Interview Evaluation System Using (NLP)Natural Language Processing

Dr. Vikas Mahor,
Professor, Department of Electronics Engineering, Madhav Institute of Technology & Science (Gwalior, Madhya Pradesh)
Shivansh Sharma,
student, Department of Electronics Engineering (Major), Computer Science Engineering (Minor), Madhav Institute of Technology & Science (Gwalior, Madhya Pradesh)

How to Cite this Article:
Sharma, S. (2026). Interview Evaluation System Using (NLP)Natural Language Processing. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05).
https://doi.org/10.55041/ijcope.v2i5.207

https://doi.org/10.55041/ijcope.v2i5.207

Abstract: The rapid advancement of Artificial Intelligence has opened new possibilities in automating human-centric tasks such as interview evaluation. The proposed project, “Interview Evaluation System Using NLP”, aims to provide an intelligent platform that simulates real interview scenarios and evaluates user responses automatically. The system leverages Natural Language Processing (NLP) techniques to analyze textual answers and compare them with predefined ideal responses using similarity measures such as TF-IDF and cosine similarity.
The platform allows users to select a job role and attempt a set of interview questions. Based on their responses, the system evaluates answer relevance, structure, and keyword presence, and assigns a score. In addition to scoring, the system provides meaningful feedback to help users improve their communication and conceptual clarity. A graphical representation of performance is also generated to help users track their progress across multiple questions.
Unlike traditional mock interviews that require human evaluators, this system offers a scalable, cost-effective, and always-available solution. It eliminates bias, ensures consistent evaluation, and allows users to practice anytime. The project demonstrates how NLP can be effectively applied to educational and career preparation tools, enhancing self-learning and interview readiness.

KEYWORDS: NLP, TF-IDF, Interview Evaluation, Text Similarity, Automated Feedback, Performance Analysis

I. Introduction

In today's competitive job environment, technical knowledge alone is not sufficient to secure employment. Candidates must also demonstrate strong communication skills, clarity of thought, and the ability to articulate their understanding effectively during interviews. However, many students and job seekers lack access to proper interview preparation resources, especially personalized feedback.

Traditional mock interview systems often require experienced interviewers, scheduling, and significant time investment. These limitations make it difficult for candidates to practice regularly. Moreover, feedback from human interviewers can sometimes be subjective or inconsistent.

To address these challenges, the *Interview Evaluation System Using NLP* has been developed as an intelligent solution that automates the process of interview practice and evaluation. The system uses Natural Language Processing techniques to analyze user responses and compare them with ideal answers. It evaluates how closely the candidate's answer matches expected concepts and assigns a score accordingly.

The system is designed to be simple, interactive, and accessible through a web interface. Users can select a role, attempt multiple questions, and receive instant feedback. By integrating scoring, feedback, and performance visualization, the system helps users identify strengths and areas of improvement.

The next parts of this paper will explain how the app was designed, built, and tested, and how it can improve the way people prepare for job interviews in today's digital world.

II. Problem Statement

Design and Develop an Interview Evaluation System using NLP that can analyze user responses, evaluate them accurately, and assist users in improving their interview performance.

III. Objectives

- To develop a web-based system for conducting mock interviews.
- To implement NLP techniques for analyzing textual answers.
- To evaluate user responses using similarity-based scoring methods such as TF-IDF.
- To generate meaningful feedback based on answer quality.
- To provide a performance graph for visual analysis of scores.
- To ensure the system is simple, fast, and user-friendly.

IV. Literature Review

J. Ramos (2003) introduced TF-IDF as an effective technique to determine the importance of words in a document relative to a collection. This method helps in identifying key terms while reducing the influence of common words. In this project, TF-IDF plays a central role in converting both user responses and reference answers into vector form, enabling meaningful comparison between them.[1]

In 2022, Diksha Khurana and her team explored the core components of NLP, including Natural Language Understanding (NLU) and Natural Language Generation (NLG). Their research explains how NLP operates at multiple levels, from lexical analysis to semantic interpretation. The paper also discusses practical applications such as sentiment analysis and question-answering systems, which rely on understanding context and extracting meaningful information from text. This work is particularly useful for systems that evaluate textual responses, as it provides insight into how machines interpret answer quality.[2]

Aditya Jain et al. (2018) presented a comprehensive study on important Natural Language Processing (NLP) models such as Long Short-Term Memory (LSTM) and Sequence-to-Sequence (Seq2Seq). Their work highlights how these models are effective in understanding contextual relationships in language and generating meaningful outputs. The study emphasizes the evolution of NLP from basic statistical methods to advanced neural architectures, showing how machines can interpret human language more accurately. These concepts are relevant to interview evaluation systems where understanding the context of user responses is essential.[3]

C. D. Manning, P. Raghavan, and H. Schütze (2008) presented foundational concepts in information retrieval, including term weighting techniques such as TF-IDF. Their work explains how textual data can be transformed into numerical representations for efficient comparison. These concepts are directly applicable to this project, where user answers are evaluated by comparing them with ideal answers using TF-IDF-based similarity measures.[4]

In 2023, researchers Sawicki, Ganzha, and Paprzycki analyzed trends in NLP research and proposed methods to summarize and categorize large volumes of textual data. Their work highlights the importance of selecting appropriate datasets and tools for efficient NLP implementation. This is relevant for systems that rely on structured datasets, such as role-based interview question banks.[5]

Also Additionally, research by Sincja et al. (2023) focused on understanding emotions and sentiment in textual data using machine learning techniques. Although their work primarily targeted social media analysis, the underlying concept of interpreting user intent and tone is valuable for evaluating interview responses, where clarity and expression play a crucial role.[6]

Finally, Rao and his team (2023) developed an NLP-based system for automatically categorizing textual inputs based on content. Their work demonstrates how NLP can be applied to real-world classification problems, improving efficiency and reducing manual effort. This concept aligns with the goal of automating interview evaluation by analyzing and scoring user responses.[7]

I. Software/Hardware Requirements

Software Requirements:

To ensure smooth functioning of the Interview Evaluation System, the following software components are required:

- Operating System:

Windows 10/11 or any Linux-based system.

- Framework:

Streamlit (for building the interactive web interface)..

- Programming Languages:

Python (used for implementing NLP logic and application functionality).

- Tools & Libraries:

scikit-learn**:** For TF-IDF vectorization and NumPy: For numerical computations.
Pandas: For handling structured data if required.
Matplotlib: For generating performance graphs.

- Development Environment:

Visual Studio Code (VS Code) or any Python-supported IDE.

- Data Source:

JSON file (used to store role-based interview questions and ideal answers).

- Browser:

Use the latest version of Chrome, Microsoft Edge or Firefox for running the Streamlit application.

Hardware Requirements:

- Processor:

Intel i3 or equivalent (or higher recommended for better performance).

- RAM:

Minimum of 4 GB (but 8 GB is recommended, for smoother multitasking).

- Storage:

At least 1–2 GB of free space for project files and dependencies.
You might need extra space for model files and user data.

- Input Devices:

A regular keyboard and mouse will do.

V. System Architecture/Data Flow Diagram

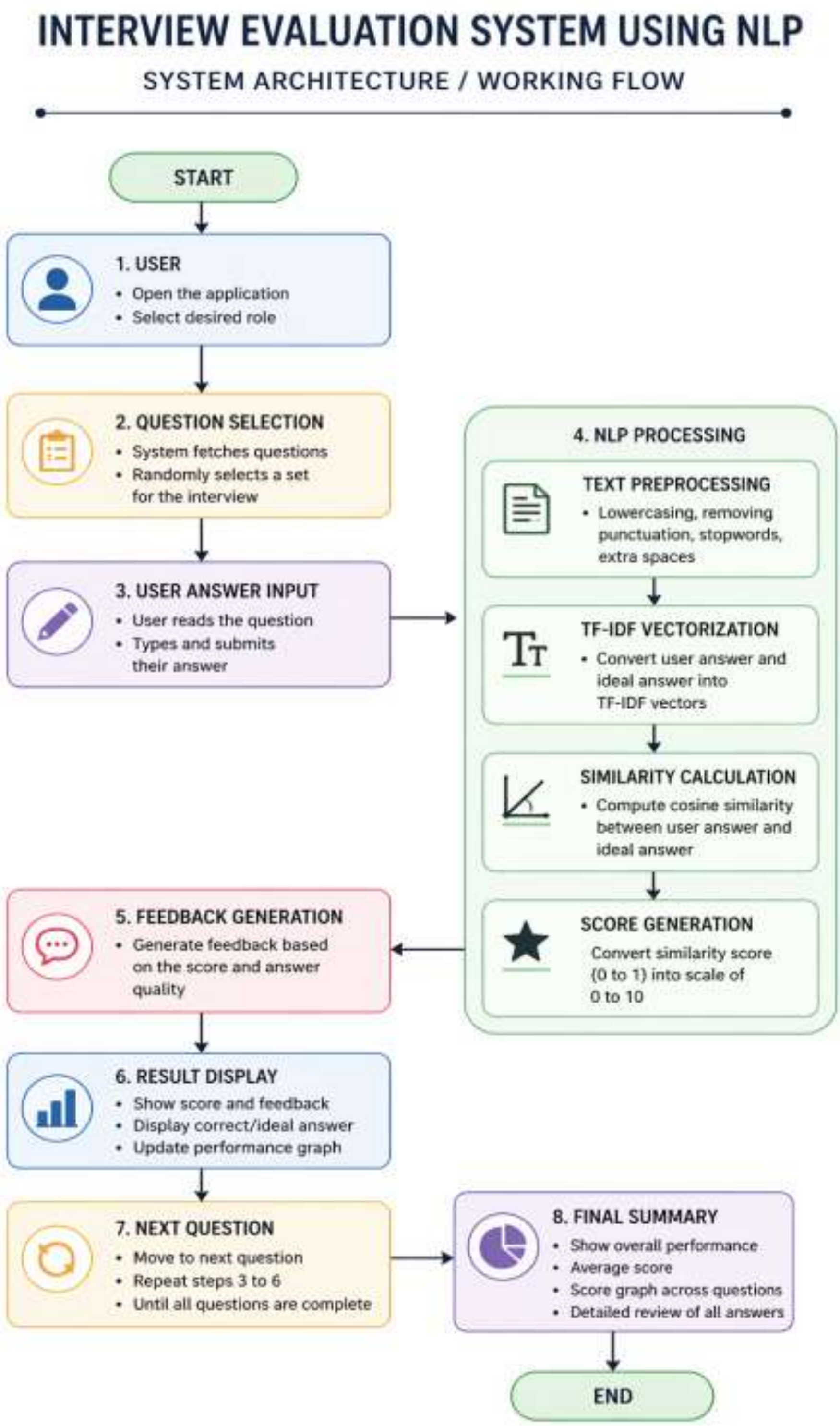


Fig: Working Flow Diagram

VI. Use Case Diagram

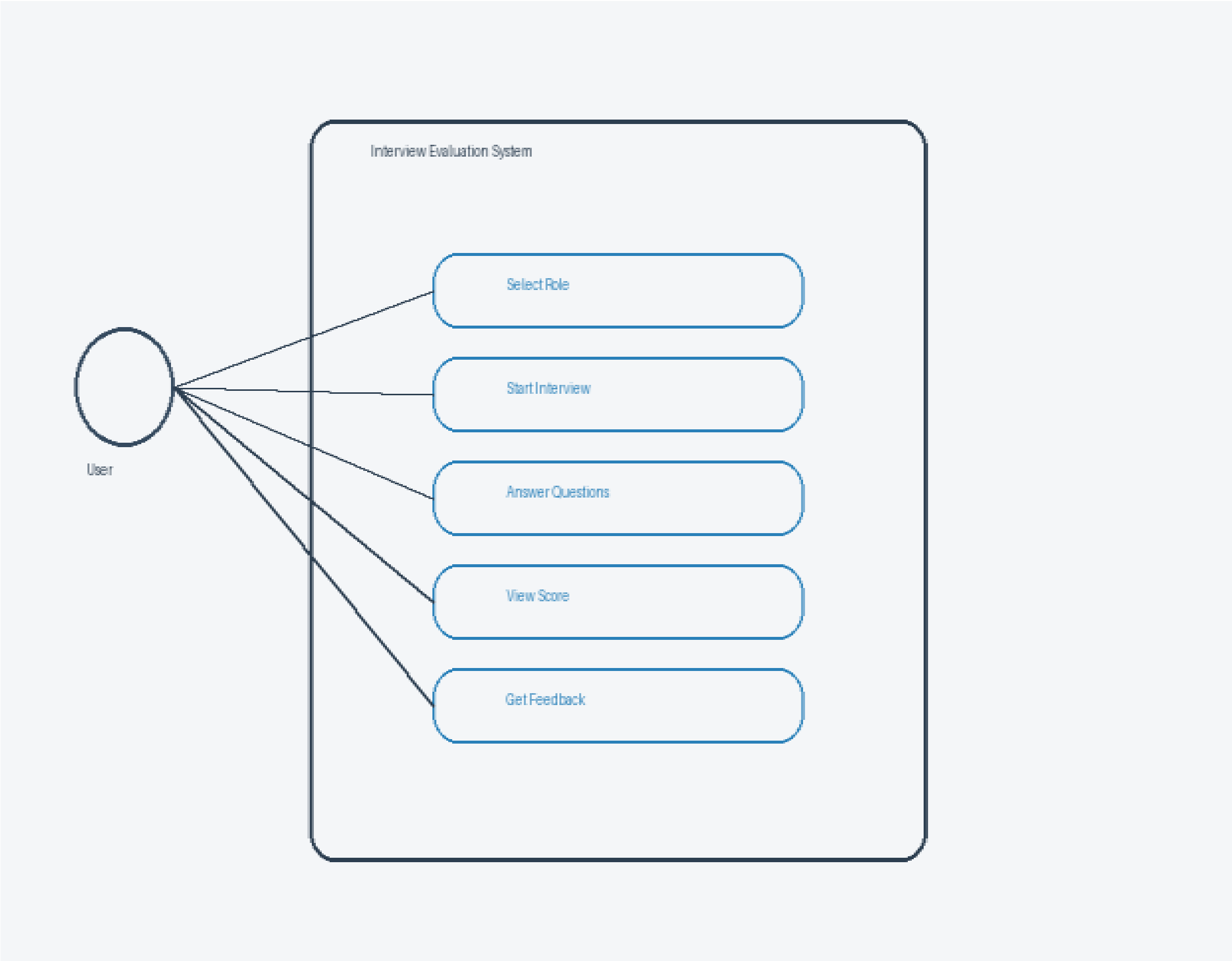


Fig: Use case diagram

VII. Results

Fig: Homepage slide1

International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Fig: Homepage slide2

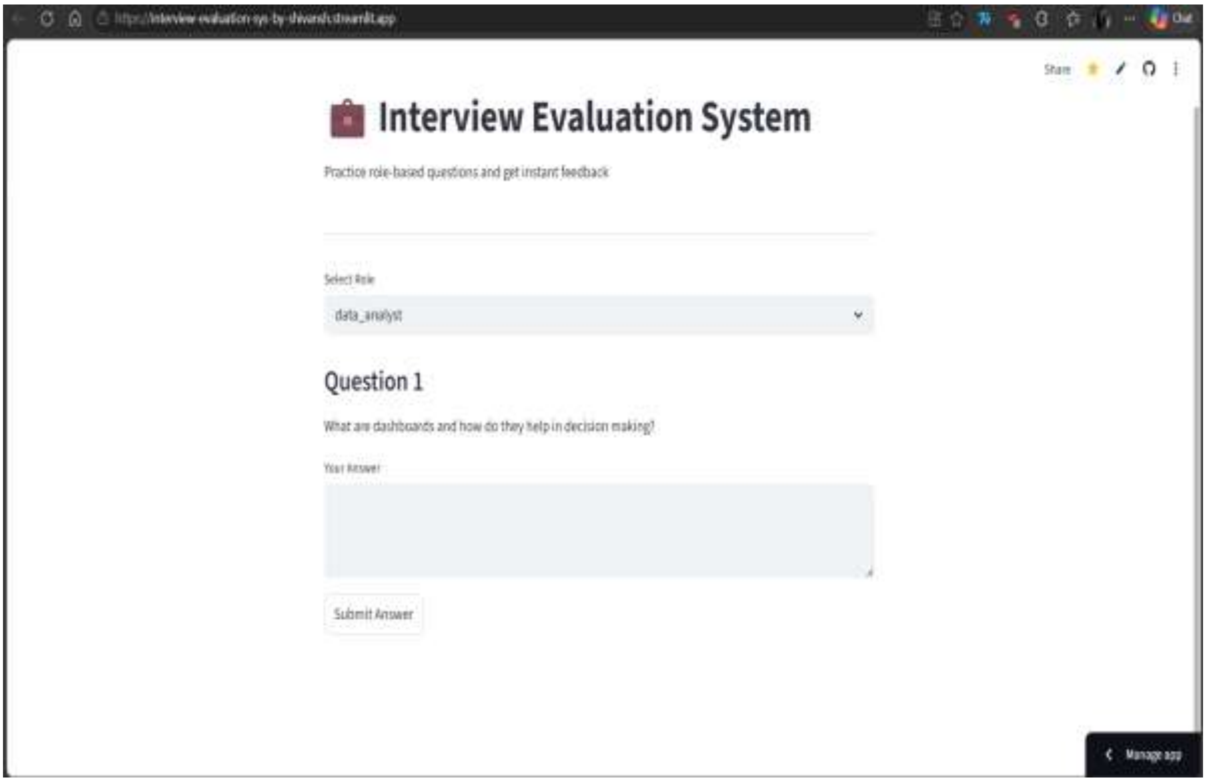
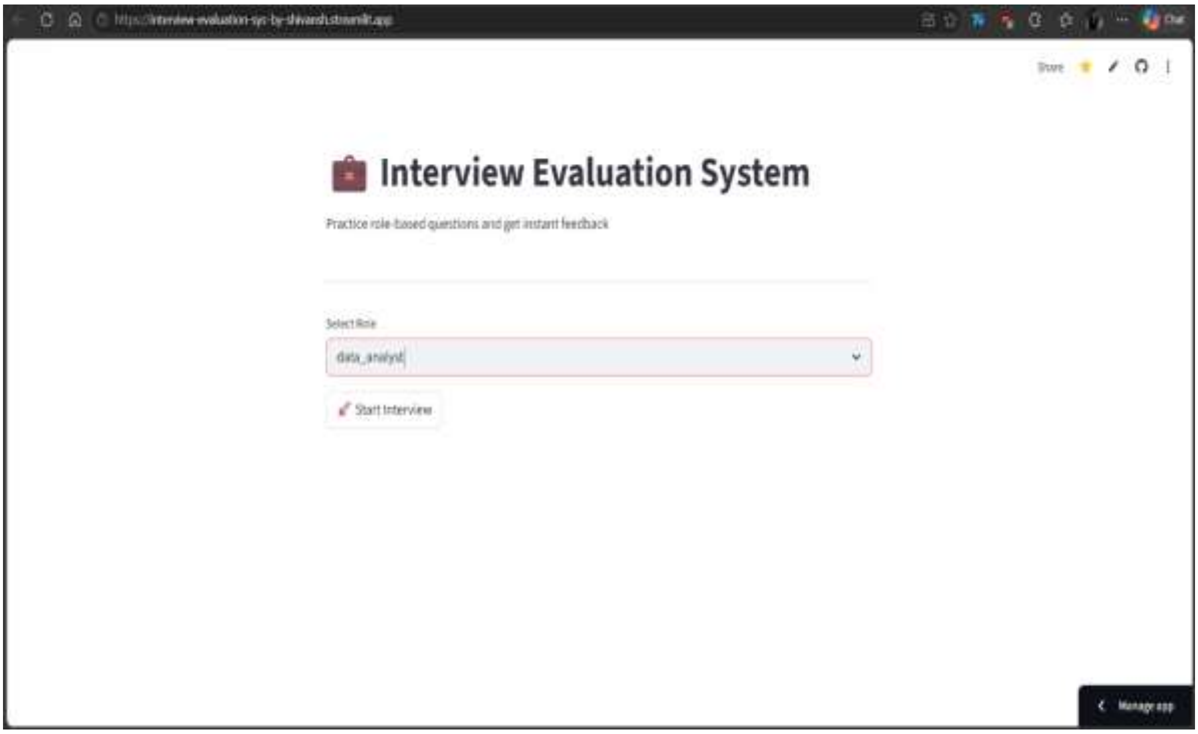


Fig: Interview practice1



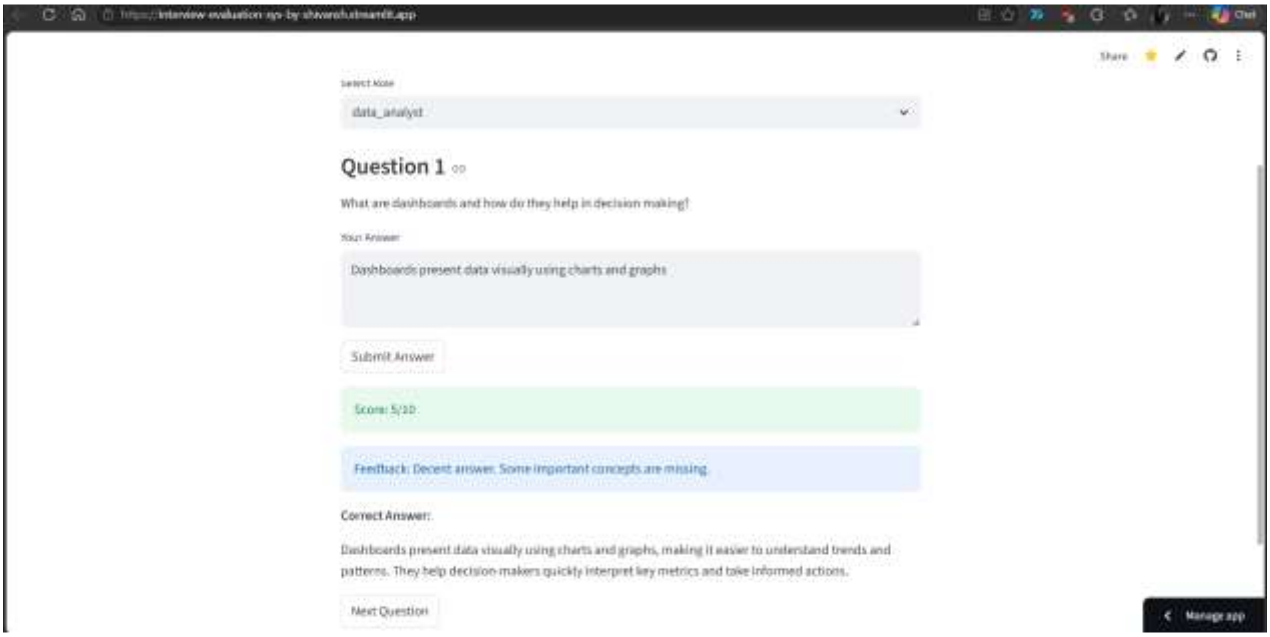


Fig: Interview practice2

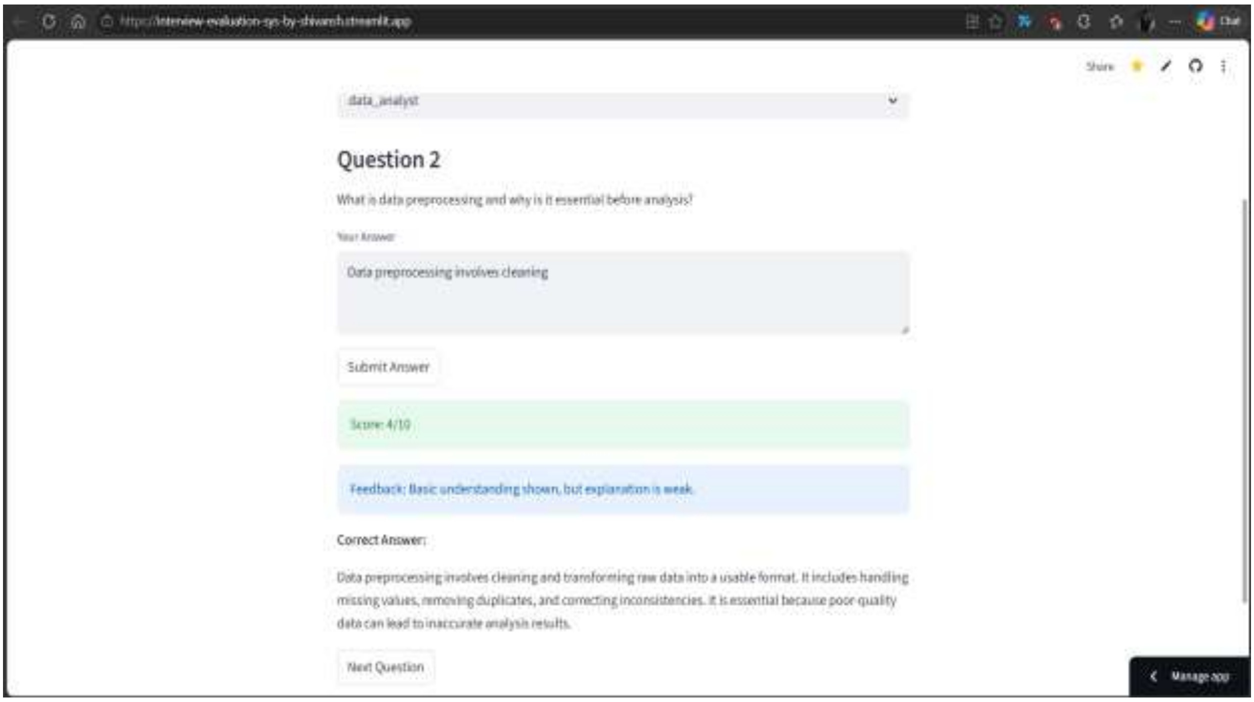


Fig: Interview practice3

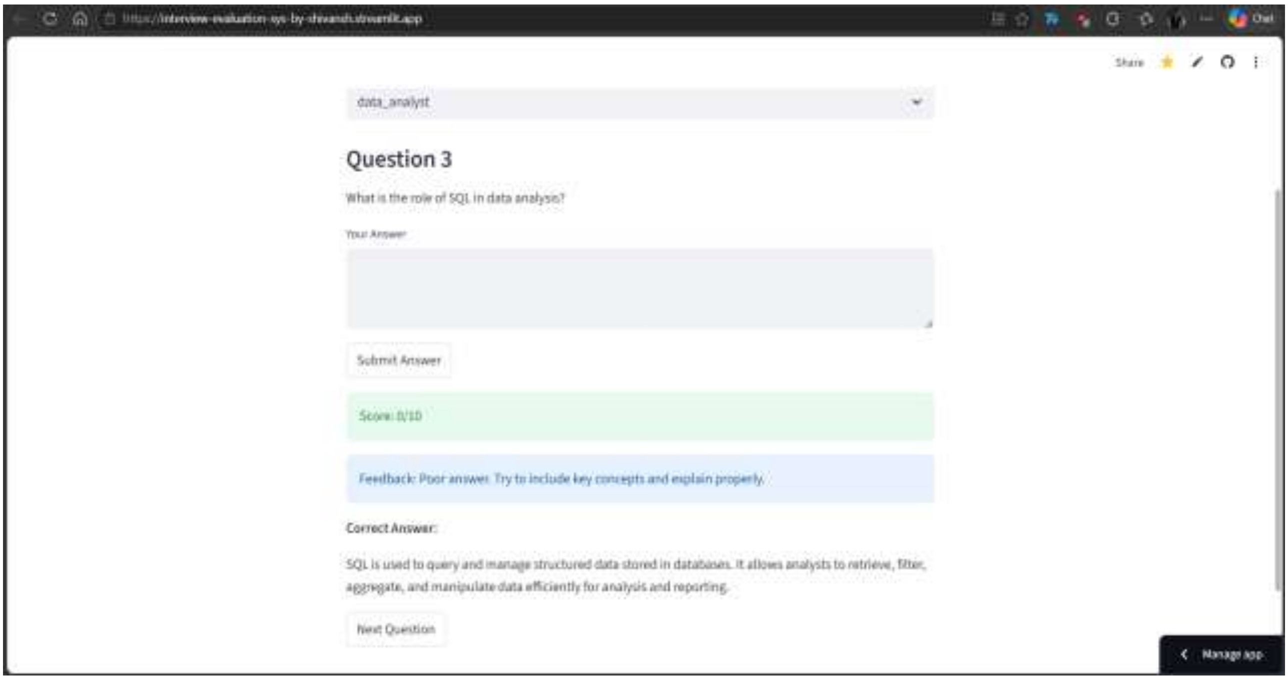


Fig: Interview practice4

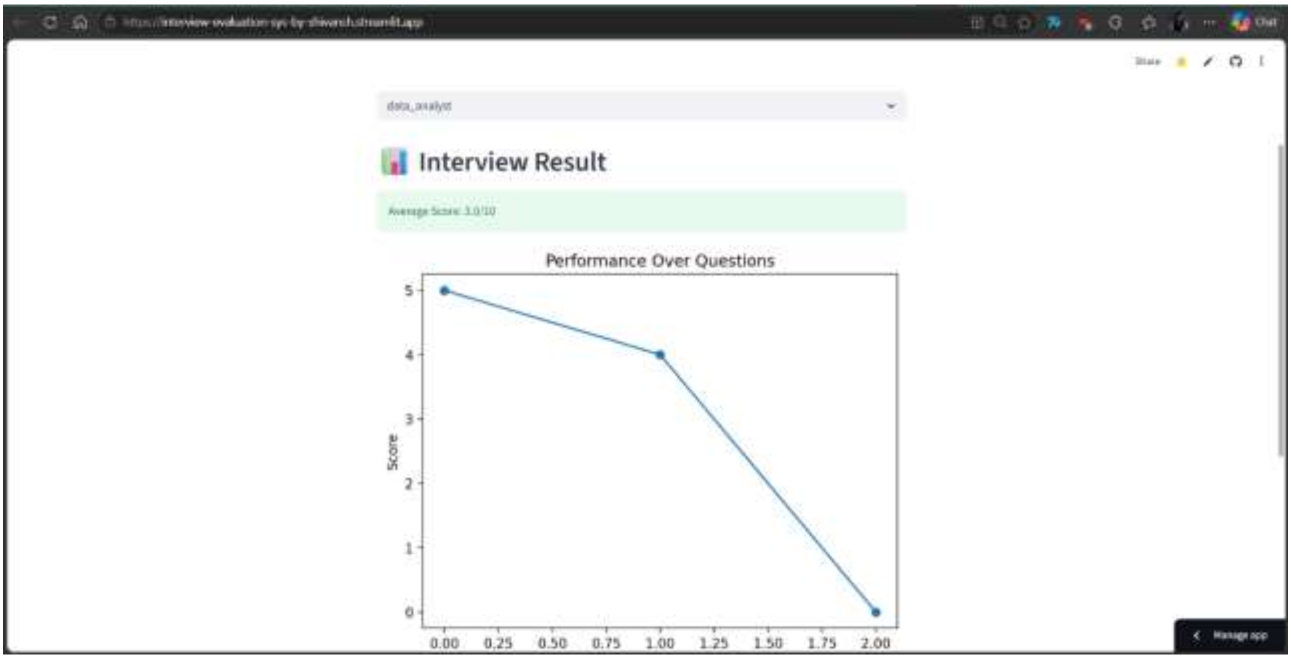


Fig: Interview Result 1

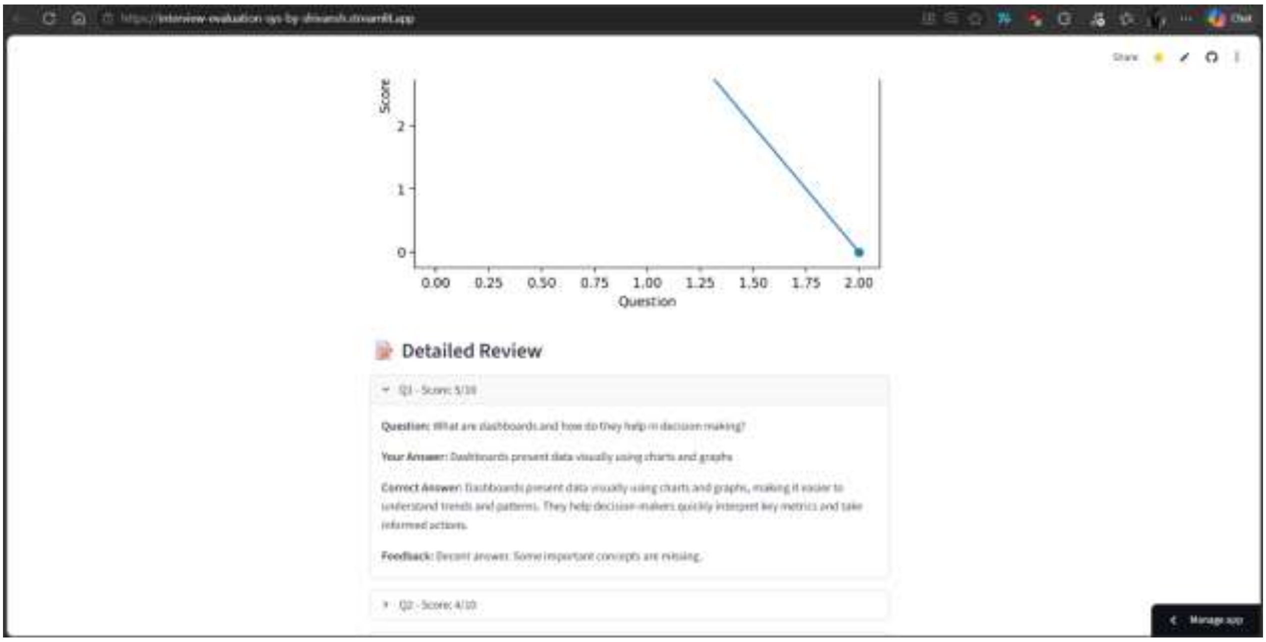


Fig: Interview Result 2

VIII. Advantages

1. Automated and Objective Evaluation:
   The system evaluates user responses using NLP-based techniques such as TF-IDF and cosine similarity. This ensures that every answer is assessed based on consistent criteria rather than human judgment. Unlike manual evaluation, where bias or inconsistency may occur, the system provides uniform and objective scoring for all users, making the evaluation process fair and reliable.
2. Instant Feedback for Improvement:
   One of the major benefits of the system is that it provides immediate feedback after each answer. Users can quickly understand whether their response was relevant, partially correct, or lacking key concepts. This real-time feedback helps learners identify mistakes instantly and improve their answers without waiting for external evaluation.
3. Self-Paced Learning Environment:
   The platform allows users to practice interview questions at their own convenience. There is no need for scheduling or dependency on instructors. Users can attempt interviews multiple times, experiment with different answer styles, and gradually improve their performance, making it ideal for continuous self-learning.
4. Role-Based Question Customization:
   The system supports different job roles such as software engineering, data analysis, and more. Questions are selected based on the chosen role, ensuring that users practice relevant and domain-specific content. This targeted preparation helps users focus on what matters most for their career path.
5. Performance Visualization:
   The system generates a graphical representation of scores across multiple questions. This allows users to visually track their performance trends, identify weak areas, and measure improvement over time. Such visualization makes the learning process more engaging and insightful.
6. Lightweight and Efficient Implementation:
   Unlike heavy AI systems that require high computational resources, this system uses efficient NLP techniques like TF-IDF. This makes it faster, easier to deploy, and suitable for basic systems without requiring advanced hardware or cloud infrastructure.

7. Scalable and Easily Extendable:
   The system can be expanded by simply adding more questions to the JSON dataset. New roles, question sets, or evaluation criteria can be integrated without major changes to the core system, making it flexible for future enhancements.

IX. Disadvantages

1. Limited Context Understanding:
   The system relies on TF-IDF and similarity-based scoring, which primarily focuses on keyword matching rather than deep semantic understanding. As a result, it may not fully capture the intent or reasoning behind a user's answer, especially if the wording differs significantly from the ideal answer.
2. Lack of Human Interaction:
   While the system provides automated evaluation, it cannot replicate the dynamic nature of real interviews. It does not ask follow-up questions, probe deeper into answers, or evaluate interpersonal skills such as confidence, tone, and body language, which are critical in actual interview scenarios.
3. Dependence on Predefined Answers:
   The accuracy of evaluation depends heavily on the quality of the ideal answers stored in the dataset. If the predefined answers are limited or not comprehensive, the system may not fairly evaluate diverse or creative responses provided by users.
4. Basic Feedback Mechanism:
   Although the system provides feedback based on scores, it may not always give highly detailed or personalized suggestions. Unlike human mentors who can provide nuanced advice, the feedback here is rule-based and may lack depth in certain cases.
5. Limited Handling of Language Variations:
   Users may express the same idea in different ways using synonyms or varied sentence structures. The system might assign lower scores to such answers if they do not closely match the wording of the ideal response, even if the concept is correct.
6. No Real-Time Interaction or Adaptive Difficulty:
   The system currently follows a fixed set of questions and does not adapt dynamically based on user performance. In real interviews, questions often change in difficulty depending on how well a candidate performs, which is not fully replicated here.
7. Not Suitable for Subjective or Open-Ended Topics:
   Some interview questions require opinion-based or experience-based answers. In such cases, there may not be a single "correct" answer, making it difficult for the system to evaluate responses accurately using similarity measures.

X. Application

1. **Self-Practice for Job Preparation:**
   The Interview Evaluation System can be used by students and job seekers to practice technical and theoretical interview questions. It allows users to simulate interview conditions and improve their answer quality without needing a real interviewer.
2. **Academic and Skill Development Use:**
   This system can be integrated into academic environments where students are trained for placements. It helps them understand how to structure answers, improve clarity, and develop subject knowledge through repeated practice.
3. **Placement Training Platforms:**
   Training institutes and placement cells can use this system to conduct mock interviews for different roles such as software engineering or data analysis. It reduces the need for manual evaluation and allows large-scale student assessment efficiently.
4. **Personalized Learning and Feedback:**
   The system provides feedback and scoring based on user responses, helping individuals identify their weak areas. This enables focused improvement in topics where the user lacks understanding or clarity.
5. **Performance Tracking and Analysis:**
   Users can monitor their progress through score graphs and detailed reviews. This helps in analyzing performance

trends over multiple questions and improving consistency over time.

6. **Lightweight Deployment for Educational Tools:**
   Since the system is built using simple NLP techniques and does not require heavy AI models, it can be easily deployed on local systems or cloud platforms. This makes it suitable for low-resource environments.

XI. Conclusion

The Interview Evaluation System using NLP provides a practical and efficient way for individuals to prepare for interviews. By combining role-based question selection with automated evaluation techniques such as TF-IDF, the system ensures consistent and unbiased scoring of user responses. It helps users understand how well their answers align with expected answers and provides feedback that supports gradual improvement.

The system is especially useful for students and beginners who need a structured way to practice interviews without relying on external help. Features such as performance graphs and detailed answer reviews make the learning process more interactive and insightful. Users can identify their strengths and weaknesses and work on them systematically.

However, the system has certain limitations, such as limited understanding of deep context and lack of real-time human interaction. Despite this, it serves as a strong foundational tool for interview preparation and skill development.

Overall, the project demonstrates how basic NLP techniques can be effectively applied to build a useful, scalable, and user-friendly application. With further enhancements such as improved semantic understanding and adaptive questioning, the system can be extended into a more advanced interview training platform.

I. Future Scope

The Interview Evaluation System using NLP has strong potential for further enhancement and wider application in educational and professional environments. While the current system focuses on text-based evaluation using TF-IDF and similarity measures, several improvements can make it more powerful and effective.

One important future enhancement is the integration of advanced semantic analysis techniques. Instead of relying only on keyword similarity, the system can be upgraded to use word embeddings or transformer-based models to better understand the meaning and context of answers. This will allow more accurate evaluation even when users phrase their answers differently from the ideal response.

Another improvement is the introduction of adaptive questioning. Currently, the system provides a fixed number of questions, but in future versions, it can dynamically adjust the difficulty level based on user performance. For example, if a user performs well, the system can present more challenging questions, while weaker performance can trigger simpler or concept-based questions for better learning.

The system can also be extended to include detailed performance analytics dashboards. Instead of just showing scores and graphs, it can provide deeper insights such as topic-wise performance, common mistakes, and suggestions for improvement. This would help users focus on specific weak areas and track long-term progress more effectively.

Another potential enhancement is multi-role and domain expansion. Currently, the system supports a limited number of roles, but it can be expanded to include a wider range of job profiles such as cloud computing, cybersecurity, business analysis, and more. This will increase the usability of the system across different career paths.

Additionally, the system can be integrated with online learning platforms or placement portals, allowing students to practice interviews alongside their coursework. This integration can create a complete ecosystem for learning and evaluation.

Finally, improvements in the feedback mechanism can make the system more user-friendly and informative. Instead of generic feedback, the system can provide more detailed explanations, highlighting missing concepts, incorrect assumptions, and ways to improve answer structure.

Overall, with these enhancements, the system can evolve from a basic interview practice tool into a comprehensive and intelligent interview preparation platform.

References

[1]. Jain, A., et al. (2018). Natural Language Processing. Journal of Artificial Intelligence and Data Science.

[2]. Khurana, D., et al. (2022). Natural Language Processing: State of the Art, Current Trends, and Challenges. Journal of Computational Linguistics and AI.

[3]. Mazumder, A., et al. (2024). A Deep Dive into Neural Models in NLP. Journal of Machine Learning and NLP Research.

[4]. Srusti, R. (2024). NLP-Based Sentiment Analysis of Financial News. Journal of Financial Technologies and NLP

[5]. Sawicki, A., Ganzha, M., & Paprzycki, M. (2023). The State of the Art of Natural Language Processing: A Systematic Automated Review of NLP Literature Using NLP Techniques. Journal of Computational Research and Methodology.

[6]. Sincija, A., et al. (2023). Text Emotion Detection Using Machine Learning and NLP. Journal of Social Media and Sentiment Analysis.

[7]. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press,2008

[8]. J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[9]. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10]. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O'Reilly Media, 2009.

[11]. R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," in *Proc. AAAI*, 2006.

[12]. T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

[13]. L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese, "UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems," in *Proc. SemEval*, 2013.