International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

AI-Powered Resume Analysis, PDF Summarization, and Interview Preparation System Using NLP and Deep Learning

SALADHI SAHITHI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

K. Venkatesh

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

In today's competitive job market, candidates require efficient tools to enhance their resumes, prepare for interviews, and quickly understand large volumes of information. This project presents an AI-powered web-based system developed using Django that integrates Natural Language Processing (NLP) and deep learning techniques to assist users in resume evaluation, document summarization, and technical interview preparation.The system provides three primary functionalities. First, it evaluates resumes by extracting text from PDF and Word documents and analyzing the quality using tokenization and spell-checking techniques. The scoring mechanism estimates correctness based on lexical accuracy, helping users improve the quality of their resumes.Second, the system performs automatic document summarization using transformer-based models. By leveraging pre-trained models such as T5 via Hugging Face's transformer pipeline, the application extracts key information from uploaded PDF documents. This helps users quickly grasp the essential content without reading the entire document. Additionally, the system generates interactive questions from summarized content, enabling active learning.Third, the application includes an intelligent interview preparation module. It conducts a quiz-based technical assessment in programming languages such as Java, C, and C++. Based on user responses, it evaluates performance and provides targeted skill improvement suggestions. The system also enhances user interaction by converting questions into speech using text-to-speech technology, improving accessibility.The backend is implemented using Django, with MySQL for data storage and libraries such as PyPDF2, NLTK, and transformers for text processing. Features like speech synthesis using gTTS and spell correction using autocorrect further enhance usability.This system aims to bridge the gap between traditional resume building and intelligent career preparation tools by integrating AI-driven insights into a unified platform. It offers a scalable solution for students, job seekers, and professionals seeking automated feedback and efficient learning tools.

Keywords: Natural Language Processing, Text Summarization, Resume Analysis, Transformer Models, Deep Learning, Interview Preparation, Artificial Intelligence, Django Web Application, Text-to-Speech, Document Processing

I. INTRODUCTION

With the rapid growth of digital information and increasing competition in the job market, individuals require intelligent tools to enhance productivity and improve their career prospects. Traditional methods of resume evaluation, interview preparation, and document reading are time-consuming and often lack personalized feedback. This has led to the emergence of AI-driven systems that automate and enhance these processes.This project introduces an integrated web-based application that combines resume analysis, document summarization, and interview preparation into a single platform. The system leverages Natural Language Processing (NLP) and deep learning techniques to deliver intelligent insights and recommendations to users.Resume evaluation is one of the critical aspects of job preparation. Many candidates struggle with formatting, grammar, and content quality. The proposed system addresses this by extracting text from resumes in PDF or Word formats and analyzing it using tokenization and spell-checking methods. It generates a score indicating the overall quality of the resume, helping users identify areas for improvement.Another major challenge is handling large documents efficiently. Reading lengthy PDFs can be time-consuming and ineffective. The system integrates transformer-based summarization models to extract concise summaries from documents. This allows users to quickly understand key concepts and information without reading the entire text. Additionally, the system creates interactive questions from the summarized content, promoting active engagement and better retention.Interview preparation is also a crucial component of career readiness. The system includes a quiz module that tests users' knowledge in programming languages such as Java, C, and C++. Based on their responses, it evaluates performance and suggests areas for improvement. This personalized feedback helps users focus on specific skills that need enhancement.The application is developed using Django, ensuring scalability, security, and efficient handling of user requests. It integrates various libraries such as PyPDF2 for document parsing, NLTK for text processing, and Hugging Face transformers for summarization. The inclusion of text-to-speech functionality further improves accessibility and user experience.Overall, this system provides a comprehensive solution that combines multiple AI-driven functionalities to support users in their career development journey. It reduces manual effort, improves efficiency, and delivers intelligent insights, making it a valuable tool for modern job seekers.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Recent advancements in Artificial Intelligence and Natural Language Processing have significantly influenced the development of automated systems for text analysis, summarization, and career assistance tools. Several studies have explored the use of machine learning techniques for resume screening, document summarization, and

interview preparation systems.Early resume screening systems relied on keyword matching and rule-based approaches. These systems were limited in their ability to understand context and semantics. With the introduction of NLP techniques such as tokenization, part-of-speech tagging, and named entity recognition, more advanced systems were developed to analyze resumes more effectively. However, these approaches still lacked deep contextual understanding.

The emergence of deep learning models, particularly transformer-based architectures, revolutionized text processing tasks. Models like T5 (Text-to-Text Transfer Transformer) and BERT have demonstrated exceptional performance in tasks such as summarization, question answering, and text classification. These models can capture contextual relationships in text, enabling more accurate and meaningful analysis.In the domain of document summarization, both extractive and abstractive techniques have been explored. Extractive methods select important sentences directly from the text, while abstractive methods generate new sentences that capture the essence of the content. Transformer-based models have shown superior performance in abstractive summarization, producing coherent and human-like summaries.Interview preparation systems have also evolved over time. Traditional systems focused on static question banks without personalization. Recent approaches incorporate adaptive learning techniques that adjust questions based on user performance. Some systems also use speech recognition and natural language understanding to simulate real interview scenarios.Spell-checking and grammar correction tools have been widely used in text processing applications. Libraries such as autocorrect and advanced grammar-checking models help improve text quality by identifying and correcting errors. These tools play a crucial role in resume evaluation systems.

Text-to-speech (TTS) technology has also gained importance in enhancing accessibility. Tools like gTTS allow applications to convert text into speech, making systems more interactive and user-friendly.Despite these advancements, many existing systems focus on a single functionality, such as resume screening or document summarization. There is a lack of integrated platforms that combine multiple features into a unified system. This project addresses this gap by integrating resume analysis, summarization, and interview preparation into a single application.

III. EXISTING SYSTEM

Existing systems for resume evaluation, document summarization, and interview preparation are typically developed as separate tools, each addressing a specific problem. Resume screening systems often rely on keyword-based filtering or Applicant Tracking Systems (ATS), which match resumes against predefined job descriptions. While these systems are useful for recruiters, they provide limited feedback to candidates and do not help in improving resume quality.Document summarization tools are widely available, but many of them use basic extractive techniques that simply select important sentences from the text. These methods often fail to capture the overall meaning and may produce disjointed summaries. Advanced summarization tools using deep learning models exist, but they are often not integrated into user-friendly web applications.Interview preparation

platforms typically provide static question banks or multiple-choice quizzes. These systems lack personalization and do not offer detailed feedback or suggestions for improvement. Additionally, they do not adapt to the user's performance or identify specific skill gaps.Most existing solutions also lack multimodal interaction features such as text-to-speech, which can enhance user engagement and accessibility. Furthermore, users often need to switch between multiple platforms to access different functionalities, leading to inefficiency and inconvenience.

In summary, the existing systems are fragmented, lack integration, and provide limited intelligent feedback. This creates a need for a unified platform that combines resume analysis, document summarization, and interview preparation with AI-driven insights, which is addressed by the proposed system.

IV. PROPOSED METHOD

The proposed system is an AI-powered web application that integrates resume analysis, document summarization, and interview preparation into a unified platform. The system is designed using Django and incorporates advanced Natural Language Processing (NLP) and deep learning techniques to deliver intelligent insights.The system consists of three main modules: Resume Evaluation, PDF Summarization, and Interview Assessment. In the resume evaluation module, the system accepts PDF or Word documents, extracts textual content, and evaluates its quality using tokenization and spell-checking algorithms. The score generated reflects the correctness and readability of the resume.The PDF summarization module utilizes transformer-based models, specifically the Text-to-Text Transfer Transformer (T5), to generate concise summaries from lengthy documents. Transformer models have shown superior performance in abstractive summarization tasks by capturing contextual relationships in text. The system processes multiple pages and generates meaningful summaries along with question-based interactions to enhance user engagement.The interview preparation module conducts a structured quiz based on programming languages such as Java, C, and C++. It evaluates user responses and provides targeted suggestions for skill improvement. Additionally, the system incorporates text-to-speech functionality using gTTS to convert questions into audio, improving accessibility.The integration of these modules into a single platform provides a comprehensive solution for career development. The system is scalable, user-friendly, and capable of delivering personalized feedback, making it highly beneficial for students and job seekers.

V. IMPLEMENTATION

The implementation of the proposed system is carried out using the Django web framework, which follows the Model-View-Template (MVT) architecture. The backend is developed in Python, integrating various libraries for text processing, machine learning, and file handling.The system begins with user authentication, where users can register and log in securely. User credentials are stored in a MySQL database using the PyMySQL connector. After successful login, users can access different modules of the application.For document processing, the system supports both PDF and Word formats.

The PyPDF2 library is used to extract text from PDF files, while the python-docx library is used for Word documents. The extracted text is preprocessed using the NLTK library, which performs tokenization and sentence segmentation.In the resume evaluation module, the extracted text is analyzed using the autocorrect library to detect spelling errors. Each word is compared with its corrected form, and a score is calculated based on the number of correctly spelled words. This provides a quantitative measure of resume quality.

The PDF summarization module uses the Hugging Face transformers pipeline with a pre-trained T5 model. The model processes text in chunks and generates summaries using sequence-to-sequence learning. Transformer models such as T5 are highly effective in generating coherent summaries due to their attention mechanisms and contextual understanding. The summarized text is further processed into sentences, and interactive questions are generated for user engagement.The interview module is implemented using predefined question sets stored in lists. User responses are evaluated against correct answers, and performance is calculated as a score. Based on incorrect answers, the system suggests areas such as Java, C, or C++ for improvement.Additionally, the system incorporates text-to-speech functionality using the gTTS library. Questions are converted into audio files and stored dynamically, allowing users to listen to questions instead of reading them.File management is handled using Django's FileSystemStorage, ensuring efficient storage and retrieval of uploaded files. The frontend is designed using HTML templates, providing a user-friendly interface.Overall, the system integrates multiple technologies, including NLP, deep learning, and web development frameworks, to deliver a robust and efficient solution.

VI. ALGORITHMS

Resume Scoring Algorithm

1. Input resume document (PDF/DOCX)
2. Extract text using PyPDF2/docx
3. Convert text to lowercase
4. Tokenize text using NLTK
5. For each word:
    1. Apply spell checker
    2. Compare corrected word with original
6. Count correct words
7. Compute score = correct_words / total_words
8. Display resume quality score

2. PDF Summarization Algorithm

1. Input PDF document
2. Extract text page-wise
3. For each page:

1. Pass text to T5 transformer model
2. Generate summary

4. Concatenate summaries
5. Perform sentence tokenization
6. Output summarized text
7. Generate questions from summary

Transformer-based summarization models use sequence-to-sequence learning with attention mechanisms to produce high-quality summaries.

3. Interview Evaluation Algorithm

1. Initialize question set and answers
2. Display questions sequentially
3. Accept user responses
4. Compare responses with correct answers
5. Calculate score
6. Identify incorrect answers
7. Map errors to skill categories (Java, C, C++)
8. Generate improvement suggestions

VII. SYSTEM DESIGN

The system follows a modular architecture based on the Django MVT (Model-View-Template) framework. It is divided into three major layers: Presentation Layer, Application Layer, and Data Layer.

1. Presentation Layer

This layer consists of HTML templates that interact with users. It includes pages such as Login, Registration, Resume Upload, PDF Upload, and Interview Interface. The user interface is designed to be simple and intuitive, allowing easy navigation between modules.

2. Application Layer

The application layer contains the core logic of the system implemented in Django views. It handles request processing, file uploads, data validation, and interaction with machine learning models.

- **Resume Module:** Processes uploaded resumes and computes scores
- **Summarization Module:** Uses transformer models to generate summaries
- **Interview Module:** Conducts quizzes and evaluates responses

The system uses a pipeline-based architecture for summarization, where input text is processed through a transformer model to generate output summaries.

3. Data Layer

The data layer uses MySQL to store user information such as login credentials and personal details. File storage is managed using Django's FileSystemStorage.

4. Architecture Flow

1. User uploads document
2. System extracts text
3. NLP processing is applied
4. Model generates output (score/summary)
5. Results are displayed to the user

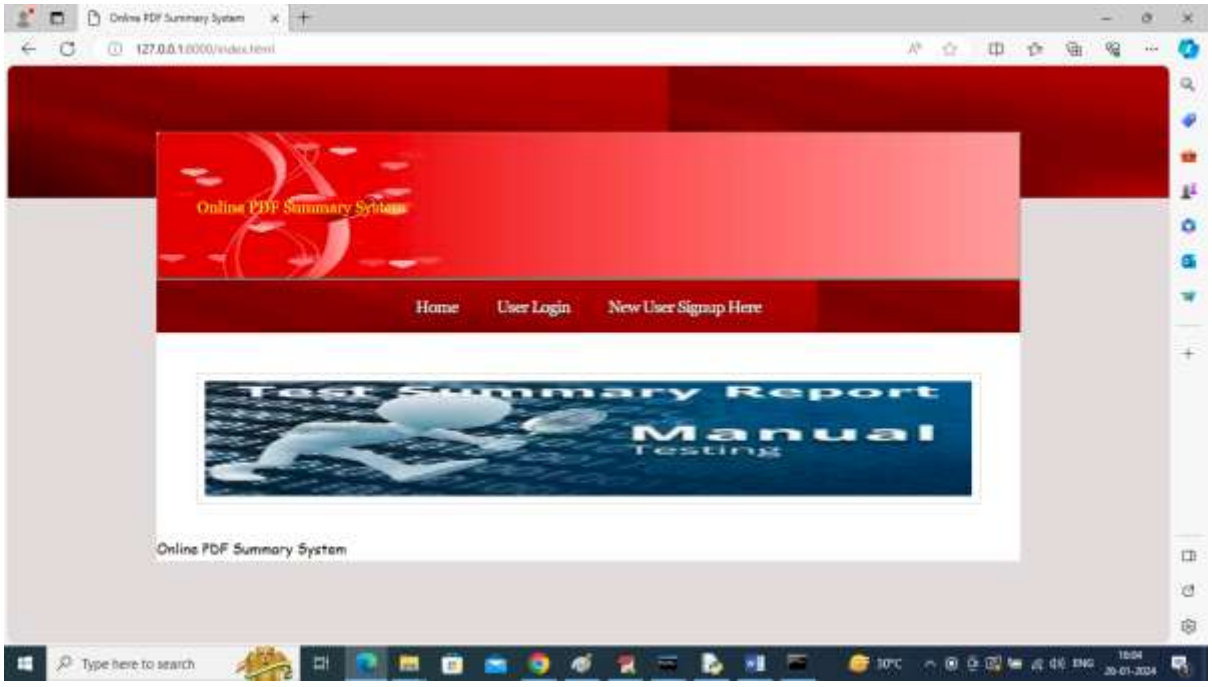
5. Key Design Features

- Modular architecture for scalability
- Integration of NLP and deep learning models
- Real-time processing of user inputs
- Audio support using text-to-speech
- Secure user authentication

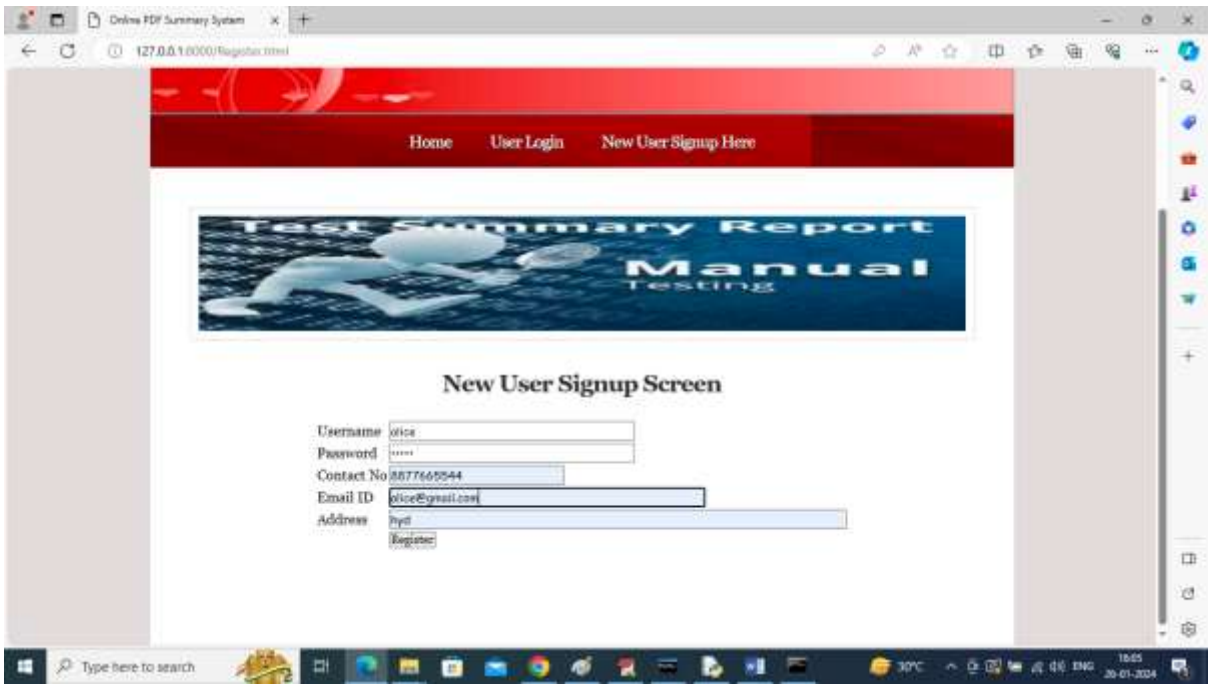
The system design ensures efficient processing, scalability, and maintainability, making it suitable for real-world deployment.

International Journal of Engineering Research and Science & Technology

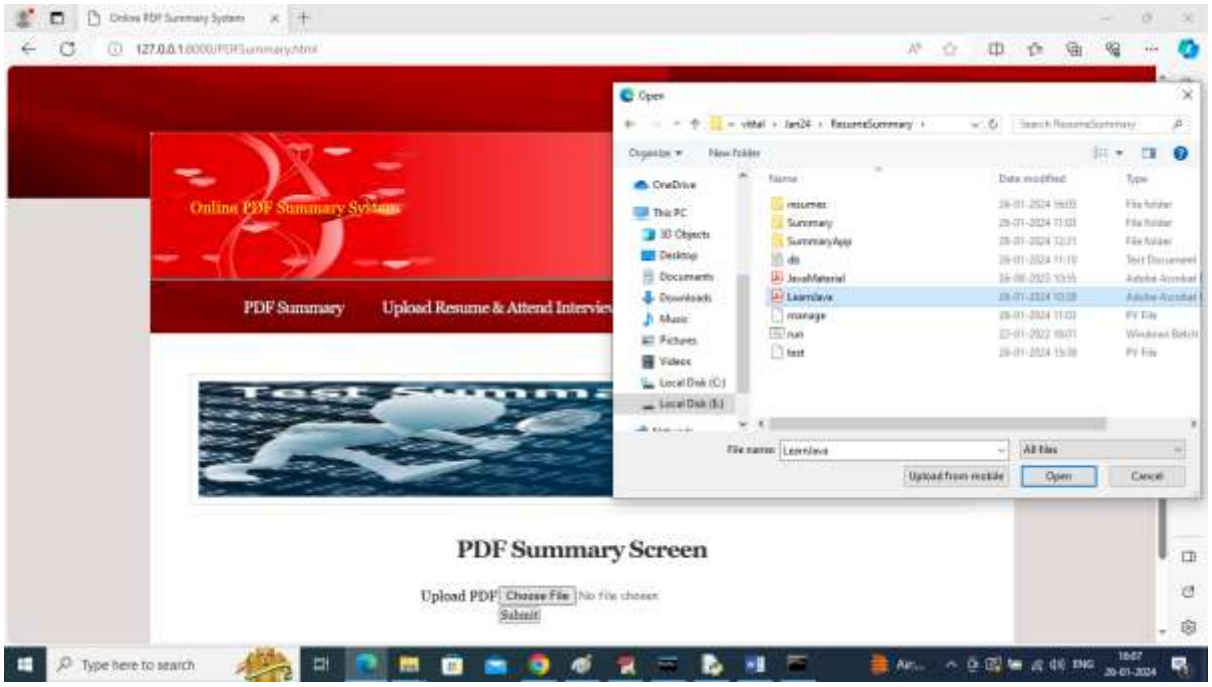
SYSTEM DESIGN IMAGES



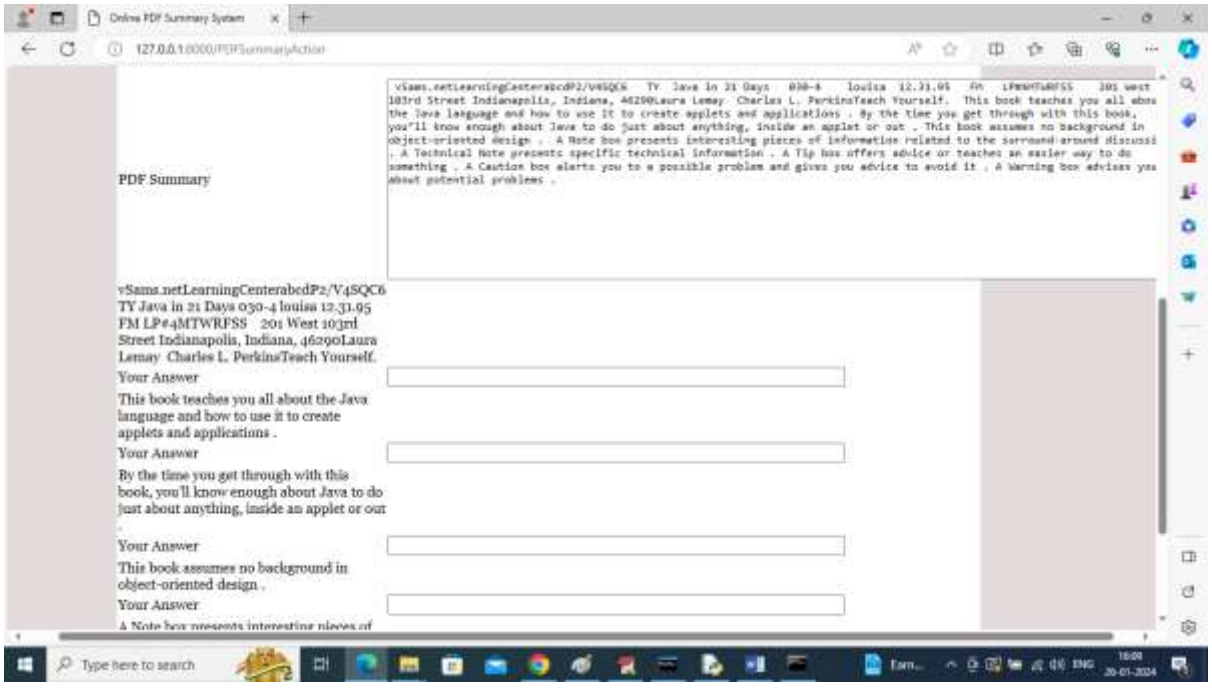
In above screen click on 'New User Signup Here' link to get below page



In above screen user is entering sign up details and then press button to get below page

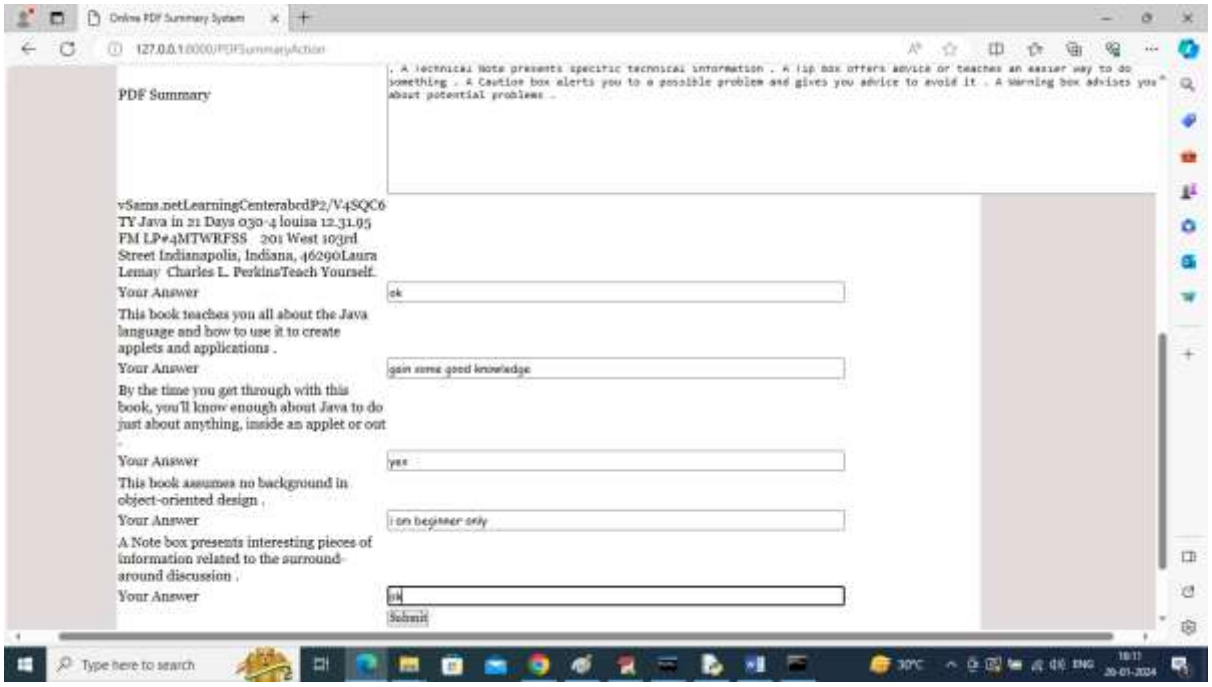


In above screen upload any PDF and then click on ‘Open’ and ‘Submit’ button to get below page



In above screen in text area can see extracted summary from PDF and then can see some questions and in text field user can write some answers and submit them.

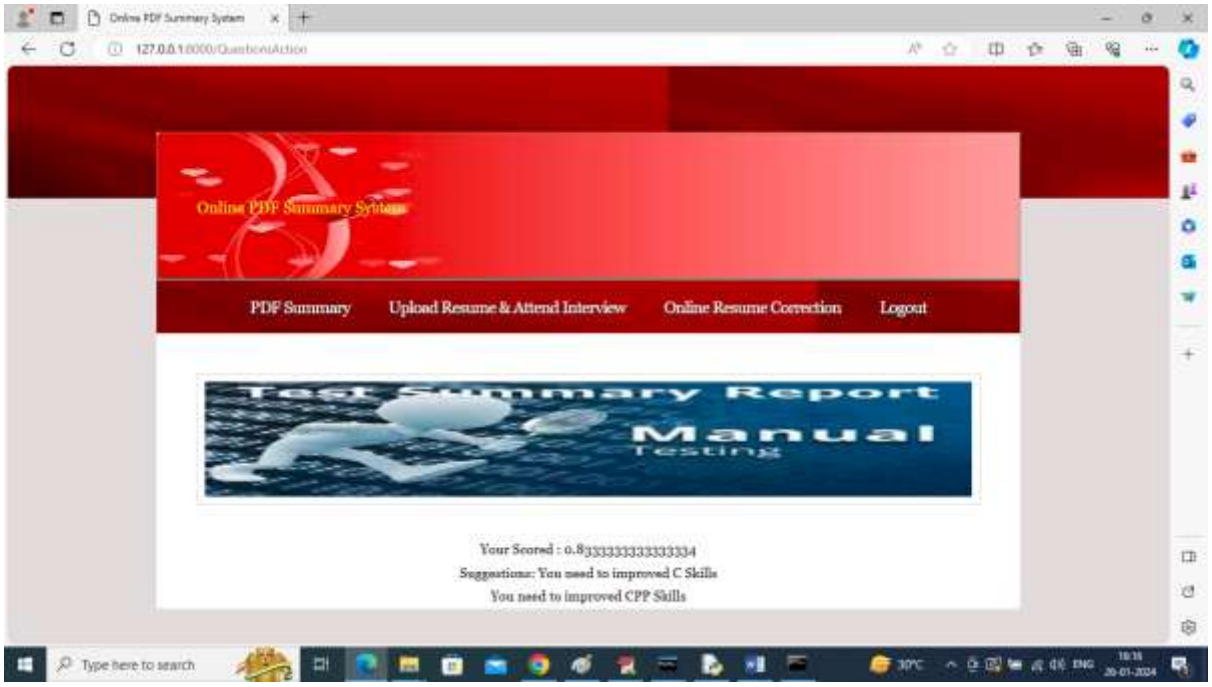
International Journal of Engineering Research and Science & Technology



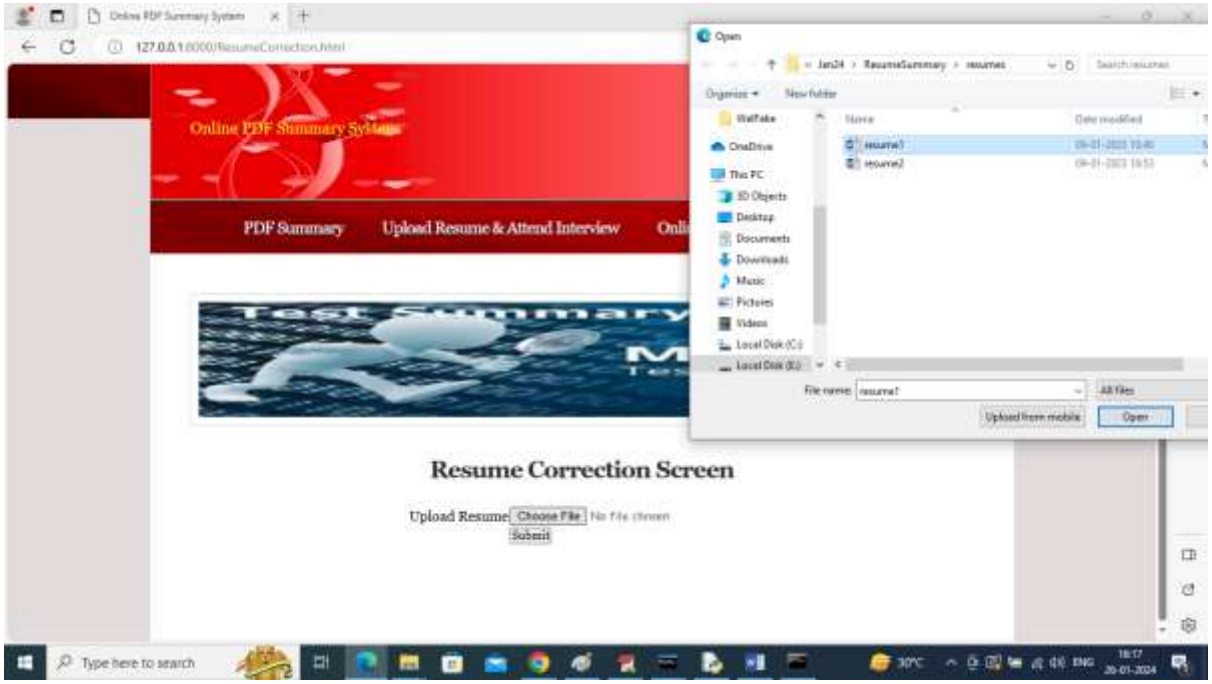
In above screen user will write some answers and then press button to submit answers



In above screen interview questions will sounded to user as ‘Speech to text’ which user can hear and after hearing question he will write correct option and press submit button

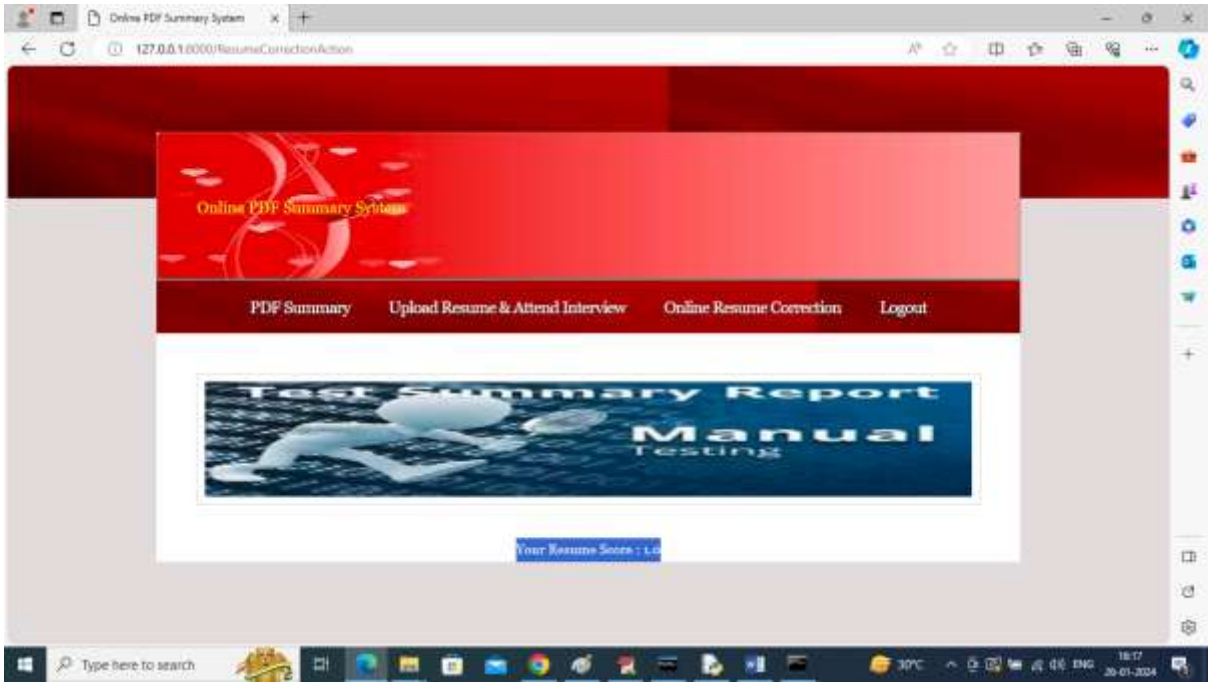


In above screen can see interview questions score and can see suggestions from the application and now click on 'Online Resume Correction' link to get below page

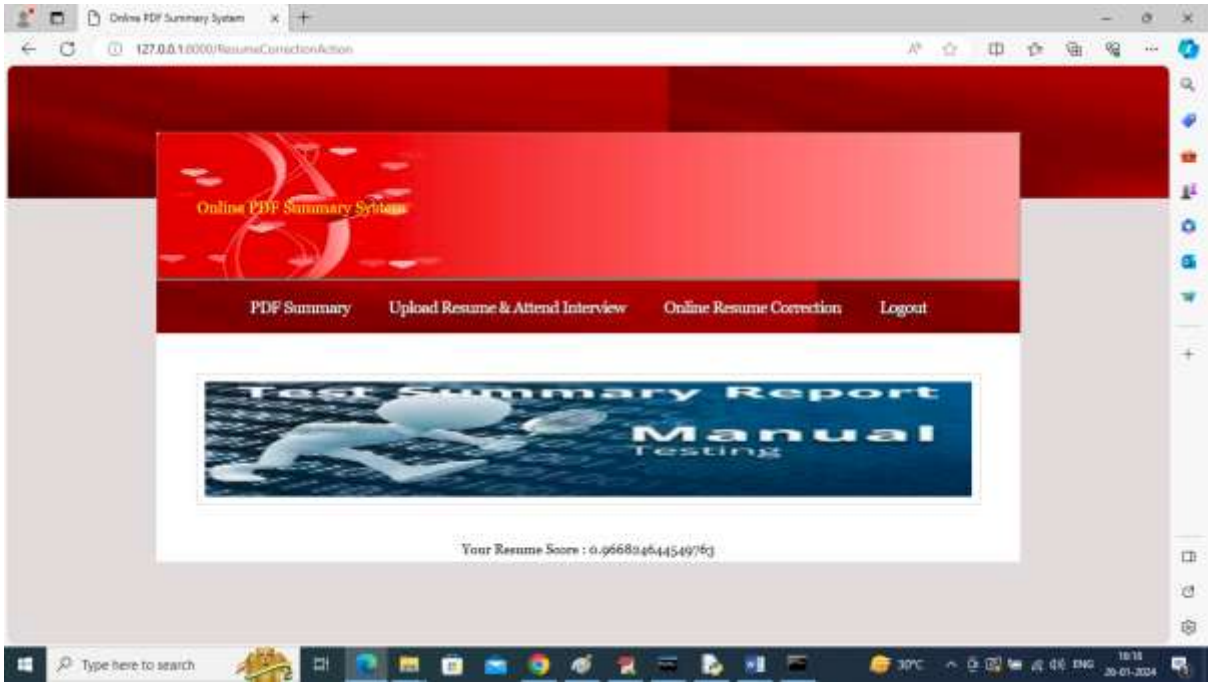


In above screen selecting and uploading resume and then click 'Open' and 'Submit' button to get below page

International Journal of Engineering Research and Science & Technology



In above screen uploaded resume got score as 100% and below is another example



In above screen resume score is 96% and similarly by uploading other resumes you can get scoree

VIII. CONCLUSION

The proposed system successfully integrates multiple advanced technologies, including Natural Language Processing, deep learning, and web development, into a single unified platform. It provides an intelligent solution for resume evaluation, document summarization, and interview preparation, addressing key challenges faced by job seekers.The use of transformer-based models such as T5 significantly enhances the quality of text summarization by generating meaningful and context-aware summaries.
The resume evaluation module provides a simple yet effective way to assess document quality, while the interview module offers personalized feedback to improve technical skills.One of the major strengths of the system is its integration of multiple functionalities into a single platform, eliminating the need for users to rely on separate tools. The addition of text-to-speech functionality further improves accessibility and user engagement.However, the system has certain limitations, such as reliance on basic spell-checking for resume evaluation and predefined questions for interview assessment. Future enhancements may include advanced grammar analysis, AI-based question generation, and adaptive learning mechanisms.In conclusion, the system demonstrates the potential of AI-driven applications in enhancing career preparation processes. It provides a scalable, efficient, and user-friendly solution that can be further extended with emerging technologies.

REFERENCES

1. · Wang, M., et al. “T5-Based Model for Abstractive Summarization,” 2023.
2. · Rao, R., et al. “Automatic Text Summarization Using Transformer Models,” 2024.
3. · Mahmood Abadi, V., et al. “mT5 Transformer for Text Summarization,” 2025.
4. · Lubis, A. R., et al. “Improving Text Summarization Using T5,” 2023.
5. · Tawmo, T., et al. “Comparative Analysis of T5 Model,” 2022.
6. · MDPI. “Comparison of PEGASUS, BART, and T5 Models,” 2024.
7. · Ketineni, S., et al. “Recursive Transformer Networks for Summarization,” 2024.
8. · Anbarasi, M. S., et al. “Abstractive Summarization Using T5 & QLoRA,” 2023.
9. · Xue, L., et al. “mT5: Multilingual Text-to-Text Transformer,” 2020.
10. · Raffel, C., et al. “Exploring Transfer Learning with T5,” 2020.
11. · Devlin, J., et al. “BERT: Pre-training of Deep Bidirectional Transformers,” 2019.

12. · Lewis, M., et al. “BART: Denoising Sequence-to-Sequence Pretraining,” 2020.

13. · Zhang, J., et al. “PEGASUS: Pre-training for Abstractive Summarization,” 2020.

14. · Bahrainian, S. A., et al. “Topic-Focused Abstractive Summarization,” 2023.

15. · Gabín, J., et al. “T5-Based Keyphrase Generation,” 2024.