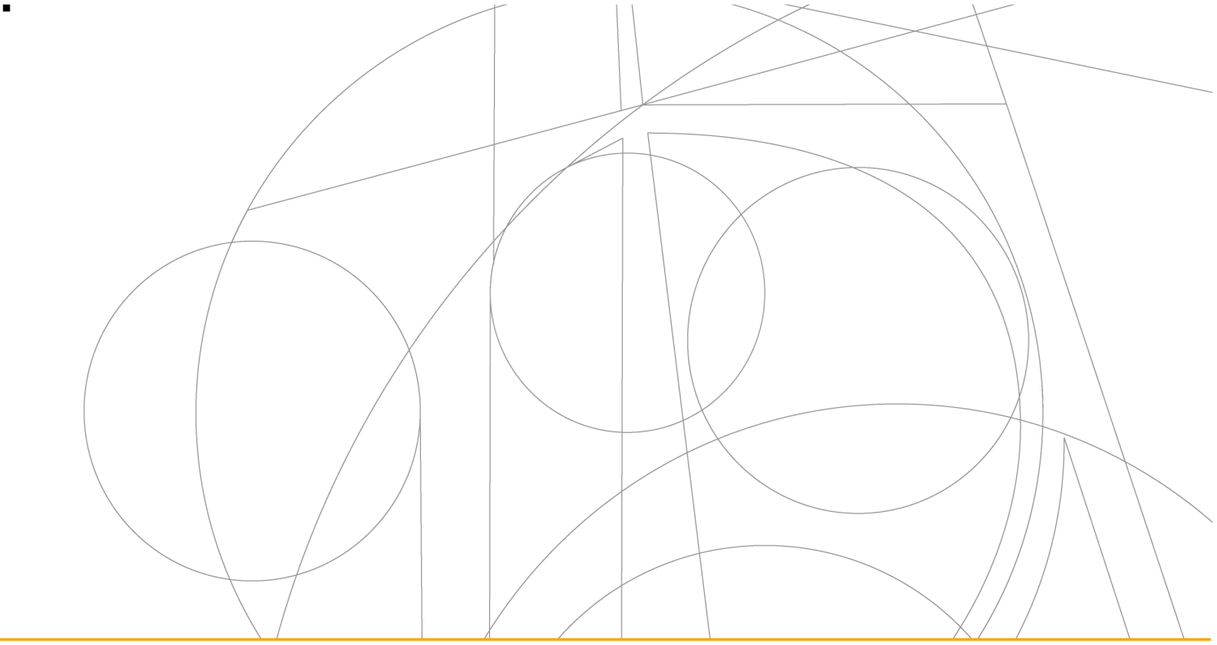


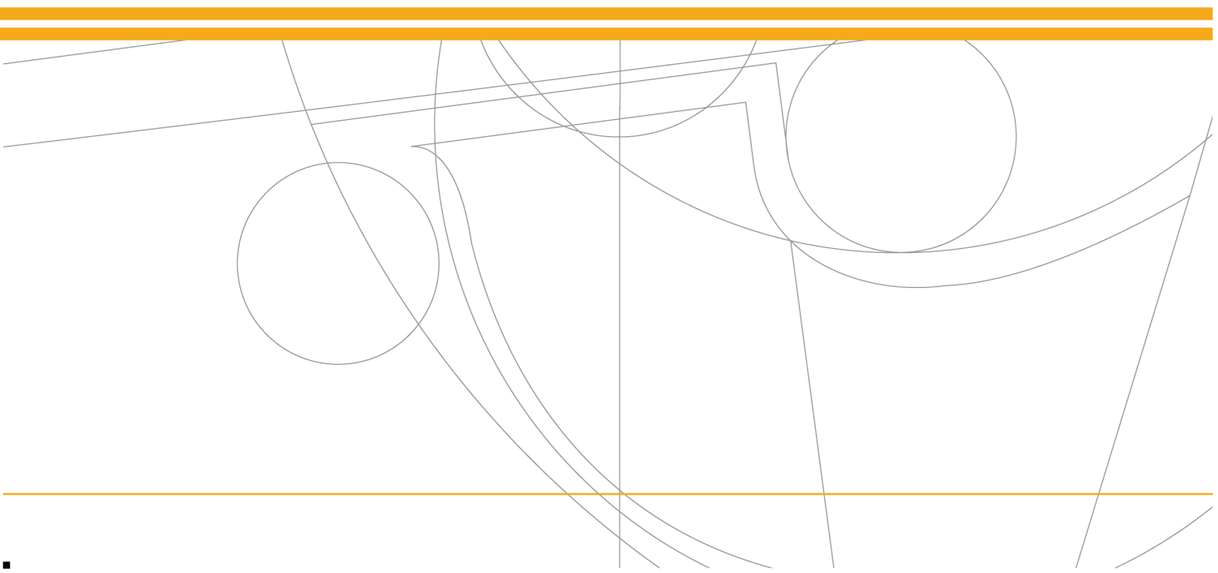
INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

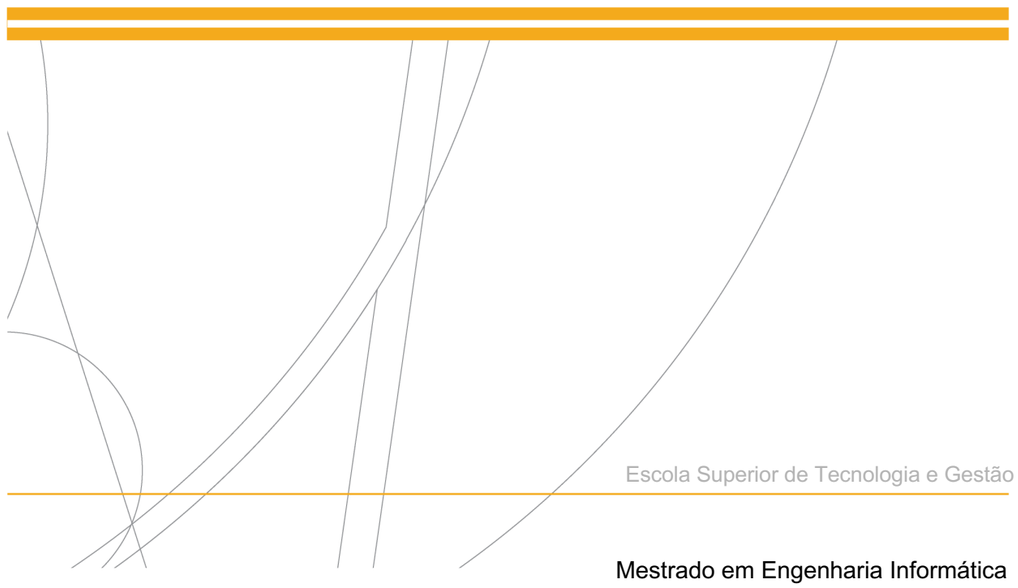
ESTG

Humanizing the inhumane recruitment process using Application Tracking System

Humanizando o processo de recrutamento inhumano baseado em Application Tracking System

Valdo Voliz Mpinga





Humanizing the inhumane recruitment process using Application Tracking System
Valdo Mpinga

2025

Escola Superior de Tecnologia e Gestão

Mestrado em Engenharia Informática

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

Humanizing the inhumane recruitment process using Application Tracking System

Humanizando o processo de recrutamento inhumano baseado em *Application Tracking System*

Valdo Voliz Mpinga

MSC PROJECT WORK

MSC PROJECT WORK
Escola Superior de Tecnologia e Gestão

Orientador: Professor Doutor António Miguel Cruz

July 25, 2025

Resumo

Os Application Tracking Systems (ATS) evoluíram significativamente desde a sua criação em 1996, transitando de simples repositórios de currículos para ferramentas avançadas baseadas em inteligência artificial (IA). Embora estas inovações tenham melhorado a eficiência do recrutamento, introduziram desafios éticos e de direitos humanos significativos. Vieses nos dados de treino da IA e a excessiva dependência da tomada de decisão algorítmica resultaram em violações da dignidade humana, igualdade, privacidade e outros direitos fundamentais. Este estudo analisa estes desafios e propõe uma abordagem centrada no ser humano para os processos de recrutamento, integrando os ATS com salvaguardas éticas.

A metodologia envolve o desenvolvimento de uma Aplicação de Serviço de Humanização para mitigar vieses e reforçar a supervisão humana. As funcionalidades principais incluem:

1. Validação dos requisitos de vagas de emprego;
2. Identificação e correção de triggers de viés nos algoritmos de recrutamento;
3. Exigência de assinaturas digitais de profissionais qualificados para aprovar anúncios de emprego.

Ao incorporar IA generativa e blockchain, este sistema promove conformidade, transparência e justiça no recrutamento, abordando as preocupações éticas da contratação algorítmica. Crucialmente, este trabalho demonstra que as organizações podem alcançar uma humanização significativa dos seus processos de recrutamento, incluindo a mitigação de vieses e o fornecimento de feedback personalizado (entre muitos outros passos potenciais), através de intervenções direcionadas que requerem investimentos financeiros, prazos e recursos computacionais surpreendentemente modestos. Este estudo sublinha a importância de combinar avanços tecnológicos com considerações éticas para criar processos de recrutamento equitativos de uma forma acessível e eficiente em termos de recursos.

Abstract

Application Tracking Systems (ATS) have evolved significantly since their inception in 1996, transitioning from simple resumé repositories to AI-driven tools with advanced capabilities. Although these innovations have improved recruitment efficiency, they have introduced significant ethical and human rights challenges. Bias in AI training data and over reliance on algorithmic decision making have led to violations of human dignity, equality, privacy, and other fundamental rights. This study examines these challenges and proposes a human-centered approach to recruitment processes that integrates ATS with ethical safeguards. This methodology involves the development of a Humanization Service Application to mitigate biases and enhance human oversight. Key features include:

1. validating job vacancy requirements;
2. identifying and addressing bias triggers in recruitment algorithms;
3. requiring digital signatures from qualified professionals to approve job postings;

By incorporating generative AI and blockchain, this system facilitates compliance, transparency, and fairness in recruitment, thereby addressing the ethical concerns of algorithmic hiring. Crucially, this work demonstrates that organizations can achieve meaningful humanization of their recruitment processes, including the mitigation of bias and provision of personalized feedback (among many other potential steps) through targeted interventions that require surprisingly modest financial investment, time commitment, and computational resources. This study underscores the importance of combining technological advancement with ethical considerations to create equitable recruitment processes in an accessible and resource-efficient manner.

Agradecimentos

Dedico o fim desta etapa principalmente aos meus pais, pois sem eles nada disso seria possível, Eles olharam para um filho que toda a adolescência nunca se interessou por estudar e nunca demonstrou hipótese nenhuma de sequer terminar alguma licenciatura, sinceramente quando olho para trás me pergunto como isso lhes fez sentido? Mesmo assim eles apostam em mim neste aspeto e como resultado, concluí uma licenciatura e estou a caminho de terminar um mestrado. Grato a vários amigos, como Mauro Luís, Jackson Barreto, Leonardo Lopez e Pedro Costa que se não fossem eles provavelmente não saberia tudo o que sei sobre programar, não que seja muito ou pouco, muito longe de ser o melhor, porem sei o suficiente para resolver problemas no ramo e o mais importante, fazer grande parte dos softwares que me vêm à cabeça! Muito grato ao Professor Doutor António Cruz, não só por ser o meu orientador deste trabalho do mestrado, mas por ter-me dado a primeira oportunidade de trabalho neste ramo como investigador, pois isso me mostrou que vale a pena continuar a trabalhar duro, pois pelo menos por cá, a meritocracia funciona, pelo menos bem melhor de onde eu venho...

Valdo Voliz Mpinga

"The purpose of computing is insight, not numbers. "

Richard Hamming

Contents

List of Figures

List of Tables

Abbreviations and symbols

AI	Artificial Intelligence
ATS	Application Tracking System
BPMN	Business Process Model and Notation
HRATS	Humanizing Recruitment using Application Tracking System
HR	Human Resource
HRM	Human Resource Management
JSON	JavaScript Object Notation
LLaMA	Large Language Model Architecture
LLM	Large Language Model
XML	Extensible Markup Language

Chapter 1

Introduction

In 1996, the first ATS emerged as a crucial software tool for companies, primarily utilized by management and HR teams to efficiently manage job advertisement applications [11]. At that time, the core functionalities included:

1. Storing resumés
2. Keyword filtering and searches
3. Advertising job vacancies
4. Candidate tracking

Over the years, with the continuous evolution of technology, ATS software has expanded its capabilities significantly. The integration of Artificial Intelligence (AI) into these systems promises increased efficiency and automation of recruitment. However, although these advancements were intended to enhance productivity, they also introduced a range of ethical and operational challenges. AI-driven recruitment tools have been found to exhibit biases related to race, gender, and age, leading to discriminatory hiring practices. Furthermore, increasing reliance on algorithmic decision-making has resulted in a degree of detachment among HR professionals, as recruitment processes have become predominantly automated, reducing the emphasis on human judgment.

These issues extend beyond the technical limitations and enter the realm of ethical concerns. The primary concern is not merely the use of AI-powered ATS tools but rather the manner in which they are designed and implemented. Numerous studies have highlighted how algorithmic decision making in hiring processes can infringe upon fundamental human rights, including the right to work, equality, non-discrimination, privacy, and freedom of expression and association. Unfair algorithmic evaluations [4] can compromise human dignity by making hiring decisions based on flawed or biased data, rather than an applicant's true professional potential.

To address these injustices and human rights violations in the recruitment process, this research aims to develop a service that evaluates and humanizes job vacancies by identifying misalignments in job requirements, unrealistic experience expectations, and excessive workload demands. Additionally, the proposed system is designed to detect biases within recruitment data, including

discrimination based on gender, age, race, and other personal characteristics that may unfairly influence hiring outcomes. By employing AI and data validation techniques, this research seeks to eliminate biased data that do not contribute to an objective assessment of candidates' qualifications and competencies.

Furthermore, the study proposes a balanced approach to mitigating overreliance on algorithmic recruitment by emphasizing the necessity of human oversight in the final selection process. This approach aims to harmonize the efficiency of AI with the indispensable role of human judgment, ensuring that hiring decisions are both fair and free from algorithmic biases. Ultimately, the objective is to create a more inclusive and equitable recruitment framework that upholds ethical principles while leveraging the benefits of technological advancement.

1.1 Problem awareness and identification

First, these problems were identified online. Suddenly, on social media platforms such as Twitter and Reddit, users began to raise concerns regarding various aspects of the recruitment process. Some of the most frequently cited issues include the following.

1. Unrealistic job requirements
2. Unfair screening due to use of ATS
3. Ghost jobs
4. Poor or non-existent feedback

Ghost jobs are job posts for positions that are not actually open or for which the company has no intention of filling. This practice, often used for purposes such as collecting CVs or creating an illusion of growth, can be misleading and frustrating for jobseekers. While this work does not directly address the issue of ghost jobs, it is important to acknowledge their prevalence in the current recruitment landscape.

1.1.1 Unrealistic job requirements

Unfortunately, unrealistic job requirements and discrepancies have become increasingly common, raising significant concerns regarding fairness and human rights violations [4]. An illustrative example is presented in Figure 1.1. In this case, a developer who had created a software framework discovered a job listing that required more years of experience with the framework than the existing tool. Even the creator of the framework did not meet the specified criteria. Such issues often arise when HR departments rely on ATS-generated role requirements without involving domain experts who fully understand actual job demands.

Figure 1.1: Unrealistic requirements.

In Figure 1.2, a misconfigured job requirement led to a major HRM team dismissal. In this case, the role remained open for three months without any successful candidates. When the team manager confronted HR about the issue, they simply stated that no qualified applicants were found. Suspicious of this outcome, the manager created a new email account, modified his own CV, and applied it for the position, only to be automatically rejected by the ATS. This incident is particularly revealing, as the manager himself knew precisely what qualifications were required. Upon further investigation, the issue was traced to a typo in job requirements, which prevented suitable candidates from passing the ATS filtering process.

A similar issue is demonstrated in Figure 1.3, where an applicant is rejected exactly one minute after submitting its application. This highlights how candidate data are processed almost exclusively by the ATS before reaching the HR personnel. While automation can enhance efficiency, it also reinforces the perception that applicants are treated purely as data points rather than as individuals. The core issue is not the use of ATS or AI-driven recruitment systems themselves, but rather the complacency and lack of human intervention in decision making. Companies must strike a balance between automation and ethical hiring practices to ensure that recruitment processes remain fair and humane.

1.2 Research Objectives

Contemporary recruitment methodologies often grapple with issues pertaining to transparency, equity, and efficient allocation of resources. This thesis posits that readily available open-source

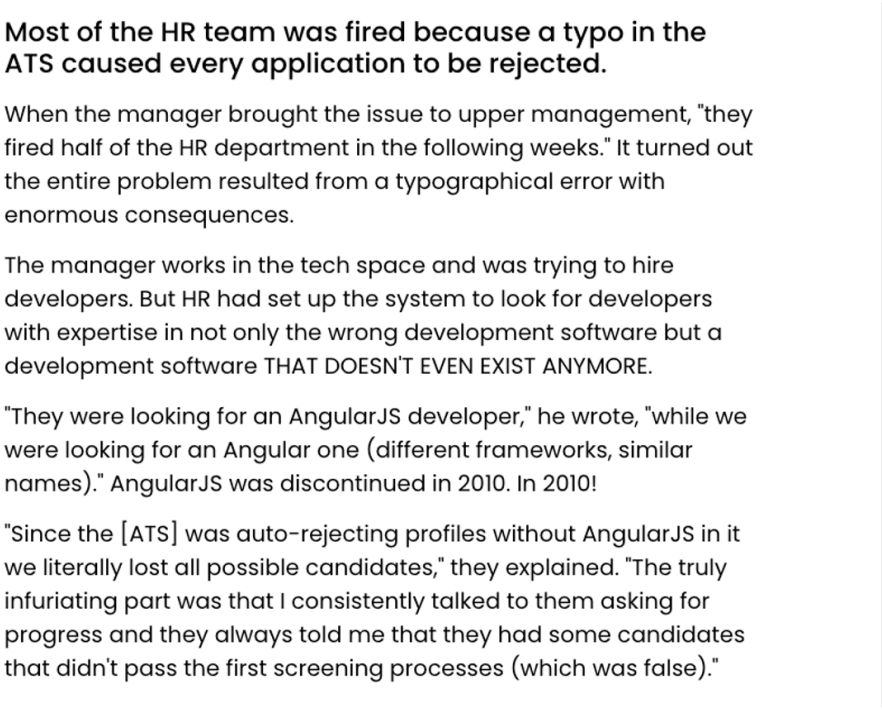


Figure 1.2: Typo on requirement.

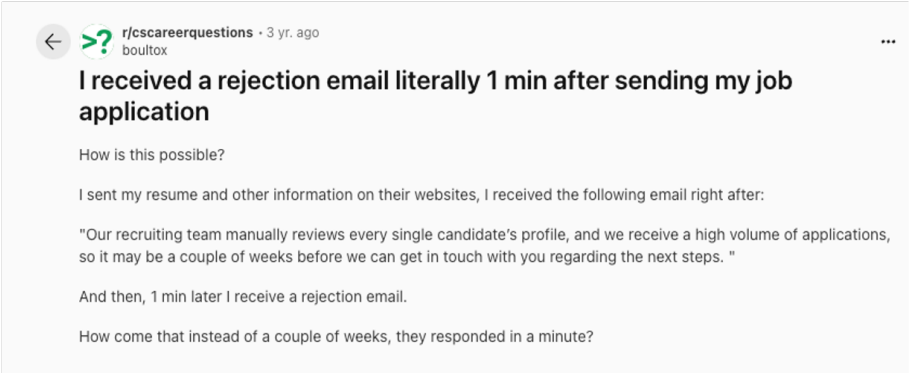


Figure 1.3: Instant rejection.

technologies, specifically Large Language Models (LLMs) and blockchains, offer a viable pathway to address these challenges. The overarching goal of this research is to demonstrate the feasibility of implementing recruitment processes that are not only more transparent, equitable, and humane, but also achievable with minimal investment in both temporal and financial capital by effectively harnessing these existing technological frameworks. The specific research objectives undertaken to substantiate this central premise are as follows.

1. **Vacancy Requirement Validation:** Contemporary recruitment practices frequently exhibit incongruities between job titles and associated requirements. This research implemented the following AI-driven validation protocols:

    - Analyze coherence between job titles and qualification requirements

- Assess temporal consistency of experience demands relative to market realities
- Identify contradictory or exploitative requirement patterns
- Generate standardized compliance metrics for published vacancies

2. **Bias Trigger Identification:** To address systemic discrimination in hiring practices [13], the proposed system will:
    - Implement anonymization pipelines for protected attributes (age, gender, sexual orientation, race/ethnicity, etc.)
3. **Digital Signature of Relevant Human Actors -** Building upon distributed accountability systems, we propose a blockchain-based validation framework to address unrealistic job requirements through:
    - Technical validation by domain experts (e.g., project managers) via smart contracts
    - Immutable audit trails for requirement modifications

Complete technical implementation which resulted in a web-based platform featuring the following features:

1. AI-powered feature for requirement analysis
2. Bias mitigation feature
3. Publicly accessible blockchain signature validation code

1.3 Document Structure

This document is structured into 6 main chapters, each addressing a critical aspect of the research and development of the recruitment humanization process.

Chapter 1: Introduction

This chapter outlines the motivation behind the research, problem statement, and objectives of the study. It also highlights the contributions of this research to the domain of the recruitment process and provides an overview of the structure of the document.

Chapter 2: Theoretical background

Chapter 2 delineates the foundational concepts that are essential for comprehending the role of AI within the domain of recruitment. Specifically, this chapter encompasses the following aspects.

- **Algorithmic Hiring and Ethical Considerations:** An introduction to the integration of AI in recruitment processes, addressing critical aspects such as algorithmic bias, fairness metrics, safeguarding human dignity and rights, and the imperative for regulatory frameworks.
- **Technological Advancements for Trust and Accuracy:** An overview of innovative technological solutions aimed at enhancing transparency and precision in AI-driven recruitment, including blockchain integration and sophisticated resumé parsing methodologies.
- **Perceptions and Acceptance of AI in HRM:** A discussion concerning employee attitudes towards the adoption of AI in human resource management, alongside an examination of organizational factors that influence its successful integration.

This chapter establishes the requisite theoretical context for subsequent expositions within this document.

Chapter 3: Methodology

Chapter 3 outlines the Design Science Research (DSR) [9] methodology employed to develop and evaluate humanization service applications for fairer and more transparent ATS-driven recruitment. This section details the application of the six DSR activities [5] in this study.

- **Problem Identification and Motivation:** Addresses the issues of bias and dehumanization in current ATS processes, emphasizing the need for fairness and transparency in AI-driven hiring.
- **Objectives Definition:** Defines the research objectives, focusing on designing and developing a humanization service application to validate vacancy requirements, remove bias triggers from applicant data, and implement a digital signature mechanism for human reviewers.
- **Design and Development:** Describes the creation of the humanization service application modules, including the Vacancy Requirement Validation Module, the Bias Triggers Removal Module (both using Python, FastAPI, and LLaMA), and the Digital Signature Module (using Blockchain).
- **Demonstration:** Showcases the application's functionality through validating job postings, processing applicant data for bias removal, simulating the blockchain validation process, and a web interface (`https://joblimpo.valdompinga.com/`).
- **Evaluation:** Assesses the effectiveness of the application by analyzing vacancy validation results on a large dataset, verifying bias removal, and testing the blockchain protocol.
- **Communication:** Explains how the research findings are communicated to both technology- and management-oriented audiences, highlighting the application's technical aspects and its benefits for fairness, transparency, and risk mitigation.

Chapter 4: Design and Implementation

Chapter 4 details the design and implementation of the Humanization Service Application. Key aspects of architecture, technologies, and implementation include the following.

- **Architecture:** The system employs a modular **ATS Humanization Service** with a central **Humanization Service** component and a **Data Processing** module potentially utilizing an LLM. The frontend is a web application interacting with a FastAPI backend via a Nginx reverse proxy with containerization using Docker and Docker Compose.
- **Technologies:** The application leverages a LLaMA LLM for text analysis, Python and the FastAPI framework for the backend API (`/requirements` and `/candidate` endpoints), JavaScript and SvelteKit for the interactive frontend (`https://joblimpo.valdompinga.com/`), Docker and Docker Compose for deployment, Nginx as a reverse proxy, and Ethereum smart contracts (Solidity, tested with Hardhat, repository at `https://github.com/ValdoMpinga/clean-job`) for the decentralized validation protocol.
- **Implementation Points:**
  - **Vacancy Requirement Validation:** Uses a LLaMA prompt for multi-faceted analysis, outputting structured JSON.
  - **Bias Triggers Removal:** Uses a LLaMA prompt to identify and remove bias-related data while preserving the original format.
  - **Decentralized Validation Protocol:** Implements dual cryptographic authorization via Ethereum smart contracts, ensuring on-chain verification and immutable records for job posting approvals.

Chapter 5: Validation and Results

Section 5 presents the validation outcomes of the application of Humanization Services.

5.1 Vacancy Validation:

Analysis of 21,701 job postings revealed substantial issues, with approximately 64.76% of requirements misaligned with the role, around 35.99% exhibiting unreasonable experience demands, roughly 66.56% presenting unrealistic workload expectations, and about 38.11% containing internal discrepancies.

5.2. Evaluation of different Large Language Models by number of parameters using Job Requirement Analysis

An assessment of various open-source LLMs, ranging from 0.5B to 8 B parameters, revealed a trade-off between model size and performance in job requirement analysis. Smaller models exhibited faster processing times, but struggled with structured output generation, often resulting in parsing errors. Larger models, particularly Mistral-7B, demonstrated higher accuracy and consistency in identifying issues, such as role misalignment, irrational experience demands, unfeasible workloads, and internal discrepancies, albeit with longer processing times.

5.3 Functional Verification of Bias Trigger Identification:

The bias trigger identification service endpoint was functionally tested using mock candidate data in JSON format, simulating typical ATS workflows. Input data included various demographic attributes. The tests successfully confirmed that the service accurately processed the input and filtered out potentially biased demographic details (e.g., age, gender, race, religion), retaining only professionally relevant information, such as languages spoken and address, as demonstrated by the provided example input and expected output.

5.4 Blockchain Validation:

Comprehensive testing of the blockchain validation system using the Hardhat testing environment confirmed robust security and operational integrity. Access control mechanisms strictly enforced HR manager exclusivity and validated field-expert assignments by job category. Rigorous signature verification testing achieved a 100% detection of invalid signatures and prevented unauthorized approvals and duplicate submissions. End-to-end workflow integrity testing confirmed mandatory expert assignment, consistent state management across CRUD operations, and an immutable approval history. Overall, the testing validated the system's robust role-based access control, rigorous cryptographic requirements, and full adherence to the functional specifications.

5.5 System Demonstration via Public Web Interface:

This section details the public web interface developed to showcase the system's core functionalities. It highlights two key interactive demonstrations: the **Interactive Job Requirement Analysis** tool, accessible at `https://joblimpo.valdompinga.com/requirements`, which provides real-time evaluation of job postings for human-centered criteria using natural language processing; and the **Bias-Free Candidate Presentation Interface**, available at `https://joblimpo.valdompinga.com/candidate`, which demonstrates the anonymization of candidate data in JSON, XML, and plain text formats by removing sensitive demographic information to mitigate potential biases.

5.6 Blockchain Implementation Showcase:

This section presents the blockchain implementation of the system, with access details provided at https://joblimpo.valdompinga.com/validation. A direct link is included in the project's open-source GitHub repository, which hosts the complete Solidity smart contract code, a comprehensive Hardhat test suite, and deployment scripts for the Ethereum network, offering full transparency and verifiability of blockchain validation logic. This section demonstrates the functionality and effectiveness of the proposed solution.

Chapter 6: Conclusions

Chapter 6 summarizes the key findings and contributions of this research on humanizing recruitment processes using AI and related technologies. This study addressed the challenges of bias and dehumanization in ATS-driven recruitment through the design, implementation, and validation of a humanization service application.

Chapter 2

Theoretical background

Problems with recruitment that rely on AI and ATS have garnered increasing attention from researchers. This has led to several scholarly articles discussing the potential violations of fairness and human rights that can arise from the use of these technologies. While these AI-powered tools and algorithms were initially developed to enhance the recruitment process for both recruiters and applicants, the core issue often lies in how these systems are implemented and the inherent biases present in the training data used to develop the AI models. These biases can inadvertently perpetuate and amplify existing societal inequalities in recruitment pipelines. To address these challenges, emerging solutions incorporate advanced technologies, including Large Language Models for bias detection and validation, as well as blockchain technology to ensure transparency, accountability, and immutable audit trails in the hiring process. Blockchain's decentralized validation protocols can enforce human oversight by requiring cryptographic authorization from qualified professionals before job postings are approved, thereby maintaining the essential human element in recruitment decisions. This chapter delves into the multifaceted landscape of AI in recruitment and explores the challenges, potential solutions, and perceptions surrounding its use.

2.1 Ethical, Fairness, and Human Rights Implications of AI Recruitment

A central concern regarding the increasing adoption of AI in recruitment processes is the significant potential for ethical breaches and violation of fairness and fundamental human rights [3, 8, 1, 13, 4]. Inherent biases within AI algorithms can lead to discriminatory outcomes against candidates based on sensitive attributes such as age, disability, gender, religion, race, ethnicity, and sexual orientation [4]. These biases can originate from skewed training data, disparities in self-promotion across demographic groups, or discriminatory patterns embedded in the algorithmic design itself.

These biases and the reductionist nature of certain AI hiring assessments pose significant threats to human dignity [1]. By focusing on singular, quantifiable metrics, these assessments can severely limit a candidate's ability to authentically represent their multifaceted self, thereby

infringing upon their inherent dignity and autonomy over self-representation [1]. This can also lead to broader infringements on fundamental human rights, including the right to work, equality and nondiscrimination, privacy, free expression, and association [13]. The absence of human intervention in purely algorithmic assessments, while aiming to reduce human prejudices, introduces new challenges related to systemic biases embedded in data and algorithms [8].

Addressing these deeply ingrained biases necessitates comprehensive and rigorous application of various fairness measures. This includes a thorough evaluation of both group fairness and the granularity of fairness analysis, as well as prioritizing the interpretability of AI decision-making processes [4].

A complex ethical challenge involves navigating the intricate issue of historical disadvantages faced by certain communities [4]. The debate remains open on whether algorithms should proactively favor candidates from historically disadvantaged groups to redress past inequities or strictly adhere to present-day equal opportunity principles. This complex ethical dilemma necessitates continuous research and societal dialogue to establish clear ethical guidelines to ensure fairness in AI-driven recruitment.

2.2 Strategies for Risk Mitigation and Trust Enhancement

Several strategies have been proposed to mitigate the risks associated with AI use during recruitment and enhance trust in these systems. The implementation of human rights impact assessments (HRIAs) is crucial for proactively identifying and addressing potential human rights violations arising from AI deployment [13]. Additionally, Multi-Agent Systems designed for ethical and legal auditing, particularly in less regulated areas, such as AI-driven video interviews, can help prevent discriminatory practices [3].

Research has also emphasized the importance of balancing transactional efficiency with relational considerations in AI-driven recruitment [8]. Overrelying solely on algorithmic assessments risks violating human dignity by neglecting the holistic view of candidates and the significance of human interaction in the recruitment process [8].

2.3 Technological Advancements for Transparency and Accuracy

Technological solutions have been explored to enhance the transparency and accuracy of AI recruitment. The integration of AI with Blockchain Technology (BCT) presents an avenue for increased transparency and trust. For instance, the proposed TAIRA-BSC architecture leverages blockchain's verifiable and secure features for candidate data and job matching with the aim of improving data management, security, and traceability [2]. This platform uses AI and blockchain to establish a trustworthy job matching system where job seekers can create authenticated CVs, recruiters can post job descriptions, and smart contracts can generate pre-selection lists [2].

Improving the accuracy of the input data is also crucial. Hybrid models, such as those utilizing YOLOv5 for resumé section detection and DistilBERT for information extraction, have shown a

high resumé parsing accuracy (96.2%), thus enhancing the reliability of candidate evaluations by minimizing data extraction errors [6]. This approach refines the Applicant Tracking System (ATS) process, offering a more precise method for resume evaluation by accurately extracting and classifying relevant information such as contact details, work experience, and skills with a high degree of accuracy (96.3%) [6].

2.4 Employee Perceptions and Organizational Implementation

Studies generally indicate positive employee intentions towards AI adoption in recruitment and selection, driven by perceived benefits, such as time and cost savings, speed, and accessibility [7]. However, successful AI integration in HRM hinges on organizational practices and fostering trust [10] as trust mediates the positive impact of user perception on organizational commitment, subsequently enhancing engagement and performance.

Research by Năstase et al. [7] using the Technology Acceptance Model (TAM) in Romanian companies, revealed that perceived ease of use positively influences AI adoption intent in recruitment and selection. Specific benefits, such as time and cost reduction, speed, accessibility, equal opportunities, work diversity, competency focus, and CV updating, drive recruitment AI acceptance. In selection, cost-effectiveness, privacy, feedback, reduced favoritism, efficient screening, skill matching, video interviewing, quick evaluation, and data extraction are influential. Interestingly, non-discrimination in recruitment and security in selection showed lower perceived influence.

Prasad et al. [10] found that trust significantly mediates the relationship between positive user perception of generative AI and organizational commitment in the IT sector, leading to higher employee engagement and performance. Optimism and innovativeness shape user perception, which in turn boosts trust and commitment. Organizational commitment positively affects engagement, which in turn enhances performance.

In conclusion, while perceived efficiency drives positive employee attitudes towards AI in HRM, successful implementation requires careful consideration of the organizational context and cultivation of trust. Addressing employee perceptions and fostering a trusting environment are crucial for effectively leveraging AI's potential of AI in human resource management.

2.5 Technologies

The implementation of the humanization service application required a carefully selected technology stack to address the complex challenges of bias detection, job requirement validation, and human oversight in recruitment processes. The technology choices were guided by principles of scalability, maintainability, and accessibility, ensuring effective deployment across organizations of varying technical capabilities. This section outlines the key technologies employed, from Large Language Models for intelligent analysis to modern web frameworks for user interfaces, containerization for deployment, and blockchain for decentralized validation.

2.5.1 LLM

The main features implemented to enhance humanization in this system are the validation of vacancy requirements before publication, and the identification of bias triggers. To address these challenges efficiently, artificial intelligence was integrated into the process, specifically using an LLM called LLaMA. Developed by Meta-AI [12], LLaMA is a state-of-the-art natural language processing (NLP) model designed to advance research in generative AI. It is a transformer-based model trained on an extensive and diverse corpus of text, offering high performance with relatively fewer parameters than other large-scale models such as GPT-3, while achieving competitive results across various NLP benchmarks [12].

Developers interact with LLM's primarily through prompts, which are textual inputs or instructions that guide the model's response. These prompts can vary in complexity from simple queries to detailed commands, and the level of specificity directly affects the quality of the output. When processing a prompt, the input is tokenized, a process in which the text is broken down into smaller units or tokens, enabling the model to interpret and generate a response accurately.

The rise of open-source LLM's has opened up numerous opportunities as they allow developers to accelerate workflows. Tasks that require coding small, repetitive features can now be simplified with the right prompt, which can produce a functional output efficiently. However, a significant limitation of LLM's is their consistency. Owing to their probabilistic nature, these models generate responses word-by-word based on the likelihood of each word following the previous one [12]. Consequently, the output can vary even when the same prompt is used multiple times.

Fortunately, this inconsistency can often be mitigated in scenarios that require predictable behavior. By crafting well-structured prompts that include specific rules or formats, a model is more likely to produce consistent outputs. This makes LLMs highly useful for smaller, well-defined tasks and systems, where consistency is essential. When the prompt sets clear expectations, the model can reliably generate responses in the desired format, thus making it a practical tool for many applications.

2.5.2 Python and FastAPI

Python, a high-level interpreted programming language, served as the foundational language for developing the backend infrastructure of the proposed system. Its versatility and rapid development capabilities make it an ideal choice for constructing the core logic of both vacancy requirement validation and bias-related data-removal services. Specifically, the FastAPI framework, a modern, high-performance web framework for building APIs with Python based on standard Python type hints, was employed to create robust and efficient endpoints for the "ATS Humanization Service." FastAPI's inherent support for data validation, serialization, and asynchronous operations has facilitated the development of a scalable and maintainable backend architecture capable of handling diverse data inputs and processing demands from various Applicant Tracking Systems. The utilization of Python's rich set of libraries for natural language processing and data manipulation further enhances the analytical capabilities of the system.

2.5.2.1 Insomnia REST Client for API Interaction and Testing

To facilitate the development, testing, and demonstration of the FastAPI backend API endpoints, the Insomnia REST Client was utilized. Insomnia provides a user-friendly interface for constructing and sending HTTP requests to the API, allowing developers to inspect responses and validate the functionality of each endpoint. This tool proved invaluable for iteratively developing and debugging the `/requirements` endpoint for vacancy validation, and the `/candidate` endpoint for bias trigger removal. Insomnia's features, such as environmental variables and request chaining, streamlined the testing process across different configurations and data scenarios, ensuring the reliability and correctness of backend services. The ability to organize API requests into collections also aids in documenting and sharing API usage patterns.

2.5.3 Javascript and SvelteKit

The front-end interface for the validation workflow was constructed using JavaScript, a ubiquitous scripting language for web development, in conjunction with the SvelteKit framework. SvelteKit, a meta-framework built on top of the Svelte compiler, enables the creation of performant and user-friendly web applications by shifting much of the compilation work to the building step. This results in smaller bundle sizes and faster client-side rendering, providing a responsive experience for project managers or other designated experts tasked with reviewing and approving the vacancy requirements. JavaScript's inherent interactivity and SvelteKit's component-based architecture facilitated the development of an intuitive graphical user interface for visualizing and validating job specifications, ensuring ease of use for non-technical stakeholders involved in the approval process.

2.5.4 Docker and Docker Compose

To ensure consistent deployment across different environments and to simplify the setup process, Docker and Docker Compose were employed. **Docker** was used to containerize both the Python/FastAPI backend and the JavaScript/SvelteKit frontend into isolated and reproducible units. Each container encapsulates all necessary dependencies, libraries, and configurations required for the respective application to run. This eliminates environment-specific issues and ensures that the applications behave identically regardless of the underlying infrastructure. **Docker Compose** was utilized to define and manage multi-container Docker applications. A `docker-compose` is a. yml file that specifies the services (backend, frontend, and potentially other dependencies), their configurations, networks, and volumes, allowing for easy orchestration and deployment of the entire system with a single command.

2.5.5 Nginx

Nginx was implemented as a reverse proxy server to sit in front of the application containers. This provides several key benefits, including enhanced security, improved performance through load

balancing and caching, and simplified management of the SSL/TLS certificates. Nginx acts as the entry point for all external requests, routing them to the appropriate backend or front-end container based on the defined rules. For the front-end SvelteKit application, Nginx can serve static assets directly, further improving performance. For the FastAPI backend, Nginx can handle tasks such as request routing, rate-limiting, and protection against common web vulnerabilities.

2.5.6 Git and Github

Version control and collaborative development for this project were managed through Git, a distributed version control system, and GitHub, a web-based platform that provides hosts for Git repositories. Git's robust branching and merging capabilities enabled parallel development of different modules and features, while ensuring a clear history of changes and facilitating seamless integration of contributions from multiple developers. GitHub serves as the central repository for the project's codebase, providing tools for issue tracking, code review, and collaborative project management. Furthermore, the open-source nature of the envisioned blockchain application necessitates a publicly accessible and version-controlled platform for community contributions and transparency, making it an indispensable tool for hosting and managing project evolution.

Chapter 3

Methodology

The methodology applied to humanizing recruitment processes that use ATS, is Design Science Research (DSR) [9]. DSR includes the identification of real-world problems, design and development of innovative artifacts, and evaluation of their effectiveness through rigorous testing and refinement. This function ensures that technological progress aligns with moral principles, regulatory compliance, and practical practices. In this study, DSR guides the construction and implementation of Humanization Services applications and integrates human inspection mechanisms and prejudice strategies to improve fairness and transparency in recruitment processes.

To provide a structured approach to this study, we followed six activities [5] of the DSR process (see figure 3.1), adapted from [5] for this work's specific context:

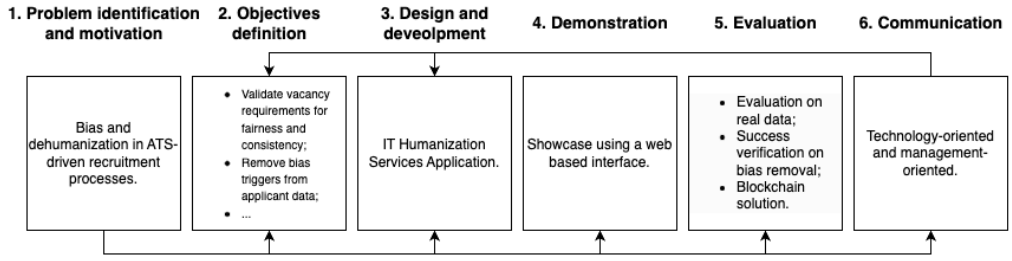


Figure 3.1: DSR diagram applied to the project's context.

3.1 Problem Identification and Motivation

This study addresses the critical problem of bias and dehumanization within current ATS-driven recruitment processes. As detailed in Chapter 1, motivation stems from the need to ensure that fairness, transparency, and ethical considerations are integrated into AI-driven hiring practices. The specific problems identified include the following.

- The potential for algorithmic bias in screening candidates, leading to unfair disadvantages for certain groups (see Chapter 2).

- The lack of transparency in how AI evaluates candidates, hindering trust in the hiring process.
- The need to maintain human dignity and ensure ethical considerations are paramount when integrating AI into recruitment.

The value of addressing these problems is significant as it promotes more equitable hiring practices, enhances organizational reputation, and aligns technological advancements with societal values. This aligns with the DSR principle of developing technology-based solutions for relevant business problems.

3.2 Objectives definition

The objectives of this research are to design and develop a humanization service application that

- Validates vacancy requirements to ensure fairness and consistency in job postings. This involves analyzing job descriptions to identify and rectify misalignments, unreasonable demands, unrealistic expectations, and internal discrepancies. (See Chapter 4 and 5)
- Removes bias triggers from applicant data to mitigate the risk of algorithmic bias. This focuses on identifying and removing sensitive information that could lead to unfair evaluations, while preserving the integrity of the original data. (See Chapter 4 and 5)
- Implements a digital signature mechanism for human reviewers to enhance accountability and transparency. This involves a decentralized validation protocol using blockchain to ensure that both HR personnel and subject-matter experts authorize vacant publications. (See Chapter 4 and 5)

These objectives are qualitative, focusing on how the application of Humanization Services improves the fairness and transparency of recruitment processes. They are designed to create disruptions that trigger strategic responses from organizations.

3.3 Design and Development

This activity involved the design and construction of humanization service applications. Artifacts encompass several key components:

- **Vacancy Requirement Validation Module:** This module was developed using Python and the Fast API, incorporating LLaMA prompts to evaluate job postings. It analyzes job descriptions and provides structured feedback regarding fairness and consistency.
- **Bias Triggers Removal Module:** This module, also implemented with Python, the Fast API and LLaMA prompts, identifies and removes bias-related fields from applicant data. This design ensures that the removal process maintains the original data format.

- **Digital Signature of Relevant Human Actors Module:** A decentralized validation protocol was designed, leveraging Blockchain technology, to mandate dual cryptographic authorization from HR personnel and subject-matter experts.

These artifacts are objects for which development research contributes, including models, methods, and new technical properties.

3.4 Demonstration

The demonstration of the humanization service application involved showcasing its functionality in a real-world or simulated environment. The key demonstrations include the following:

- Validating a set of job postings to illustrate the application's ability to identify inconsistencies and biases.
- Processing applicant data to demonstrate the removal of bias triggers while preserving data integrity.
- Simulating the blockchain-based validation process to confirm the correct implementation of access control and signature verification.
- A web interface `https://joblimpo.valdompinga.com/` was developed to showcase job analysis, bias-free candidate view, and blockchain integration.

These demonstrations illustrate the use of artifacts to solve instances of the identified problems.

3.5 Evaluation

The evaluation phase assessed the effectiveness of the application of Humanization Services in achieving defined objectives. The evaluations included the following:

- Analyzing the results of vacancy validation on a dataset of 21,701 job postings, measuring the percentage of postings with misalignments, unreasonable demands, unrealistic expectations and internal discrepancies.
- Verifying the successful removal of bias-related fields from applicant data.
- Testing the blockchain validation protocol to ensure correct access control, signature verification, and workflow integrity.

The evaluation involved comparing the results with the objectives defined in Step 2 and employing relevant analysis techniques and metrics.

3.6 Communication

The findings of this research are communicated in this document, highlighting the relevance of the problem, the design and utility of the Humanization Services application, and the results of the evaluation. Communication is tailored to both technology- and management-oriented audiences.

- For technology-oriented audiences (e.g., AI developers, software engineers), detailed information is provided on the application's architecture, implementation using Python, Fast API and LLaMA, and the blockchain protocol.
- For management-oriented audiences (e.g., HR managers, organizational leaders), the focus is on the application's ability to enhance fairness and transparency, mitigate legal risks, and improve organizational reputation.

Chapter 4

Design and Implementation

4.1 Introduction

The core objective of this study was to enhance human involvement in key recruitment processes. This is achieved by pre-processing job requirements before they are published and by addressing biased applicant data before they are analyzed by the ATS. To this end, a Humanization Service Application was developed that incorporates the necessary features to implement these enhancements. In Figure 4.1, the architecture is detailed, and the requirements implemented at the top are

1. **Validation of vacancy requirements to be published -** Contemporary analysis of the employment landscape reveals a prevalent issue: the dissemination of job vacancies characterized by incongruous and often unattainable prerequisites. These discrepancies range from entry-level positions, stipulating multi-year experience levels, to roles demanding expertise exceeding the temporal existence of the relevant industry or technology. To address these systemic inconsistencies, an intelligent automation framework was developed for rigorous evaluation of job vacancy postings. This framework undertakes a multifaceted assessment to identify potential contradictions between designated role titles and articulated requirements. Beyond the detection of unrealistic experience demands, the system was engineered to scrutinize job descriptions for a broader spectrum of potential issues. This includes the identification of misaligned skill sets, evaluation of workload feasibility within the scope of a single position, and detection of any internal inconsistencies within the vacancy description itself. Furthermore, the analytical capabilities extend to the formulation of actionable recommendations for rectifying identified issues, such as suggested adjustments to the role title, modifications to specific requirements, and revisions to experience-level expectations. Critically, the framework incorporates a module dedicated to the identification of potential violations of ethical and human-centered employment practices, ensuring a more equitable and transparent recruitment process. The output of this automated validation process encompasses a comprehensive evaluation, including a detailed breakdown of identified discrepancies, a set of targeted recommendations for improvement, and an overall assessment

of the vacancy's compliance with the established criteria.

2. **Mitigation of Bias Triggers -** A significant concern in contemporary hiring practices pertains to the potential for discriminatory biases arising from the collection and utilization of sensitive personal information. Attributes such as age, gender identity, sexual orientation, and racial or ethnic background have historically served as triggers for prejudiced decision-making. To counteract these inequitable scenarios, a methodology focused on the identification and subsequent reduction of Bias Triggers within recruitment data has been developed. This proactive approach aimed to facilitate the development and refinement of recruitment models that operate with enhanced fairness and impartiality. As an initial step in this process, the automated system receives the specifications for a given employment vacancy. Subsequently, an analytical process is initiated to evaluate the alignment between the stated job requirements and the role title, ascertain the rationality of experience prerequisites relative to the position's level, assess the feasibility of the outlined responsibilities and requirements for a single incumbent, and detect any logical inconsistencies or contradictions within the job description. Furthermore, this evaluation extends to the identification of potentially exploitative practices embedded within vacancy details, culminating in a summarized assessment accompanied by a quantitative compliance score derived from these analytical metrics. Complementary to this vacancy analysis, a dedicated mechanism was implemented to process the candidate data. This mechanism is specifically designed to ingest input data, meticulously preserve the original structural format, and systematically eliminate attributes recognized as potential sources of bias. These attributes include but are not limited to name, age, gender, sexual orientation, race or ethnicity, religious affiliation, disability status, and marital or parental status, thereby promoting a more equitable evaluation of candidate qualifications.

3. **Digital signature of the relevant human actors -** A significant impediment to efficient talent acquisition arises from the prevalence of incongruous job specifications. A substantial portion of these issues could be mitigated through the implementation of a validation mechanism involving subject matter experts with a profound understanding of the requisite technical skills. This additional layer of scrutiny serves not only to prevent the inadvertent exclusion of suitably qualified candidates, but also reinforces the accuracy and relevance of the initial requirements definition. To facilitate this crucial validation step, a graphical user interface is implemented, enabling at least one individual with pertinent field expertise, such as a project manager, to review and explicitly approve the vacancy requirements prior to public dissemination, subsequent to the automated humanization process. The underlying technology employed to ensure the integrity and traceability of this approval workflow is a distributed ledger system based on the blockchain principles.

4.2 Architecture

The proposed system architecture illustrated in Figure 4.1 delineates the implementation of the aforementioned functionalities as a modular service, thereby facilitating seamless integration into existing recruitment ecosystems. On the left-hand side of the diagram, an entity or corporation utilizing an ATS is depicted as the primary consumer of the **ATS Humanization Service.** This service is designed as a centralized module capable of executing two key processes: a comprehensive analysis of vacancy requirements, and the systematic removal of bias-related data from applicant information. Interaction is initiated when the ATS transmits either a vacancy description or applicant résumés to the Humanization Service. Within this service, a dedicated **Humanization Service** component orchestrates initial processing. Subsequently, a **Data Processing** module undertakes the core analytical tasks, potentially leveraging an LLM for sophisticated text analysis and pattern recognition. The output of this processing, **humanized data** , which may encompass insights derived from vacancy analysis or bias-redacted candidate information, is then relayed back to the originating ATS. This service-oriented architecture promotes modularity and reusability, allowing for straightforward incorporation of advanced humanization capabilities into diverse ATS platforms.

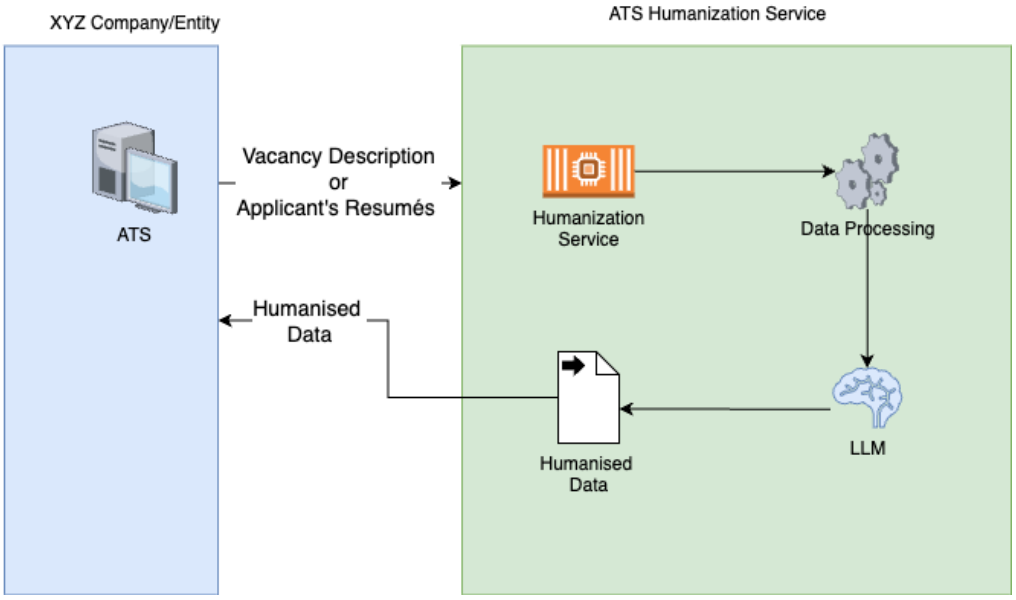


Figure 4.1: Solution architecture.

4.2.1 Blockchain - Ethereum, Hardhat, and Smart Contracts

To underpin the integrity and auditability of the expert validation process, distributed ledger technology based on blockchain principles was implemented. Specifically, the Ethereum platform, which is a decentralized open-source blockchain with smart contract functionality, was chosen

because of its robust ecosystem and widespread adoption. Smart contracts, self-executing contracts with the terms of the agreement directly written into code, were developed to record and store the approval status of each job vacancy requirement set. This ensured a transparent and tamper-proof record of the validation process, providing a high degree of trust and accountability. Hardhat facilitated the development and testing of these smart contracts, a local Ethereum development network designed for efficiency and ease of use. Hardhat provided a controlled environment for compiling, deploying, and testing smart contracts before their potential deployment in a live Ethereum network. The use of smart contracts on the Ethereum blockchain guarantees that, once a vacancy's requirements are approved by a designated expert, this approval is permanently recorded and cannot be unilaterally altered, thereby enhancing the trustworthiness of the entire recruitment process.

4.3 Implementation

4.3.1 Interactive Web-Based Frontend: Architectural Design and Implementation

To effectively illustrate the practical applicability and facilitate user interaction with the proposed solution, a publicly accessible web interface was developed and deployed. An architectural overview of this interactive platform is shown in Figure 4.2.

As illustrated, the user engages with a *Frontend Application*, accessible via the URL `https://joblimpo.valdompinga.com/`. This frontend serves as the primary point of interaction, allowing users to input queries, configure parameters, and visualize the outputs generated by the underlying system. The front-end application communicates bidirectionally with the *Nginx* web server. The Nginx server acts as a reverse proxy that directs incoming requests to the backend infrastructure. This design choice enhances security and load-balancing capabilities.
The core logic of our solution resides within a *Backend Application* containerized using *Docker*. This containerized environment ensures consistency and portability across the different deployment environments.

The backend application was built using the FastAPI framework, chosen for its convenience for data handling and robust APIs. The FastAPI application is responsible for receiving requests from the front-end (via Nginx), processing them according to the implemented methodology, and crucially interacting with the LLM (see Figure 4.2.

The user interface of the system was implemented as an interactive website designed to provide intuitive access to its core functionalities. The website available at `https://joblimpo.valdompinga.com/` is structured around four principal pages, each serving a distinct purpose in facilitating the validation of vacancy requirements and the mitigation of bias in the candidate data.

The **Landing Page**, illustrated in Figure 4.3, serves as the initial point of contact for the users. This page provides an overview of the key features of the application, highlighting its capabilities in enhancing fairness and objectivity in the recruitment processes.

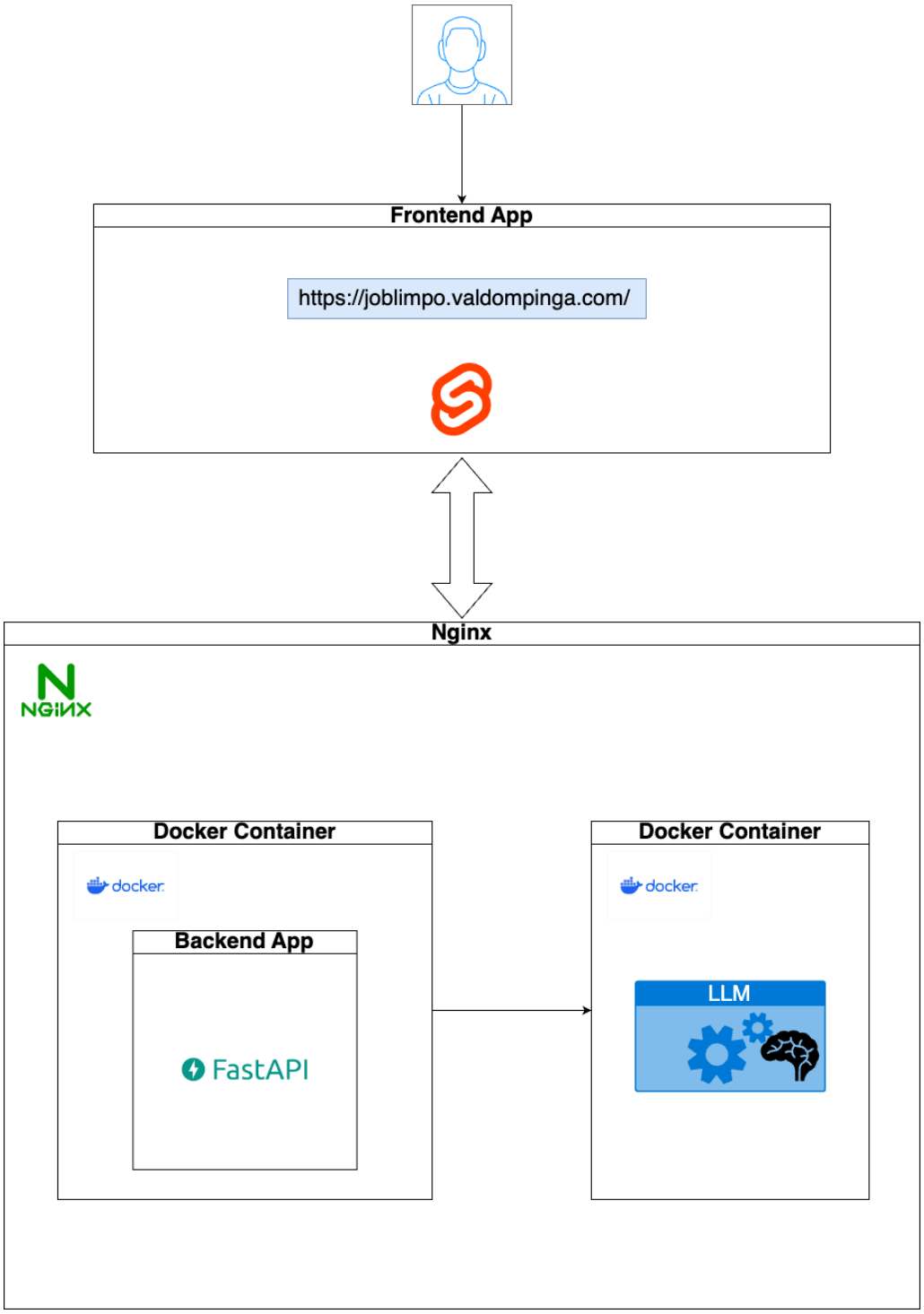


Figure 4.2: Architectural Overview of the Interactive Web Interface.

The **Vacancy Requirement Validation Page** depicted in Figure 4.4 allows any user to analyze a job description against predefined ethical and practical criteria. Users can input job-vacancy text and receive an evaluation of its compliance, identifying potential issues related to role alignment, experience rationality, workload feasibility, and internal discrepancies.

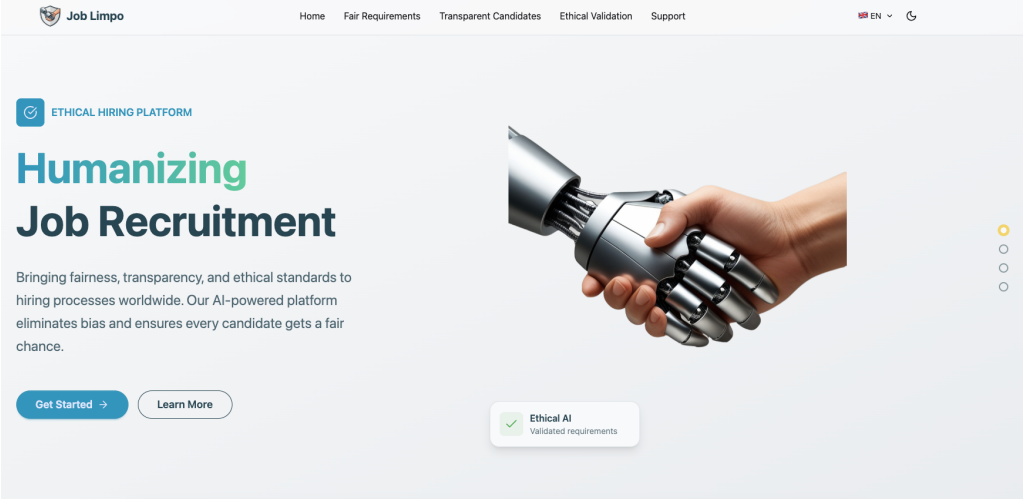


Figure 4.3: Overview of the application's features as presented on the landing page.

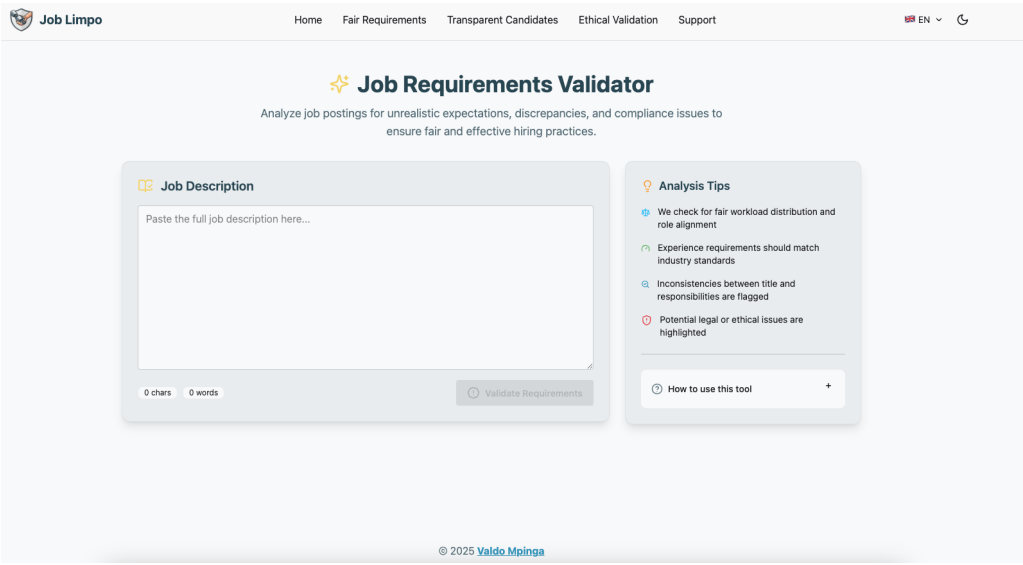


Figure 4.4: Interface for analyzing job vacancy descriptions and receiving compliance evaluations.

The **Candidate Data Bias Removal Page** shown in Figure 4.5 provides a feature for users to process candidate-like data with the aim of removing or anonymizing information that could potentially trigger biases in automated applicant tracking systems. This functionality supports a more objective assessment of candidate qualifications.

Finally, the **Ethical Validation Page**, illustrated in Figure 4.6, provides information regarding ethical considerations and the implementation of a blockchain-based solution for handling the validation of code signatures on platforms such as GitHub. This page emphasizes the system's commitment to transparency and integrity in its operations.

These four main pages collectively offer a comprehensive and user-friendly interface for interacting with the system's functionalities, from understanding its core features to actively utilizing

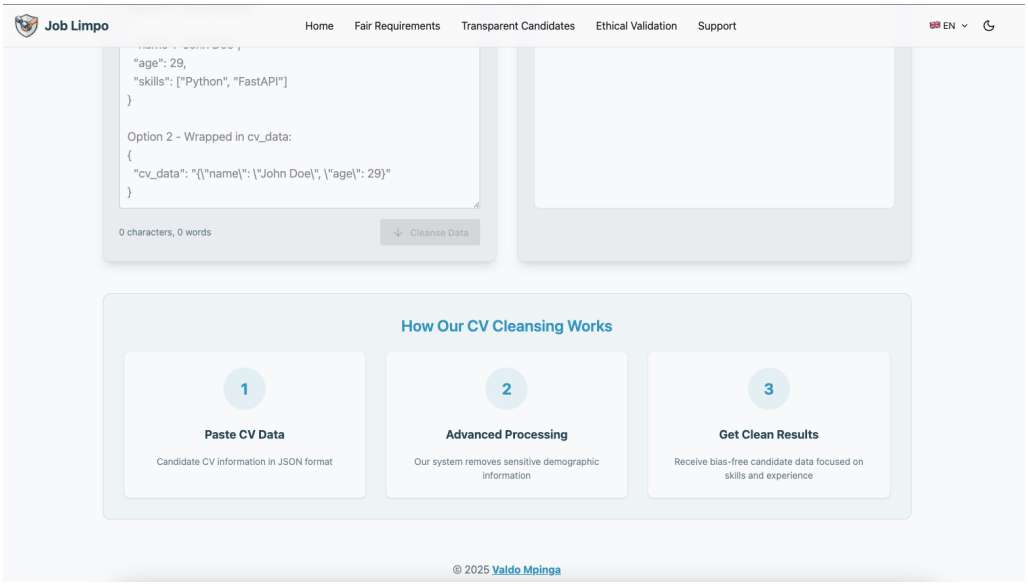


Figure 4.5: Tool for processing candidate data to mitigate potential bias triggers.

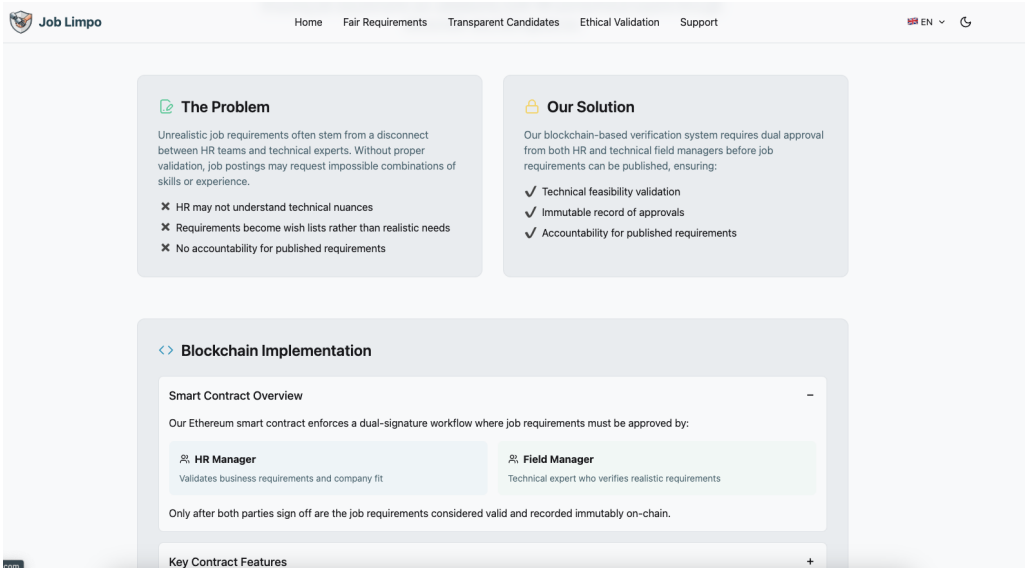


Figure 4.6: Information on the ethical framework and the blockchain-based code signature validation mechanism.

its tools for vacancy analysis and bias mitigation. The website adheres to responsive design principles, ensuring accessibility across various screen sizes, as shown in 4.7.

4.3.2 Backend Implementation - Validation of Vacancy Requirements and Mitigation of Bias Triggers

The backend infrastructure of the system, implemented as a service utilizing the FastAPI framework in Python, provides critical functionalities for both the evaluation of job vacancy requirements and the identification and removal of potential bias triggers within candidate data. The

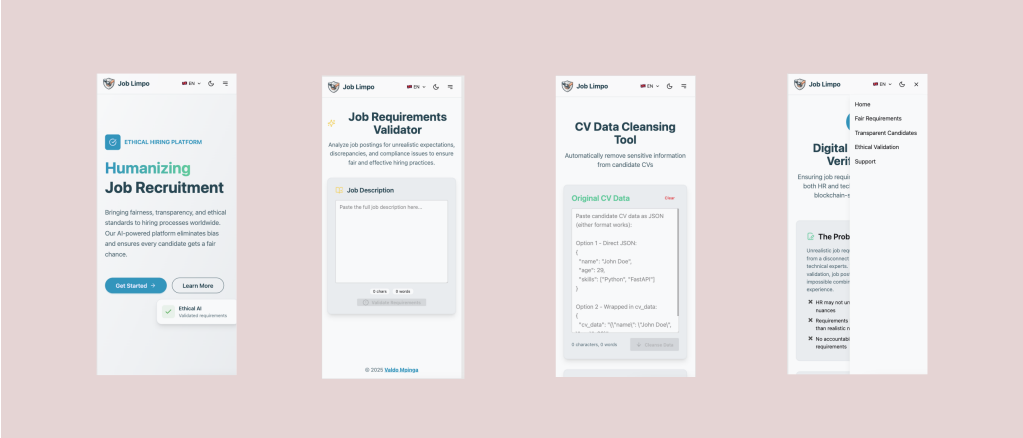


Figure 4.7: Illustration of Responsive Website Design Across Different Screen Sizes.

architecture is designed with modularity and a clear separation of concerns to ensure maintainability and scalability.

The primary entry point for the backend service is defined within the main application file, as illustrated in Figure 4.8. This file configures the core routing and dependencies of the FastAPI application, serving as the central orchestrator for various functionalities exposed through distinct API endpoints.

```python
# main.py
from fastapi import FastAPI
from utils.limiter import limiter, exception_handler
from slowapi.errors import RateLimitExceeded

# Import and include routers after app initialization
from routes.candidate import router as candidate_router
from routes.requirements import router as requirements_router
from routes.health import router as health_router

app = FastAPI()

# Initialize rate limiting
app.state.limiter = limiter
app.add_exception_handler(RateLimitExceeded, exception_handler)


app.include_router(health_router)
app.include_router(candidate_router)
app.include_router(requirements_router)
```

Figure 4.8: Core routing configuration of the FastAPI backend service.

To enhance user experience and prevent unnecessary interactions with a potentially unavailable backend, a health check mechanism was implemented. Prior to executing any API request related to core functionalities, such as vacancy requirement validation and bias data removal, a request is made to the health endpoint. A successful response, indicated by an HTTP status code of 200, signifies that the system is operational, allowing the user interface to render these features. Conversely, if the health check fails, the interface is not displayed and a message indicating server

unavailability is presented to the user, as illustrated in Figure 4.9. This proactive approach saves user time by avoiding interactions with non-responsive systems.

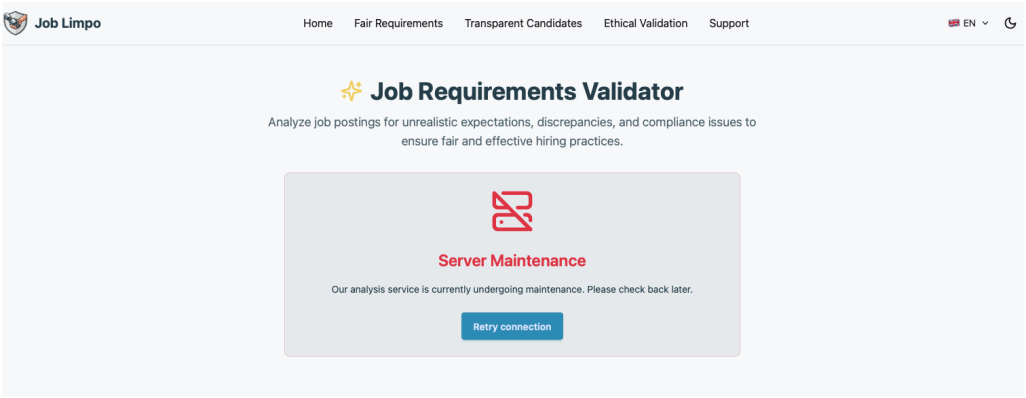


Figure 4.9: User interface displayed when the server is offline.

One of the key functionalities offered by the backend is the `/requirements` end point. As shown in Figure 4.10, this endpoint is responsible for receiving job vacancy data, typically including the job title and a detailed description of the role and its requirements. Upon receiving this information, the backend service invokes the job vacancy validator, which performs the analysis based on defined criteria (role alignment, experience rationality, workload feasibility, and internal discrepancies). The results of this evaluation, including the compliance score and specific feedback, were returned to the requesting client.

```
15  OLLAMA_URL = os.getenv("OLLAMA_URL")
16  openai.api_key = os.getenv("OPENAI_KEY")
17  USE_OPENAI = os.getenv('USE_OPENAI', 'False').strip().lower() == 'true'
18
19  logging.basicConfig(
20      level=logging.INFO,
21      format='%(asctime)s - %(levelname)s - %(message)s',
22      handlers=[
23          logging.FileHandler("app.log"),
24          logging.StreamHandler()
25      ]
26  )
27  logger = logging.getLogger(__name__)
28
29
30  if not OLLAMA_URL:
31      logger.critical("OLLAMA_URL environment variable is not set.")
32      raise HTTPException(status_code=500, detail="OLLAMA_URL environment variable is not set.")
33
34
35  router = APIRouter(prefix="/requirements", tags=["requirements"])
36
37  class VacancyRequest(BaseModel):
38      vacancy: str = Field(..., min_length=100, description="Vacancy description must be at least 100 characters long.")
39
40  @router.post("/")
41  @limiter.limit("10/hour")
42  def evaluate_single_vacancy(request: Request,
43                              vacancy_request: VacancyRequest):
44      logger.info("Received vacancy evaluation request")
45      vacancy = vacancy_request.vacancy
46
47      if not vacancy or not isinstance(vacancy, str):…
54
55      job_validator_prompt = f"""…
108
109     try:…
153
154     except Exception as e:…
161
```

Figure 4.10: Functionality of the `/requirements` endpoint for job vacancy evaluation.

In addition to the vacancy evaluation, the backend also hosts a `/candidate` endpoint, the primary function of which is to process candidate data and mitigate potential bias triggers commonly associated with automated applicant tracking systems (ATS). As shown in Figure 4.11, this endpoint receives candidate information and employs a series of predefined rules and potentially more advanced techniques for identifying, removing, or anonymizing data points that could inadvertently introduce bias during the screening process. This functionality aims to promote a more equitable and merit-based evaluation of the candidates.

```python
#routes.candidate.py
from fastapi import APIRouter, HTTPException, Request
import requests
from pydantic import BaseModel, Field
from dotenv import load_dotenv
import os
from utils.limiter import limiter
import openai
import logging

logger = logging.getLogger(__name__)

load_dotenv()

OLLAMA_URL = os.getenv("OLLAMA_URL")
openai.api_key = os.getenv("OPENAI_KEY")
USE_OPENAI = os.getenv('USE_OPENAI', 'False').strip().lower() == 'true'

router = APIRouter(prefix="/candidate", tags=["candidate"])

class BiasRequest(BaseModel):
    cv_data: str = Field(..., min_length=50, description="CV data must be at least 50 characters long.")

@router.post("/")
@limiter.limit("10/hour")
async def removeBiasFromCandidateData(request: Request, candidate_request: BiasRequest):
    logger.info("Bias removal request received")
    cv_data = candidate_request.cv_data
    logger.debug(f"Input CV data (truncated): {cv_data[:200]}...")

    prompt = f""" …

    try: …

    except Exception as e:
        logger.exception("Unexpected error during bias removal")
        raise HTTPException(
            status_code=500,
            detail=f"Error processing request: {str(e)}"
        )
```

Figure 4.11: Functionality of the `/candidate` endpoint for bias trigger removal from candidate data.

The separation of these critical functionalities into distinct API endpoints within the FastAPI backend allows for a clear and organized approach to both the proactive improvement of job vacancy descriptions and reactive mitigation of bias in candidate processing, contributing to a more human-centered recruitment process.

To ensure the correct operation and reliability of the backend API endpoints, the Insomnia REST Client was employed for comprehensive testing. Figures 4.12 and 4.13 illustrate the process of testing the `/requirements` and `/candidate` endpoints, respectively, using insomnia.

Figure 4.12 shows a typical test scenario for the `/requirements` endpoint, in which a sample job vacancy description is submitted to the API. The figure highlights the structure of the request, including the input data and the corresponding response received from the backend. This

allows for the verification of the validation logic and format of the returned compliance evaluation.

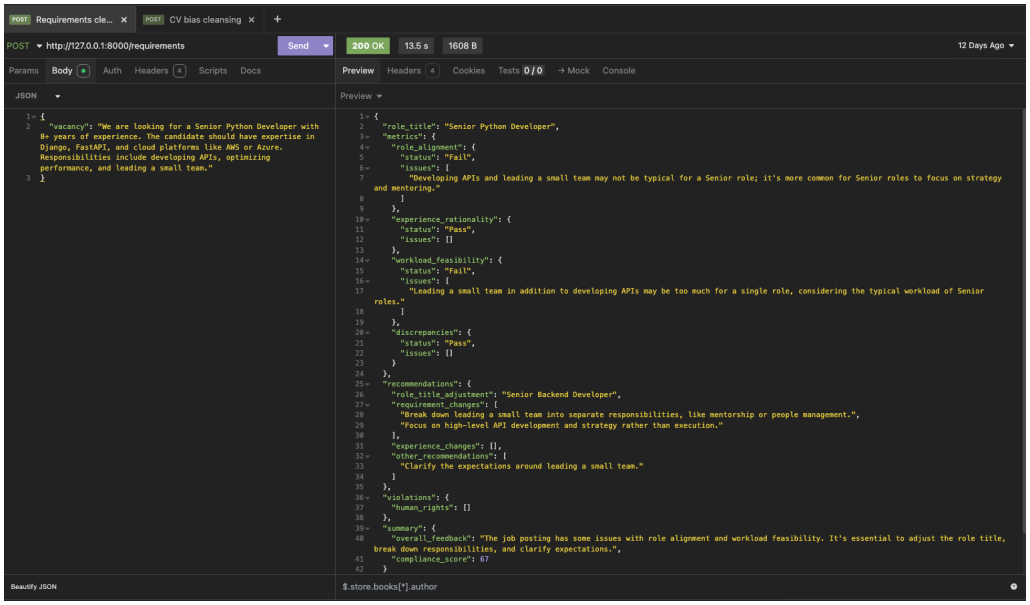


Figure 4.12: Testing the `/requirements` endpoint with a sample job vacancy description using Insomnia REST Client.

Similarly, Figure 4.13 shows the testing of the `/candidate` endpoint. In this scenario, the sample candidate data containing potential bias triggers were sent to the API. The figure illustrates the input data and processed response, demonstrating the effectiveness of the bias removal mechanisms implemented in the backend.

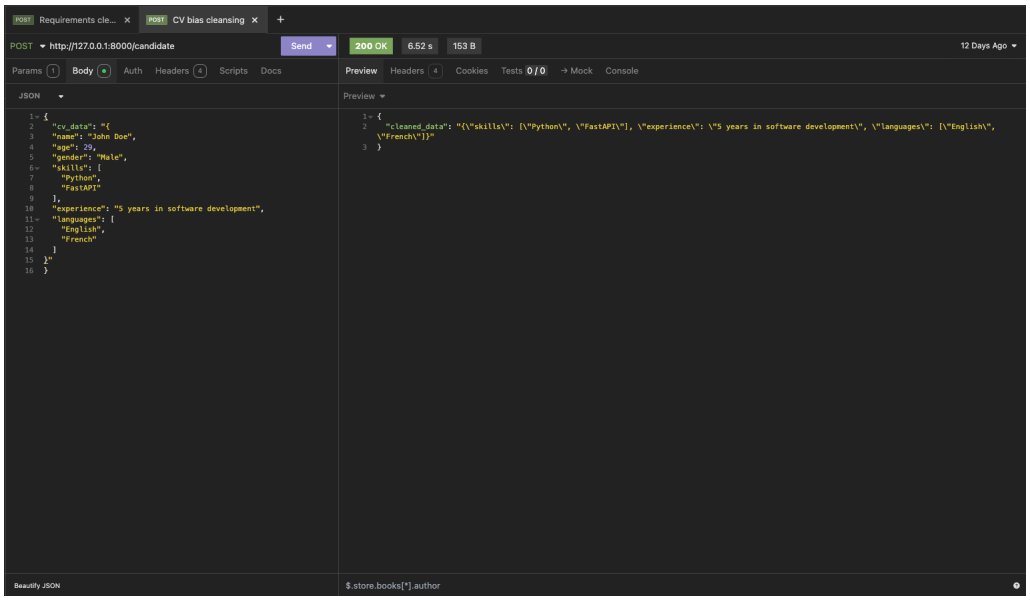


Figure 4.13: Testing the `/candidate` endpoint with sample candidate data for bias trigger removal using Insomnia REST Client.

The use of insomnia facilitated a systematic approach to verifying the functionality of each

backend endpoint, ensuring that the vacancy requirement validation and bias mitigation services operate as intended. The visual representation of requests and responses, as captured in these figures, provides clear evidence of the testing procedures employed during the development process.

+

4.3.3 Validation of Vacancy Requirements - In-depth

This module is designed to receive and analyze employment vacancy posts, typically sourced from Human Resources departments or integrated Applicant Tracking Systems (ATS). Recognizing that such initial posts may inadvertently lack alignment with equitable and human-centered considerations, this endpoint performs a rigorous evaluation to ensure adherence to predefined standards of fairness and rationality. The primary objective is to proactively identify and flag potential issues before a vacancy is publicly disseminated.

To achieve consistent and structured analysis, a meticulously engineered prompt was developed and iteratively refined. This prompts the LLM to process the vacancy description and output its assessment in a standardized JavaScript Object Notation (JSON) format. The final iteration of the prompt utilized for this critical endpoint is presented below:

"You are a Job Requirement Validator. Your task is to evaluate and assess job posts for fairness, rationality, and alignment with the specified role title. Your analysis should specifically address the following aspects:

- ***Role Alignment:*** *Determine the relevance of each listed job requirement to the stated job title.*
- ***Experience Rationality:*** *Assess the reasonableness of the required years of experience in relation to the seniority and complexity of the role.*
- ***Workload Feasibility:*** *Evaluate whether the combined responsibilities and requirements are realistically achievable within the scope of a single position.*
- ***Discrepancy Check:*** *Identify any logical inconsistencies or contradictions present within the job description.*
- ***Human-Centered Feedback:*** *Highlight any elements that may be indicative of potentially exploitative or unfair employment practices.*

*Evaluate the following job posting: {**`ATS VACANCY GOES HERE`**}*
The analysis should be returned in the following structured JSON format:

- ***role_title:*** *"Job Title"*
- ***metrics:***
  - ***role_alignment:***

        * ***status:*** *"Pass/Fail"*
        * ***issues:*** *["List of misaligned requirements"]*
    - ***experience_rationality:***
        * ***status:*** *"Pass/Fail"*
        * ***issues:*** *["Details about unreasonable experience requirements"]*
    - ***workload_feasibility:***
        * ***status:*** *"Pass/Fail"*
        * ***issues:*** *["Details about unrealistic workloads"]*
    - ***discrepancies:***
        * ***status:*** *"Pass/Fail"*
        * ***issues:*** *["Details about discrepancies"]*
- ***recommendations:***
    - ***role_title_adjustment:*** *"Suggested new title if necessary"*
    - ***requirement_changes:*** *["Suggested changes to requirements"]*
    - ***experience_changes:*** *["Suggested changes to experience requirements"]*
    - ***other_recommendations:*** *["Additional advice or changes"]*
- ***violations:***
    - ***human_rights:*** *["List of detected violations, if any"]*
- ***summary:***
    - ***overall_feedback:*** *"Summary of the evaluation"*
    - ***compliance_score:*** *"Percentage of compliance based on metrics (0-100)"*
- ***isJob:*** *"True/Fail"*

If the input is not a job posting, immediately set `isJob` *to "False" and return the rest of the JSON with empty objects or strings. Ensure that the output is strictly in JSON format, without any additional explanations or formatting."*

This carefully constructed prompt is instrumental in ensuring that the LLM consistently generates an analysis in the predefined JSON structure, thereby establishing a reliable and predictable vacancy validation process.

4.3.4 Mitigation of Bias Triggers - In-depth

This module addresses the critical need to mitigate potential biases in recruitment processes by automatically identifying and removing data fields that could inadvertently influence decision making. The endpoint is designed to receive structured data pertaining to job applicants, which may be in JSON, XML (Extensible Markup Language), or plain text format. Upon processing, the endpoint returns the data in an identical original format, but with all the identified bias-inducing attributes systematically removed.

"You are an AI tool specifically designed to sanitize structured data by removing fields associated with potential bias.
The following categories of fields are considered bias-related and must be removed:

- *Personal Identifiers:* Name
- *Demographic Information:* Age, Gender, Sexual orientation, Race or ethnicity, Religion, Disability status
- *Familial Status:* Marital status, Parental status (e.g., fields indicating the presence or number of children)
- *Physical Attributes:* Any form of photos or descriptive text pertaining to physical appearance.

Strict Output Formatting Rules:

1. *The output must be in **exactly the same format** as the input data (JSON, XML, or plain text).*
2. *Do **not** include any supplementary explanations, code snippets, illustrative examples, inline comments, or any other extraneous text. The output should consist solely of the cleaned data.*
3. *The output should **not** be enclosed within code fences (e.g., " ') or any surrounding markdown syntax or comments.*
4. *If the input is valid JSON, the output must be valid JSON.*
5. *If the input is valid XML, the output must be valid XML.*
6. *If the input is plain text, the output must be the cleaned plain text.*
7. *Any key-value pairs where the value corresponds to bias-related information should be entirely removed from the output. Do not retain keys with empty values resulting from the removal process.*
8. *Information regarding languages spoken by the applicant is explicitly **not** considered a bias-related attribute and should be preserved.*

Input Data:
{APPLICANT DATA GOES HERE}
Output Data:
(Return the cleaned input data strictly adhering to the specified format.)"

This precisely defined prompt ensures that the LLM consistently processes applicant data and returns them in the original format, effectively stripping away information that could introduce bias into subsequent evaluation stages.

By implementing these two distinct endpoints, the proposed system directly addresses critical aspects of fairness and equity in the recruitment lifecycle: proactively ensuring that job vacancy postings are human-centered and rational and reactively eliminating bias-inducing personal attributes from applicant data. The effectiveness and reliability of both implementations are predicated on the careful design and refinement of the prompts that guide Large Language Model in performing these specific tasks.

4.3.5 Digital signature of the relevant human actors - In-depth

Traditional methodologies for creating job posts frequently encounter challenges related to the inclusion of unrealistic requirements, often stemming from a lack of sufficient domain-specific knowledge during the drafting phase. To mitigate this critical issue, we propose a decentralized validation protocol based on the necessity of dual cryptographic authorization from both HR personnel and subject-matter experts, prior to the formal publication of any vacancy.

4.3.5.1 Decentralized System Architecture

The proposed system strategically leverages the capabilities of Ethereum smart contracts to rigorously enforce several key operational parameters, the details of which are listed in Table 4.1.

- **Role-Based Access Control**: Implementation of distinct permission frameworks tailored for HR managers and field-specific experts.
- **Multi-Signature Validation**: Utilization of the Elliptic Curve Digital Signature Algorithm (ECDSA) to ensure robust multi-signature verification.
- **Immutable Approval Records**: Secure and transparent recording of all approval processes through the immutable state transitions inherent to the blockchain.

4.3.5.2 Operational Workflow

The workflow follows the following sequential steps:

1. **Job Requirement Formulation**: HR personnel initiate the process by drafting comprehensive job requirements, including the job title, a detailed description of responsibilities, and the relevant job category.

2. **Domain Expert Evaluation**: A designated subject-matter expert with pertinent domain expertise meticulously evaluates the technical feasibility and appropriateness of the drafted job posting.

3. **Cryptographic Endorsement**: Upon satisfactory review, both the responsible HR personnel and the designated domain expert cryptographically sign the finalized job proposal using their private keys.

4. **On-Chain Verification and Recording**: The Ethereum smart contract autonomously verifies the authenticity and validity of the provided digital signatures. Upon successful verification, the contract records the approval of job posting on the blockchain, ensuring an immutable audit trail.

4.3.5.3 Smart Contract Implementation

The complete implementation of the smart contract is in Solidity, the programming language for Ethereum smart contracts, along with comprehensive test cases, is publicly available in our dedicated GitHub repository: `https://github.com/ValdoMpinga/clean-job`.

The key features incorporated within smart contract implementation include the following:

- **Gas-Efficient Signature Recovery**: Optimized utilization of the `ecrecover` precompiled contract to minimize the computational cost (gas) associated with signature verification.

- **Duplicate Submission Prevention**: Implementation of job hashing mechanisms to generate unique identifiers for each job posting, thereby preventing the submission and approval of identical job specifications.

- **Event-Driven Architecture**: Design incorporating event emitters that trigger upon significant state changes (e.g., job approval, rejection), facilitating seamless off-chain monitoring and integration with external systems (see 4.14).

Figure 4.15 presents a visual representation of the data model underpinning the Hardhat Job Approval smart contract. This model outlines the key entities managed by the contract, including HR Managers, Field Managers, Job Types, and Approved Jobs, along with their respective attributes and relationships established between them. The smart contract employs mappings and arrays to maintain and access these data on the Ethereum blockchain, ensuring data integrity and facilitating the job approval process.

4.3.5.4 System Integration and Workflow Phases

The integration and operational lifecycle of the blockchain-based job validation system are structured in two distinct yet interconnected phases:

- The initial smart contract deployment, showcased in figure 4.16 and system configuration.

```
// Event emitted when a job is successfully approved
event JobApproved(…
);

// Event emitted when an HR manager is added
event HRManagerAdded(address hrManager, string name);

// Event emitted when an HR manager is removed
event HRManagerRemoved(address hrManager);

// Event emitted when an HR manager is updated
event HRManagerUpdated(address hrManager, string newName);

// Event emitted when a field manager is added
event FieldManagerAdded(…
);

// Event emitted when a field manager is removed
event FieldManagerRemoved(address fieldManager);

// Event emitted when a field manager is updated
event FieldManagerUpdated(…
);
```

Figure 4.14: Smart contract events.

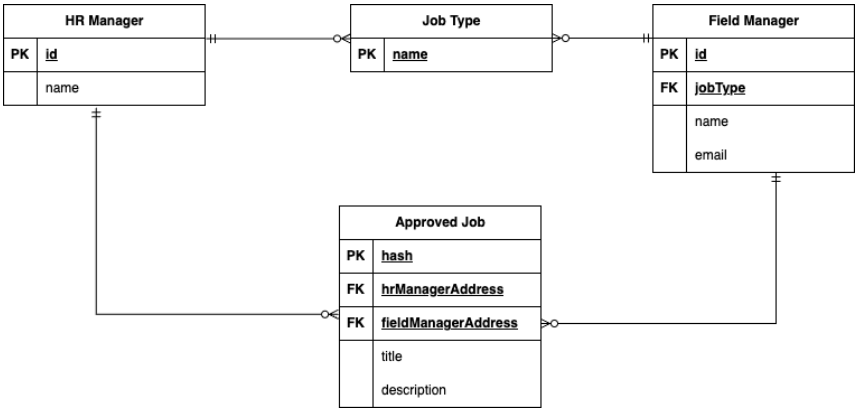


Figure 4.15: Data model of the Hardhat Job Approval smart contract.

- The ongoing validation process for individual job postings.

These critical workflows are visually represented using the Business Process Model and Notation (BPMN) diagrams in Figures 4.17 and 4.20. The core concept is to enable each interested entity or company to establish a local copy of the blockchain and operate it within its organizational scope. Consequently, the BPMN diagrams were designed to reflect this decentralized operational model.

4.3.5.5 Smart Contract Deployment Phase

The initial deployment of the smart contract necessitates the designation of at least one authorized HR manager during the contract's instantiation. As illustrated in Figure 4.17, this initial HR manager is granted administrative privileges that enable them to

Table 4.1: Smart Contract Characteristics

Characteristic	Details
Digital Signature Scheme	ECDSA (secp256k1 curve) via \texttt{ecrecover}
Smart Contract Functionality	Manages HR Managers, Field Managers (by job type and unique email), Records approved jobs (by hash).
Access Control and HR Manager role-based access via a modifier.	
Job Approval Mechanism	Requires valid ECDSA signatures from the HR Manager and designated Field Manager for job type.
Data storage and mapping for field managers (job type), job approval (hash), arrays for HR Managers, and job types.	
Events emitted and tracked job approval and HR/field manager additions/removals/updates for auditing.	

- **HR Personnel Onboarding**: Enroll additional HR personnel into the system using the `addHRManager()` function.
- **Domain Expert Registration**: Register qualified domain experts within the system, associating them with specific job-type taxonomies using the `addFieldManager(jobType)` function.
- **Job-Type Taxonomy Configuration**: Define and manage the various categories of job types recognized by the system (e.g., "Software Engineering", "Biomedical Engineering").

4.3.5.6 Job Posting Validation Phase

Figure 4.20 provides a detailed depiction of the cryptographic validation protocol employed for each job post submitted to the system.

1. **Requirement Drafting**: HR personnel initiate the process by drafting the complete job specifications, including the title, a comprehensive description of the role, and the designated job type.
2. **Expert Assignment and Verification**: The system automatically verifies the existence of a registered field expert who is associated with the specified job type.
    - If no qualified expert is currently registered for the given job type, the job posting transaction is automatically reverted, preventing further processing until an appropriate expert has been onboarded.

```
scripts > deploy.js > main
const hre = require("hardhat");

async function main()
{
    const [hr] = await hre.ethers.getSigners();

    console.log("Deploying JobApproval contract with HR:", hr.address);

    // Get the contract factory
    const JobApproval = await hre.ethers.getContractFactory("JobApproval");
    console.log("Contract factory loaded");

    // Deploy the contract
    console.log("Deploying contract...");
    const jobApproval = await JobApproval.deploy(hr.address, "Alice");

    // Wait for deployment to complete
    await jobApproval.waitForDeployment();

    // Fetch the deployment transaction (Ethers v6 way)
    const deploymentTransaction = jobApproval.deploymentTransaction();

    if (!deploymentTransaction)
    {
        throw new Error("deploymentTransaction is undefined");
    }

    // Log deployment details
    console.log("JobApproval deployed to:", await jobApproval.getAddress());
    console.log("Deployment transaction hash:", deploymentTransaction.hash);
}

main().catch((error) =>
{
    console.error(error);
    process.exitCode = 1;
});
```

Figure 4.16: Smart contract deploy script.

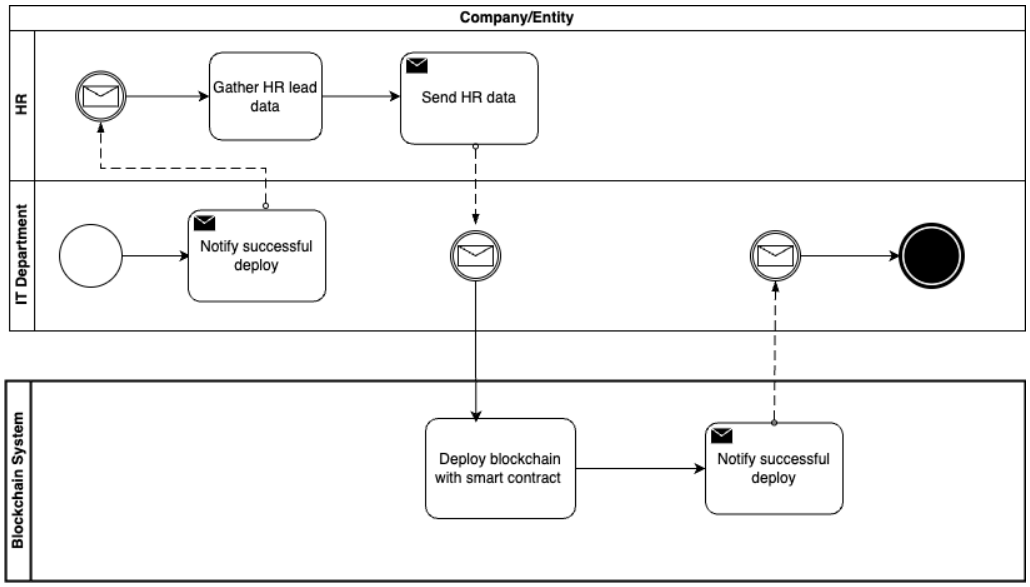


Figure 4.17: Smart contract deployment workflow.

3. **Dual-Signature Validation Flow**:

    - Both the responsible HR personnel and the designated field expert independently generate a digital signature for the cryptographic hash of the job posting details.

- The smart contract then rigorously verifies the authenticity and (see figure 4.18) of both submitted signatures using the `ecrecover` (see figure 4.19) precompiled contract.

```
function verifyJob(
    string memory title,
    string memory description,
    string memory jobType,
    bytes memory hrSignature,
    bytes memory fieldManagerSignature
) public {
    require(
        fieldManagers[jobType].id != address(0),
        "No field manager assigned"
    );

    bytes32 jobHash = getMessageHash(title, description);
    bytes32 ethSignedMessageHash = getEthSignedMessageHash(jobHash);

    // Check if the job has already been approved
    require(!approvedJobs[jobHash], "Job already approved");

    address hrSigner = recoverSigner(ethSignedMessageHash, hrSignature);
    address fieldSigner = recoverSigner(
        ethSignedMessageHash,
        fieldManagerSignature
    );

    bool isHRManager = false;
    for (uint i = 0; i < hrManagers.length; i++) {
        if (hrManagers[i].id == hrSigner) {
            isHRManager = true;
            break;
        }
    }
    require(isHRManager, "Invalid HR signature");

    require(
        fieldSigner == fieldManagers[jobType].id,
        "Invalid Field Manager signature"
    );

    approvedJobs[jobHash] = true;
    emit JobApproved(title, description, hrSigner, fieldSigner);
}
```

Figure 4.18: Job approval function.

```
function recoverSigner(
    bytes32 ethSignedMessageHash,
    bytes memory signature
) public pure returns (address) {
    (bytes32 r, bytes32 s, uint8 v) = splitSignature(signature);
    return ecrecover(ethSignedMessageHash, v, r, s);
}
```

Figure 4.19: Signatures recover function.

4. **Immutable On-Chain Recording**:

   - Upon successful verification of both signatures, the approved job posting is securely stored within the `approvedJobs` mapping on the blockchain, creating an immutable record.
   - In instances where the signature verification fails or other validation criteria are not met, the smart contract emits a `JobRejected` event, signaling the rejection of the proposal.

This robust two-phase approach to job validation ensures the following critical properties.

- **Enhanced Accountability**: All actions performed within the system, including the drafting and approval of job postings, are immutably linked to the cryptographic identities of the participating HR personnel and domain experts.
- **Automated Fail-Safety**: The smart contract incorporates automated checks and verifications, causing transactions to revert in the event of invalid states or unmet criteria, thereby ensuring the integrity of the validation process.
- **Comprehensive Auditability**: The inherent transparency and immutability of the blockchain provide a complete and auditable history of all job posting approvals and rejections, fostering trust and accountability within the hiring process.

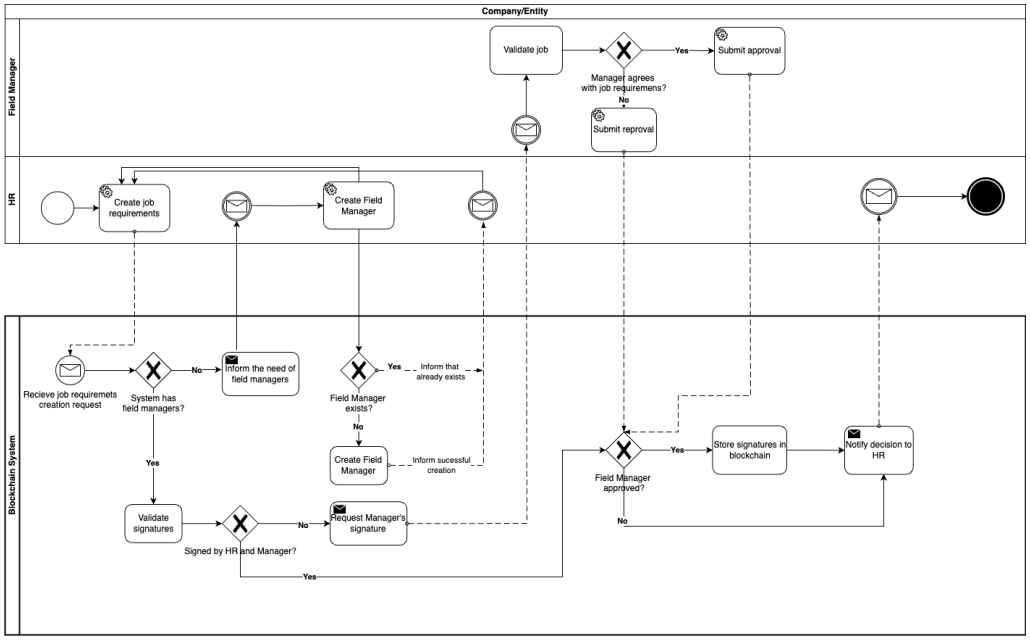


Figure 4.20: Detailed job posting validation workflow utilizing dual cryptographic signatures.

Chapter 5

Validation and Results

A suite of targeted tests and practical system demonstrations were conducted to rigorously assess the efficacy and robustness of the proposed system. The subsequent subsections detail the methodologies and outcomes of the evaluations.

5.1 Empirical Validation of Job Vacancy Requirements for Publication

To rigorously assess the efficacy of the job vacancy requirement validation service, an expanded data-driven evaluation was conducted using a substantial corpus of real-world job vacancy listings. The publicly accessible "US Jobs on Monster.com" Kaggle dataset[1], a comprehensive collection encompassing 22,000 job posts originating from the United States, served as the foundational data source for this enhanced analysis. In contrast to prior assessments, 21,701 job posts were scrutinized by the validation service. The difference of 299 job posts was due to unusable data entries that contained missing information, incomplete job descriptions, or formatting issues that prevented proper analysis by the validation system.

The computational infrastructure employed for this evaluation consisted of a domestic desktop system featuring an AMD Ryzen 7 770X processor, 64 GB DDR5 RAM, and an NVIDIA GeForce RTX 4070 graphics processing unit. The validation service was executed within a Windows Subsystem for Linux (WSL) environment running Ollama 3.1:8b. The complete analysis of the 21,701 job posts was completed in less than 40 h (see Figure 5.1). This timeframe, indicative of efficient processing, suggests potential underutilization of dedicated GPU resources. Monitoring during execution revealed a GPU utilization rate consistently below 10%, indicating a possible bottleneck or configuration issue within the WSL environment that limited the full exploitation of the parallel processing capabilities of RTX 4070. Despite this observed limitation, the total processing time remains a notable achievement for the scale of the analyzed dataset.

[1] https://www.kaggle.com/datasets/PromptCloudHQ/us-jobs-on-monstercom

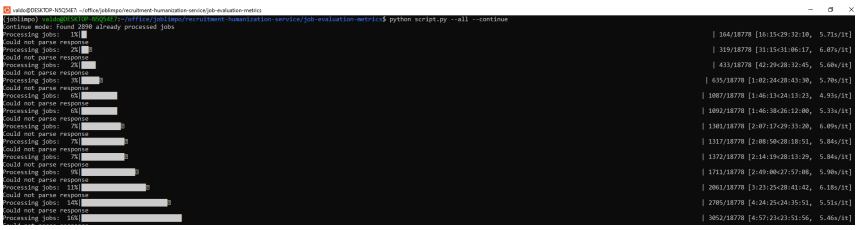


Figure 5.1: Visual representation of the job vacancy validation service in operation, processing the dataset of 21,701 job postings.

5.1.1 Distribution of Overall Compliance Scores

Following a comprehensive analysis of 21,701 job vacancy listings, the distribution of the derived overall compliance scores provides valuable insights into the general quality and adherence to the principles of fairness and rationality within contemporary job postings. Figure 5.2 presents a graphical depiction of this distribution, illustrating the frequency of compliance scores across the analyzed dataset.

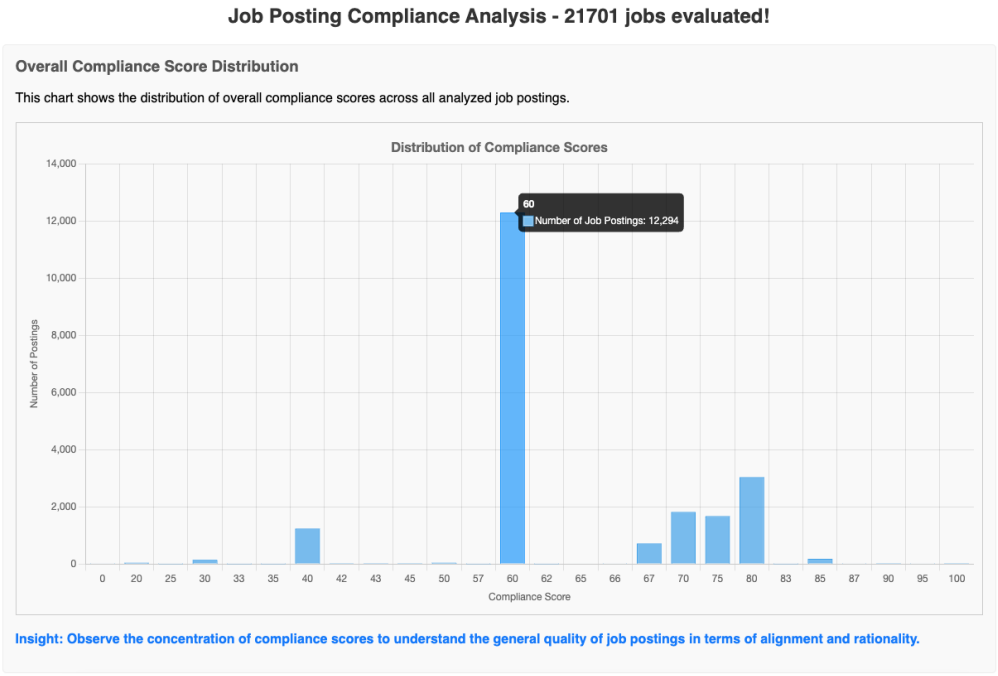


Figure 5.2: Distribution of overall compliance scores across 21,701 analyzed job vacancy postings.

As shown in Figure 5.2, the distribution of compliance scores exhibits a notable concentration within the lower to mid-range of the spectrum. A striking observation is the prominent peak at a compliance score of 60, accounting for 12,294 job posts. This significant clustering suggests that a large proportion of the analyzed job vacancies achieve a moderate level of compliance according to the defined metrics of role alignment, experience rationality, workload feasibility, and internal consistency.

However, the distribution also reveals that a considerable number of job posts fell below this central peak. Scores in the range of 40 and below, while less frequent than the 60-point peak, still represent a non-negligible portion of the dataset, indicating the presence of job posts with more pronounced issues across the evaluated criteria. Conversely, the frequency of job postings with higher compliance scores (above 80) appears comparatively lower, suggesting that truly exemplary job postings exhibiting strong alignment, rationality, and consistency are less prevalent within the analyzed sample.

The insight provided in Figure 5.2 underscores the importance of distributional analysis. The observed concentration around the 60% compliance mark, coupled with the presence of lower scores, indicates a prevalent pattern of deficiencies, as identified by the job vacancy validator. These deficiencies, as determined by the underlying evaluation prompt, generally pertain to the issues of role alignment, experience rationality, workload feasibility, and internal discrepancies. Subsequent analysis of individual metric scores will further elaborate on the prevalence of these specific categories of issues within the dataset, aligning with the criteria defined for automated evaluation.

5.1.2 Compliance Status by Individual Metric

To gain a more granular understanding of the specific areas where job vacancy posts frequently deviate from the principles of fairness and rationality, an analysis of the compliance status for each individual metric was conducted. Figure 5.3 presents a comparative overview of the number of job posts that passed or failed for each of the four key evaluation criteria: Role Alignment, Experience Rationality, Workload Feasibility, and Discrepancies.

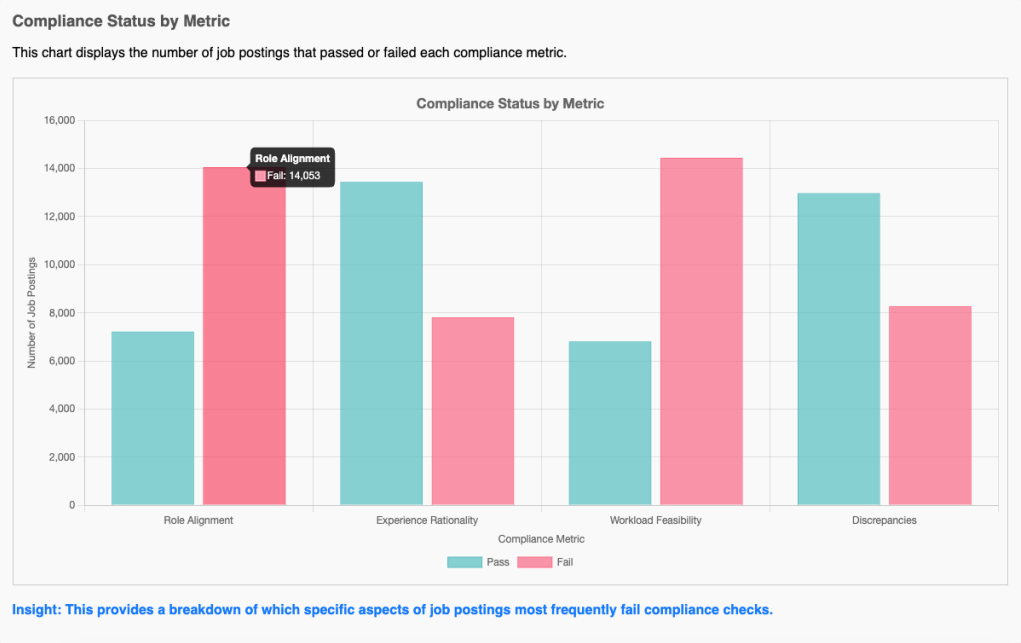


Figure 5.3: Compliance status (Pass/Fail) for each evaluation metric across 21,701 analyzed job vacancy postings.

The data presented in Figure 5.3 reveal distinct patterns of compliance across different metrics. Regarding **Role Alignment**, 14,053 job postings were identified as failing to demonstrate a clear and logical connection between the stated requirements and core responsibilities typically associated with the job title. In contrast, 7,216 posts were deemed compliant.

In the assessment of **Experience Rationality**, the results indicate a more balanced distribution. While 7,816 job postings failed to justify their experience demands as reasonable for the described role and industry standards, 13,444 postings were considered to have rational experience requirements.

The evaluation of **Workload Feasibility** highlights a significant challenge. A considerable majority of the analyzed job postings, totaling 14,445, were flagged as presenting potentially unrealistic workload expectations for a single individual within standard working hours. Only 6,812 posts were assessed to have feasible workload descriptions.

Finally, the analysis of **Discrepancies** within the job posts revealed that 8,270 posts contained internal inconsistencies or contradictory information. A larger proportion (12, 997 posts) were found to be free of such internal discrepancies.

The insight provided in Figure 5.3 is clearly supported by these findings. The metric exhibiting the most frequent failures is **Workload Feasibility**, where a significant majority of posts suggest potentially unrealistic expectations. This is closely followed by **Role Alignment**, indicating a common disconnect between the job requirements and the core responsibilities of the role. While **Experience Rationality** shows a better pass rate, a substantial number of posts still demand potentially unreasonable levels of prior experience. Finally, **Discrepancies**, although less frequent than the other failure points, still represents a notable lack of internal consistency within a portion of the analyzed job descriptions. These patterns collectively underscore the need for greater diligence in crafting job posts that are clear, realistic, and accurately reflective of the intended roles.

5.1.3 Top Recommended Role Title Adjustments

The analysis conducted by the job vacancy validator also included the generation of recommendations for adjusting job titles, where the initial title appeared misaligned with the described responsibilities and requirements. Figure 5.4 presents the top ten most frequently recommended adjustments to job titles identified across the analyzed dataset of 21,701 job postings.

As illustrated in Figure 5.4, the most frequent recommendation involved adjusting the title to "Store Manager or Assistant Store Manager," with 58 such recommendations. This suggests a recurring pattern in which the initial job titles in these posts may have been less specific or potentially misleading regarding the actual level and scope of the role.

The second most frequent recommendation was for the Customer Service Representative: " although the precise number of recommendations for this adjustment was not explicitly labeled in the provided image. Following this, "Sales Representative" and "Administrative Assistant" also appear as commonly recommended adjustments, indicating potential ambiguities or inaccuracies in the original titling of these roles.

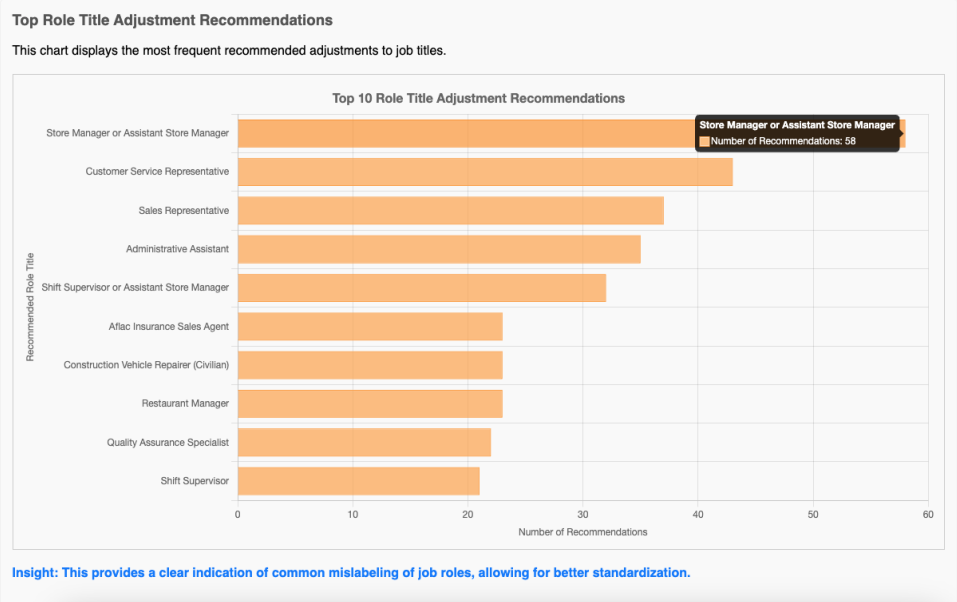


Figure 5.4: Top 10 most frequent recommended adjustments to job titles.

The insight provided below the chart, "This provides a clear indication of common mislabeling of job roles, allowing for better standardization," highlights a significant issue within the analyzed job postings. The frequency of these title adjustment recommendations suggests a lack of precision or consistency in the definition and labelling of job roles. This mislabeling can lead to confusion for jobseekers, potentially resulting in applications for unsuitable positions and inefficiencies in the recruitment process.

The prevalence of recommendations for titles such as "Store Manager or Assistant Store Manager" and the various "Representative" roles suggests a need for more specific and descriptive job titles that accurately reflect the responsibilities and required skills. Addressing these inconsistencies in job titling can contribute to a more transparent and efficient job market.

5.1.4 Top Recommended Requirement Changes

In addition to evaluating overall compliance and suggesting role title adjustments, the job vacancy validator provided specific recommendations for modifying the listed requirements to enhance clarity, rationality, and feasibility. Figure 5.5 illustrates the top 10 most frequently suggested changes to job requirements identified within the analyzed dataset of 21,701 job posts.

The analysis of the recommended requirement changes, as depicted in Figure 5.5, reveals a strong emphasis on the need for greater specificity and manageability in job descriptions. The most frequent recommendation, with 155 occurrences, was to "**Break down the responsibilities into more specific and manageable tasks**" which suggests a prevalent tendency in the analyzed job posts to list broad or overwhelming sets of responsibilities without a clear delineation or realistic expectations for a single role.

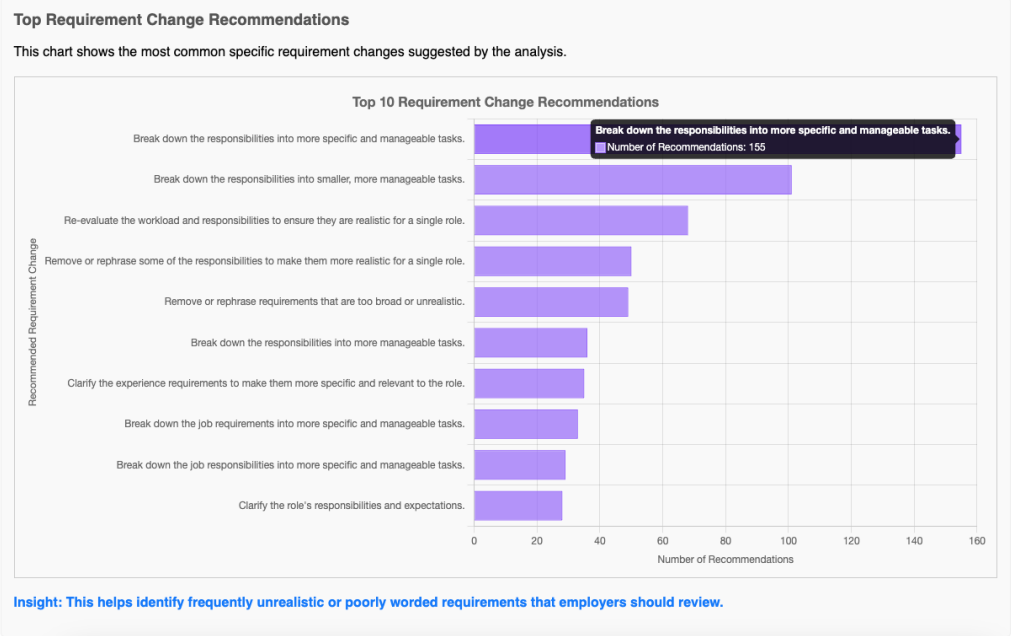


Figure 5.5: Top 10 most frequent recommended changes to job requirements.

Following this, another common recommendation was to "**Break down the responsibilities into smaller, more manageable tasks**" (the number of recommendations is not explicitly labeled but visually significant). These two recommendations underscore the recurring issue of potentially overloaded or vaguely defined roles.

Several other frequent recommendations also point towards the need to refine the articulation of responsibilities and workload. These include:

- *Re-evaluate the workload and responsibilities to ensure they are realistic for a single role.*
- *Remove or rephrase some of the responsibilities to make them more realistic for a single role.*
- *Remove or rephrase requirements that are too broad or unrealistic.*

Recommendations focusing on clarity and specificity were also prominent, such as "*Clarify the experience requirements to make them more specific and relevant to the role*" and "*Clarify the role's responsibilities and expectations.*" Additionally, suggestions to "*Break down the job responsibilities into more specific and manageable tasks*" appeared multiple times within the top 10, further reinforcing the need for granular and realistic descriptions.

The insight provided below the chart, "This helps identify frequently unrealistic or poorly worded requirements that employers should review," directly reflects the findings. The prevalence of recommendations to break down responsibilities, re-evaluate workload, and clarify requirements indicates that a significant portion of the analyzed job postings suffer from a lack of specificity, potentially unrealistic expectations, or overly broad statements. Addressing these

issues through more detailed and realistic descriptions would likely lead to more effective recruitment processes and a better alignment between job expectations and the capabilities of potential candidates.

5.1.5 Overall Conclusion of the Analysis

The comprehensive analysis of 21,701 job vacancy listings reveals a discernible trend, indicating significant shortcomings in the quality and human-centricity of a substantial portion of contemporary job postings. The distribution of overall compliance scores, peaking at a modest 60%, suggests that while many job descriptions meet a basic level of adherence to the evaluation criteria, a considerable number fall short, and truly exemplary postings are relatively rare.

The examination of compliance status using individual metrics further illuminates specific areas of concern. Notably, **Workload Feasibility**, **Role Alignment**, and **Experience Rationality** exhibited high failure rates, indicating a widespread tendency towards unrealistic workload expectations, a disconnect between stated requirements and the job role, and potentially exclusionary or disproportionate experience demands.

Furthermore, the prevalence of recommendations for **Role Title Adjustments** underscores the lack of precision and consistency in job titling practices, potentially causing confusion and hindering effective matching between job seekers and opportunities. The most frequent **Requirement Change Recommendations**, particularly the repeated emphasis on breaking down responsibilities into more specific and manageable tasks, strongly suggest that many job descriptions suffer from vagueness, overambition, and a lack of clarity regarding the actual scope and demands of the role.

Collectively, these analytical findings provide compelling empirical evidence that supports the notion that a significant proportion of current job vacancy listings exhibit characteristics that can be deemed dehumanizing. The lack of clear role alignment, imposition of unreasonable experience demands, implication of unrealistic workloads, and presence of internal inconsistencies contribute to a recruitment landscape that may be opaque, discouraging, and potentially detrimental to both job seekers and employers. These data strongly advocate the need to humanize the recruitment process through interventions that promote clarity, rationality, and realistic expectations in job vacancy descriptions, thereby fostering a more equitable and efficient job market.

5.2 Evaluation of different Large Language Models by number of parameters using Job Requirement Analysis

This section presents an evaluation of the various open-source LLMs. The goal was to assess their capability to complete tasks that require pattern-like outputs (see figure 5.6). Because the job requirements prompt has many requirements to check, it is a good use case to evaluate LLMs from very small to medium-sized ones.

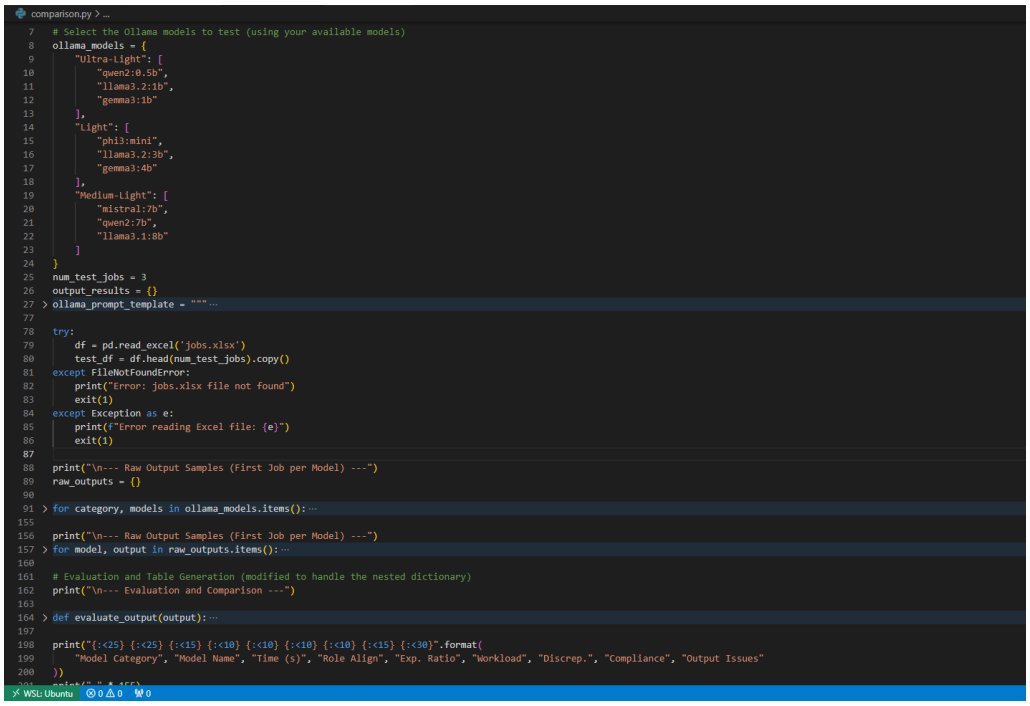


Figure 5.6: LLMs comparison script.

5.2.1 Large Language Model Parameters and Significance

LLMs are complex statistical models trained on massive datasets of text and code [12]. Their ability to generate human-like text is governed by a key parameter, the number of *parameters*. A parameter in an LLM is a variable that the model learns during the training. These parameters determine the strength of the connections between words and concepts, thereby enabling the model to predict the next word in a sequence.

Generally, a larger number of parameters allows an LLM to capture more intricate patterns in language, leading to improved performance in various tasks. However, increasing the parameter count also significantly increases the model size, thereby requiring more computational resources (memory and processing power) for inference [12]. This tradeoff between performance and resource requirements is a crucial consideration when deploying LLMs in real-world applications.

5.2.2 Selected Large Language Models

To evaluate the performance of LLMs in job requirement analysis, we selected a range of open-source models categorized into three size groups to represent different computational resource constraints.

- **Ultra-Light Models:** These models are designed for deployment in resource-constrained environments, such as edge devices or applications with strict latency requirements. They typically have parameter counts in the sub-billion-to-one-billion range. The models in this category are as follows:

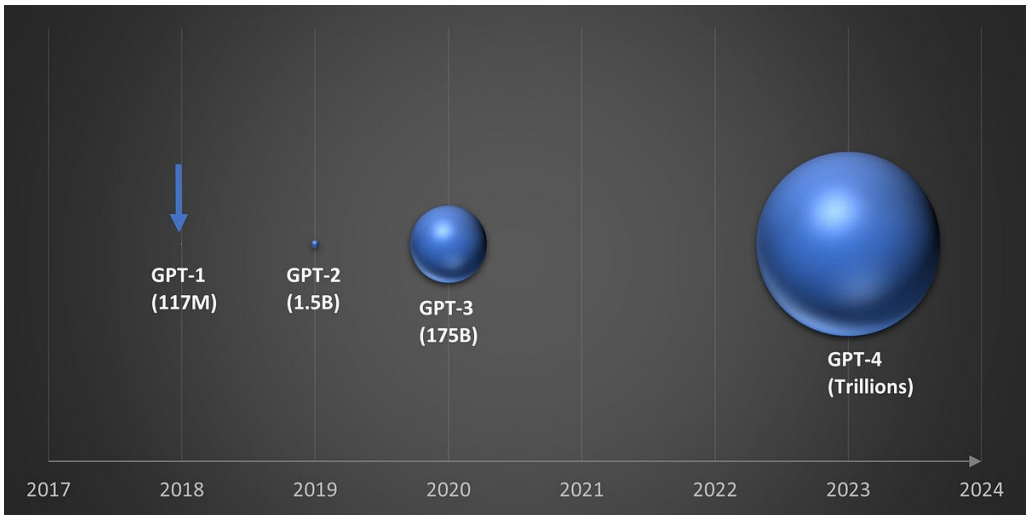


Figure 5.7: LLMs parameters sizes visual demonstration.

  - **Qwen2-0.5B:** A smaller variant of the Qwen series, known for its efficiency.
  - **Llama-3.2-1B:** A compact version of Meta's Llama 3 model, optimized for speed.
  - **Gemma-3-1B:** A lightweight model from Google's Gemma family, emphasizing strong performance with fewer parameters.

- **Light Models:** These models offer a balance between performance and resource usage. They generally have a few billion parameters and can operate on standard hardware with moderate computational resources. The selected models are as follows:

  - **Phi-3-Mini:** A model from Microsoft's Phi-3 family, designed for efficiency and reasoning capabilities.
  - **Llama-3.2-3B:** A slightly larger variant of the Llama 3 model, providing improved performance over the 1B version.
  - **Gemma-3-4B:** A model from Google's Gemma family, offering enhanced capabilities compared to the 1B version.

- **Medium-Light Models:** These models represent a higher capacity for complex language understanding and generation, typically with parameter counts around 7-8 billion. They may require specialized hardware (e.g., GPUs) for optimal performance. The models in this category are:

  - **Mistral-7B:** A model from Mistral AI, known for its strong performance and efficiency.
  - **Qwen2-7B:** A larger model from the Qwen series, capable of handling more complex tasks.
  - **Llama-3.1-8B:** A larger variant of Meta's Llama 3, offering robust performance.

5.2.3 Evaluation Methodology

The selected LLMs were evaluated using a set of job posts extracted from a real-world dataset (Table 5.1). The models were prompted to analyze each posting based on the following criteria.

1. **Role Alignment:** Assessing the relevance of listed requirements to the job title.
2. **Experience Rationality:** Evaluating the reasonableness of experience requirements.
3. **Workload Feasibility:** Determining if the responsibilities and requirements are realistic.
4. **Discrepancy Check:** Identifying inconsistencies in the job description.
5. **Human-Centered Feedback:** Highlighting potential exploitative practices.

The models were instructed to return their analyses to a structured JSON format. We then measured the following metrics.

- **Processing Time:** The time taken by the model to analyze a single job posting.
- **Output Quality:** A qualitative assessment of the model's ability to produce valid and complete JSON output.
- **Evaluation Accuracy:** A subjective evaluation of the model's correctness in assessing the job posting (where available).

5.2.4 Results and Discussion

The evaluation results are summarized in the following table:

5.2.4.1 Analysis of Results

Output Parsing Errors: A significant observation is the prevalence of parsing errors, particularly with the smaller models. Models such as Qwen2-0.5B and Phi-3-Mini frequently failed to produce valid JSON outputs, rendering their evaluations unusable. This indicates that these models, although efficient in terms of computational cost, may struggle with the precision required for structured output generation. Parsing errors are likely due to the model's tendency to include extraneous text or deviate from the specified JSON format.

Processing Time: As expected, there is a general trend of increased processing time with larger models. The Ultra-Light models (e.g., Qwen2-0.5B, Llama-3.2-1B) exhibited the fastest processing times, often below 5 s per job posting. The Medium-Light models (e.g., Mistral-7B, Llama-3.1-8B) generally took longer, with some exceeding 10 seconds. This highlights the trade-off between the speed and model capacity.

Table 5.1: Evaluation of Large Language Models for Job Requirement Analysis

Model Category	Name	Time (s)	Role Align	Exp. Ratio	Work-load	Dis-crep.	Compl.	Output Issues
Ultra-Light	qwen2:0.5b	2.37	N/A	N/A	N/A	N/A	N/A	Could not parse....
Ultra-Light	qwen2:0.5b	1.19	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Ultra-Light	qwen2:0.5b	0.27	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Ultra-Light	llama3.2:1b	4.15	Pass	Fail	Pass	Pass	50	None
Ultra-Light	llama3.2:1b	1.86	Pass	Fail	Pass	N/A	Missing 'summary' key....	
Ultra-Light	llama3.2:1b	1.95	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Ultra-Light	gemma3:1b	5.31	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Ultra-Light	gemma3:1b	4.35	Pass	Fail	Pass	Pass	48/100	None
Ultra-Light	gemma3:1b	3.38	Pass	Fail	Pass	Pass	65/100	None
Light	phi3:mini	14.01	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Light	phi3:mini	4.59	N/A	N/A	N/A	N/A	N/A	Parsing Error ....
Light	phi3:mini	3.84	Pass	Pass	Fail	Fail	60	None
Light	llama3.2:3b	10.79	Fail	Pass	Fail	Pass	60	None
Light	llama3.2:3b	3.47	Fail	Pass	Fail	Pass	60	None
Light	llama3.2:3b	2.68	Fail	Pass	Fail	Fail	0	None
Light	gemma3:4b	14.65	Pass	Fail	Fail	Pass	75	None
Light	gemma3:4b	7.78	Pass	Fail	Fail	Pass	40	None
Light	gemma3:4b	7.11	Pass	Fail	Fail	Pass	65	None
Medium-light	mistral:7b	13.31	Pass	Pass	100	Pass		
Medium-light	mistral:7b	5.97	Pass	Fail	Pass	Fail	40	None
Medium-light	mistral:7b	6.16	Pass	Pass	Fail	Pass	80	None
Medium-light	qwen2:7b	14.14	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Medium-Light	qwen2:7b	1.32	N/A	N/A	N/A	N/A	N/A	Parsing Error: ....
Medium-light	qwen2:7b	5.98	Pass	Fail	Pass	Fail	50	None
Medium-light	llama3.1:8b	17.86	Fail	Pass	Fail	Pass	70	None
Medium-Light	llama3.1:8b	5.50	Fail	Pass	Pass	Pass	67	None
Medium-light	llama3.1:8b	6.25	Pass	Fail	Pass	Fail	40	None

Evaluation Accuracy and Consistency: Among the models that produced parsable output, Mistral-7B consistently demonstrated strong performance in terms of evaluation accuracy, as reflected in the higher compliance scores. However, Mistral-7B showed variability across different job postings. The smaller models (e.g., Llama-3.2-1B and Gemma-3-1B) showed more inconsistency and a tendency to miss some nuances in the job descriptions.

Model Capacity and Complexity: The results suggest that larger models with higher parameter counts are better equipped to handle the complexity of the job requirement analysis task. This task requires a nuanced understanding of language, the ability to identify subtle biases, and the capacity to reason the implications of different requirements. Smaller models, while faster, may lack the capacity to reliably perform these complex evaluations.

Humanization Potential: Despite the challenges with output formatting and consistency in some models, the overall feasibility of using LLMs to automate job requirement analysis is promising. Models such as Mistral-7B have demonstrated the potential to provide valuable insights into

the fairness and rationality of job postings. This technology can assist human reviewers in identifying potential issues, ultimately contributing to a more human-centered recruitment process.

5.2.5 Conclusion

Our evaluation demonstrates that LLMs can be valuable tools for automating and enhancing the analysis of job requirements. However, careful consideration must be given to the trade-off between the model size, computational cost, and output reliability. Although smaller models offer speed and efficiency, larger models generally provide more accurate and consistent evaluations. Further research is needed to improve the ability of smaller LLMs to reliably generate structured output and to develop robust methods for validating the accuracy of LLM-generated evaluations.

5.3 Functional Verification of Bias Trigger Identification

The bias trigger identification service endpoint underwent functional testing to ensure its operational efficacy in processing candidate data and selectively extracting relevant professional information, while omitting potentially biased demographic details. In typical Applicant Tracking System (ATS) workflows, candidate Curriculum Vitae (CV) data are parsed and subsequently transformed into structured data formats such as JSON, XML, or plain text. To simulate this input and rigorously evaluate the functionality of the service, mock candidate data were generated, adhering to the representative JSON format presented below:

```
{
  "name": "John Doe",
  "age": 29,
  "gender": "Male",
  "sexual_orientation": "Heterosexual",
  "race": "Caucasian",
  "religion": "Christian",
  "disability_status": "None",
  "marital_status": "Single",
  "children": 0,
  "languages_spoken": ["English", "Spanish"],
  "photo": "http://example.com/photo.jpg",
  "address": "123 Main St, Cityville, USA"
}
```

The anticipated output from the bias trigger identification service, designed to retain only professionally relevant information, was the following JSON structure.

```
{
  "languages_spoken": ["English", "Spanish"],
```

```
  "address": "123 Main St, Cityville, USA",
  "email": "johndoe@example.com",
  "phone": "123-456-7890",
}
```

The tests confirmed that the bias trigger identification endpoint successfully processed the mock candidate data and accurately produced the expected output, effectively filtering out specified demographic attributes while retaining essential contact and professional information.

5.4 Comprehensive Test Results for the Blockchain Validation System

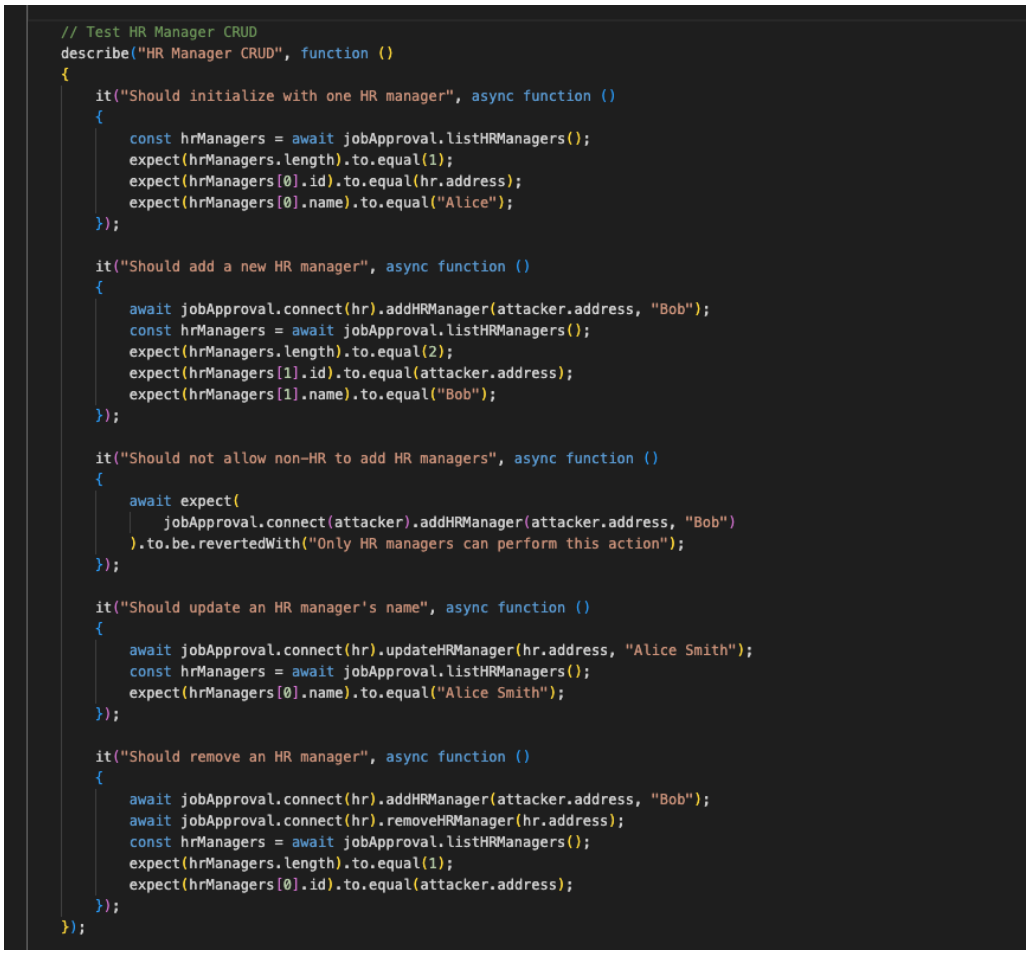
The core validation logic within the blockchain validation system's smart contract was subjected to a rigorous verification process using a test suite developed with Hardhat for the testing environment and Chai for assertions. This comprehensive suite specifically targets three critical aspects fundamental to the system's security and operational integrity.

5.4.1 Thorough Access Control Testing

- **HR Manager Exclusivity**: The tests (showcased in 5.8) successfully confirmed that the system strictly enforces HR manager exclusivity for administrative functions, effectively rejecting all unauthorized attempts to modify critical system parameters or roles by non-HR actors.
- **Field Expert Assignment Validation**: The test suite (see figure 5.9) validated the system's ability to enforce the assignment of field experts based on specific job categories, ensuring that only designated experts are authorized to provide validation for relevant job postings.
- **Secure Role Revocation**: Functionality for the revocation of assigned roles was tested and confirmed to operate correctly without causing any corruption or inconsistencies in the system's internal state (see figure 5.10).

5.4.2 Robust Signature Verification Testing

- **Invalid Signature Detection**: The tests demonstrated a 100% 5.11 detection rate for both invalid HR and field manager digital signatures, ensuring that only cryptographically authorized personnel can endorse job postings.
- **Unauthorized Approval Prevention**: The ECDSA (Elliptic Curve Digital Signature Algorithm) validation mechanism effectively prevented all attempts at unauthorized job approvals by entities lacking the required valid digital signatures.



```
// Test HR Manager CRUD
describe("HR Manager CRUD", function ()
{
    it("Should initialize with one HR manager", async function ()
    {
        const hrManagers = await jobApproval.listHRManagers();
        expect(hrManagers.length).to.equal(1);
        expect(hrManagers[0].id).to.equal(hr.address);
        expect(hrManagers[0].name).to.equal("Alice");
    });

    it("Should add a new HR manager", async function ()
    {
        await jobApproval.connect(hr).addHRManager(attacker.address, "Bob");
        const hrManagers = await jobApproval.listHRManagers();
        expect(hrManagers.length).to.equal(2);
        expect(hrManagers[1].id).to.equal(attacker.address);
        expect(hrManagers[1].name).to.equal("Bob");
    });

    it("Should not allow non-HR to add HR managers", async function ()
    {
        await expect(
            jobApproval.connect(attacker).addHRManager(attacker.address, "Bob")
        ).to.be.revertedWith("Only HR managers can perform this action");
    });

    it("Should update an HR manager's name", async function ()
    {
        await jobApproval.connect(hr).updateHRManager(hr.address, "Alice Smith");
        const hrManagers = await jobApproval.listHRManagers();
        expect(hrManagers[0].name).to.equal("Alice Smith");
    });

    it("Should remove an HR manager", async function ()
    {
        await jobApproval.connect(hr).addHRManager(attacker.address, "Bob");
        await jobApproval.connect(hr).removeHRManager(hr.address);
        const hrManagers = await jobApproval.listHRManagers();
        expect(hrManagers.length).to.equal(1);
        expect(hrManagers[0].id).to.equal(attacker.address);
    });
});
```

Figure 5.8: HR actions test suite.

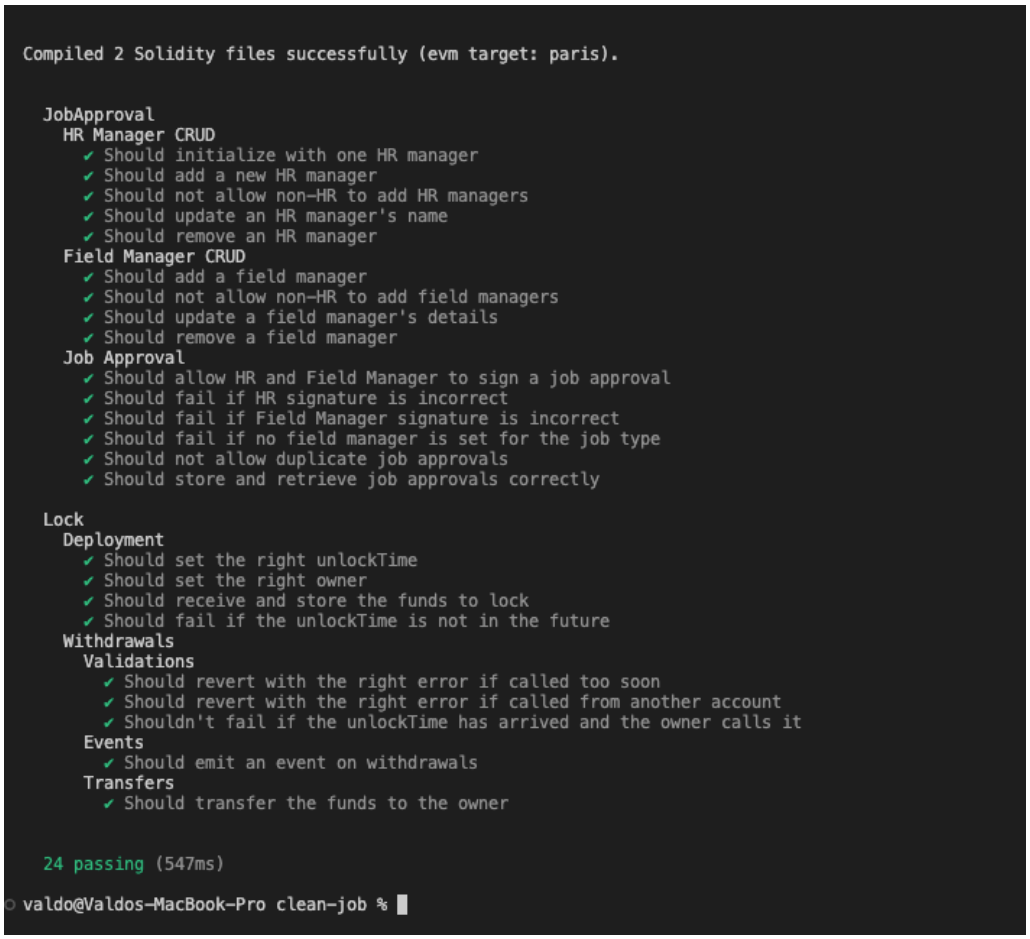
```
// Test Field Manager CRUD
describe("Field Manager CRUD", function ()
{
    beforeEach(async function ()
    {
        // Add a field manager for testing
        await jobApproval.connect(hr).addFieldManager(fieldManager.address, "Charlie", "IT", "charlie@example.com");
    });

    it("Should add a field manager", async function ()⋯
    });

    it("Should not allow non-HR to add field managers", async function ()⋯
    });

    it("Should update a field manager's details", async function ()⋯
    });


    it("Should remove a field manager", async function ()⋯
    });
});
```

Figure 5.9: Field managers actions test suite.

- **Duplicate Submission Blocking**: The implemented job hashing mechanism successfully blocked all attempts to submit and approve duplicate job postings, maintaining the integrity and uniqueness of validated vacancies.

```
    // Test Job Approval
    describe("Job Approval", function ()
    {
        beforeEach(async function ()
        {
            // Add a field manager for testing
            await jobApproval.connect(hr).addFieldManager(fieldManager.address, "Charlie", "IT", "charlie@example.com");
        });

        it("Should allow HR and Field Manager to sign a job approval", async function ()
        });

        it("Should fail if HR signature is incorrect", async function ()
        });

        it("Should fail if Field Manager signature is incorrect", async function ()
        });

        it("Should fail if no field manager is set for the job type", async function ()
        });

        it("Should not allow duplicate job approvals", async function ()
        });

        it("Should store and retrieve job approvals correctly", async function ()
        });
    });
});
```

Figure 5.10: Job approval test suite.



```
Compiled 2 Solidity files successfully (evm target: paris).


  JobApproval
    HR Manager CRUD
      ✔ Should initialize with one HR manager
      ✔ Should add a new HR manager
      ✔ Should not allow non-HR to add HR managers
      ✔ Should update an HR manager's name
      ✔ Should remove an HR manager
    Field Manager CRUD
      ✔ Should add a field manager
      ✔ Should not allow non-HR to add field managers
      ✔ Should update a field manager's details
      ✔ Should remove a field manager
    Job Approval
      ✔ Should allow HR and Field Manager to sign a job approval
      ✔ Should fail if HR signature is incorrect
      ✔ Should fail if Field Manager signature is incorrect
      ✔ Should fail if no field manager is set for the job type
      ✔ Should not allow duplicate job approvals
      ✔ Should store and retrieve job approvals correctly

  Lock
    Deployment
      ✔ Should set the right unlockTime
      ✔ Should set the right owner
      ✔ Should receive and store the funds to lock
      ✔ Should fail if the unlockTime is not in the future
    Withdrawals
      Validations
        ✔ Should revert with the right error if called too soon
        ✔ Should revert with the right error if called from another account
        ✔ Shouldn't fail if the unlockTime has arrived and the owner calls it
      Events
        ✔ Should emit an event on withdrawals
      Transfers
        ✔ Should transfer the funds to the owner


  24 passing (547ms)

valdo@Valdos-MacBook-Pro clean-job %
```

Figure 5.11: 100% test coverage.

5.4.3 End-to-End Workflow Integrity Testing

- **Mandatory Expert Assignment Enforcement**: The tests confirmed that the system enforces the mandatory assignment of a qualified field expert for a given job category before the validation process can be initiated, ensuring that all job postings receive appropriate

domain-specific review.

- **Consistent State Management**: The test suite verified that the system maintains a consistent and accurate internal state across all Create, Read, Update, and Delete (CRUD) operations related to HR managers, field experts, and job postings.
- **Approval History Immutability**: The tests confirmed that the approval history of job postings, once recorded on the blockchain, remains immutable and cannot be retroactively altered or tampered with after transaction finalization.

The collective results of this comprehensive test suite provide strong empirical evidence that the blockchain validation system correctly and securely implements the following.

- Robust role-based access control mechanisms, ensuring that only authorized entities can perform specific actions.
- Rigorous cryptographic requirements validation through ECDSA signatures, guaranteeing the authenticity and integrity of approvals.
- Full adherence to all functional requirements as originally outlined in the smart contract's design specifications.

5.5 System Demonstration via Public Web Interface

This interactive platform implements three core functionalities, providing tangible demonstrations of the system's capabilities.

5.5.1 Interactive Job Requirement Analysis

- **Accessibility**: The job requirement analysis tool is publicly accessible via the following URL: `https://joblimpo.valdompinga.com/requirements`.
- **Real-Time Evaluation**: This interface enables users to perform real-time evaluations of job postings against a predefined set of human-centered criteria, providing immediate feedback on potential issues.
- **Natural Language Processing**: The tool leverages the underlying validation framework to process natural language input from job postings, identifying and highlighting areas of concern based on the defined metrics.

5.5.2 Bias-Free Candidate Presentation Interface

- **Accessibility**: The candidate anonymization service demonstration is available at: `https://joblimpo.valdompinga.com/candidate`.

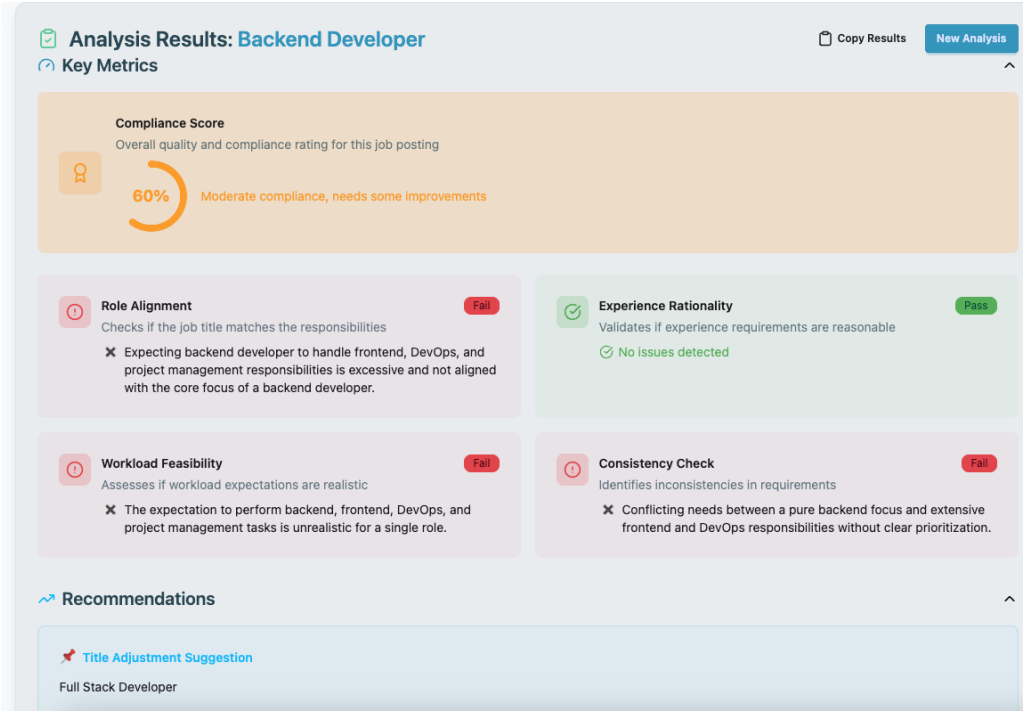


Figure 5.12: Interactive web interface for job requirement evaluation, displaying dehumanization detection metrics derived from natural language analysis.

- **Demonstration Across Multiple Data Formats**: This section showcases the practical implementation of the proposed candidate anonymization solution, illustrating its effectiveness in processing candidate data presented in JSON, XML, and plain text formats.
- **Bias Indicator Removal**: The following figures demonstrate how the system effectively identifies and removes sensitive demographic indicators from candidate profiles provided in each of these formats, while preserving essential professional qualifications and experience details, thereby mitigating potential unconscious biases.

JSON Format Figure 5.13 illustrates the anonymization process for candidate data provided in the JSON format.

XML Format Figure 5.14 illustrates the anonymization process for candidate data provided in the XML format.

Plain Text Format Figure 5.15 illustrates the anonymization process for the candidate data provided in plain-text format.

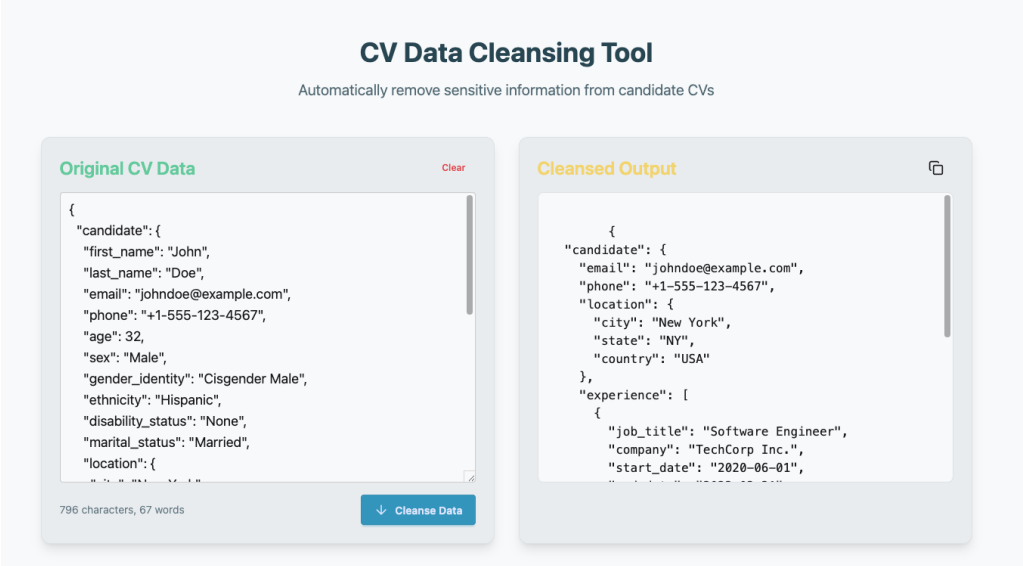


Figure 5.13: Web interface demonstrating candidate profile processing with bias-inducing information removal for JSON format.

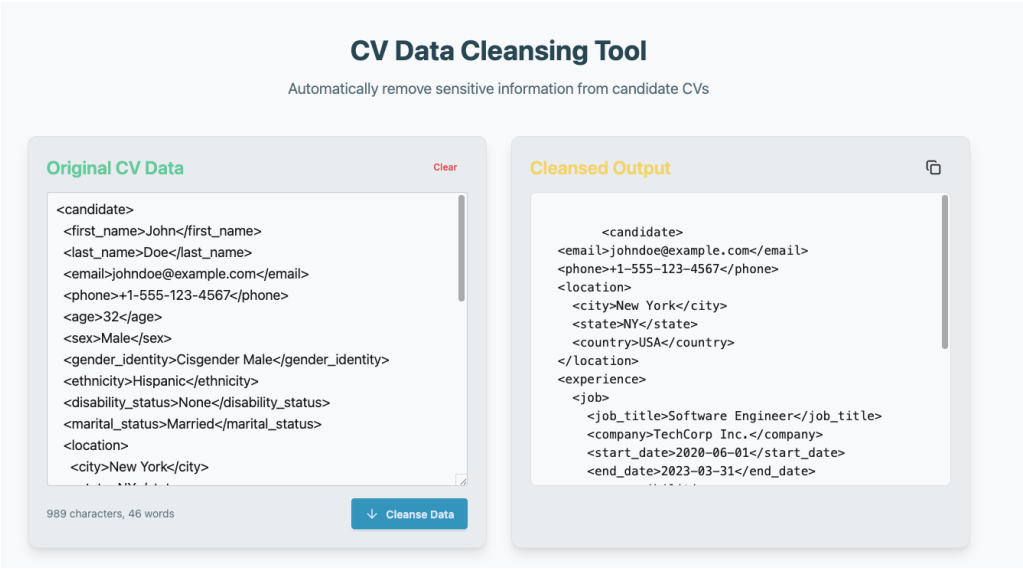


Figure 5.14: Web interface demonstrating candidate profile processing with bias-inducing information removal for XML format.

5.6 Blockchain Implementation Showcase

- **Platform Access**: The interface providing access to the blockchain implementation details is hosted at: `https://joblimpo.valdompinga.com/validation`.
- **Open-Source Repository Link**: This section provides a direct link to the project's public GitHub repository, which contains the following critical components:

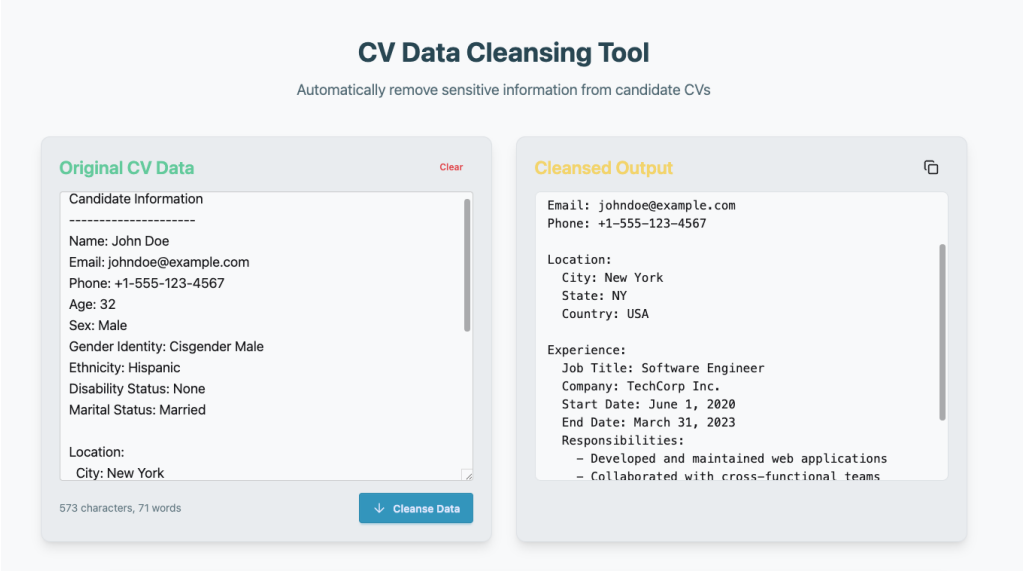


Figure 5.15: Web interface demonstrating candidate profile processing with bias-inducing information removal for plain text format.

- The complete Solidity source code for the smart contracts implementing the blockchain validation logic.
- The comprehensive Hardhat test suite used for rigorous contract verification.
- Deployment scripts facilitating the deployment and initialization of the smart contracts on the Ethereum network.

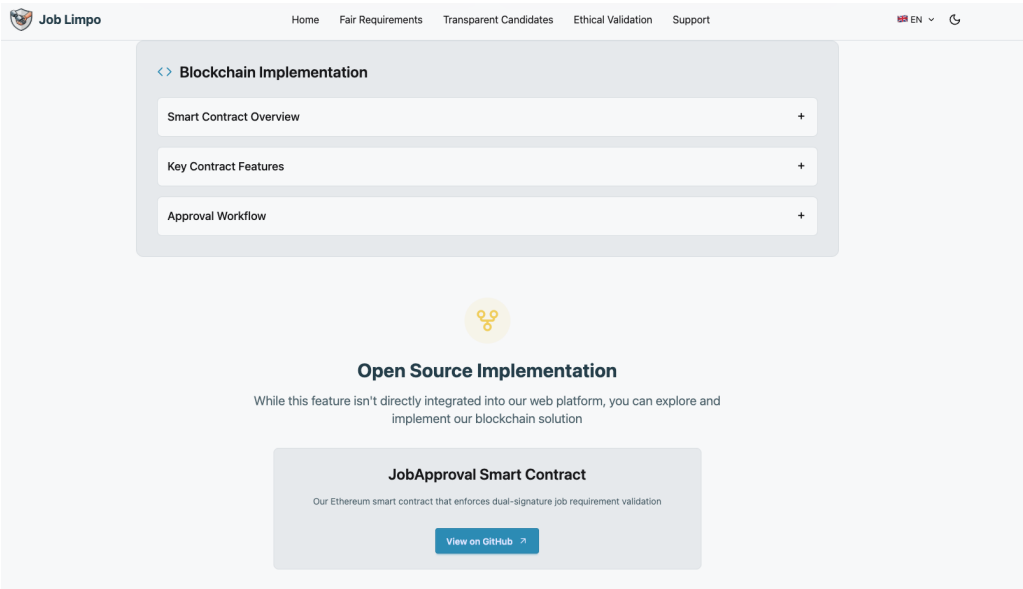


Figure 5.16: Web interface providing a link to the open-source blockchain solution repository.

Chapter 6

Conclusions

The persistent dehumanization of job roles and inadvertent violation of fundamental human rights within recruitment practices constitute a significant and ethical concern. Empirical evidence robustly demonstrates that the neglect of these critical issues not only undermines the inherent dignity of individuals but also directly contravenes the established tenets of human rights law. The accelerating proliferation of AI across various societal domains, including talent acquisition, has led to the increasing deployment of sophisticated AI algorithms within ATS and broader recruitment workflows. Alarmingly, these algorithms frequently exhibit a propensity to inherit and amplify pre-existing biases present in their training data. This phenomenon often arises unintentionally, wherein the processes of data curation, feature selection, and model training are inadvertently embedded and perpetuate societal and historical inequities.

To cultivate a more inclusive talent acquisition environment that is capable of attracting a diverse pool of highly qualified candidates, the presence of rational and equitable job vacancies within the labor market is indispensable. Regrettably, this ideal is not always realized in practice. Companies often construct job postings based on internally perceived "appropriate" keywords, sometimes leading to the inclusion of unrealistic or overly specific requirements, and occasionally even stipulating levels of prior experience that exceed pragmatic industry norms. The central tenet of this work is to provide compelling evidence that organizations can effectively humanize the multifaceted aspects of their recruitment processes without incurring exorbitant expenditures of financial capital, extensive time commitments, or the need for vast computational infrastructure. By focusing on targeted interventions and leveraging accessible technologies, this study demonstrates a pathway towards more ethical and equitable talent acquisition.

The imperative to humanize recruitment extends beyond mitigating the bias in job requirements and user data. Consider the often-neglected aspect of the candidate feedback. It is demonstrably inefficient and arguably dehumanizing to expect human resource professionals to provide personalized feedback to thousands of applicants at each vacancy. However, LLMs offer a scalable and cost-effective solution. By leveraging LLM, organizations can generate tailored feedback based on the specific requirements of the roles and attributes of individual candidates. This not only enhances the candidate experience, but also fosters a sense of respect and value, transforming

a traditional impersonal process into a more human-centric interaction.

The practical implementation of such solutions is surprisingly accessible even for enterprises with budgetary constraints. First, numerous powerful LLM models are available under open-source licenses, significantly reducing software acquisition costs. Second, the technical barrier to entry for deploying these models is relatively low, often requiring only basic programming skills and not an extensive computational infrastructure. Standard enterprise-grade hardware is frequently used in these applications. Consequently, the integration of LLMs for tasks such as personalized feedback represents a tangible and affordable step for organizations to actively contribute to a more ethical and equitable recruitment landscape, shifting away from practices that inadvertently violate the fundamental rights and dignity of jobseekers.

References

[1] Evgeni Aizenberg, Matthew J. Dennis, and Jeroen van den Hoven. Examining the assumptions of AI hiring assessments and their impact on job seekers' autonomy over self-representation. *AI and Society*, 2023. All Open Access, Hybrid Gold Open Access.

[2] Monirah Aleisa, Mona Alshahrani, Natalia Beloff, and Martin White. TAIRA-BSC - Trusting AI in recruitment applications through blockchain smart contracts. In *2022 IEEE INTERNATIONAL CONFERENCE ON BLOCKCHAIN (BLOCKCHAIN 2022)*, pages 376–383, 2022. 5th IEEE International Conference on Blockchain (Blockchain), Espoo, FINLAND, AUG 22-25, 2022.

[3] María Del Carmen Fernández Martínez and Alberto Fernández. AI in recruiting. multi-agent systems architecture for ethical and legal auditing. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August:6428 – 6429, 2019.

[4] Alessandro Fabris, Nina Baranowska, Matthew J. Dennis, David Graus, Philipp Hacker, Jorge Saldivar, Frederik Zuiderveen Borgesius, and Asia J. Biega. Fairness and bias in algorithmic hiring: a multidisciplinary survey. *ACM Transactions on Intelligent Systems and Technology*, 2024.

[5] Estrela Ferreira Cruz and António Miguel Rosado da Cruz. Design science research for IS/IT projects: Focus on digital transformation. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6, 2020.

[6] Shakti Kinger, Divija Kinger, Shivam Thakkar, and Devashish Bhake. Towards smarter hiring: resume parsing and ranking with YOLOv5 and DistilBERT. *Multimedia Tools and Applications*, 83(35):82069 – 82087, 2024.

[7] Marian Nastase, Gabriel Croitoru, Nicoleta Valentina Florea, Nicoleta Cristache, and Ramona Lile. The perceptions of employees from romanian companies on adoption of artificial intelligence in recruitment and selection processes. *AMFITEATRU ECONOMIC*, 26(66), MAY 2024.

[8] Dhyana Paramita, Simon Okwir, and Cali Nuur. Artificial intelligence in talent acquisition: exploring organisational and operational dimensions. *International Journal of Organizational Analysis*, 32(11):108 – 131, 2024.

[9] Ken Peffers, Tuure Tuunanen, Charles E Gengler, Matti Rossi, Wendy Hui, Ville Virtanen, and Johanna Bragge. Design science research process: A model for producing and presenting information systems research, 2020.

[10] K. D. V. Prasad and T. De. Generative AI as a catalyst for HRM practices: mediating effects of trust. *Humanities and Social Sciences Communications*, 11:1362, 2024.

[11] Recruitify.ai. The evolution of applicant tracking systems (ats): From manual processes to ai-powered recruitment. https://www.recruitify.ai/blog/en/the-evolution-of-applicant-tracking-systems-ats-from-manual-processes-to-ai 2023. Accessed: 2025-05-01.

[12] Hugo Touvron, Mathilde Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[13] Josephine Yam and Joshua August Skorburg. From human resources to human rights: Impact assessments for hiring algorithms. *Ethics and Information Technology*, 23(4):611 – 623, 2021.