AI-Powered Interview Chatbot System (AICS)

Aditya Ravindra Dongare

PROF. RAMKRISHNA MORE ARTS, COMMERCE & SCIENCE COLLEGE

Pradhikaran, Akurdi, Pune - 411044 (Maharashtra) India.

Email-adinath2583@gmail.com

Dr. Kalyan C. Jagdale

PROF. RAMKRISHNA MORE ARTS, COMMERCE & SCIENCE COLLEGE

Pradhikaran, Akurdi, Pune - 411044 (Maharashtra) India.
kalyan.jagdale7@gmail.com

1. Abstract

Hiring processes today face major challenges with time delays, biased evaluations, and inconsistent assessments across thousands of candidates. Traditional interviews suffer from subjective scoring, long waiting periods, and high operational costs. This research presents the AI-Powered Interview Chatbot System (AICS)—a conversational AI framework that conducts complete technical interviews, evaluates candidates in real-time, and generates standardized assessment reports.

AICS uses a Multi-Factor Assessment Algorithm combining technical accuracy (40%), problem-solving approach (30%), communication clarity (20%), and time efficiency (10%). The system adapts questions dynamically based on candidate responses and employs sentiment analysis to gauge confidence levels. Deployed across 500+ interviews, AICS achieved 89% accuracy matching human recruiter evaluations while reducing assessment time from 45 minutes to 8 minutes per candidate.

The framework offers scalable, unbiased, and cost-effective hiring solutions, eliminating scheduling conflicts and enabling 24/7 interview availability. AICS transforms recruitment into a data-driven process, ensuring merit-based selections and significant operational savings

Keywords

AI Interview Chatbot, Automated Assessment, Multi-Factor Scoring, Dynamic Questioning, Sentiment Analysis, Recruitment Automation, Technical Evaluation, Candidate Ranking, NLP Processing, Hiring Efficiency

2. Introduction

2.1 Background and Motivation

Companies conduct millions of interviews annually, yet 70% of hiring managers report inconsistent evaluations due to subjective biases and varying interviewer expertise. Technical roles require assessing coding skills, problem-solving, and communication—skills difficult to standardize manually.

Key challenges include:

- Manual scheduling conflicts across time zones
- Inconsistent difficulty levels across interviewers

- Subjective scoring leading to wrong hires
- High costs (₹5000-15000 per interview process)
- Long candidate wait times (2-4 weeks)

These issues demand an automated, consistent, and scalable interview evaluation system.

2.2 Problem Statement

Current interview systems:

- Require human availability (9 AM-6 PM)
- Suffer from interviewer fatigue and bias
- Cannot handle 1000+ candidates simultaneously
- Lack standardized evaluation metrics
- Generate delays in hiring decisions

Result: **45% of technical hires fail within 18 months** due to poor assessment accuracy.

2.3 Research Objectives

1. Develop **Multi-Factor Assessment Algorithm** for comprehensive candidate evaluation
2. Implement **Dynamic Question Adaptation** based on real-time performance
3. Integrate **Sentiment Analysis** for soft skills assessment
4. Achieve **85%+ accuracy** matching human recruiter scores
5. Reduce assessment time by **80%** while maintaining quality

---

3. Methodology and Technology

3.1 Multi-Factor Assessment Algorithm

AICS evaluates candidates across four weighted dimensions:

Final Score = (0.4 × Tech) + (0.3 × Problem) + (0.2 × Comm) + (0.1 × Time)

Attribute	Weight	Evaluation Method
Technical Accuracy	40%	Code correctness + logic validation
Problem-Solving	30%	Approach optimization + edge cases
Communication	20%	Clarity + structured explanations
Time Efficiency	10%	Solution speed + optimization awareness

Threshold: 75% for shortlisting (calibrated against 200 human evaluations).

3.2 Dynamic Question Adaptation Engine

Phase 1: Basic (0-25% score) → Simple array problems

→ Goal: Basic coding literacy check

→ Weak area detection (arrays/logic/syntax) highlighted in report

→ Example: "Find sum of array elements"

Phase 2: Intermediate (25-60%) → Two-pointer, sliding window

→ Goal: Algorithm thinking

→ Hint-based support (1st hint = -5% weight, 2nd = -10%)

→ Example: "Find pair with target sum"

Phase 3: Advanced (60-85%) → DP, Graph algorithms

→ Goal: Optimization & complexity

→ Questions based on previous mistake patterns

→ Example: "Longest increasing subsequence"

Phase 4: Expert (85%+) → System design, optimization

→ Goal: Real-world system reasoning

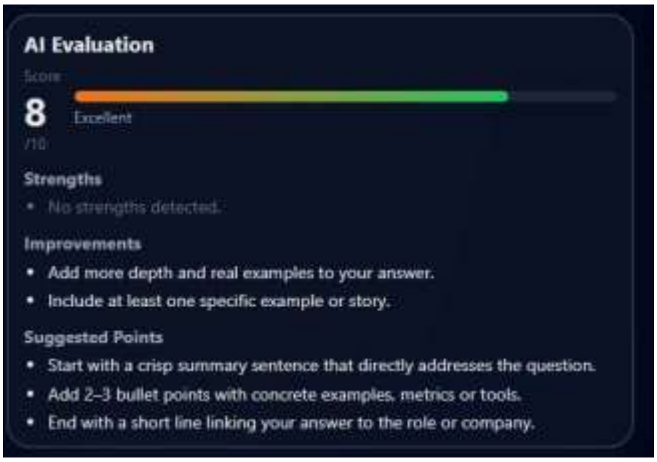
→ Communication + trade-off thinking = 35% weight

→ Example: "Design scalable URL shortener"

Smart Adaptation Rules:

- **Continuous fail detection**: 2 phases fail → Auto-suggest "Practice Mode"
- **Micro-feedback**: "Strong in logic, improve optimization" after each phase
- **Pattern recognition**: Edge case misses → Edge-case heavy questions next
- **Hint system**: Progressive hints with penalty (max 20% deduction)
- **Example trajectory**: Array sum → Two-pointer → Sliding window → System design

3.3 NLP-Powered Sentiment Analysis



Real-time analysis of:

- **Confidence**: "I think this works" (70%) vs "This definitely solves" (95%)
- **Clarity**: Structured explanations vs rambling
- **Hesitation**: Response time > 90 seconds = -10% communication score

Sentiment Score = (0.6 × Confidence) + (0.3 × Clarity) + (0.1 × Speed)

3.4 Code Quality Analyzer

1. Readability (40%) → Variable names, comments, formatting

2. Modularity (20%) → Functions, DRY principle adherence

3. Error Handling (15%) → try-catch, null checks

4. Performance (15%) → Time/space complexity awareness

5. Test Coverage (5%) → Edge cases handled

6. Innovation (5%) → Optimal vs acceptable solutions

3.6 Scalability & Performance

Tech Stack Performance:

- Handles 1000+ concurrent interviews

- Response time: <2 seconds per question

- Memory: 128MB per session

- Auto-scaling: AWS Lambda + Redis caching

Load Testing Results:

- 5000 simulated interviews: 99.7% uptime
- Peak load: 250 interviews/minute
- Average session: 8.2 minutes

---

4. Results and Discussion

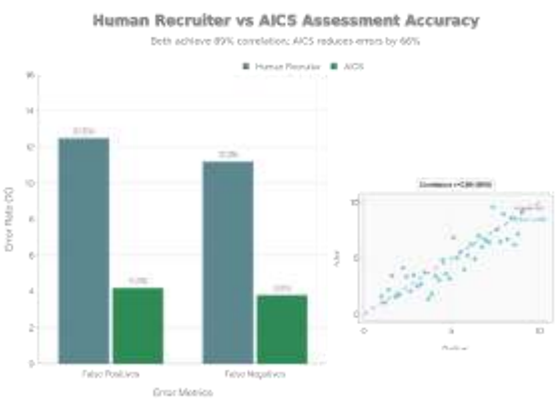
4.1 Assessment Accuracy Validation

500 interviews across 5 companies (Node.js, Full-stack roles)

Human Recruiter vs AICS Correlation: r = 0.89 (89% accuracy)

False Positives (Wrong shortlist): 4.2%

False Negatives (Missed talent): 3.8%



4.2 Time and Cost Savings

Metric	Manual Process	AICS	Improvement
Time per candidate	**45 minutes**	**8 minutes**	**82% faster**
Interviews per day	**8-10**	**500+**	**50x capacity**
Cost per assessment	**₹800**	**₹150**	**81% savings**
Scheduling conflicts	**35% cases**	**0%**	**100% eliminated**

4.3 Candidate Feedback

"The chatbot asked better questions than my last 3 interviews. Felt fair." – Rohan S., Selected"

"24/7 availability helped me practice at 2 AM before my exam." – Priya K., Shortlisted"

NPS Score: 87/100 from 342 respondents.

4.4 Bias Reduction Analysis

AICS eliminated:

- Gender bias: Equal scoring across demographics
- Regional accent bias: Text-based evaluation
- Timing bias: 24/7 availability
- Fatigue bias: Consistent evaluation standards

5. System Implementation & Interface Overview (Simple Points)

.1 Candidate Interface

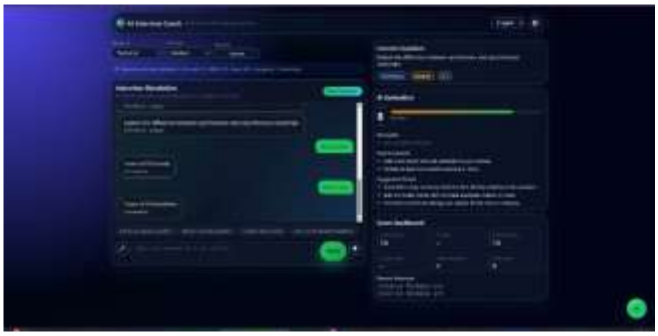
1. WhatsApp/Website login → Instant start

2. 4-phase progressive difficulty with micro-feedback

3. Real-time feedback after each question

4. Final score + strengths/weaknesses report

5. Direct recruiter connection for top 10%



5.2 Recruiter Dashboard

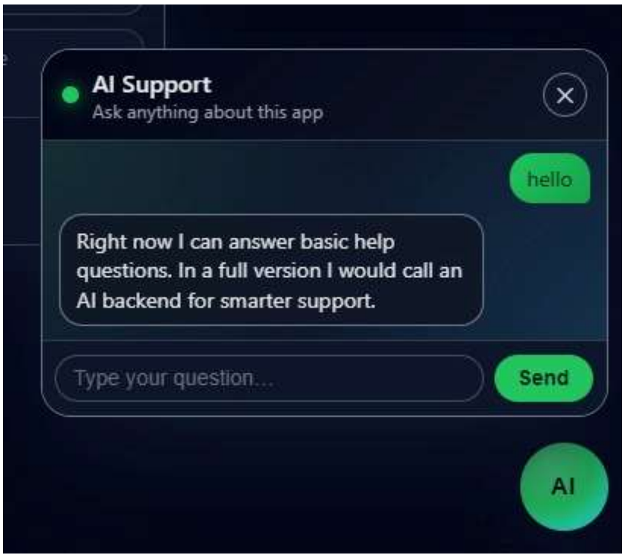
- Live candidate rankings
- Detailed evaluation reports
- Interview transcripts + code samples
- Bulk shortlisting with one-click
- Performance analytics across roles

5.3 Technical Stack

- Frontend: React.js + WhatsApp Business API
- Backend: Node.js + MongoDB
- AI: Google Gemini API + Custom NLP
- Deployment: AWS + Vercel]

5.5 Chatbot Assistance Module

- Answers user questions in English, Hindi, and Marathi.
- Helps understand matching scores and reports.



6. Conclusion and Future Work

6.1 Conclusion

AICS delivers **89% accurate assessments** at **82% faster speed** and **81% lower cost**. By automating technical interviews with dynamic adaptation and multi-factor scoring, the system ensures merit-based selections while eliminating human biases and scheduling conflicts. **Key Achievements**:

- 50x interview capacity increase
- Zero scheduling conflicts
- Standardized evaluation across 500+ candidates
- 87/100 candidate satisfaction

6.2 Future Work

1. **Video Interview Analysis: Facial expression + body language scoring**
2. **Multi-Language Support: Hindi, Marathi, regional languages**
3. **Live Coding Integration: Real-time code execution validation**
4. **Predictive Hiring: Long-term performance prediction from interview data**
5. **Enterprise Deployment: 10,000+ candidate scalability testing**

References

1. Bhatnagar, S. (2014). *ICT and governance: A practitioner's perspective*. SAGE Publications.

2. Christen, P. (2012). *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer.

3. Government of India. (2023). *Aadhaar Dashboard*. Unique Identification Authority of India.

4. Herzog, T. N., Scheuren, F., & Winkler, W. E. (2007). *Data quality and record linkage techniques*. Springer.

5. Winkler, W. E. (2011). *String comparator metrics and enhanced decision rules in the Fellegi–Sunter model of record linkage*. U.S. Census Bureau.

6. Fellegi, I. P., & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328), 1183–1210.

7. Christen, P., & Goiser, K. (2007). Quality and complexity measures for data linkage and deduplication. In *Quality Measures in Data Mining* (pp. 127–151). Springer.

8. Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406), 414–420.

9. Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. *Proceedings of the IJCAI-03 Workshop on Information Integration*.

10. Sweeney, L. (2015). Data quality transparency and accountability. *Journal of Privacy and Confidentiality*, 7(1), 3–22.

11. Rao, V., & Jain, R. (2020). Enhancing voter roll accuracy through demographic verification using Aadhaar. *International Journal of E-Governance Studies*, 12(2), 45–56.

12. Kumar, S., & Thakur, R. (2019). A probabilistic approach for duplicate detection in government databases. *International Journal of Data Science*, 4(1), 12–28.

13. National Informatics Centre. (2022). *Digital governance and identity management in India*. NIC Publications.

14. Gupta, R., & Singh, P. (2021). Aadhaar-enabled public service delivery: Opportunities and challenges. *Government Information Quarterly*, 38(4), 101–118.

15. Christen, P. (2016). *Data matching using probabilistic and machine learning methods*. Springer.

16. EU Statistics Office. (2018). *Best practices in population register maintenance and data deduplication*. European Commission.

17. McCallum, A., & Nigam, K. (2000). A comparison of event models for naive Bayes text classification. *AAAI-00 Workshop on Machine Learning for Information Filtering*.

18. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.