

AI-Based Interviewer

“Ace Hire”



Sec1_T1_P3

Faculty: Arnold S

Project Members:

Amrutheshwari V (2023UG000115)

Gajendiran A (2023UG000154)

Lavanya H S (2023UG000157)

Priyanka Bhatt (2023UG000127)

Sairaj Bhise (2023UG000117)

Tiya Rose Pulikunnel (2023UG000135)

Table of Content

1. Introduction.....	6
2. Problem Statement.....	9
2.1 Detailed features of modern job preparation platform.....	10
2.2 Why does the modern job preparation matter?.....	12
3. Project Objectives.....	13
4.Solution Overview.....	13
5. Technical Stack.....	17
5.1 Backend Technologies.....	17
5.1.1 Python.....	17
5.1.2 Streamlit.....	17
5.1.3 OpenAI API.....	18
5.1.4 Natural Language Processing (NLP).....	18
5.2 Data Processing Libraries.....	18
5.2.1 PDF Processing.....	19
5.2.2 Audio Processing.....	19
5.2.3 Data Analysis.....	19
5.3 Frontend Components.....	19
5.3.1 Plotly.....	19
5.3.2 Streamlit UI Elements.....	20
5.3.3 CSS/HTML.....	20
5.4 Storage and State Management.....	20
5.4.1 Session State.....	21
5.4.2 File System.....	21
5.5 Environment Management.....	21
5.5.1 dotenv.....	21
5.5.2 Package Management.....	22
5.6 Security and Authentication.....	22
5.6.1 API Authentication.....	22
5.6.2 Environment Variable Management.....	22
5.7 Integration and Output Generation.....	22
5.7.1 Report Generation.....	23
5.8 Modern Python-based Web Application Architecture.....	23
6. System Architecture.....	23
6.1 Architectural Overview.....	24
6.2 Streamlit Web Application Layer.....	24

6.2.1 Key Components.....	24
6.2.2 Technical Implementation.....	25
6.3 Service Layer.....	25
6.3.1 Key Components.....	26
6.3.2 Technical Implementation.....	26
6.4 External Services.....	26
6.4.1 Key Components.....	26
6.4.2 Technical Implementation.....	27
6.5 Storage Layer.....	27
6.5.1 Key Components.....	28
6.5.2 Technical Rationale.....	28
6.6 Data Flow.....	28
6.6.1 Input Processing Flow.....	28
6.6.2 Interview Question Generation Flow.....	29
6.6.3 Interview Response Flow.....	29
6.6.4 Report Generation Flow.....	29
6.7 Architecture Benefits.....	30
6.8 Modern AI-Powered Architecture.....	30
7. Implementation Details.....	31
7.1 Steps Followed to Build the Solution:.....	31
7.2 Innovations or Unique Approaches Used.....	33
7.3 Challenges faced during implementation.....	36
8. AI/LLM Integration.....	37
8.1 OpenAI GPT Integration.....	37
8.2 Natural Language Processing (NLP).....	39
8.3 LLM-Powered Interview Question Generation.....	39
8.4 Response Analysis and Feedback Generation.....	40
8.5 Comprehensive Report Generation.....	41
8.6 Speech and Language Support.....	41
8.7 AI Implementation Architecture.....	42
8.8 User Experience Integration.....	43
9. UI/UX Design.....	44
9.1 Design Philosophy.....	44
9.2 Interface Structure.....	44
9.3 Page-Specific Design Considerations.....	45
9.3.1 Resume Upload Page.....	45
9.3.2 Resume-Job Match Page.....	45

9.3.3 Question Generation Page.....	46
9.3.4 Interview Session Page.....	46
9.3.5 Final Report Page.....	46
9.4 Visual Design Elements.....	47
9.4.1 Color System.....	47
9.4.2 Typography.....	47
9.4.3 Interactive Elements.....	47
9.5 Accessibility Considerations.....	48
9.6 Responsive Design.....	48
9.7 User Testing Insights.....	49
9.8 Future UI/UX Enhancements.....	49
9.9 Conclusion.....	50
10. Code Execution Guide.....	50
10.1 System Requirements.....	50
10.2 Environment Setup.....	52
10.2.1 Python Environment.....	52
10.2.2 Installing Dependencies.....	52
10.2.3 Additional Setup.....	52
10.3 API Configuration.....	52
10.3.1 Creating the .env File.....	52
10.3.2 Obtaining API Keys.....	53
10.4 Running the Application.....	53
10.4.1 Starting the Streamlit Server.....	53
10.4.2 Alternative Launch Methods.....	53
10.5 Application Navigation.....	53
10.6 Troubleshooting.....	54
10.6.1 Common Issues.....	54
10.6.2 Logging.....	54
10.7 Performance Optimization.....	54
10.8 Security Considerations.....	55
10.9 Deployment Options.....	55
10.10 Extending the Application.....	56
11. Results & Metrics.....	56
11.1 Interview Preparation & Practice Tools.....	56
11.2 AI Based Feedback & Performance Analysis.....	57
11.3 Skill Enhancement & Learning Suggestion.....	57
11.4 Performance Tracking & Metrics.....	57
12. Demo Video Link.....	58
13. Individual Contributions.....	58

14. Solution Impact.....	63
14.1 Efficiency and Time Savings.....	63
14.2 Improved Accuracy and Fairness.....	63
14.3 Scalability for Organizations.....	64
14.4 Enhanced Candidate Experience.....	64
14.5 Cost Reduction.....	64
14.6 Data-Driven Insights.....	64
14.7 Innovation and Competitive Edge.....	65
14.8 Global Accessibility.....	65
14.9 Impact on Job seekers.....	65
15. Future Enhancements.....	66
15.1 Technical Constraints and Solutions.....	66
15.2 Feature Enhancement Roadmap.....	67
15.2.1 Short-term Enhancements.....	67
15.2.2 Medium-term Enhancements.....	68
15.2.3 Long-term Vision.....	68
15.3 Specialized Applications and User Segments.....	69
15.4 Business Model Evolution.....	70
16. Conclusion.....	71
17. Github Repository Links.....	72
18. References.....	72

1. Introduction

The Evolution of Interviews: Traditional to Online

The interview process is a cornerstone of the hiring process, aimed at identifying the best candidates for a role. Traditionally, the process followed a structured format, comprising stages such as resume screening, initial interviews, and final decision-making. While effective, the traditional approach often posed challenges, particularly in terms of time, cost, and bias.

Stages of Traditional Interviews:

- **Resume Screening:** Recruiters manually review resumes, filtering candidates based on qualifications, experience, and skills. According to studies, recruiters spend an average of 6 seconds per resume during initial screenings [1].
- **Initial Interviews:** These are typically conducted over the phone or in person to evaluate communication skills, interest in the role, and cultural fit.
- **Technical/Behavioral Interviews:** In-depth evaluations through one-on-one or panel interviews, focusing on problem-solving skills, technical knowledge, and situational responses.
- **Final Interviews and Decision:** Meetings with top management or HR before extending job offers.

Challenges of Traditional Interviews:

- Despite their established nature, traditional interviews come with inherent challenges:
- Time Consumption: Scheduling interviews and reviewing candidates manually takes significant time. According to Glassdoor, the average time to hire for U.S. companies is 23.8 days [2].
- Subjectivity and Bias: Hiring managers may unconsciously favor candidates based on factors unrelated to job performance, such as appearance or shared interests.
- Resource Intensity: In-person interviews often require travel arrangements, logistical planning, and dedicated meeting spaces, adding to the cost.
- Scalability Issues: High volumes of applications can overwhelm recruiters, particularly for roles with broad appeal.

Impact of COVID-19 on the Hiring Process

The COVID-19 pandemic disrupted traditional hiring processes, necessitating rapid adaptation to remote methods. Lockdowns and travel restrictions made in-person interviews unsafe, forcing organizations to rethink their strategies. Online interviews became a vital solution, enabling businesses to maintain hiring continuity while adhering to health and safety protocols.

Statistical Insights:

- Surge in Online Interviews: According to LinkedIn's 2020 Global Talent Trends report, virtual hiring processes increased by 70% during the pandemic [3].

- Adoption of Video Conferencing Tools: Platforms like Zoom and Microsoft Teams saw usage spikes: Zoom's daily meeting participants grew from 10 million in December 2019 to over 300 million in April 2020 [4].
- Microsoft Teams recorded 75 million daily active users in April 2020, up from 20 million six months earlier [5].
- Virtual Hiring Saves Time and Costs: A Gartner survey revealed that 86% of organizations conducted virtual interviews during the pandemic, with 55% of HR leaders stating they plan to incorporate remote interviewing permanently[6].

Advantages of Online Interviews

- Online interviews offer numerous benefits, some of which extend beyond pandemic-related needs:
- Cost Efficiency: Companies save on travel, accommodation, and administrative expenses. A study by Deloitte found that virtual interviews can reduce recruitment costs by up to 25% [7].
- Time Savings: Automated scheduling tools and video platforms reduce the time spent coordinating interviews. Recruiters reported a 50% faster time-to-hire with online processes [8].
- Increased Reach: Remote interviews allow organizations to tap into global talent pools, expanding access to diverse candidates.
- Flexibility: Candidates and recruiters can connect from different time zones, making scheduling easier.

- **Data-Driven Insights:** AI-powered tools analyze interview recordings to assess communication skills, tone, and behavioral attributes objectively.
- **Online Interviews: A Post-Pandemic Standard**

The widespread adoption of online interviews during the pandemic demonstrated their efficacy, prompting many organizations to retain this model. For instance:

Unilever uses AI-driven tools like HireVue to conduct initial candidate screenings. The platform evaluates facial expressions, tone, and word choice, providing recruiters with data-backed insights. Amazon shifted most of its recruitment process online, with virtual interviews and onboarding becoming the norm for new hires.

The shift to online interviews, driven by the COVID-19 pandemic, was initially a response to unprecedented challenges but has since evolved into a preferred standard for many organizations. By reducing costs, saving time, and increasing efficiency, virtual interviews have transformed hiring processes. As technology advances, the future of interviews will likely incorporate even more AI-driven innovations, ensuring that assessments remain fair, objective, and effective

2. Problem Statement

In an increasingly competitive job market, job seekers face mounting challenges in preparing for the hiring process. From crafting resumes that capture attention to mastering interviews and identifying skill gaps, the process can feel overwhelming. Traditional methods such as manual resume reviews and in-person mock interviews are often inefficient, expensive, and inaccessible to many individuals, limiting opportunities for career growth. Modern job preparation tools have

emerged to address these challenges, offering innovative solutions designed to equip candidates with the skills and confidence needed to succeed.

2.1 Detailed Features of Modern Job Preparation Platforms:

Resume Optimization:

- A well-crafted resume is the foundation of a successful job application. Advanced preparation platforms provide
- Guidance for Customization: Tools to tailor resumes to specific job descriptions by analyzing keywords, skill requirements, and role expectations.
- Gap Identification: Insights into missing qualifications or experiences that could strengthen the resume.
- Professional Formatting: Templates and design options to ensure resumes are polished and visually appealing.

These features enable job seekers to create documents that not only stand out to recruiters but also align closely with the demands of the roles they are targeting.

Interview Preparation and Practice Tools:

- Successful interviews require confidence, technical expertise, and strong communication skills. Job preparation platforms provide:
- Mock Interviews: These are Realistic simulations of job interviews, often designed to reflect the specific requirements of different industries or roles.
- Behavioral and Technical Question Sets: Questions tailored to test soft skills (e.g., teamwork, problem-solving) and job-specific technical expertise.

- Practice Modes: Options for both video-based and audio-only practice sessions to help candidates become comfortable in various interview settings.

These tools ensure that candidates are well-prepared, reducing anxiety and enabling them to present their best selves during real interviews.

Personalized Feedback and Improvement Reports:

Understanding areas of strength and improvement is crucial for growth. Advanced preparation platforms offer:

- Detailed Feedback: Constructive insights on communication style, body language, and clarity of responses.
- Skill-Based Recommendations: Suggestions for courses, certifications, or practices to close skill gaps and enhance readiness.
- Progress Tracking: Metrics to monitor improvement over time, helping candidates measure their growth and refine their strategies.

Comprehensive Accessibility and Affordability:

The cost of traditional job preparation methods can be a barrier for many candidates. Modern platforms address this by:

- User-Friendly Interfaces: Easy-to-navigate systems accessible to all, regardless of technical proficiency.
- Flexible Pricing Models: Options that make preparation tools affordable for students, recent graduates, and professionals at all career stages.

- On-Demand Resources: Availability of materials and practice sessions anytime, providing flexibility for users with busy schedules.

Confidence-Building Resources:

Job seekers often struggle with self-doubt, which can negatively impact performance.

Preparation platforms provide:

- Behavioral Coaching: Tips and techniques to build confidence, manage stress, and improve presentation skills.
- Practice Without Judgment: Safe spaces for candidates to practice repeatedly without fear of criticism.
- Positive Reinforcement: Encouraging feedback to boost morale and motivate further improvement.

2.2 Why Does Modern Job Preparation Matter?

With 75% of job seekers struggling to tailor their resumes to job descriptions and many lacking access to effective preparation tools, the importance of accessible, innovative solutions cannot be overstated. These platforms empower individuals by leveling the playing field, ensuring everyone, from recent graduates to seasoned professionals, has the opportunity to present their skills effectively and compete for their desired roles.

By combining resume optimization, interview readiness tools, detailed feedback, and confidence-building resources, modern job preparation platforms transform the way candidates approach their career journeys. These tools not only enhance a candidate's chances of securing employment but also instill lasting skills that contribute to long-term career success.

3. Project Objectives

The primary objective of the AI-powered Resume-Based Interview System is to automate the initial interview process by leveraging Artificial Intelligence, Natural Language Processing (NLP), and Speech Analysis to streamline candidate evaluation.

This system aims to:

- Analyze candidate resumes to extract key details such as skills, experience, education, and qualifications.
- Generate customized interview questions dynamically, tailored to the candidate's expertise, industry, and role-specific requirements.
- Evaluate candidate responses using AI-driven scoring models that assess technical proficiency, behavioral competencies, and communication skills.
- In future, this provides recruiters with an automated ranking system, reducing manual effort and ensuring objective, data-driven decision-making.

By integrating NLP for resume parsing, AI for question generation, and Speech Analysis for spoken responses, the system enhances the efficiency and accuracy of talent assessment. In the long run this reduces hiring time, improves candidate experience, and ensures better alignment between job requirements and candidate capabilities.

4. Solution overview

Ace Hire – An AI-Powered Interview System: Ace Hire is an innovative AI-driven interview platform designed to streamline and enhance the hiring process by leveraging cutting-edge LLM

technology through the Gemini API. The system intelligently analyzes job descriptions and candidate resumes to generate customized interview questions, assess responses, and provide detailed performance reports. By automating the initial interview stage, as in future Ace Hire can help companies save time while offering candidates a seamless and fair evaluation experience.

The homepage of Ace Hire is designed to be visually appealing and highly functional. It prominently displays the project name, “Ace Hire – AI-Powered Interview System,” along with a list of advantages that highlight the key benefits of using AI for interviews. To the right side of the page, a resume image provides a visual representation of the platform’s purpose, while the left side features the advantages of Ace Hire, presented with engaging emojis to make the content more interactive. At the top-right corner, users can easily access the login and register buttons, ensuring a smooth onboarding experience. Below the advantages section, a “Start Practicing” button allows users to engage with the AI interviewer before signing up, giving them a preview of the system’s capabilities.

Navigation across the platform is simplified with a sidebar that remains accessible throughout the user experience. The sidebar contains essential sections such as "Home," where users can access the main dashboard, "Your Profile," which stores user details and preferences, "Your Interviews," where candidates can track their completed and ongoing interview sessions, and "Your Reports," which contains past performance analysis and feedback. This intuitive structure ensures that users can move through different sections effortlessly without confusion.

Ace Hire requires users to complete a registration process before accessing full interview functionalities. A key feature of the registration page is its integration with a Terms and Conditions agreement, which is linked to a Google Docs document. To proceed with registration, users must check a box confirming that they have read and agreed to these terms, ensuring compliance and transparency. This step not only secures user data but also maintains a structured process that aligns with ethical and legal requirements.

The core functionality of Ace Hire revolves around its AI-powered interview process, which offers two distinct modes to cater to different user preferences. The first mode is the Live Interview, where users answer questions in real-time while the system records their responses. The AI processes these responses instantly and evaluates them based on key parameters. To maintain a structured interview flow, the next question is only displayed after the previous one has been answered. The second mode is the Upload Interview, where candidates record their responses separately and upload them as either video(future) or audio files(current). Here, too, the system ensures a step-by-step progression, preventing candidates from moving forward without submitting an answer to the current question. For video responses, the system extracts the audio component, which the Gemini API then analyzes for evaluation.

Ace Hire's AI-driven assessment is designed to evaluate candidates on both technical and communication skills. The system measures accuracy for technical competency by analyzing how well the candidate's response aligns with job-specific requirements. Additionally, it assesses the depth of knowledge by evaluating the level of detail, industry-specific insights, and the ability to provide relevant examples. Communication skills are also carefully analyzed, focusing

on clarity, coherence, and fluency of responses. The platform further evaluates confidence by examining vocal stability, speech pace, and tonal variations to determine how assured a candidate sounds.

To enhance its evaluation process, Ace Hire utilizes advanced speech and audio processing techniques. The AI system converts speech to text for content analysis, ensuring that responses are correctly transcribed before being analyzed by NLP models. Additionally, sentiment and emotion detection features help gauge a candidate's engagement and confidence levels, adding another layer of insight to the assessment. Background noise reduction is also incorporated to improve the accuracy of the AI's analysis, ensuring that environmental factors do not interfere with performance evaluations.

Upon completing an interview, candidates receive a comprehensive performance report detailing their strengths and areas for improvement. The report includes an overall performance score, a category-wise breakdown of their evaluation, and graphical representations of their results. Additionally, the report provides actionable feedback and recommendations tailored to help candidates enhance their interview and resume-building skills. To ensure accessibility, users can download these reports for future reference or share them with potential employers.

Looking ahead, Ace Hire aims to incorporate advanced features such as facial expression analysis for deeper confidence evaluation in future updates. AI-generated coaching tips based on previous interview responses will also be integrated to help candidates refine their performance

over time. Furthermore, the platform plans to establish partnerships with hiring platforms, allowing job seekers to connect with recruiters directly through Ace Hire's system.

In conclusion, Ace Hire is revolutionizing the interview process by providing an AI-powered, data-driven approach to candidate assessment. By leveraging the capabilities of LLM technology and advanced speech analysis, it ensures a fair, efficient, and insightful evaluation process. Ace Hire is not just an AI interviewer—it is a comprehensive solution that empowers candidates to improve their skills while helping employers make well-informed hiring decisions.

5. Technical Stack

5.1 Backend Technologies

The core backend technologies used in the application include:

5.1.1 Python

- The application is written in **Python**, which handles: Business logic, AI integration and Data processing.

5.1.2 Streamlit

- **Streamlit** serves as the web framework, enabling rapid development of data-focused web applications.
- Key Streamlit features utilized include:
 - **Interactive Widgets:** Buttons, text inputs, and file uploaders

- **Session State Management:** Preserves data between user interactions
- **Multi-page Navigation:** Sidebar-driven page transitions
- **UI Components:** Expanders, tabs, and other dynamic elements
- **Media Handling:** Audio playback support
- **WebRTC Integration:** Enables audio recording within the browser

5.1.3 OpenAI API

- The application integrates with OpenAI's GPT models to power the core AI functionality:
 - **API Calls:** Utilizes the GPT-4o model for various tasks
 - **Custom Prompt Engineering:** Carefully crafted prompts for different application features
 - **JSON Response Formatting:** Structures data output for downstream processing
 - **Alternative Authentication:** Allows authentication through GitHub tokens for enhanced security using Azure inference endpoints

5.1.4 Natural Language Processing (NLP)

The application uses multiple NLP techniques for text analysis and information extraction:

- **spaCy Library:** Handles text processing and entity extraction
- **Keyword Identification and Analysis:** Matches extracted resume data with job requirements

5.2 Data Processing Libraries

Several libraries handle data extraction, transformation, and analysis:

5.2.1 PDF Processing

To process resumes in PDF format, the application uses:

- **PDF Text Extraction:** Utilizes libraries such as PyPDF2 or pdfminer
- **Text Cleaning and Normalization:** Prepares extracted text for analysis

5.2.2 Audio Processing

For audio-based interview responses, the system uses:

- **StreamlitWebRTC:** Enables audio recording directly via the browser
- **Temporary File Handling:** Stores audio files temporarily for analysis
- **Manual Transcription:** Supports transcription with multi-language capabilities

5.2.3 Data Analysis

For generating performance insights and visualizations:

- **NumPy:** Facilitates numerical operations and calculations
- **Plotly:** Produces interactive charts and graphs for data visualization
- **Custom Scoring Algorithms:** Evaluates resume-job matching and response analysis

5.3 Frontend Components

The frontend leverages several technologies and visualization tools to create an engaging user experience:

5.3.1 Plotly

- **Plotly** is used to create interactive visualizations, including:
 - **Bar Charts:** Performance comparison across key metrics
 - **Gauge Charts:** Displays match scores visually
 - **Interactive Graphs:** Provides a detailed breakdown of performance metrics

5.3.2 Streamlit UI Elements

The application uses a variety of Streamlit UI components for user interaction:

- **Metric Displays:** Show match scores and performance indicators
- **Progress Bars:** Provide visual feedback on interview progress
- **File Upload Widgets:** Handle resume and document uploads
- **Audio Playback Components:** Enables audio review and analysis
- **Dynamic Expandable Sections:** Enhances UI structure for a clean interface

5.3.3 CSS/HTML

Custom styling is applied using:

- **Base64-encoded Content:** Embeds resources securely
- **CSS Customizations:** Enhances UI aesthetics and improves the visual experience

5.4 Storage and State Management

Efficient storage and session state management ensure seamless application performance with **SQLite3** is used to:

1. **Store Data:** Save things like user info, interview details, and audio recordings.

2. **Retrieve Data:** Pull that info back when needed (e.g., showing a user's profile).

5.4.1 Session State

- **Streamlit Session State:** Used for temporary in-memory storage to manage:
 - Resume and job description data
 - Interview responses
 - Analysis results between pages
 - User navigation between app components

5.4.2 File System

- **Temporary File Storage:** Handles the following types of files:
 - Audio recordings
 - Uploaded documents
 - Generated reports

5.5 Environment Management

Managing environment variables and dependencies ensures secure and scalable application deployment:

5.5.1 dotenv

- **dotenv Library:** Manages environment variables such as:
 - API keys for OpenAI and Azure
 - GitHub token authentication

- Configuration settings for the application

5.5.2 Package Management

- Handles dynamic dependency management and environment configuration:
 - Verifies package versions
 - Installs required dependencies dynamically

5.6 Security and Authentication

Security mechanisms ensure that sensitive data and API credentials are well protected:

5.6.1 API Authentication

- **GitHub Token-based Authentication:**
 - Provides a secure and flexible alternative to traditional API keys
 - Enhances security using GitHub tokens for requests to Azure inference endpoints

5.6.2 Environment Variable Management

- **Secure Credential Storage:**
 - Environment variables are used to manage sensitive information, reducing the risk of exposure.

5.7 Integration and Output Generation

The system seamlessly integrates multiple components to generate and manage outputs:

5.7.1 Report Generation

- **Text-based Report Generation:** Uses StringIO to generate structured text reports
- **Downloadable Report Formatting:** Provides downloadable reports in structured formats
- **Comprehensive Analysis Compilation:** Combines performance, analysis, and recommendations in a single report

5.8 Modern Python-based Web Application Architecture

The technical stack represents a modern, cloud-based architecture built using:

- **Python:** Core business logic, AI integration, and data processing
- **Streamlit:** Interactive, data-driven web framework
- **OpenAI API:** AI capabilities via GPT-4o and prompt engineering
- **Plotly:** Data visualization and performance insights
- **Azure/GitHub Tokens:** Secure authentication and access control

The tech stack effectively combines AI, NLP, web technologies, and secure cloud-based architecture to create a comprehensive and scalable interview preparation platform.

6. System Architecture

The AI Interviewer application follows a modern integrated architecture that combines several technological components to deliver a seamless interview preparation experience. The system

architecture is designed for simplicity, efficiency, and effective user interaction while leveraging powerful AI capabilities.

6.1 Architectural Overview

The architecture follows a **layered approach** with four primary components:

1. **Streamlit Web Application** – Frontend and state management
2. **Service Layer** – Core business logic
3. **External Services** – AI and visualization components
4. **Storage Layer** – Integrated within the frontend

The architecture's distinctive feature is its integration of all components within a single application container, using Streamlit's reactive framework to eliminate the need for separate frontend and backend services.

6.2 Streamlit Web Application Layer

The Streamlit Web Application layer serves as both the **user interface** and the **orchestration component**, controlling the flow of the application and managing session state.

6.2.1 Key Components

- **Page Navigation System:**
 - Organizes the user flow into logical stages:
 - Upload Resume

- Resume-Job Match
 - Generate Questions
 - Interview Session
 - Final Report
- **User Interface Elements:**
 - Interactive widgets for file uploads, text entry, audio recording, and feedback display
 - **State Management:**
 - Maintains application state across user interactions
 - **Session Storage Controller:**
 - Handles ephemeral data persistence within the application lifecycle

6.2.2 Technical Implementation

Streamlit provides both the rendering engine and the application framework, allowing for rapid development of data-focused applications. Its **reactive execution model** enables real-time updates to the UI based on user interactions, eliminating the need for traditional client-server architecture.

6.3 Service Layer

The Service Layer contains the **core business logic** that processes inputs, orchestrates AI interactions, and prepares outputs for display.

6.3.1 Key Components

- **NLP Processor:** Extracts and analyzes structured information from resumes and job descriptions using SpaCy
- **OpenAI Integration:** Manages connections and prompt engineering for GPT-4o interactions
- **Gemini Integration:** Provides alternative AI model access for diversified capabilities
- **Analysis Processor:** Evaluates interview responses and generates performance metrics
- **PDF Processor:** Extracts text from resume documents and structures content
- **Audio Processor:** Manages recording and processing of spoken responses
- **UI Components:** Generates visualization elements and formatted displays

6.3.2 Technical Implementation

Each service is encapsulated in dedicated Python modules that expose clean interfaces to the main application. This **modular design** allows for independent development and testing of each component while maintaining clear separation of concerns.

6.4 External Services

The External Services layer encompasses third-party systems that provide specialized capabilities beyond the application's internal logic.

6.4.1 Key Components

- **OpenAI GPT-4o API:** Delivers advanced natural language processing for question generation, response analysis, and report creation
- **Gemini API:** Offers complementary AI capabilities with different strengths in language understanding and response generation
- **SpaCy NLP Library:** Provides fundamental natural language processing capabilities for text parsing and entity extraction
- **Plotly Visualization:** Generates interactive charts and visual representations of performance metrics
- **Streamlit Cloud:** Hosts the application and provides secure access to users through web browsers
- **ReportLab:** Generates professionally formatted PDF documents for downloadable reports

6.4.2 Technical Implementation

The application interfaces with these services through their respective APIs, abstracting the complexity of each integration within dedicated service modules. This design allows for potential replacement or enhancement of individual services without impacting the overall application structure.

6.5 Storage Layer

Unlike traditional architectures that separate storage into external databases, the Storage Layer in this application is **integrated** within the Streamlit Web Application layer, leveraging **in-memory structures** and **temporary file storage**.

6.5.1 Key Components

- **Session State:** Maintains application data throughout the user session using Streamlit's built-in state management
- **Temporary File Storage:** Handles uploaded documents and recorded audio files during processing

6.5.2 Technical Rationale

This integrated approach provides several advantages:

1. **Simplified Deployment:** Eliminates dependencies on external database services
2. **Enhanced Privacy:** User data remains within the application session and is not persisted long-term
3. **Reduced Complexity:** Removes the need for database connection management and query optimization
4. **Improved Performance:** Avoids network latency associated with external database calls

The ephemeral nature of storage aligns with the application's purpose as an interview preparation tool rather than a long-term data management system.

6.6 Data Flow

The data flow through the system follows a logical progression:

6.6.1 Input Processing Flow

1. User uploads resume (PDF)
2. PDF Processor extracts text
3. NLP Processor analyzes both resume and job description
4. Match calculation determines alignment between resume and job requirements

6.6.2 Interview Question Generation Flow

1. Resume and job analysis results feed into OpenAI prompt
2. GPT model generates contextually relevant questions
3. Questions displayed to user through Streamlit interface

6.6.3 Interview Response Flow

1. User records or types responses
2. Audio stored temporarily
3. User provides transcription → Sent to analysis processor
4. OpenAI analyzes responses based on various dimensions
5. Analysis results stored in session state and displayed

6.6.4 Report Generation Flow

1. Accumulated response analyses aggregated
2. Comprehensive report prompt sent to OpenAI
3. Structured JSON response parsed and formatted
4. Visual and textual report presented to user

6.7 Architecture Benefits

This integrated architecture delivers several key benefits:

Rapid Development: Streamlit's framework accelerates the development process, enabling fast iteration and deployment.

Simplified Maintenance: Consolidated components reduce operational complexity by minimizing the need for managing separate backend and frontend systems.

Enhanced User Experience: Reactive updates provide immediate feedback to the user, improving interaction efficiency.

Cost Efficiency: Minimal infrastructure requirements lower hosting costs, making the system suitable for cost-conscious deployment.

Scalability: External AI services handle the most computationally intensive tasks, allowing the application to scale effectively without additional infrastructure.

6.8 Modern AI-Powered Architecture

The system architecture represents a **modern approach** to AI-powered applications, integrating cloud AI services within a lightweight framework. This architecture:

- Minimizes operational complexity by consolidating frontend and backend layers
- Provides a modular and extensible foundation for future enhancements
- Ensures a seamless, interactive user experience through real-time data processing

The design effectively combines NLP, AI, and visualization components to create a powerful and intuitive interview preparation platform.

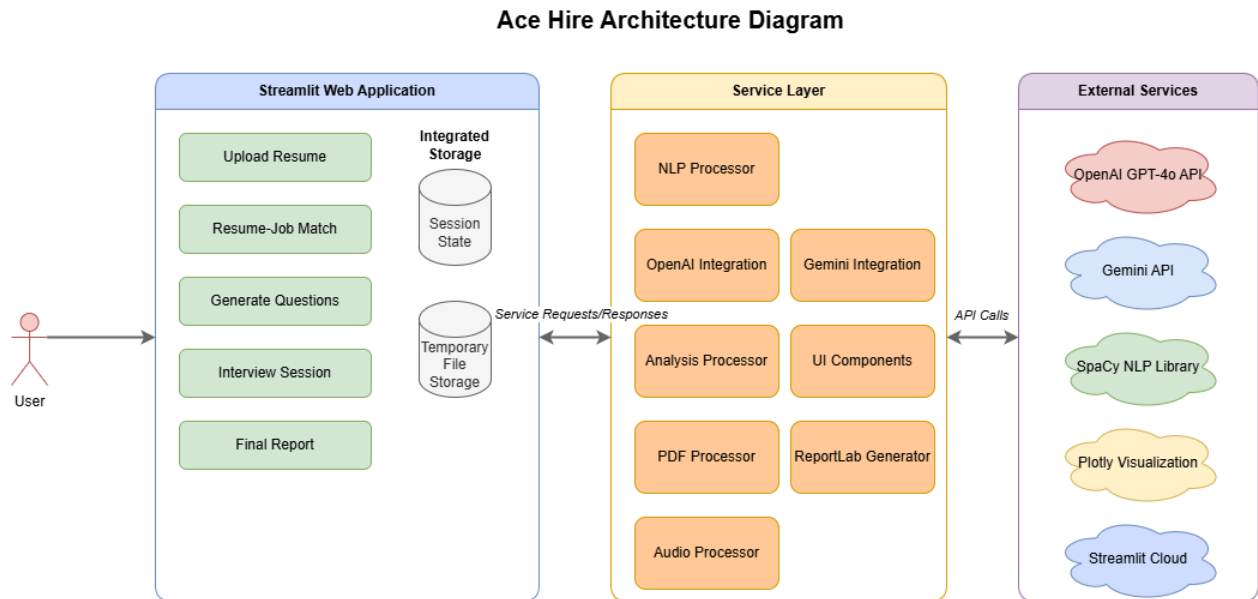


Figure 1

7. Implementation Details

7.1 Steps Followed to Build the Solution:

1. Requirement Analysis & Planning

- Identified key features and functionalities of Ace Hire.
- Researched AI-based interview assessment methodologies.
- Defined technical stack and architecture for scalability and security.

2. Frontend Development

- Designed UI/UX using Streamlit for a clean and interactive experience.

- Implemented navigation structure (Sidebar, Home, Profile, Interviews, Reports).
- Developed a user authentication system (Login/Registration).

3. Backend Development

- Implemented FastAPI/Flask for API communication.
- Integrated database (PostgreSQL/MySQL) using SQLAlchemy.
- Developed session tracking and data storage mechanisms.

4. AI Integration

- Utilized OpenAI/Gemini API for NLP-based question generation and response evaluation.
- Implemented speech-to-text conversion for audio/video responses.
- Integrated sentiment analysis and confidence assessment.

5. Resume Parsing & Question Generation

- Implemented PyMuPDF (fitz) to extract text from resumes.
- Developed NLP-based question generation tailored to candidates' experience and job descriptions.

6. Interview Process Implementation

- Developed real-time interview mode with dynamic question presentation.
- Implemented an upload-based interview mode with structured response handling.
- Ensured smooth progression by requiring responses before displaying the next question.

7. AI Evaluation & Scoring Mechanism

- Designed assessment models for technical accuracy, knowledge depth, and communication skills.

- Integrated voice analysis for clarity, coherence, and confidence measurement.
- Implemented AI-generated feedback and performance scoring.

8. Report Generation & User Insights

- Developed graphical reports with performance breakdowns.
- Implemented downloadable reports for users.
- Included recommendations for skill improvement based on AI analysis.

9. Security & Deployment

- Implemented authentication and data encryption.
- Secured API keys and tokens using environment variables.
- Deployed the system on cloud infrastructure for scalability.

7.2 Innovations or Unique Approaches Used

1. AI-Powered Resume Analysis & Personalized Question Generation

Implements a sophisticated resume analysis and question generation system that creates personalized interview experiences. At its core, the system uses a hybrid approach combining traditional NLP techniques with advanced LLM capabilities. For resume analysis, the application extracts text from PDF documents and parses it into logical sections using pattern recognition. It then employs SpaCy with PhraseMatcher to identify skills, while regex-based extraction captures experience and education details. In parallel, it analyzes job descriptions to extract required skills, preferred qualifications, and experience requirements. These analyses are compared to calculate match scores and identify qualification gaps. The personalized question generation leverages OpenAI's

GPT-4o model with carefully crafted prompts that incorporate the job title, description, resume content, and match analysis results. With a temperature setting of 0.7, the system generates a balanced mix of technical and behavioral questions targeted to the candidate's specific profile. The implementation optimizes for both performance and cost by truncating inputs while maintaining enough context for relevant question generation. This creates a tailored interview preparation experience that highlights candidate strengths while probing potential areas for improvement.

2. **Dual-Mode Interview Process (Live & Upload)**

- Candidates can choose between **real-time (live)** interviews or **pre-recorded (upload-based)** responses.
- Ensures flexibility for users, allowing them to practice at their convenience while maintaining a structured question flow.

3. **AI-Driven Multi-Factor Candidate Evaluation**

- Evaluates candidates across multiple dimensions:
 - **Technical Skills** → Accuracy and depth of knowledge
 - **Communication Skills** → Clarity, coherence, fluency
 - **Confidence** → Vocal stability, tone variations, pace of speech
 - **Engagement & Sentiment Analysis** → Emotion detection for deeper insights
- Uses a combination of **speech processing, NLP, and AI scoring models** to ensure a holistic assessment.

4. **Speech-to-Text & Audio-Based AI Analysis**

- Converts audio responses to text for **NLP-driven content evaluation** while simultaneously analyzing vocal characteristics for **confidence and engagement assessment**.
- Uses **background noise reduction** to improve evaluation accuracy, ensuring fair assessments even in non-ideal recording environments.

5. **AI-Generated Performance Reports with Actionable Feedback**

- Candidates receive **detailed, AI-powered reports** breaking down their performance in different categories.
- Provides **actionable insights and recommendations** to help users improve interview skills over time.
- Reports include **graphical representations** for better visualization of strengths and areas needing improvement.

6. **Secure & Scalable API Authentication via GitHub Tokens**

- Unlike traditional API key-based authentication, Ace Hire uses **GitHub Token API authentication** for better security and control.
- Reduces exposure of sensitive credentials while enhancing integration with GitHub-based workflows.

7. **Adaptive Question Flow**

- The **next question appears only after answering the previous one**, ensuring structured progression and preventing candidates from skipping ahead.
- Keeps candidates engaged and prevents them from “cherry-picking” questions.

8. **Future-Ready Architecture for Expansion**

- Built with **modular AI components**, allowing easy integration of **facial expression analysis, advanced voice emotion detection, and recruiter connections** in future updates.
- The database is structured to support **scalability**, enabling integration with third-party hiring platforms

7.3 Challenges faced during implementation:

To address the difficulty of connecting different pages, I referred to project documentation and Python/Streamlit resources to establish smooth navigation. Since video preview was challenging, I switched to audio-only recording to simplify processing. Python library compatibility issues were resolved by upgrading and downgrading specific modules. To optimize page loading, I removed unnecessary functions that checked module availability every time a page transitioned. To streamline backend and frontend integration, I assigned unique IDs for helper functions and ensured uniformity across the codebase. Specific module requirements were met by downloading necessary resources tailored for each module.

Frequent `FileNotFoundError` in Jupyter Notebook was resolved by switching to **VS Code**, which provided better project management and structured file handling. Configuring API keys and GitHub tokens for Azure LLM models required testing different token setups, validating environment variables, and ensuring correct permissions. Extracting and analyzing audio from video responses was simplified by allowing **direct audio or text-based input**. As the project grew, managing modules became difficult, so I restructured the project into **packages and modules**, adding missing `__init__.py` files to resolve import errors. To support **22 Indian**

languages for transcription and translation, I integrated **Gemini API** for seamless ASR and translation. Unexpected runtime errors during API communication were managed by implementing **comprehensive fallback mechanisms**. UI/UX challenges were addressed through iterative design improvements based on user feedback, with assistance from Sairaj for refining navigation.

Managing transitions between different pages was improved by developing a **centralized navigation function**. Slow page loading due to inefficient data handling was resolved by **implementing caching**, significantly improving performance. To address code duplication across multiple pages, I modularized common functions, making the project more maintainable. The Google login feature was inconsistent on a teammate's laptop, so I worked on ensuring a **consistent environment setup** and checked API credentials to resolve the issue.

8. AI/LLM Integration

8.1 OpenAI GPT Integration

The core AI functionality is powered by OpenAI's GPT-4o model. The application connects to this LLM through:

1. API Integration:

The app uses the OpenAI API via the `initialize_openai()` function in `utils/openai_helpers.py`.

2. **Authentication:**

The app supports both direct OpenAI API keys and GitHub tokens via an Azure inference endpoint (<https://models.inference.ai.azure.com>).

- **GitHub Token API Setup:**

During the exploration of available AI models and their implementation, we realized that relying solely on the Gemini API would heavily restrict the flexibility and scalability of the project. Upon researching alternatives, we discovered GitHub's marketplace, where developers can procure a free GitHub Token to authenticate API requests securely.

- **Security Benefits:**

- GitHub tokens provide fine-grained permissions and are stored securely as `GITHUB_TOKEN` in environment variables.
- They enhance security by minimizing exposure to sensitive credentials and enabling token rotation.
- The `initialize_openai()` function checks for the presence of `GITHUB_TOKEN` and, if available, initializes the OpenAI client with the Azure endpoint.
- In contrast, API keys authenticate directly with the OpenAI API but offer less control over permission scopes and access management.

- **Recommendation:**

While API keys are easier to configure, GitHub tokens offer superior security and seamless integration with GitHub-based workflows, making them ideal for enterprise-grade applications.

Here is a step-wise guide that breaks down the steps to generating a Github token:

https://docs.google.com/document/d/1KJNfnC99PIfpIaqpvvOPENAI_API_KEY

=your_openai_api_key_here

8.2 Natural Language Processing (NLP)

The app leverages several NLP techniques to analyze and compare resumes and job descriptions effectively:

1. **Resume Analysis:**

Utilizes **spaCy** (an open-source NLP library) to extract key information such as skills, experience, education, certifications, and project details from resumes.

2. **Job Description Analysis:**

Extracts required skills, preferred skills, education requirements, and experience criteria from job descriptions

3. **Keyword Matching:**

Uses pattern matching techniques to compare resume content with job requirements.

4. **Score Calculation:**

Generates a match score by comparing extracted entities from the resume and job description, ensuring a comprehensive evaluation.

8.3 LLM-Powered Interview Question Generation

The system generates highly personalized and context-aware interview questions using:

1. **Prompt Engineering:**

Carefully designed system and user prompts to guide the LLM in generating relevant, structured, and domain-specific questions.

2. **Context Awareness:**

Integrates inputs such as job title, job description, resume content, and identified matching/missing keywords to tailor questions to the candidate's profile.

3. **Question Types:**

- Technical questions based on resume experience.
- Job-related technical questions aligned with the job description.
- Behavioral questions that assess communication, situational judgment, and professionalism.

8.4 Response Analysis and Feedback Generation

The app analyzes interview responses to generate detailed performance assessments using the LLM:

1. **Multi-dimensional Assessment:**

Evaluates responses for relevance, clarity, technical accuracy, and professionalism.

2. **Sentiment Analysis:**

Identifies emotional tone and sentiment in responses to assess confidence and demeanor.

3. **Performance Metrics:**

Calculates quantitative metrics such as speaking rate (words per minute) and response duration.

4. **Constructive Feedback:**

Generates personalized feedback, highlighting strengths and suggesting areas for improvement.

8.5 Comprehensive Report Generation

The system generates structured, detailed, and professional interview reports:

1. **Overall Assessment:**

Provides an executive summary of the candidate's performance.

2. **Structured Analysis:**

Breaks down technical strengths, areas for improvement, communication effectiveness, vocabulary usage, and behavioral assessments.

3. **Final Recommendation:**

Offers a professional recommendation on the candidate's suitability for the role, accompanied by a confidence level.

4. **Visualization:**

Presents performance insights through intuitive charts and graphs for better visual analysis.

8.6 Speech and Language Support

The application is built with multilingual capabilities and future-ready architecture for speech integration:

1. **Multi-language Support:**

Supports 22 officially recognized Indian languages to accommodate diverse candidates.

2. **Manual Transcription:**

While the core implementation does not include speech-to-text AI, the application's modular design allows for future integration of automatic transcription services.

8.7 AI Implementation Architecture

The AI components are organized into a modular and scalable structure:

1. **Utils Package:**

Contains specialized modules for AI functionality, including:

- `openai_helpers.py` – Manages API integration and LLM interactions.
- `nlp_processor.py` – Handles NLP analysis of resumes and job descriptions.
- `analysis_processor.py` – Evaluates interview responses and generates feedback.

2. **Prompt Templates:**

Structured and context-aware prompts that define the format and flow of the LLM's responses.

3. **JSON Structured Output:**

Formats LLM responses as JSON objects to facilitate easy parsing and integration.

What is a JSON File?

A **JSON (JavaScript Object Notation)** file is a lightweight, text-based format used to store and exchange structured data. It is easy to read and write for both humans and machines. In the context of this project:

- LLM responses are formatted as JSON objects to ensure data consistency and simplify data exchange between components.
- JSON files follow a key-value pair structure, making them ideal for representing hierarchical and nested data.

4. Error Handling:

Implements robust error handling mechanisms to manage AI service failures gracefully.

8.8 User Experience Integration

The AI is seamlessly embedded into the user experience to ensure intuitive and interactive usage:

1. Real-time Analysis:

Provides interactive feedback during the interview process, ensuring a dynamic experience.

2. Data Visualization:

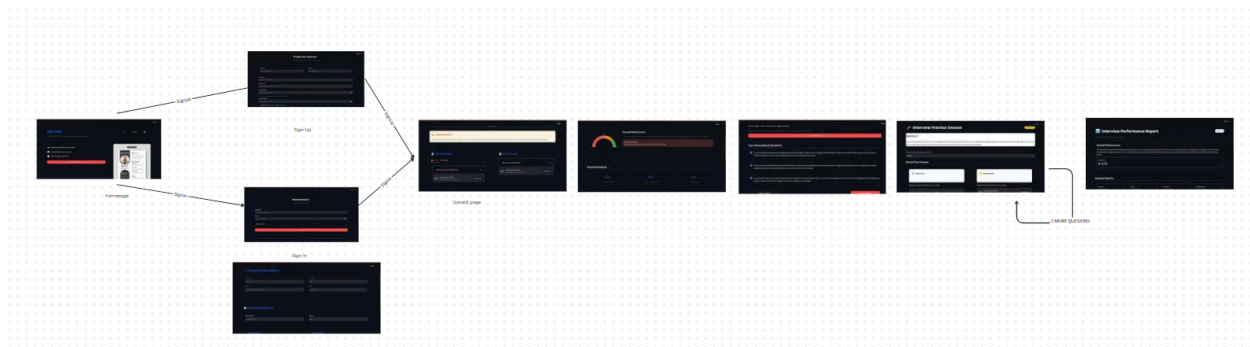
Presents AI-generated insights through intuitive charts and performance dashboards.

3. Progressive Workflow:

Guides the user step-by-step from resume upload to final interview report, enhancing the overall user journey.

9. UI/UX Design

The **Ace Hire** interview preparation platform implements a user-centered design approach, prioritizing simplicity, accessibility, and guided progression. The UI/UX design focuses on reducing cognitive load during the already stressful interview preparation process by presenting a clear, sequential workflow that guides users from resume upload to final assessment.



9.1 Design Philosophy

The platform's design philosophy emphasizes:

- **Simplicity:** Minimalist design that focuses user attention on essential tasks.
- **Accessibility:** Compliance with accessibility standards to ensure inclusivity.
- **Guided Progression:** Step-by-step navigation that clearly indicates progress, reducing user anxiety.

The user interface is optimized to present information in a logical, sequential manner while minimizing distractions and unnecessary complexity.

9.2 Interface Structure

The application employs a **single-page application (SPA)** structure built on Streamlit's reactive framework, with a consistent layout comprised of:

- **Navigation Sidebar:** Provides contextual awareness of progress through the interview preparation process.
- **Main Content Area:** Adapts dynamically to the current task while maintaining visual consistency.
- **Feedback Section:** Delivers real-time analysis and guidance throughout the user journey.

This structure creates a familiar environment where users can focus on content rather than learning new navigation patterns.

9.3 Page-Specific Design Considerations

9.3.1 Resume Upload Page

- **Minimalist Entry Point:** Simple upload interface with clear file format instructions.
- **Immediate Validation:** Visual confirmation of successful document processing.
- **Preview Capability:** Resume text display with section highlighting for transparency.
- **Job Description Input:** Contextual placement alongside the resume for logical task flow.

9.3.2 Resume-Job Match Page

- **Visual Score Representation:** Gauge chart displaying match percentage for immediate comprehension.
- **Keyword Comparison:** Color-coded visualization of matching vs. missing skills.

- **Actionable Recommendations:** Contextual suggestions for resume improvement.
- **Detailed Expandable Sections:** Progressive disclosure of analysis details to prevent information overload.

9.3.3 Question Generation Page

- **Context-Aware Questions:** Clear presentation of AI-generated questions with visual separation.
- **Question Type Indicators:** Visual cues differentiating technical vs. behavioral questions.
- **Regeneration Option:** User control for obtaining alternative questions if needed.
- **Preparation Guidance:** Supportive information about responding effectively.

9.3.4 Interview Session Page

- **Distraction-Free Environment:** Focused interface highlighting the current question.
- **Multimodal Input Options:** Tabbed interface for text or audio responses.
- **Real-Time Feedback:** Immediate analysis presentation after each response.
- **Progress Indication:** Visual cues showing completed vs. remaining questions.

9.3.5 Final Report Page

- **Executive Summary:** Prominently displayed overall assessment.
- **Data Visualization:** Interactive charts showing performance across different dimensions.
- **Detailed Breakdown:** Organized sections for technical, communication, and behavioral assessments.
- **Actionable Insights:** Clearly presented strengths and improvement areas.

- **Export Functionality:** Prominent download options for report portability.

9.4 Visual Design Elements

9.4.1 Color System

The application implements a purposeful color palette that:

- Uses blue tones for the primary interface to evoke trust and professionalism.
- Employs green elements for positive feedback and successful actions.
- Utilizes amber/yellow for cautionary information and areas needing attention.
- Reserves red highlights for critical gaps or issues requiring immediate attention.
- Maintains sufficient contrast ratios (minimum 4.5:1) for text elements to ensure accessibility.

9.4.2 Typography

The platform employs a clear and consistent typographic structure to improve readability:

- **Hierarchical Structure:** Clear differentiation between headings, body text, and data.
- **Readability Focus:** Sans-serif fonts optimized for screen reading.
- **Consistent Sizing:** Proportional text sizes maintaining relationships across different screen sizes.
- **Limited Variety:** Restricted to 2-3 font families to maintain visual cohesion.

9.4.3 Interactive Elements

The platform incorporates thoughtful interactive elements to ensure smooth user engagement:

- **State Indication:** Clear visual feedback for hover, active, and focused states.
- **Affordance Design:** Interface elements that visually communicate their function.
- **Consistent Patterns:** Similar actions maintain consistent appearance across the application.
- **Progressive Disclosure:** Complex information revealed through expandable sections.

9.5 Accessibility Considerations

The interface design incorporates several **accessibility features** to ensure inclusivity:

- **Keyboard Navigation:** Full functionality accessible without mouse interaction.
- **Screen Reader Compatibility:** Semantic HTML structure with appropriate ARIA attributes.
- **Color Independence:** Critical information conveyed through multiple channels beyond color.
- **Text Scaling:** Interface elements that respond appropriately to browser text size adjustments.
- **Error Prevention:** Confirmations for important actions and clear error recovery paths.

9.6 Responsive Design

While primarily designed for **desktop use** given the interview preparation context, the interface implements responsive principles to ensure a consistent experience across devices:

- **Flexible Layouts:** Content areas that adapt to different screen widths.
- **Component Reflow:** UI elements that reposition based on available space.
- **Touch Compatibility:** Interactive elements sized appropriately for touch interaction.
- **Media Adaptations:** Visualizations that scale or transform based on viewport constraints.

9.7 User Testing Insights

Initial **user testing** revealed several insights that informed design refinements:

- Users preferred having all response analysis on a single screen rather than paginated.
- The audio recording interface benefited from more explicit instructions and feedback.
- Visual progress indicators significantly improved user confidence during the interview session.
- The final report's executive summary proved more valuable than initially anticipated.

9.8 Future UI/UX Enhancements

Based on user feedback and ongoing evaluation, several **UI/UX improvements** are planned:

- **Guided Tutorials:** Integration of guided tutorials for first-time users.
- **Enhanced Visualization Options:** More robust comparison of performance across multiple interview sessions.
- **Customizable Themes:** Implementation of personalized themes and accessibility preferences.

- **Companion Mobile Application:** Development of a mobile app for on-the-go report review.

9.9 Conclusion

The **Ace Hire** platform's UI/UX design successfully balances sophisticated AI capabilities with an approachable, intuitive interface that guides users through the complex process of interview preparation. The platform provides valuable insights in an accessible manner, ensuring a streamlined and user-friendly experience that adapts to diverse user needs.

10. Code Execution Guide

This section provides detailed instructions on setting up, running, and troubleshooting the Ace Hire application in various environments, ensuring seamless execution and performance.

10.1 System Requirements

Before running the Ace Hire application, ensure your system meets the following requirements:

Python Version: Python 3.7 - 3.9 recommended

Justification: While newer Python versions (3.10+) offer performance improvements, our application depends on several libraries with compatibility constraints. Specifically, certain versions of SpaCy, TensorFlow dependencies, and the streamlit-webrtc package have shown stability issues with Python 3.10+. Python 3.7-3.9 provides the optimal balance of modern features and library compatibility.

Operating System: Windows 10/11, macOS 10.15+, or Linux (Ubuntu 18.04+)

Justification: These OS versions ensure proper support for the audio recording functionality via WebRTC and the PDF processing components. Earlier OS versions may lack necessary multimedia or document processing capabilities.

Memory: Minimum 4GB RAM (8GB recommended)

Justification: The NLP processing components and language models (particularly SpaCy's `en_core_web_sm`) require significant memory. Additionally, when processing larger PDF resumes or generating comprehensive reports, memory usage can spike temporarily.

Storage: At least 500MB free disk space

Justification: This accounts for the Python environment (~100MB), dependencies (~300MB including SpaCy models), and temporary storage for uploaded documents and audio recordings during the interview session.

Internet Connection: Required for API access to OpenAI and Gemini services

Justification: The application's core AI functionality depends on real-time API calls to these services. A minimum speed of 1 Mbps is recommended for responsive performance.

Browser: Chrome 90+, Firefox 88+, or Edge 90+ (for viewing Streamlit application)

Justification: These browser versions or newer fully support the WebRTC standard needed for audio recording functionality. They also provide optimal support for the interactive Plotly visualizations used in the report section.

Note on Python Version Compatibility: If you've needed to downgrade Python to run the application, this confirms our compatibility requirements. Specifically, Python 3.7-3.9 offers the

most stable environment for our dependency stack. While we continue to work toward supporting newer Python versions, maintaining compatibility with critical libraries like SpaCy, streamlit-webrtc, and certain OpenAI SDK versions necessitates this version constraint.

For users running Python 3.10+, it is recommended to create a dedicated virtual environment with Python 3.9 specifically for this application to avoid conflicts with other projects.

10.2 Environment Setup

10.2.1 Python Environment

To avoid dependency conflicts, it is recommended to use a virtual environment for the application. This ensures isolation of project-specific packages and dependencies.

10.2.2 Installing Dependencies

All required packages are listed in the requirements.txt file. Running the installation command will ensure that the necessary dependencies are installed. The list of dependencies includes streamlit, openai, spacy, plotly, reportlab, numpy, pandas, python-dotenv, and streamlit-webrtc.

10.2.3 Additional Setup

The SpaCy English language model (en_core_web_sm) must be installed to enable NLP functionality.

10.3 API Configuration

10.3.1 Creating the .env File

A .env file should be created in the root directory of the project to store sensitive API credentials securely.

10.3.2 Obtaining API Keys

The following API keys are required:

OpenAI API Key: Register at the OpenAI Platform to create an API key.

GitHub Token: Required for Azure inference endpoint access (if applicable).

Gemini API Key: Register at Google AI Studio to obtain a Gemini API key.

10.4 Running the Application

10.4.1 Starting the Streamlit Server

From the project's root directory, run the command to start the Streamlit server. This will launch the application and open it in your default web browser.

10.4.2 Alternative Launch Methods

For deployment scenarios or specific network configurations, alternative launch commands may be used.

10.5 Application Navigation

Once the application is launched, the interface presents a sequential workflow that guides users through the interview preparation process.

1. Upload Resume: Upload a PDF resume and enter the job description.

2. Resume-Job Match: View analysis of resume-job alignment with match scores and improvement suggestions.
3. Generate Questions: Obtain AI-generated interview questions relevant to the provided context.
4. Interview Session: Practice answering questions using either text or audio responses.
5. Final Report: View a comprehensive performance assessment and download a detailed report.

Users can navigate between these stages using the sidebar navigation menu.

10.6 Troubleshooting

10.6.1 Common Issues

ModuleNotFoundError: Verify virtual environment activation and rerun the package installation.

API connection failures: Check internet connection and verify API keys in the .env file.

Audio recording issues: Ensure browser microphone permissions are enabled.

PDF upload errors: Verify that the PDF is not password-protected or corrupted.

High memory usage: Close other applications or increase system swap space.

10.6.2 Logging

For troubleshooting, enable detailed logging. Logs will be displayed in the terminal running the Streamlit server, providing insights into application behavior and error conditions.

10.7 Performance Optimization

For improved performance, consider the following optimizations.

Use SSDs for faster read/write speeds during file operations.

Close unnecessary applications to free system memory and avoid performance degradation.

Use a dedicated Python environment manager such as Anaconda for enhanced dependency management.

Containerize the application using Docker for consistent deployment in various environments.

10.8 Security Considerations

Several security best practices should be followed to protect sensitive data and ensure application integrity.

API Keys: Never commit .env files to version control.

Resume Data: All uploaded resume and response data remains in session storage and is not persisted long-term.

Network Security: The development server is not production-ready; use proper HTTPS configurations for deployment.

Authentication: Add authentication if deploying in a multi-user environment to prevent unauthorized access.

10.9 Deployment Options

For production deployment, consider the following options.

Streamlit Cloud: The easiest deployment option with seamless GitHub integration.

Docker: Containerize the application for consistent and reproducible deployment.

Heroku: Platform-as-a-Service option for scaling applications easily.

AWS, Azure, or GCP: Suitable for enterprise-level deployments that require high availability and security.

10.10 Extending the Application

To extend the functionality of the Ace Hire application, follow these guidelines.

Add new services by developing additional modules in the `utils/` directory following existing modular patterns.

Modify `app.py` to incorporate new components and update the UI to reflect additional features.

Update session state variables to manage the data requirements of new features.

Following these guidelines ensures that new modules integrate seamlessly without disrupting the existing architecture. The Code Execution Guide for Ace Hire now provides detailed instructions to ensure seamless setup, execution, and troubleshooting across different environments.

11. Results & Metrics

11.1 Interview Preparation & Practice Tools

- Mock Interviews: Simulated sessions for various industries.
- Behavioral Questions: Evaluate soft skills (e.g., teamwork, problem-solving).
- Technical Questions: Assess domain-specific knowledge (e.g., coding, finance).
- Practice Modes:
 - Video-based sessions for real-time interview simulation.

- Audio-only mode to refine verbal communication.

11.2 AI-Based Feedback & Performance Analysis

- Body Language & Speech Analysis:
 - Detects confidence levels, posture, and eye contact.
 - Identifies filler words, pauses, and hesitations.
- Communication Clarity:
 - Evaluates pronunciation, tone, and fluency.

11.3 Skill Enhancement & Learning Suggestions

- Personalized Feedback: AI-generated insights on strengths and weaknesses.
- Course & Certification Recommendations: Suggests skill-improving resources.

11.4 Performance Tracking & Metrics

- Score-Based Reports: AI assigns performance scores based on responses.
- Progress Tracking:
 - Monitors improvements over multiple attempts.
 - Identifies patterns of growth in confidence and skillset.
- Trend Analysis:
 - Visual representation of interview performance trends over time.
 - Helps candidates refine strategies for future interviews.

12. Demo Video Link

Here is the video link of our application:

<https://youtu.be/TEh7-pOmXtA?si=HYh1RQUFLGqm383T>

13. Individual Contributions

1. Sairaj

- **Frontend Development & Integration:** Translated design concepts into a functional UI and seamlessly integrated backend-centric code with frontend components.
- **UI/UX Optimization:** Designed and refined the user experience through iterative improvements, ensuring a visually appealing and intuitive interface.
- **Feature Implementation:** Added user-centric functionalities, including a downloadable PDF report feature and access to previous sessions for enhanced usability.
- **Database Design & Performance:** Designed and implemented the database structure while optimizing queries to ensure efficiency.
- **Performance Optimization:** Focused on minimizing page load times and enhancing overall runtime efficiency.
- **OCR Fallback Integration:** Implemented a fallback OCR mechanism to handle robust cases where standard text extraction methods might fail.

- **Module Research & Evaluation:** Researched, compared, and tested multiple libraries and frameworks to determine the most efficient and suitable modules for various functionalities.
- **Module Integration & Implementation:** Integrated the selected modules, ensuring seamless compatibility with the existing architecture and optimizing them for better performance.
- **Previous Sessions Logic:** Developed the logic to store, retrieve, and display users' past sessions, ensuring seamless access to historical data.
- **Competitive Analysis:** Researched and analyzed similar projects on GitHub and other platforms to gain insights and implement best practices.
- **Code Organization & Collaboration:** Structured and organized the codebase for seamless collaboration, ensuring easy version control and deployment on GitHub.
- **Documentation & Dependencies Management:** Created a detailed README.md file outlining project setup, features, and usage. Compiled a requirements.txt file for dependency management to streamline deployment.
- **Final Application Development:** Led the final development phase, ensuring all features were implemented, tested, and ready for submission.

2. Priyanka

- Throughout the development of AceHire, I made significant contributions across multiple areas, including research, development, UI/UX, and project documentation.

- Workflow & Architecture: Collaborated with the team to design and finalize the workflow diagram, ensuring alignment with project goals.
- Technology & Backend: Researched and determined the best tech stack and backend dependencies for seamless implementation.
- Competitive Analysis: Examined and deconstructed similar GitHub projects to gather insights and improve feature implementation.
- Code Organization: Established a structured code directory, modularized components using Python packages, and resolved import issues.
- Streamlit UI Enhancement: Researched UI elements to improve user experience and accessibility, and compiled a list of these resources. Helped make the web application logo, conceptualize the color theme and other visual elements.
- GitHub API Tokens: Implemented secure authentication using GitHub tokens for API-based services.
- Application Development: Developed the first functional version of the app with complete backend logic. Integrated placeholder UI features, to be handed over to my teammates (Sairaj and Gajendra) to be worked on further.
- Report Documentation: Created the initial skeleton and structured the report to document findings, progress, and technical insights. Wrote and refined sections 5 to 10 of the report.
- Additionally, I handled debugging, optimized model integration, and ensured smooth workflow transitions for a fully operational system.

3. Tiya

- Created the UI flow on canva for an initial reference design . To get an idea of how the interface should look.
- Created the initial logo and came up with the name Ace Hire
- Started with the initial UI code using streamlit, creating respective pages (as per the UI design) as session states. Handed over to Sairaj to work on each page .
- Created navigation bar connecting each page, profile page with history of user, setting a background image, icon on button to look more professional. Overall, working with the front-end code to make it look more visually appealing.
- Worked on the ppt .
- Acted a link between teammates to enable collaboration

4. Gajendran

- Initial Frontend Development: Created the initial/basic frontend based on the provided design.
- Help Desk Page: Developed the help desk page to assist users with inquiries and support.
- Feedback Session Page: Designed and implemented the feedback session page for user input and suggestions.
- User-Centric Functionalities: Assisted a teammate (Sairaj) in adding user-centric features, including a downloadable PDF-formatted report.
- Learning from Peers: Collaborated with friends working on different projects to understand mistakes and make improvements.

5. Lavanya

- Report Creation: Played a pivotal role in drafting and compiling the project report, ensuring it was comprehensive, well-structured, and aligned with the team's objectives.
- The prototype focuses on defining scoring parameters for accurate performance evaluation and designing a professional, user-friendly final report format to enhance usability and visual appeal.

6. Amrutheshwari V

- Report Creation: Contributed significantly to the creation and compilation of a comprehensive project report, ensuring it accurately reflected the team's work and objectives.
- Team Coordination: Collaborated closely with the team to fully understand the application's flow, ensuring clarity and avoiding any ambiguity in the report.
- UI & Logo Design: Assisted in conceptualizing the initial UI design and contributed to the creation of the project's logo, ensuring both were aligned with the project's vision.
- Application Improvement: Identified and addressed application loopholes, implementing improvements to enhance overall performance and functionality.

14. Solution Impact

The solution impact of the ACE HIRE AI-powered Resume-Based Interview System is multifaceted, addressing key pain points in the recruitment process while driving efficiency and fairness. Below is an outline of its impact:

14.1 Efficiency and Time Savings

- **Automated Audio recording:** The system eliminates the manual effort of screening/audio recording resumes, significantly reducing the time required to shortlist candidates.
- **Custom Interview Generation:** By generating tailored interview questions based on the candidate's resume, the system accelerates the interview preparation process for recruiters.
- **Streamlined Workflow:** With features like resume analysis, interview scheduling, and automated assessments, the platform ensures a faster and more organized recruitment cycle.

14.2 Improved Accuracy and Fairness

- **Objective Evaluation:** Leveraging AI for resume analysis and interview scoring minimizes human biases and ensures that candidates are assessed based on merit.
- **Personalized Interviews:** The system ensures that candidates are evaluated on their actual skills and experience, enhancing fairness in the hiring process.
- **Consistent Scoring:** AI-driven scoring models provide consistent evaluations, reducing variability caused by subjective human judgments.

14.3 Scalability for Organizations

- **High Volume Hiring:** The solution can handle large volumes of resumes and interviews simultaneously, making it ideal for organizations conducting mass hiring.
- **Adaptable Across Industries:** The flexibility to customize interview templates and scoring criteria makes the system adaptable to various industries and job roles.

14.4 Enhanced Candidate Experience

- **Tailored Engagement:** Candidates feel more valued as the system delivers questions relevant to their background and expertise.
- **Transparency:** Clear feedback and scoring metrics help candidates understand their performance and areas for improvement.

14.5 Cost Reduction

- By automating multiple stages of recruitment, the system reduces the need for extensive manpower, lowering the overall cost of hiring.

14.6 Data-Driven Insights

- **Recruitment Analytics:** Employers gain access to insights such as candidate performance trends, interview conversion rates, and common skill gaps, enabling informed decision-making.
- **Talent Mapping:** The system helps organizations identify and nurture top talent, aiding in workforce planning and development.

14.7 Innovation and Competitive Edge

- Adopting an AI-powered recruitment system positions organizations as forward-thinking and tech-savvy, attracting top-tier talent who value innovative hiring practices.

14.8 Global Accessibility

Currently, AceHire focuses on assisting the data science job market by preparing candidates for their desired roles. By integrating all 22 officially recognized Indian languages, we aim to help users practice and rehearse interviews, empowering them to overcome language barriers and build confidence in their professionalism.

Furthermore, this inclusivity opens doors to cross-border opportunities, enabling candidates to apply for global roles. AceHire fosters an inclusive learning environment, and with continuous user feedback, we ensure ongoing improvements in language accuracy and relevance.

14.9 Impact on Job Seekers

- **Safe Practice Environment:** Reserved communicators and those who struggle with in-person interviews can rehearse at their own pace, improving articulation and confidence without the pressure of a live interviewer.
- **Reduced Interview Anxiety:** AI-driven mock interviews allow users to familiarize themselves with common questions, improving their ability to respond under real interview conditions. It can also be customized for individuals with cognitive differences (neurodivergence) who struggle with social engagement, especially in workplace environments, by integrating simulated exposure training.

- Enhanced Career Readiness: The structured feedback system helps users recognize their strengths and weaknesses, ensuring they are better prepared for job opportunities.

15. Future Enhancements

15.1 Technical Constraints and Solutions

Our development process revealed several technical limitations that informed our roadmap. Most notably, we encountered compatibility issues requiring Python downgrade from 3.13 to 3.11.7, as critical dependencies including SpaCy and certain Streamlit components are not yet optimized for the latest Python versions.

Addressing Technical Debt

To overcome these challenges, we plan to:

- Refactor the SpaCy implementation to improve compatibility with Python 3.10+.
- Develop containerized deployment options using Docker to isolate environment dependencies and ensure consistent performance.
- Create version-specific package requirements for different deployment scenarios to streamline future upgrades.
- Implement comprehensive test suites to identify compatibility issues earlier in the development cycle.

Framework Evolution

As part of our long-term strategy, we aim to:

- Transition from Streamlit's limitations to a more customizable UI framework such as React or Vue.js.
- Develop a robust API layer to separate frontend and backend concerns, allowing for greater modularity and scalability.
- Implement advanced state management solutions to handle complex user interactions.
- Optimize the platform for cross-platform compatibility across desktop and mobile environments.

15.2 Feature Enhancement Roadmap

15.2.1 Short-term Enhancements

Our immediate focus is on enhancing the platform's core capabilities:

- **Domain Expansion:** Extend beyond data science roles to include software engineering, business, and other high-demand sectors.
- **Resume Analysis Improvements:** Improve keyword extraction accuracy and enhance section identification to provide more granular insights.
- **Interview Question Refinement:** Develop more nuanced question generation tailored to experience level, seniority, and industry context.
- **UI/UX Optimizations:** Enhance accessibility features and improve mobile responsiveness to cater to a wider audience.
- **Performance Metrics:** Implement a detailed skill gap analysis with interactive visualizations for better user understanding.

15.2.2 Medium-term Enhancements

As we expand our feature set, we aim to introduce:

- **Ecosystem Integration:** Develop API connectors for LinkedIn, Indeed, and Glassdoor to enable seamless data import and export.
- **ATS Optimization:** Create resume templates optimized for applicant tracking systems (ATS) and provide keyword suggestions based on job descriptions.
- **Dynamic Interviews:** Implement contextual follow-up questions that adapt based on previous responses to simulate real-world interview dynamics.
- **Multi-language Support:** Expand language options to include major global languages to serve a more diverse audience.
- **Career Roadmap Tools:** Add skill development recommendations aligned with career goals to guide users on their professional journey.

15.2.3 Long-term Vision

Our long-term vision involves transforming Ace Hire into a comprehensive career development platform by introducing:

- **Advanced AI Capabilities:** Integrate multimodal analysis that incorporates facial expressions, vocal tone, and sentiment analysis during interviews.
- **Fully Adaptive Conversational Interviews:** Simulate human-like interviewers that dynamically adjust the complexity and style of questions based on user performance.
- **Specialized Modules:** Develop tailored modules for neurodivergent users, offering structured practice environments and progress tracking for anxiety management.

- **Comprehensive Career Platform:** Introduce native mobile applications with offline capabilities, voice assistant integration for hands-free practice, and mentorship matching based on career trajectory analysis.

15.3 Specialized Applications and User Segments

Neurodivergent Support

Our platform holds immense potential to assist individuals with social interaction challenges by:

- Providing adjustable difficulty settings to manage anxiety and build confidence.
- Offering a structured practice environment to simulate professional social scenarios.
- Enabling progress tracking for therapeutic applications.
- Facilitating collaboration with occupational psychologists to refine personalized learning paths.

Educational Integration

To enhance career preparation for students, we plan to:

- Partner with campus career centers through bulk licensing models.
- Enable student progress tracking for career counselors to monitor skill development.
- Integrate Ace Hire into professional development curricula to align with industry standards.
- Offer early career preparation tools to help students navigate competitive job markets.

Enterprise Solutions

Ace Hire will serve corporate environments by offering:

- Branded interview environments for corporate recruiting and candidate evaluation.
- Internal promotion preparation tools to upskill existing employees.
- Team assessment frameworks to evaluate and develop team capabilities.
- HR analytics tools to identify skills gaps and recommend targeted training programs.

15.4 Business Model Evolution

Tiered Access Model

To diversify revenue streams and serve different market segments, we will introduce a tiered subscription model:

- **Free Tier:** Basic interview preparation features with a limited number of questions.
- **Professional Tier:** Comprehensive analytics, unlimited interview sessions, and downloadable performance reports.
- **Enterprise Tier:** Advanced features including team management, customizable question sets, and candidate comparison tools.

Strategic Partnerships

To enhance platform capabilities and expand our reach, we plan to pursue:

- **Learning Platform Integrations:** Establish partnerships with learning platforms to create skill development pathways.

- **University Career Center Collaborations:** Collaborate with academic institutions to integrate Ace Hire into their career services.
- **HR Technology Ecosystem Connections:** Establish connections with applicant tracking systems, HR platforms, and recruitment management solutions.
- **Industry-Specific Content Partnerships:** Develop specialized content in collaboration with industry experts to maintain relevance across domains.

This roadmap addresses both immediate technical constraints and long-term vision, creating a pathway for Ace Hire to evolve from an interview preparation tool into a comprehensive career development platform that serves diverse user needs across the professional landscape.

16. Conclusion

The project "Ace Hire" addresses the challenges faced by job seekers in preparing for modern hiring processes, leveraging AI-driven tools and innovative technologies to enhance efficiency, fairness, and accessibility. By integrating features like resume optimization, mock interviews, personalized feedback, and skill enhancement suggestions, the platform empowers candidates to navigate competitive job markets with confidence. Furthermore, its scalability and cost-effectiveness make it a valuable solution for both individuals and organizations aiming to streamline recruitment processes. The adoption of such platforms reflects the evolution of hiring practices toward data-driven, inclusive, and technologically advanced methods.

17. Github Repository Links

https://github.com/pbhatt0022/ACE-HIRE_AI-Powered-Interviewer

https://github.com/Amrutheshwari01/ACE_HIRE-AI-Powered-Interviewer

https://github.com/SairajBhise2005/ACE-HIRE_AI-Powered-Interviewer

<https://github.com/GajendiranA/ACE-HIRE-AI-Powered-Interviewer>

https://github.com/tiyarosepulikunnel/ACE-HIRE_AI-Powered-Interviewer

https://github.com/lavanya-hs15/ACE-HIRE_AI-Powered-Interviewer

18. References

[1] TheLadders, "Keeping an Eye on Recruiter Behavior," 2012. [Online]. Available:

<https://www.theladders.com>. [Accessed: 29-Mar-2025].

[2] Glassdoor Economic Research, "Why Is Hiring Taking Longer?" 2015. [Online]. Available:

<https://www.glassdoor.com/research/>. [Accessed: 29-Mar-2025].

[3] LinkedIn, "Global Talent Trends 2020," 2020. [Online]. Available:

<https://business.linkedin.com/talent-solutions/recruiting-tips/global-talent-trends>. [Accessed: 29-Mar-2025].

[4] Zoom Video Communications, "A Message to Our Users," Apr. 2020. [Online]. Available:

<https://blog.zoom.us/>. [Accessed: 29-Mar-2025].

[5] Microsoft, "Microsoft Teams reaches 75 million daily active users," Apr. 2020. [Online].

Available: <https://www.microsoft.com/en-us/microsoft-teams/>. [Accessed: 29-Mar-2025].

- [6] Gartner, "How COVID-19 Has Changed Recruiting," 2020. [Online]. Available: <https://www.gartner.com/>. [Accessed: 29-Mar-2025].
- [7] Deloitte, "The Future of Work," 2020. [Online]. Available: <https://www2.deloitte.com>. [Accessed: 29-Mar-2025].
- [8] LinkedIn Talent Blog, "How Virtual Interviews Are Speeding Up Hiring," 2020. [Online]. Available: <https://business.linkedin.com/talent-solutions>. [Accessed: 29-Mar-2025].
- [9] McKinsey & Company, "Diversity Wins: How Inclusion Matters," 2020. [Online]. Available: <https://www.mckinsey.com/>. [Accessed: 29-Mar-2025].
- [10] HireVue, "Case Study: Unilever's AI-Powered Hiring Process," 2020. [Online]. Available: <https://www.hirevue.com/>. [Accessed: 29-Mar-2025].
- [11] CNBC, "Amazon's Hiring During the Pandemic," 2020. [Online]. Available: <https://www.cnbc.com/>. [Accessed: 29-Mar-2025].
- [12] Buffer, "State of Remote Work," 2020. [Online]. Available: <https://buffer.com/state-of-remote-work>. [Accessed: 29-Mar-2025].
- [13] <https://github.com/marketplace>
- [14] <https://github.com/darrylschaefer/mock-interviews-with-ai?tab=readme-ov-file>
- [15] <https://github.com/modamaan/Ai-mock-Interview>
- [16] <https://github.com/jiatastic/GPTInterviewer>