

Prototyp Stellenmanager KI

Prototypname

Stellenmanager KI

Ziel

Ziel ist es, die Prozesse bei Stellenbeschreibungen und Stellenbewertungen substantiell zu beschleunigen und qualitativ zu verbessern. Für Nutzer stets als „Decision Support System“ konzipiert.

Themenfeld / Bedarfscluster

Automatisierung (mit KI)

Ausschreibungsstart / Kickoff

September 2024 / November 2024

Fokus

KI-Unterstützung bei Stellenbeschreibungen (insb. für Führungskräfte)

Entwicklerfirma

YOUniquehorns GmbH



Funktionalitäten

Der KI Stellenmanager ist darauf ausgerichtet, Prozesse rund um die Arbeitsplatzbeschreibung substantiell zu beschleunigen und qualitativ zu verbessern. Im Zentrum steht die Prüfung, wie KI in der Personalverwaltung eingesetzt werden kann, um gezielt zu transparenten, fairen und effizienten Prozessen beizutragen.

Die technische Umsetzung erfolgt durch einen Retrieval Augmented Generation (RAG) Ansatz, der bestehende, bereits bewertete Arbeitsplatzbeschreibungen als Referenzdatenbank integriert. Dies gewährleistet, dass die KI-gestützten Vorschläge und Bewertungen auf fundierten Vergleichsdaten basieren. Die Integration fortschrittlicher Sprachmodelle ermöglicht:

1. Detaillierte und strukturierte Evaluierung der beschriebenen Aufgaben und deren Anforderungsniveau

Einleitung

Stellenprofil

Aufgabenbeschreibung

Schlüsselkompetenzen

Anforderungen

Prüfung

Aufgabenbeschreibung

Bitte nennen Sie wesentlichen Aufgaben für die Stelle sowie eine ausführliche, vollständige und verständliche Beschreibung der Tätigkeit. Führen Sie für die Erledigung der Tätigkeit konkret benötigte Fachkenntnisse auf und jennen Sie jeweils Beispiele.

KI Unterstützung
Über den grünen Button in der Aktionsleiste kann eine Prüfung der Aufgabenbeschreibung durchgeführt werden.

1. Aufgabenbereich

Entfernen

Allgemeine Aufgabendarstellung

Anforderungsanalyse und Spezifikation

Zeitanteil

35%

Detaillierte Beschreibung der Tätigkeit

Analyse der Anforderungen der Kunden, Spezifikation der notwendigen Funktionen und Eigenschaften, Beratung der Kunden bei not-wendigen Prozessänderungen, Koordinierung der unterschiedlichen Interessen der Stakeholder, Anfertigung von User Storys

Fachliche Kompetenzen

Vertiefte Kenntnisse in UML-Notationen, Anwendungskennntnisse von MindManager, EnterpriseArchitekt, Figma, Erfahrungen in der Erstellung von Anforderungsanalysen, vertiefte Kenntnisse in der Erstellung von User Storys

2. Aufgabenbereich

Entfernen

Allgemeine Aufgabendarstellung

Festlegen der Softwarearchitektur und Management des Softwareprojektes

Zeitanteil

25%

Detaillierte Beschreibung der Tätigkeit

Festlegung der Organisation, Struktur und Hierarchie des Softwaresystems in Zusammenwirkung der verschiedenen Komponenten
Planung, Überwachung und Steuerung von Softwareprojekten, Konzeption der Schnittstellenanbindung

Fachliche Kompetenzen

Projektmanagement mit Scrum Kenntnisse
Erfahrungen mit den Tools Git und Jira

Analyse des Aufgabenbereichs
Ergebnis:
Die Aufgabenbeschreibung ist klar, jedoch fehlt eine stärkere Fokussierung auf die Ergebnisse der Architekturentscheidungen und deren Auswirkungen auf das Projekt.
Alternative:
Definieren Sie konkrete Ergebnisse, wie z.B. 'Erstellung von Architekturdigrammen' oder 'Dokumentation der Softwarearchitektur'.

3. Aufgabenbereich

Entfernen

Allgemeine Aufgabendarstellung

Testen

Zeitanteil

20%

Detaillierte Beschreibung der Tätigkeit

Definition of Ready (DoR) und Definition of Done (DoD) – Erstellung

Hilfe

Zurück

Weiter

2. Automatisierte Generierung passender Schlüsselkompetenzen basierend auf der Stellenbeschreibung

Einleitung

Stellenprofil

Aufgabenbeschreibung

Schlüsselkompetenzen

Anforderungen

Prüfung

Schlüsselkompetenzen

Bitte bewerten Sie die folgenden Kompetenzbereiche nach ihrer Relevanz für Ihre Position und geben Sie an, welche Fähigkeiten in Ihrem Arbeitsumfeld besonders wichtig sind. Diese Kompetenzen umfassen intellektuelle/kognitive Fähigkeiten, soziale Kompetenzen und Persönlichkeitsmerkmale, die für eine effektive Führung und beruflichen Erfolg erforderlich sind. Nutzen Sie die Liste, um Prioritäten zu setzen und Schwerpunkte herauszustellen, soweit sie zwingend für die Stelle erforderlich sind.

📘 KI Unterstützung

Über den grünen Button in der Aktionsleiste kann automatisierte Generierung der Schlüsselkompetenzen durchgeführt werden. Dieser Schritt berücksichtigt die getätigten Angaben in diesem Anforderungsprofil und vergleicht diese mit anderen ähnlichen Profilen aus der Vergangenheit.

Intellektuelle / kognitive Kompetenz

Analytische Kompetenz	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Umsetzungskompetenz	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Lösungsorientierung	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Kreativität	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
+ Intellektuelle / kognitive Kompetenz hinzufügen			

Sozialkompetenz

Kooperationsfähigkeit	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Kommunikationsfähigkeit	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Teamfähigkeit	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️
Führungskompetenz	Kriteriumsart Musskriterium Vorkriterium	Relevanz 1 - sehr wichtig 2 3 4 5	🗑️

🔍 Hilfe

📝

✍️ Schlüsselkompetenzen generieren

← Zurück

Weiter →

✅ Generierung erfolgreich

Die Schlüsselkompetenzen wurden erfolgreich generiert.

3. KI-basierte Orientierungshilfe und Vorschläge zur objektiven Eingruppierung der Entgeltstufe mit ausführlicher Begründung

Einleitung

Stellenprofil

Aufgabenbeschreibung

Schlüsselkompetenzen

Anforderungen

Prüfung

Mögliche Entgeltstufe

EG12

Eingruppierung prüfen

Begründung der Einstufung

Niedrigere Entgeltstufe

Höhere Entgeltstufe

📘 Begründung der Einstufung

Die Einstufung in EG12 könnte gerechtfertigt sein, da die Stelle besondere Leistungen erfordert, die über die Anforderungen der EG11 hinausgehen. Die umfassende Verantwortung für die Softwarearchitektur, das Management komplexer Softwareprojekte sowie die Notwendigkeit, vertiefte Kenntnisse in verschiedenen Technologien und Methoden zu besitzen, deuten auf eine besondere Schwierigkeit und Bedeutung der Tätigkeit hin. Zudem wird eine mehrjährige praktische Erfahrung in der Softwareentwicklung gefordert, was die Einstufung in die EG12 unterstützt.

Kurze Prozessschritt-Beschreibung des Stellenmanager KI

Der KI-Stellenmanager nutzt einen mehrstufigen Prozess zur Erstellung und Analyse von Stellenbeschreibungen.

1. **Datengrundlage:** Für den Prototyp müssen bewertete Arbeitsplatzbeschreibungen vorliegen und Informationen wie Jobtitel, Arbeitsziel, Aufgabenbeschreibung, erforderliche Kompetenzen und Ausbildung enthalten. Diese Arbeitsplatzbeschreibungen wurden durch eine große Stadt in Baden-Württemberg zur Verfügung gestellt.
2. **Vektor-Datenbank:** Die Arbeitsplatzbeschreibungen werden in eine Vektor-Datenbank (Qdrant) eingebettet, um semantisch ähnliche Beschreibungen zu finden. Dieser Schritt nutzt lokale Embedding (Transformer)-Modelle.
3. **Formular zur Arbeitsplatzbeschreibung (Frontend):** Ein mehrstufiges Formular im Frontend führt Benutzer durch die Erstellung einer Arbeitsplatzbeschreibung. Es bietet Hilfestellungen und Beispiele die den Benutzer anleiten.
4. **LLM-Unterstützte Anwendungsfälle:** Große Sprachmodelle (LLMs) werden verwendet, um Aufgabenbeschreibungen zu analysieren und zu bewerten, Schlüsselkompetenzen zu generieren, sowie eine Entgelteinstufung zu empfehlen.
5. **Retrieval-Augmented Generation (RAG):** Bei Anfragen werden ähnliche Stellenbeschreibungen aus der Vektor-Datenbank abgerufen und als Kontext für die Anfrage verwendet.

Nutzen

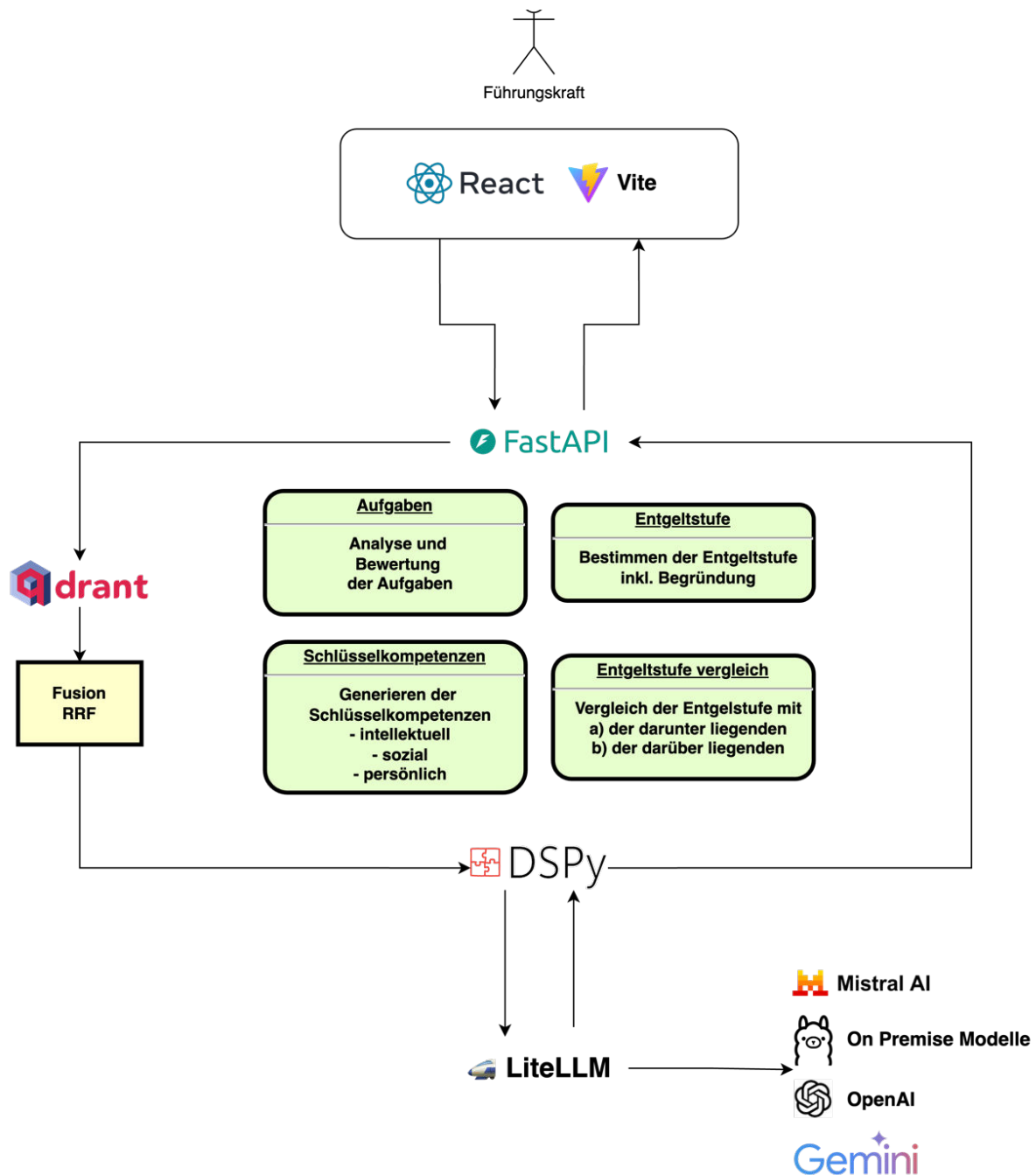
- Die KI soll die Erstellung von Arbeitsplatzbeschreibungen begleiten und durchführen. Schritt für Schritt, mit Hinweisen zum jedem Erstellungsschritt.
- Die KI soll eine transparente und nachvollziehbare Darstellung der Entgelteinstufung ermöglichen (Eingruppierung, Vergütungsniveau).
- Der KI-Prototyp soll flexibel anpassbar sein, damit eine Weiterentwicklung und eine breite Anwendbarkeit in Kommunen ermöglicht wird.
- Die erstellten Profile orientieren sich an den tarifrechtlichen Vorgaben.

Generelle Prototyp-Anforderungen

- Das Projekt umfasst die Entwicklung einer KI-basierten Lösung unter Einsatz geeigneter Programmiersprachen, die auf der Grundlage vorhandener Arbeitsplatzbeschreibungen über den spezifischen Kontext verfügt und über standardisierte Schnittstellen verfügen.
- Eine intuitive Benutzeroberfläche mit nutzerzentrierter Benutzerführung, die es auch nicht-technischen Benutzern eine einfache Bedienung ermöglicht.
- Der Prototyp wird als Open-Source-Lösung konzipiert, um Flexibilität, Anpassbarkeit und kontinuierliche Weiterentwicklung für verschiedene kommunale Kontexte zu ermöglichen
- Die KI-gestützten Bewertungen und Einstufungen sollen transparent und für alle Beteiligten nachvollziehbar dargestellt werden, mit klaren Begründungen für Vergütungsvorschläge und Eingruppierungen.

Ausblick 2025

Bereitstellung des Prototyps als Open Source samt Dokumentation



Darstellungen von Konzeption und Entwicklung

Dieser Prototyp verwendet den Vector Store der Firma Qdrant. Qdrant ist eine Open-Source Vektor-Datenbank, geschrieben in Rust, und bietet eine performante und skalierbare Lösung für die Speicherung und Suche von Vektoren. Das Unternehmen hinter Qdrant ist ein deutsches Startup mit Sitz in Berlin.

Ein wichtiges Designprinzip dieser Anwendung ist die dynamische Nutzung und Austauschbarkeit von LLMs. Dies stellt sicher, dass die Anwendung zukunftssicher bleibt und neue Generationen von Sprachmodellen problemlos integrieren kann, sobald diese verfügbar sind. Durch die Ermöglichung der dynamischen Modellauswahl und die Bereitstellung einer Abstraktionsschicht ist die Anwendung zukunftssicher konzipiert. Wenn neue und verbesserte LLMs verfügbar werden, können diese einfach in die Anwendung integriert werden.

Die Integration erfolgt über das Framework LiteLLM. LiteLLM abstrahiert die Unterschiede zwischen verschiedenen LLM-Anbietern und bietet eine einheitliche API für den Zugriff auf Modelle wie OpenAI, Azure OpenAI, Cohere, Anthropic usw. Dies vereinfacht den Austausch und die Nutzung verschiedener Modelle erheblich, da keine spezifischen Client-Bibliotheken oder API-Schlüssel für jeden Anbieter benötigt werden. LiteLLM unterstützt auch Funktionen wie automatische Wiederholungsversuche, Protokollierung und Kostenverfolgung.

Hinzu kommt das Framework DSPy, das von der Stanford University veröffentlicht wurde. DSPy (Declarative Self-improving Language Programs) ist ein Framework, das die Entwicklung von Anwendungen mit LLMs vereinfacht, indem es eine deklarative Art der Programmierung mit LLMs ermöglicht. Anstatt LLMs direkt zu programmieren, beschreibt man in DSPy die gewünschte Funktionalität und überlässt es dem Framework, die optimale Konfiguration der LLMs zu finden. DSPy unterstützt Techniken wie Prompt Engineering, Chain-of-Thought Reasoning und Knowledge Retrieval.

Die Anwendung ist als RESTful API konzipiert. Dies ermöglicht eine einfache Integration mit verschiedenen Clients und Systemen. Durch die Einhaltung der REST-Prinzipien können neue Anwendungsfälle einfach hinzugefügt werden, ohne bestehende Funktionalitäten zu beeinträchtigen. Neue Endpunkte können hinzugefügt werden, um zusätzliche Funktionen bereitzustellen, ohne die bestehende API-Struktur zu verändern.

Kontaktieren Sie uns gerne für Ideen, Feedback und Einbindung in die Entwicklung.

Ihr KommHUB Team.

KI Stellenmanager

Dieses Repository ist Teil eines Forschungsprojektes der Komm.ONE in Kooperation mit der Landeshauptstadt Stuttgart sowie der Entwickler-Firma YOUniquehorns GmbH und umfasst einen Prototyp zum Einsatz von Sprachmodellen im Personalbereich bzw. im Prozess der Arbeitsplatzbeschreibung.

In a Nutshell

- 📄 Automatisierte Prüfung und Einstufung (inkl. Begründung) von Arbeitsplatzbeschreibungen unter Verwendung von Sprachmodellen
- 🔍 Vektorbasierte Ähnlichkeitssuche für ähnliche Arbeitsplatzbeschreibungen
- 🌐 [FastAPI-Backend](https://github.com/fastapi/fastapi) mit Swagger-Dokumentation
- 🎨 React-Frontend ([Chakra-UI](https://github.com/chakra-ui/chakra-ui)) und [vite.js](https://github.com/vitejs/vite)

Technologien

- ****Sprachmodelle****: Unterstützt alle gängigen Sprachmodelle sowie die Provider durch [LiteLLM](https://docs.litellm.ai/docs/providers)

Getestet wurde der Prototyp mit OpenAI's `GPT-4o-mini` Modell in der Microsoft Azure Umgebung und Mistral AI's `mistral-large-2407`
- ****Embedding Modelle****: Verwendung von [FastEmbed](https://qdrant.github.io/fastembed/)
 - Dense Vektoren: [jinaai/jina-embeddings-v2-base-de](https://huggingface.co/jinaai/jina-embeddings-v2-base-de), das explizit für Deutsche und Englische Texte trainiert wurde und eine Kontext-Länge von über 8000 Tokens verfügt.
 - Sparse Vektoren: [Qdrant/bm25](https://huggingface.co/Qdrant/bm25), das eine breite Liste von Sprachen unterstützt.
- ****Vektor Datenbank****: [Qdrant](https://qdrant.tech/qdrant-vector-database/) (Open Source, deutsches Startup aus Berlin)
- ****Backend****: Python Version: 3.12
- ****Frontend****: TypeScript Version: 5.6

Prototyp

Kurzbeschreibung - Funktionalitäten

Der KI Stellenmanager ist darauf ausgerichtet, Prozesse rund um die Arbeitsplatzbeschreibung substantiell zu beschleunigen und qualitativ zu verbessern. Im Zentrum steht die Prüfung, wie KI in der Personalverwaltung eingesetzt werden kann, um transparentere, fairere und effizientere Prozesse zu realisieren.

Die Integration fortschrittlicher Sprachmodelle ermöglicht:

- Automatisierte Generierung von strukturierten Aufgabenspezifikationen basierend auf dem Jobtitel und Arbeitsziel
- Detaillierte und strukturierte Evaluierung der beschriebenen Aufgaben und deren Anforderungsniveau

![Analyse_Aufgaben](frontend/docs/imgs/Analyse_Aufgaben.png)

- Automatisierte Generierung passender Schlüsselkompetenzen basierend auf der Stellenbeschreibung

![Generierung_Kompetenzen](frontend/docs/imgs/Generierung_Kompetenzen.png)

- KI-basierte Orientierungshilfe und Vorschläge zur objektiven Eingruppierung der Entgeltstufe mit ausführlicher Begründung

![Einstufung_Entgeltstufe](frontend/docs/imgs/Einstufung_Entgeltstufe.png)

Darüber hinaus bietet der Prototyp ein webbasiertes Formular für die strukturierte Erstellung von Arbeitsplatzbeschreibungen, das traditionelle Word-Dokumente ersetzt. Diese Digitalisierung ermöglicht es zu evaluieren, wie der gesamte Prozess effizienter, schneller und transparenter gestaltet werden kann.

Die technische Umsetzung erfolgt durch einen Retrieval Augmented Generation (RAG) Ansatz, der bestehende, bereits bewertete Arbeitsplatzbeschreibungen als Referenzdatenbank integriert. Dies gewährleistet, dass die KI-gestützten Vorschläge und Bewertungen auf fundierten Vergleichsdaten basieren.

Nutzen

Der KI Stellenmanager bietet signifikante Vorteile für kommunale Personalverwaltungen:

- **Prozessoptimierung**: Deutliche Reduzierung des Zeit- und Arbeitsaufwands bei der Erstellung und Bewertung von Stellenbeschreibungen
- **Qualitätssteigerung**: Konsistente und qualitativ hochwertige Arbeitsplatzbeschreibungen durch standardisierte KI-Unterstützung
- **Transparenz**: Nachvollziehbare und gut dokumentierte Entscheidungsprozesse bei der Eingruppierung
- **Wissenssicherung**: Systematische Erfassung und Nutzung von Erfahrungswissen für neue Stellenbewertungen
- **Rechtssicherheit**: Unterstützung bei der Einhaltung tarifrechtlicher Vorgaben und Gleichbehandlungsgrundsätze

Als anpassungsfähige Open-Source-Lösung unterstützt der KI Stellenmanager das kontinuierliche Lernen und die Kompetenzentwicklung im kommunalen Umfeld durch den verantwortungsvollen Einsatz künstlicher Intelligenz im Personalbereich.

Generelle Prototyp-Anforderungen

- **Transparenz und Nachvollziehbarkeit**: Die KI-gestützten Bewertungen und Einstufungen müssen transparent und für alle Beteiligten nachvollziehbar dargestellt werden, mit klaren Begründungen für Vergütungsvorschläge und Eingruppierungen.
- **Open-Source-Entwicklung**: Der Prototyp wird als Open-Source-Lösung konzipiert, um Flexibilität, Anpassbarkeit und kontinuierliche Weiterentwicklung für verschiedene kommunale Kontexte zu ermöglichen.
- **Tarifrechtliche Konformität**: Alle erstellten Arbeitsplatzbeschreibungen und -bewertungen entsprechen den geltenden tarifrechtlichen Vorgaben und unterstützen deren korrekte Anwendung.
- **Datenschutzkonformität**: Strikte Einhaltung datenschutzrechtlicher Bestimmungen mit besonderem Fokus auf den Schutz personenbezogener Daten im öffentlichen Dienst.

Ausblick

- **Erweiterung der KI-Anwendungsfälle**: Integration zusätzlicher Analysemodule für die Prüfung der Kohärenz zwischen Aufgaben, Zielen und Anforderungen sowie automatisierte Identifikation von Inkonsistenzen und Optimierungspotentialen.
- **Personalisiertes Session-Management**: Implementierung eines sicheren Benutzermanagements zur Speicherung und Verwaltung von Arbeitsplatzbeschreibungen mit differenzierten Zugriffsrechten.
- **Automatisierte Stellenausschreibungen**: Entwicklung einer Funktion zur automatisierten Generierung vollständiger Stellenausschreibungstexte aus den vorhandenen Arbeitsplatzbeschreibungen unter Berücksichtigung von Zielgruppenspezifika.

- ****Integrierte Workflow-Steuerung****: Abbildung des gesamten Prozesses von der Initiierung über Freigaben bis zur finalen Veröffentlichung der Stellenbeschreibung mit automatisierten Statusaktualisierungen.

Entwicklung

Projekt Struktur

```markdown

stellenmanager/

```
├── api/ # Python Backend (FastAPI)
| ├── src/ # Quellcode
| | ├── api/ # API bezogen
| | ├── core/ # Kernfunktionalitäten
| | ├── scripts/ # Setup und Hilfs-Skripte
| | └── main.py # API Startpunkt
| ├── docs/ # Dokumentation
| ├── tests/ # Backend Tests
| └── pyproject.toml # Backend Abhängigkeiten
├── frontend/ # React Frontend mit vite (Chakra-UI)
| ├── src/ # React Quellcode
| ├── public/ # Statische Inhalte
| ├── docs/ # Dokumentation
| ├── package.json # Frontend Abhängigkeiten
| └── tsconfig.json # TypeScript Konfiguration
└── docker/ # Docker Dateien
```

...

### ### Setup

1. Python FastAPI Backend aufsetzen: [\[LINK\]\(./api/README.md\)](#)

2. TypeScript Frontend aufsetzen: [\[LINK\]\(./frontend/README.md\)](#)

## ## Lizenz & Sicherheit

- Lizenziert unter der MIT Lizenz - siehe [\[Lizenz\]\(License\)](#)
- Sicherheitsprobleme bitte direkt an die Maintainer melden