IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

AI-Driven Personalized Career Guidance and Skill Gap Analysis System

P. D. Lanjewar[1], Gayatri Vijay Patil[2], Neha Atul Patil[3], Harsha Ananta Dahivadkar[4], Rituja Girdhar Bhavsar[5]

Professor, Department of Artificial Intelligence & Machine Learning, R. C. Patel Institute of Technology, Shirpur, India[1]

Student, Department of Artificial Intelligence & Machine Learning, R. C. Patel Institute of Technology, Shirpur, India[2,3,4,5]

Abstract: Abstract: The process of selecting a career is challenging, and the general counseling provides limited knowledge about the necessary skills in a particular profession. Our proposed web application identifies the content of a resume in PDF form and uses it to extract its TF-IDF features to classify it using a model to suggest the domain of the career, whether software engineer or data scientist. It then evaluates the skills present in the resume based on the skills required by the job, suggests the skill gaps, and suggests some courses, certificates, and topics that should be studied for interviews. In addition to this, there is also a chatbot, an interview simulator, resume generator, and ATS score checker as other parts of the application. This application is built using Python, Flask, MySQL, and Scikit-learn. All the modules performed as expected in their tests.

Keywords: Artificial Intelligence, Machine Learning, NLP, Career Guidance, Career Forecasting, Skill Gap Analysis, Resume Screening, TF-IDF, Job Interview Preparation, ATS Score

I. INTRODUCTION

Choosing a career is among the crucial decisions a student faces but has no clue about. They may not know about what the employers expect from a candidate, what skills are required and valued, how their experience and education stack up. Rarely someone explains them clearly what they have and what they miss.

Traditional career advising normally involves a session with a career counselor, some aptitude tests, and review of the student's academic performance. The process takes some time and does not pay attention to the document that reveals much about the applicant – the resume. Resume contains information about education, technical skills, certifications, projects, and work experience. Therefore, it is natural to start the analysis with it. On the other hand, selection of a proper job position is not the whole story. The person should understand his/her gaps, learn how to fill them, prepare for an interview and make a good resume.

Previous research on career recommendation [1] offered useful ways of suggesting jobs, analyzing resume and evaluating skills. However, the same challenges appear again. Gaps in skills are virtually ignored, the matching process depends heavily on keywords from resume, there is almost no support for preparation for an interview, the suggestions are generic, they rely on the quality of input data and cannot be combined. Almost all of the existing solutions solve just one of these issues; hence, one needs multiple apps in order to get the necessary help.

Our objective in this project was to develop a platform where a person can perform all these tasks at once. For recommending the job domain, our system analyzes the uploaded resume using NLP techniques, converts the text into TF-IDF features and lets the classifier decide about the appropriate career domain. After that, our system compares the skills of the person with the requirements of the selected career and identifies the missing items. The possible actions in order to fill those gaps are suggested to the user. Our platform also features a chatbot, interview simulator, resume building and resume analysis tools. All predictions are saved so the user can access them later. What needs to be emphasized immediately is that our tool can provide only recommendations, not guarantees.

II. LITERATURE REVIEW

There exist several types of prior work. In this case, the research can be divided into three major groups: career recommendations research, resume analysis and categorization research, and resume text similarity measurement research.

Review of literature on machine learning methods for career recommendation [1] claims that recommendations can be given based on data on a candidate or a learner. However, the importance of such recommendations will depend on the

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

completeness and accuracy of provided input data. Almost all existing solutions recommend a career but do not provide recommendations on how one can become a suitable candidate for this position.

In Resume2Role [2], TXT and PDF resumes are read, preprocessed with the help of lemmatization, lowercasing, and stop-words removal, transformed to TF-IDF vectors, and then classification algorithms are used. As a result of experiments, it was found that Random Forest was the best classifier, that oversampling technique was useful for the problem of imbalance of classes, and that whole algorithm was implemented as Flask app for resume uploading, domain classification, and job recommendation. It is close to our prediction platform. But while the main goal of the research is to classify and recommend jobs, our prediction platform aims at more complex tasks. Onukwugha et al. [3] tried to improve resumes with the use of machine learning techniques, which shows that the focus of research in this field is changing from classification of resumes to resume improvement. Random Forest [4], which is an ensemble of decision trees implemented in Scikit-learn [5], is usually used for such text classification tasks.

For comparison of resumes to job descriptions, the use of Sentence-BERT embeddings [6] is interesting because it allows to capture semantics instead of word similarities. The popular TF-IDF weightings [7] are used because of fast implementation of the algorithm, easy explanation, and the fact that it can be used even in case of limited training data. Information about text preprocessing, tokenization, and vectorization can be found in [8] and [9]. Comparison of these approaches to our proposed platform is given in Table I.

TABLE I COMPARISON OF RELATED WORK

Author(s)	Main focus	Gap addressed in this work
A. Jain and B. B. Gupta [1]	Review of ML-based career recommendation	Adds explicit skill-gap identification and learning guidance after prediction
A. Dubey, N. Dhanda, A. Lovewanshi and V. Yadav [2]	Resume parsing, TF-IDF and domain classification in a Flask application	Extends prediction with skill-gap analysis, recommendations, interview support and a resume builder
C. Onukwugha, C. Ofoegbu, O. Aliche and C. Betrand [3]	Resume optimization using ML techniques	Combines resume improvement (builder, ATS evaluation) with career prediction in one platform
N. Reimers and I. Gurevych [6]; G. Salton and C. Buckley [7]	Semantic embeddings and TF-IDF weighting for text matching	Uses TF-IDF for lightweight, interpretable features; embeddings are left as a future extension

After analyzing each of the studies, it becomes apparent that each of the studies serves only one purpose and that can either be advice on careers or resume building. Skills assessment, mock interviews, and resume creation are not offered in one program together very often, and that is what we intend to do.

III. METHODS AND MATERIALS

The methodology of the research includes six stages: resume processing, feature extraction, career prediction, skill gap identification, recommendations generation and data storage. The interview and resume writing tools were developed following the above pipeline.

A. System Architecture

Four-layer architecture was used for implementation of the platform (Fig. 1). The first layer, the presentation layer, comprises all user-visible components: the dashboard, resume upload, career prediction, skill gap analysis, resume writing using AI, ATS analysis, learning roadmap, chatbot and interview simulation. Then, there is the Flask application layer, providing user authentication and authorization, resume processing, prediction, skill gap analysis, recommendations generation, resume writing, ATS analysis, AI career mentor and interview simulation. Next, there is the AI/ML layer, dealing with analytics tasks (preprocessing, feature extraction, prediction, skills matching, resume creation and ATS analysis), and having access to external services for accessing API of AI, PDF processing, speech and text to speech recognition. Finally, there is the MySQL database layer storing users, resumes, predictions, skill gap

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

analysis results, learning roadmaps, recommendations, chat history, interview sessions and answers, generated resumes and ATS analysis results.

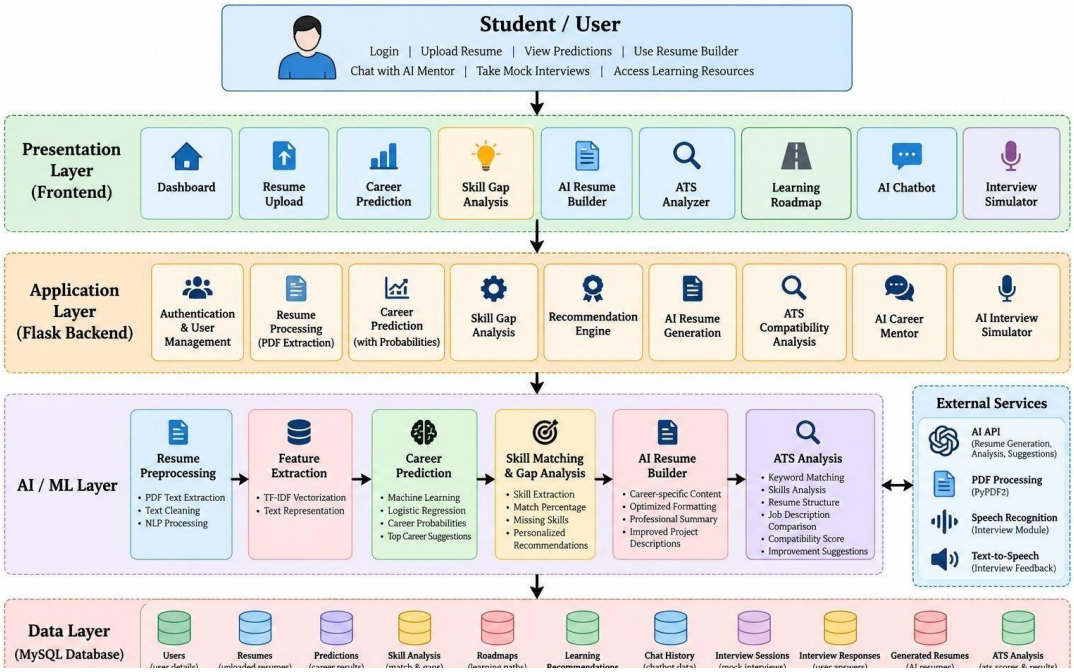


Fig. 1. Proposed system architecture

The typical process would be as follows. The user signs up, logs in, and uploads his/her resume in the PDF format. Using the PyPDF2 library, the text is extracted from the PDF, and NLP preprocessing is performed. The system identifies skills, qualifications, certificates, and projects of the user, creates TF-IDF vector out of the text, and requests the career domain from the classifier. Then, the skill-gap module gets the career domain and identifies skills that are needed in this particular domain, compares them with the skills in the resume, and provides recommendations.

B. Resume processing and feature extraction

The processing starts right away after the PDF resume is received and its text is read aloud. We clean up all the extra symbols and unimportant words according to common NLP preprocessing rules [8]. Then we extract skills, qualifications, certifications and keywords and use TF-IDF vectorization [7], [9] to convert our text to numerical values which the model can work with. This way we do not have to go through each resume manually. But still, our outcome depends on the structure of the resume, extraction process, and information provided.

C. Career Prediction

To predict the career choice of a person, the module loads the model that is already trained with the features extracted from the resume. Skills, education, certification, and project details are taken into account, vector representation is prepared, and these vectors are matched against the training vectors. The output for the user would be a career domain along with the probability score. Career domains that we support are Software Engineer, Data Scientist, Web Developer, Cloud Engineer and Database Developer. Logistic regression and Random Forest [4] classifiers, which use several decision tree models to increase accuracy, are very suitable for this problem, and there are Scikit-learn [5] implementations available.

D. Skill Gap Analysis

After selecting the area to which the candidate belongs, the software identifies the skills required for that particular role and verifies each skill against the candidate's skills list. Every missing skill is considered a skill gap, and the report recommends certain certification and learning material and interview questions. In the case of a data analyst role, the skills that could be required include Python, SQL, statistics, and data visualization; if there are no such skills in the resume or some of them, then those lacking are considered skill gaps.

E. Recommendations for Learning

These recommendations will stem from the predicted career along with the gaps identified. Technical skills, certifications, study materials, subjects, projects, and interview preparation may be suggested based on the resources that have been

DOI: 10.17148/IJARCCE.2026.151102

provided in the application. Because these recommendations are based on the resume of the user himself/herself along with the predicted domain, these will be personalized.

F. AI Chatbot and Interview Simulator

The chatbot provides the ability for users to have a dialogue and ask questions regarding careers, skills, and preparation. Answers generated will depend on how the chatbot is implemented, as well as on any additional external AI-based services used for that purpose, therefore, the scope of questions and their quality will depend on these components as well. In case of the interview simulator, the user practices answering questions, which are similar to the ones he or she will have in the real-life situation. With voice inputs available where appropriate, the interview process becomes even closer to reality.

G. Resume Builder and ATS-Friendly Resume Evaluator

The resume builder enables users to organize the information into a proper format. Resumes created by the resume builder are stored in the database. ATS-Friendly Resume Evaluator offers a score indicative of your resume content, keywords, formatting and the level of compliance with a job description. Suggestions for improvements are also offered by the tool. It is worth noting that the score is not the score of an official applicant tracking system.

H. Implementation and Technology Stack

We have implemented our platform as a web application in Python using Flask and MySQL technologies. The frontend provides the interaction with the user, while the backend receives the request and communicates with the database. The interface is designed using HTML, CSS, and JavaScript for providing interactivity. Data processing is carried out with Pandas, numerical calculations – with NumPy, machine learning – with Scikit-learn and TF-IDF. We have used PyPDF2 for extracting the text from the PDF resumes. Training and testing of the model was conducted in Jupyter Notebook, and the development was done in Visual Studio Code.

The new user signs up using his name, e-mail address and password. The information provided is stored in the database. The returning user signs in using his e-mail and password. Flask verifies the credentials, giving access to the dashboard to valid users. Depending on the features that are used by the application, MySQL stores the information about the user, resumes, career prediction, skill analysis, learning recommendation, roadmap, interviews, questions and answers, chatbot conversations, resumes generated by the platform and its analysis by the ATS.

IV. RESULTS AND DISCUSSION

A. Testing Strategy

The system was tested both in isolation per module and as a whole when all components were integrated. Testing of modules entailed studying the behavior of individual features of the system whereas testing the integration aspect entailed checking the interaction between the frontend, backend, and the database. Modules that underwent testing include: Registration & Login, Resume Upload, Text Extraction, Career Prediction, Skill Gap Analysis, Recommendations, Interview Suggestions, Prediction History and Dashboard. In all tests, appropriate input was fed and compared to expected output..

B. Findings

All major modules we tested functioned properly. Registration and login worked correctly. The upload page was able to accept PDF format resumes and the extracted texts were processed using NLP techniques, allowing for a proper analysis of skills, qualifications, certifications and projects. This allows users to avoid a lot of manual work involved in resume reading.

For example, during one of our testing sessions, the prediction module predicted Software Engineer as the best career match with 50.0% confidence (Fig. 5). Skill-gap page corresponding to the tested resume showed 55.56% skill compatibility and 44.44% educational gap. The skills detected by the program were Python, Java, Coding, Git and Frontend and the recommended road map for skill development included four stages: Backend, OOP, Data Structures and Algorithms (Fig. 6). On the learning resources page, intermediate-level courses for each of these topics with a duration of 6–8 weeks (8–10 weeks for Algorithms) and other sources (YouTube, practice, projects and studying materials) are provided (Fig. 7).

Such an approach is better than just naming the right career, as it allows a user to see how he or she can become skilled enough to take this position. Moreover, the platform maintains a history of predictions and puts all of its main modules into one interface - the dashboard.

C. System Walk-through

A new user begins by registering themselves through the registration page (Figure 2) and logging in via the login page (Figure 3). The next step involves uploading the resume via the upload page (Figure 4), where a PDF file not exceeding

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

10MB can be dragged-and-dropped or clicked. The output is shown in Figure 5 as the predicted career domain and the degree of accuracy. Then the skill gap page (Figure 6) reveals the user's skills, the skill gaps, and a learning plan designed for the individual.

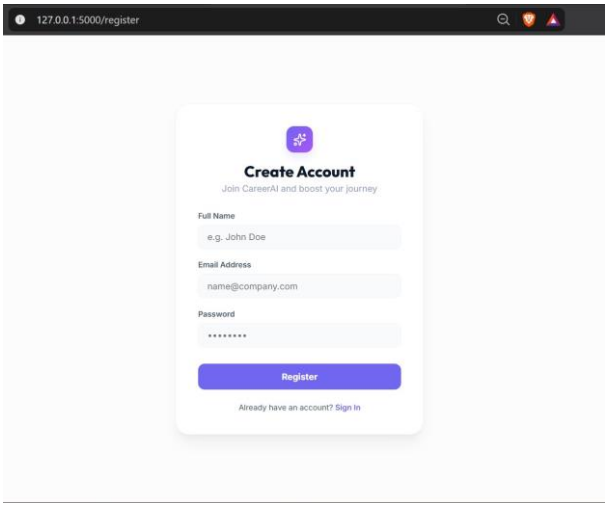


Fig. 2. User register page

Registration page on which a new user provides his/her name, e-mail ID, and password for registering himself/herself.

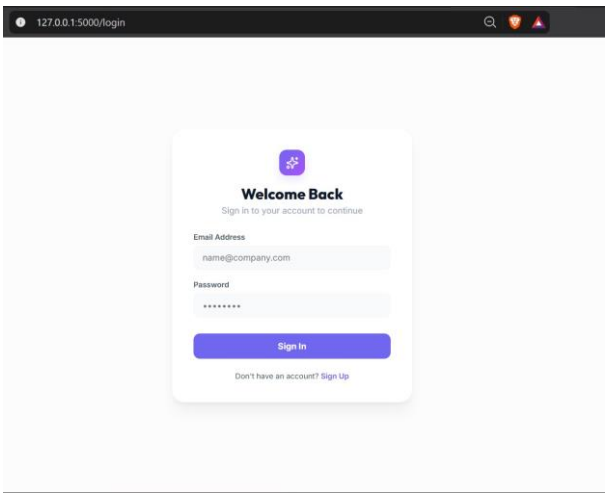


Fig. 3. User login page

Page to log in for the registered user using email id and password.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

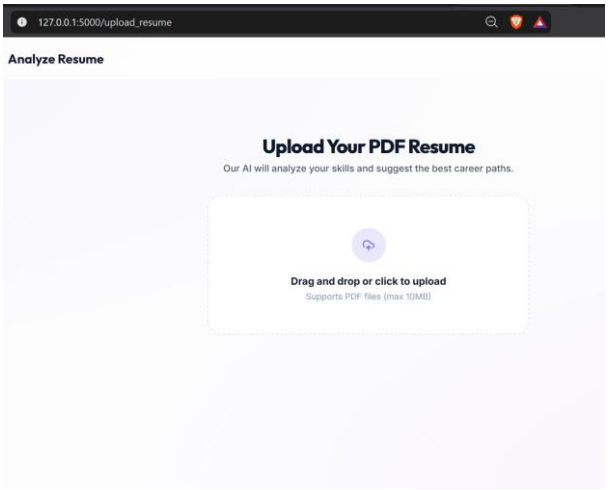


Fig. 4. Resume upload page

Resume Upload Page that requires a maximum file size of 10MB in PDF format to be uploaded.

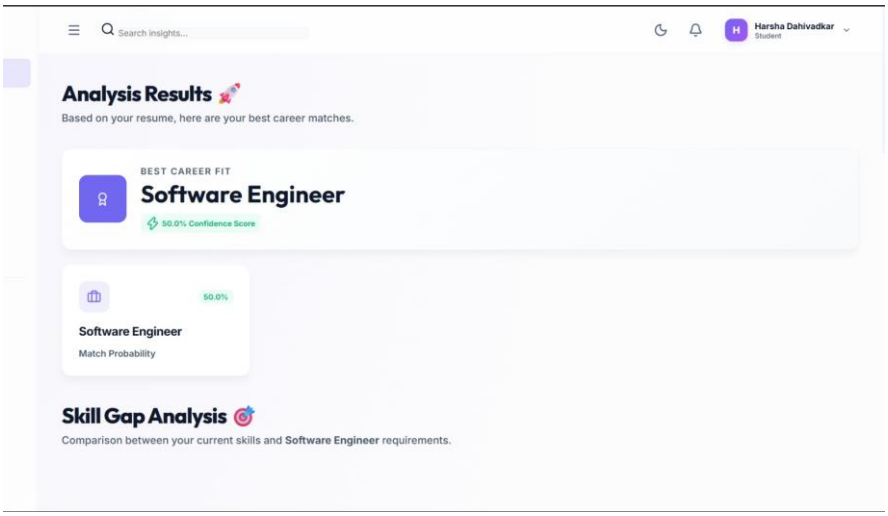


Fig. 5. Resume analysis and career prediction page

Results page displaying the predicted optimal career match along with its level of confidence, Software Engineer.

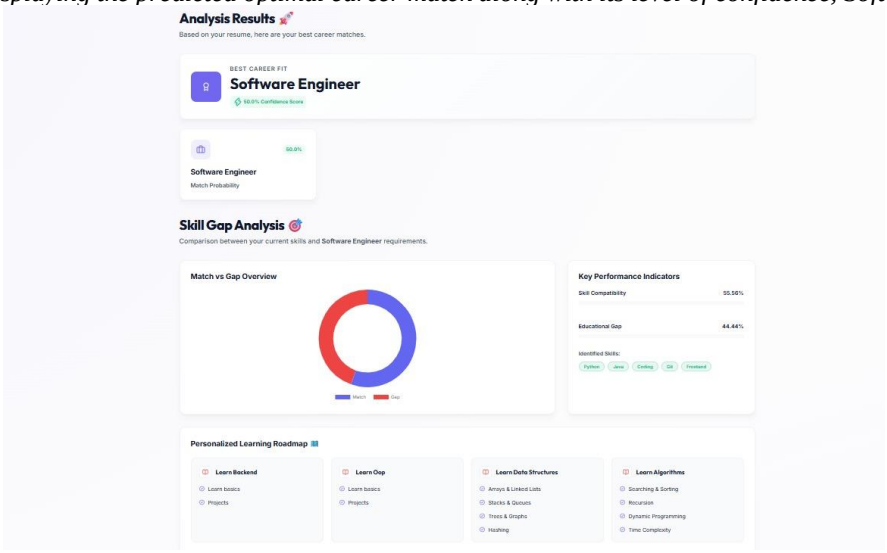


Fig. 6. Skill gap analysis and career guidance page

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

Gap analysis page containing the match versus gap graph, key indicators, skills and a custom learning path.

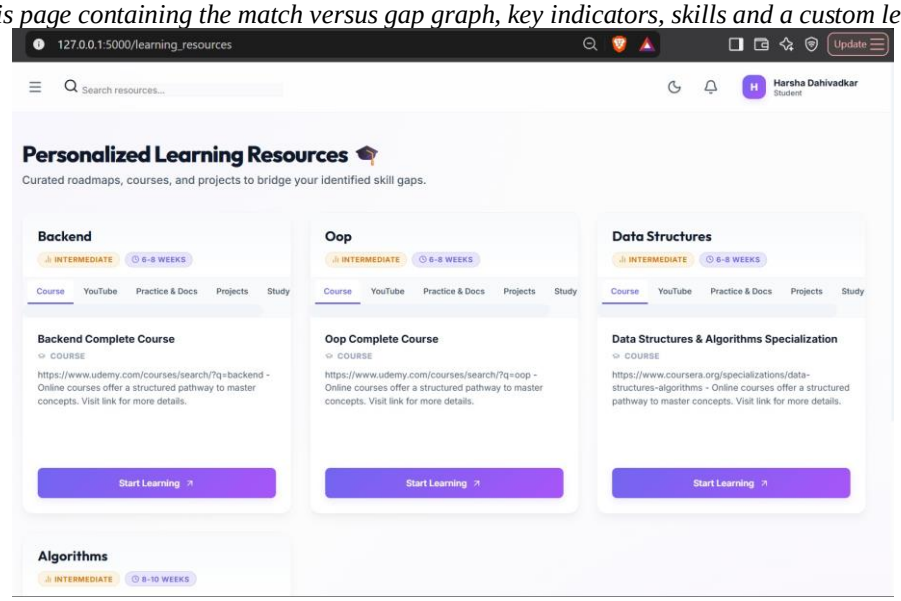


Fig. 7. Personalized learning resources page

The interview simulation tool follows next. Prior to starting the session, the user picks its length (10-30 min), the desired role, the type and level of the interview, while on the session setup page they can see the profile that the questions are based upon: the desired career, the skills that appear on the resume and the skill gaps which should be tested (Fig. 8). In the course of the session, the questions are asked by the AI interviewer one by one, the progress bar displays the current stage, the timer counts down the remaining time and the status of the camera and microphone is shown. In case the voice input does not work properly, the user can enter the answer manually (Fig. 9). After the session ends, the performance report provides an overall score, the scores for the sub-factors: technical knowledge, communication skills, problem solving skills, resume knowledge and project understanding, as well as the strengths, areas for improvement, the improvement plan and the analysis of each individual question (Fig. 10).

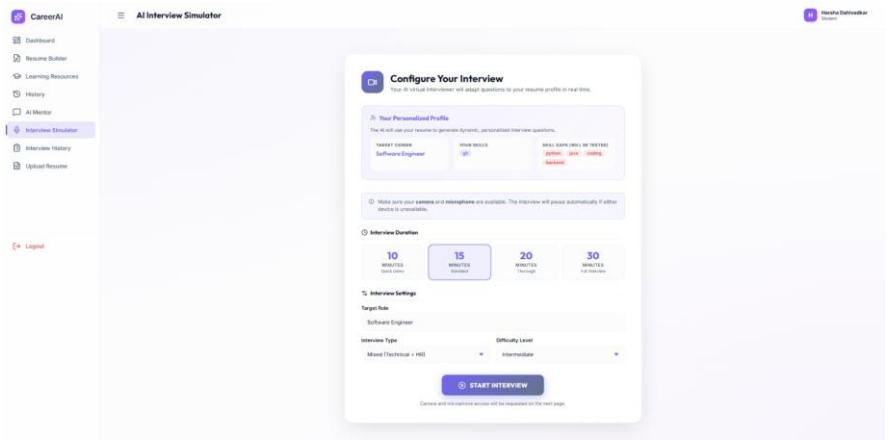


Fig. 8. Interview simulator setup page

Setup page for choosing duration, role, interview type and difficulty, with the personalized profile used for questions.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

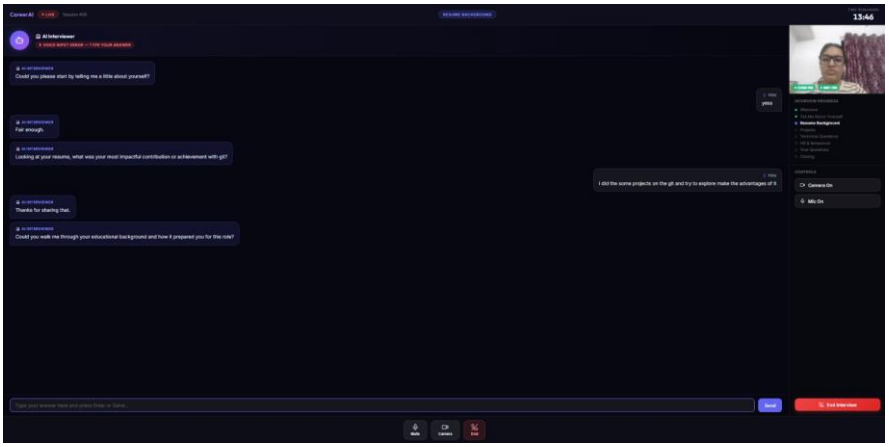


Fig. 9. Live interview session page

Live interview page showing the AI interviewer's questions, progress tracker, timer and typed-answer fallback.

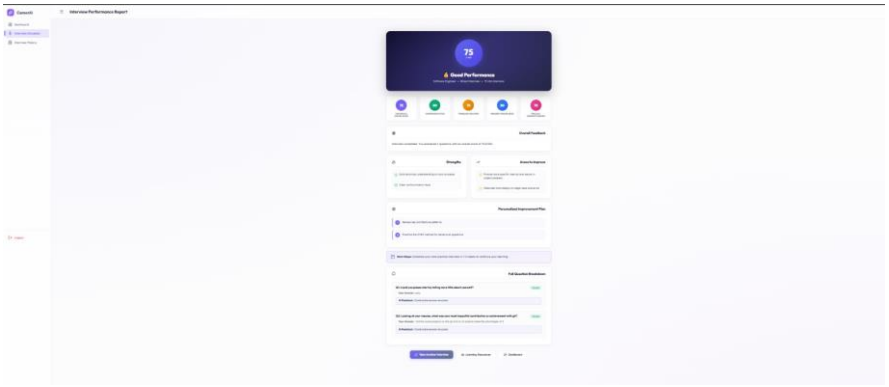


Fig. 10. Interview performance report page

modules went well, and we believe the platform can help users see where their skills stand, find fitting career

D. Conclusion and limitations

The confidence score obtained is quite moderate, indicating that there is a need for a bigger and more balanced training data set. Our model has not been calculated for accuracy, precision, recall and F1-score yet, and we will provide such metrics only after calculating them on a test data set. The ATS score is indicative only. The skills gap report is dependent on how effectively the skills are extracted from the CV and how the job requirements are defined within the application. The chatbot and interview questions depend on the AI technology behind them. For this reason, the results provided by the platform can only serve as an initiation to a career conversation.

V. CONCLUSION AND FUTURE WORK

In an effort to create one solution to assist students and job seekers with exploring and preparing for careers, a web-based platform capable of doing just this using only one resume was created. Skills, qualifications, certifications and projects are identified, TF-IDF feature extraction technique turns the raw text into useful features, classifier classifies the resume into career domain and compares skills of a person with those required for the selected career. On top of all this, the platform offers its users career learning recommendations, certification recommendations, chatbot, interview simulator, resume creation, ATS evaluation of resume and past predictions. Testing of core components of the platform was conducted successfully and we believe that this platform will assist users in evaluating their skills, identifying career domains they fit and developing further steps towards these goals.

The recommendations will always depend on completeness of the resume and career domain data. In the future, we plan to train on larger datasets and continuously updating industry requirements, use live job market data, trends and online education platforms, improve resume identification process, further personalize the output, make the solution mobile-friendly and evaluate it using appropriate test cases and datasets.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151102

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Prof. P. D. Lanjewar for his guidance, support, and technical inputs during the course of this project. The authors are grateful to Prof. Dr. U. M. Patil, Head of the Department; Prof. Dr. J. B. Patil, Director; and other faculty members of the Department of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur.

REFERENCES

[1] A. Jain and B. B. Gupta, "Machine learning based career recommendation systems: A review," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, pp. 125-132, 2021.

[2] A. Dubey, N. Dhanda, A. Lovewanshi, and V. Yadav, "Resume2Role: Intelligent resume parsing and domain prediction using NLP and classification models," ITM Web of Conferences, vol. 86, Art. no. 01008, 2026, doi: 10.1051/itmconf/20268601008.

[3] C. Onukwugha, C. Ofoegbu, O. Aliche, and C. Betrand, "Resume optimization model using machine learning techniques," International Journal of Intelligent Information Systems, no. 5, pp. 109-116, 2024.

[4] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

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

[6] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982-3992.

[7] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988.

[8] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA, USA: O'Reilly Media, 2009.

[9] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008.

 This work is licensed under a Creative Commons Attribution 4.0 International License