**ATS- Resume Builder Using NLP**

**7COM1039 – Advanced Computer Science Masters Project**

**Sanofer Sameena Riyas (22080596)**

**Table of** **Contents**

[Aim of the Project 3](#_Toc169194414)

[Research Question/Hypothesis 3](#_Toc169194415)

[Objectives 3](#_Toc169194416)

[Description 4](#_Toc169194417)

[Methodology 5](#_Toc169194418)

[Bibliography 6](#_Toc169194419)

[Appendices 6](#_Toc169194420)

## **Aim of the Project**

The primary aim of this project is to develop an interactive, user-friendly resume builder application that generates ATS (Applicant Tracking System)-friendly resumes using ReactJS for the front-end and Natural Language Processing (NLP) techniques to enhance the content quality. A "user-friendly" application features an intuitive and interactive interface that simplifies complex processes for the user, often through guided input forms, clear navigation, and real-time feedback. This reduces the learning curve and minimizes user errors. An "ATS-optimized " system, on the other hand, ensures that outputs, such as resumes, are formatted and structured to be easily processed by Applicant Tracking Systems (ATS). This involves using standard templates, including relevant keywords, and avoiding formats or elements that could confuse automated systems.

## **Research Question/Hypothesis**

**Research Question:** How can Natural Language Processing (NLP) be integrated into a resume builder application to optimize resumes for ATS compatibility and enhance content quality?

**Hypothesis:** Integrating NLP techniques for keyword extraction, grammar checking, and content enhancement will significantly improve the effectiveness and quality of resumes generated by the application, thereby increasing the chances of resumes passing through ATS filters.

## **Objectives**

1. **Develop a User-Friendly Interface:** Create an intuitive and interactive front-end using ReactJS.
2. **Implement User Authentication:** Ensure secure sign-up, log-in, and profile management functionalities
3. **Facilitate Resume Creation:** Provide guided input forms for users to input their personal details, work experience, education, and skills.
4. **Optimize for ATS Compatibility:** Design pre-formatted resume templates that meet ATS standards.
5. **Integrate NLP for Content Enhancement:**

* Implement grammar and spell check functionalities.
* Provide keyword suggestions based on job descriptions.
* Offer content enhancement suggestions, such as action verbs and impactful phrases.

1. **Enable Customization and Export Options:**

* Offer multiple resume templates and customization options.
* Provide download options in various formats (PDF, DOCX) and shareable resume links.

1. **Conduct Market Research:** Understand user needs and preferences through surveys.
2. **Assess Technological Feasibility:** Investigate the integration challenges of ReactJS and NLP.
3. **Ensure Legal Compliance:** Adhere to data privacy and copyright laws.
4. **Evaluate Resource and Cost Requirements:** Determine the availability of skilled developers and estimate the project's budget.

## **Description**

The project aims to develop a sophisticated resume builder application that leverages Natural Language Processing (NLP) to enhance the quality and ATS compatibility of resumes. Users will benefit from a seamless and interactive experience, with features such as guided input forms, real-time previews, and NLP-driven content suggestions. The application will also include multiple customizable templates and export options to cater to diverse user needs.

**Natural Language Processing (NLP):** Natural Language Processing (NLP) is a branch of artificial intelligence[5] that focuses on the interaction between computers and human language. It involves creating algorithms and models that enable computers to understand, interpret, and generate human language in a meaningful and useful manner. Key components of NLP include text analysis[6], which involves breaking down text into smaller units (tokenization) and identifying grammatical parts of speech (part-of-speech tagging). Language understanding tasks such as Named Entity Recognition (NER) detect and classify entities like names, dates, and locations, while sentiment analysis determines the emotional tone of text. NLP also encompasses language generation tasks such as text summarization, which creates concise summaries of larger texts, and machine translation, which translates text between languages. Additionally, content enhancement features like grammar and spell checking help identify and correct errors, and keyword extraction identifies important terms relevant to the content. Information retrieval applications, such as search engines and document classification systems, benefit from NLP by enhancing query understanding and categorizing documents based on their content. By integrating techniques from computer science, linguistics, and machine learning, NLP facilitates more natural and intuitive interactions with technology, powering applications like chatbots, voice assistants, and automated content analysis tools.

**Usage of NLP in ATS:** In the context of an Applicant Tracking System (ATS), Natural Language Processing (NLP) plays a crucial role in enhancing the effectiveness and efficiency of resume screening and matching candidates to job postings [5]. ATS is software used by employers to manage the hiring process by automatically sorting through resumes to identify the best candidates. However, traditional ATS systems often face challenges in accurately parsing and evaluating resumes due to variations in formatting, language, and content quality. This is where NLP becomes instrumental.

NLP techniques enable the ATS to better understand and process human language by performing tasks such as keyword extraction, grammar and spell checking, and content enhancement [6]. For instance, NLP can identify key skills and experiences from a resume and match them to the requirements of a job description [1], improving the chances of relevant resumes passing through the initial filters. Additionally, NLP can analyze the sentiment and context of the text, ensuring that resumes are not just keyword-stuffed but genuinely align with the job criteria [2]. By leveraging NLP, an ATS can offer more accurate and fair assessments of candidates, streamline the recruitment process, and help employers find the most suitable candidates efficiently.

## **Methodology**

1. **Literature Review:**

* Review existing literature on NLP applications in resume building and ATS systems.
* Study current resume builder tools to identify gaps and opportunities.

1. **Data Collection:**

* Gather data on user preferences through surveys and interviews.
* Collect sample job descriptions and resumes to train NLP models.

1. **Model Development:**

* Use ReactJS for front-end development and Node.js/Express.js for the back-end.
* Develop NLP algorithms using Python libraries such as NLTK, spaCy, and transformers.

1. **Feature Implementation:**

* **User Authentication:** Implement secure sign-up and log-in functionalities with profile management.
* **Resume Creation**: Create guided input forms and real-time preview features.
* **NLP Integration:** Develop and integrate NLP algorithms for keyword extraction, grammar checking, and content enhancement.
* **Template Customization:** Design multiple ATS-friendly templates and customization options.
* **Export Options:** Provide options to download resumes in various formats and generate shareable links.

1. **Testing and Validation:**

* Conduct extensive testing, including unit testing, integration testing, and end-to-end testing.
* Collect user feedback to refine the application.

1. **Deployment and Documentation:**

* Prepare the application for deployment using containerization tools like Docker and host it on cloud services such as AWS, GCP, or Azure.
* Create comprehensive user documentation and tutorials to assist users.

## **Bibliography**

1. Bozzon, A., & Silvestri, M. (2020). Enhancing Job Seeker Experience through Natural Language Processing Techniques. *International Journal of Information Management, 50,* 26-34.
2. Jatowt, A., & Kazama, K. (2019). NLP for Improving Job Matching and Resume Screening. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* 645-655.
3. Ruder, S. (2019). Neural Transfer Learning for Natural Language Processing. *University of Edinburgh.*
4. Jadhav, A. M., & Gadekar, D. P. (2014). A survey on text mining and its techniques. *International Journal of Science and Research (IJSR),* 3(11), 2110-2113.
5. Sharma, A., Singhal, S., & Ajudia, D. (2021, September). Intelligent Recruitment System Using NLP. In 2021 *International Conference on Artificial Intelligence and Machine Vision (AIMV)* (pp. 1-5). IEEE.
6. Jaiswal, G., Uttam, A., Dubey, D. D., & Mall, P. K. (2024, March). Resume Analyser and Job Recommendation System Based on NLP. *In 2024 2nd International Conference on Disruptive Technologies (ICDT)* (pp. 1584-1587). IEEE.
7. Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna *Design and Development of Machine Learning based Resume Ranking System* Global Transitions Proceedings (2021).
8. Nirali Bhaliya, Jay Gandhi, Dheeraj Kumar Singh , *NLP based Extraction of Relevant Resume using Machine Learning* in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-7, May 2020.
9. *Resume Screening Using Machine Learning and NLP* : A Proposed System Bhushan Kinge 1, Shrinivas Mandhare2, Pranali Chavan3, S. M. :https://doi.org/10.32628/CSEIT228240 Chaware4, doi.

## **Appendices**

**Appendix A: Detailed Timeline:**

**Phase 1: Planning and Design (Weeks 1-2)**

* Define project requirements and scope.
* Design wireframes and UI mockups.
* Set up project repository and development environment.

**Phase 2: Front-End Development (Weeks 3-5)**

* Implement user authentication and profile management.
* Develop guided input forms for resume creation.
* Create a real-time resume preview feature.

**Phase 3: Back-End Development (Weeks 6-8)**

* Set up the back-end server with Node.js and Express.js.
* Implement APIs for user data and resume storage.
* Integrate authentication mechanisms.

**Phase 4: NLP Integration (Weeks 9-11)**

* Develop NLP algorithms for keyword extraction, grammar check, and content enhancement.
* Integrate NLP functionalities with the front-end.

**Phase 5: Template Customization and ATS Optimization (Weeks 12-13)**

* Design and implement multiple resume templates.
* Ensure templates are ATS-friendly.
* Add customization options for users.

**Phase 6: Testing and Debugging (Weeks 14-15)**

* Conduct unit testing, integration testing, and end-to-end testing.
* Fix bugs and optimize performance.

**Phase 7: Deployment and Final Review (Weeks 16-17)**

* Prepare for deployment (containerization, CI/CD setup).
* Deploy the application on the chosen hosting service.
* Conduct a final review and gather feedback for improvements.

**Phase 8: Documentation and User Training (Week 18)**

* Create user documentation and tutorials.
* Provide training materials for end-users.

**Appendix B: Survey Questions for Market Research**

1. What features do you find most important in a resume builder application?
2. How frequently do you use resume builder tools?
3. What challenges do you face when creating a resume?
4. How familiar are you with ATS (Applicant Tracking System) and its requirements?
5. Would you find NLP-driven content enhancement suggestions useful?

**Appendix C: Technical Specifications**

* **Front-End:** ReactJS, Redux (for state management), CSS/SCSS, Bootstrap or Material-UI.
* **Back-End:** Node.js, Express.js, MongoDB (for user data and resumes storage).
* **NLP Technologies:** Python with libraries such as NLTK, spaCy, or transformers for NLP functionalities.
* **Hosting and Deployment:** AWS/GCP/Azure, Docker for containerization.