**A logo of a university

Description automatically generated**

**CPSC 597 - Project**

**Project: AI – Interviewer**

**Final Report**

**Professor: Lidia Morrison**

**Submitted by:**

**Shail Bhaveshkumar Patel**

Department of Computer Science and Engineering

California State University, Fullerton

Spring 2024

Reviewed By - Drashti Amin

Date- 2nd May 2024

**Table of Contents**

Abstract……………..……………….………………………………………………………….…. 4

1. Introduction ……………………….………………………………...………….………...….....5

1.1 Purpose …………………………………...……………….………….……………….5

1.2 The Inception of AI-Based Interview Training…………………….….…………... 5

1.3 The Contribution of GPT4……………….….…………………………….………... 5

1.4 Thorough Preparation Across Various Domains……………..….….…………....6

2. Related Work.….………………….………………………………...…………………...….....,6

2.1 Deep Learning Interviews …………….....……………….………………………….6

2.2 Relation to this Project ………………….…………………………………………....6

2.3 Similarities and Differences……………………..………………….….…………....6

3. Requirements.….………………….………………………………...…………………...….....7

3.1 Tools and Technologies…………………….….…………………………………....7

3.2 Data Requirements………………………….….…………………………………....8

3.3 Comparison to Traditional Interviews……………………………………………....9

4. Methodology.….………………….……..…………………………...…………………...…...10

4.1 Data Engineering…………………………….….…………………………………..10

4.2 Prompt Engineering……………………...….….…………………………………..11

4.3 Feature Engineering…………………………………………………………...…...13

4.4 Visualization…………….………………...….….…………………………………..13

5. Data Science Algorithm & Features.…………….…………………..………………...…....15

5.1 KDD **……………………….**…………………….….…………………………………..16

5.2 High Level Architecture Design……………………..……………………………..17

5.3 Client-Side Design…………………………………………………………………..19

5.4 Interfaces **………**………………..…………….….…………………………………..20

5.5 Data Patterns Used …………………………………..……………………………..21

5.6 AutoML / Serverless AI……………………………………………………………..22

6. Experiments Section……………………………….…………………..………………...…...24

6.1 Testing **…………………....**…………………….….…………………………………..24

6.2 Model Deployment………………….………………..……………………………..25

7. Workflow.………………………………………………………………..………………...…...26

7.1 Data Flow Diagram **………**………………….….…………………………………...27

8. Active Learning or Feedback Loop ..…………….…………………..………………...…...30

9. Screenshots.……………………………….……….…………………..………………...…...31

10. Conclusion ……………………….....…………….…………………..………………...…...35

11. References.…………………………...….……….…………………..………………...…...37

**List of Figures**

Fig 1: Word Map ……………..……………….……………………………………………….…15

Fig 2: KDD ……………..………….………….……………………………………………….…17

Fig 3: High Level Architecture Diagram…………….……………………………………….…19

Fig 4: Deployed Application ……………..….……………………………………………….…26

Fig 5: DFD …………..………..……………….……………………………………………….…29

Fig 6: Homepage (I) ……………..…………..……………………………………………….…31

Fig 7: Homepage (II) ……………..…………..…………...………………………………….…31

Fig 8: Technical Interview (I) ……………..………………………………………………….…32

Fig 9: Technical Interview (II) …...……………..…………………………………………….…32

Fig 10: Technical Interview (III) ……………..……………………………………………….…32

Fig 11: Interview Guide ……………..……………………………….……………………….…33

Fig 12: Feedback (I) ……………..…………….……………………………………………......33

Fig 13: Feedback (II) ……………..…………….…………………………………………….…33

Fig 14: Resume Interview (I) ……………..…………….………………..……………….….…34

Fig 15: Resume Interview (II) ……………..…………….………………..………………….…34

Fig 16: Resume Interview (III) ……………..…………….………..……..………………….…34

Fig 17: Behavioral Interview (I) ……………..…….…….………………..………………….…35

Fig 18: Behavioral Interview (II) ……………..………….………………..………………….…35

**Abstract**

In today's ever-evolving job market, AI-enhanced tools for interview preparation are emerging as groundbreaking solutions, revolutionizing how individuals gear up for job interviews. These programs merge cutting-edge technology with a profound comprehension of the interviewing process, catering to the varied requirements of job applicants across different sectors. Central to this tool is the Open AI GPT-4 API, known for its superior language processing abilities, enabling the simulation of diverse interview situations. This feature is further supported by integrating the Whisper model, which transforms spoken language into text, offering a more dynamic and realistic practice environment. The tool is adept at scrutinizing a user's resume and job specifications to forge customized training sessions aimed at honing particular areas, whether it be technical expertise, interpersonal skills, or experiential gaps. The project is meticulously segmented into three distinct modules: behavioral, technical, and resume-based interviews, each crafted to tackle the specific hurdles associated with different types of interviews. For instance, the behavioral interview module assists users in refining fundamental skills, utilizing job descriptions to customize training for specific industry contexts. The technical interview segment concentrates on fortifying the user's technical acumen and problem-solving abilities, crucial for technical roles. Conversely, the Resume-based interview module guides users in showcasing their professional journey and competencies, ensuring they can handle queries about their work history with assurance. This AI-driven tool blends technological ingenuity with practical application, not just serving as a preparatory platform but also as an all-encompassing guide that equips users to face interviews with assurance and sophistication. It represents a significant advancement in employing artificial intelligence for personal and professional growth, offering a tailored, user-focused, and effective method for mastering the craft of interviewing.

# **1. Introduction: The Era of AI-Driven Interview**

## **1.1 Purpose**

In an era where technological advancements are reshaping all facets of life, the realm of job interview preparation is undergoing a significant transformation. The once-dominant traditional methods are now giving way to more sophisticated, individualized, and tech-savvy approaches. This evolution is epitomized by the emergence of an AI-enhanced interview preparation tool, crafted to arm job seekers with an advantage in the competitive employment landscape.

**1.2 The Inception of AI-Based Interview Training**

This tool was developed to meet a pressing need: modernizing interview preparation to align with the complexities and diversities of the contemporary job market. Conventional interview prep techniques often fall short in addressing the unique requirements of individuals, offering generic advice that lacks personal relevance. Utilizing the capabilities of Open Ai’s GPT-4 and Whisper models, this AI-driven tool heralds a new era in tailored interview coaching. By employing sophisticated AI, it offers a bespoke experience tailored to each user. Analyzing resumes and job descriptions, the AI pinpoints crucial areas for each candidate to focus on, ensuring that the preparation is not only exhaustive but also pertinent to their career goals and the specific demands of their desired job role.

## **1.3 The Contribution of GPT- 4 & Whisper in Simulating Real Interviews**

Here, the integration of Open Ai’s GPT-4 is pivotal. Renowned for its language processing prowess, GPT-4 renders realistic portrayals of varied interview contexts. It assists candidates in navigating through diverse interview challenges, from standard queries to intricate technical questions. This is augmented by the Whisper model, which adeptly transforms speech into text, infusing an extra layer of realism into the practice, enabling users to rehearse spoken responses as though in an actual interview setting.

## **1.4 Thorough Preparation Across Various Domains**

The tool encompasses three distinct segments, each homing in on crucial elements of the interview process: behavioral, technical, and resume-focused interviews. In the Behavioral Interview section, AI zeroes in on soft skills pertinent to the job description, guiding candidates in areas like communication and leadership. The Technical Interview segment concentrates on the specific technical demands of the job, with AI customizing to these specialized requirements. The Resume-Based Interview component offers a detailed review of the candidate's resume, equipping them to effectively articulate their experiences and skills.

# **2. Related Work**

## **2.1 Related Work: Deep Learning Interviews**

A pivotal work relevant to our project is the second edition of "Deep Learning Interviews," an extensive collection of fully solved problems encompassing a broad spectrum of AI's key topics. This resource is particularly valuable for machine learning MSc/PhD students and job seekers preparing for interviews or exams, offering a comprehensive overview of the field. It stands out due to its challenging problems, presented within thought-provoking questions and engaging narratives. This approach not only enhances technical skills but also equips users with the confidence and clarity needed to tackle interview questions effectively.

## **2.2 Relation to this Project**

Our project, focusing on AI-enhanced interview preparation tools, shares a common goal with "Deep Learning Interviews": to provide robust preparation for AI-related job interviews. Both resources aim to empower users with the knowledge and skills required to navigate the complexities of AI-related job interviews successfully.

## **2.3 Similarities and Differences**

* **Content Focus**: Both our tool and "Deep Learning Interviews" cover a range of AI topics. However, while "Deep Learning Interviews" offers a large inventory of problems and solutions, our tool leverages AI to create dynamic, interactive interview simulations.
* **Interactivity and Personalization**: A key distinction lies in the interactive nature of our tool. Unlike the static format of a book, our AI-powered tool offers personalized feedback and adapts to the user’s performance and needs in real-time. This interactivity is further enhanced by integrating speech-to-text capabilities using Whisper model, allowing for a more immersive interview practice experience.
* **Target Audience**: While both are invaluable to individuals preparing for AI roles, "Deep Learning Interviews" is particularly beneficial for academic preparation, such as for MSc/PhD students. In contrast, our tool caters to a broader range of job seekers, offering tailored preparation that aligns with specific job descriptions and individual backgrounds.
* **Technological Utilization**: Our project leverages advanced AI technologies like Open Ai’s GPT-4 and Whisper model, enabling a simulation of real-world interview scenarios. This contrasts with the more traditional, problem-solution format of "Deep Learning Interviews."
* **Practical Application**: "Deep Learning Interviews" provides a theoretical and problem- solving foundation, while our tool focuses on practical interview skills, including verbal communication and response structuring, offering a more holistic interview preparation.

# **3. Requirements**

**3.1 Tools & Technologies**

* **Lang chain**: Utilized for tasks involving natural language processing or AI-based operations.
* **PyPDF2**: This Python library is dedicated to manipulating PDF files, particularly useful for processing resumes in PDF format.
* **OPENAI**: A crucial library for integrating with Open Ai’s GPT-4 and other AI models, pivotal in our AI-driven functionalities.
* **Wave**: A Python module designed for reading and writing WAV files, indicative of its role in audio processing tasks.
* **Streamlit** (version 1.25.0): Employed as a Python framework to develop web applications, it's instrumental in crafting the user interface for our system.
* **Tiktoken**: This tool is deployed for managing the tokenization aspect of text processing.
* **NLTK (Natural Language Toolkit):** A well-regarded Python library for handling text-based human language data.
* **Audio\_Recorder\_streamlit**: An extension for Streamlit, this tool is integrated for adding audio recording features within the web application.
* **streamlit-option-menu**: A Streamlit add-on that provides additional user interface components.
* **FAISS-CPU**: This library is essential for performing efficient similarity searches and clustering of dense vectors, reflecting its use in our project's data processing.
* **boto3**: The AWS SDK for Python, this toolkit is used for interactions with Amazon Web Services.
* **Ipython**: An interactive Python shell that is likely utilized for the purposes of development and testing within our project.

# **3.2 Data Requirements**

For this project, I am working with two primary types of data: resumes and job descriptions.

## **3.2.1 Resumes**

1. **Type of Data**: The resumes are in PDF format, containing textual information about candidates' education, experience, skills, etc.
2. **Source**: The data comes directly from the candidates applying for the interviews. Each resume is unique and tailored to the individual's professional background.
3. **Volume of Data**: The amount of data depends on the number of resumes processed. Each resume is a separate document that contributes to the overall dataset.

## **Preprocessing and Treatment**

* + PDF Reading: Our code uses a PdfReader to extract text from each resume page.
  + Text Chunking: The NLTKTextSplitter splits the resume text into manageable chunks, likely to facilitate more effective processing.
  + Embedding Generation: Using OpenAIEmbeddings, you convert these text chunks into embeddings, which are high-dimensional vectors representing the text's semantic information.
  + Embedding Storage: The embeddings are stored using FAISS.from\_texts, which likely facilitates efficient similarity searches and retrieval.

## **3.2.2 Job Descriptions**

1. Type of Data: Textual data outlining the requirements, responsibilities, and qualifications for specific job roles.
2. Source: Job descriptions are typically sourced from job postings or provided by companies looking to hire.
3. Volume of Data: Like resumes, the volume depends on the number of job descriptions processed.
4. Preprocessing and Treatment:
   * Text Splitting: As with resumes, the NLTKTextSplitter is used to break down the job description text.
   * Embedding Creation: OpenAIEmbeddings is used again to transform the text into semantic embeddings.
   * Embedding Indexing: These embeddings are indexed using FAISS.from\_texts for efficient retrieval.

Our project involves conducting behavioral, technical, and resume-based interviews. The processed data from resumes and job descriptions is critical for tailoring these interviews to the specific context of each candidate. For instance:

* Behavioral Interviews: Our system can use the resume data to understand a candidate’s background and tailor behavioral questions accordingly.
* Technical Interviews: The job description data helps identify the key technical skills required for the job, enabling the system to ask relevant technical questions.
* Resume-Based Interviews: The embeddings from the resumes allow our system to generate questions directly related to the candidate's experience and skills.

## **3.3 Comparison to Traditional Interview Preparation**

The approach in our project differs significantly from traditional interview preparation methods. Traditional methods often rely on general questions that might not be directly relevant to the candidate's experience or the specific job they are applying for. In contrast, our AI-driven approach offers a more personalized and relevant experience by analyzing specific data points from both the candidate's resume and the job description.

In summary, our project leverages advanced AI techniques to preprocess and utilize data from resumes and job descriptions, enabling a highly personalized and efficient interview preparation process. This approach represents a significant advancement over traditional methods, providing tailored and context-specific preparation for candidates.

# **4. Methodology**

# **4.1 Data Engineering**

In the AI Interviewer platform, data engineering plays an indispensable role by ensuring data is managed, processed, and stored with efficiency and precision. This field focuses on constructing and upholding the systems necessary for data flow, acquisition, and examination, which are crucial for the platform's operational success.

## **4.1.1 Data Collection and Ingestion**

* Diverse Data Types: The system captures a variety of data types, including structured data such as user details and interview preferences, as well as unstructured forms like resumes, job descriptions, and audio inputs.

## **4.1.2 Organizing and Maintaining Data**

* Data Storage Solutions: The FAISS database is implemented for its ability to swiftly perform similarity searches, making it ideal for managing the structured data within our system.
* Secure Data Repositories: The platform utilizes robust and expandable storage solutions to house unstructured data like documents and voice recordings, ensuring security and scalability.

## **4.1.3 Refining and Modifying Data**

* Data Preparation Methods: We apply a range of data preparation techniques, including tokenization, stemming, and lemmatization, to refine text data for subsequent analysis.
* Insightful Feature Engineering: Leveraging OpenAI's sophisticated algorithms, we extract significant features from the data, which are crucial for the training of models and the generation of inferences.

# **4.2 Prompt Engineering**

This section delves into the structured prompts employed in our AI Interviewer system. These prompts, fundamental to guiding the AI in generating interview questions, are categorized into behavioral, technical, and resume-based templates.

## **4.2.1 Technical Prompt Engineering**

## The technical prompt guides the AI to assess a candidate's technical knowledge. It bases its questions on the job description, asking about specific skills relevant to the job, such as data mining or statistics.

## **Prompt**

Act in the role of the interviewer and not the candidate. Progressively, based on the job description, formulate a set of interview questions that evaluate the candidate's technical proficiency in key areas.

For instance:

For roles requiring data mining expertise, the AI interviewer might ask, "Can you explain overfitting, or describe how backpropagation works?"

If statistical knowledge is essential, questions like "What distinguishes Type I from Type II error?" may be posed.

Ensure each question is unique and not repetitive. Context: {context}

Question: {question} Answer: {answer}

## **4.2.2 Resume-Based Prompt Engineering**

This prompt is centered around the candidate's resume. It directs the AI to craft questions related to the candidate's background, work experience, and projects, ensuring the questions are relevant to their specific experiences and skills as a Full Stack Developer.

## **Prompt**

Take on the role of an interviewer rather than a candidate.

Focus progressively on the resume and develop a set of interview guidelines assessing the candidate's competencies for a Cloud Engineer role.

The interview should cover three primary areas based on the resume content:

1. The candidate's background and skills.
2. Their work experience.
3. Projects they've undertaken, if applicable.

Ensure that the questions are unique and not repetitive.

Context: {context} Question: {question} Answer: {answer}

## **4.2.3 Behavioral Prompt Engineering**

The behavioral prompt is designed to elicit soft skill-focused questions from the AI interviewer. It instructs the AI to act as the interviewer, formulating questions based on identified keywords that reflect the candidate's soft skills.

## **Prompt**

Act in interviewer's role, not the candidates.

Methodically, based on specific keywords, develop a set of guidelines for a behavioral interview aimed at assessing the candidate's interpersonal skills.

Ensure diversity in your questions and avoid repetition. Keywords: {context}

Question: {question} Answer: {answer}

In each case, the AI is instructed to avoid repeating questions, ensuring a diverse range of queries. These prompts are pivotal in tailoring the interview process to each candidate's unique profile, thereby enhancing the system's effectiveness and personalization.

# **4.3 Feature Engineering**

In the domain of AI-based interview preparation, the process of feature engineering is essential for converting raw data into a format that is not only meaningful but also optimized for efficient machine learning. At the heart of our project lies a carefully crafted feature engineering approach that greatly enhances the predictive capabilities and functionality of our models.

## **4.3.1 Rich Feature Extraction from Resumes**

Our process starts with the analysis of resumes, rich with unstructured data. In this process, akin to an art, we extract crucial information such as educational qualifications, skill sets, professional experience, and project participation. Each piece of information is methodically converted into structured data, forming the cornerstone for our AI algorithms.

## **4.3.2 Customizing Features for Different Interview Styles**

We adapt our feature extraction technique to suit various types of interviews - Behavioral, Technical, and Resume-based. For Behavioral interviews, our emphasis is on extracting soft skills such as communication abilities and leadership qualities. In Technical interviews, we concentrate on hard skills, for example, programming expertise in languages like Python. This tailored approach ensures our AI models are finely tuned to the specific requirements of diverse interview formats.

## **4.3.3 Enhancing Semantic Depth with OPENAI Embeddings**

Utilizing OPENAI's advanced embedding methods, we introduce a layer of semantic depth to our features. This step is vital in grasping the contextual subtleties of the text, significantly enhancing the precision and relevance of our similarity searches within the FAISS database.

# **4.4 Visualizations**

## **4.4.1 Extracting Keywords from Resumes**

The process of extracting keywords from resumes is an essential aspect of our AI Interviewer system, particularly for the visualization of vital data points. This process begins with the systematic analysis of the resume text, employing advanced Natural Language Processing (NLP) techniques.

## **4.4.2 Keyword Extraction Methodology**

* The system utilizes a combination of NLP methods to parse the resume content. Techniques like tokenization, frequency analysis, and relevance scoring are used to identify and extract significant keywords. These keywords typically include skills, tools, job roles, and other pertinent professional terms.
* To ensure accuracy and relevance, the extraction process also incorporates contextual analysis, which helps differentiate between general and job-specific keywords.

## **4.4.3 Importance and Relevance Scoring**

Each extracted keyword is assigned a relevance score based on its frequency and context within the resume. This scoring mechanism helps in prioritizing the keywords, ensuring that the most significant terms are highlighted in the visualization.

## **4.4.4 Creating a Word Map**

The extracted keywords are then visually represented in a word map, which offers a quick, intuitive view of the candidate’s skills and experiences.

## **4.4.5 Design of the Word Map**

The word map is designed to display keywords in varying sizes and colors. The size of each keyword in the map corresponds to its relevance score, with larger sizes indicating higher importance.

Color coding is used to categorize keywords into different groups, such as technical skills, soft skills, industry-specific terminology, etc. This categorization aids in quick visual assessment of the candidate’s skill set and areas of expertise.

Now, let us look at the word maps of our resumes.

A close-up of words

Description automatically generated

Fig 1: Word Map

# **5. Data Science Algorithms & Features**

The AI Interviewer platform incorporates a suite of sophisticated algorithms and functionalities, each meticulously chosen and fine-tuned to enrich the user's interview training experience. These integral components empower the system to provide comprehensive and nuanced training as well as insightful feedback.

* **Advanced NLP Techniques:** The platform uses advanced NLP methods such as tokenization and parsing to meticulously parse resumes and job descriptions, structuring the extracted information for further use.
* **Customized Question Crafting via Machine Learning:** Employing advanced machine learning techniques, the system crafts customized interview queries. It evaluates the data from the user's resume and the job description to formulate questions that are highly pertinent to the individual's career profile and the role they aspire to secure.
* **Sophisticated Speech-to-Text Capabilities:** The system integrates the OPENAI Whisper technology for its state-of-the-art speech recognition capabilities, essential for transcribing the user's spoken responses throughout the mock interview.

## **Lifelike Speech Synthesis:** For realistic speech output, the system deploys AWS Polly, transforming text responses into natural-sounding speech, thereby enhancing the interactivity and realism of the interview simulation.

## **Semantic Search via Embeddings:** To ensure that the generated questions are precisely matched to the user's qualifications and the job's demands, the platform leverages OPENAI's embedding algorithms in harmony with the FAISS database for an efficient similarity search.

## **Intelligent Feedback and Personalized Recommendations:** Following the interview simulation, the platform utilizes an intricate mix of heuristic and machine learning techniques to dissect the user's answers, providing tailored feedback and actionable recommendations to guide their ongoing development.

# **5.1 KDD**

1. **Data Collection**: Transforming interview replies into text via the Whisper model's speech-to-text capabilities & utilizing individual resumes for tailored interview practice.
2. **Data Preparation**: Gathering data and transforming it into vectors with OPENAI Embeddings, followed by storage in the FAISS database for efficient similarity-based searches. The outcomes from these searches are then incorporated into specialized prompts to input into the AI model.
3. **Data Mining**: Leveraging the linguistic prowess of OPENAI's GPT-4 to scrutinize and formulate interview queries and replies.
4. **Evaluation and Interpretation**: Critically evaluating the responses to offer feedback, ratings, and concise summaries.

A diagram of a data processing process

Description automatically generated**Diagram**:

Fig 2: KDD

# **5.2 High Level Architecture Design**

Here's the workflow described in the diagram:

## User Interaction

* + The process begins with a user inputting a query into the system.

## Query Processing

* + The user's input is directed towards a Conversation Focused GPT Model.
  + If the model has been fine-tuned, the input query is processed by the fine-tuned model instead.

## GPT Model Interactions

* + The GPT-4 Model is used to generate an answer based on the user's query.
  + The answer is extracted and used to create a dynamic query, presumably to fetch relevant data or generate follow-up content.

## Data Handling and Preprocessing

* + A resume is preprocessed to extract relevant information. This preprocessing step likely involves cleaning, normalizing, and structuring data.
  + The preprocessed data from the resume is then embedded, which involves creating numerical representations of the text data that can be processed by machine learning algorithms.

## FAISS Vector Database

* + The embeddings are stored in a FAISS Vector Database, which allows for efficient similarity search and retrieval.
  + This database can retrieve conversation examples and messaging frameworks relevant to the user's input and the extracted answer from the GPT model.

## Feature Extraction and Question Generation

* + The system focuses on extracting data from the provided document (the resume) and its content, using this to inform the generation of interview questions.
  + Based on prompts, the system generates a series of interview questions that are likely to be used in a subsequent interview or interaction with the user.

## Throughout the workflow, there are two main focuses.

* The first is on generating conversational replies based on user answers, which are informed by conversational data and the user's profile.
* The second focus is on extracting data from documents (resumes) and using it to generate interview questions. This involves transforming unstructured data (text) into a structured format that the AI models can utilize.

A diagram of a process flow

Description automatically generated

Fig 3: High Level Architecture Diagram

# **5.3 Client-Side Design**

In the AI Interviewer platform, the client interface is developed using Streamlit, a cutting-edge framework that streamlines the creation of interactive data applications. Streamlit's capabilities enable rapid and visually engaging user interface construction, perfectly aligning with the needs of our system's user-facing side.

## **5.3.1 Utilizing Streamlit for User Interface Creation**

* Accelerated Development and Immediate Rollout: The strength of Streamlit lies in its efficiency, enabling quick development and immediate rollout of data-centric applications. Its intuitive design and robust set of functionalities support fast-paced creation and real-time visualization of user interfaces.
* Python-Based Development: Streamlit stands out by allowing front-end development directly through Python, bypassing the conventional reliance on HTML, CSS, and JavaScript. This Python-centric development pathway is highly advantageous for our Python-skilled development team, enabling them to effectively craft and maintain the platform's interface.

## **5.3.2 Engaging and Responsive User Interface**

* Assortment of Interactive Elements: Streamlit comes equipped with a suite of interactive elements such as buttons, sliders, and text boxes, which are integral to building the engaging components necessary for an AI-guided interview process and for soliciting user responses.
* Instantaneous Interface Updates: The platform's interface is designed to refresh instantaneously in response to user actions, a key feature for delivering AI-generated responses and real-time feedback throughout the interview experience.

# **5.4 Interfaces**

## **5.4.1 User Interaction Frontend**

* **Web Interface**: The system provides a user-friendly web portal that acts as the main touchpoint for users. This interface streamlines the process of resume upload, job description entry, interview type selection, and participation in the AI-conducted interview sessions.
* **Performance Insights**: After completing the interview, users have access to a breakdown of their performance, along with actionable feedback and recommendations, all presented in a user-friendly format.

## **5.4.2 Application Programming Interfaces**

* **OPENAI GPT-4 API:** The backbone of the system's intelligent questioning and dialogue capabilities is the integration with the OPENAI GPT-4 API. This connection is indispensable for facilitating the AI-powered elements of the interview.
* **OPENAI Whisper API**: The system utilizes the Whisper API for its reliable speech-to-text functionality, ensuring that spoken responses are captured with precision.
* **AWS Polly API**: The AWS Polly API is employed to convert text into realistic speech outputs, adding an auditory layer to the user experience that simulates a conversational environment.

## **5.4.3 Data Management Backend**

## FAISS Database Connectivity: The platform employs the FAISS database to handle the extensive data it accumulates, particularly for storing resume information. This database link is essential for the system's ability to conduct accurate similarity searches and efficiently manage the data repository.

# **5.5 Design Patterns Used**

In crafting the AI Interviewer platform, the adoption of established architectural design patterns has been critical in bolstering the system’s stability, expandability, and ease of maintenance. These design patterns serve as proven templates for addressing frequent software design challenges, elevating the code’s quality and development efficacy.

## **5.5.1 Singleton Design Pattern**

* Application in Our System: The Singleton pattern is employed for the singular management of connections to the FAISS database and AWS services, guaranteeing a single shared instance of the connection object that minimizes overhead and assures uniformity throughout the platform.
* The Singleton pattern facilitates regulated access to common resources, which is vital for preserving data coherence and operational performance.

## **5.5.2 Factory Method Design Pattern**

* Application in Our System: The platform leverages the Factory Method to instantiate objects for various interview categories – Behavioral, Technical, and Resume-based. It wraps the instantiation process, contributing to the system’s modularity and its capacity to accommodate future alterations.
* The Factory Method brings about an increase in the system's adaptability and potential for growth, allowing the introduction of new interview categories without extensive changes to the code.

## **5.5.3 Observer Design Pattern**

* Application in Our System: The Observer pattern is integrated to oversee the live feedback and recommendation system. Upon the user’s completion of an interview, multiple system components receive notifications and proceed to update accordingly.
* This pattern is instrumental in realizing a heightened level of user interaction and system reactivity, which are crucial for delivering an engaging and dynamic user interface.

## **5.5.4 Model-View-Controller (MVC) Design Pattern**

* Application in Our System: The MVC architecture is vital to our web interface's organization, delineating the data model, the user interface, and the control logic into distinct sections. This demarcation is key to the system’s long-term manageability and growth potential.
* The MVC pattern encourages a well-defined separation of roles, simplifying the management, testing, and enhancement of the application.

# **5.6 AutoML / Serverless AI**

In the AI Interviewer platform, while there is no deployment of AutoML, the system is powered by sophisticated AI technologies such as OPENAI's GPT-4, the Whisper Model, and AWS Polly. These elements are integral to facilitating a smooth and intelligent user engagement, delivering an enhanced interview simulation experience.

## **5.6.1 Engaging with Users via OPENAI GPT-4**

* Intelligent Response Formulation: At the heart of our platform lies OPENAI GPT-4, a cutting-edge linguistic model, responsible for crafting responses that are both intelligent and relevant to the interview's context. It operates as the driving force behind the AI interviewer, tailoring dialogue to the nuances of each session.
* Advanced Language Processing: The superior language processing abilities of GPT-4 enable a conversation flow that closely mirrors human interaction, contributing to a more authentic and immersive mock interview experience.

## **5.6.2 Translating Speech to Text with OPENAI Whisper**

* Precise Voice Recognition: The system incorporates the Whisper model for its outstanding speech recognition accuracy, allowing it to precisely convert spoken words from users into text during the interactive sessions, which is fundamental for prompt analysis and feedback delivery.
* Interactive Enhancement: The Whisper model's proficient speech-to-text conversion ensures a fluid integration of user speech into the AI-powered interview framework, bolstering the system's interactivity.

## **5.6.3 Simulating Conversational Feedback with AWS Polly**

* Authentic Audio Feedback: The AI platform utilizes AWS Polly to transform AI-generated text into speech that sounds convincingly human. This capability is essential for providing a realistic interview ambiance, echoing the dynamics of an actual human exchange.
* Diverse Vocal Characteristics: With AWS Polly's array of vocal attributes, the system can offer feedback in various timbres and accents, accommodating a wide-ranging user demographic.

## **5.6.4 Leveraging Serverless Architecture for Robust Performance**

* Resource Optimization: The platform's serverless infrastructure is key to addressing the processing demands associated with running sophisticated AI models like GPT-4 and Whisper. This architecture facilitates resource optimization, ensuring the system remains responsive and efficient, even with increasing user numbers.

# **6. Experiments Section**

**6.1 Testing**

For the AI Interviewer platform, our approach to testing is methodical and targeted, with a particular focus on validating the application's functionality and user experience. This strategy is aimed at confirming that all elements of the system perform optimally, especially where user engagement and AI-generated responses are concerned.

## **6.1.1 Testing for Functionality**

* Interaction Verification Tests: These examinations are designed to confirm that user interactions with the platform occur seamlessly. This encompasses processes such as the uploading of resumes, the entry of job specifications, the selection of interview formats, and communication with the AI interviewer.
* Checking AI Response Validity: A pivotal part of our testing routine is to ascertain that the AI interviewer's replies are pertinent and precise. This entails assessing the relevance and correctness of the AI-generated inquiries and the feedback provided during practice interviews.

## **6.1.2 Assessing the User Interface**

* Workflow Functionality of the Interface: Tests on the user interface aim to certify its operational efficiency. This ensures the user can easily traverse through the application and that all the interactive components are functioning as expected.
* Aesthetics and User-Friendly Testing: We evaluate the visual components and the overall user-friendliness of the interface to ensure they adhere to design specifications and create an enjoyable user experience.

## **6.1.3 Comprehensive System Evaluation**

* Verification of the Entire Workflow: Comprehensive system evaluations encompass full- scale trials of the application's process flow, from initial input to the concluding AI- conducted interview and feedback stage, confirming the system's integrity and seamless operation.
* Testing in Realistic Conditions: We conduct tests under conditions that mirror actual usage scenarios to confirm the system's preparedness for genuine user engagement.

# **6.2 Model Deployment**

In the deployment phase of our AI-enhanced interview preparation tool, we utilize a sophisticated combination of GitHub Webhooks and Streamlit Cloud. This integration not only ensures the tool's seamless updates and management but also addresses crucial aspects of continuous integration, security, and user engagement.

## **Continuous Integration and Deployment (CI/CD) Workflow**

GitHub Webhooks for Automation: Our deployment strategy starts with setting up GitHub Webhooks in the project's repository. These webhooks are configured to trigger an automatic update process whenever there's a new commit or merge in the repository. This setup ensures that any changes in the codebase are immediately reflected in the deployed application, maintaining the tool’s current state with the latest features and fixes.

* **Streamlit Cloud for Hosting**

The application is hosted on Streamlit Cloud, a platform chosen for its ease of use and compatibility with our tool’s technology stack. Streamlit Cloud simplifies the deployment process, allowing our team to focus on developing the tool's core functionalities rather than on the intricacies of web hosting.

* **Monitoring and Updating the Tool**

Performance Monitoring: On Streamlit Cloud, we continuously monitor the application's performance. This involves tracking usage metrics, load times, and user feedback to ensure the tool operates smoothly and efficiently. Any performance issues are promptly addressed to maintain a high standard of user experience.

* **Iterative Updates and Maintenance**

Leveraging the CI/CD pipeline, we regularly update the tool, incorporating user feedback and new developments in AI technology. This iterative process ensures that the tool remains cutting-edge and tailored to the evolving needs of job applicants. Regular maintenance is scheduled to keep the application running optimally, with minimal downtime.

**Deployed Application:** <https://shail-ai-interview.streamlit.app/>

A screenshot of a computer

Description automatically generated

Fig 4: Deployed Application

# **7. Workflow**

## Uploading Resumes and Inputting Job Specifications

* + Participants begin by submitting their resumes and entering job descriptions that align with their interview training objectives.
  + This initial step is critical as it shapes the direction of the interview preparation process that follows.

## Extraction of Data from Resumes and Job Descriptions

* + The platform analyzes the provided resume and job description, pinpointing essential elements like qualifications, experiences, and required job competencies.

## Choosing the Interview Modality

* + Participants choose the interview modality they want to train for: Behavioral, Technical, or Resume-centric.
  + Each modality is customized to concentrate on distinct competencies and job-related necessities.

## AI-Managed Interview Interaction

* + The interview session commences, conducted by an AI interviewer driven by OPENAI's GPT-4, posing questions derived from the participant's resume, job details, and the chosen type of interview.
  + Respondents answer orally, and their spoken responses are transcribed into text through OPENAI's Whisper technology.

## Immediate Interaction and Feedback

* + The platform offers instant feedback and interaction, utilizing AWS Polly to create spoken feedback.
  + This engaging approach mimics a genuine interview setting, improving the participant's practice experience.

## Evaluation and Tailored Advice Post-Interview

* + Following the interview, the platform evaluates the participant's answers to provide customized advice and critiques.
  + This evaluation is vital for pinpointing areas for enhancement and steering the participant's ongoing practice.

## Progressive Learning and System Refinement

* + The system progressively refines its questioning and feedback based on user engagements, enhancing its effectiveness.
  + This continuous refinement guarantees that the interview preparation stays pertinent and efficacious as needs and contexts evolve.

**7.1 Data Flow Diagram**

The Data Flow diagram illustrates the steps of an interactive interview preparation system where the user goes through a series of actions to complete an interview simulation:

1. **Start**: The process begins.
2. **User Selects Interview Type**: The user chooses the type of interview they wish to prepare for. There are three options:
   * Technical
   * Behavioral
   * Resume-based.
3. **Input Query**: After selecting the interview type, the user inputs a query. This could be a question or a request for an interview question.
4. **Process Input**: The user's input is then processed by understanding the query and determining the best response or action.
5. **Generate Question using GPT-4**: For the input query, an interview question is generated using the GPT-4 model. This step implies that the system uses OpenAI's GPT-4 to create relevant interview questions based on the type of interview chosen.
6. **Present Question to User**: The generated interview question is then presented to the user.
7. **User Submitted Response**: The user responds to the interview question.
8. **Provide Feedback**: After the user submits their response, the system provides feedback. This feedback could be based on the content of the response, the style, or other metrics relevant to the interview preparation.
9. **Ask Another Question (Optional Loop)**: If there is a need to ask another question (which might depend on the user's choice or the system's design for a comprehensive interview session), the process loops back to the "Input Query" step.
10. **End**: If there is no need to ask another question, the process ends.

This workflow allows for an iterative interview practice session where a user can go through multiple rounds of questioning and feedback to improve their interview skills.

A diagram of a process flow

Description automatically generated

Fig 5: DFD

# **8. Active Learning or Feedback Loop**

The AI Interviewer system's adoption of active learning and iterative feedback mechanisms is designed to perpetually refine its efficacy and precision. This evolving methodology leverages interactions and system feedback, enabling ongoing enhancements to the AI algorithms and user experience.

## **Continuous Learning within AI Frameworks**

* + Leveraging User Engagement for Enhancement: Each engagement with the user, particularly their inputs and the system's feedback during interviews, is harnessed as an opportunity for learning. The AI entities responsible for crafting questions and providing feedback are engineered to evolve and optimize based on these interactions.
  + Progressive Model Enhancement: The system forgoes a one-time training approach in favor of a progressive training model that integrates new data regularly. This ensures that the AI models remain up-to-date and attuned to the changing requirements of users and the job market.

## **Evaluation of Data for Insights**

* + Assessment of User Interactions and System Feedback: The system conducts a thorough analysis of the data gathered, focusing on understanding user tendencies, challenges encountered, and the areas where users commonly require additional practice.
  + Identification of Areas for Refinement: Through this evaluative process, the system identifies specific aspects for potential enhancement, which may include the accuracy of the AI models, the design of the user interface, or the comprehensiveness of the interview questions posed.

# **9. Screenshots**

**A screenshot of a computer

Description automatically generated**

Fig 6: Homepage (I)

**A screenshot of a computer

Description automatically generated**

Fig 7: Homepage (II)

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generatedFig 8: Technical Interview (I)

A screenshot of a computer

Description automatically generatedFig 9: Technical Interview (II)

Fig 10: Technical Interview (III)

A screenshot of a computer

Description automatically generated

Fig 11: Interview GuideA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedFig 12: Feedback (I)

Fig 13: Feedback (II)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedFig 14: Resume Based Interview (I)

A screenshot of a computer

Description automatically generatedFig 15: Resume Based Interview (II)

Fig 16: Resume Based Interview (III)

A screenshot of a computer

Description automatically generated

A screenshot of a black screen

Description automatically generatedFig 17: Behavioral Interview (I)

Fig 18: Behavioral Interview (II)

# **10. Conclusion**

## **10.1 Summary of Key Results and Learnings**

This project has demonstrated significant advancements in the realm of interview preparation using AI technologies. By integrating the OpenAI GPT-4 model for natural language processing and the Whisper model for speech-to-text conversion, I have successfully created a platform that offers personalized interview training. My key results include.

* **Effective Personalization**: The ability to tailor interview questions and feedback based on individual resumes and specific job descriptions, leading to a more relevant and targeted interview preparation experience.
* **Advanced Data Processing**: The use of OPENAI Embeddings and FAISS database for similarity searches has enabled us to effectively match interview content with the candidate's profile and job requirements.
* **Enhanced User Interaction**: The incorporation of speech-to-text technology has allowed for a more natural and interactive interview practice environment, closely mimicking real- life scenarios.
* **Comprehensive Evaluation**: Our system not only generates pertinent interview questions but also provides insightful feedback, scores, and summaries, aiding candidates in understanding and improving their performance.

## **10.2 Learnings**

## From this project, I have learned the importance of integrating diverse AI technologies to create a holistic and user-friendly system. The combination of language understanding, speech processing, and data analysis has proven to be highly effective in simulating realistic interview environments. Additionally, I have gained insights into the nuances of personalized training, understanding that a one-size-fits-all approach is less effective in interview preparation.

## **10.3 Future Extensions and New Applications**

## **Looking forward, there are few ideas for extending this project.**

1. **Broader Domain Coverage**: Expanding the range of job roles and industries covered by our tool to cater to a wider audience.
2. **Career Development Features**: Adding modules for career advice, skill development, and job market trends analysis.
3. **Language Expansion**: Incorporating multiple languages to support non-English speaking users.
4. **Educational Sector Applications**: Adapting the tool for educational purposes, helping students in universities and colleges prepare for their career paths.

By building upon these ideas, the project can evolve to meet the changing demands of the job market and continue to support job seekers in their interview preparation journey more effectively.

# **11. References**

1. Shlomo Kashani, M.Sc, QMUL, UK. (2022). “DEEP LEARNING INTERVIEWS”, “Deep Learning Interviews: Hundreds of fully solved job interview questions from a wide range of key topics in AI”. Available: https://arxiv.org/abs/2201.00650v2
2. Bird, S., Klein, E., & Loper, E. (2009). "Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit," O'Reilly Media, Inc. [Online]. Available: https://[www.nltk.org/book/](http://www.nltk.org/book/)
3. Brown, T. et al. (2020). "Language Models are Few-Shot Learners," OpenAI. [Online]. Available: https://arxiv.org/abs/2005.14165
4. Johnson, J., Douze, M., & Jégou, H. (2017). "FAISS: A library for efficient similarity search and clustering of dense vectors," arXiv:1702.08734. [Online]. Available: https://arxiv.org/abs/1702.08734
5. McKinney, W. (2010). "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference. [Online]. Available: https://conference.scipy.org/proceedings/scipy2010/mckinney.html
6. OpenAI. (2023). "GPT-4: Language Models are Unsupervised Multitask Learners," OpenAI.com. [Online]. Available: https://[www.openai.com/gpt-4/](http://www.openai.com/gpt-4/)
7. Streamlit. (2023). "Streamlit: The fastest way to build and share data apps," Streamlit.io. [Online]. Available: https://[www.streamlit.io/](http://www.streamlit.io/)