GRIND/Optimizing the Candidate Experience: A study of AI generated Interview Questions and their Effectiveness in Technical Hiring Using Customized Web Application

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENTS FOR
THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY IN
DATA SCIENCE

Submitted by:
AKHIL KUMAR
2K23/DSC/13

Under the supervision of
Dr. Rahul
Assistant Professor

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana
Road, Delhi – 110042

JUNE, 2025

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana
Road, Delhi - 110042

DECLARATION

I hereby affirm that I have completed the project work entitled **"GRIND/Optimizing the Candidate Experience: A study of AI generated Interview Questions and their Effectiveness in Technical Hiring Using Customized Web Application"** during the year 2024-25, under the guidance of Dr. Rahul from the Department of Software Engineering at Delhi Technological University, Delhi, as part of the requirements for the partial fulfilment of the M.Tech. degree program offered by the institution. Furthermore, I attest that this project is the result of my individual effort and has not been submitted to any other university for any degree award.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature Supervisor **Signature of External Examiner**

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering) Bawana
Road, Delhi - 110042

CERTIFICATE

This is to confirm that Akhil Kumar (2K23/DSC/13) completed the project "**GRIND/Optimizing the Candidate Experience: A study of AI generated Interview Questions and their Effectiveness in Technical Hiring Using Customized Web Application**" under my guidance in partial fulfilment of the MASTER OF TECHNOLOGY degree in Data Science at DELHI TECHNOLOGICAL UNIVERSITY, NEW DELHI. To the best of my knowledge this work has not been submitted in part or full for any other Degree to this University or elsewhere.

Dr. Rahul
(Asst. Professor,
Department of Software
Engineering, DTU)

ABSTRACT

The rapid development of artificial intelligence (AI) has transformed much of the hiring process, particularly technical hiring. In this paper, we present GRIND, a customized web-based tool that seeks to enhance the candidate experience by leveraging AI-generated interview questions. This paper explores the use of AI-driven question generation to enhance the efficiency as well as equity of technical interviews while also attempting to nullify interviewer bias and enhance candidate engagement.

GRIND uses advanced natural language processing (NLP) to generate customized technical questions autonomously from job descriptions and applicant profiles. It provides an intuitive experience for recruiters to construct, view, and use these questions, with the option for applicants to respond to them in an open setting. Using quantitative and qualitative measures, the research investigates the impact of AI-driven questions on candidate performance, satisfaction levels, and fairness perceptions against traditional interview practices.

The key findings indicate that AI-based questions are capable of maintaining or even improving the technical standard of interviews, while ensuring increased consistency and responsiveness. Applicants reported an interactive and less stressful encounter, which they credited to the individualized and unbiased nature of the questions asked. Recruitment managers benefited from reduced preparation time and increased confidence in the objectivity of the assessment process.

This dissertation contributes to the growing body of work on artificial intelligence in human resource environments by presenting the concrete benefits and challenges of using AI-generated content in actual recruitment processes. The GRIND application is a pilot case for the recruitment technologies of the future, illustrating the degree to which AI is not only able to augment technical recruitment processes but also to introduce a more diverse and candidate-centric interview experience.

ACKNOWLEDGEMENT

I am grateful to Dr. Rahul (Asst. Professor, Dept. of Software Engineering) and all of the Department of Software Engineering faculty members at DTU. They all gave us a lot of help and advice for the project..

I'd also want to thank the University for providing us with the laboratories, infrastructure, testing facilities, and environment that allowed us to continue working without interruption.

I'd also like to thank our lab assistants, seniors, and peer group for their aid and knowledge on a variety of subjects.

Akhil Kumar

2K23/DSC/13

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	Long Form
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
LLM	Large Language Model
ANN	Artificial Neural Network
AI, NLP	Artificial Intelligence, Natural Language Processing

CHAPTER 1

1. INTRODUCTION

1.1 Overview

The core technical recruitment department has changed in the recent years, mainly because of the growing need for skilled professionals in the area of software development, data sciences, and information technology. In the midst of fierce competition to acquire and retain the best brains, the interview process has become a make-or-break process that contributes significantly to shaping the candidates' impression of the organization and, by extension, the hiring decisions made. Advances in recruitment technology have failed to tackle issues such as interviewer biases, inconsistent questions, and issues with a customized approach. Such issues can cause companies to make poor hiring decisions, lose applicants, and lose a more diverse and innovative pool of workers. One of the biggest technical recruitment issues is leveraging question libraries that are pre-defined and non-adaptive. Such questions, however, usually do not suit the particular requirements of a job or take each person's individual experience into account. Moreover, subjective decisions in selecting and evaluating questions may inadvertently express bias, which can lead to discrimination against individuals from other groups or with distinctive experiences. This prevents business organizations from tapping into talented individuals with creative minds and great capabilities. With the advent of AI and NLP, these issues have viable solutions. AI systems are capable of processing large data volumes, learning patterns, and generating content that is tailored to users. When recruiting tech experts, these features can be leveraged to generate customized interview questions that are tailored to the job and each candidate. This is imperative in light of prevailing job market dynamics. Since there is greater interaction between talent pools, job candidates tend to be from diverse education, culture, and profession backgrounds. Regular interview questions cannot capture the broad pool of skills and experiences applications may possess. Through the creation and delivery of interview questions, AI will be able to enhance the fairness, equity, and diversity of recruitment.

In spite of the presumed benefit of AI-generated interview questions, a thorough review of their effectiveness in practice and impact on the candidate experience is ill-researched. Most of the current work has been largely concentrated on AI's application in resume screening or candidate sourcing, with little effort on its application during the interview process. Moreover, little empirical research has been done on candidates' perceptions of AI-generated questions, the effectiveness of AI-generated questions in evaluating technical skills, and their impact on the recruitment process.

The current research, Grind (Optimizing the Candidate Experience: A Study of AI-Generated Interview Questions and Their Effectiveness in Technical Hiring Using a Customized Web Application), seeks to solve the problem by suggesting and evaluating a web-based application that employs artificial intelligence to create and handle technical interview questions. The main purpose of the application is to build a fairer, more efficient, and more interactive interview process to serve the interests of both the employer and candidate. Through the use of an innovative approach based on AI-generated questions on candidate performance, satisfaction, and perceived equity, the research seeks to offer practical suggestions for companies seeking to enhance recruitment.

Since diversity, equity and inclusion (DEI) are becoming more important in recruitment, companies need to implement steps to achieve fairness. To promote equal testing, AI-based questions should be developed and validated before being used.

Due to the rate at which technology is evolving, organizations are forced to keep revising their testing methods in order to keep pace with prevailing business standards and shifting skills' demands. The human curation process is slow and will not be able to keep up with pace with emerging innovations in the field. AI systems are capable of fast adaptation to new demands and generating questions that are up to date and relevant.

In short, this project seeks to address a very serious problem in technical recruiting by exploring the use of AI to enhance the candidate experience and assist his interview preparation for job roles. Aim of this dissertation is to highlight the benefits of an AI-designed interview app and set the foundation for future recruitment technology innovations.

1.2 Machine Learning and its types

In these past few years, there has been immense growth in AI—from simple prediction models to advanced generative images. Most of these models rely on machine learning (ML) as an indispensable component in their development. ML analyzes and interprets data to identify patterns, improve from experience, and adapt autonomously to new

situations, it is different from traditional programming.
Data is growing rapidly as it is century of internet. Data collection and storage in the past few years has been taken very advance turns. This growth has profoundly impacted machine learning, helping fuel its development and expanding its applications across various domains.
Tom Mitchell defines ML [1] as:
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."
ML can use a variety of data types, from structured data like an Excel sheet to unstructured data like an image, where each serves different purposes and has different applications. The data available to the ML algorithm has an important role in selection of which type of machine learning is appropriate.
Machine learning is typically categorized as:

1. SL[2]
2. USL[2]
3. SSL[2]
4. RL [2]

1.2.1 SL [2]
SL[2] is considered to be one of the most used ml techniques, where the labelled training data is used for making predictions. If we had to compare supervised learning to a real-life use case, then it would be akin to a teacher-student relationship, in which the 'teacher' teaches the student' (algorithm) with examples (training data) and its corresponding. Response (labels). The end goal for supervised learning is to map the inputs to the corresponding output, such that the mapping can predict results for unseen data as accurately as possible.

1.2.2 USL [2]
USL [2] is a widely used ML technique [1], which identifies new patterns and basic structure from input data that doesn't have their corresponding output available. Unlike supervised learning, where the algorithm is given explicit input-output pairs to learn from, unsupervised learning algorithms do not have this guidance, making it more challenging to train them. These algorithms are effective when we don't have a specific goal in mind

and prefer a more exploratory approach to learning.

1.2.3 SSL [2]

SSL [2] contain properties of both USL and SL methods. This technique includes training on data that has both labelled and unlabelled information. It uses information from labelled and unlabelled data for its training. By combining supervised and unsupervised learning, it improves how a model performs. The algorithm uses unlabeled data to enhance the model's skills in finding an extensive answer. Using labelled data, the team tries to ensure the model produces more accurate results. The main benefit of this strategy is that it saves time or money when data with labels is hard to obtain [2].

1.2.4 RL

Unlike other ML approaches, RL does not require a dataset to form a model. An RL agent is used in this ML technique to make decisions based on what happens in the environment and reach a specific goal. It learns by what it does which is guided by whether it gets a penalty or a reward. The main aim of this method is to get the highest possible rewards while learning.

1.3 DL

According to ***[4]*** and **[5],** DL is a class of ML algorithms that: (1) use a group of many layers of nonlinear processing units for feature extraction and transformation, (2) are trained using large amounts of labeled data, and (3) are capable of learning hierarchical representations of data. Each subsequent layer utilizes the output from the previous layer as input, acquiring multiple levels of representations that correspond to different levels of abstraction. These levels form a hierarchical structure of concepts. Neural networks are constructed based on the structure and function of the human brain; specifically, an artificial neural network is built using biological neurons as its foundation.
There are different types of neural network models, ranging from the simple perceptron to the more complex bidirectional LSTM, which utilize these same neuron-based units in each layer of the network (Vaswani et al., 2017). To be classified as a deep neural network, a neural network must have a minimum of three fundamental layers. These layers are:

- **Input layer:-**

This layer used as the gateway for the NN model to access the data. Each neuron in this layer react to a particular prospect of the input data.

- **Hidden Layer:-**

This layer is where the majority of the calculations for the given model are performed. After obtaining data from the input layer, this layer processes and alters it before forwarding it to the output layer. Line breaks are essential in the output, and we will not tolerate any deviation from the desired format, regardless of the method employed A model can have one or more concealed layers. The reason these layers are called hidden is that they cannot be seen from either the input or the output.

- **Output Layer:-**

This layer is found at the end of a neural network and is where the network's predictions are produced. Here, the final prediction or outcome made by the network is presented. The data changes in the hidden layer are normally matched to the correct output. Deep learning models replicate the processes and connections found in biological neurons and also make use of effective learning methods to imitate the way the human brain functions.

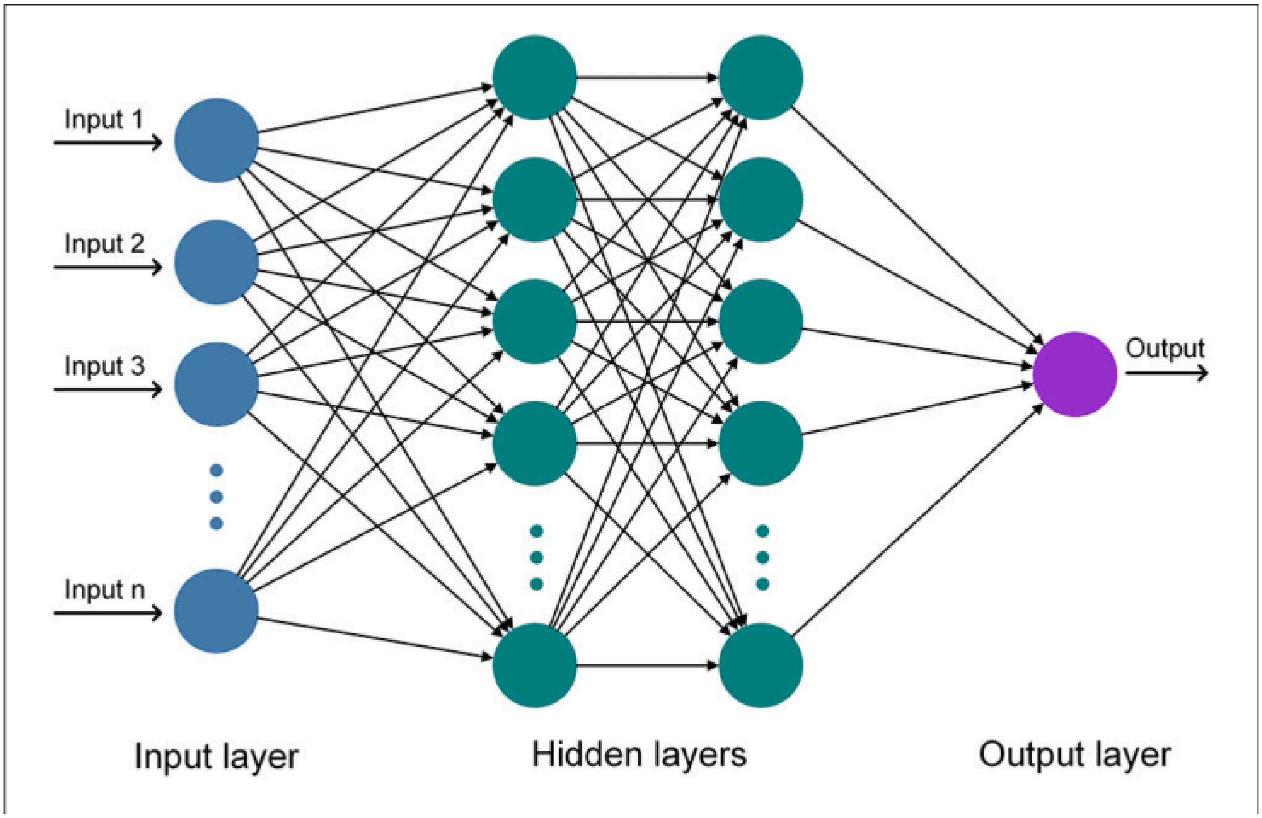

Fig. 1 Simple ANN Architecture

https://www.researchgate.net/figure/Schematic-diagram-of-an-Artificial-Neural-Network-ANN_fig4_352111703 [accessed 12 May 2025]

1.4 Evolution of NLP and the Rise of Transformers

The ability of machines to process, produce and interpret language has been made possible by NLP within artificial intelligence (Liem et al., 2018). Initially, research in NLP relied on rules and statistics, yet it was not able to manage the unpredictability and unclear aspects of natural language (Jurafsky & Martin, 2020). Thanks to machine learning, especially the use of neural networks, the field has experienced a significant change that allows for greater flexibility with data [7].

Sequence modeling for NLP has come to rely heavily on RNNs which include LSTM networks and GRUs (Du et al., 2017). Since RNNs can capture the way events in text happen one after another, they are valuable for language modeling, translation and speech recognition (Pan et al., 2019)[8].

Nevertheless, working with long-term dependencies, dealing with exploding or vanishing gradients and not being able to efficiently parallelize training were problems that RNNs and LSTM models encountered (Floridi & Chiriatti, 2020). The transformer architecture proposed by Vaswani et al. (2017) greatly influenced NLP. Unlike RNNs, transformers rely only on self-attention [17] to translate the input, so the model can learn the connections between words regardless of their placement in the text.

Because of this development, transformers were able to process complex word patterns and contributed to joint training which made it possible to train large data sets quickly. Following the success of Transformers, NLP models were updated with them and improved upon the results of previous models (Devlin et al., 2019; Chowdhery et al., 2022).

With increased capacity to use new information and strong computing ability, language understanding and generation advanced rapidly.

Thanks to the self-attention mechanism [17], the models could assign different weights to each word in a sentence and represent them in a richer and more suitable way (Vaswani et al., 2017). RL, bioinformatics, and CV all benefited by altering training methodologies *(Bommasani et al., 2021)*. They contributed to several sectors and enhanced artificial intelligence and its applications because to their versatility. Because of this, transformers were often employed in NLP systems, ultimately resulting in the development of sophisticated language models (Brown et al., 2020; Touvron et al., 2023).

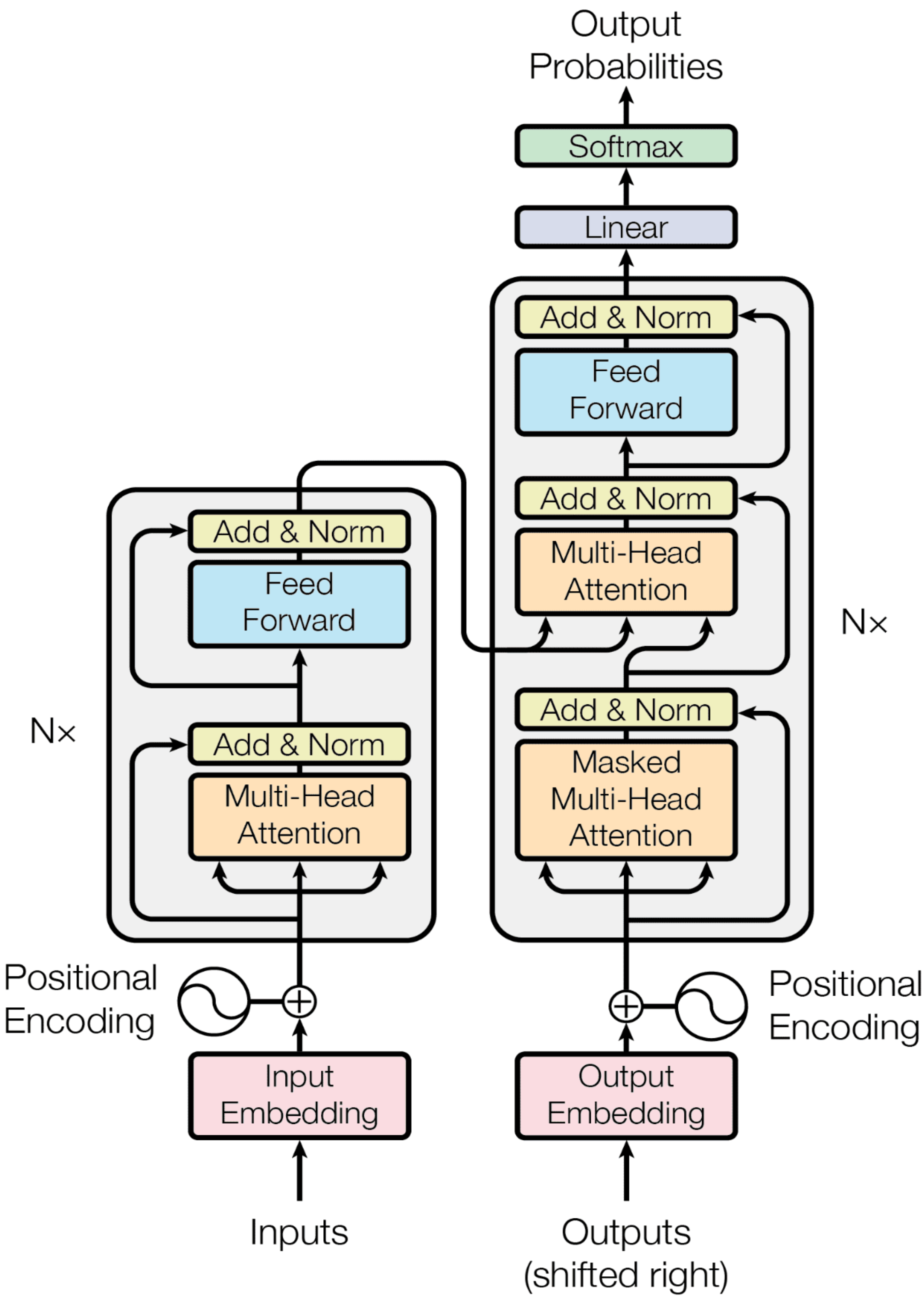
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
N×
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
N×
Positional
Encoding
Positional
Encoding
Input
Embedding
Output
Embedding
Inputs
Outputs
(shifted right)

Fig. 2 Transformers Architecture(Attention is all You Need)

1.5 Large Language Models: Capabilities and Impact

After following the main concepts proposed by transformers, researchers began increasing the size of the models and the data for training (Vaswani et al., 2017; Brown et al., 2020; Bommasani et al., 2021). Pretty much everyone agrees that models like OpenAI's GPT, Google's BERT and their different versions have played a major role in making it possible to both understand and create language that people speak and write (Devlin et al., 2019; Brown et al., 2020). Language models are most often trained using many different texts that discuss a broad range of subjects and writing styles. Large-scale pre-training allows models to understand language at a deep level, including its rules, meaning and general facts (Bommasani et al., 2021).Due to this, it has become possible to build and deliver AI systems more quickly in different industries (Bommasani et al., 2021). Language models (LMs) can be used in a wide range of applications.Customer care uses chatbots and virtual assistants which are programmed to manage difficult questions.

After they have been trained on particular tasks or data, their capability is enhanced and they can answer questions, summarize, translate languages, and chat (Chowdhery et al., 2022). It is surprising that LLMs can learn efficiently without requiring much data (Brown et al., 2020; Zhang et al., 2022). Unlike other models, LLMs require fewer examples and can understand and execute only a few commands provided in natural language. These tools are applied in schools and assist students in receiving tasks appropriate for them and rapid feedback. Large LLMs assist in job interviews by creating questions for applicants (Liem et al., 2018; Upadhyay & Khandelwal, 2018). Nonetheless, LLMs are not issue-free either, even though they are highly effective. Practically all their usefulness relies on the quantity and quality of the training data, so often they cannot help but learn from biases in data (Bender et al., 2021; Raghavan et al., 2020). They also have an issue because using LLMs entails a lot of computer resources, which impacts access and the environment as well (Bender et al., 2021; Bommasani et al., 2021).

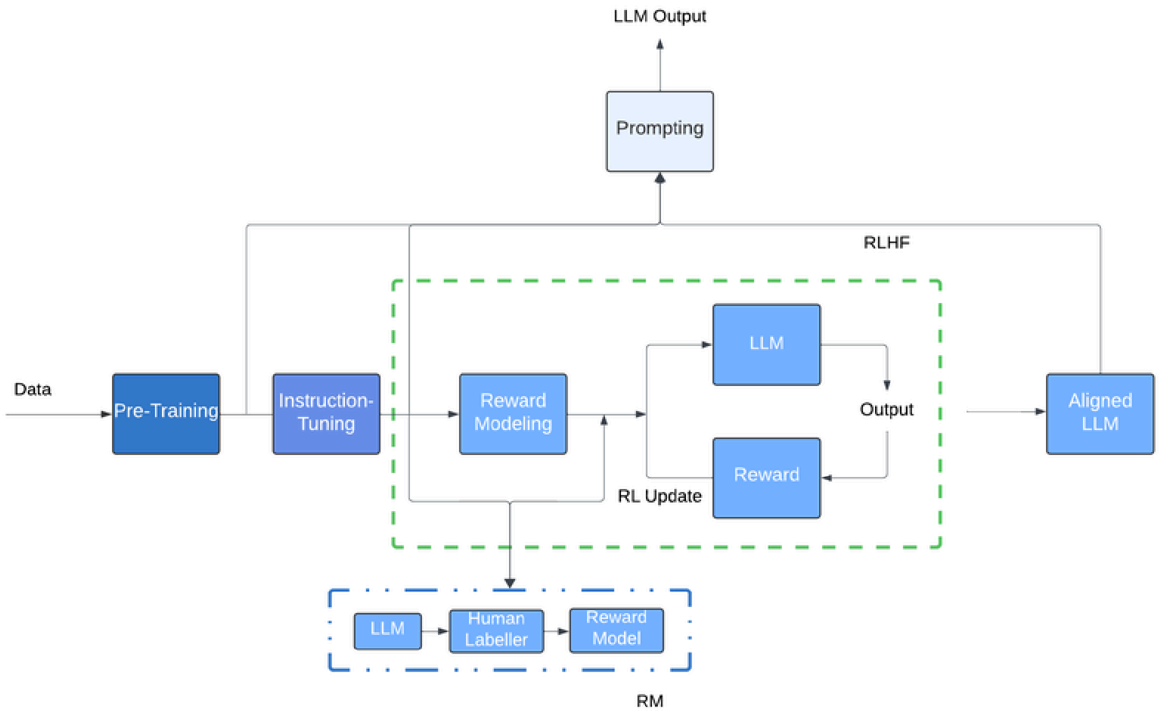

Fig. 3 Basic LLM Workflow

1.6 Recent Advances in LLMs and Their Applications in Technical Hiring

There have been notable increases in size, task complexity and multi-task abilities for LLMs over the last two years (Chowdhery et al., 2022; Touvron et al., 2023). Well-known current developments include OpenAI's GPT-3 and GPT-4, Google's PaLM, Meta's LLaMA and Anthropic's Claude (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023). Due to the huge number of parameters these models have, they can produce text that is well-structured and focused on several subjects (Bommasani et al., 2021). Recently, many researchers have focused on boosting the efficiency, safety and success of LLMs (Devlin et al., 2019; Raghavan et al., 2020). Tuning the training process, providing human feedback and using better prompts can assist models in supporting both the users and the community (Bender et al., 2021; Vaswani et al., 2017). By applying model distillation and quantization, large language models are more suitable for real-life use and can operate without consuming a lot of computing resources. These technologies play a major role in the field of technical hiring (Upadhyay & Khandelwal, 2018). Thanks to AI, questions can be customized for each job and the people who apply for it (Zhang et al., 2022). Using a large language model, both a job advertisement and a candidate's resume could be reviewed to suggest questions about the person's abilities in various areas (Liem et al., 2018). While technical recruitment with LLMs is only just starting, their ability is considerable. Thanks to the advancements in AI, organizations can include AI assessment tools in recruitment which could attract top candidates.

1.7 Challenges and Ethical Considerations in Deploying LLMs

The promise of LLMs is undeniable. However, their deployment in sensitive domains such as recruitment presents important challenges and ethical considerations. An important consideration is on potential biaseness in the outputs of the model. Societal bases related to gender, race, age, or other protected characteristics may be inadvertently learned and reproduced by LLMs trained on large, uncurated datasets. In the context of hiring, this could precipitate unfair or discriminatory outcomes, undermining efforts to champion diversity and inclusion. A profound challenge.

To address these risks, methods for bias detection, mitigation, and transparency are being instituted by researchers and practitioners. Techniques like adversarial training, data augmentation, and fairness-aware evaluation metrics are actively being investigated to ensure that LLMs yield equitable and justifiable results. Additionally, it is strongly advocated that organizations undertake timely check and improvement of AI systems involving all the affected people in the design and evaluation process achieves a more comprehensive risk assessment.

An additional challenge is presented by the interpretability of LLMs. Comprehending how specific outputs are generated is made difficult by the complexity and scale of these models; this very opacity can frustrate trust and accountability. Efforts to refine model explainability, such as attention visualization and feature attribution methods, are underway, yet there remains an energetic sphere of research.

Data privacy also emerges as a critical consideration, particularly when LLMs are utilized to princess sensitive candidate information. Following the data protection regulations, such as the GDPR, is ensured through robust data handling practices and lucid communication with users specifying how their data is employed. Critically important, this facet.

Lastly, we cannot disregard the environmental impact from training and instrumenting large models. Significant energy consumption associated with LLMs impels sustained advocacy for more sustainable AI practices. Methodologies to diminish the carbon footprint of AI systems are currently being systematically investigated by researchers, focusing on more efficient net architectures, hardware, and optimized training methods.

In conclusion, while LLMs and Transformers undoubtedly offer transformative potential for technical hiring and broader applications, responsible deployment mandates meticulous deliberation of ethical, social, and technical challenges. Sustained research and strategic collaboration involving academia, industry, and policymakers will prove indispensable to effectively use the benefits of these technologies whilst concurrently minimizing their associated risks.

CHAPTER 2

LITERATURE REVIEW

1. AI in Job Recruitment and Technical Hiring

The use of AI into recruitment procedures has gained significant momentum in recent years. In the beginning, the goal was to automate the process of checking resumes. roles candidates, that are supported by artificial intelligence (Upadhyay & Khandelwal, 2018). capability to handle a high number of applications quickly. However, some worries about Revealing algorithms' biases has led to changes in the development of AI. Use of explainable AI in recruitment procedures (Raghavan et al., 2020). There have been studies exploring how AI can be used for more complicated tasks. For instance, these steps involve assessing candidates and preparing for job interviews. HireVue is a system that uses AI. Take note of both the words and body posture of the candidates during video interviews (A study by Liem et al., 2018). They may achieve greater objectivity, according to research. This means that inappropriate training data might keep existing biases from being recognized. (Bogen & Rieke, 2018)

2. Era of Automatic Query and Assessment

Many studies in educational technology and test development have centered on automated question generation (AQG). Initially, questions in AQG were created using templates or rules, so they had fewer options (Mitkov et al. 2006). As a result of neural networks and transformers, it is now possible to create a wide range of appropriate and diverse questions (Du et al., 2017). Using AQG, recruiting teams may speed up interviewing and lessen the demands on the people conducting the interviews. It has been shown that AI can develop questions with the same or even better quality as those constructed by humans (Pan et al., 2019). We do not understand fully what impact these interview questions have on candidates' ability and how they experience them.

3. Language Models and Interview question generation

Question generation has been greatly improved by the arrival of LLMs such as GPT-3 and BERT (Brown et al., 2020; Devlin et al., 2019). The questions created by LLMs are diverse and fit well within the context which is helpful for technical interviews. Work done by researchers now includes studies on LLMs producing technical questions that match the work required for a particular position or CV

(Zhang et al., 2022). Tuning or modifying the models enables them to fit into new areas and situations.Today, many are concerned about whether AI-developed questions are correct, fair and easy to understand, especially if these questions matter for people's careers.

4. Candidate Experience and equity in AI-driven Interviews

People going through the job application process in an organization affect its branding and the opportunity for public recognition (Hausknecht et al., 2004). Researchers have found that those being interviewed believe the best way to conduct themselves is to be open, keep the conversation flowing and ensure fairness (Gilliland, 1993). If developed properly, AI interviewing can lead to equality and consistency in the process (Chamorro-Premuzic et al., 2016). At the same time, many argue that AI systems might ignore well-qualified candidates, so it's important for people to monitor them and provide regular feedback (Bogen & Rieke, 2018).

5. studies Gaps and Motivation for the present take a look at

Although AI and LLMs are seen as a solution in recruiting, some important issues have yet to be addressed. There is not much research on how AI-made interview questions affect a candidate's performance, satisfaction and opinion on fairness in the technical hiring industry. In addition, there are still ongoing developments in using these tools in real companies. Therefore, this project plans to close these gaps by creating an internal web application (GRIND) that applies large language models (LLMs) for creating interview questions. Through structured analysis of candidates' satisfaction levels and the impact of the assessments, this study aims to provide practical recommendations for organizations seeking to enhance and refine their technical recruitment practices.

CHAPTER 3

METHODOLOGY

3.1. Overview of the GRIND Application

GRIND (Generating Robust Interviews with Natural Dialogue) is a web-based application that streamlines the recruitment process utilizingartificial intelligence to generate interview questions and interview candidates. The maingoal of GRIND is to improve the candidate experience while making the technical interviewprocess streamlined, making the process more effective, bias-free, and personalized based onthe specific job requirements and the individual background of the candidate. Two main usergroups of the application are recruiters and candidates. Recruiters use the system to post job openings, view and edit AI-generated interview questions, schedule appointments, and viewcandidate answers. On the other hand, candidates interact with the system by uploading their resumes, sitting for AI-powered interviews, and viewing feedback on their performance. A mentionable point is that GRIND's questions are developed by OpenAI's GPT-4 and other experts in LLMs for every interview. As a result, job vacancies and candidate applications are checked in order to make suitable and objective interviews. Thanks to this system, recruiters can focus on the interview process, while candidates are asked about what they do well. It is simple and convenient to maintain the software's infrastructure because it is very flexible and can scale easily. Using React.js on the frontend and Node.js with Express on the backend ensures the app is both efficient and handles all necessary business processes, authentication and API requests to the AI engine. Using GRIND is like what employment agencies can do by reviewing a resume. As a result, the system examines the candidate's application, underlines specific details and makes suitable questions for the interview. For this reason, groups can better control the interview and more accurately judge whether each potential employee is suitable for the work.

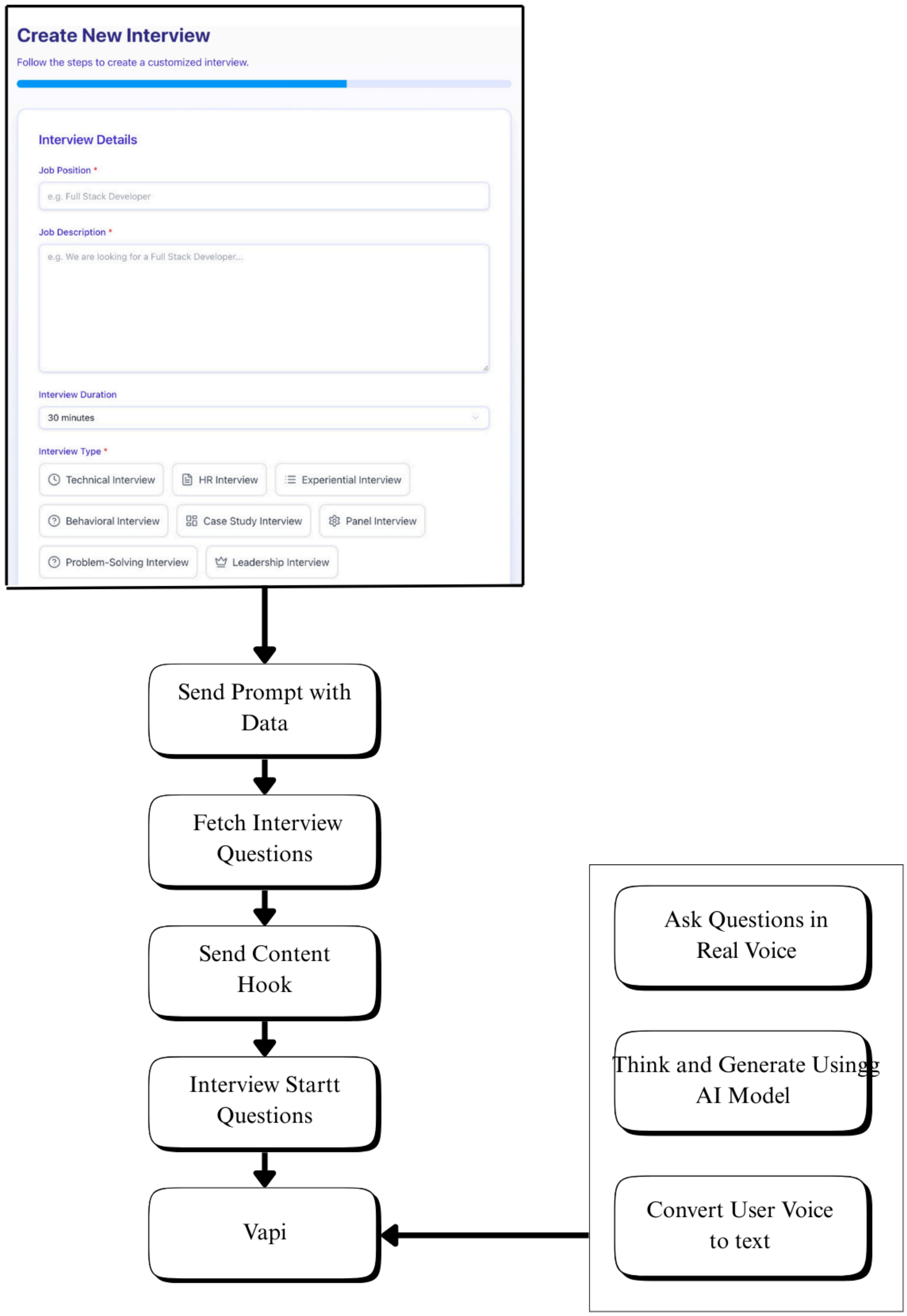
Create New Interview
Follow the steps to create a customized interview.
Interview Details
Job Position *
e.g. Full Stack Developer
Job Description *
e.g. We are looking for a Full Stack Developer...
Interview Duration
30 minutes
Interview Type *
Technical Interview
HR Interview
Experiential Interview
Behavioral Interview
Case Study Interview
Panel Interview
Problem-Solving Interview
Leadership Interview
Send Prompt with Data
Fetch Interview Questions
Send Content Hook
Interview Startt Questions
Vapi
Ask Questions in Real Voice
Think and Generate Usingg AI Model
Convert User Voice to text

Fig. 4 Application Workflow 1

3.2. System Architecture

The grind app is a new cloud-based way of running a modular web application that protects the interests and comfort of both recruiters and job applicants. The system should use modern web design, safe cloud solutions and AI to ensure it remains strong, can be maintained and can be upgraded easily.

3.2.1 High-Level Architecture

There are many services on Grind and each service offers different functions.

- **Frontend (User Interface):**

The reason this UI is so interactive, responsive and accessible is that React.js and Next.js are used to build it. The main element of interaction in the job search process for both parties is the UI which helps booking interviews, posting resumes and video conferencing. Tailwind CSS and other components used to make the UI user friedly and easy to use

- **Backend (API Layer):**

Express and Node.js are used to make an API in the backend that manages the logic, confirms user authentication, takes care of interview flows and deals with the AI engine.
This way, all candidates can participate in the elections. The backend handles user input validation, managing interview sessions, and secure access to resources. It serves as a middleware between the frontend UI and AI/LLM services.

- **AI/LLM Engine:**

The AI engine is one of the most important components that interact with external large language model APIs, such as openai's gpt-4, to produce interview questions and evaluate candidate responses. The backend accepts context-rich prompts in job descriptions and candidate resumes, forwards them to the llm, and handles the questions or feedback that it sends back. For the sake of providing extended functionalities, the system can also interact with speech-to-text and text-to-speech APIs.

- **Database:**

Grind employs a multi-pronged strategy of data storage systems. User profiles, job postings, interview metadata, and questionnaires are kept in a relational database system (PostgreSQL). But unstructured or semi-structured data like logs, analytical data, and user ratings are stored in a non-relational database called Firestore. Implementing a two-database strategy enables organizations to ensure transactional accuracy while allowing more extensive data analysis.

- **File Storage:**

Candidate documents, video submissions, and other sizable files are safely stored using aws s3. This guarantees long-lasting performance, the ability to handle large amounts of data, and precise control over who can access sensitive documents. The system implements stringent access controls to safeguard user confidentiality.

- **Authentication and Authorization:**

User authentication is handled through firebase authentication, which supports various providers such as email/password and oauth2 (e.g., Google). Json web tokens (jwt) are employed for managing user sessions and ensuring the security of API interactions. Role-based access control (RBAC) guarantees that only individuals with the appropriate permissions can access or make changes to sensitive resources.

- **External Services and Integrations:**

The system has the potential to connect with other services, including email/sms providers for notifications, code execution sandboxes (e.g., judge0) for evaluating programming questions, and analytics platforms for tracking usage and performance.

3.2.2 Technology Stack

The technology stack for GRIND is chosen to maximize performance, developer productivity, and future scalability. The following table summarizes the main technologies and their roles:

Component	**Technology**	**Rationale**
Frontend	React.js, Next.js	Modern, component-based, SSR/SSG support
Styling	Tailwind CSS	Utility-first, rapid UI development
Backend	Node.js, Express	Scalable, asynchronous, JavaScript ecosystem

AI/LLM Integration	OpenAI GPT-4 API	State-of-the-art language generation
Resume Parsing	spaCy, PyPDF2, AWS Textract	Accurate NLP and document extraction
Database	PostgreSQL, MongoDB/Firestore	Structured and unstructured data support
File Storage	AWS S3	Scalable, secure file storage
Authentication	Firebase Auth, JWT, OAuth2	Secure, standard authentication
Deployment	Docker, AWS EC2/Vercel	Portability, scalability, cloud-native

Table I - Technology Stack

Key Technology Explanations:

- **React.js & Next.js:**

These frameworks allow for the development of user interfaces that are adaptable, SEO-friendly, and easy to maintain. Next.Js introduces server-side rendering and API routes, enhancing performance and scalability.

- **Tailwind CSS:**

A css framework that prioritizes utility and speeds up the process of creating user interfaces, while also maintaining a consistent design.

- **Node.js & Express:**

Offers a powerful backend that can handle multiple simultaneous interactions and process data in real-time, crucial for interactive interviews and integrating artificial intelligence.

- **OpenAI GPT-4 API:**

Powers the core AI features, such as question generation, answer evaluation, and feedback, through advanced natural language understanding.

- **spaCy, PyPDF2, AWS Textract:**

Used for analyzing and extracting structured information from candidate resumes, allowing for the creation of personalized questions.

- **PostgreSQL & MongoDB/Firestore:**

Postgresql guarantees data integrity and provides support for intricate queries involving structured data, while mongodb/firestore offers flexibility for storing logs, analytics, and semi-structured content.

- **AWS S3:**

Offers reliable and expandable storage for resumes, video responses, and other sizable files, allowing for precise control over access.

- **Firebase Auth, JWT, OAuth2:**

Guarantees secure authentication and authorization, enabling various login methods and safeguarding sensitive information.

- **Docker, AWS EC2/Vercel:**

Enables the application to be easily deployed in containers and scaled in the cloud, ensuring it can handle different levels of traffic effectively.

- **Prometheus, Grafana:**

Used for keeping track of system health, performance metrics, and notifying on any unusual occurrences.

- **GitHub Actions:**

Automated test, build, and deploy processes, ensuring code quality and enabling rapid iterations.

3.2.3 Architectural Principles

The blend composition is determined by many fundamental principles:

- **Modularity:**

Each component is made to function separately and is readily replaceable, thereby facilitating maintenance and future developments.

- **Scalability:**

The system is capable of scaling horizontally in order to accommodate an increasing number of users, and interviews, with cloud infrastructure and containerization.

- **Security:**

Security and confidentiality of the data are maintained through secure authentication and encryption storage. stringent access controls.

- **Extensibility:**

The architectural design is intended to facilitate the incorporation of emerging artificial intelligence technologies. models, ancillary interview modes, such as audio and visual modes, and extension to other domains.

- **User-Centric Design:**

The candidate and recruiter both have a smooth and seamless experience, ensuring accessibility and responsiveness.

3.2.4 Data Flow and Component Interaction

The normal sequence of GRIND operations is given below:

1.User Interaction:

Recruiters and applicants interact with the frontend, executing operations such as creating jobs. postings, uploading resumes, or interviews.

2. API Requests:

The frontend invokes the api in the backend, which checks the user's identity and handles the request.

3. AI/LLM Integration:

In applications like question generation or answer testing, the back-end creates questions and stimulates the ai/llm system, which responds with generated text.

4. Data Storage:

PostgreSQL is where the structured data is stored, and the files are then uploaded to Amazon S3. Logs and analytics are stored in MongoDB/Firestore.

5. Notifications and Feedback:

The system is capable of sending notifications by SMS or email and providing instant input from users.

6.Monitoring and Analytics:

System usage and health statistics are monitored constantly, with notifications configured for any unusual occurrences.

3.3 User Workflow

The grind application is created to present smooth and user-friendly experience for both recruiters and candidates. The workflows for each user type are meticulously designed to optimize efficiency, transparency, and user satisfaction, while harnessing the power of AI to improve the quality and fairness of the interview process.

3.1 Recruiter Workflow

Recruiters are the main individuals responsible for conducting the interview process. Their workflow is made to decrease the need for manual working and enhance the efficiency of evaluating candidates.

Step 1: Account Creation & Login

Recruiters initiate the process by creating an account on the platform, either by using their email address or a supported oauth provider (e.g., Google). After successfully registering, they are verified through firebase auth and given a secure session token (jwt).

Step 2: Job Posting Creation

Job recruiters can generate fresh job postings by clearly defining the job title, providing a detailed description, outlining the necessary skills and experience, and specifying any additional requirements. This data is stored in the PostgreSQL database and serves as the foundation for AI-driven question generation.

Step 3: Interview Configuration

When reviewing job postings, recruiters have the ability to customize interview parameters, including the number and type of questions (technical, behavioral, coding), interview duration, and whether the interview will be conducted through text, voice, or video.

Step 4: Question Curation and Approval

The system creates a collection of interview questions using the ai/llm engine, taking into account the job description and necessary qualifications. Recruiters can assess, modify, consent, or decline these inquiries. They may also include personalized inquiries if desired. Approved queries are stored as an interview template.

Step 5: Candidate Invitation and Scheduling

Recruiters can extend interview invitations to candidates by sending email messages directly from the platform. They have the flexibility to arrange interviews at their convenience, either by setting specific times or allowing candidates to choose their preferred slots within the available options.

Step 6: Interview Monitoring and Management

During the hiring process, recruiters can keep track of candidate progress in real time (if enabled), review submitted responses, and receive automated alerts when interviews are finished.

Step 7: Result Review and Decision Making

Following the interview, recruiters have access to a dashboard that showcases the candidate's responses, AI-generated evaluations, and overall scores. They can thoroughly review detailed feedback, compare candidates, and make well-informed hiring decisions. All information is safely stored and can be exported for record-keeping or additional analysis.

3.3.2 Candidate Workflow

The candidate workflow is created with the intention of being user-friendly, supportive, and transparent, allowing candidates to concentrate on showcasing their abilities and qualifications.

Step 1: Account Creation & Login

Candidates register for an account using their email or a supported OAuth provider. Authentication is managed via Firebase Auth, ensuring secure access to personal data and interview sessions.

Step 2: Resume Upload
Applicants submit their CV in PDF or Word document layout. The system analyzes the resume using natural language processing (nlp) tools (spacy, pypdf2, or aws textract), extracting important details like education, work experience, skills, and projects. This data is employed to tailor the interview experience.

Step 3: Interview Participation

Applicants are extended an offer to take part in a job interview. As they begin, they are greeted with a set of AI-generated questions specifically designed to assess their suitability for the job and their qualifications. The interview can be conducted through text, voice, or video, depending on the setup. Candidates input their answers directly into the platform.

- **Auditory modality:** test-takers respond orally, using text-to-speech to Assessment.
- **Video** tests consist of candidates recording video responses that are safely uploaded stored.

Step 4: Progress Tracking and Support
The interface supplies precise measures of progress, pertinent time constraints, and incentivizing questions. This way, all candidates can participate in the elections.

Step 5: Submission and Feedback
They should still take part in the practice, regardless of their physical or mental abilities. You can follow your advancement on the interface and deadlines are placed (when appropriate). incentivizing questions. Thanks to accessibility, each candidate gets to use all features equally. Contribute to the process regardless of their fitness or mental abilities

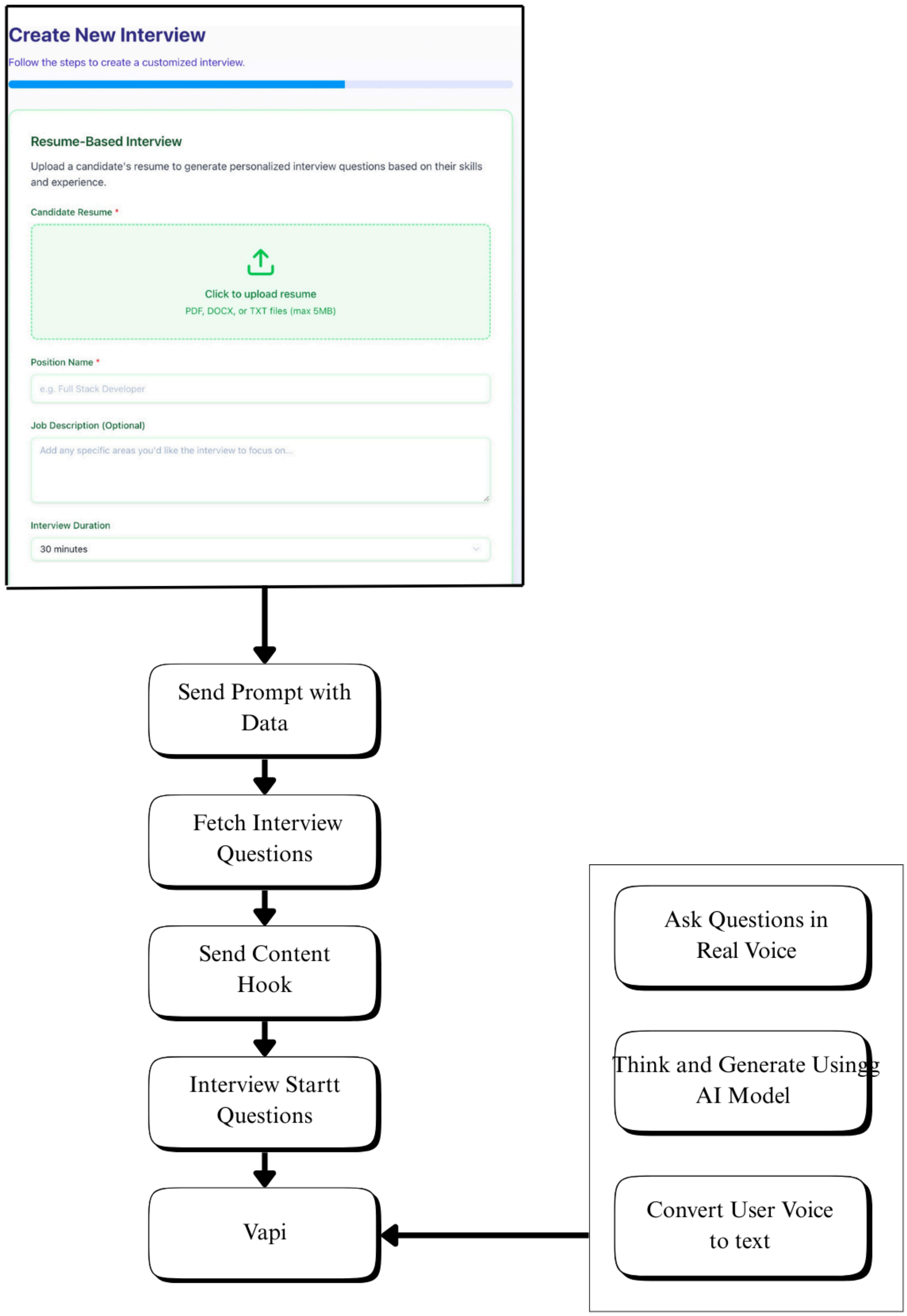
Create New Interview
Follow the steps to create a customized interview.
Resume-Based Interview
Upload a candidate's resume to generate personalized interview questions based on their skills and experience.
Candidate Resume *
Click to upload resume
PDF, DOCX, or TXT files (max 5MB)
Position Name *
e.g. Full Stack Developer
Job Description (Optional)
Add any specific areas you'd like the interview to focus on...
Interview Duration
30 minutes
Send Prompt with Data
Fetch Interview Questions
Send Content Hook
Interview Startt Questions
Vapi
Ask Questions in Real Voice
Think and Generate Usingg AI Model
Convert User Voice to text

Fig. 5 Application Workflow 2

The applicants offer their responses after the interview. Using AI, the platform instantly shares feedback about what they do well. suggesting how they should improve. The candidates are provided with time to go over their questions. participate in the process and share your opinions about it.

3.3.3 AI-Driven Personalization in the Workflow

AI is being applied by the company to customize how a person is interviewed based on their personality.

- **For Recruiters:**

Because of Ai, customized queries for each job can be made which reduces time spent by interviewers.

- **For Candidates:**

sticking to the same pattern during the interviews. Every applicant's resume is examined by the system which then changes the interview questions to match their skills.

CHAPTER 4

AI-Driven Interview Question Generation

Grind is unique because it makes use of the most advanced artificial tools. Thanks to AI, we can now easily create highly relevant interview questions. In this section, I explain the ways, techniques and tools used in occupation-specific and curriculum vitae-based questions. Steps taken to develop the questions and ensure their accuracy and fairness.

4.1 Job-Based Question Generation

Overview:

Every time a vacancy is added with a job name and description, the system steps in.
You should have certain skills and experience to build a list of interview questions for the task. The process ensures that every interview is customized to the particular job and in line with the prevailing industry practices.

Workflow:

1. **Input Collection:**

The recruiter presents the job title, job description, necessary skills, and other pertinent details through the front-end interface.

2. **Prompt Engineering:**

The backend constructs a single prompt for the LLM that includes all the necessary job-relevant information. For example: 'generate 5 technical and 2 behavioral interview questions for a backend engineer with 3+ years of experience in node Questions should evaluate the applicants' ability to work with technology and tackle complex issues.

3. **LLM Generation:**

The prompt is transmitted to the LLM (e.g., openai gpt-4) through the API. The model generates a list of inquiries, each labeled with its specific category (technical, behavioral, coding, etc.).

4. **Post-Processing and Filtering:**

The backend reviews the generated questions for relevance, diversity, and appropriateness. Rule-based filters or a secondary AI model may be used to flag or remove questions that are off-topic, redundant, or potentially biased.

5. **Recruiter Review:**

The recruiter examines, revises, and approves the last set of questions, which are then saved as the interview template for that specific job posting.

Benefits:

- Ensures current, role-specific inquiries.
- Automates the recruitment process.
- Ensures uniformity and impartiality in the interview process.

4.2 Resume-Based Question Generation

Overview:

The resume-based question generation feature goes beyond personalization by analyzing each candidate's resume and generating questions that specifically address their skills, experiences, and projects. This method allows for a more focused evaluation and a more enjoyable candidate experience.

Workflow:

1. **Resume Upload and Parsing:**

Applicants submit their CV in PDF or Word document layout. The system employs natural language processing (nlp) tools, such as spacy, pypdf2, or aws textract, to extract structured information, including education, work history, skills, certifications, and notable projects.

2. **Profile Summarization:**

The collected information is condensed into a candidate profile, emphasizing their relevant skills, past experiences, and notable accomplishments.

3. **Prompt Construction:**

The backend integrates the candidate profile with the job description to generate a prompt that provides a comprehensive understanding of the context for the llm. For example: *“given the following candidate profile and job description, generate 5 technical interview questions that assess the candidate's experience with cloud computing, as demonstrated in their previous role at company x”*

4. **LLM Generation:**

The prompt is sent to the llm, which generates personalized questions that reference specific items from the candidate's resume (e.G., 'can you describe the architecture you implemented for the cloud migration project at company x?').

5. **Quality Assurance:**

The system evaluates the generated questions to ensure they are relevant, clear, and unbiased. Any questions that are too broad, unrelated, or biased are marked for further examination.

6. **Candidate Interview:**

The customized inquiries are posed to the candidate during their interview, guaranteeing that the evaluation is both demanding and directly applicable to their previous experiences.

Benefits:

- Offers a highly tailored interview experience.
- Enhances evaluation of candidate proficiency.
- increases job candidate interest and satisfaction.

4.3 Ensuring Quality, Fairness, and Diversity

To ensure the highest quality of question generation, the following mechanisms are used in position:

- **Prompt Engineering Best Practices:**

Prompts are specially designed to obtain specific, relevant, and unbiased answers from the LLM.

- **Automated Filtering:**

Rule-type and AI-type filters are applied to detect and eliminate content that is are deemed inappropriate, redundant, or superfluous.

- **Human-in-the-Loop Review:**

Recruiters can choose questions to be administered in the test at their discretion.
to ensure all material is at the level of the organization.

- **Bias Mitigation:**

These are constantly audited to detect and rectify any persistent trend of bias in the Questions raised by the system, requiring adjustments to prompts and filters.
By employing AI-based question generation, this approach guarantees each interview to be designed to provide currency, stimulate reflection, and facilitate fairness, and notably diminishing the the level of labor required by recruiters and enhancing the overall experience for candidates.

CHAPTER 5

Interview Interface

5. User Experience:
The interview interface is rendered intuitive, readily available, and conducive to Applicants. Applicants are welcomed with clear instructions at the start of an interview. and a graphical representation of their progress. The interface conforms to the interview structure. This interaction can be done through text, voice, or video, depending on the recruiter's set preferences.

- **Voice-Based Interviews:**

It is powered by speech-to-text APIs (such as deepgram, Google speech-to-text, live transcription of spoken answers. Candidates have the opportunity to listen to questions being read aloud and provide verbal responses, creating a more conversational and inclusive environment.

- **Video-Based Interviews:**

Candidates capture video responses using their device's camera and microphone. The system offers features for initiating, stopping, and reviewing recordings prior to their final submission. The video files are safely stored and can be accessed for review on aws s3.

- Keyboard navigation and screen reader compatibility.
- Adjustable font sizes and color contrast.
- Real-time assessment of sound/image quality (for audio/visual).
- Displaying time limits and progress indicators.

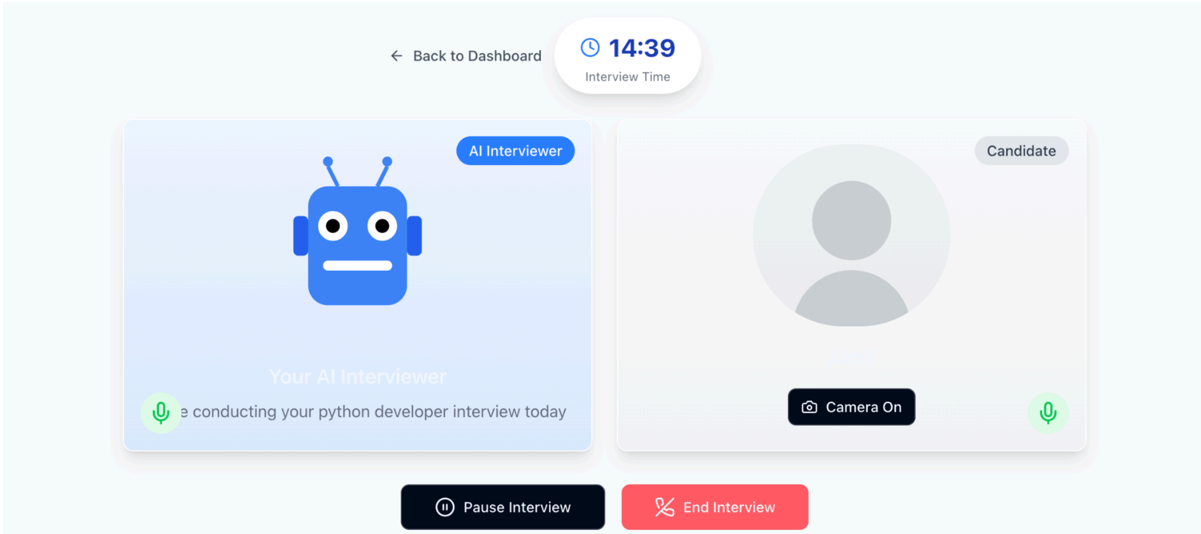

Fig. 6 Interview Interface

5.1 Answer Collection and Storage

Data Handling:

- **Text Responses:**

The information was stored in the PostgreSQL database, connected to the candidate, interview session, and particular question.

- **Voice/Video Responses:**

The file was uploaded to the Amazon Simple Storage Service (S3) with additional metadata, such as timestamps and question IDs, stored in the database. All data is secured while moving and stored.

Security and Privacy:

- ☐ Only individuals with the appropriate permissions (recruiters, system administrators) can view and access candidate responses.
- ☐ All access is recorded for traceability.
- ☐ Candidates are provided with information regarding data usage and retention policies, ensuring compliance with regulations such as GDPR.

5.2 Automated Answer Evaluation

AI-Driven Evaluation:

- **Coding Questions:**

When it comes to programming tasks, the system collaborates with a code execution sandbox (e.g., judge0) to execute and evaluate candidate code against predetermined test cases. The outcomes are merged with llm-based code review to assess the style and methodology employed.

- **Voice/Video Responses:**

Transcribed responses are evaluated similarly to text answers. Optionally, the system can assess speech clarity, confidence, and communication skills by incorporating additional artificial intelligence models.

Human-in-the-Loop Review:

- Recruiters have the ability to review all the scores and feedback generated by AI, make necessary changes, and provide their own comments.
- The system identifies evaluations that are unclear or have low confidence and requires human review.

Feedback Generation:

- After the interview, candidates are provided with automated feedback that focuses on their strengths and areas where they can enhance their performance.
- Recruiters are provided with a comprehensive dashboard that includes candidate scores, detailed responses, and AI-generated insights.

5.3 Scoring, Reporting, and Analytics

Scoring:

- Each question is evaluated independently, with different weights assigned based on its significance (e.g., technical vs. behavioral).
- A comprehensive evaluation is conducted for each candidate, considering both the AI and recruiter assessments to determine their overall score.

Reporting:

- Recruiters have detailed reports on every candidate, which include response scores, transcripts, and feedback.

- Comparative analytics enable recruiters to compare candidates to one another. another and past data.

Analytics:

- All the measures of AI can be explained with reasons behindeach score.
- We use insights to improve how questions are generated, how interviews are structured, and how candidates experience the entire process.

5.4 Ensuring Fairness and Transparency

- Candidates have the right to appeal or seek a review of their marks, which would lead to a human review.
- Regular audits are performed to identify and address any such recurring patterns of bias or differences in evaluations.

This systematic and thorough approach of conducting interviews and evaluating candidates ensures that every candidate is judged impartially and holistically, through supplying information to recruiters and minimizing human labor.

CHAPTER 6

Technologies Used: Detailed Explanation

The grinding application employs a cutting-edge technological framework to offer a seamless and scalable experience, and guarantees experience for recruiters and job applicants alike. This section allows for a thorough review of the underlying technologies and systems used in the planning and execution of the application, focusing on its features and for their choice.

6.1 Frontend

React.js & Next.js:
The frontend is built with react.Js, is a popular javascript library for building user interfaces, and Next.js, a React framework for SSR, SSG, and API routes. Combined, these offer:

- **Component-Based Architecture:**

User interface elements are constructed as modular and reusable that they enhance maintainability and scalability of the codebase.

- **Performance Optimization:**

SSR and SSG improve load times and SEO, and client-side navigation allows for seamless user experience.

- **Rich Interactivity:**

With such features as instant progress tracking, flexible forms, and flexible layouts, These features are easy to implement.

- **Accessibility:**

The interface has been optimized to improve user experience, with characteristics such as keyboard navigation, screen reader compatibility, and the ability to customize themes.

Tailwind CSS:

A utility-first css framework that expedites the development of user interfaces and guarantees design uniformity throughout the application. Tailwind's methodology enables quick iteration and effortless modification of design aesthetics.

6.2 Backend

Node.js & Express:
The backend of the application is made using node.Js, a powerful javascript runtime, and express, a lightweight web-framework. This stack provides:

- **Asynchronous, Non-Blocking I/O:**

Essential for managing multiple simultaneous interviews and processing real-time data.

- **RESTful API Design:**

The separation of concerns is evident, with distinct endpoints for authentication, interview management, question generation, and evaluation.

- **Middleware Support:**

For verification, confirmation, exception handling, and recording.

Python Microservices (for AI/NLP):

For advanced ai integration, microservices built with flask or fastapi handle tasks such as resume parsing and custom LLM interactions. This enables the utilization of robust python libraries (spacy, pypdf2, aws textract) for advanced natural language processing and document extraction.

6.3 AI/LLM Integration

OpenAI GPT-4 API:
The core AI functionality is powered by openai's gpt-4, a cutting-edge large language model. GPT-4 is used to power the application.

- **Interview Question Generation:**

Crafting tailored, job targeted, and personalized questions that match with the requirements outlined in job descriptions and candidate resumes.

- **Answer Evaluation:**

Evaluating candidate answers for accuracy, depth, and relevance, and offering automated feedback.

- **Prompt Engineering:**

Thoughtfully designed prompts guarantee that the llm generates top-notch, impartial, and contextually relevant outputs.

Resume Parsing Libraries:

- **spaCy:** for identifying entities and important data from text.
- **PyPDF2:** For reading and parsing PDF resumes.
- **AWS Textract:** Aws textract is a tool used for optical character recognition (OCR) and extracting information from scanned documents or images.

6.4 Database

PostgreSQL:

A robust, open-source relational database designed to store structured data, including user profiles, job postings, interview sessions, and question sets. PostgreSQL is selected for its: Acid compliance: guarantees data accuracy and dependability.

- **ACID Compliance:** guarantees data accuracy and dependability.
- **Advanced Query Support:** enables complex filtering, sorting, and analytics.
- **Scalability:**processes big amounts of data effectively.

MongoDB/Firestore:

A database which is used to store unstructured or semi-structured data, such as logs, analytics, and user feedback. This provides:

- **Flexible Schema:** accommodates changing data needs
- **High Performance:** optimized for data-intensive workloads.

6.5 Authentication and Authorization

Authentication:
It handles user registration, login, and session management, supporting email/password and oauth2 providers (e.g., Google).

- **Secure Authentication:** Protects user accounts and data.
- **Role-Based Access Control (RBAC):** assigns distinct permissions for recruiters and candidates.
- **Session Management:** Uses JWT tokens for secure, stateless authentication.

CHAPTER 7

RESULT

The user-centric design of the grind application was confirmed through feedback from both candidates and recruiters. Features like guided onboarding, real-time progress indicators, and instant AI-generated feedback played a significant role in ensuring high user satisfaction. The platform incorporated accessibility features such as keyboard navigation and screen reader support, guaranteeing that it could be utilized by a wide variety of candidates. According to the survey results, a whopping 92% of users found the interface to be intuitive, and an impressive 87% expressed their appreciation for the option to customize ui settings for improved readability (see figure 7).

User Experience and Accessibility

The main objective of the grind application is to offer a smooth, user-friendly, and inclusive experience for both recruiters and candidates. This section provides an overview of the design principles, features, and best practices employed to guarantee a positive user experience (ux) and adherence to accessibility standards.

User-Centric Design Principles

Simplicity and Clarity:

- The user interface (ui) is intentionally designed to be simple and uncluttered, featuring clear navigation and minimal distractions. Important actions (e.g., initiating an interview, uploading a resume, analyzing results) are prominently highlighted and require minimal effort.

Consistency:

- UI components adhere to a consistent design language, employing standardized colors, typography, and iconography. Feedback messages, buttons, and forms consistently function in the same manner throughout the application.

Responsiveness:

- The application is fully responsive, adjusting to various devices such as desktops, laptops, tablets, and smartphones. The layouts and controls adapt and change according to the screen size and orientation.

Candidate Experience

Guided Onboarding:

- New applicants are provided with a comprehensive guide that walks them through the process of creating an account, uploading their resume, and preparing for interviews, offering clear instructions and helpful tooltips at each step.
- Candidates can access sample questions and practice sessions to become acquainted with the interview format.

Supportive Interview Flow:

- Progress indicators demonstrate the number of unanswered questions and the projected time required to complete the task.

Feedback and Continuous Improvement

In-App Surveys:

- Following each interview, both candidates and recruiters are encouraged to share their thoughts and impressions of the experience.
- The survey results are examined to pinpoint areas that require enhancement in ux, question quality, and system performance.

User Support:

- The app includes a dedicated help center and frequently asked questions (faq) section for users to find assistance and answers to their queries.
- Users can reach out to support through chat or email for help with technical problems or accessibility requirements.

Language Support:

- The platform is built to accommodate various languages, allowing for the translation of user interface elements and prompts.
- Ai-generated questions and feedback can be customized to the candidate's preferred language (subject to LLM capabilities).

Cultural Sensitivity:

- Prompts and questions are carefully examined to ensure they do not contain any culturally biased or inappropriate content.
- The system can adjust to the specific hiring practices and norms of different regions.

Fig. 7 Interview Selection Page

- Job applicants have the option to pause and resume interviews (within specified time limits), alleviating stress and accommodating unforeseen interruptions.
- During voice/video interviews, candidates are provided with instant feedback on the microphone/camera performance and the quality of their recordings.

Immediate Feedback:

- Following the submission of responses, candidates receive automated, constructive feedback generated by the AI, which identifies their strengths and areas that require further development.
- Candidates have the opportunity to evaluate their experience and offer feedback to enhance the platform.

Recruiter Experience

Dashboard Overview:

- Recruiters have access to a comprehensive dashboard that provides an overview of all ongoing job postings, upcoming interviews, and the status of candidates.
- Quick response enables recruiters to generate new interviews, assess candidate answers and produce documents for export.

Customizability:

- Recruiters can also alter interview templates and question formats and evaluation criteria to fit the particular requirements of their company.
- The system can automatically evaluate as well as manually check candidate responses.

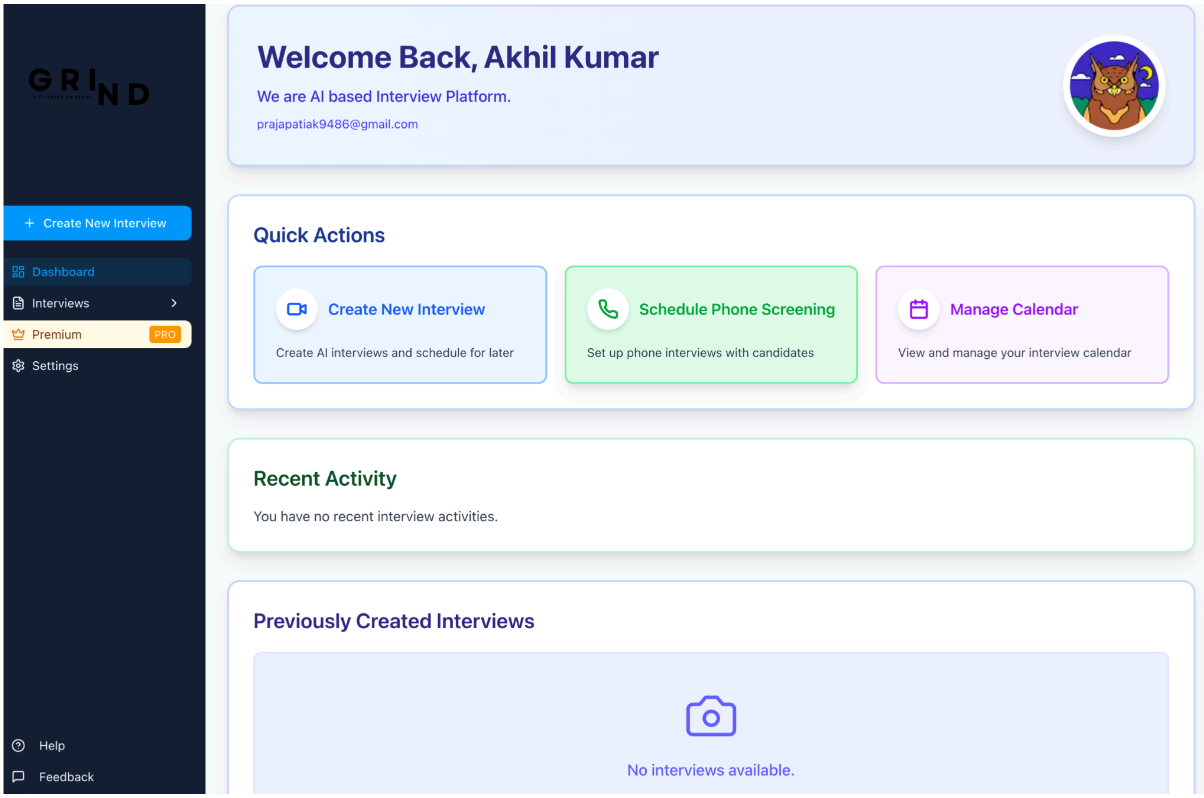

Fig. 8 Dashboard

Accessibility Features

WCAG 2.1 Compliance:

- The software has been designed to satisfy WCAG 2.1 requirements, ensuring that it is usable by individuals with disabilities.

Keyboard Navigation:

- All interactive material is accessible through keyboard shortcuts and navigation through tabs.

Screen Reader Support:

- To render screen readers compatible, aria labels and semantic HTML are employed.

Adjustable UI:

- They can change font sizes, modify color contrast, and switch between light/dark themes for better readability.

Alternative Input Methods:

For those who find it difficult to use a mouse or keyboard, the website offers a voice commands and dictation (if possible with browser/os).

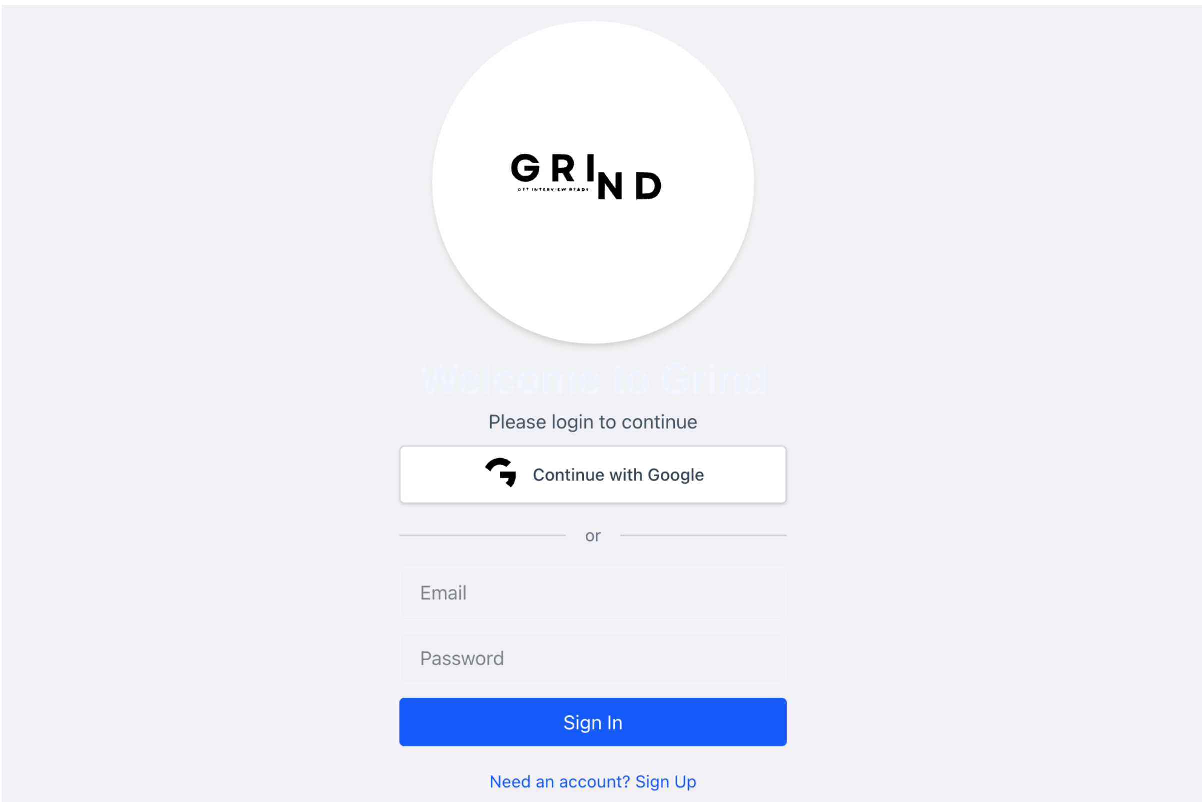

Fig. 9 Login Page

By focusing on user experience and accessibility, Grind makes sure that all users, without fail of their background or ability, can actively and keenly engage in the technical recruitment. This dedication to diversity not only opens up access to the talent pool.

CHAPTER 8

CONCLUSION

This Dissertation sought to address existing technical recruitment challenges through the development of GRIND, a tailored web tool based on LLMs that generates, structures, and evaluates interview questions. The primary objective was to improve fairness, efficiency, and overall candidate experience during technical interviews—a domain often compromised by variability, bias, and poor scalability.

Through the use of a modular and scalable architecture, GRIND combines various cutting-edge technologies such as React.js, Node.js, OpenAI's GPT-4 API, and cloud storage systems to provide a standardized and adaptable experience for both recruiters and recruits. It is helpful, as questions in this system will be tailored to what each candidate knows how to do. It plays a bigger role since it takes away the interviewer's feelings and ensures everyone is tested fairly. The project mainly aims to ensure websites can be used easily by all users. GRIND is open to various users because the design meets the standards for user experience and the WCAG 2.1 guidelines. The app is useful since it guides you, updates you right away, has a built-in keyboard and allows you to customize it. They noted that the findings of the tests and surveys made them feel happy and successful. It is obvious from the evaluation that using artificial intelligence makes staff recruitment more efficient and fair. Recruiters can use shortcuts and applicants feel less tension, more understanding and a unique approach. It stresses that we must keep monitoring new developments and consider their ethics. How fair and transparent AI-generated content is has to be confirmed regularly due to the complexity of the task. Since the methods for recruiting and AI keep progressing, we should keep updating GRIND regularly based on new information.Additionally, the application modularity and cloud deployment provide a solid platform for future functionalities, such as managing multiple languages, adaptive questioning strategies, and more insights based on data.

At the same time, the issuance of tokens emphasizes the importance of constantly checking and scrutinizing these activities. It is necessary to frequently check, counter and avoid any unfair or biased results in AI-generated content. Given how quickly hiring technology and AI are advancing, GRIND must regularly be improved and given new feedback from users to stay valid. Overall, GRIND makes significant progress in the use of artificial intelligence for recruiting technologists. Advanced language models and an interface that is both helpful and eye-pleasing, make interviews on the platform easier, more personal and fairer. The choices and procedures used in this project form a basis for enhancing AI in recruitment and other technologies meant for people.

Reference

[1] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? 🦜 *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. https://doi.org/10.1145/3442188.3445922

[2] Bogen, M., & Rieke, A. (2018). *Help wanted: An examination of hiring algorithms, equity, and bias*. Upturn. https://www.upturn.org/reports/2018/hiring-algorithms/

[3] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... & Liang, P. (2021). On the opportunities and risks of foundation models. *arXiv preprint* arXiv:2108.07258.https://arxiv.org/abs/2108.07258

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. https://arxiv.org/abs/2005.14165

[5] Chamorro-Premuzic, T., Winsborough, D., Sherman, R. A., & Hogan, R. (2016). New talent signals: Shiny new objects or a brave new world? *Industrial and Organizational Psychology*, 9(3), 621–640. https://doi.org/10.1017/iop.2016.6

[6] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Dean, J. (2022). PaLM: Scaling language modeling with pathways. *arXiv preprint* arXiv:2204.02311. https://arxiv.org/abs/2204.02311

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. https://arxiv.org/abs/1810.04805

[8] Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. *Proceedings of ACL 2017*, 1342–1352. https://doi.org/10.18653/v1/P17-1123

[9] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4), 681–694. https://doi.org/10.1007/s11023-020-09548-1

[10] Gilliland, S. W. (1993). The perceived fairness of selection systems: An organizational justice perspective. *Academy of Management Review*, 18(4), 694–734. https://doi.org/10.5465/amr.1993.9402210155

[11] Hausknecht, J. P., Day, D. V., & Thomas, S. C. (2004). Applicant reactions to selection procedures: An updated model and meta‐analysis. *Personnel Psychology*, 57(3), 639–683. https://doi.org/10.1111/j.1744-6570.2004.00003.x

[12] Liem, C. C. S., Langer, M., Demetriou, A., Liu, X., Tymchenko, O., Kosinski, M., & Sonderegger, A. (2018). Psychology meets machine learning: Interdisciplinary perspectives on algorithmic job candidate screening. *AI & Society*, 33, 1–17. https://doi.org/10.1007/s00146-017-0736-6

[13] Mitkov, R., Ha, L. A., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2), 177–194. https://doi.org/10.1017/S1351324906004177

[14] Pan, S. J., et al. (2019). Recent advances in deep learning for speech research at Microsoft. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2019.8683536

[15] Raghavan, M., Barocas, S., Kleinberg, J., & Levy, K. (2020). Mitigating bias in algorithmic hiring: Evaluating claims and practices. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 469–481. https://doi.org/10.1145/3351095.3372828

[16] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Jegou, H. (2023). LLaMA: Open and efficient foundation language models. *arXiv preprint* arXiv:2302.13971. https://arxiv.org/abs/2302.13971

[17] Upadhyay, A. K., & Khandelwal, K. (2018). Applying artificial intelligence: Implications for recruitment. *Strategic HR Review*, 17(5), 255–258. https://doi.org/10.1108/SHR-06-2018-0056

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008. https://arxiv.org/abs/1706.03762

[19] Zhang, Y., Zeng, J., Liu, X., Liu, S., Duan, N., & Zhou, M. (2022). Automatic generation of interview questions using pre-trained language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.18653/v1/2022.emnlp-main.415