

# **AI-Powered Interview Automation System**



A Project report submitted in partial fulfillment of  
requirements for the award of degree of

## **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

by

**BANDARI UCHITH KUMAR (219X1A05A4)**

**PYREDDY SREEJA REDDY (219X1A05K4)**

**BUKKE SIVA NAIK (219X1A05K8)**

Under the esteemed guidance of

**Smt. C. Ratna Prabha**

Assistant Professor

Department of C.S.E.

**Department of Computer Science and Engineering**

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL**

(Affiliated to JNTUA, ANANTAPURAMU)

**2024 – 2025**

## Department of Computer Science and Engineering

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL**

**(Affiliated to JNTUA, ANANTAPURAMU)**



### **CERTIFICATE**

**This is to certify that the Project Work entitled ‘AI-Powered Interview Automation System’ is a bonafide record of work carried out by**

**BANDARI UCHITH KUMAR (219X1A05A4)**

**PYREDDY SREEJA REDDY (219X1A05K4)**

**BUKKE SIVA NAIK (219X1A05K8)**

**Under my guidance and supervision in partial fulfillment of the requirements for the award of degree of**

### **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING**

.....  
**Smt. C. Ratna Prabha**

Assistant Professor,  
Department of C.S.E.,  
G. Pulla Reddy Engineering College,  
Kurnool.

.....  
**Dr. N. Kasiviswanath**

Professor & Head of the Department,  
Department of C.S.E.,  
G. Pulla Reddy Engineering College,  
Kurnool.

## **DECLARATION**

I hereby declare that the project titled “**AI-POWERED INTERVIEW AUTOMATION SYSTEM**” is an authentic work carried out by me as the student of **G. PULLA REDDY ENGINEERING COLLEGE(Autonomous) Kurnool**, during 2024-25 and has not been submitted elsewhere for the award of any degree or diploma in part or in full to any institute.

**Bandari Uchith Kumar**  
**(219X1A05A4)**  
**Pyreddy Sreeja Reddy**  
**(219X1A05K4)**  
**Bukke Siva Naik**  
**(219X1A05K8)**

## **ACKNOWLEDGEMENT**

We wish to express our deep sense of gratitude to our project guide **Smt C. Ratna Prabha, Assistant Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for his immaculate guidance, constant encouragement and cooperation which have made possible to bring out this project work.

We are grateful to our project incharge **Dr. K. Ishthaq Ahmed, Associate Professor** of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

We are thankful to our Head of the Department **Dr.N.Kasiviswanath**, for his whole hearted support and encouragement during the project sessions.

We are grateful to our respected Principal **Dr.B.Sreenivasa Reddy**, for providing requisite facilities and helping us in providing such a good environment.

We wish to convey our acknowledgements to all the staff members of the Computer Science and Engineering Department for giving the required information needed for our project work.

Finally we wish to thank all our friends and well wishers who have helped us directly or indirectly during the course of this project work.

## **ABSTRACT**

The AI-Powered Interview Automation System project revolutionizes recruitment by leveraging cutting-edge technologies such as AI, machine learning, and NLP to optimize the end-to-end hiring process. It not only automates repetitive tasks like scheduling, candidate screening, and interview coordination, but also intelligently assesses candidate responses in real-time. Through AI-powered analysis of both textual and spoken inputs, the system provides unbiased evaluations and immediate feedback during interviews.

One standout feature is the **AI-generated interview** questions, which are tailored to job-specific criteria, ensuring that each candidate is assessed based on relevant skills. The platform also includes a robust **candidate management system**, allowing HR teams to track applications seamlessly from submission through post-interview reviews. Furthermore, **post-interview analytics** and **performance reports** help hiring managers make data-backed decisions, promoting transparency and reducing subjective judgment.

By integrating these advanced technologies, the system significantly reduces time-to-hire, boosts scalability, and improves the overall effectiveness of the recruitment process, making it adaptable to organizations of any size. This innovative approach helps companies identify the best talent efficiently while ensuring a streamlined, unbiased, and data-driven hiring experience.

## **TABLE OF CONTENTS**

1.INTRODUCTION .....	1
1.1 INTRODUCTION .....	2
1.2 MOTIVATION.....	3
1.3 PROBLEM DEFINATION .....	4
1.4 OBJECTIVE OF THE PROJECT .....	4
1.5 LIMITATIONS OF THE PROJECT .....	5
1.6 ORGANIZATION OF THE PROJECT .....	5
2.SYSTEM SPECIFICATIONS .....	7
2.1 SOFTWARE SPECIFICATIONS .....	8
2.2 HARDWARE SPECIFICATIONS : .....	10
3.LITERATURE SURVEY .....	12
3.1 INTRODUCTION .....	13
3.2 EXISTING SYSTEM .....	14
3.3 LIMITATIONS OF EXISTING SYSTEM .....	14
3.4 PROPOSED SYSTEM.....	16
4.DESIGN AND IMPLEMENTATION .....	20
4.1 INTRODUCTION .....	21
4.2 UML DIAGRAMS.....	22
4.3 IMPLEMENTATION.....	27
4.4 SOURCE CODE.....	33
5.RESULTS .....	37
6.CONCLUSION.....	45
6.1 CONCLUSION .....	46
6.2 FUTURE ENHANCEMENT.....	46
REFERENCES .....	49

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>FIGURE NAME</b>	<b>PAGE NO.</b>
4.2.1	Use Case Diagram	23
4.2.2	Class Diagram	24
4.2.3	Activity Diagram	25
4.2.4	Sequence Diagram	26

# **1.INTRODUCTION**



## 1.1 INTRODUCTION

Recruitment has always been a fundamental aspect of organizational growth, evolving significantly over time. In the early days, hiring was a personal and informal process, where employers selected candidates based on personal recommendations or direct interactions. As industries expanded and businesses grew, the need for structured hiring methods led to the introduction of formal interviews, written tests, and skill assessments. The 20th century saw the rise of standardized recruitment practices, where companies began using structured interviews, group discussions, and aptitude tests to evaluate candidates more objectively.

With the advancement of **technology in the late 20th and early 21st centuries**, recruitment began to shift towards digital platforms. The introduction of **online job portals and applicant tracking systems (ATS)** allowed recruiters to reach a larger pool of candidates, making hiring more efficient. Video interviews, online assessments, and resume screening tools became common, helping companies manage large-scale recruitment. However, despite these improvements, many challenges remained—such as **time-consuming screening processes, scheduling conflicts, interviewer biases, and inefficiencies in candidate evaluation**.

To address these issues, **interview automation** emerged as a revolutionary approach. Instead of relying solely on manual efforts, companies started incorporating **automated tools to streamline hiring**. Automated interview systems can now handle various aspects of recruitment, including **resume screening, chatbot-based interviews, and AI-driven candidate analysis**. These advancements have significantly reduced hiring time and improve the accuracy of candidate evaluations.

Moreover, automated interviews enhance the **candidate experience** by providing faster responses, reducing long waiting periods, and offering instant feedback. With the ability to conduct **real-time assessments and structured interviews**, companies can efficiently hire top talent while ensuring a transparent and engaging recruitment process.

As organizations continue to grow and technology advances, **interview automation is set to become a crucial part of modern recruitment**. From early face-to-face interactions to AI-powered hiring solutions, the evolution of recruitment reflects the changing needs of businesses in a fast-paced world. By adopting **interview automation**, companies can ensure a **more efficient, unbiased, and data-driven hiring process**, ultimately leading to better workforce management and business success.

## **1.2 MOTIVATION**

The Interview Automation System is developed to address the growing challenges in the hiring process by integrating technology-driven solutions.

### **Inefficiencies in Traditional Hiring**

Manual recruitment involves time-consuming resume screening, interview scheduling, and subjective evaluations, leading to delays. The high dependency on HR personnel increases operational costs and hiring time. Automation helps streamline these processes, improving efficiency and reducing effort.

### **Fair and Unbiased Hiring**

Human biases can lead to unfair hiring decisions and lack of diversity in recruitment. Automated systems use standardized evaluation criteria to ensure fair and objective candidate selection. This promotes merit-based hiring, reducing discrimination and errors.

### **Scalability and Efficiency**

Companies handling large volumes of applications struggle with manual screening and assessments. Automation enables simultaneous evaluation of multiple candidates, reducing hiring time. This improves scalability, making mass recruitment easier and more structured.

### **Enhanced Candidate Experience**

Long wait times and lack of communication frustrate candidates and affect company reputation. Automated interview systems provide real-time updates, structured interview formats, and instant feedback. This creates a positive experience, attracting top talent to the organization.

## **Cost Reduction**

Recruitment costs include advertising, HR salaries, and consultancy fees, which increase expenses. Automating interview processes helps reduce hiring costs by minimizing manual effort. Companies can reallocate resources efficiently, optimizing HR functions.

## **Technological Advancement**

The rise of AI and automation has transformed hiring processes globally. Organizations adopting smart hiring solutions improve decision-making and operational efficiency. Interview automation ensures data-driven hiring and long-term workforce optimization.

## **1.3 PROBLEM DEFINATION**

Traditional recruitment processes are **time-consuming, labor-intensive, and prone to human biases**, making it difficult for organizations to hire efficiently. **Manual resume screening, interview scheduling conflicts, and subjective evaluations** often lead to delays, increased hiring costs, and inconsistencies in candidate selection. As the number of applicants increases, recruiters struggle to manage **large-scale hiring** while maintaining fairness and accuracy.

Additionally, candidates often experience **long waiting times and lack of feedback**, leading to a poor recruitment experience. The absence of a **standardized assessment method** results in **inconsistent evaluations** and potential hiring errors. These challenges highlight the need for an **automated interview system** that can **streamline recruitment, enhance efficiency, and ensure objective hiring decisions**.

This project aims to **develop an Interview Automation System** that reduces **manual effort, improves evaluation accuracy, accelerates hiring timelines, and enhances the candidate experience**. By automating key recruitment tasks, organizations can **optimize their hiring process, reduce costs, and select the best talent efficiently**.

## **1.4 OBJECTIVE OF THE PROJECT**

The primary objectives of this project are:

- To automate the initial screening process by evaluating resumes and matching candidates based on job requirements.

- To enable AI-based interview assessments that analyze responses, assess soft skills, and rank candidates accordingly.
- To provide an automated scheduling system that eliminates manual coordination efforts between recruiters and candidates.
- To generate instant feedback and reports based on structured evaluation metrics, reducing delays in decision-making.
- To minimize human bias by ensuring a standardized and fair evaluation process for all candidates.
- To improve candidate experience by offering quick responses and an efficient recruitment process.

## 1.5 LIMITATIONS OF THE PROJECT

- **Limited Human Judgment** – The system may not effectively assess creativity, emotions, and critical thinking required for certain roles.
- **Dependence on Internet and Technology** – Network failures and technical issues can disrupt the interview process.
- **AI Evaluation Limitations** – Automated assessments may misinterpret responses and fail to analyze non-verbal cues accurately.
- **Resistance to Adoption** – Some recruiters and candidates prefer traditional face-to-face interviews over automated ones.
- **Security and Data Privacy Concerns** – Handling sensitive candidate data requires strong security to prevent breaches.
- **Customization Challenges** – Different companies need customized hiring processes, requiring frequent updates and modifications.

These limitations highlight the need for continuous improvements to enhance the effectiveness of interview automation.

## 1.6 ORGANIZATION OF THE PROJECT

The Interview Automation System aims to enhance the hiring process by eliminating inefficiencies, reducing biases, and improving scalability. Throughout this report, we have explored various aspects of the system:

- **Chapter 1 (Introduction):** Established the need for automation in recruitment, highlighting key challenges in traditional hiring methods.
- **Chapter 2 (System Specifications):** Defined the software and hardware requirements essential for implementing the system effectively.
- **Chapter 3 (Literature Survey):** Analyzed existing hiring processes, their disadvantages, and proposed an improved automated approach.
- **Chapter 4 (Design and Implementation):** Outlined the system architecture, workflow, and functionality, showcasing how automation streamlines recruitment.

By integrating automation, this project reduces manual effort, speeds up hiring, and ensures a fair, data-driven evaluation process.

## **2.SYSTEM SPECIFICATIONS**

These are the requirements for doing the project. Without using these tools and software, we cannot do the project. So, we need two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

## **2.1 SOFTWARE SPECIFICATIONS**

### **1. Operating System (OS):**

The system should be compatible with multiple operating systems to ensure flexibility in deployment.

- Windows 10/11 – Most commonly used for development and deployment.
- Linux (Ubuntu, CentOS, Red Hat, etc.) – Preferred for servers due to stability, security, and open-source nature.
- macOS – Used in development environments, mainly by Apple ecosystem users.

### **2. Programming Languages:**

The selection of programming languages depends on the backend, frontend, and AI requirements.

- Backend: Python, Java, or Node.js for handling server-side logic.
- Frontend: JavaScript (React.js, Next.js) for building a dynamic user interface.

### **3. Database Management System (DBMS):**

A reliable database is required to store candidate information, test scores, interview recordings, and logs.

- MySQL / PostgreSQL – Relational databases for structured data storage.
- MongoDB – NoSQL database for flexible and scalable data handling.
- Firebase / AWS DynamoDB – Cloud-based databases for real-time data synchronization.

### **4. Web Technologies:**

A responsive and interactive web application is essential for candidates and recruiters.

- Frontend:
  - HTML5, CSS3 for structure and styling.
  - JavaScript frameworks (React.js, Angular.js, or Vue.js) for dynamic and interactive UI.
  - Bootstrap / Tailwind CSS for responsive design.

- Backend:
  - Django (Python), Spring Boot (Java), or Express.js (Node.js) for API and business logic.
  - Flask (Python) for lightweight backend development.

## **5. Frameworks & Libraries**

Additional libraries and frameworks enhance the efficiency of the system.

- Machine Learning & AI : TensorFlow, OpenCV, or Scikit-learn for facial recognition, voice analysis, and evaluation.
- Frontend Frameworks: Bootstrap, Tailwind CSS for modern UI design.
- Backend Frameworks: Express.js (Node.js), Flask (Python), or Spring Boot (Java).

## **6. Server Requirements:**

The system can be hosted on either a local server or a cloud-based environment

- Cloud Services (AWS, Azure, Google Cloud, Firebase) – Used for scalability and remote accessibility.
- Docker / Kubernetes – Containerization and orchestration for easy deployment and management.

## **7. Integrated Development Environment (IDE) & Development Tools:**

Developers need a reliable IDE for writing and debugging code efficiently.

- VS Code – Lightweight and supports multiple languages.
- PyCharm – Best for Python-based development.
- Postman – API testing tool for backend services.
- Git/GitHub/GitLab – Version control for collaborative development.

## **8. Security & Authentication:**

Security is essential for handling confidential candidate data.

- OAuth / JWT (JSON Web Token) – Secure authentication for user access.
- SSL/TLS Encryption – Protects data transmission.
- Firewall & Anti-DDoS Protection – Ensures security from external threats.



## **2.2 HARDWARE SPECIFICATIONS :**

### **1. Processor:**

A powerful processor is required to handle multiple user requests, AI computations, and database operations.

- **Minimum:** Intel Core i5 (10th Gen) / AMD Ryzen 5
- **Recommended:** Intel Core i7/i9 (11th Gen and above) / AMD Ryzen 7/9

### **2. RAM (Random Access Memory):**

Sufficient RAM is needed to ensure smooth execution of web applications, database queries, and AI models.

- **Minimum:** 8GB
- **Recommended:** 16GB or higher

### **3. Storage:**

Fast storage is required for handling resumes, interview recordings, and logs.

- **Minimum:** 256GB SSD
- **Recommended:** 512GB SSD or higher

### **4. Graphics Card :**

For AI-based facial recognition and video analysis, a dedicated GPU is required.

- **Minimum:** Integrated Intel UHD Graphics
- **Recommended:** NVIDIA GTX 1650 or higher

### **5. Network Requirements:**

Stable internet connectivity is required for real-time interviews and cloud-based operations.

- **Minimum:** 50 Mbps
- **Recommended:** 100 Mbps or higher

### **6. Camera & Microphone:**

A good-quality camera and microphone are necessary for video interviews.

- **Webcam:** 1080p HD resolution for clear video interviews.
- **Microphone:** Noise-cancelling microphone for better audio quality.

## **7. Cloud Storage:**

If the system stores candidate data and interview recordings in the cloud, it requires:

- **Google Drive / AWS S3 / Firebase Storage** – Secure storage for resumes, video interviews, and evaluation reports.

## **8. Additional Peripherals:**

- **Speakers/Headphones** – For clear audio in video interviews.
- **Power Backup (UPS)** – Ensures uninterrupted operations in case of power failure.

### **3.LITERATURE SURVEY**

### 3.1 INTRODUCTION

Recruitment has always been a critical function in every organization, but over time, the hiring process has become increasingly complex and time-consuming. Traditional interview methods involve multiple stages—screening resumes, conducting preliminary tests, scheduling interviews, evaluating candidates—and each step is often handled manually. This not only requires significant human effort but also introduces subjectivity, delays, and inconsistencies.

In the era of digital transformation, where companies strive for speed, accuracy, and efficiency, the manual process of recruitment is proving to be a bottleneck. Organizations are now facing challenges in handling large volumes of applicants, ensuring objective assessments, and delivering a smooth candidate experience. This is where the need for Interview Automation arises.

Interview Automation is a modern approach that leverages technology to transform the hiring process. It uses software systems to automate various stages of an interview—right from application intake and question delivery to recording responses, analyzing them, and providing performance feedback. The system ensures standardized evaluation by using predefined rubrics or even AI-based models for scoring. This results in fairer assessments, reduced bias, and considerable time savings for HR departments.

Moreover, interview automation systems are scalable and can handle hundreds of applicants simultaneously, making them ideal for mass hiring or campus placements. With the integration of technologies such as Natural Language Processing (NLP), speech recognition, and machine learning, these systems can now simulate realistic interview environments and offer personalized feedback to candidates.

In summary, Interview Automation bridges the gap between human expertise and digital efficiency, allowing recruiters to focus more on decision-making rather than repetitive administrative tasks. It is a smart solution for modern-day recruitment challenges—offering accuracy, scalability, and fairness in candidate evaluation.

### **3.2 EXISTING SYSTEM**

In the existing system of recruitment and interview management, the entire process is heavily manual and dependent on human intervention. The workflow typically begins with collecting resumes, followed by shortlisting candidates based on qualifications or experience. Once shortlisted, candidates are scheduled for interviews—often involving multiple rounds with HR, technical panels, and management.

Interview questions are asked in real-time by human interviewers, and responses are evaluated manually. Feedback and scoring vary from person to person, and in some cases, interviews are not documented, leading to difficulties in re-evaluation or bias checking. Moreover, coordinating interview schedules between candidates and interviewers is time-consuming, especially in large-scale hiring or campus recruitment drives.

The existing systems may use some digital tools like email for communication, spreadsheets for score tracking, and basic applicant tracking systems (ATS) for storing resumes. However, there is a lack of integration, automation, and intelligence. This results in a process that is inefficient, inconsistent, and not scalable.

Despite the increasing availability of online platforms and recruitment tools, most organizations still rely on traditional face-to-face or telephonic interviews, with limited automation in question delivery, candidate evaluation, or performance analytics.

### **3.3 LIMITATIONS OF EXISTING SYSTEM**

Despite being the standard approach for decades, the traditional interview and recruitment system exhibits several critical limitations that affect the efficiency, fairness, and quality of hiring decisions. These limitations become more apparent in large-scale recruitment scenarios or when consistency and speed are vital.

#### **1. Time-Consuming Process**

The manual interview process involves several stages—from resume screening to multiple rounds of interviews—that require significant time and effort from HR

personnel and technical panels. This leads to delays in hiring, increased workloads, and a longer time-to-fill for open positions.

## **2. High Dependence on Human Involvement**

Every stage of the traditional interview process relies heavily on human input—from shortlisting candidates to conducting interviews and scoring them. This creates dependency and limits the number of candidates that can be processed efficiently in a given time.

## **3. Inconsistency and Human Bias**

Different interviewers may have varying evaluation standards, leading to inconsistent scoring and decision-making. Unconscious bias, favoritism, or subjective judgment may affect the fairness and objectivity of the interview outcome.

## **4. Scheduling and Coordination Challenges**

Aligning the availability of interviewers and candidates is often difficult, especially across different locations or time zones. This results in scheduling conflicts, delays, and even missed opportunities for qualified candidates.

## **5. Lack of Real-Time Evaluation Metrics**

Manual interviews do not provide instant feedback or analytics. There is no immediate way to analyze performance trends, compare candidate scores objectively, or identify skill gaps unless everything is recorded and analyzed manually.

## **6. Poor Documentation and Record-Keeping**

In many cases, interview feedback is noted down informally or only partially. Without recorded sessions or standardized documentation, it becomes challenging to revisit past interviews for verification, audit, or reference.

## **7. Scalability Limitations**

Traditional systems are not designed for high-volume recruitment. Managing thousands of candidates during campus drives or bulk hiring campaigns becomes impractical without an automated solution.

## **8. Limited Candidate Experience**

The manual process may lead to longer wait times, inconsistent communication, and unclear feedback—resulting in a negative experience for the candidate and potentially harming the organization’s employer brand.

These limitations clearly indicate the need for an intelligent, automated system that can streamline the interview process, improve evaluation accuracy, reduce human involvement, and provide a better experience for both recruiters and candidates.

### **3.4 PROPOSED SYSTEM**

To address the shortcomings of the existing recruitment process, the **Interview Automation System** is proposed as an advanced solution that leverages technology to streamline and enhance candidate evaluation. It introduces automation, consistency, and scalability while minimizing manual involvement and human error.

#### **1. Overview of the Proposed System**

The proposed system aims to automate every stage of the interview process, from candidate registration to final evaluation. It uses web technologies, databases, and intelligent scoring algorithms to create a structured, unbiased, and efficient interview environment.

Candidates log in to the platform, complete interviews online by responding to predefined questions (text/audio/video), and receive automated assessments. This ensures a standardized process, reduces the burden on HR teams, and accelerates decision-making.

#### **2. Automation of Interview Process**

The core of the system is a fully automated interview interface. After registration, candidates are presented with questions either randomly or based on predefined criteria (role, skill, or difficulty level). Their answers are recorded in real-time.

Textual responses may be processed through Natural Language Processing (NLP) techniques, while audio/video responses can be analyzed using speech or

sentiment analysis tools if integrated. This replaces live interviewers, making the process available 24/7 and globally accessible.

### 3. Smart Evaluation and Scoring

The system uses predefined rubrics or machine learning models to evaluate candidate responses. Metrics such as relevance, correctness, grammar, confidence (for video/audio), and coherence are used to generate a final score.

This automated scoring mechanism ensures **consistency**, **objectivity**, and **real-time feedback**, eliminating bias and providing recruiters with a reliable performance report for each candidate.

### 4. User Roles and Dashboard Management

The system is designed with two major user roles: **Admin/Recruiter** and **Candidate**. Each has access to their own dashboard.

- **Admin Dashboard** allows for:
  - Adding/modifying question sets
  - Monitoring interview sessions
  - Viewing and exporting performance reports
  - Managing candidate data
- **Candidate Dashboard** allows for:
  - Profile management
  - Taking interviews
  - Viewing their feedback/report card

### 5. Real-Time Report Generation

Once an interview is completed, the system automatically generates a performance report. This includes:

- Section-wise scores



- Overall percentage or grade
- Time taken
- Weakness and strength analysis (optional, if ML/NLP is used)

This helps recruiters quickly make informed hiring decisions.

## **6. Scalability and Bulk Recruitment**

The platform supports bulk user registrations and simultaneous interview sessions, making it ideal for campus placements or large-scale hiring drives. This removes logistical challenges and saves tremendous time for both companies and applicants.

## **7. Data Storage and Security**

All candidate data, responses, and evaluation results are stored in a secure database. Access control mechanisms and encryption techniques can be applied to ensure data privacy and integrity, which is essential for handling sensitive personal and performance data.

## **8. Integration with Other Systems**

The system can be enhanced with integration capabilities for:

- Resume screening tools
- Video proctoring solutions
- HRMS platforms for end-to-end recruitment
- Email/SMS APIs for notifications

This makes the interview automation platform flexible and expandable in real-world enterprise environments.

## 9. Ethical and Fair Use

To ensure fairness, the system is designed to minimize bias in evaluation by using objective scoring logic. For added transparency, interviewers or HR admins can view detailed evaluation logs or override scores where necessary.

Additionally, candidates are informed about how their data will be used, and consent is obtained during registration to maintain ethical standards.

## 10. Benefits of the Proposed System

- **Time-Saving:** Drastically reduces time spent on manual interviews
- **Consistency:** Same set of questions and rules applied to all
- **Scalability:** Handles thousands of candidates without logistical constraints
- **Transparency:** Clear evaluation criteria and recorded responses
- **Accessibility:** Available 24/7 across geographical boundaries
- **Fairness:** Reduces unconscious bias by standardizing evaluation

This **Interview Automation System** represents a significant step forward in modernizing recruitment. By combining technology, efficiency, and ethical design, it offers a powerful solution for organizations seeking to transform their hiring process.

## **4.DESIGN AND IMPLEMENTATION**

## 4.1 INTRODUCTION

Design and implementation are crucial phases in the software development lifecycle, transforming theoretical concepts and requirements into a functional system. In this chapter, we focus on how the Interview Automation System is architected and brought to life using software design principles and development tools.

The **design phase** involves modelling the structure and behaviour of the system through **UML (Unified Modelling Language) diagrams** such as use case, class, sequence, and activity diagrams. These visual representations help in understanding system interactions, workflows, and data flow.

The **implementation phase** covers the actual coding of the system components, integrating front-end and back-end technologies, and connecting them to the database. The system is built to ensure scalability, user-friendliness, and automation of interview processes.

### 4.1.1 PURPOSE

The purpose of the Interview Automation System is to streamline and automate the traditional interview process by developing a digital platform that minimizes manual intervention. This system is designed to:

- Reduce the time and effort involved in scheduling and conducting interviews.
- Enable consistent and unbiased evaluation of candidates.
- Allow interviewers to focus on quality decision-making rather than administrative tasks.
- Provide an accessible and user-friendly environment for candidates to attend interviews remotely.
- Maintain a structured database of interview results, reports, and candidate performance for future reference.

By achieving these goals, the system contributes to a more efficient, transparent, and scalable recruitment process.

### 4.1.2 SCOPE

The scope of the Interview Automation System includes the development of a web-based application that facilitates the end-to-end automation of the interview process. It

provides functionalities for different types of users, including **Administrators**, **Interviewers**, and **Candidates**.

The key features covered within the scope of this project are:

- **User Authentication:** Secure login and role-based access for admins, interviewers, and candidates.
- **Interview Scheduling:** Admins can schedule interviews and assign them to candidates.
- **Question Management:** Uploading and managing question banks (MCQs, coding questions, or subjective).
- **Interview Process:** Candidates can attend interviews online by answering the given set of questions.
- **Auto-Evaluation:** The system evaluates objective questions automatically and stores results.
- **Reports Generation:** Final scores and performance metrics are generated for review by the admin/interviewer.
- **Data Management:** Secure storage of candidate data, interview records, and question sets.

The system is scalable and can be further enhanced with AI-based voice/video interview evaluations or integration with existing HR systems.

## 4.2 UML DIAGRAMS

Unified Modeling Language (UML) diagrams are standardized visual representations used in software engineering. They include Use Case Diagrams for interactions, Class Diagrams for structure, Sequence Diagrams for chronological interactions, Activity Diagrams for workflows, and State Machine Diagrams for state transitions. UML diagrams enhance communication, simplify system comprehension, and serve as blueprints for software development by providing a standardized visual language.

### 4.2.1 USE CASE DIAGRAM

The use case diagram represents the interactions between the system and its users (actors). It highlights the major functionalities available to each type of user.

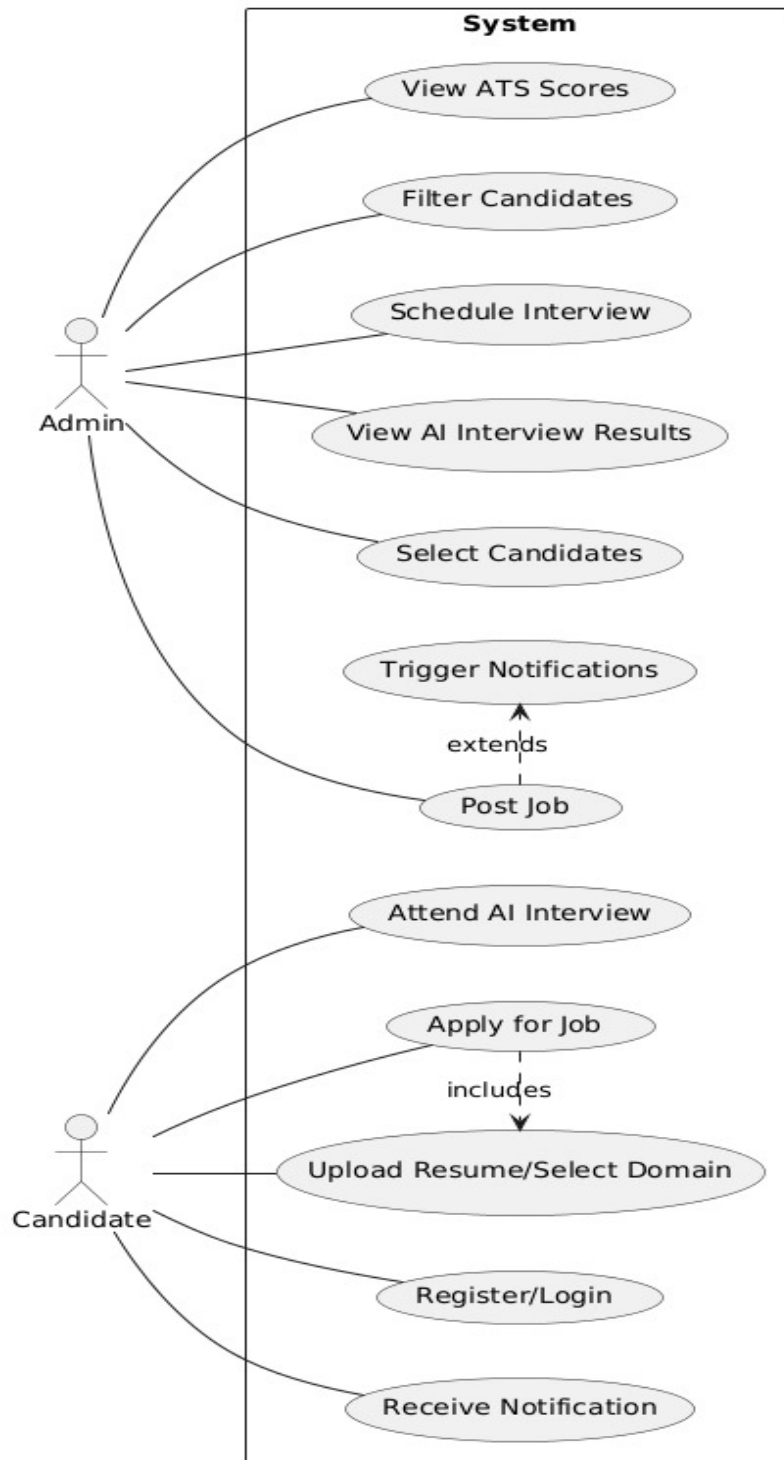


Fig 4.2.1 Use Case

### 4.2.2 CLASS DIAGRAM

The class diagram outlines the structure of the system in terms of its classes, attributes, and methods, along with relationships between them.

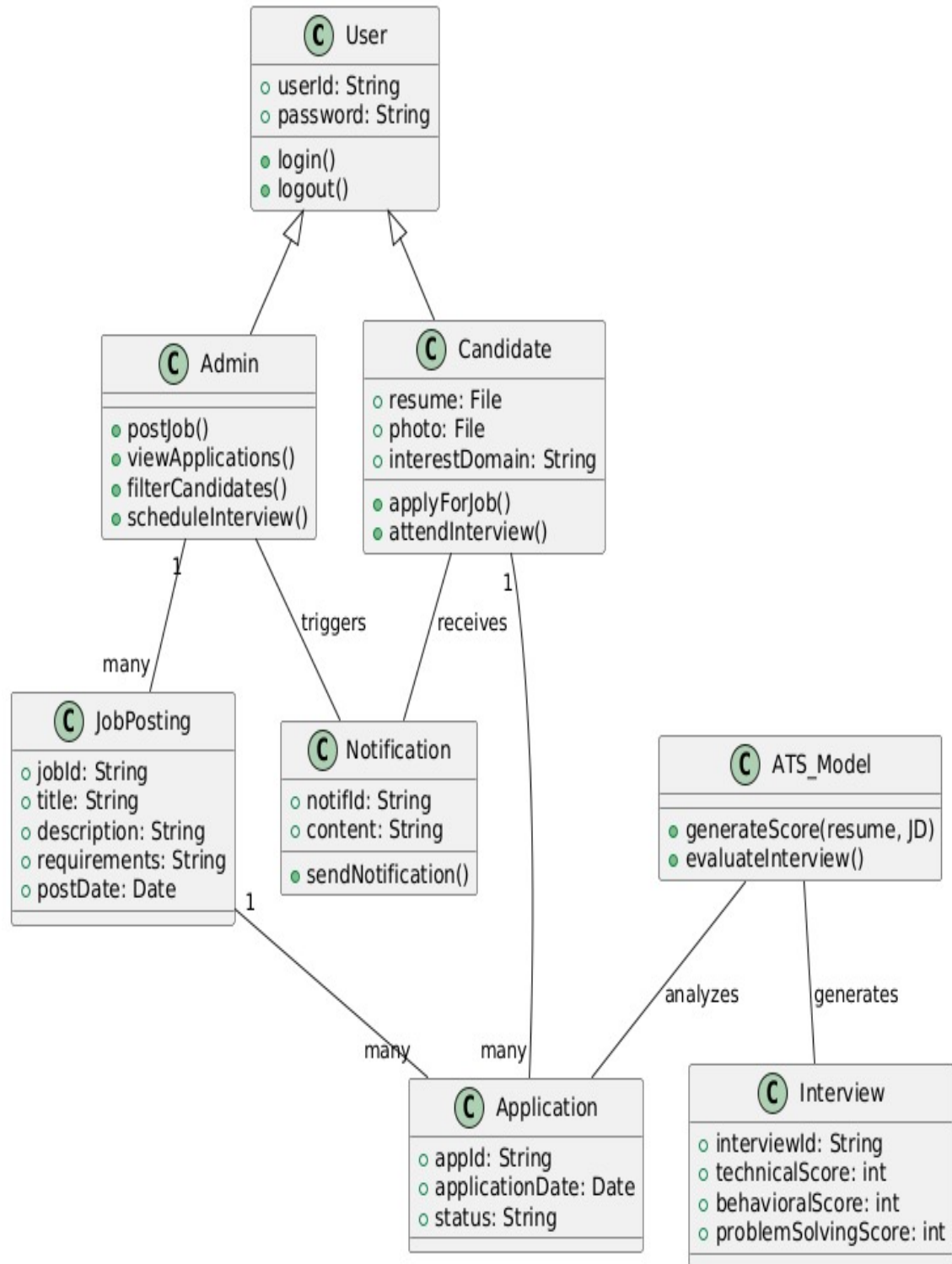


Fig 4.2.2 Class Diagram

### 4.2.3 ACTIVITY DIAGRAM

The activity diagram shows the flow of control from one activity to another in a process.

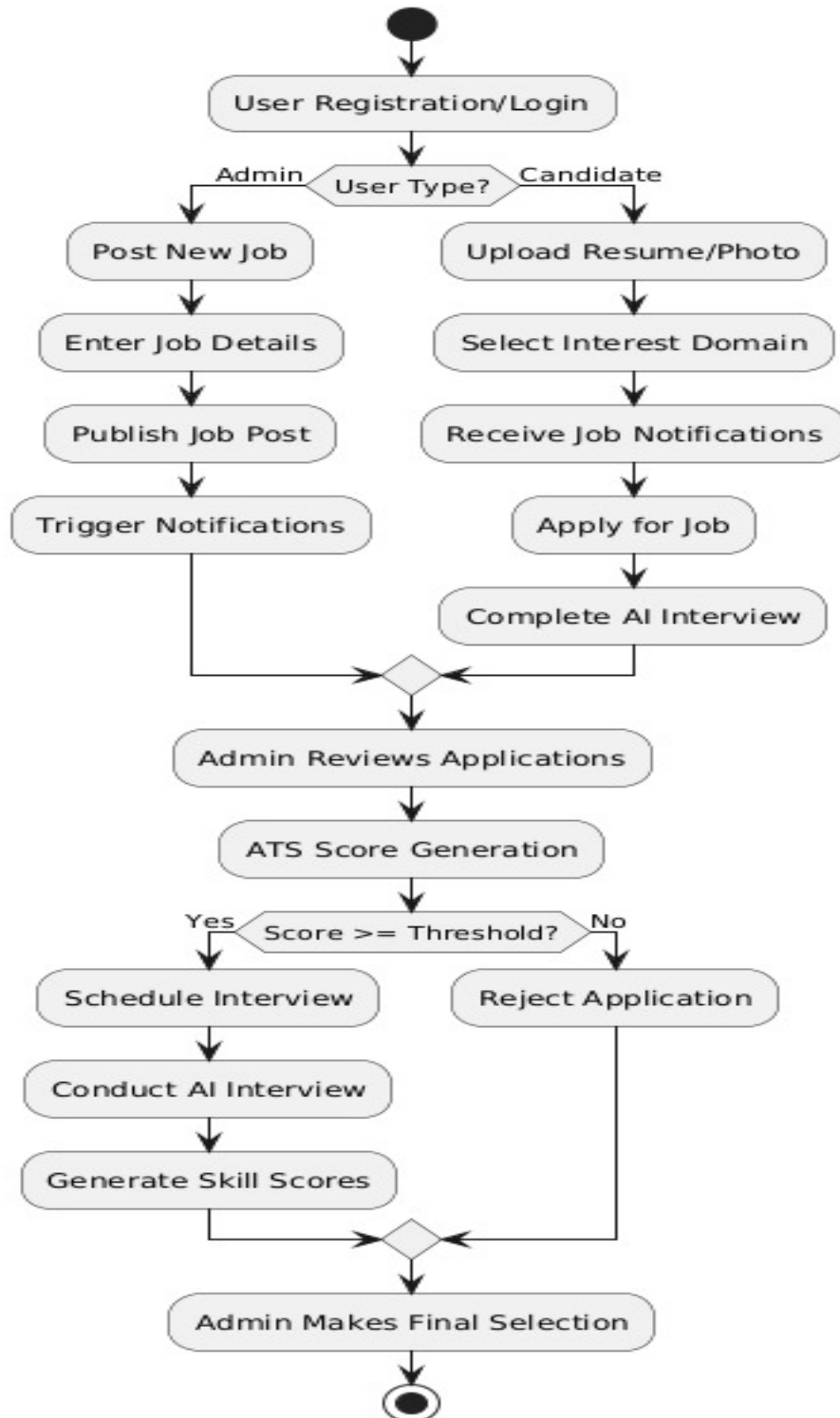


Fig 4.2.3 Activity Diagram



#### 4.2.4 SEQUENCE DIAGRAM

The sequence diagram shows the order of operations and the interaction between system components during a specific process, for example, a candidate attending an interview.

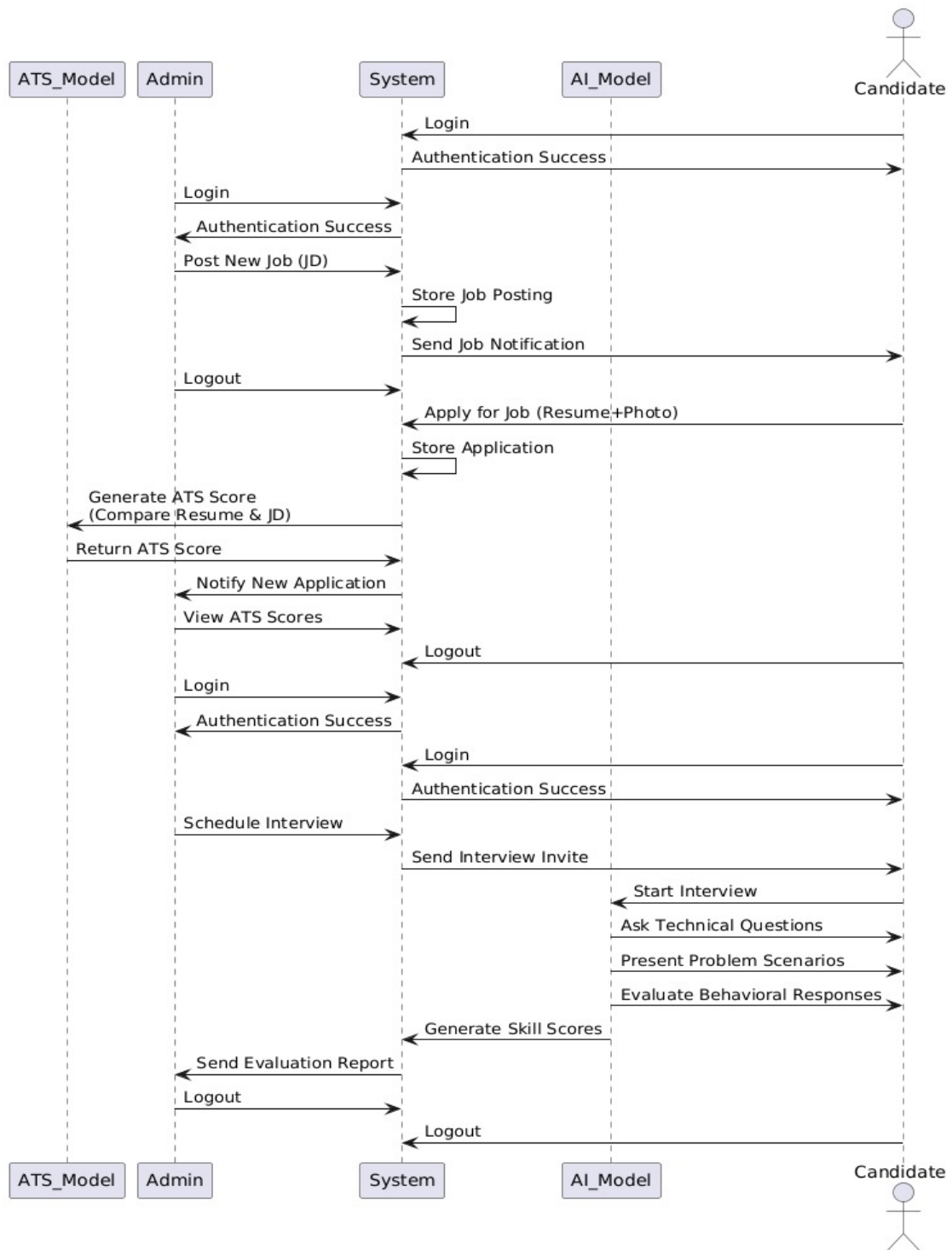


Fig 4.2.4 Sequence Diagram

## **4.3 IMPLEMENTATION**

The implementation phase transforms the system design into a fully functional software application. This phase involves writing the actual source code, integrating modules, setting up databases, and ensuring proper communication between the front-end and back-end components. The Interview Automation System was developed using a modular approach to ensure scalability, maintainability, and efficient performance.

### **4.3.1 METHODS AND CATEGORIZATION**

The Interview Automation System is designed with a modular and method-driven architecture to promote code reusability, clarity, and scalability. This section explains the core methods (functions) used in the system and how the system is logically categorized for ease of development and maintenance.

#### **A. Content analysis**

Content Analysis in the context of interview automation refers to the automatic understanding and evaluation of submitted answers or user interactions.

#### **Applications in the System:**

- **Text Analysis:** Analyzing subjective answers using NLP techniques like sentiment analysis, keyword extraction, and semantic scoring.
- **Keyword Matching:** Checking whether key technical terms or concepts are mentioned in the response.
- **Code Quality Evaluation:** In coding questions, the system can analyze logic, syntax correctness, and performance metrics.

#### **Benefits:**

- Reduces human effort in evaluating long subjective answers.
- Ensures fair and consistent grading across candidates.
- Can detect plagiarism using similarity checks.

## **B. Feature Engineering**

Feature Engineering involves extracting and selecting meaningful data points (features) from user inputs and system activity logs to train machine learning models.

### **Examples of Features:**

- Time taken per question
- Number of code compilations before success
- Length and complexity of answers
- Sentiment score of HR answers
- Keyboard/mouse activity (for behavior analysis)

### **Use in the System:**

- These features can be used to build a candidate performance profile.
- Helps in classifying candidates based on readiness levels (e.g., Beginner, Intermediate, Advanced).
- Can also assist in detecting suspicious behavior (e.g., copy-paste detection).

## **C. Deep Learning Integration**

Deep Learning techniques can significantly enhance the intelligence of the interview system.

Possible Applications:

### **1. Automatic Grading of Subjective Answers:**

- Using transformer-based models like BERT or GPT for semantic understanding.
- Model compares student answers with expected ones and predicts a score.

## 2. Speech & Video Analysis:

- Voice-based interviews can be analyzed for tone, confidence, and stress level using CNN/RNN models.
- Facial expression recognition for engagement detection (using deep CNNs).

## 3. Question Recommendation Systems:

- Based on candidate's previous answers and performance, recommend personalized follow-up questions.
- Uses Recurrent Neural Networks (RNNs) or Collaborative Filtering.

### D. System Categorization

Module	Description
User Management	Registration, login, and role-based dashboard
Interview Scheduling	Schedule slots for candidates and assign interviewers
Question Management	Add, edit, categorize questions (MCQ, coding, HR, etc.)
Interview Process	Conducts timed interviews with auto-save, navigation, and progress tracking
Answer Evaluation	Auto-evaluates objective answers; integrates AI models for subjective ones
Report Generation	Scorecards with AI-generated insights and feedback
AI (Optional)	Module NLP & deep learning models for intelligent grading and prediction

### 4.3.2 PREREQUISITES

Before implementing the Interview Automation System, certain prerequisites need to be fulfilled in terms of hardware, software, technical knowledge, and frameworks/libraries. These ensure a smooth development, deployment, and execution process.

Developing and deploying an Interview Automation System involves a combination of technical, software, and infrastructure requirements. This section outlines the essential prerequisites needed to successfully build and maintain the application.

#### 1. Technical Knowledge

To contribute to the development of this project, the following technical skills are essential:

- **HTML, CSS, JavaScript:** Basic web technologies used for UI development.
- **React & Next.js:** Frameworks used for building the frontend with server-side rendering and routing capabilities.
- **Tailwind CSS:** A utility-first CSS framework used to style components efficiently.
- **TypeScript (optional but recommended):** Enhances code reliability through static typing.
- **RESTful APIs:** Understanding of HTTP methods (GET, POST, PUT, DELETE) to communicate with the backend.
- **Authentication/Authorization:** Knowledge of role-based access control and secure session handling (via Clerk.js).

#### 2. Software Tools and Packages

Before development begins, install and configure the following tools:

- **Node.js (v16 or above)** – Runtime environment for executing JavaScript server-side code.

- **NPM or Yarn** – Package managers used to install project dependencies.
- **Visual Studio Code (VS Code)** – Recommended code editor for development with useful extensions.
- **Git** – Version control system for code tracking and collaboration.
- **Supabase Account** – For managing the backend database, storage, and real-time updates.
- **Clerk.js Account** – For handling secure user authentication and role-based access.

### 3. Backend Setup

- **Supabase Configuration:**
  - Create a new project on Supabase.
  - Set up tables for users, questions, responses, and scores.
  - Configure Row-Level Security (RLS) to protect sensitive data.
- **Database Design:**
  - Design schemas to manage user roles, question banks, and answer storage.
  - Use SQL or Supabase's GUI for database management.

### 4. Authentication System

- **Clerk.js Setup:**
  - Register your application and obtain the frontend/backend API keys.
  - Configure sign-up/sign-in UI components.
  - Enable session-based access and role detection (admin vs. candidate).

## 5. Deployment Requirements

- **Vercel (Recommended for Next.js):** For deploying the web application seamlessly.
- **Netlify/Render (Alternatives):** Can also be used with minor configuration.
- **Environment Variables:** Securely store API keys in the deployment platform's settings.

## 6. Collaboration & Version Control

- Use **GitHub** to manage the source code repository.
- Follow **branching strategies** (e.g., main, dev, feature/\*) for teamwork.
- Enable GitHub Actions for continuous integration if needed.

## 4.4 SOURCE CODE

### proctoring/route.js

```
const rekognitionClient = new RekognitionClient({
  region: "us-east-1",
  credentials: {
    accessKeyId: process.env.AWS_ACCESS_KEY_ID,
    secretAccessKey:
process.env.AWS_SECRET_ACCESS_KEY,
  },
});

if (response.FaceMatches &&
response.FaceMatches.length > 0) {
  return new Response(JSON.stringify({
    matched: true,
    similarity:
response.FaceMatches[0].Similarity
  }), {
    status: 200,
    headers: { 'Content-Type':
'application/json' },
  });
} else {
  return new Response(JSON.stringify({
    matched: false
  }), {
    status: 200,
    headers: { 'Content-Type':
'application/json' },
  });
}
```



### Speech-to-text/route.js

```
const startListening = () => {
  const SpeechRecognition =
    window.SpeechRecognition ||
    window.webkitSpeechRecognition;
  if (!SpeechRecognition) return alert("Speech
  recognition not supported!");

  const recognition = new SpeechRecognition();
  recognition.continuous = false;
  recognition.interimResults = false;
  recognition.lang = "en-US";

  recognition.onstart = () =>
    setIsListening(true);
  recognition.onresult = (event) => {
    const transcript =
      event.results[0][0].transcript;
    setCurrentAnswer(transcript);
    handleAnswer(transcript);
  };
  recognition.onerror = (event) => {
    console.error("Speech recognition error:",
    event);
    setIsListening(false);
  };
  recognition.onend = () =>
    setIsListening(false);
  recognition.start();
  recognitionRef.current = recognition;
};
```

### email-update/route.js

```
const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: process.env.EMAIL_USER,
    pass: process.env.EMAIL_PASS,
  },
});
const mailOptions = {
  from: process.env.EMAIL_USER,
  to: email,
  subject: subject,
  text: textContent(name, job_title, Jobcompany),
  html: htmlContent(name, job_title, Jobcompany),
};
```

### generate-question/route.js

```
const QUESTION_TYPES = {
  TECHNICAL: 'technical',
  BEHAVIORAL: 'behavioral',
  PROBLEM_SOLVING: 'problem_solving',
  CLOSING: 'closing'
};

function createQuestionPrompt(data, count) {
  const questionTypes = Object.values(QUESTION_TYPES);
  // Cycle through question types (skipping CLOSING)
  const type = questionTypes[(count - 1) %
(questionTypes.length - 1)];
  const context = `Job: ${data.jobDescription}\nSkills:
${data.techStack}\nPrevious answers:
${JSON.stringify(data.conversationHistory.slice(-2))}`;

  return `Generate a ${type} question
about:\n${context}\nFormat: [${type.toUpperCase()}]
Question text`;
}
```

### evaluate-answer/route.js

```
async function evaluateAnswer(question, answer) {
  try {
    const evaluationPrompt = `Evaluate this interview
answer in valid JSON format:
{
  "technicalKnowledge": <0-10>,
  "communicationClarity": <0-10>,
  "problemSolving": <0-10>,
  "overall": <0-10>,
  "feedback": "<constructive feedback>"
}
Question: ${question}
Answer: ${answer}`;

    const response = await
fetch('https://api.openai.com/v1/chat/completions', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer
${process.env.OPENAI_API_KEY}`
  },
  body: JSON.stringify({
    model: 'custom-trained',
    messages: [{ role: 'user', content:
evaluationPrompt }],
  })
});
  };
```

## resume\_extractor\_with\_nlp.py

```
# Load Models
nlp_spacy = spacy.load("en_core_web_trf")
ner_pipeline = pipeline("ner", grouped_entities=True)

# Text Extraction
def extract_text_from_file(path):
    ext = os.path.splitext(path)[1].lower()
    if ext == ".pdf": return extract_pdf_text(path)
    elif ext == ".docx":
        doc = Document(path)
        return "\n".join([p.text for p in doc.paragraphs])
    else:
        with open(path, encoding="utf-8", errors="ignore")
as f:
    return f.read()

# Name Extraction
def extract_name(text):
    ents = [e['word'] for e in ner_pipeline(text[:500]) if
e['entity_group'] == 'PER']
    if ents: return ents[0]
    for line in text.splitlines():
        if line.strip(): return line.strip()
    return None

# Section Extraction
def extract_sections(text, headers):
    lines, header_map = text.splitlines(), {h.lower(): h for
h in headers}
    sections, current = {h: [] for h in headers}, None
    for line in lines:
        key = line.strip().rstrip(':').lower()
        if key in header_map: current = header_map[key];
    continue
    if current:
        if key in header_map: current = header_map[key];
    continue
    sections[current].append(line)
    return {h: ("\n".join(sections[h]).strip() or None) for
h in headers}

# Main Execution
uploaded = files.upload()
path = next(iter(uploaded.keys()))
text = extract_text_from_file(path)

fields = ["Education", "Experience", "Projects",
"Achievements", "Awards"]
sections = extract_sections(text, fields)
links = re.findall(r'https?:\/\/\S+', text) or None
```

## **5.RESULTS**

## Interview Automation

Rectangular Snip

# Welcome to Interview Automation

Upload your resume, answer questions, and receive a personalized interview report.

Start Your Interview

Login

Signup

## Interview Automation

Admin Registration

Candidate Registration

Add New Job

Jobs

My Applications

Reports

Schedule

Interview



Uchith Kumar

# Welcome to Interview Automation

Upload your resume, answer questions, and receive a personalized interview report.

Start Your Interview

## Interview Automation

Admin Registration

Candidate Registration

Add New Job

Jobs

My Applications

Reports

Schedule

Interview

 Uchith Kumar

### Admin Registration

Admin Name

Enter your name

Email

Enter your email

Company Name

Enter company name

Password

Enter your password

Register

### Candidate Registration

Candidate Name

Enter your name

Email

Enter your email

Password

Enter your password

Company

Enter your company

Upload Resume

No file chosen

Upload Photo

No file chosen

Skills

Enter your skills

Highest Education

Enter highest education

College Name

Enter college name

Address

Enter your address

Interest Domains

Enter your interest domains

Register

## Interview Automation

Admin Registration

Candidate Registration

Add New Job

Jobs

My Applications

Reports

Schedule

Interview

 Uchith Kumar

## Add New Job

Job Title

 Enter job title

Company Name

 Enter company name

Domain

 Enter domain

Last Date to Apply

 mm/dd/yyyy 

Job Description

Normal  Normal  **B** *I* U      

Enter job description...

Eligibility Criteria

Normal  Normal  **B** *I* U      

Enter eligibility criteria...

Add Job

## Interview Automation

Admin Registration

Candidate Registration

Add New Job

Jobs

My Applications

Reports

Schedule

Interview

 Uchith Kumar

## Job Listings

All Companies 

Show Active Only: ☒

### Software Development Engineer Intern

Company: HackerRank

Last Date: April 9, 2025

[Get Details](#)

[Apply](#)

### Software Engineer - Frontend

Company: Tech Innovators Pvt. Ltd.

Last Date: April 18, 2025

[Get Details](#)

[Apply](#)

## Job Listings

All Companies

### Software Development Engineer Intern

Company: HackerRank

Last Date: April 9, 2025

[Get Details](#)

#### Take Photo & Upload Resume

Upload Resume:

[Choose File](#) Uchith\_Resume.pdf

[View](#)

Upload Photo:



[Start Camera](#)

[Capture Photo](#)

[Upload](#)

Active

[Apply](#)

## Job Applications

☐ Select All ☐ Reject All

Job ID

All Job IDs

College

Filter by College

Education

Filter by Education

Sort by Date

Select Date Order

Sort by Score

Select Score Order

Selection Status

All Candidates

Score Range

From

To

S. No	Job ID	Candidate ID	Candidate Name	Highest Education	College Name	Applied On	Details	Score	Select/Reject	
1	1	3	Uchith	CDE	GPREC	11/10/2024	<a href="#">View details</a>	20	<a href="#">Select</a>	<a href="#">Reject</a>
2	7	3	Uchith	CDE	GPREC	11/10/2024	<a href="#">View details</a>	75	<a href="#">Select</a>	<a href="#">Reject</a>



1 of 467



8:02PM (2 minutes ago) ☆ 😊 ↩ ⋮



## My Interviews

### SDE

Company: Google

Start Time: 3/28/2025, 4:55:26 PM

Time Remaining: **Interview expired**

Start Interview

Request Reschedule

### Software Development Engineer Intern

Company: HackerRank

Start Time: 3/28/2025, 4:55:26 PM

Time Remaining: **Interview expired**

Start Interview

Request Reschedule

### Software Engineer - Frontend

Company: Tech Innovators Pvt. Ltd.

Start Time: 3/28/2025, 4:55:26 PM

Time Remaining: **Interview expired**

Start Interview

Request Reschedule

## Technical Interview



Match: 100.00%

Question 2 of 5

Interviewer

Behavioral

Let's start with your introduction. Please tell us about yourself and your technical experience.

Your Answer

Score: 7/10

my name is bhuchal Kumar I am studying G Pulla Reddy Engineering College and I have skills in Java Python and data structures and also done project in full stack as a Avishkar event

Great to hear about your technical skills and project experience. It would be beneficial to work on improving communication clarity to effectively convey your knowledge and experiences.

Interviewer

Behavioral

Can you provide an example of a challenging technical problem you faced during your project in full stack development, and how you solved it?

Press to Answer

my name is bhuchal Kumar I am studying G Pulla Reddy Engineering College and I have skills in Java

## Interview Automation

Admin Registration

Candidate Registration

Add New Job

Jobs

My Applications

Reports

Schedule

Interview

 Uchith Kumar

## Interview Reports

Job ID	Overall Score Range	Avg Technical Score	Avg Behavioral Score
<input type="text" value="All Jobs"/>	<input type="text" value="Min"/> <input type="text" value="Max"/>	<input type="text" value="Min"/> <input type="text" value="Max"/>	<input type="text" value="Min"/> <input type="text" value="Max"/>

S. No	Application ID	Job ID / Title	Candidate ID / Name	Avg Scores (T/B/P)	View Report	Action
1	77	7 Software Development Engineer Intern	3 Uchith	1.5 / 1.3 / 1.3	<a href="#">View Report</a>	<a href="#">Select</a> <a href="#">Reject</a>
2	77	7 Software Development Engineer Intern	3 Uchith	3.2 / 3.4 / 3.8	<a href="#">View Report</a>	<a href="#">Select</a> <a href="#">Reject</a>

[Notify Selected/Rejected Candidates](#)

## **6.CONCLUSION**

## 6.1 CONCLUSION

The Interview Automation System successfully demonstrates how recruitment processes can be modernized using web technologies and automation tools. It offers a structured and efficient alternative to traditional interviews by reducing human intervention, minimizing biases, and improving the consistency of candidate evaluation.

By using Next.js for frontend development, Clerk.js for secure authentication, and Supabase for database management, the system ensures a smooth user experience and secure data handling. The admin dashboard facilitates real-time monitoring and decision-making, while candidates benefit from a responsive and intuitive interview interface.

This project not only addresses the challenges of scalability, scheduling, and evaluation in conventional hiring methods but also opens the door for integrating advanced technologies for smarter and faster recruitment decisions. It stands as a robust foundation for organizations aiming to digitize and streamline their hiring workflow.

## 6.2 FUTURE ENHANCEMENT

To increase the utility, efficiency, and intelligence of the Interview Automation System, several future enhancements can be implemented. These upgrades will transform the current system into a more adaptive, intelligent, and enterprise-grade recruitment platform.

### 1. AI-Based Evaluation

By integrating **Natural Language Processing (NLP)** and **Machine Learning (ML)** models, subjective candidate responses can be analyzed for content relevance, tone, confidence, and coherence. This eliminates manual evaluation effort and ensures more unbiased and standardized scoring mechanisms. Over time, the models can learn from past assessments to improve accuracy.

### 2. Voice and Facial Expression Analysis

Incorporating **computer vision** and **audio processing** technologies can help assess candidates based on facial expressions, eye movement, tone of voice, hesitation, and

stress levels. This would provide **emotional and behavioral insights** that are crucial for roles requiring communication or leadership skills.

### **3. Real-Time Chatbot Assistance**

A built-in AI chatbot can offer real-time assistance to candidates during interviews. It can answer FAQs, provide next steps, give guidance on technical issues, or help with navigation. This reduces confusion and increases candidate satisfaction, especially for first-time users.

### **4. Integration with HR Systems**

For large-scale deployments, integration with **Applicant Tracking Systems (ATS)** and **Human Resource Management Systems (HRMS)** will ensure seamless data synchronization. It would allow HR teams to directly import/export candidate data, schedule interviews, and generate reports without switching platforms.

### **5. Dynamic Question Generation**

Enhance the system with an **AI-based question bank** that generates questions dynamically based on the candidate's role, skill level, and previous answers. This adaptive interview approach keeps candidates engaged and ensures a fair, customized assessment experience.

### **6. Multilingual Interface**

Supporting multiple languages will make the system more inclusive and applicable for multinational companies. Candidates can interact with the system in their preferred language, thereby improving accessibility and reducing language barriers.

### **7. Gamification Features**

Adding gamification elements like scores can make the platform more engaging for candidates. It can also reduce anxiety and promote a friendly, interactive experience during assessments.

## 8. Mobile Application Development

Developing a cross-platform mobile app can make the system even more accessible. Candidates can attend interviews on-the-go, and admins can monitor and evaluate responses anytime. Push notifications can keep users updated about interview schedules and feedback.

## 9. Advanced Analytics and Reporting

Introduce dashboards with **interactive charts, performance trends, heat maps,** and **comparative analytics** to help HR professionals make informed hiring decisions. AI-generated summaries and prediction models can help in forecasting candidate success in a particular role.

## 10. Security and Compliance Enhancements

As the platform deals with personal and performance-related data, future versions must comply with **GDPR, ISO standards,** and include **multi-factor authentication, encryption at rest,** and **regular audits** for robust security.

## **REFERENCES**

1. **Daryanto, T., Ding, X., Wilhelm, L. T., Stil, S., Knutsen, K. M., & Rho, E. H. (2024).**  
*Conversate: Supporting Reflective Learning in Interview Practice Through Interactive Simulation and Dialogic Feedback.*  
<https://arxiv.org/abs/2410.05570>
2. **Xiao, Z., Zhou, M. X., Chen, W., Yang, H., & Chi, C. (2020).**  
*If I Hear You Correctly: Building and Evaluating Interview Chatbots with Active Listening Skills.*  
<https://arxiv.org/abs/2002.01862>
3. **Han, X., Zhou, M., Turner, M., & Yeh, T. (2021).**  
*Designing Effective Interview Chatbots: Automatic Chatbot Profiling and Design Suggestion Generation.*  
<https://arxiv.org/abs/2104.04842>
4. **Lyu, D., Lu, Y., He, J., Abrar, M. M., Xie, R., & Raiti, J. (2024).**  
*InsightPulse: An IoT-based System for User Experience Interview Analysis.*  
<https://arxiv.org/abs/2410.00036>
5. **Parfenova, A. (2024).**  
*Automating the Information Extraction from Semi-Structured Interview Transcripts.*  
<https://arxiv.org/abs/2403.04819>
6. **Sahijwani, H., Dhole, K., Purwar, A., Vasudevan, V., & Agichtein, E. (2023).**  
*Contextual Response Interpretation for Automated Structured Interviews: A Case Study in Market Research.*  
<https://arxiv.org/abs/2305.00577>
7. **Naim, I., Tanveer, M. I., Gildea, D., & Hoque, M. E. (2015).**  
*Automated Analysis and Prediction of Job Interview Performance.*  
<https://arxiv.org/abs/1504.03425>
8. **Unknown Author (n.d.).**  
*Interview Automation through Speech Recognition and Semantic Matching.*  
<https://www.researchgate.net/publication/384892821>
9. **isyedahmed531.**  
*Automated Interview Evaluation System (GitHub Project).*  
<https://github.com/isyedahmed531/Automated-Interview-project>
10. **yangshun.**  
*Tech Interview Handbook (GitHub Project).*  
<https://github.com/yangshun/tech-interview-handbook>