



AI Resume Maker

Course Code & Name: CSC301 - Seminar

Prepared by: LAKSHAY AGGARWAL / RIBHAV BHALLA

Roll Number: 22CSU272 / 22CSU261

Instructor: Mr. Sumit Kumar

Date: [01/05/2025]

THE NORTHCAP UNIVERSITY

Gurugram, India

Index

1. Introduction	Page 3
2. Background / Literature Review	Page 4
3. Research Methodology	Page 5
4. Findings & Analysis	Page 6
5. Conclusion & Recommendations	Page 7
6. References	Page 8
7. Appendices (if any)	Page 9

1. Introduction

- Brief Overview:

This project focuses on building an AI-powered resume generator that transforms unstructured, paragraph-style user input into structured resume data using Large Language Models (LLMs).

- Purpose and Objectives:

The goal is to automate and simplify resume creation using AI. Key objectives include:

- Extracting structured data from user input.
- Allowing user edits through a dynamic form.
- Generating a formatted resume that can be exported.

- Scope and Limitations:

Scope: Resume generation from paragraph input, real-time form editing, and PDF output.

Limitations: Currently deployed locally; uses a single resume template; accuracy of AI parsing depends on prompt quality.

- Significance of the Study:

Helps job seekers quickly generate professional resumes without requiring formatting or writing expertise. Demonstrates practical use of LLMs and full-stack development.

2. Background / Literature Review

Relevant Background:

Traditional resume builders require users to manually enter data. AI provides a way to automate this by interpreting natural language descriptions.

Key Findings from Past Research:

Studies show LLMs like ChatGPT and DeepSeek can perform well in information extraction and text classification tasks.

Theoretical Framework:

Based on NLP concepts such as Named Entity Recognition (NER), prompt engineering, and structured data mapping.

Recent Advancements:

The emergence of open-source LLM hosting (Ollama) and lightweight frontend tools (Vite, DaisyUI) enables fast prototyping of intelligent applications.

3. Research Methodology

Data Collection Methods:

Case study approach using user-generated paragraph inputs and observing the accuracy of AI-generated JSON.

Tools and Techniques:

- Ollama with DeepSeek/ChatGPT for AI-based JSON extraction.
- Spring Boot for API and backend logic.
- ReactJS, TailwindCSS, and DaisyUI for the frontend.
- JSON used as the data format for frontend-backend communication.

Sampling and Data Sources:

Test cases created with different user profiles and descriptions (e.g., John Cena demo input).

Justification:

Using AI models for information extraction reduces the need for manual tagging or form-filling, improving user experience and efficiency.

4. Findings & Analysis

Key Findings:

- AI can effectively parse paragraph input into resume fields like name, email, education, and experience.
- Accuracy improves with refined prompt design.
- Users found the editable form convenient and intuitive.

Discussion:

Performance is influenced by how structured or vague the input paragraph is. Most resumes generated required minimal manual edits.

Comparison with Previous Tools:

Compared to drag-and-drop resume builders, this method offers faster and more intelligent data capture.

5. Conclusion & Recommendations

Summary:

The AI resume generator successfully automates resume creation with a novel user experience. The project integrates LLMs in a full-stack application with strong usability.

Applications:

Can be scaled for job portals, HR tools, or resume review services.

Limitations:

- Not yet cloud-deployed
- AI model may misinterpret ambiguous input
- Only one resume format available

Future Work:

- Add multi-template support
- Cloud deployment (e.g., via Render, Heroku, or AWS)
- Integrate grammar and tone suggestions using AI

6. References

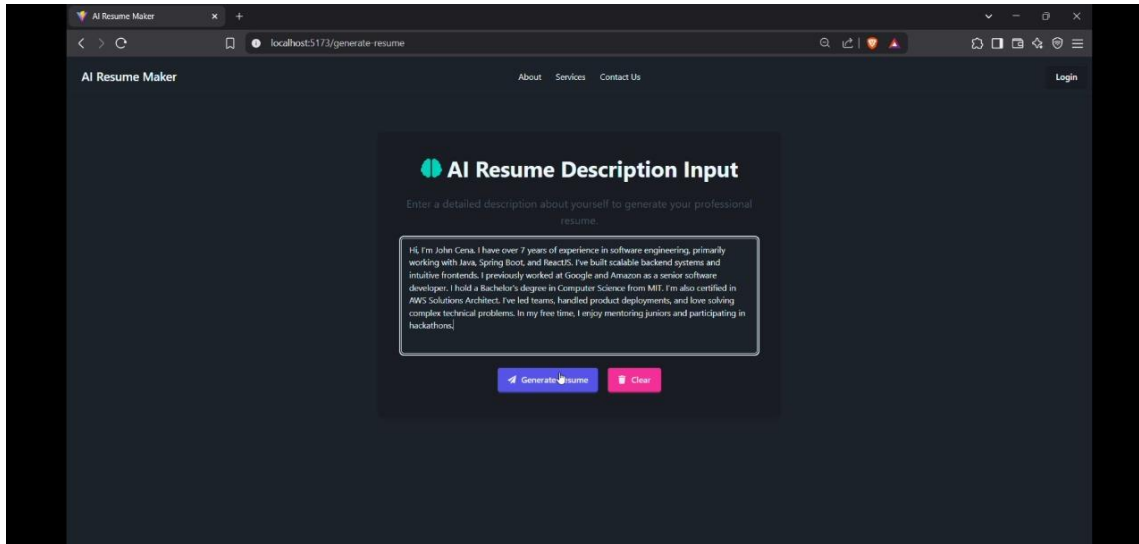
OpenAI. (2023). *ChatGPT Technical Report*. <https://openai.com/research>

DeepSeek. (2024). *DeepSeek Model Overview*. <https://deepseek.com/models>

Spring Team. (n.d.). *Spring Boot Documentation*.
<https://spring.io/projects/spring-boot>

React Docs. (n.d.). *React – A JavaScript library for building user interfaces*.
<https://reactjs.org>

7. Appendices (if any)



The screenshot shows a web browser window with the URL `localhost:5173/generate-resume`. The page title is "AI Resume Maker". The main heading is "AI Resume Description Input". Below the heading is a text input area containing a sample resume description for John Cena. At the bottom of the input area are two buttons: "Generate Resume" (blue) and "Clear" (pink).

AI Resume Maker

About Services Contact Us

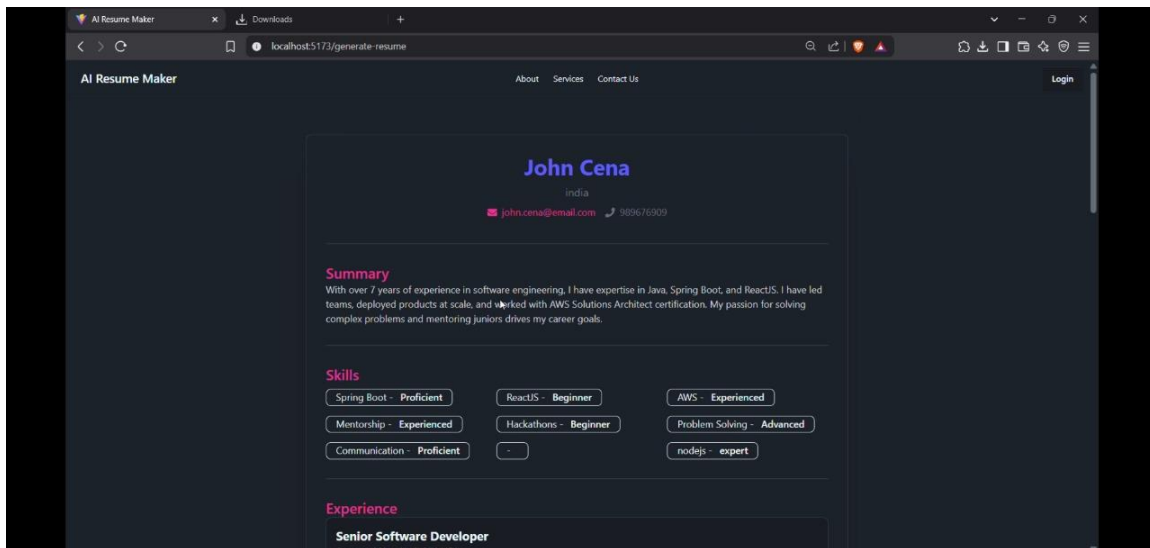
Login

AI Resume Description Input

Enter a detailed description about yourself to generate your professional resume.

Hi, I'm John Cena. I have over 7 years of experience in software engineering, primarily working with Java, Spring Boot, and ReactJS. I've built scalable backend systems and intuitive frontends. I previously worked at Google and Amazon as a senior software developer. I hold a Bachelor's degree in Computer Science from MIT. I'm also certified in AWS Solutions Architect. I've led teams, handled product deployments, and love solving complex technical problems. In my free time, I enjoy mentoring juniors and participating in hackathons.

Generate Resume Clear



The screenshot shows the same web browser window, but now displaying the generated resume for John Cena. The resume includes a summary, skills, and experience section.

AI Resume Maker

About Services Contact Us

Login

John Cena

India

john.cena@email.com 989676909

Summary

With over 7 years of experience in software engineering, I have expertise in Java, Spring Boot, and ReactJS. I have led teams, deployed products at scale, and worked with AWS Solutions Architect certification. My passion for solving complex problems and mentoring juniors drives my career goals.

Skills

Spring Boot - Proficient	ReactJS - Beginner	AWS - Experienced
Mentorship - Experienced	Hackathons - Beginner	Problem Solving - Advanced
Communication - Proficient	-	nodejs - expert

Experience

Senior Software Developer

