CareerVision

Final Presentation Group F

Daniel Rüdiger, Anton Averianov, Chingiz Kuanyshbay, Maciej Lubinski

Motivation: Helping professionals in their search for matching job positions and professional development

Problem

- In-transparent job market[1]
- Millions of job offers
- Manual search process
- *'Application spam'*

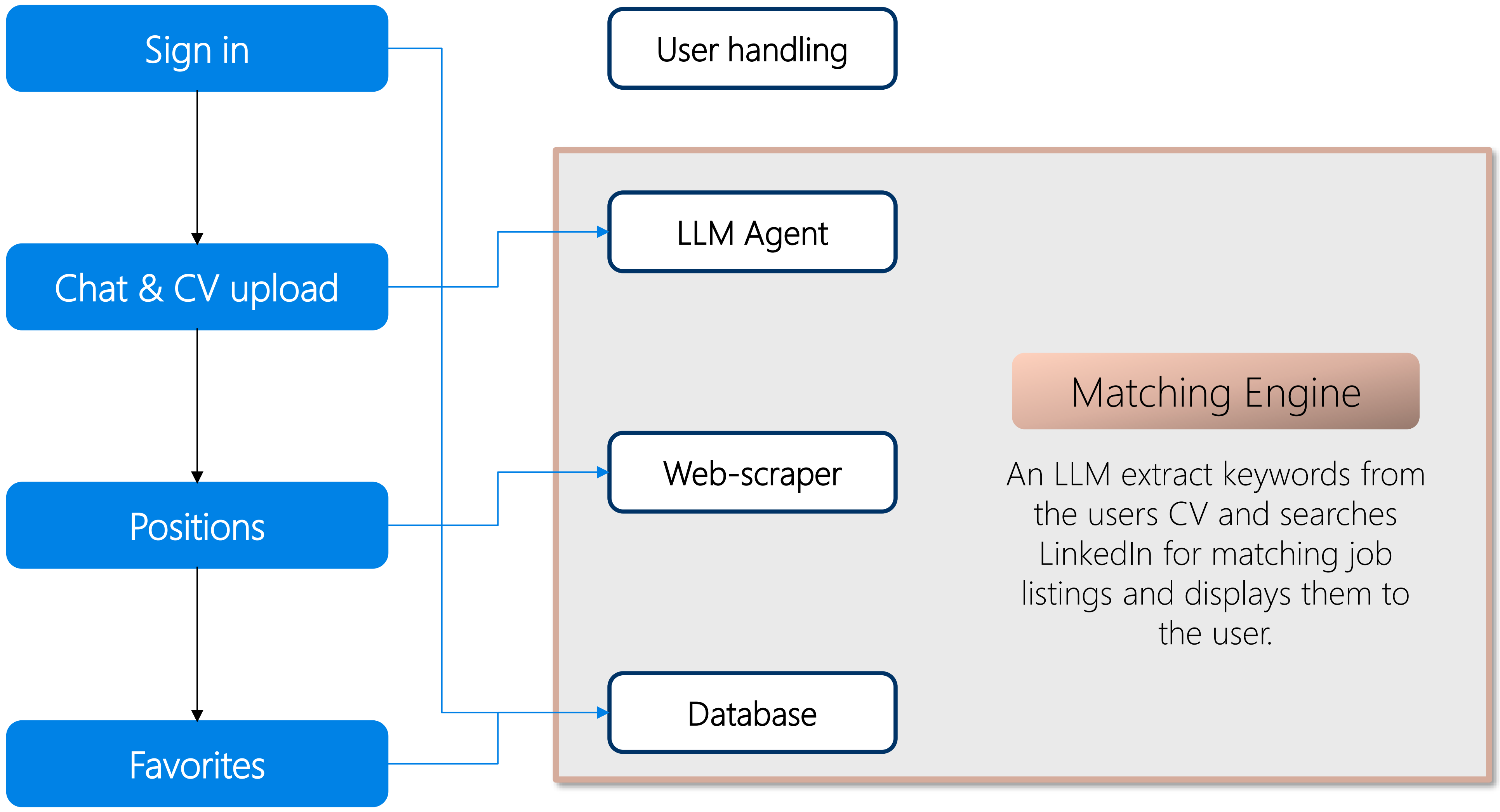
Solution: CareerVision

- Increase transparency
- Narrow down job offers[2]
- Semi-automatic search
- Targeted applications

- Career professionals looking to transition
- Explore new job opportunities
- Searching for well matching job offers

Sources: [1]Artificial intelligence and labour market matching – OECD, [2]Use Cases For Generative AI In Workforce Development – Jobs of the Future,

User-Journey and functionality break-down, the elements that make up CareerVision



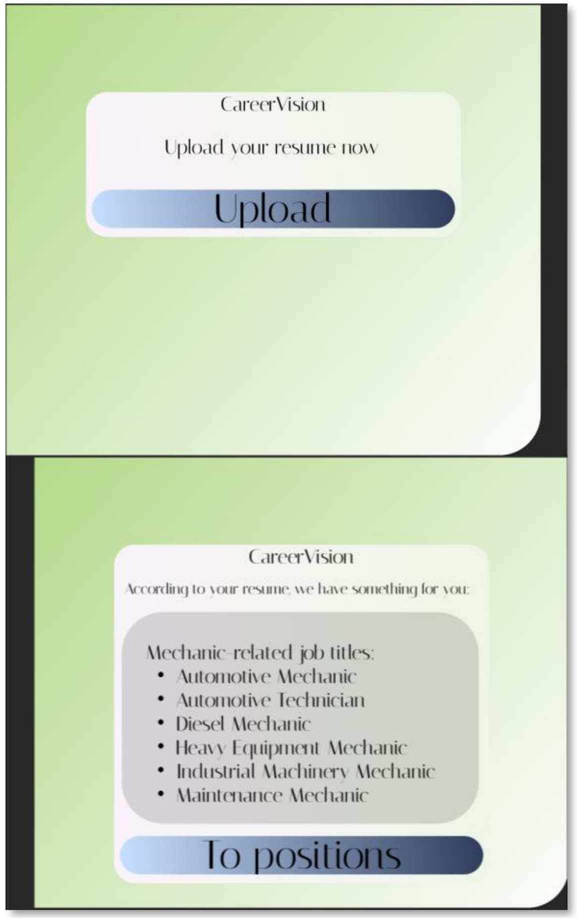
An LLM extract keywords from the users CV and searches LinkedIn for matching job listings and displays them to the user.

Technology Front-End: The tools, libraries and API we used to get our project to run

Tool

- Figma
- Basic HTML
- JavaScript & CSS
- Translations
- Connecting Back and Front

Iterating designs and layouts in Figma



CareerVision

Upload your resume now

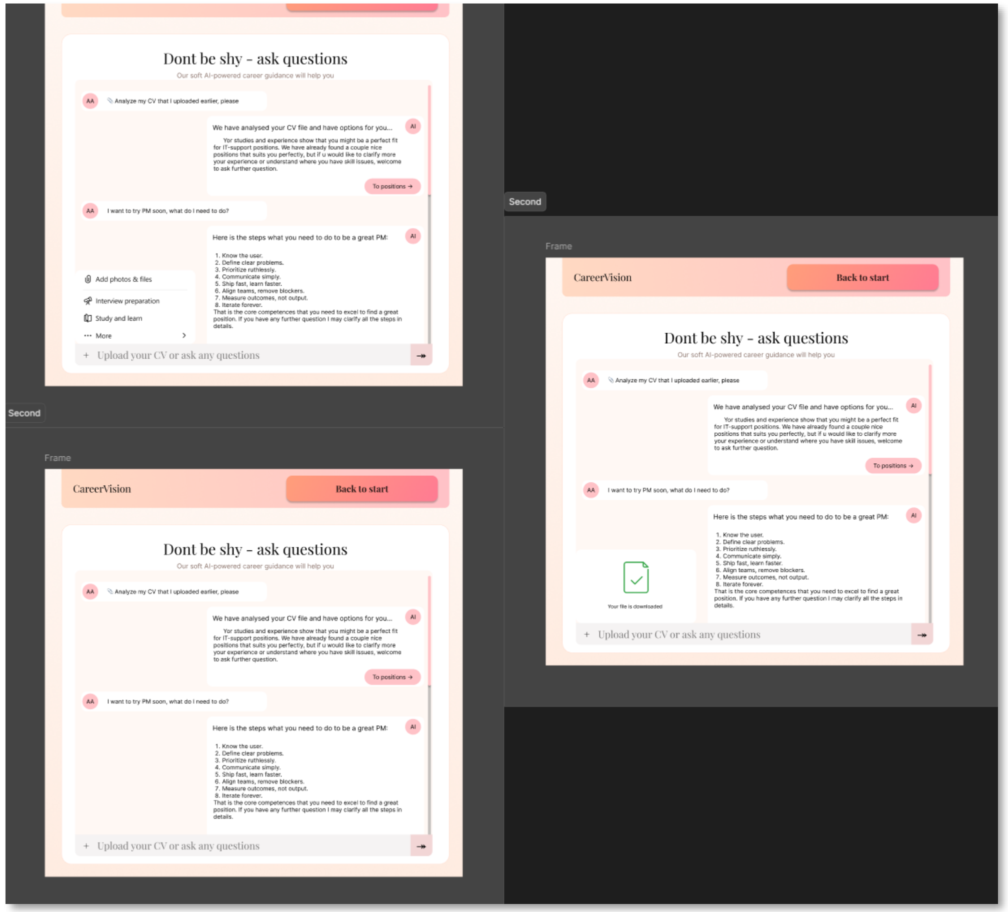
Upload

CareerVision

According to your resume, we have something for you:

Mechanic-related job titles:
- Automotive Mechanic
- Automotive Technician
- Diesel Mechanic
- Heavy Equipment Mechanic
- Industrial Machinery Mechanic
- Maintenance Mechanic

To positions



CareerVision | Back to start

Dont be shy - ask questions

Our soft AI-powered career guidance will help you

AA: Analyze my CV that I uploaded earlier, please

AI: We have analysed your CV file and have options for you...

Yor studies and experience show that you might be a perfect fit for IT-support positions. We have already found a couple nice positions that suits you perfectly, but if u would like to clarify more your experience or understand where you have skill issues, welcome to ask further question.

To positions →

AA: I want to try PM soon, what do I need to do?

AI: Here is the steps what you need to do to be a great PM:

1. Know the user.
2. Define clear problems.
3. Prioritize ruthlessly.
4. Communicate simply.
5. Ship fast, learn faster.
6. Align teams, remove blockers.
7. Measure outcomes, not output.
8. Iterate forever.

That is the core competences that you need to excel to find a great position. If you have any further question I may clarify all the steps in details.

Technology Front-End: The tools, libraries and API we used to get our project to run

Tool

Figma

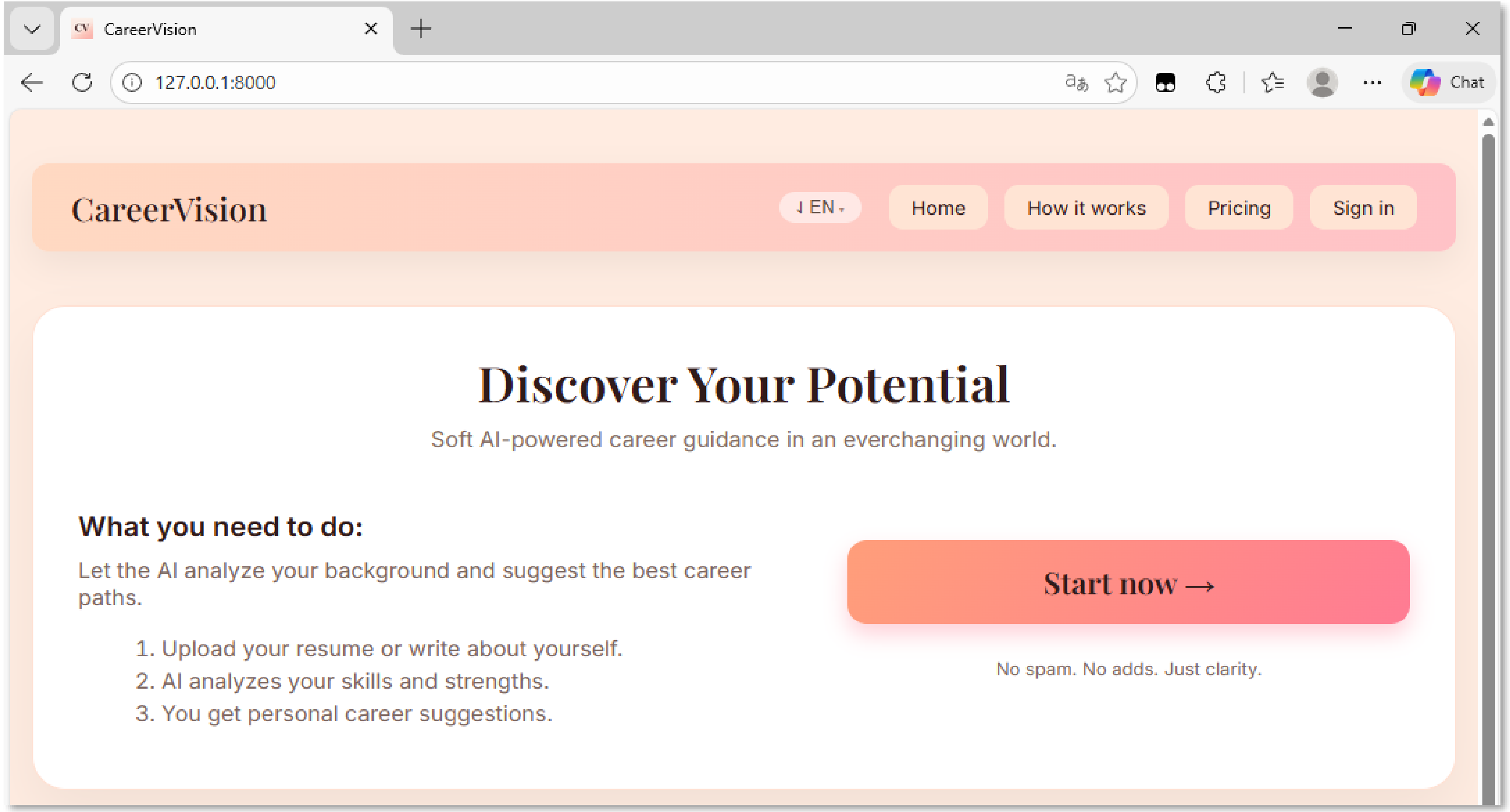
Basic HTML

JavaScript & CSS

Translations

Connecting Back and Front

Transferring Figma designs to HTML code



CareerVision

127.0.0.1:8000

CareerVision

EN

Home

How it works

Pricing

Sign in

Discover Your Potential

Soft AI-powered career guidance in an everchanging world.

What you need to do:

Let the AI analyze your background and suggest the best career paths.

1. Upload your resume or write about yourself.
2. AI analyzes your skills and strengths.
3. You get personal career suggestions.

Start now →

No spam. No adds. Just clarity.

Technology Front-End: The tools, libraries and API we used to get our project to run

Tool

- Figma
- Basic HTML
- **JavaScript & CSS**
- Translations
- Connecting Back and Front

Implementing layout and direct user interactions with JavaScript

Layout & Structure

- Fixed-height layout to prevent UI jumping during streaming
- Flexbox for alignment and spacing
- Responsive message bubbles with max-width for readability

Styling & UX

- Gradient buttons for visual emphasis
- Smooth CSS transitions for state changes
- Consistent color scheme across the interface

Dynamic Behavior

- Streaming text animation using timed intervals
- Auto-scroll to keep the latest messages visible
- Dynamic DOM manipulation (elements created on the fly)
- Conditional rendering based on application state

Technology Front-End: The tools, libraries and API we used to get our project to run

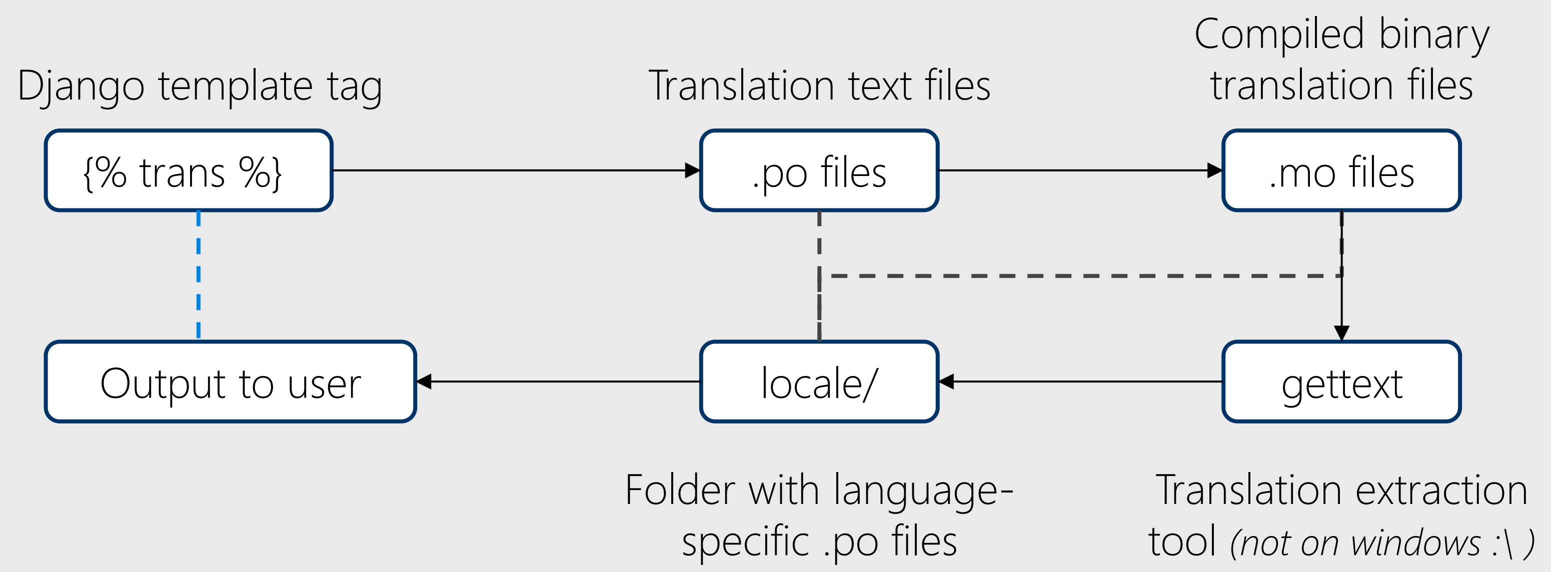
Tool

- Figma
- Basic HTML
- JavaScript & CSS
- **Translations**
- Connecting Back and Front

Language accessibility through Django i18n internationalisation

Overview of i18n - Internationalization

LANGUAGE_CODE - Django setting (settings.py)


Django template tag
{% trans %}
Translation text files
.po files
Compiled binary translation files
.mo files
Output to user
locale/
gettext
Folder with language-specific .po files
Translation extraction tool *(not on windows :\ )*

Technology Front-End: The tools, libraries and API we used to get our project to run

Tool

- Figma
- Basic HTML
- JavaScript & CSS
- Translations
- Connecting Back and Front

API Endpoints that connect front-end to back-end databases and features

$$\begin{tikzcd}\text{Front - localhost:8001} \arrow[r] & \text{Back - localhost:8000} \arrow[r] & \text{LLM - localhost:8002}\end{tikzcd}$$

User Auth

- User login
- User registration
- Access token requests
- Backend returns tokens
- Frontend stores tokens

Matching Engine

CV
- Upload CV PDF
- Fetch analysis

Jobs
- Select city
- Fetch job lists
- Display results

Favorites
- Add job favorite
- View favorites list
- Remove favorite

Chat

- Load chat history
- Send user message
- Combine CV + chat
- LLM generates reply
- Save messages

Clear Chat
- Clear conversation
- Delete chat history
- Reset chat UI

Technology Back-End: The tools, libraries and API we used to get our project to run

LLM Agent

QWEN2.5 3B Instruct: Chat & document analysis

- Run locally through GGUF format
- ~3 billion parameters (lightweight, fast) Large Language Model
- Different instruction configurations

Web-scraper

LinkedIn Voyager API (scraping data)

- internal API that its own web app uses to fetch data like jobs, profiles, messages, and search results

Uniform Resource Name (URN): To lable and identify job listings

User Auth

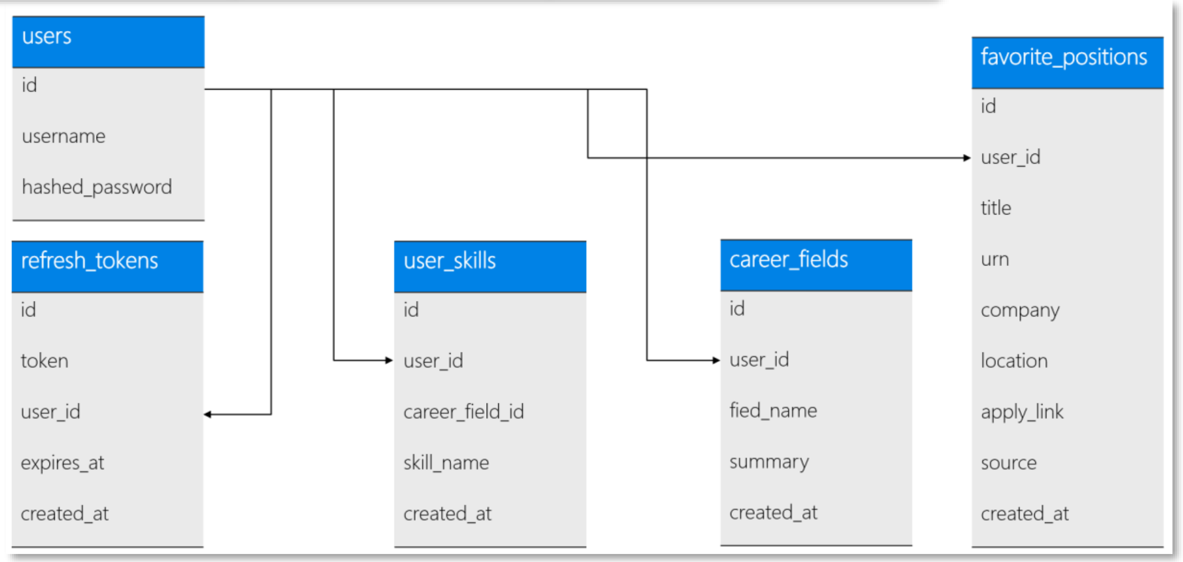
JSON Web Token (JWT): secure user information handling
signed token used to prove identity between a client and a server to manage access

Database

PostgreSQL: Handle user data extracted from CV and job listing scraped from LinkedIn

SQLAlchemy (Python library) to build the database in python, easier interaction with SQL, safer, cleaner

Diagram of resulting Database



Technology Back-End: The tools, libraries and API we used to get our project to run

LLM Agent

QWEN2.5 3B Instruct: Chat & document analysis

- Run locally through GGUF format
- ~3 billion parameters (lightweight, fast) Large Language Model
- Different instruction configurations

Web-scraper

LinkedIn Voyager API (scraping data)

- internal API that its own web app uses to fetch data like jobs, profiles, messages, and search results

Uniform Resource Name (URN): To lable and identify job listings

User Auth

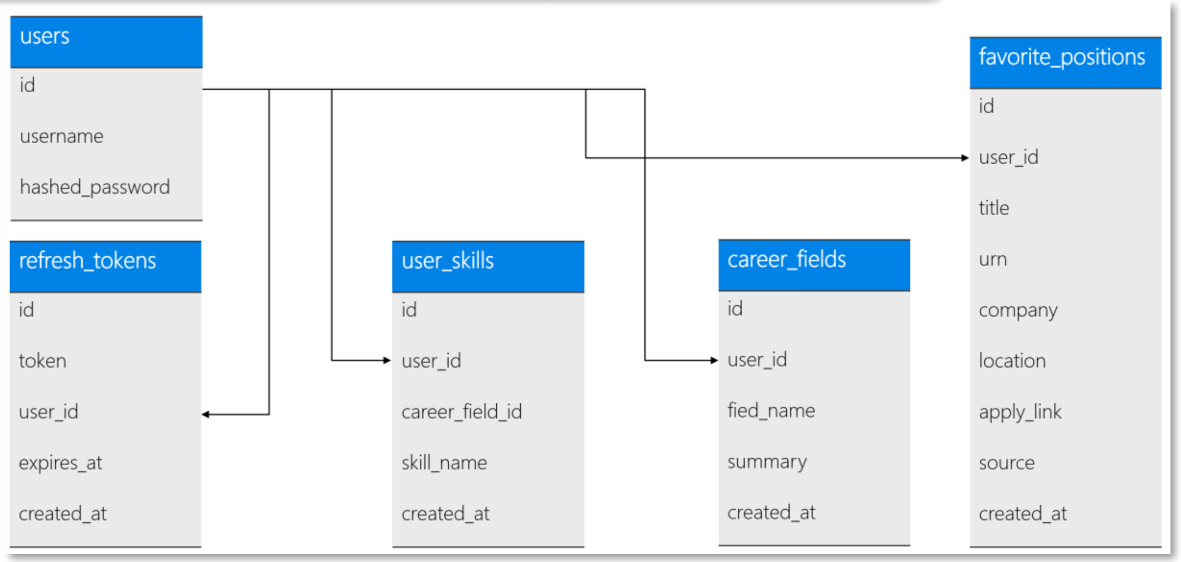
JSON Web Token (JWT): secure user information handling
signed token used to prove identity between a client and a server to manage access

Database

PostgreSQL: Handle user data extracted from CV and job listing scraped from LinkedIn

SQLAlchemy (Python library) to build the database in python, easier interaction with SQL, safer, cleaner

Diagram of resulting Database



Technology Back-End: The tools, libraries and API we used to get our project to run

LLM Agent

QWEN2.5 3B Instruct: Chat & document analysis

- Run locally through GGUF format
- ~3 billion parameters (lightweight, fast) Large Language Model
- Different instruction configurations

Web-scraper

LinkedIn Voyager API (scraping data)

- internal API that its own web app uses to fetch data like jobs, profiles, messages, and search results

Uniform Resource Name (URN): To lable and identify job listings

User Auth

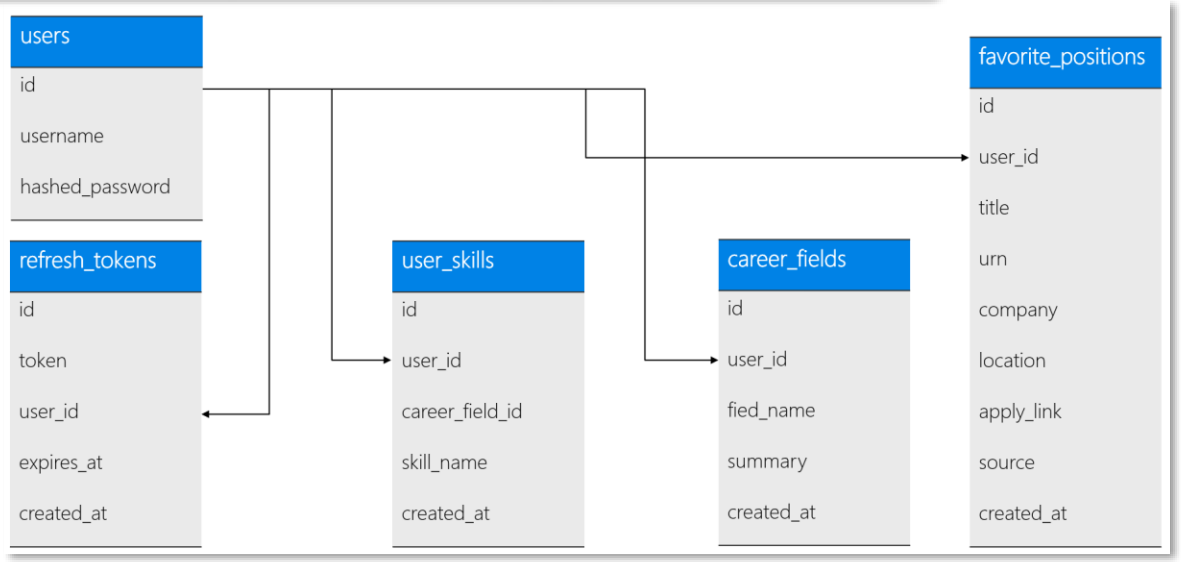
JSON Web Token (JWT): secure user information handling signed token used to prove identity between a client and a server to manage access

Database

PostgreSQL: Handle user data extracted from CV and job listing scraped from LinkedIn

SQLAlchemy (Python library) to build the database in python, easier interaction with SQL, safer, cleaner

Diagram of resulting Database



Technology Back-End: The tools, libraries and API we used to get our project to run

LLM Agent

QWEN2.5 3B Instruct: Chat & document analysis
- Run locally through GGUF format
- ~3 billion parameters (lightweight, fast) Large Language Model
- Different instruction configurations

Web-scraper

LinkedIn Voyager API (scraping data)
- internal API that its own web app uses to fetch data like jobs, profiles, messages, and search results

Uniform Resource Name (URN): To lable and identify job listings

User Auth

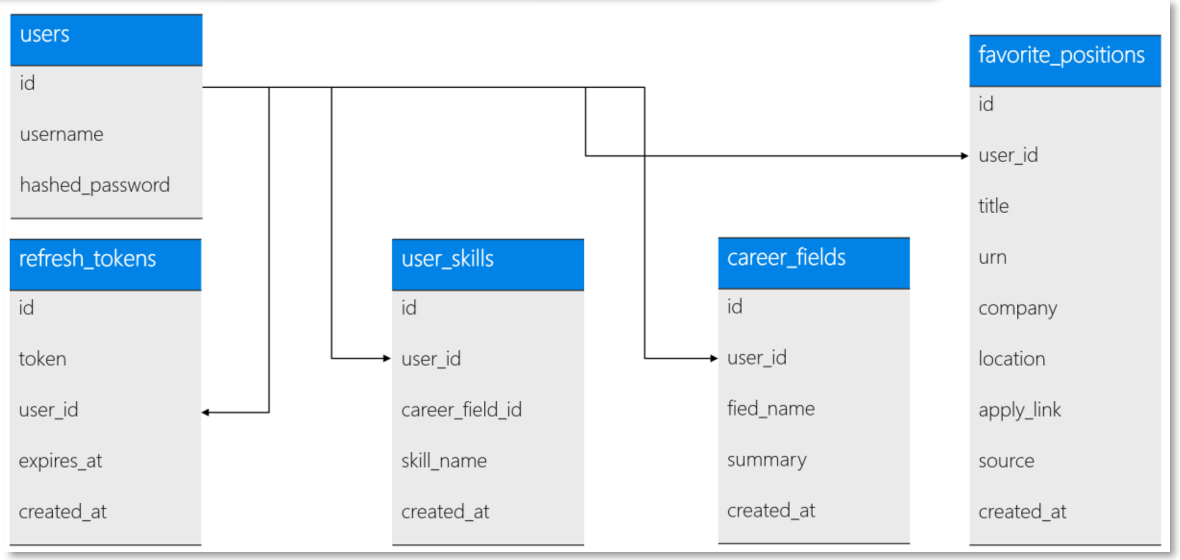
JSON Web Token (JWT): secure user information handling
signed token used to prove identity between a client and a server to manage access

Database

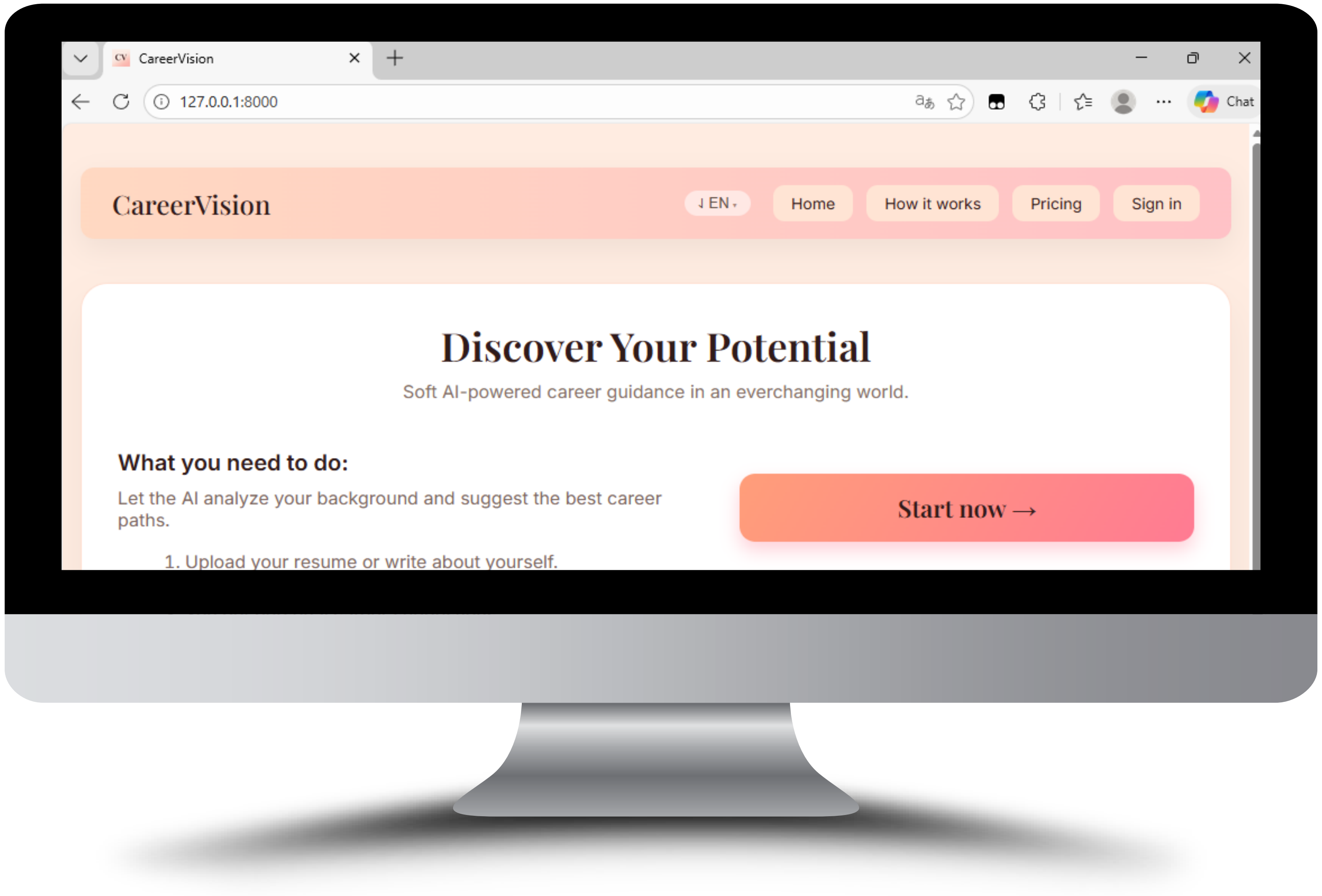
PostgreSQL: Handle user data extracted from CV and job listing scraped from LinkedIn

SQLAlchemy (Python library) to build the database in python, easier interaction with SQL, safer, cleaner

Diagram of resulting Database



Project demo: CareerVision in action



CareerVision
127.0.0.1:8000

CareerVision
EN Home How it works Pricing Sign in

Discover Your Potential

Soft AI-powered career guidance in an everchanging world.

What you need to do:

Let the AI analyze your background and suggest the best career paths.

1. Upload your resume or write about yourself.

Start now →

Thank you!

Questions?

If you have feedback, feature ideas or just want to say hi drop a line to:

anton.averianov@tum.de
daniel.Ruediger@tum.de