

VeriHire: The NextGen Interview Application

Vivek Yadav¹, Ayush Rai², Abhishek Yadav³, Abhyuday Rai⁴, Aryan Singh⁵

Department of Computer Science and Engineering, Prasad Institute of Technology, Jaunpur, Uttar Pradesh, India

-Guided By: Satendra Kumar

Abstract - Traditional automated interview systems often rely on static question banks, resulting in a generic and impersonal candidate experience. Such systems frequently fail to accurately assess an applicant's unique skill set. VeriHire addresses these challenges by introducing an adaptive, AI-powered interview platform designed to conduct dynamic and personalized interviews.

The system begins by analysing a candidate's professional profile — including resumes, portfolios, and repositories — to understand their expertise. Using Natural Language Processing (NLP) and a Large Language Model (LLM), VeriHire generates tailored, real-time questionnaires and adapts subsequent questions based on the candidate's live responses. This approach enables deeper exploration of technical knowledge, problem-solving skills, and project experience, closely simulating the process of a human interviewer.

MCP (*Model Context Protocol*) is a new standard that allows language models to securely communicate with external tools, databases, and APIs, making AI systems more extensible and capable. RAG (*Retrieval-Augmented Generation*) is a powerful technique that improves model accuracy by fetching relevant information from a knowledge source before generating a response. LangChain is a popular open-source framework that helps developers build applications using language models by chaining prompts, tools, and memory into a structured workflow. The system begins by analysing a candidate's professional profile — including resumes, portfolios, and repositories — to understand their expertise. Using **Natural Language Processing (NLP)** and a **Large Language Model (LLM)**, VeriHire generates tailored, real-time questionnaires and adapts subsequent questions based on the candidate's live responses.

Key Words: MCP, RAG, LANG-CHAIN

1. INTRODUCTION

Technical hiring is often burdened by time-consuming and inconsistent screening methods that lack personalization and are prone to bias. Traditional solutions, from static coding tests to rigid asynchronous interviews, fail to capture a candidate's full potential and unique experience. To address this, VeriHire proposes an adaptive, AI-driven interview platform. By ingesting a candidate's professional profile, the system leverages a Large Language Model (LLM) to generate and dynamically adapt real-time questions. This approach simulates a human-like interview, ensuring every session is unique and provides recruiters with fair, data-rich insights, ultimately modernizing the technical screening.

2. LITERATURE REVIEW

In recent years, technology has significantly changed the way companies hire talent, especially in technical roles. Many platforms now use AI to speed up interviews and evaluations. However, most of these systems still rely on predefined questions and fixed flows, which makes the interview process feel robotic and impersonal. This is where new research in AI, particularly in the fields of Natural Language Processing (NLP) and Large Language Models (LLMs), is starting to make a real difference. One of the most important breakthroughs is a technique called Retrieval-Augmented Generation (RAG). Introduced by researchers like Patrick Lewis and his team, RAG allows AI to pull in relevant information from external sources before generating a response. This means that instead of giving generic answers or questions, the AI can tailor its output based on what it "learns" on the fly — an ideal approach for interviews where every candidate has a unique background.

Building on this, newer models like Active RAG go a step further by constantly retrieving information even while the conversation

is happening. This creates a more dynamic and interactive experience, making the interview feel more like a real-time discussion rather than a scripted session. For example, if a candidate mentions a project, the system can immediately look into that project (from their GitHub or resume) and ask deeper, more specific questions.

Another tool that's helping developers build smarter applications like VeriHire is LangChain. It provides a way to organize language models with tools like memory, APIs, and document storage — kind of like giving the AI a long-term memory and the ability to use external knowledge. This makes it easier to build interviews that flow naturally and stay on topic, adapting based on previous answers.

In addition, a new concept called Model Context Protocol (MCP) is emerging. MCP acts like a secure bridge between AI models and outside systems like databases, resumes, or code repositories. This makes it possible for an AI interviewer to not just understand the candidate's resume but to also ask meaningful questions about specific projects or skills they've listed.

Despite all these advancements, many existing platforms (like HackerRank or HireVue) still fall short. They might be efficient, but they often lack the ability to adjust their questions based on the candidate's real skills or previous answers. They also miss the human touch — the ability to follow up on interesting points or ask more when a candidate says something insightful.

This is where our project, VeriHire, comes in. By combining the latest in AI research — from RAG and LangChain to NLP and MCP — we aim to create a smarter, more engaging interview experience. One that doesn't just test candidates, but actually understands them. Studies suggest that such systems introduce biases due to generic assessment formats and do not effectively simulate real interview dynamics.

The literature thus supports the need for a more nuanced, AI-driven system that leverages modern NLP techniques, adaptive workflows, and external data integration. VeriHire emerges from this gap, combining state-of-the-art research with practical implementation to offer a scalable yet personalized interview solution.

3. EXISTING SOLUTIONS AND GAPS

Current market solutions fall primarily into two categories:

- **Static Assessment Platforms (e.g., Hacker Rank):** Efficient for coding challenges but unable to personalize questions based on candidate background.

- **Asynchronous Interview Tools (e.g., HireVue):** Allow video responses but cannot ask follow-up questions or adapt in real-time.

Key Gaps Identified:

- **Static Question Flow:** No ability to dynamically adjust based on candidate responses.
- **Limited Context Awareness:** Cannot parse resumes or project portfolios to ask targeted, meaningful questions.
- **Impersonal Candidate Experience:** Lacks real-time engagement, failing to capture communication and critical-thinking skills.

4. PROPOSED SOLUTION

VeriHire is an intelligent, next-generation interview platform designed to offer candidates a more personalized and meaningful interview experience — one that closely mirrors how real, human interviews unfold. Unlike traditional systems that rely on static question banks or one-size-fits-all assessments, VeriHire takes a smarter approach. It begins by carefully analyzing each candidate's professional profile — including their **resume**, **GitHub repositories**, **project portfolios**, and other relevant data — to gain a well-rounded understanding of their background, skills, and interests.

Once the candidate's profile is processed, VeriHire's **AI-driven engine** steps in to create a customized interview experience. It generates **questions specifically tailored** to the individual's expertise, such as asking about technologies they've actually used in past projects or exploring challenges they've solved in their code repositories. This ensures that candidates are evaluated based on what they know and have done — not just on general knowledge.

What truly sets VeriHire apart is its ability to **adapt dynamically** during the interview. As the candidate responds, the system intelligently follows up with relevant, deeper questions — just like a skilled human interviewer would. This results in a **more natural, two-way conversation** rather than a rigid Q&A session. Candidates feel more engaged, and recruiters gain deeper insights into how candidates think, communicate, and solve problems in real time.

Overall, VeriHire is a more effective and human-centric tool for modern technical hiring.

5. PROJECT OBJECTIVES

The project aims to:

1. **Parse and Understand Candidate Profiles:** Build AI modules to extract key skills and project details from resumes and repositories.
2. **Enable Real-Time Question Generation:** Implement an NLP-driven engine that formulates context-aware, adaptive questions during the interview.
3. **Develop a Full-Stack Web Platform:** Provide a seamless interface for candidates and recruiters, enabling smooth interaction with the AI interviewer.
4. **Ensure Fairness and Consistency:** Reduce bias by standardizing evaluations, focusing on skills rather than subjective human judgment.

5.

6.METHODOLOGY AND SYSTEM ARCHITECTURE

6.1 SYSTEM ARCHITECTURE

□ Frontend:

- Built using **React.js** or **Next.js**, the frontend provides an interactive and user-friendly interface.
- Candidates can easily upload their documents such as resumes and portfolios.
- The interface facilitates smooth interaction between the candidate and the AI interviewer, making the experience engaging and seamless.

□ Backend:

- Powered by a Python-based server using frameworks like **Flask** or **Django**.
- Manages all business logic including user authentication and authorization.
- Provides APIs that enable communication between the frontend and other system components, ensuring a cohesive workflow.

□ AI/ML Core:

- **Document Parsing Module:** Utilizes NLP libraries like **spaCy** to extract important details such as skills, projects, and keywords from candidate profiles. This module converts unstructured text data into structured information that can be used for question generation.

- **Question Generation Module:** Employs a fine-tuned Large Language Model (LLM) that generates relevant and personalized interview questions. This module also creates adaptive follow-up questions based on candidate responses, enabling a dynamic interview process.

□ Database:

- Uses **MongoDB** or **PostgreSQL** to store all relevant data securely.
- Stores user information, interview history, and processed candidate data, allowing recruiters and the system to retrieve past interactions and analytics efficiently.



Fig -1: Methodology

6.2 DEVELOPMENT METHODOLOGY

To ensure a structured, efficient, and iterative development process, the project will be executed using an Agile methodology divided into clearly defined phases. Each phase focuses on a specific aspect of the system, allowing for continuous development, testing, and refinement.

The project will follow an **Agile, phase-based approach**:

Phase 1 – Data Ingestion & Profile Parsing

- The first step involves building **robust data ingestion pipelines** capable of reading and interpreting various forms of candidate data such as **resumes, GitHub profiles, and project portfolios**.
- Advanced **Natural Language Processing (NLP)** tools like **spaCy** will be used to extract meaningful information, including **skills, experience, project domains, and tools/technologies used**.

- The goal of this phase is to create a strong foundation of **structured, machine-readable candidate data**, which will feed into the AI-driven question generation process.

Phase 2 – AI Model Development

- In this phase, the core intelligence of the platform will be developed by **fine-tuning a pre-trained Large Language Model (LLM)** such as GPT or a transformer-based architecture.
- The model will be trained to generate **context-aware, technically relevant questions** and intelligent follow-ups based on candidate profiles and live responses.
- Emphasis will be placed on **adaptive learning**, where the AI can adjust the depth and direction of questioning in real-time — replicating a natural interview flow.

Phase 3 – System Integration

- This phase focuses on **integrating the AI core with the backend server**, APIs, and the supporting infrastructure.
- Backend development (using Python frameworks like Flask or Django) will handle **authentication, API endpoints, session management**, and interaction between the frontend and the AI model.
- Careful integration will ensure **data security, scalability**, and smooth communication between all system components.

Phase 4 – Frontend Development & Testing

- The final phase includes developing a **responsive and user-friendly frontend interface** using frameworks like React.js or Next.js.
- The interface will allow **candidates to interact with the AI interviewer** and recruiters to view results, analytics, and reports.
- This phase also includes **rigorous testing** — including unit testing, usability testing, and performance testing — to ensure the platform is stable, intuitive, and production-ready.

7. TECHNOLOGY STACK

Layer	Technologies
Frontend	React.js / Next.js, TailwindCSS
Backend	Python (Flask/Django), REST APIs
AI/ML	PyTorch/TensorFlow, Hugging Face Transformers, spaCy, NLTK
Database	MongoDB / PostgreSQL
Deployment	Docker, DigitalOcean / AWS
Authentication	JWT / OAuth 2.0

Table -1: Technologies Used

8. EXPECTED OUTCOMES AND APPLICATIONS

8.1 Expected Outcomes (Expanded)

1. Working Prototype of VeriHire

- The project will deliver a fully functional prototype of the VeriHire platform, demonstrating its ability to conduct dynamic, adaptive, and AI-driven interviews. This prototype will showcase the end-to-end flow — from candidate profile ingestion to personalized question generation and real-time interaction.

2. Real-Time Generation of Context-Aware Questions

- The system will be able to generate intelligent, personalized questions on-the-fly based on the candidate's background, skills, and responses during the interview. This will highlight the system's capacity to simulate a natural, human-like conversation, unlike traditional fixed-question platforms.

3. Interactive Recruiter Dashboard

- A clean, intuitive dashboard will be developed for recruiters to view interview summaries,

candidate rankings, AI-driven insights, and analytics. The dashboard will support decision-making by providing performance metrics, skill mapping, and comparison tools — streamlining the hiring process.

8.2 Applications & Future Scope

1. Technical Hiring for Companies

- VeriHire can be directly integrated into corporate hiring workflows to automate technical screening, reduce interviewer workload, and improve candidate experience by delivering tailored interviews aligned with job requirements.

2. Mock Interviews for Students and Job Seekers

- The platform can be used by educational institutions, training centers, or individuals for mock interview preparation, allowing users to practice in a realistic environment that adapts to their responses and offers constructive feedback.

3. Skill Evaluation for Bootcamps and Training Programs

- Coding bootcamps, online learning platforms, and training institutes can use VeriHire to assess student progress and technical readiness, offering tailored interviews based on the curriculum and real-world job expectations.

CONCLUSIONS

In today's fast-paced and competitive job market, traditional hiring processes often fall short in accurately evaluating a candidate's true capabilities, particularly in technical domains. Manual interviews, while thorough, lack scalability, and automated systems, though efficient, tend to be impersonal and generic. This creates a significant gap in the recruitment process — one that compromises both fairness and effectiveness. **VeriHire** is designed to address this challenge head-on by blending the scalability of automation with the contextual understanding and adaptability of human-led interviews.

By leveraging advanced technologies such as **Artificial Intelligence (AI)**, **Natural Language Processing (NLP)**, **Large Language Models (LLMs)**, and modern frameworks like **RAG**

(Retrieval-Augmented Generation) and **LangChain**, VeriHire enables a more dynamic, context-aware, and personalized interview experience. The system is capable of analyzing candidate resumes, project portfolios, and coding repositories to generate questions that are relevant and tailored to individual profiles. As the interview progresses, the platform adapts in real-time to the candidate's responses, much like a human interviewer would — diving deeper into areas of strength or exploring specific technologies mentioned.

This personalized approach not only improves the candidate experience by making it more engaging and meaningful but also empowers recruiters with deeper insights into each applicant's skill set, communication style, and problem-solving abilities. By reducing bias, standardizing evaluations, and ensuring that questions are based on merit and context, VeriHire promotes a **fairer, more inclusive hiring process**.

The broader impact of VeriHire goes beyond recruitment efficiency. It paves the way for more **data-driven, transparent, and equitable hiring practices** that value candidates not just for how well they perform on a standardized test, but for their actual skills, experience, and potential. It also opens up opportunities for wider applications, such as mock interviews for job seekers, technical assessments for training programs, and skill evaluation for educational institutions.

ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere and heartfelt gratitude to our esteemed guide, **Professor Satendra Kumar**, whose continuous support, expert supervision, and unwavering encouragement played a pivotal role in the successful completion of our project, *VeriHire: The NextGen Interview Application*. His vast knowledge, constructive feedback, and thoughtful suggestions consistently guided us in the right direction and helped us overcome numerous technical and conceptual challenges throughout the development process. His mentorship has not only enriched this project but also significantly contributed to our academic and personal growth.

We are also deeply thankful to the **Department of Computer Science and Engineering, Prasad Institute of Technology, Jaunpur**, for providing us with the necessary academic environment, infrastructure, and access to learning resources that enabled us to carry out our work smoothly. The department's

emphasis on innovation, research, and practical application laid the foundation for this project and motivated us to push our boundaries.

We would like to extend our special thanks to the **open-source developer communities** and contributors whose efforts have given rise to powerful frameworks and tools such as **LangChain**, **Hugging Face Transformers**, **spaCy**, and various other AI and NLP libraries. These tools formed the technological backbone of our system and allowed us to explore and implement cutting-edge solutions. Without the generosity and collaborative spirit of these communities, developing a project like VeriHire would not have been possible.

Last but certainly not least, we would like to express our deepest gratitude to our **families and friends**, whose constant motivation, emotional support, and patience gave us the strength to stay focused and committed throughout this journey. Their belief in our abilities helped us stay grounded and determined.

This project is the result of the collective support, guidance, and encouragement we have received from all these individuals and institutions. We are truly grateful for each contribution.

REFERENCES

1. Lewis, P., Perez, E., Piktus, A., Karpukhin, V., Goyal, N., K ttler, H., ... & Riedel, S. (2020). **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. *Advances in Neural Information Processing Systems (NeurIPS) 2020*. <https://arxiv.org/abs/2005.11401>
2. Li, H., Yin, P., Xiong, C., Socher, R., & Zettlemoyer, L. (2022). **A Survey on Retrieval-Augmented Text Generation**. *arXiv preprint arXiv:2206.06668*. <https://arxiv.org/abs/2206.06668>
3. Jiang, Y., Xiao, C., Yin, D., & Zhang, B. (2023). **Active Retrieval Augmented Generation**. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. <https://aclanthology.org/2023.emnlp-main.1/>
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is All You Need**. *Advances in Neural Information Processing Systems*. <https://arxiv.org/abs/1706.03762>
5. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). **Transformers: State-of-**

the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

<https://arxiv.org/abs/1910.03771>

6. LangChain Documentation – A framework for building applications with language models. <https://docs.langchain.com/docs/>
7. spaCy: Industrial-Strength Natural Language Processing in Python. <https://spacy.io/>