**Technical Document: AI Fire4s CV/Resume Builder**

The above document outlines a technical overall of the AI Fire4s resume builders breaking it’s structure into detail and how it’s created and functions.

**Project Overview**

* We decided to implement HTML coded application of Resume generator with the aim of achieving to aid the end-user by making things simpler for them in creating their resumes, we implemented a simple user interface including basic inputs such as personal details with their background educational information as well as employment experience, the main objective is to achieve certain functionalities in this project to generate the input required information of the user resulting it to have a preview before having it exported as a final product with different file formats such as PDF, HTML and DOCX.
* The generator also has a feature where it contests job description with the focus of climax how well the resume/CV aligns with the specific job description which we used HTML, CSS and JavaScript to develop, using also external information such as **FIGMA**

**Structure**

* The structure of the application is contained in a single index.html file, which forms the basis of both its layout and functionality. Viewport and charset are two crucial meta tags in the  section that provide correct character encoding and responsive performance on all devices. External resources are connected to improve styling and interactivity, and a tag defines the page title. These consist of Google Fonts (especially Inter) for crisp, contemporary font, Tailwind CSS via CDN for utility-first styling, and Font Awesome for icons.

While the navigation bar (nav.main-nav) has a responsive design with links for "Build Resume," "Match Job Description," dropdown menus for "Templates" and "Edit," and a mobile hamburger menu for smaller devices, the header has a h1 element that displays the title of the application. Multiple sections comprise the main content: the "Build Resume" section has a form with text areas and input fields for user data, action buttons for creating previews and exporting the resume in different formats (such as PDF, DOCX, and HTML), and options for clearing fields and switching between dark mode and preview. Users can paste a job description into the "Match Job Description" box, which shows data such as progress bars and matching percentages for different categories (Skills).

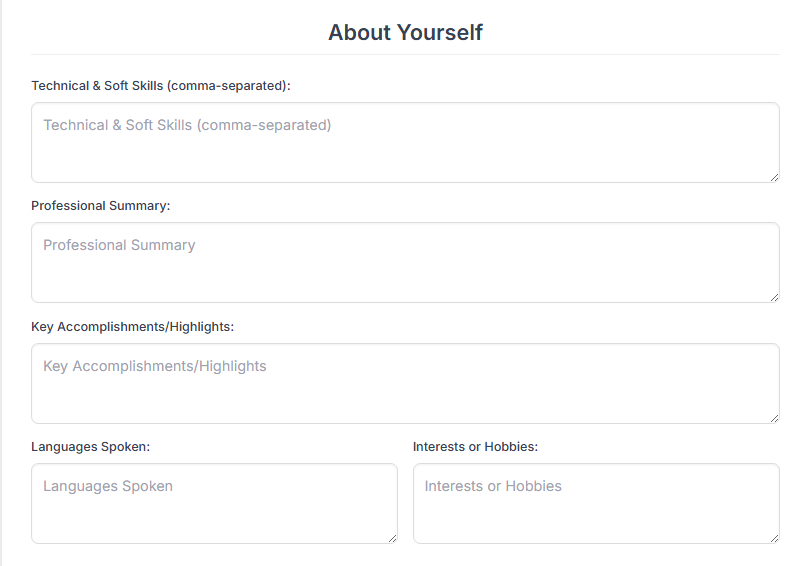
To handle features like section toggling, form handling, preview generation, file exports, and dark mode toggling, the script.js file integrates with external JavaScript libraries, such as html2pdf.bundle.min.js for PDF export and html-docx.js for DOCX export. Utilizing Tailwind CSS's grid layout, conditional visibility, and fluid width utility classes, the application is completely responsive, guaranteeing a smooth experience on a range of devices. Though they are yet hypothetical in nature, AI integration is intended for services like creating a professional summary and matching a user's résumé with job descriptions. Tailwind's responsive design makes sure that layouts and visibility are adjusted for desktop and mobile views using the viewport meta tag and other classes.

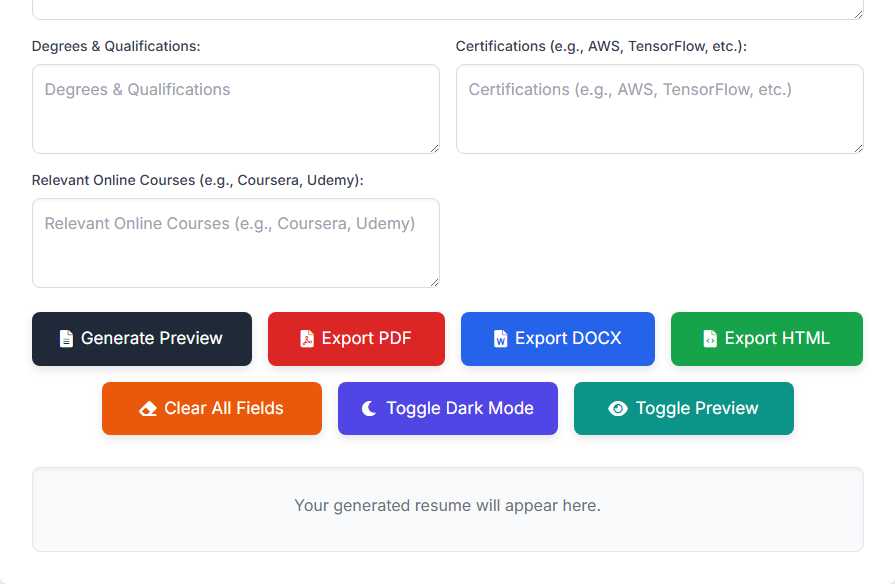
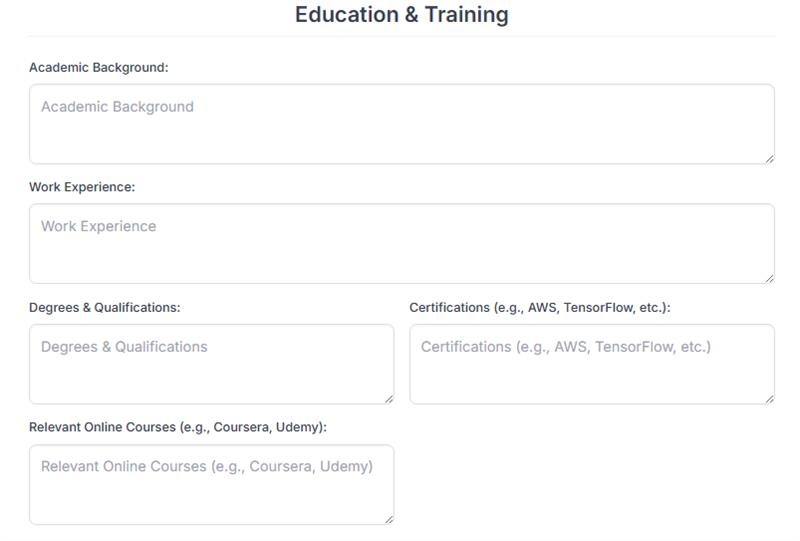
**Layout**

* We employed Figma to develop a comprehensive prototype of the entire layout and user interface, effectively illustrating the potential functionality of the final product and demonstrating its intended usability across various user interactions

Screenshots of clear Resume Generator







Screenshots of Resume Generator with it’s results