**AI-BASED RESUME BUILDER**

**GROUP MEMBER**

**G RUESHIKESH REDDY**

**2320030285**

**Abstract:**

In the contemporary job market, where competition for employment opportunities is fierce, the ability to present a compelling resume is paramount. However, many job seekers struggle to create resumes that effectively showcase their skills, experiences, and achievements. Traditional resume builders, while helpful, often lack the sophistication to customize content for specific job requirements or leverage recent advancements in natural language processing (NLP) and machine learning. To address this gap, we propose an innovative system that utilizes a Sequence-to-Sequence (Seq2Seq) model with an attention mechanism to generate personalized and contextually relevant resumes.

Our system collects and processes data from resumes and job descriptions using a Seq2Seq model. It customizes resumes to match job requirements by analyzing the user's work history, education, and skills. The system includes a user-friendly interface for inputting and customizing resumes and features ATS optimization, error checking, and privacy measures. It aims to provide job seekers with a powerful tool to create compelling resumes that enhance their chances of securing interviews in a competitive job market.

**Tools Used:**

* NLP Library:

Transformers: This library provides pre-trained models and can handle most NLP tasks, including text generation. It includes implementations of Seq2Seq models with attention mechanisms.

* Data Preprocessing and Storage:

Pandas: For data manipulation and resume and job description data preprocessing.

SQLite: A simple and lightweight database for storing user data, job descriptions, and generated resumes.

* User Interface (UI) Development:

HTML/CSS/JavaScript: For the front-end interface.

Flask: A lightweight Python web framework for the back-end, handling user input and serving the generated resumes.

* Version Control:

Git: For version control.

GitHub or GitLab: This is used to host and collaborate on the repository.

* Continuous Integration and Deployment:

GitHub Actions: For automating the build and deployment process.

* Privacy and Security:

Flask-HTTP AUTH: For basic authentication and securing user data.

Environment Variables: For storing sensitive information like API keys and database credentials.