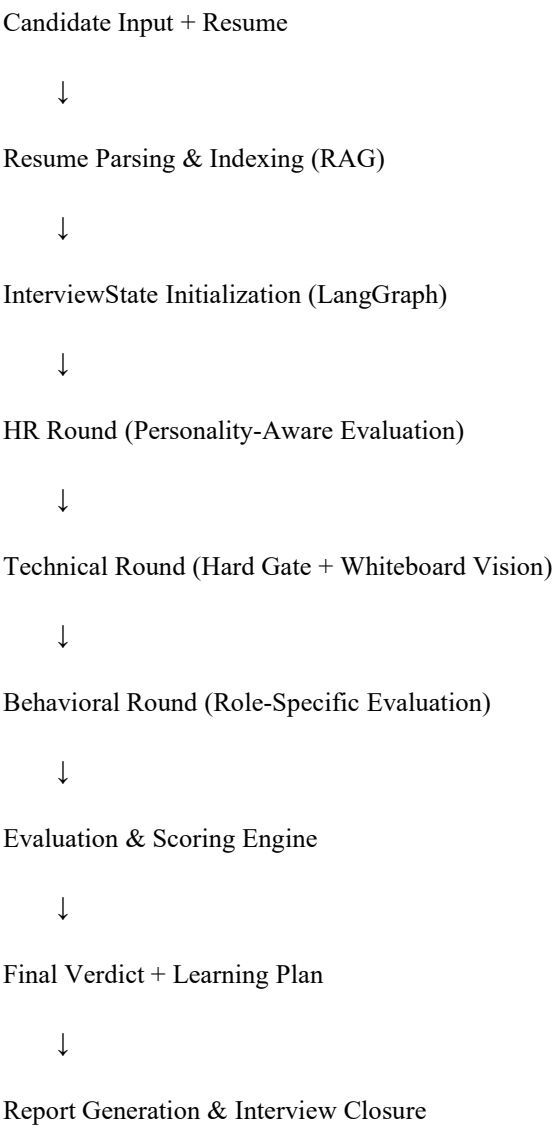
RAG Layer (ChromaDB)

- Resume is indexed into a vector database
- Resume context is retrieved during question generation
- Ensures interview questions are personalized and grounded

Learning Plan Generator

- Converts interview weaknesses into a personalized roadmap
 - Suggests practice tasks and curated learning resources
 - Adapts recommendations based on interviewer personality
-

2.End-to-End Interview Pipeline Flow



2.1 Candidate Input + Resume

The interview begins when the candidate enters basic details and uploads a resume through the Streamlit interface. The candidate also selects the interviewer personality and role. This information is captured as the initial input and stored for use throughout the interview lifecycle.

2.2 Resume Parsing & Indexing (RAG)

The uploaded resume is parsed to extract text from supported formats such as PDF, DOCX, or TXT. The extracted text is cleaned, chunked, and converted into vector embeddings. These embeddings are stored in a vector database and later retrieved during question generation to ensure that interview questions are personalized and grounded in the candidate's background.

2.3 InterviewState Initialization (LangGraph Control)

After resume indexing, the system initializes a structured InterviewState object. This object stores all runtime information including candidate profile, selected interviewer personality, current round, difficulty level, answer history, hint usage, and scoring metadata. A LangGraph state machine uses this state to control interview flow and enforce strict transition rules.

2.4 HR Round (Personality-Aware Evaluation)

The interview starts with the HR round. Personality-aware HR questions are generated to assess communication skills, clarity of expression, and confidence. Candidate responses are evaluated, and limited retries or follow-up questions allowed for weak answers. Completion of this round triggers an automatic transition to the technical stage.

2.5 Technical Round (Hard Gate + Whiteboard Vision)

The technical round acts as a hard gate for the interview. Resume-aware technical questions are generated using retrieved resume context, interviewer personality, and adaptive difficulty logic. Candidate answers are evaluated for correctness and technical depth. Poor performance beyond a defined threshold immediately terminates the interview.

During this round, candidates may optionally upload whiteboard diagrams. These diagrams are analyzed by a vision-language model, which provides concise structural feedback and generates exactly one follow-up technical question.

2.6 Behavioral Round (Role-Specific Evaluation)

Candidates who pass the technical round proceed to the behavioral round. This stage focuses on role-specific behavioral questions designed to evaluate real-world reasoning, decision-making, and ownership.

2.7 Evaluation & Scoring Engine (Multi-Dimension)

After the interview rounds are completed or terminated early, all recorded answers are aggregated. The system computes scores across multiple dimensions including technical depth, correctness, clarity, confidence, and communication. Hint penalties, repetition checks, and whiteboard bonuses are applied. Round-wise and overall scores are then calculated.

2.8 Final Verdict + Learning Plan

Based on aggregated scores and interview rules, the system determines the final verdict (Selected or Not Selected). In parallel, weak topics are identified and converted into a personalized learning plan consisting of recommended practice tasks and curated learning resources.

2.9 Report Generation & Interview Closure

Finally, a structured interview report is generated containing the interview summary, scores, strengths, improvement areas, final evaluation, and learning plan. The report can be downloaded in DOCX format. The interview is then politely closed, completing the interview process.

3. Interview Flow and Agent Logic

3.1 Interview Rounds

The interview always follows **three fixed rounds**:

1. HR Round

- Personality-aware situational questions
- Controlled retries and follow-ups
- Evaluates communication, clarity, and confidence

2. Technical Round

- Core gatekeeper round
- Adaptive difficulty (levels 1–5)
- Resume-based, system design, coding, and theory questions
- Whiteboard interaction allowed
- Failure immediately ends interview

3. Behavioral Round

- Role-specific behavioral questions
- Enforced minimum and maximum question counts
- Evaluates decision-making, ownership, and reasoning

3.2 State-Driven Control (LangGraph)

The interview is implemented as a LangGraph state machine:

- `InterviewState` carries all runtime data
- Each user input triggers exactly one graph step
- Hard guards prevent:
 - Duplicate final evaluations
 - Continuing after technical failure
 - Skipping mandatory rounds

3.3 Personality-Based Behavior

The interviewer behavior changes dynamically based on selected personality:

Personality	Behavior
Friendly Coach	Supportive, learning-oriented
Professor	Theory-heavy, formal correctness
Startup Hiring Manager	Practical, real-world trade-offs
Strict FAANG	High difficulty, zero tolerance

Personality affects:

- Question difficulty escalation
- Hint limits
- Strictness of evaluation
- Final verdict logic

3.4 Logical Agents in the AI Interviewer System

The AI Interviewer system is implemented using multiple logical agents. Each agent is responsible for a specific function in the interview lifecycle and operates over a shared `InterviewState`. The agents are coordinated using a LangGraph state machine to ensure controlled execution, fairness, and explainability.

3.4.1 Interview Orchestrator Agent

- Controls round transitions (HR → Technical → Behavioral)
- Enforces hard technical gate
- Prevents duplicate final evaluations
- Terminates interview when failure conditions are met

3.4.2 Question Generation Agent

- Generates HR, technical, and behavioral questions
- Uses resume context (RAG), role, personality, and difficulty
- Produces follow-up questions for weak answers

3.4.3 Evaluation and Scoring Agent

- Evaluates answers across multiple dimensions
- Applies hint penalties and weak-answer limits
- Computes round-wise and final scores
- Determines final verdict

3.4.4 Intent Detection and Repetition Control Agent

- Detects user intent (answer, hint, unsure, clarification)
- Blocks repeated or reused answers
- Limits excessive hint usage

3.4.5 Whiteboard Vision Agent

- Analyzes uploaded diagrams during technical round
- Provides concise structural feedback
- Generates exactly one follow-up technical question
- Applies capped whiteboard bonus

3.4.6 Learning Plan Generation Agent

- Identifies weak topics after evaluation
- Generates personalized practice tasks
- Suggests curated learning resources

4. Use of Multimodal Inputs (Whiteboard)

4.1 Whiteboard Integration

The system supports diagram-based answers during the Technical round:

- Candidate opens an external drawing canvas
- Exports diagram as PNG
- Uploads the image into the interview

4.2 Vision-Based Analysis

The whiteboard image is analyzed using a vision-language model:

- Identifies visible elements (boxes, arrows, flows)
- Detects missing steps, risks, or design flaws
- Produces concise diagram-specific feedback
- Generates exactly one follow-up technical question

If vision analysis fails, the system safely falls back to text-only reasoning using the candidate's explanation.

5. Memory and State Management Strategy

5.1 InterviewState (Single Source of Truth)

All interview data is stored inside a structured state object:

- Candidate profile and resume
- Current round and question index
- Answer records
- Hint usage and retry counters
- Difficulty level and streak tracking
- Final verdict and evaluation

5.2 Answer Records

Each answer is stored as an `AnswerRecord` containing:

- Question text
- Candidate answer
- Topic and question type
- Quality label (good / weak)
- Hints used
- Round name

- Whiteboard usage metadata

These records drive:

- Final scoring
- Topic breakdown
- Learning plan generation
- Downloadable report content

5.3 Repetition and Concept Memory

The system actively prevents:

- Reusing the same answer across different questions
- Over-helping using repeated hints

It also supports one-time concept reminders:

- If a candidate previously explained a concept well
 - The agent encourages applying the same idea to a new scenario
-

6. Design Decisions and Known Limitations

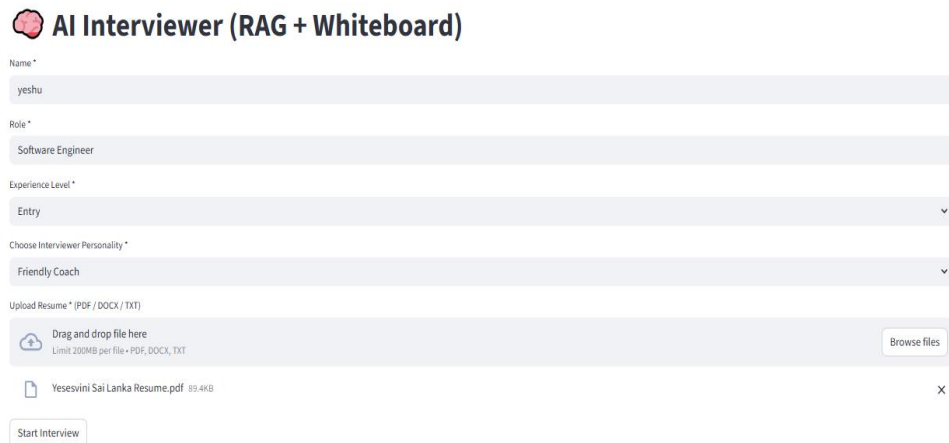
6.1 Key Design Decisions

- **Hard Technical Gate:** Technical failure immediately stops the interview
- **Strict Finalization Guard:** Prevents duplicate summaries or verdicts
- **No Silent Fallbacks:** All LLM failures are explicit and controlled
- **Explainability First:** Every score is traceable to recorded answers
- **Safety-First State Machine:** No undefined transitions or loops

6.2 Known Limitations

- Vision models depend on external APIs and may face latency
- Diagram understanding is limited to visible structural cues
- Learning resources rely on curated mappings when LLM fails
- Interview realism is partially dependent on the reasoning quality and consistency of the underlying language models.
- Offline usage is not supported due to model dependencies

7. Example interview transcripts



The screenshot shows the 'AI Interviewer (RAG + Whiteboard)' interface. It features a brain icon and the title. Below the title are several input fields: 'Name' with the value 'yeshu', 'Role' with 'Software Engineer', 'Experience Level' with a dropdown menu showing 'Entry', and 'Choose Interviewer Personality' with a dropdown menu showing 'Friendly Coach'. There is an 'Upload Resume' section with a file upload area showing 'Drag and drop file here' and a 'Browse files' button. A file named 'Yeesivini Sai Lanka Resume.pdf' (89.4KB) is listed below. At the bottom is a 'Start Interview' button.

Fig:7.1 Interview initialization screen used to collect candidate details, configure interviewer personality, and ingest resume for RAG-based question generation.

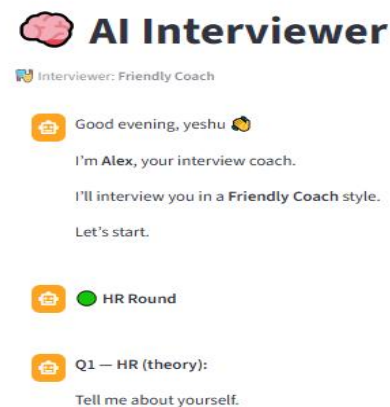


Fig:7.2 Interview start screen showing a time-based personalized greeting, interviewer introduction, selected interviewer personality, and the beginning of the HR round with the first interview question.

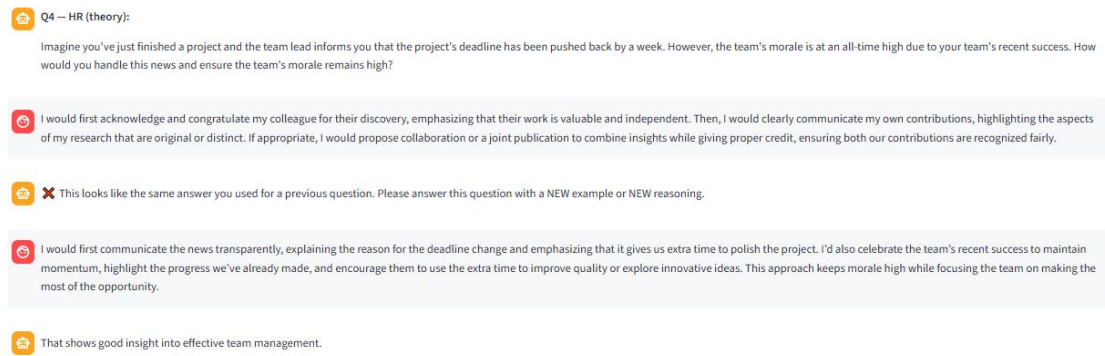


Fig:7.3: Screen showing detection of a repeated response, a request for a revised answer, evaluation of the improved response, and positive feedback.

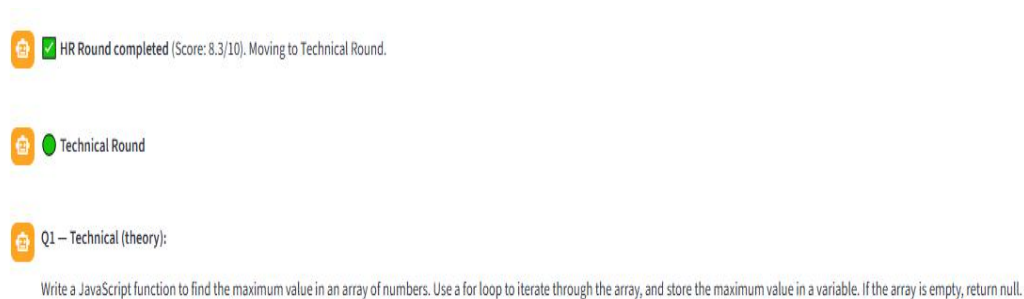


Fig:7.4 Screen illustrating the completion of the HR round with score feedback, the automatic transition to the technical round, and the presentation of the first technical interview question.



Fig:7.5 Technical question screen displaying an “Open Whiteboard” button, allowing candidates to visually explain concepts using diagrams.

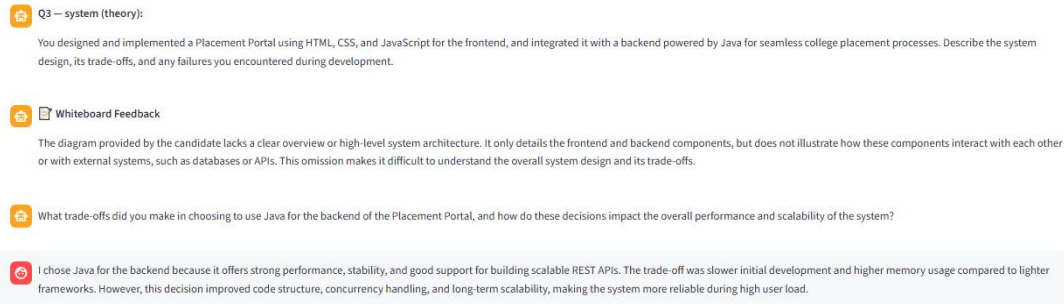


Fig:7.6 Technical round screen illustrating a system design question, automated whiteboard feedback, vision-based follow-up questioning, and candidate response evaluation.

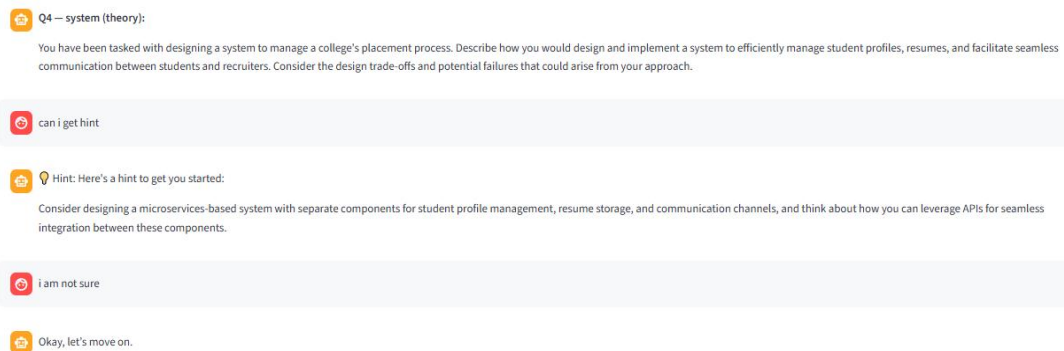


Fig:7.7 Technical round screen showing how the system gives a hint when the candidate asks for help, detects when the candidate is unsure, and automatically moves to the next question.

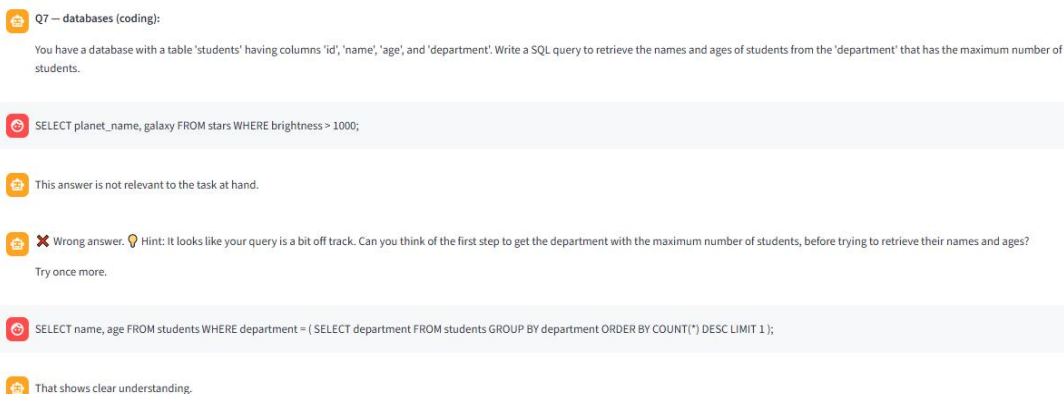


Fig:7.8 Technical coding question screen illustrating incorrect answer detection, guided hint feedback, a second attempt, and confirmation of the correct SQL solution.

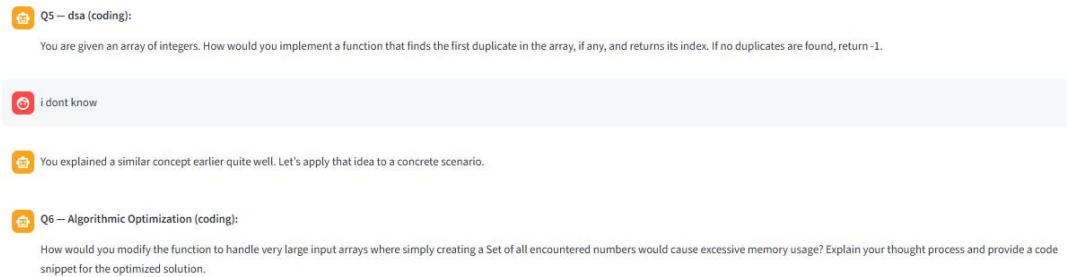


Fig:7.9 Screen showing that the system detects when the user is unsure, gives a one-time reminder of a previously explained idea, and then moves to a follow-up question about improving the solution.

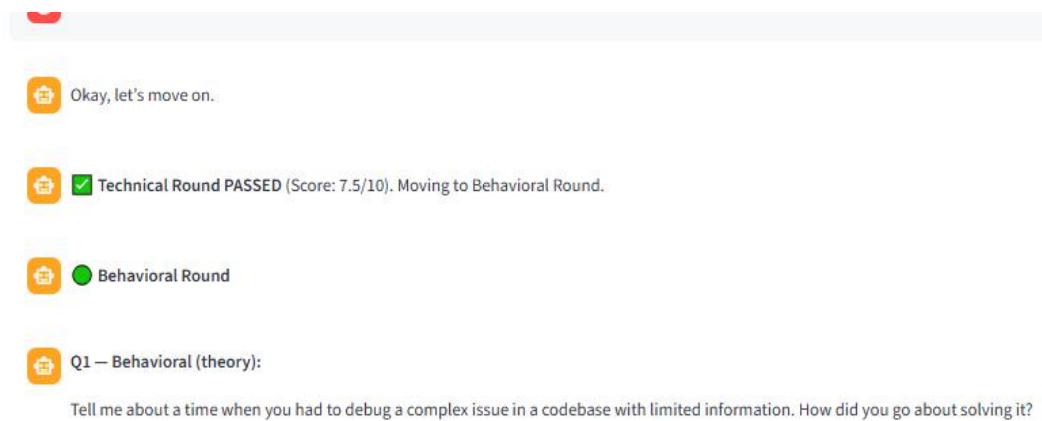


Fig:7.10 Screen showing technical assessment completion with score feedback, automatic transition to the next evaluation stage, and presentation of the first behavioral question.

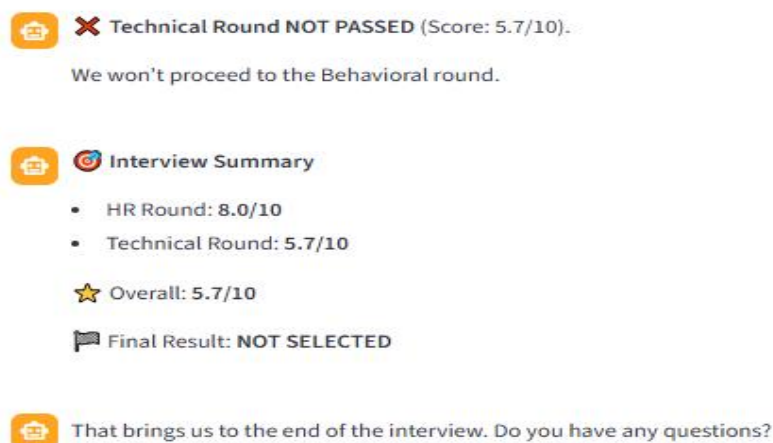


Fig:7.11 Screen showing interview termination after technical evaluation, score summary, and final non-selection outcome.

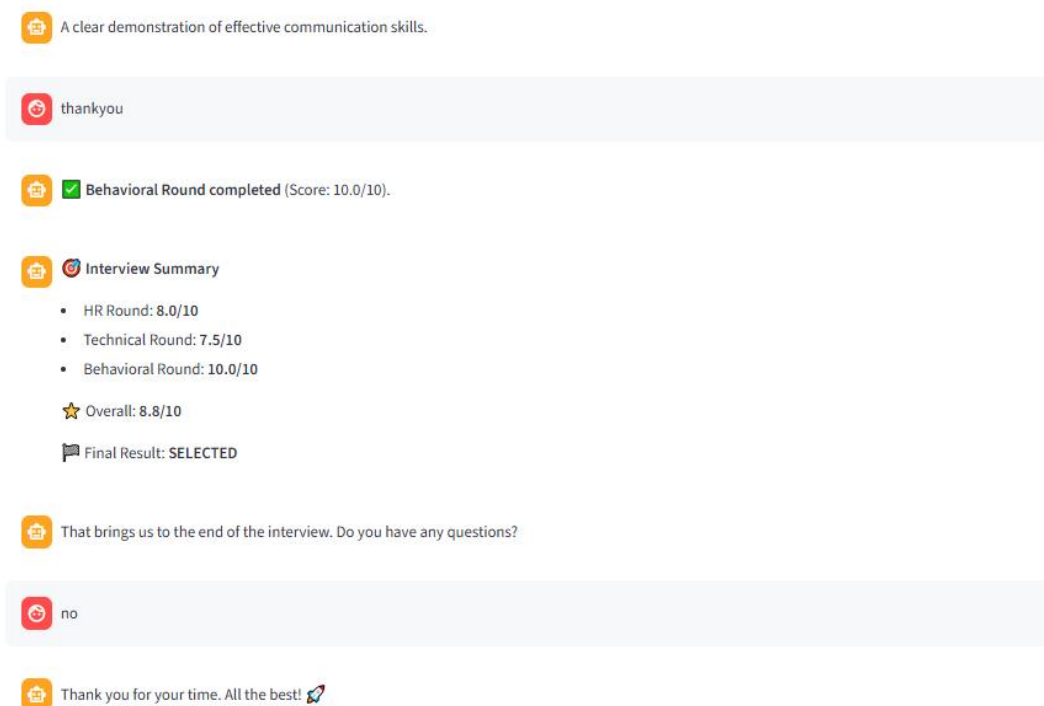


Fig:7.12 Final interview screen showing round-wise score summary, overall evaluation, selection decision, and polite interview closure.

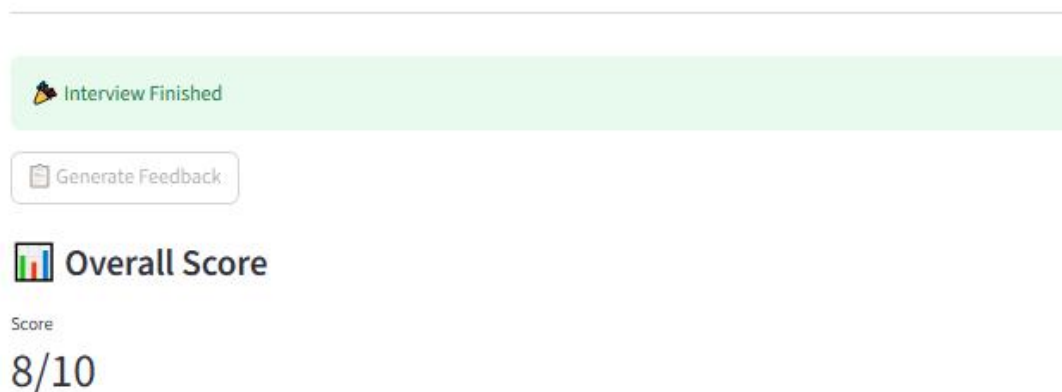
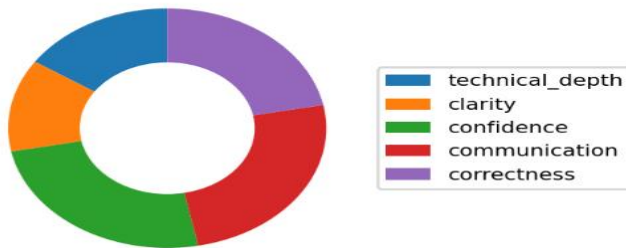


Fig: 7.13 Screen showing interview completion status and the overall score of the candidate.

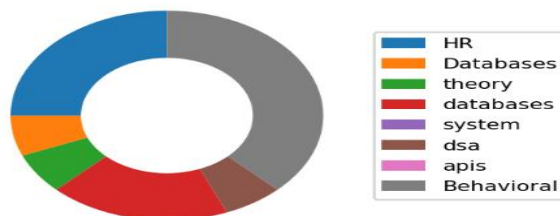
Skill Breakdown

Skill Dimensions



Topic Performance

Topics



Strengths

Technical Depth: strong performance

Confidence: strong performance

Communication: strong performance

Correctness: strong performance

Improvements

Summary

Overall performance score: 8/10.

Strong areas: Technical Depth: strong performance, Confidence: strong performance, Communication: strong performance, Correctness: strong performance.

Hints used during interview: 1

This learning plan is based on your overall interview performance (8/10), considering all rounds and observed weaknesses.

Learning Roadmap

- Practice system design questions for a e-commerce platform with 10M+ users
- Design a RESTful API for a social media platform supporting user profile management
- Write a simple API gateway using Node.js and Express.js
- Read and implement 5 different sorting algorithms in Python
- Design a database schema for a travel booking website
- Practice structured explanations using problem -> approach -> solution format.
- Practice advanced problems, edge cases, and optimizations.
- Focus on theoretical correctness and formal explanations.

Recommended Resources

System Design

System Design Primer: <https://github.com/doornemartin/system-design-primer>

Grokking System Design Interview: <https://www.designgurus.io/>

Apis

REST API Design: <https://restfulapi.net/>

FastAPI Documentation: <https://fastapi.tiangolo.com/>

Fig:7.14 Post-interview evaluation dashboard showing skill breakdown, topic performance, strengths, summary, personalized learning roadmap, and recommended resources.



Fig:7.15 Screen showing the option to download the interview report in DOCX format.

Conclusion

This project presents an AI-based interview system that conducts structured HR, Technical, and Behavioral interviews using resume-based personalization and adaptive evaluation. The system provides transparent scoring, identifies strengths and weaknesses, and generates a personalized learning roadmap with recommended resources. By combining agent-driven orchestration, multimodal whiteboard analysis, and explainable feedback, the platform delivers a realistic and effective interview assessment and preparation experience.