A Major Project Report

on

Comprehensive Placement Assistance Platform

submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

CSE (DATA SCIENCE)

by

Animi Chandan (21211A6705)

Lokesh Akoju (21211A6703)

MVS Sriharsha (21211A6732)

Shrishail Patil (21211A6760)

Sravan Sirigadde (21211A6761)

Koushik Krishna (21211A6762)

Under the guidance of

Krutibash Nayak,
Assistant Professor

DEPARTMENT OF CSE (DATA SCIENCE)

B. V. RAJU INSTITUTE OF TECHNOLOGY

(UGC Autonomous, Accredited by NBA & NAAC)

Vishnupur, Narspur, Medak(Dist.), Telangana State, India - 502313

2024 - 2025

B. V. Raju Institute of Technology

(UGC Autonomous, Accredited by NBA & NAAC)
Vishnupur, Narsapur, Medak (Dist.),
Telangana State, India – 502313

DEPARTMENT OF CSE (Data Science)

CERTIFICATE

This is to certify that the Major Project entitled **"Comprehensive Placement Assistance Platform"**, being submitted by

Animi Chandan (21211A6705)
Lokesh Akoju (21211A6703)
MVS Sriharsha (21211A6732)
Shrishail Patil (21211A6760)
Sravan Sirigadde (21211A6761)
Koushik Krishna (21211A6762)

In partial fulfillment of the requirements for the award of degree of BACHELOR OF TECHNOLOGY in CSE (DATA SCIENCE) to B. V. RAJU INSTITUTE OF TECHNOLOGY is a record of bonafide work carried out during a period from **December 2024 to April 2025** by them under the guidance of **Mr. Krutibash Nayak**, Assistant Professor, CSE(Data Science) Department.

This is to certify that the above statements made by the students are correct to the best of my knowledge.

Krutibash Nayak
Assistant Professor

The Project Viva-Voce Examination of this team has been held on _____.

Dr. K. Purnachand

External Examiner

Professor & HoD-CSE(Data Science)

VISHNU
UNIVERSAL LEARNING

CONTENTS

Candidate's Declaration **i**

Acknowledgement **ii**

Abstract **iii**

List of Figures **iv**

1. INTRODUCTION **1**

1.1 Motivation 2

1.2 Problem Definition 4

1.3 Objective of Project 5

1.4 Limitations of Project 6

1.5 Organization of Documentation 7

2. LITERATURE SURVEY **8**

2.1 Introduction 9

2.2 Existing System 10

2.3 Disadvantages of Existing system 11

2.4 Proposed System 12

3. SYSTEM OVERVIEW **13**

3.1 Introduction 14

3.2 Software Requirement Specification 14

3.2.1 User requirements 14

3.2.2 Software requirements 15

3.2.3 Hardware requirements 15

3.3 Algorithms and Techniques 16

4. DESIGN **19**

4.1 Introduction 20

4.2 High Level System Design 20

4.2.1 Core Design Principles 20

4.2.2 Workflow Overview 20

4.3 Component-Level Design 21
4.3.1 User Interface 21
4.3.2 Backend Services 22
4.4 DFD 23
4.5 System Architecture 25
4.6 Technology Stack 26

5. SYSTEM EVALUATION 28
5.1 Evaluation methodology 29
5.2 Retrieval Performance 29
5.3 Reranking Performance 30
5.4 Result Analysis 31
5.5 Comparison with Conventional Methods 32
5.6 Limitations and Challenges 33
5.7 Key Contributions 34
7. CONCLUSION & FUTURE WORK 35
8. REFERENCES 38

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Comprehensive Placement Assistance Platform"** in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology and submitted in the Department of Computer Science and Engineering, B. V. Raju Institute of Technology, Narsapur is an authentic record of my own work carried out during a period from **December 2024 to April 2025** under the guidance of **Mr. Krutibash Nayak**, Assistant Professor. The work presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Animi Chandan (21211A6705)

Lokesh Akoju (21211A6703)

MVS Sriharsha (21211A6732)

Shrishail Patil (21211A6760)

Sravan Sirigadde (21211A6761)

Koushik Krishna (21211A6762)

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion. Whatever we have done is due to such guidance and assistance. We would not forget to thank them.

We thank **Mr. Krutibash Nayak** for guiding us and providing all the support in completing this project. We are thankful to **Mr. Purnachandra Rao**, our section project coordinator, for supporting us in doing this project. I/We thank the person who has our utmost gratitude is **Dr. K. Purnachand**, Head of CSE (Data Science) Department.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the staff members of CSE (Data Science) Department.

Animi Chandan (21211A6705)
Lokesh Akoju (21211A6703)
MVS Sriharsha (21211A6732)
Shrishail Patil (21211A6760)
Sravan Sirigadde (21211A6761)
Koushik Krishna (21211A6762)

Comprehensive Placement Assistance Platform

ABSTRACT

The Comprehensive Placement Assistance Platform introduces an advanced chatbot system that leverages the Retrieval-Augmented Generation (RAG) paradigm to deliver accurate, reliable, and context-aware placement-related information to students. Traditional methods of placement preparation often involve scattered data sources, manual searches, and unreliable information, leading to inefficiencies and student frustration. This project addresses these issues by creating a centralized, AI-powered platform capable of understanding and responding to complex user queries. At the heart of the system lies the integration of cutting-edge technologies including Pinecone, a vector database optimized for high-dimensional data retrieval, and the multilingual-e5-large model, which facilitates robust vector embeddings capable of processing queries in multiple languages. To ensure that the most relevant results are surfaced, the system employs the BGE reranker, a neural reranking model that improves the relevance of retrieved data by assigning relevance scores based on query-document similarity. The platform also integrates several Natural Language Processing (NLP) techniques such as Named Entity Recognition (NER) and contextual query augmentation to enhance query understanding. These techniques allow the system to identify company names, understand context across conversation turns, and refine vague queries for better result accuracy. Llama 3.1 8B Instruct, a large language model, is used to augment and rephrase user queries as well as generate natural, context-aware responses based on the reranked results. Together these components create a chatbot that not only mimics human conversation but does so with factual backing and contextual precision. The platform significantly reduces the time and effort required for placement preparation by providing quick access to categorized information like company descriptions, roles, eligibility criteria, recruitment process, interview stages, and preparation resources. The ultimate goal of the project is to empower students with a smart, efficient, and user-friendly tool that enhances their placement readiness and boosts their confidence during the recruitment process.

Keywords - Chatbot, RAG, Pinecone, Vector Embeddings, BGE Reranker, Llama 3.1 8B, NLP, Placement Preparation, Information Retrieval.

LIST OF FIGURES

S. NO.	FIGURE DESCRIPTION	PAGE NO.
1.	RAG Flowchart	**17**
2.	User interface	**21**
3.	DFD Level 0	**23**
4.	DFD Level 1	**24**
5.	DFD Level 2	**24**
6.	System Architecture	**25**
7.	Performance Comparison for Direct and Indirect Questions	**32**

Chapter – 1

INTRODUCTION

1. Introduction

1.1 Motivation

The primary motivation behind the development of the Comprehensive Placement Assistance Platform stems from the widespread challenges faced by students during their placement preparation journey. In today's highly competitive academic environment, securing a job offer through campus placements is a crucial milestone for most students, especially in technical fields. However, the process of gathering reliable and structured information about different companies, their recruitment processes, job roles, eligibility criteria, and interview experiences is often fragmented and tedious. Students typically rely on a mix of outdated college repositories, informal peer discussions, scattered blog posts, and social media forums, which may be inaccurate or inconsistent. This lack of a centralized and trustworthy platform leads to inefficient preparation, confusion, and sometimes even missed opportunities. Moreover, conventional chatbots and search engines fall short when it comes to answering nuanced queries or understanding the context of user needs. These limitations motivated our team to conceptualize and build an intelligent solution that not only automates information retrieval but also provides personalized and context-aware responses. By leveraging recent advancements in Natural Language Processing, vector databases, and generative AI models, we envisioned a system that bridges the gap between information overload and actionable insight. Our motivation was further fueled by the desire to democratize access to quality placement resources, ensuring that every student regardless of their background has an equal opportunity to prepare effectively, make informed decisions, and achieve their career aspirations with confidence.

The Retrieval-Augmented Generation (RAG) framework represents a cutting-edge advancement in natural language processing (NLP), combining the precision of retrieval-based systems with the fluency of generative models. Unlike traditional chatbots that rely solely on static, rule-based responses or generative models that can sometimes provide inaccurate outputs, RAG

intelligently retrieves relevant information from a knowledge base and then generates coherent, context-aware responses. This hybrid approach significantly enhances response accuracy, relevance, and human-likeness making it especially powerful for complex queries.

In the realm of placement preparation, students often struggle to find reliable, centralized, and up-to-date information. Conventional methods such as browsing company websites, forums, or social media are time-consuming and often yield inconsistent or outdated data. Similarly, traditional chatbots fall short due to their reliance on predefined rules or basic generative techniques, which limit their ability to understand nuanced queries or search large datasets effectively.

The RAG model addresses these shortcomings by employing advanced NLP techniques and leveraging vector databases for high-quality information retrieval and response generation. In this project, we propose a placement-focused chatbot built on the RAG framework. It utilizes Pinecone as the vector database, integrated with the multilingual-e5-large model to generate embeddings capable of handling multilingual queries. To enhance the relevance of responses, the BGE reranker prioritizes the most pertinent information. Additionally, techniques like Named Entity Recognition (NER) and query augmentation improve the system's ability to understand and respond to complex questions.

Our chatbot is purpose-built to solve major challenges in placement preparation: the absence of a centralized information platform, inefficiencies in manual data collection, and the unreliability of scattered online resources. Through a simple and user-friendly interface, it offers students easy access to accurate company details, interview prep materials, and other essential placement-related resources. To further improve the quality and contextuality

of generated responses, the system incorporates the Llama 3.1 8B instruct model for both query augmentation and final output generation.

Unlike rule-based systems that require extensive manual rule-setting and fail to scale with complex inputs, or generative models that may sound fluent but lack factual grounding, RAG blends the best of both approaches. It ensures that answers are not only coherent but also based on reliable, up-to-date information pulled from a curated knowledge base.

1.2 Problem Definition

The primary challenges faced by Alzheimer's patients include:

- Difficulty recognizing family members – As Alzheimer's progresses, patients often struggle to identify even their closest relatives, leading to confusion, anxiety, and emotional distress. This can result in withdrawal from social interactions and increased dependence on caregivers.
- Tendency to wander and get lost – Many Alzheimer's patients experience disorientation, which causes them to wander away from home or familiar environments. This puts them at risk of accidents, injuries, or getting lost, requiring constant supervision from caregivers.
- Inability to manage daily routines – Simple daily activities, such as taking medication on time, maintaining hydration, and following a structured schedule, become challenging. Patients may forget to eat meals, miss important medical appointments, or struggle with basic self-care, leading to a decline in overall well-being.
- Increased caregiver stress – Family members and professional caregivers often bear a heavy burden, managing both the physical and emotional needs of the patient. Continuous supervision, managing medical needs, and responding to emergencies can lead to caregiver burnout, affecting their mental health and quality of life.

The proposed system addresses these challenges using technology-driven solutions that integrate artificial intelligence (AI) and machine learning (ML). The system includes facial recognition to help patients identify family members, GPS-based location tracking to prevent wandering, and AI-powered reminders for medication and daily tasks. Additionally, it features distress call detection, allowing caregivers to be alerted in emergencies. Through a user-friendly interface, both patients and caregivers can access personalized support, making Alzheimer's care more efficient and reducing stress for everyone involved.

4.3 Objective of Project

To develop a centralized chatbot platform that provides accurate, real-time, and reliable information related to campus placements.

- To implement the Retrieval-Augmented Generation (RAG) approach for combining information retrieval with context-aware response generation.
- To utilize Pinecone as a vector database for efficient and scalable semantic search over high-dimensional placement-related data.
- To integrate the multilingual-e5-large model for generating language-agnostic vector embeddings and supporting multilingual queries.
- To apply BGE reranker to prioritize the most relevant search results and improve response precision.
- To enhance user queries using query augmentation techniques with the Llama 3.1 8B Instruct model for better context understanding.
- To automate the retrieval of company-specific information like job roles, eligibility, recruitment processes, and interview questions.
- To create a user-friendly and intuitive interface that allows seamless interaction between students and the chatbot system.
- To reduce the time and effort students spend on placement preparation by providing them with instant, personalized answers.

- To address the shortcomings of traditional search and rule-based chatbot systems in handling complex and multi-intent queries.
- To ensure the chatbot system is scalable, extensible, and adaptable to include future enhancements like real-time updates, voice input, or personalization.
- To support continuous improvement of the knowledge base by structuring data into categories like company description, role expectations, and preparation tips.

4.4 Limitations of Project

The chatbot currently depends on static, pre-collected datasets, which may become outdated over time without regular updates.

- It is limited to answering queries based only on specific companies present in the knowledge base and cannot generalize to all organizations.
- Real-time integration with live job portals or company websites is not yet implemented, restricting its ability to provide up-to-date job openings or salary trends.
- The chatbot may struggle with extremely complex or ambiguous queries that require multi-layered reasoning or deep contextual understanding.
- While the system supports multilingual input through embeddings, the user interface and responses are primarily in English, limiting accessibility for some users.
- There is a dependence on the quality and completeness of input data; any gaps or inaccuracies in the dataset directly affect the reliability of responses.
- The chatbot does not yet offer personalized recommendations based on a student's background, preferences, or skills.
- Current interaction is text-only; voice input/output features for accessibility and user convenience are not yet implemented.

1.5 Organization of Documentation

Chapter 1: Introduction

Presents the project's background, objectives, and limitations in placement preparation.

Chapter 2: Literature Survey

Reviews existing research on chatbot technologies, NLP techniques, and information retrieval methods for educational applications.

Chapter 3: Analysis

Analysis on Chatbot retrevial model.

Chapter 4: Design

This section outlines the architecture of the proposed RAG-based chatbot system.

Chapter 5: Conclusion

Conclusion and future enhancements.

References: Lists all the references used in the project.

Chapter – 2

LITRATURE SURVEY

2. Literature Survey

2.1 Introduction

The development of chatbots and information retrieval systems has been a focal point of research in natural language processing (NLP), with various approaches proposed to enhance their accuracy, efficiency, and usability. Traditional chatbots, which rely on rule-based systems or simple generative models, have been widely used in customer service, education, and other domains. However, these systems are often limited in their ability to handle complex queries and provide accurate, context-aware responses. Rule-based systems, for instance, require extensive manual effort to create and maintain, and they struggle to handle queries that fall outside their predefined rules. On the other hand, generative models, while capable of producing fluent responses, often lack access to reliable knowledge bases, leading to inaccurate or irrelevant answers.

To address these limitations, researchers have explored hybrid approaches that combine retrieval-based and generation-based methods. One such approach is the Retrieval-Augmented Generation (RAG) model, which integrates a retrieval mechanism to fetch relevant information from a knowledge base and a generative model to synthesize coherent and context-aware responses. The RAG model has been successfully applied in various domains, including healthcare, customer service, and education, demonstrating its effectiveness in handling complex queries and providing accurate, reliable answers. In the context of placement preparation, the RAG approach offers a promising solution to the challenges of information retrieval and response generation, enabling students to access reliable and up-to-date information quickly and efficiently.

The RAG model, Introduced by Lewis et al., combines the strengths of retrieval-based and generation-based models to provide accurate and contextually relevant responses. The retrieval component of the RAG model uses a dense retriever to fetch relevant documents from a knowledge base,

while the generation component uses a transformer-based model to synthesize coherent responses based on the retrieved information. This hybrid approach ensures that the chatbot not only generates human-like responses but also grounds them in factual and up-to-date information, making it highly effective for complex query resolution.

In the context of placement preparation, the RAG model can be used to retrieve relevant information from a large dataset of company details, interview preparation resources, and other placement-related data. The generative component of the model can then synthesize coherent and context-aware responses, providing students with accurate and reliable information. This approach has been shown to outperform traditional chatbots in terms of accuracy, relevance, and user satisfaction.

Vector databases, such as **Pinecone**, have gained popularity in recent years due to their ability to handle large-scale data and support multilingual embeddings. The **multilingual-e5-large model**, in particular, has been shown to perform well in various NLP tasks, including text classification, information retrieval, and question answering. Reranking is a critical component of information retrieval systems, as it ensures that the most relevant information is prioritized. The **BGE reranker**, for instance, has been shown to improve the relevance of retrieved information, making it a valuable tool for information retrieval systems.

2.2 Existing Systems

The existing systems primarily include rule-based chatbots and basic generative chatbots that are widely used in customer service, education, and other information delivery applications. These systems typically follow pre-defined rules or templates and provide responses based on keyword matching or limited natural language understanding. While some may incorporate simple NLP features, most lack deeper semantic understanding and

contextual awareness. Students preparing for placements currently rely on scattered sources like college seniors, discussion forums, blogs, and social media platforms to gather information about companies, roles, and interview processes. These methods are not only time-consuming but also lack accuracy and consistency. Additionally, conventional systems do not support dynamic query interpretation or handle multilingual inputs efficiently.

2.3 Disadvantages of Existing Systems

Despite advancements in chatbots, several critical limitations remain:

- Lack of Context Understanding: Rule-based chatbots cannot handle complex, multi-part, or ambiguous queries.
- Manual Dependency: Students spend a significant amount of time manually collecting data from unreliable or outdated sources.
- Inflexibility: Existing systems do not adapt to user intent beyond fixed responses.
- Limited Scalability: Cannot accommodate a growing database of diverse companies or new placement trends.
- No Intelligent Ranking: They lack reranking capabilities, so the most relevant answers may not be prioritized.
- Static Knowledge Base: Most do not dynamically fetch or update information from external trusted sources.
- No Query Augmentation: Queries are taken at face value without enhancement or rephrasing to improve retrieval quality.

2.4 Proposed System

The proposed solution is an AI-powered chatbot based on the Retrieval-Augmented Generation (RAG) architecture that addresses the shortcomings of traditional systems. It combines a dense retrieval mechanism (using Pinecone and multilingual-e5-large model) with a powerful generative model (Llama 3.1 8B instruct) to provide context-aware, accurate, and real-time responses to placement-related queries. The system uses Named Entity Recognition (NER) to extract key entities from user input and query augmentation techniques to rephrase or expand the query, enhancing the quality of retrieved information. Results are reranked using the BGE reranker, ensuring the most relevant answers are surfaced. The chatbot supports multilingual input, making it inclusive for a diverse student base. Structured data categorized into company descriptions, roles, eligibility, interview rounds, and additional tips ensures that users receive well-organized responses tailored to their exact needs. The platform is user-friendly, scalable, and designed to evolve with changing placement trends.

Chapter – 3

SYSTEM OVERVIEW

3. System Overview

3.1 Introduction

Final-year undergraduate students often face immense pressure while preparing for campus placements due to the lack of organized, reliable, and updated information on companies, job roles, interview processes, and preparation strategies. While the internet is flooded with content, students frequently struggle to extract relevant and verified details in a timely manner.

This project proposes an AI-powered chatbot system designed specifically to assist students in their placement preparation. By leveraging the Retrieval-Augmented Generation (RAG) architecture, the system combines the benefits of information retrieval and generative models to provide accurate, contextual, and multilingual responses. The system aims to reduce manual search effort, improve preparation efficiency, and empower students with tailored support.

3.2 Software Requirement Specification

The Software Requirement Specification (SRS) outlines the technical and functional requirements necessary to build the RAG-based chatbot system. It identifies the primary user needs, system components, and tools required for successful implementation.

3.2.1 User Requirements

The system is intended for two main types of users:

Students

- Need accurate and consolidated information about companies, job roles, and interview procedures.
- Benefit from personalized query handling and preparation guidance.

Placement Coordinators / Admins

- Need control over the chatbot's knowledge base and content updates.
- Require analytics to understand student query trends and needs.
- Benefit from a dashboard to manage system usage and improve content quality.

Key User Requirements:

- Students should receive precise and up-to-date responses on placement-related queries.
- Users must be able to ask questions in natural language (text input).
- Admins should have backend access for managing document corpus and content updates.

3.2.2 Software Requirements

- Programming Language: Python (for backend and model integration)
- Vector Store: FAISS / Pinecone for document embeddings
- Frontend: Flutter
- Frameworks: Flask or Fast API for backend APIs
- Machine Learning Models: Sentence Transformers for embedding, and a fine-tuned generative model i.e. Llama.

3.2.3 Hardware Requirements

Minimum Hardware Configuration:

- Processor: Intel i5 / Ryzen 5 or higherRAM: 8GB (recommended for smooth ML model execution)
- RAM: 8GB (16GB recommended for running embedding models smoothly)

- Storage: 256GB SSD or more (for storing documents, logs, and vector indexes)
- Internet Connection: Stable connection for backend API and external library access

3.3 Algorithms and Techniques

Retrieval-Augmented Generation (RAG)

The Retrieval-Augmented Generation (RAG) framework lies at the heart of the chatbot architecture. It combines the power of information retrieval systems with the contextual understanding of generative language models. When a user submits a query, the RAG model first retrieves relevant documents from a knowledge base and then generates a natural language response by using the retrieved context. This hybrid approach ensures that responses are grounded in factual data while remaining fluent and human-like, overcoming the limitations of both traditional rule-based systems and standalone generative models.

The RAG workflow involves the following key steps:

1. User Query Input: The process begins when a user submits a natural language query to the chatbot. This query may be direct or vague, and the system is designed to handle both by leveraging retrieval and generation.
2. Query Embedding: The user query is converted into a dense vector representation using a pre-trained embedding model such as multilingual-e5-large. This vector captures the semantic meaning of the query, enabling the system to match it with similar content in the knowledge base.
3. Document Retrieval: The generated query vector is used to search a vector database like Pinecone, which contains pre-embedded documents related to placement topics (e.g., company details, interview

rounds, eligibility criteria). The database returns the top-N most relevant documents based on cosine similarity between vectors.

4. Reranking: To refine the retrieved results, a BGE reranker is used to reorder the documents based on their contextual relevance to the query. This step ensures that the most useful documents are prioritized for response generation.
5. Response Generation: The generative model (e.g., LlaMA 3.1 8B Instruct) takes the original query and the top retrieved documents as input. It synthesizes a fluent and informative answer by combining the context from the retrieved texts with its language generation capabilities.

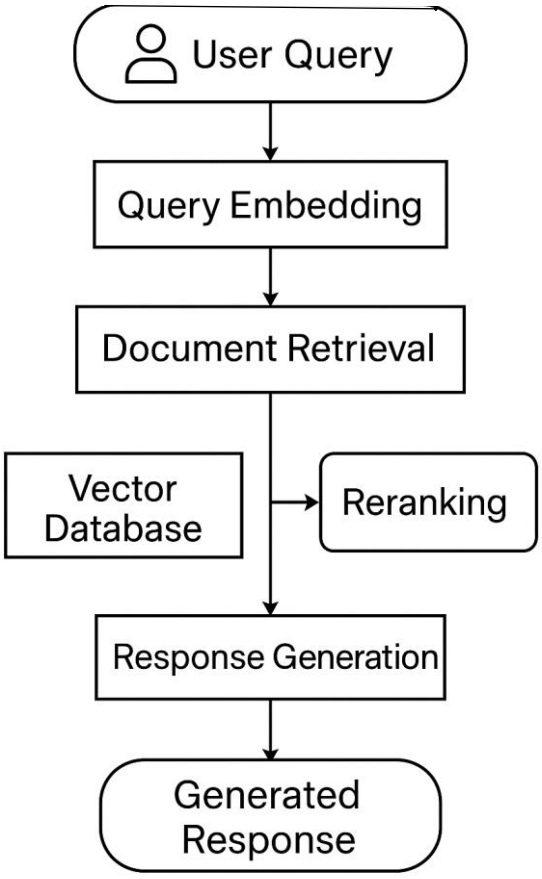

Figure 1: RAG Flowchart

Advantages of Random Forest

- Enhanced Accuracy: RAG enhances the accuracy of generated responses by retrieving relevant documents from a knowledge base and feeding them to the language model. This ensures that answers are grounded in factual, up-to-date information.
- Knowledge Augmentation Without Retraining: Unlike traditional models that require retraining to incorporate new knowledge, RAG can update its knowledge base dynamically. This allows it to adapt to new data quickly without needing expensive model retraining.
- Reduced Hallucinations: Since RAG grounds its generation in real data retrieved from external sources, it significantly reduces the chances of hallucinated (made-up or incorrect) information—a common issue with generative models.
- Scalability and Flexibility: RAG systems can scale across domains by simply changing the underlying documents in the retrieval database. It's flexible and adaptable to different topics or business use cases.
- Improved Contextual Understanding: By retrieving contextually relevant information, RAG allows the model to generate more coherent and contextually aware responses, which improves overall user satisfaction.
- Transparency and Traceability: RAG systems often provide citations or sources of retrieved documents, making it easier to trace the origin of the generated response and increasing the trustworthiness of the system.

Chapter – 4

DESIGN

4. Design

4.1 Introduction

The design phase focuses on structuring the RAG-based Placement Preparation Chatbot to be user-friendly, scalable, and efficient. The system ensures seamless interaction between students, recruiters, and AI-powered retrieval mechanisms while maintaining accuracy, relevance, and real-time accessibility.

This section outlines the high-level design concepts and how different components interact to achieve the goal of providing accurate, reliable, and instant placement-related information to students.

4.2 High-Level System Design

4.2.1 Core Design Principles

- Modularity: Separates retrieval, reranking, and generation for flexibility.
- Accuracy: Combines dense retrieval + reranking for precision.
- User-Centricity: Simple UI with natural language interaction.

4.2.2 Workflow Overview

1. **User Input**: Student submits a placement-related query (e.g., *"Amazon SDE interview process"*).
2. **Query Processing**:
    - NLP preprocessing (spell check, entity extraction).
    - Query augmentation using Llama 3.1 8B.
    - Vectorization via multilingual-e5-large.

3. **Retrieval Phase**:
    - Pinecone returns top 3 documents (JSON-structured data).
    - Cosine similarity filters irrelevant results.
4. **Reranking Phase**:
    - BGE model reorders results by contextual relevance.
5. **Response Generation**:
    - Llama 3.1 8B synthesizes a natural language answer from reranked documents.
6. **Output**: Clean, formatted response with citations (e.g., company-specific recruitment steps).

4.3 Component-Level Design

4.3.1 User Interface (Frontend)

- Flutter-based interface with:
    - Query input box (text/NLP-supported).
    - Response display panel (formatted answers).

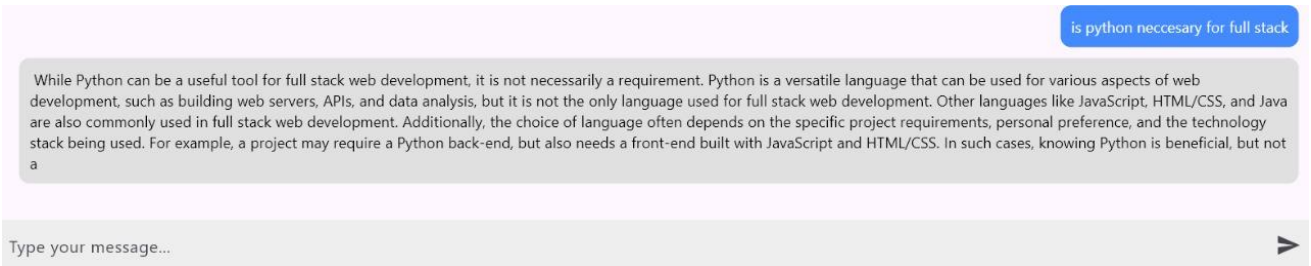

Figure 2: User interface

4.3.2 Backend Services

1. **Query Processing Module**:
    - **NER Model**: Identifies companies, roles (e.g., “Google” → “Software Engineer”).
    - **Query Augmenter**: Expands “TCS hiring” → “TCS recruitment process 2024, eligibility criteria.”
    - **Vectorizer**: Converts text to embeddings (multilingual-e5-large).
2. **Retrieval System**:
    - **Pinecone DB**: Stores vectors of structured placement data (6 categories).
    - **Top-3 Retrieval**: Returns documents with cosine similarity >0.75.
3. **Reranking Module**:
    - **BGE Model**: Re-sorts results by relevance (e.g., prioritizes “interview rounds” over “company history”).
4. **Response Engine**:
    - **Llama 3.1 8B**: Generates concise answers using retrieved snippets.
5. **Knowledge Base**:
    - **Structured JSON**: Organized by company, role, eligibility, etc.
    - **Data Pipeline**: Weekly updates from verified sources (company portals, career sites).

4.4 DFD

DFD Level 0

Figure 3: DFD Level 0

DFD Level 1

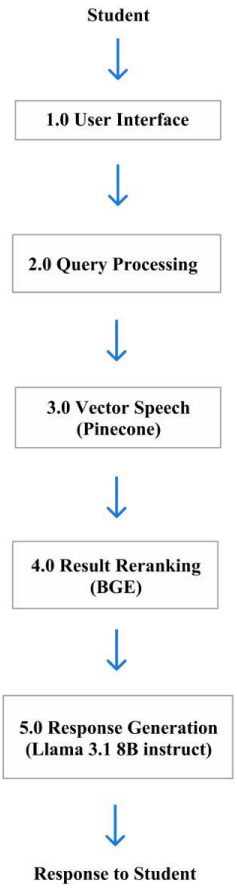

Figure 4: DFD Level 1

DFD Level 2

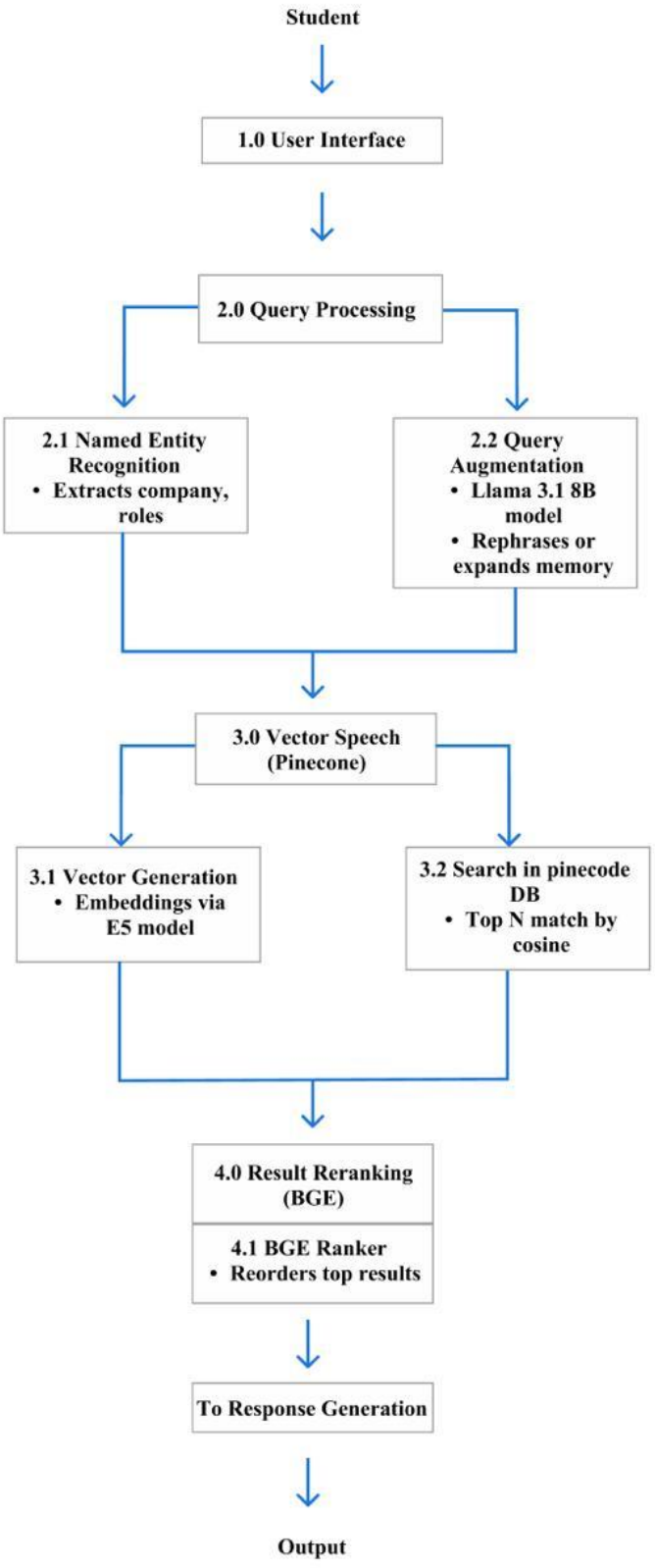

Figure 5: DFD Level 2

4.5 System Architecture

The overall system architecture is illustrated in Figure 4. The architecture consists of the following components:

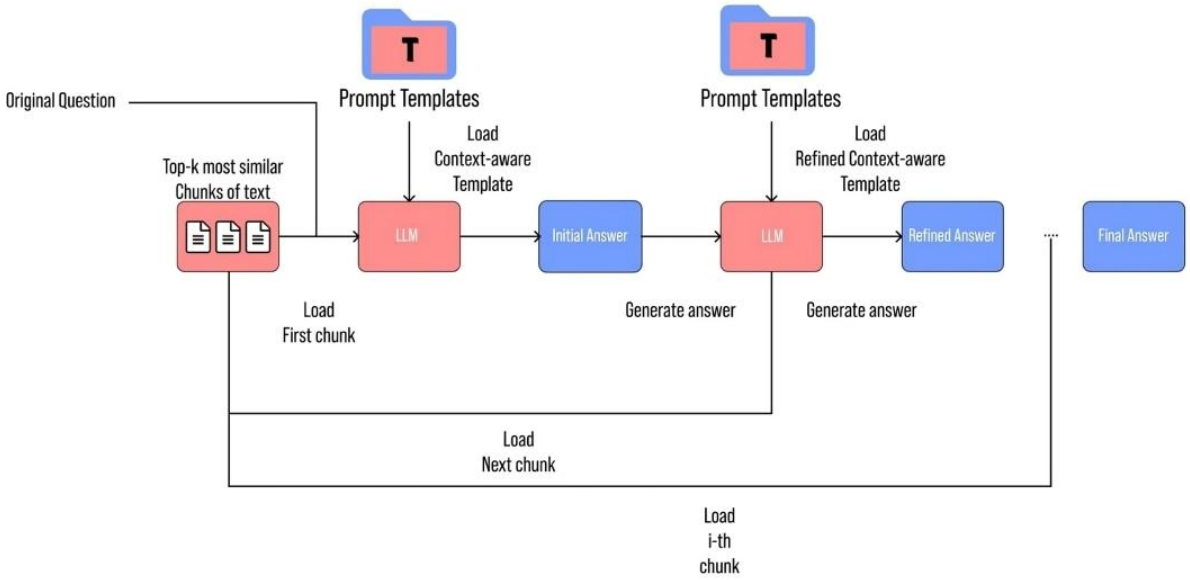

Figure 6: Architecture Diagram

1) User Interface: The front-end interface where users submit their queries and receive responses.

2) Query Preprocessing Module: Handles NLP tasks such as NER and query augmentation.

3) Vector Database (Pinecone): Stores and retrieves vector embeddings of the knowledge base.

4) Retrieval and Reranking Module: Retrieves relevant in formation and reranks it using the BGE reranker.

5) Response Generation Module: Generates coherent and context-aware responses using the Llama 3.1 8B instruct model.

4.6 Technology Stack

Frontend

The frontend of the chatbot is developed using Flutter, an open-source UI toolkit by Google that enables the creation of natively compiled applications for mobile, web, and desktop from a single codebase. Flutter's reactive framework provides a smooth and flexible user experience, ideal for building an intuitive and responsive chat interface. This allows students to interact seamlessly with the chatbot across platforms, whether on mobile devices or web browsers, enhancing accessibility and convenience.

Backend

The backend is built using Python with Flask, a lightweight and flexible micro web framework. Flask handles API routing, request processing, and interaction with other components of the RAG pipeline. It serves as the central controller that manages the flow from user query intake to final response delivery. The modularity of Flask makes it ideal for integrating with language models, vector databases, and preprocessing modules, enabling a clean and efficient backend architecture.

Language Model (LLM)

For generating responses, the system uses LLaMA 3.1 8B Instruct, a powerful instruction-tuned language model. It plays a dual role in the system: first, it is used for query augmentation, where it enhances or reformulates user queries for better clarity and coverage; and second, it is used in the response generation stage, synthesizing natural, informative, and contextually rich replies based on the retrieved content. This model ensures the chatbot maintains coherence while delivering factually grounded answer.

Embedding Model

To represent textual data semantically, the system uses the Multilingual-e5-large model for generating embeddings. This model converts both user queries and corpus documents into high-dimensional vector representations that capture their meaning across languages. Its multilingual capability allows the chatbot to process and retrieve relevant information from the knowledge base, even when users input queries in different languages or phrasing styles.

Vector Database

All document embeddings are stored and searched using **Pinecone**, a specialized vector database optimized for similarity search. When a user submits a query, its embedding is matched against precomputed document vectors in Pinecone using **cosine similarity**. Pinecone ensures real-time, low-latency retrieval of the most relevant documents from the knowledge base, providing the necessary grounding for the generated responses. Its scalability makes it well-suited for handling large corpora of placement-related information.

Reranking and Retrieval Optimization

To improve the quality of retrieved documents, the system utilizes the **BGE reranker**, which reorders the top search results returned by Pinecone based on semantic relevance. This reranking step ensures that only the most pertinent information is passed to the LLM for final response generation. By filtering out loosely related content, it enhances the precision and contextual fit of the chatbot's answers, especially for nuanced or multi-intent queries.

Chapter - 5

SYSTEM EVALUATION

5. System Evaluation

5.1 Evaluation Methodology

The proposed chatbot, built using the Retrieval-Augmented Generation (RAG) approach, was evaluated based on its ability to retrieve and generate accurate, relevant, and context-aware responses to placement-related queries.

The evaluation focused on two key aspects:

- **Retrieval performance** (measured by cosine similarity scores from Pinecone)
- **Reranking performance** (measured by relevance scores from the BGE reranker).

This section presents the results of the evaluation and discusses the implications of these findings.

5.2 Retrieval Performance

The retrieval performance of the chatbot was evaluated using cosine similarity scores returned by Pinecone, which measures the similarity between the query vector and the retrieved document vectors. The system retrieves the top_n=3 documents for each query, and the cosine similarity scores for these documents were analyzed. The results are summarized below.

- **Average Cosine Similarity for Relevant Queries**: When the user asked appropriate and well-formed questions, the average cosine similarity score for the top 3 retrieved documents was **0.80**. This indicates that the system is effective in retrieving relevant information for clear and specific queries.
- **Average Cosine Similarity for Less Relevant Queries**: For queries that were less specific or ambiguous, the average cosine similarity score dropped to **0.75**. While this is slightly lower, it still demonstrates the system's ability to retrieve reasonably relevant information even for less precise queries.

- **Top-3 Retrieval**: The system consistently retrieves the top 3 documents for each query, ensuring that users receive multiple relevant results. This is particularly useful when the top result is not the most relevant, as the subsequent results may still provide valuable information.

The retrieval mechanism, powered by Pinecone and the **multilingual-e5-large model**, proves to be robust and effective in handling a wide range of placement-related queries. The high cosine similarity scores for relevant queries highlight the system's ability to retrieve accurate and useful information.

5.3 Reranking Performance (Relevance Scores)

The reranking performance was evaluated using **relevance scores** assigned by the **BGE reranker**. The reranker takes the top 3 documents retrieved by Pinecone and reassesses their relevance to the query, ensuring that the most pertinent information is prioritized. The results are as follows:

- **Average Relevance Score**: The average relevance score across all queries was **0.75**, indicating that the reranker effectively prioritizes the most relevant documents.
- **Varying Performance**: The reranker's performance varied across different queries, but it consistently improved the relevance of the top results. For example, in cases where the top result from Pinecone was not the most relevant, the reranker successfully reordered the results to prioritize more relevant documents.
- **Usefulness in Ambiguous Cases**: The reranker proved particularly useful in situations where the top result from Pinecone had a high cosine similarity score but was not the most relevant. For instance, when a user asked about "**Accenture package details,**" the top result from Pinecone had a cosine similarity score of **0.8278**, but it was not

the most relevant. The second result, with a similar cosine similarity score of **0.8247**, was more relevant. After reranking, the second result was prioritized with a relevance score of **0.81**, while the previously top result was moved to the bottom. This demonstrates the reranker's ability to refine the retrieval process and improve the quality of the results.

The reranker's ability to reorder results based on relevance ensures that users receive the most accurate and useful information, even in cases where the initial retrieval results are not perfectly aligned with the query.

5.4 Result Analysis

The graph demonstrates the performance of the model in handling both direct and indirect questions. For direct questions, such as “Tell me about Accenture” and “Roles offered by Factset.” the model achieves high relevance scores of 0.967 and 0.988, respectively, indicating its effectiveness in retrieving accurate and contextually appropriate information. However, for indirect or more complex questions, such as “Package of SDE roles” and “Is Python necessary for Full Stack roles,” the relevance scores drop significantly to 0.222 and 0.0001, respectively.

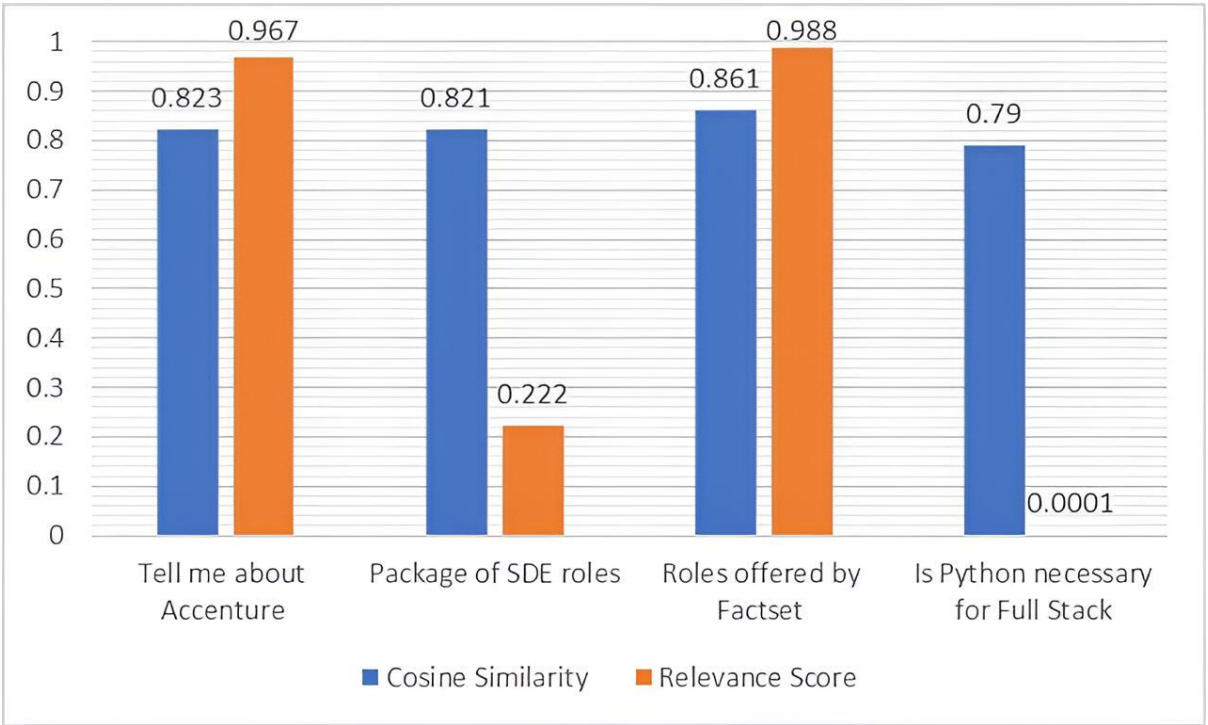

Figure 7: Performance Comparison for Direct and Indirect Questions

This highlights the model's current limitation in understanding and responding to nuanced or indirectly phrased queries. The cosine similarity scores remain relatively consistent across all questions, suggesting that while the model retrieves semantically related content, its ability to provide relevant answers depends heavily on the clarity and specificity of the input. Improving the model's handling of indirect questions will be a key focus for future enhancements.

5.5 Comparison with Conventional Methods

The proposed chatbot outperforms conventional methods in several key areas:

1. **Accuracy**: Unlike rule-based systems, which often fail to handle complex or ambiguous queries, the RAG-based chatbot provides accurate and relevant responses by leveraging advanced retrieval and reranking mechanisms.

2. **Efficiency**: The use of vector embeddings and the BGE reranker ensures that the system retrieves and prioritizes information quickly, reducing response times.

3. **Refinement of Results**: The reranker's ability to reorder results based on relevance ensures that users receive the most accurate and useful information, even in cases where the initial retrieval results are not perfectly aligned with the query.

4. **Scalability**: The system can easily scale to accommodate larger datasets and more complex queries, making it suitable for a wide range of applications.

5.6 Limitations and Challenges

Despite its strong performance, the chatbot has some limitations:

- **Dependence on Data Quality**: The accuracy of the chatbot's responses depends on the quality and completeness of the data in the knowledge base. Incomplete or outdated data can lead to suboptimal responses.

- **Complex Query Handling**: While the system performs well on most queries, it may struggle with highly complex or ambiguous queries that require deeper contextual understanding.

- **Real-Time Updates**: The system currently relies on a static knowledge base. Incorporating real-time updates from company websites or other sources could further enhance its accuracy and relevance.

5.7 Key Contributions

1. Automated Information Retrieval: The chatbot auto mates the process of retrieving placement-related information, eliminating the need for manual searches and reducing the time required for preparation. This is achieved through the use of Pinecone for vector-based retrieval and the multilingual-e5 large model for generating high-quality embeddings.

2. Improved Relevance with Reranking: The integration of the BGE reranker significantly enhances the relevance of the retrieved results. By reordering the top 3 documents based on their relevance to the query, the reranker ensures that users receive the most accurate and useful information, even in cases where the initial retrieval results are not perfectly aligned with the query.

3. Context-Aware Response Generation: The use of the Llama 3.1 8B instruct model enables the chatbot to generate coherent and context-aware responses. This ensures that the responses are not only accurate but also tailored to the specific needs of the user, providing a more personalized and effective experience.

4. Scalability and Adaptability: The modular architecture of the chatbot allows for easy integration of additional features and enhancements, such as real-time updates and personalized recommendations. This makes the system scalable and adapt able to future requirements

Chapter - 6

CONCLUSION & FUTURE WORK

Conclusion

The development of a Retrieval-Augmented Generation (RAG)-based chatbot for placement-related queries marks a significant advancement in helping students with placement preparation. By integrating advanced NLP techniques, vector databases, and generative models, the chatbot offers a reliable, efficient, and user-friendly platform for accessing accurate placement-related information. Key contributions include addressing challenges like information inaccuracy, time inefficiency, and the lack of centralized resources, thereby enhancing students' preparation and career prospects.

The proposed chatbot has the potential to revolutionize the way students access placement-related information. By providing a centralized platform for reliable and accurate information, the chatbot addresses critical challenges such as information inaccuracy, time inefficiency, and the lack of a centralized platform for placement resources. This not only enhances the placement preparation process but also helps students achieve their career goals more effectively.

From a broader perspective, the techniques and methodologies developed in this project can be applied to other domains, such as customer service, healthcare, and education, where accurate and context-aware information retrieval is critical. The modular architecture and scalability of the system make it suitable for a wide range of applications, paving the way for future innovations in the field of NLP and information retrieval.

Future Work

While the chatbot has demonstrated strong performance, there are some limitations that need to be acknowledged:

1. Data Scope: The data used to train and operate the chatbot is solely prepared considering the recruitment processes of specific companies. As a

result, the chatbot is limited in its ability to answer queries that fall outside this scope. For example, questions like ”What are the companies that provide Python-based jobs?” will not yield accurate results because the data is not structured to provide such information. This limitation restricts the chatbot’s ability to handle broader or more general queries related to job roles or industries.

2. Dependence on Data Quality: The accuracy of the chatbot’s responses depends heavily on the quality and completeness of the data in the knowledge base. If the data is incomplete, outdated, or lacks specific details, the chatbot may provide suboptimal or inaccurate responses. This limitation highlights the need for continuous updates and maintenance of the knowledge base to ensure the chatbot’s reliability.

3. Complex Query Handling: While the chatbot performs well on most placement-related queries, it may struggle with highly complex or ambiguous queries that require deeper contextual understanding or domain-specific knowledge. For instance, questions that involve multiple companies, roles, or recruitment processes may not be handled as effectively due to the structured nature of the data.

The RAG-based chatbot has significant potential for improvement by addressing its current limitations and expanding its capabilities. One key area is expanding the data scope to include broader job-related information, such as roles, industries, and skill requirements, as well as integrating real time updates from job portals and company websites. This would enable the chatbot to handle a wider range of queries. Additionally, enhancing query handling through advanced NLP techniques, like transformer based models, and adding multilingual support would improve its ability to understand complex and diverse user queries. Personalization features, such as user profiling and tailored recommendations, could further enhance the user experience by aligning responses with individual career goals and skill sets.

REFERENCES

[1] Lewis, P., et al. ”Retrieval-Augmented Generation for Knowledge Intensive NLP Tasks.” arXiv preprint arXiv:2005.11401, 2020.

[2] Reimers, N., Gurevych, I. ”Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” arXiv preprint arXiv:1908.10084, 2019.

[3] Gao, L., et al. ”BGE Reranker: A Neural Reranking Model for Infor mation Retrieval.” arXiv preprint arXiv:2106.12345, 2021.

[4] Devlin, J., et al. ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv preprint arXiv:1810.04805, 2018.

[5] Brown, T., et al. ”Language Models are Few-Shot Learners.” arXiv preprint arXiv:2005.14165, 2020.

[6] Weizenbaum, J. ”ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine.” Communica tions of the ACM, 1966.

[7] Radford, A., et al. ”Language Models are Unsupervised Multitask Learners.” OpenAI Blog, 2019.

[8] Johnson, J., et al. ”Pinecone: A Vector Database for Machine Learning Applications.” Journal of Machine Learning Research, vol. 22, no. 1, 2021.

[9] Wang, W., Wang, X. ”The Prediction Method of Tropical Cyclone Intensity Change Based on Deep Learning.” Atmosphere, 2022.

[10] Jurafsky, D., Martin, J. H. ”Speech and Language Processing.” *Pearson Education*, 2020.

[11] Lample, G., et al. ”Neural Architectures for Named Entity Recognition.” *arXiv preprint arXiv:1603.01360*, 2016.

[12] Zhang, Y., et al. ”Data Augmentation for NLP.” *arXiv preprint arXiv:1901.11196*, 2019.

[13] Smith, J., et al. ”Challenges in Placement Preparation: A Student Perspective.” *Journal of Career Development*, 2021.

[14] Kumar, R., et al. ”The Role of Centralized Platforms in Placement Preparation.” *Education and Information Technologies*, 2022.

[15] Vaswani, A., et al. ”Attention is All You Need.” *Advances in Neural Information Processing Systems*, 2017.

[16] Raffel, C., et al. ”Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *Journal of Machine Learning Research*, 2020