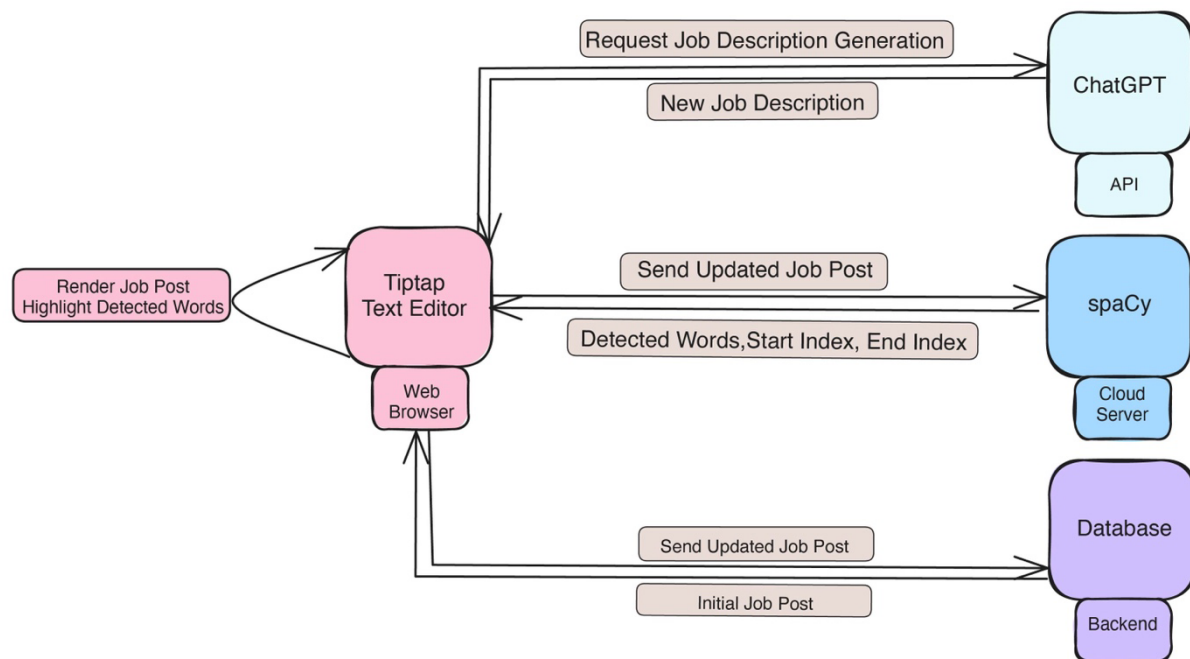


One of the features that Outhire provides to users (talent teams) is a text editor based on Tiptap. Users can paste their existing job descriptions into the text editor, receive a new AI-generated job post tailored to their needs by clicking the “regenerate” button, or edit the job description directly in the text editor.

The architecture of the text editor is illustrated in the following graph.



Architecture Diagram of Text Editor

Initially, the text editor is empty. When a user first enters the route where the text editor exists, the route sends a request for a customised job description to ChatGPT based on the information the user provided, such as the job title, work experience, and salary. Once the text editor receives the newly generated job description, it sends the job description to the database and to a cloud server where an NLP model is deployed, respectively.

When the NLP model receives the job description, it extracts gender-coded words along with their start and end indices in the entire text. The model then sends these words and indices back to the text editor. The text editor utilises the detected words and their indices to render highlights over the words, making them conspicuous to users. This feature aids users in identifying potentially discriminatory words in a job

description, encouraging them to consider changes and promoting gender equality in talent acquisition.

My involvement in the text editor task includes:

1. Training an NLP model, spaCy, to detect gender-coded words in a job description and return every word's start index and end index.
2. Writing code to enable the text editor to highlight every detected word precisely, ensuring the start index of each detected word aligns with the start of the highlight, and the end index aligns with the end of the highlight.

SpaCy is trained in Python, while Outhire is written in Typescript. To address this incompatibility, I created an API for spaCy using Flask to optimise data transfer compatibility between the text editor and spaCy. SpaCy sends the detected gender-coded words and their start and end indices back to the text editor. The text editor then renders the highlights over each detected word. These highlights must precisely overlap with the start index and end index of each word. Additionally, whenever users change the job description in the text editor, the text editor updates the highlights in real-time.

The following two graphs depict job descriptions with gender-coded words highlighted in an unnested bullet list scenario and in a nested bullet list scenario.

**What you'll do**

- Collaborate with cross-functional teams to translate business requirements into visually appealing and functional website designs.
- Design and implement responsive web layouts using Photoshop, Illustrator, and Adobe XD.
- Create and optimize graphics, icons, and images for web use.
- Conduct user research and usability testing to gather feedback and make data-driven design decisions.
- Stay up-to-date with current UI/UX design trends and best practices to continuously improve the user experience.
- Work closely with developers to ensure seamless integration of design and functionality.
- Collaborate with the marketing team to create visually engaging landing pages and promotional materials.

Feminine-coded language 'collaborate' detected

In unnested bullet list scenario

#### What you'll do

- Collaborate with cross-functional teams to translate business requirements into visually appealing and functional website designs.
- Design and implement responsive web layouts using Photoshop, Illustrator, and Adobe XD.
- Create and optimize graphics, icons, and images for web use.
- Conduct user research and usability testing to gather feedback and make data-driven design decisions.
- Stay up-to-date with current UI/UX design trends and best practices to continuously improve the user experience.
- Work closely with developers to ensure seamless integration of design and development.
- Collaborate with the marketing team to create visually engaging landing pages and promotional materials.

#### In nested bullet list scenario

This task presents challenges as the job description is saved in a tree structure, and the indices of text do not match the string version of the job description sent to spaCy. To ensure each highlight aligns with each word, the indices sent back from spaCy must be converted to their value in the text editor. I utilised depth-first search to recursively traverse each text node and used the length of each placeholder to deduce the start index and end index of each detected word in the text editor. Besides obtaining the correct indices in the text editor, the highlights need to be updated in real-time whenever users make changes, regardless of the modifications made or the number of layers in a nested bullet list.