CV Scanner

TECHNICAL INSTALLATION MANUAL

Version: 3.0.0

Date: August 20th, 2025

Status: Final

Document Type: Technical Installation Manual

Project: CV Scanner - Automated CV Analysis System

Team Members

Marcelo Parsotam

Unaisah Hassim

Talhah Karodia

Abdullah Pochee

Ronan Smart

Table of Contents

Introduction

This system is a full-stack CV scanning app composed of:

- Frontend (UI): React + TypeScript (Vite) in UI/CV-Scanner. Interacts with the API over HTTP.
- Backend (API): Spring Boot 3 (Java) in API/api. Persists data to SQL Server and exposes endpoints for saving, listing, and stats.
- AI Model (AI): Python service in AI that parses uploaded CVs and returns structured JSON to the API.
- Database: Microsoft Azure SQL Server storing candidates and CV scans.

Prerequisites Install these tools (recommended versions):

- Git: 2.46+
- Node.js: 20.x LTS and npm 10.x
- Java: 21 LTS (Spring Boot 3.5 requires Java 17+; 21 recommended)
- Maven: 3.9.x
- Python: 3.12.x (64-bit)
- SQL Server: 2019/2022 Developer or Express
- Postman (optional, for API testing)
- Visual Studio Code (optional, for development)

Pre-Requisites

- Prerequisites: Git 2.46+ for cloning
- Node.js 20.x LTS with npm 10.x for the React frontend
- Java 21 (JDK) and Maven 3.9.x for the Spring Boot API
- Python 3.12.x (64-bit) with pip for the AI service (use a virtual environment)
- Microsoft SQL Server 2019/2022 (Developer/Express) with a reachable instance and credentials;
- A modern browser (Chrome/Edge) and Postman (optional) for API testing;
- A code editor like Visual Studio Code.

Important

Ensure JAVA_HOME points to JDK 21, Node and Python are on PATH, and SQL Server is running and accessible (default port 1433).

Install Commands (Ubuntu Terminal)

```
sudo apt update
sudo apt install -y git curl python3.12 python3.12-venv maven
# Node 20
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs
# Java 21 (Temurin)
sudo apt install -y wget apt-transport-https
```

wget -O- https://packages.adoptium.net/artifactory/api/gpg/key/public | sudo tee /etc/apt/trusted.gpg.d/adoptium.asc

echo "deb https://packages.adoptium.net/artifactory/deb $(. /etc/os-release; echo $VERSION_CODENAME) main" | sudo tee /etc/apt/sources.list.d/adoptium.list

sudo apt update && sudo apt install -y temurin-21-jdk

Run AI

Open a new Terminal in Ubuntu

cd AI

python3 -m venv .venv

activate the venv (WSL / Bash)

source .venv/bin/activate

pip install -U pip

pip install -r requirements.txt

Run Uvicorn against the ASGI wrapper exported as `asgi_app`

uvicorn app:asgi_app --host 0.0.0.0 --port 5000 --log-level debug

Run UI

Open a new Terminal in Ubuntu

cd UI/CV-Scanner

npm install

npm run dev

Run API

Open a new Terminal in Ubuntu

cd API/api

mvn spring-boot:run

Running Project

In the terminal of the UI a display with:

➜ Local: http://localhost:3000/

Will appear

Click on the link and the project will run, you will now have access to the system and should be fully functional.

If you'd like to look at the User manual to show how the system can be used, you can find it on the GitHub page.

Or by clicking on the link below

https://drive.google.com/file/d/1Drewau_SXilGcay2Elxx9il7a3aqcbnF/view?usp=sharing

Running Deployment Project

Run this link in your browser and have access to the project

https://jolly-bay-0e45d8b03.2.azurestaticapps.net/

If the AI fails, run locally in terminal from main branch

Open a new Terminal in Ubuntu
cd AI

python3 -m venv .venv

```
# activate the venv (WSL / Bash)
source .venv/bin/activate
pip install -U pip
pip install -r requirements.txt
# Run Uvicorn against the ASGI wrapper exported as `asgi_app`
uvicorn app:asgi_app --host 0.0.0.0 --port 5000 --log-level debug
```