

# Software Design Document Job Finder

## 1. Úvod

- JobFinder je webová aplikace, která pomáhá firmám najít vhodné kandidáty a uživatelům porovnat volná pracovní místa na základě životopisů s využitím umělé inteligence. Zahrnuje administraci aplikace

## 2. Obecný popis

### • 2.1 Funkce systému

- Uživatel: registrace, přihlášení k účtu, zobrazení editace údajů, zobrazení všech volných míst v aplikaci (vyhledávání, filtrování), analýza životopisů, získání výsledků a uložení výsledků analýzy, získání vhodných volných míst (vyhledávání, filtrování), odpověď na volné místo, přidání volného místa do seznamu oblíbených volných míst, zobrazení seznamu oblíbených volných míst, zobrazení stavu odpovědi na volné místo.
- Firma: Registrace, přihlášení k účtu, zobrazení a úprava údajů, vytvoření volného pracovního místa, zobrazení vytvořených volných pracovních míst, zobrazení uživatelů, kteří reagovali na volné pracovní místo, reakce na odpověď uživatele na volné pracovní místo.
- Administrátor: Přihlášení k účtu, prohlížení a úprava údajů, prohlížení a mazání všech uživatelů, prohlížení a mazání všech firem, vytváření a prohlížení všech technologií.

### • 2.2 Omezení

Každý registrovaný uživatel má svou vlastní roli (ROLE\_USER, ROLE\_COMPANY, ROLE\_ADMIN), která mu umožňuje přístup ke konkrétním funkcím aplikace, které jsou popsány výše. Neregistrovaní uživatelé si mohou prohlížet pouze volná pracovní místa.

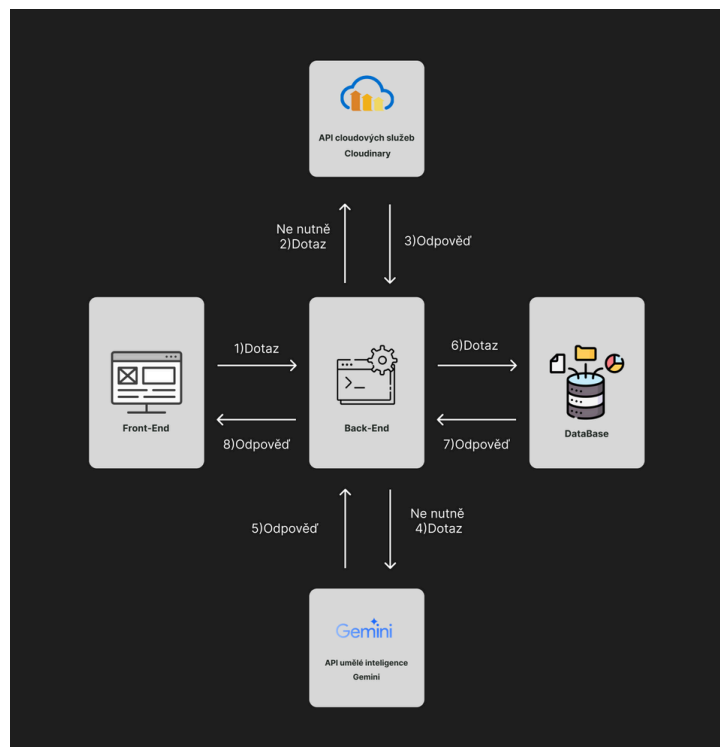
- **2.3 Závislosti**

Aplikace využívá API cloudové služby Cloudinary pro ukládání souborů, ve kterých jsou uloženy životopisy uživatelů, Google Gemini API (2.0 flash) pro analýzu životopisů, Gmail mycompanyspr@gmail.com pro odesílání dopisů s ověřovacím kódem.

## 3. Architektura systému

### • 3.1 Architektonický diagram

Popis architektury, frontend odešle požadavek na backend v závislosti na požadavku backend přistupuje buď ke cloudové službě a poté k API neuronové sítě, nebo přímo k databázi, poté obdrží odpověď z databáze, zpracuje ji a odešle frontendu.



### • 3.2 Použité technologie

- **Front-End:** (React JS, Vite, Redux, Axios, React-router, Tailwind)
- **Back-End:** (Java 17, SpringBoot, jsonwebtoken, JPA, Spring-mail-sender, ApachePDFbox, Cloudinary API)
- **DataBase:** PostgreSQL

- **Hostování aplikací:** Railway.

## 4. Návrh frontendu

- Struktura složek

jobfinder-frontend/

——	eslint.config.js	# Konfigurace ESLintu (kontrola kódu)
——	index.html	# Výchozí HTML šablona
——	package.json	# Závislosti a skripty projektu
——	vite.config.js	# Konfigurace Vite (build tool)
——	public/	# Veřejně dostupné statické soubory
——	fonts/	# Použitá písma
——	icons/	# Ikony
——	socials/	# Ikony sociálních sítí
——	src/	# Zdrojový kód aplikace
——	App.jsx	# Hlavní komponenta aplikace
——	index.jsx	# Vstupní bod React aplikace
——	components/	# Opakovatelné komponenty podle stránek/funkcí
——	admin-*/	# Komponenty pro administrátorské rozhraní
——	company-*/	# Komponenty pro firemní rozhraní
——	user-*/	# Komponenty pro uživatelské rozhraní
——	Global/	# Sdílené komponenty (např. Header, Footer)
——	all-vacancies-page/	# Komponenty pro zobrazení všech nabídek
——	analyze-cv-page/	# Komponenty pro analýzu CV
——	job-applications-page/	# Přihlášky na nabídky práce
——	my-account/	# Nastavení účtu pro uživatele/firmy
——	vacancy-details-page/	# Detailní pohled na pracovní nabídku
——	pages/	# Stránky aplikace podle rolí
——	MyAccountPage.jsx	# Stránka uživatelského účtu

— admin/	# Admin stránky
— Company/	# Firemní stránky
— public/	# Veřejně dostupné stránky (login, registrace atd.)
— user/	# Uživatelské stránky
— router/	# Ochrana tras podle role (Route Guards)
— services/	# Volání API pro různé role (admin, firma, uživatel)
— store/	# Redux store a slice pro stav aplikace
— styles/	# CSS styly
— utils/	# Pomocné funkce (např. dekodování JWT)

## • Hlavní komponenty a moduly

### Globální komponenty (src/components/Global)

- **Layout.jsx** – Základní rozložení stránky (wrapper pro obsah).
- **Header/** – Hlavičky pro různé role: admin, firma, uživatel, návštěvník.
- **Footer.jsx** – Spodní část stránky.
- **Loader.jsx** – Načítací animace.
- **VacancyElement.jsx** – Karta pracovní nabídky.
- **AdminTableElement.jsx** – Prvek pro tabulky v admin rozhraní.

### Admin moduly

**Umístění:** src/components/admin-\* a src/pages/admin/

- **MainAdmin.jsx** – Hlavní stránka admina.
- **MainAllUsers.jsx / MainAllCompanies.jsx** – Správa uživatelů a firem.
- **CreateTechnologyPage.jsx** – Stránka pro přidávání technologií.
- **SearchTechnologyForm.jsx** – Vyhledávání technologií.

### Firemní moduly

**Umístění:** src/components/company-\* a src/pages/Company/

- **MainForCompany.jsx** – Hlavní dashboard firmy.
- **MainCompanyVacancies.jsx** – Seznam nabídek firmy.
- **MainCreateJob.jsx** – Vytváření nové pracovní nabídky.
- **SearchTechnologyCompany.jsx** – Výběr technologií do nabídky.
- **MainCompanyLogin.jsx / MainCompanyRegister.jsx / MainCompanyVerify.jsx** – Přihlášení, registrace, ověření firmy.
- **CompanyVacanciesPage.jsx** – Firemní stránka nabídek.
- **CompanyJobApplicationsPage.jsx** – Přihlášky na nabídky firmy.

## Uživatelské moduly

**Umístění:** src/components/user-\* a src/pages/user/

- **MainForUserPage.jsx** – Dashboard uživatele.
- **MainUserLogIn.jsx / MainUserRegister.jsx / MainUserVerify.jsx** – Přihlášení, registrace, ověření.
- **MainUserCvDetails.jsx / CvDetailsForm.jsx** – Detail životopisu.
- **MainUserApplications.jsx / TableUserApplicationsElement.jsx** – Moje přihlášky.
- **MainUserFavoriteVacancies.jsx** – Oblíbené nabídky.
- **MainSuitableVacancies.jsx** – Nabídky vhodné pro uživatele.
- **UserPersonalInformationForm.jsx / UserLoginDetailsForm.jsx** – Úprava údajů.

## Pracovní nabídky (Vacancies)

**Umístění:** src/components/all-vacancies-page/, src/components/vacancy-details-page/

- **MainAllVacancies.jsx / VacancyFilterForm.jsx** – Filtrování a zobrazení všech nabídek.
- **MainVacancyDetails.jsx** – Stránka s detaily nabídky.
- **VacancyRequirements.jsx / VacancyOfferings.jsx / VacancyTechnologyListElement.jsx** – Detailní části nabídky.

## Analýza životopisu

**Umístění:** src/components/analyze-cv-page/

- **AnalyzeMain.jsx** – Hlavní komponenta.
- **UploadElement.jsx** – Nahrání CV.
- **AnalyzeResultElement.jsx** – Výsledek analýzy.

## Ochrana tras (routing)

**Umístění:** src/router/

- **AdminProtectedRoutes.jsx** – Ochrana přístupu pro adminy.
- **CompanyProtectedRoutes.jsx** – Ochrana pro firemní účty.
- **UserProtectedRoutes.jsx** – Ochrana pro běžné uživatele.
- **MyAccountProtectedRoutes.jsx** – Ochrana pro stránku účtu.

## Stav a služby

- **store/** – Redux store a slice pro oblíbené nabídky.
- **services/** – API služby pro různé role: adminService, companyService, userService, atd.
- **utils/jwtDecode.js** – Pomocná funkce pro dekodování JWT tokenů.

### • Stavová správa

Aplikace používá Redux Toolkit k centrální správě stavu. Konkrétně spravuje seznam oblíbených pracovních nabídek uživatele:

store/favoriteVacanciesSlice.js obsahuje slice favorites, který uchovává:

- **items:** seznam oblíbených nabídek,
- **isLoading:** indikátor načtení,
- **error:** chybové hlášení.

### Asynchronní akce:

- **fetchFavorites** – Načítá oblíbené nabídky ze serveru při spuštění.
- **toggleFavorite** – Přidává nebo odebírá nabídku ze seznamu podle aktuálního stavu.

Stav je propojen přes configureStore() ve store.js, kde je slice favorites připojen ke globálnímu stavu.

### • Volání API

Pro komunikaci s backendem aplikace používá Axios, nakonfigurovaný na základě proměnné prostředí VITE\_BACKEND\_URL.



- V souboru `services/adminService.js` je vytvořen klient `apiAdmin`, který automaticky připojuje JWT token z `localStorage` v hlavičce `Authorization`.
  - `getAdminData()` – Načítá data aktuálního administrátora.
  - `createTechnology(technologyName)` – Přidává novou technologii.
  - `deleteTechnology(technologyId)` – Maže technologii podle ID.
  - `deleteVacancy(vacancyId)` – Maže nabídku práce podle ID.
  - `getAllCompanies()` – Načítá seznam všech firem.
  - `getAllUsers()` – Načítá seznam všech uživatelů.
- Soubor `services/userService.js` obsahuje funkce pro správu účtu uživatele, životopisu, oblíbených nabídek a žádostí o práci. Axios instance `apiUser` je nakonfigurována s JWT tokenem a základní URL `/user`.

- **Správa účtu a životopisu:**

- `getUserData()` – Získává informace o uživateli.
- `updateUserPersonalInformation(updatedData)` – Aktualizuje osobní údaje uživatele.
- `getUserCvDetails()` – Načítá detaily životopisu.
- `getUserCV()` – Stahuje uživatelův nahraný životopis.
- `uploadUserCv(CV)` – Umožňuje nahrát nový životopis (formulář `FormData`).
- `analyzeUserCV()` – Spouští analýzu životopisu (např. pro doporučení nabídek).

- **Oblíbené nabídky:**

- `getUserFavoriteVacancies()` – Získává seznam oblíbených pracovních nabídek.
- `addToFavoriteVacancies(vacancyId)` – Přidává nabídku do oblíbených.
- `removeFromFavoriteVacancies(vacancyId)` – Odstraňuje nabídku z oblíbených.

- **Žádosti o zaměstnání:**

- `getUserApplications()` – Vrací všechny žádosti uživatele.

- **applyToVacancy(vacancyId)** – Odesílá žádost o konkrétní pozici.

#### ▪ Nabídky na míru:

- **getAllMatchesVacancies(filters, page, size)** – Načítá nabídky práce přizpůsobené uživateli podle zadaných filtrů. Parametry se dynamicky čistí a přenáší pomocí **URLSearchParams**.

- Soubor **services/companyService.js** obsahuje funkce pro správu firemního účtu, vytváření pracovních nabídek a zpracování žádostí uchazečů. Axios instance **apiCompany** je nastavena s JWT tokenem a základní URL **/company**.

#### ▪ Správa účtu a nabídek:

- **getCompanyData()** – Získává informace o aktuálně přihlášené společnosti.
- **createVacancy(vacancyData)** – Vytváří novou pracovní nabídku s odeslanými daty (např. popis, požadavky, technologie).
- **getCompanyVacancies()** – Načítá seznam všech nabídek dané společnosti.

#### ▪ Správa žádostí uchazečů:

- **getVacancyApplications(vacancyId)** – Vrací seznam žádostí uchazečů pro konkrétní nabídku.
- **acceptApplicant(applicantId)** – Označuje žádost uchazeče jako přijatou.
- **rejectApplicant(applicantId)** – Označuje žádost uchazeče jako odmítnutou.

- Soubor **services/authenticationService.js** zajišťuje registraci, přihlášení, ověření a správu autentizačních dat uživatelů a společností.

#### ▪ Registrace a přihlášení

- **registerUser(userData)** – Registrace běžného uživatele.
- **registerCompany(companyData)** – Registrace firmy.
- **loginUser(userLoginData)** – Přihlášení uživatele a uložení JWT tokenu a expirace do localStorage.
- **loginCompany(companyLoginData)** – Přihlášení firmy a uložení JWT tokenu a expirace.

## ▪ **Ověření účtu**

- **verifyUser(userVerifyData)** – Ověření uživatelského účtu pomocí kódu.
- **verifyCompany(companyVerifyData)** – Ověření firemního účtu pomocí kódu.
- **resendVerificationCode(email)** – Zaslání nového ověřovacího kódu na email.

## ▪ **Správa relace**

- **logout()** – Odhlášení uživatele, odstranění tokenu a přesměrování na domovskou stránku.
- **isAuthenticated()** – Kontrola, zda je uživatel přihlášen (existuje token).
- **isTokenExpired()** – Kontrola platnosti tokenu podle uložené expirace.

## • **UI knihovny a stylování**

- Pro stylování se používá framework Tailwind.

## 5. Návrh backendu

- **Architektura backendu MVC**

Všechna citlivá data, jako jsou klíče API, odkazy na databáze atd., jsou uložena v souboru .env bez přístupu pro širokou veřejnost.

- **models**

Třídy popisující obchodní subjekty. Používají se k ukládání a přenosu dat, často představují strukturu databázové tabulky.

models/

- |—— ApplicationStatusEnum.java
- |—— Company.java
- |—— Resume.java
- |—— Role.java
- |—— RoleEnum.java
- |—— Technology.java
- |—— User.java
- |—— UserFavoriteVacancy.java
- |—— Vacancy.java
- |—— VacancyApplications.java

- **repository**

Repository používá pro práci s databází. Je součástí architektonické vrstvy zvané DAO (Data Access Object) a je určena k abstrahování přístupu k datům. Úložiště umožňují pohodlně provádět operace CRUD (Create, Read, Update, Delete) bez nutnosti ručního psaní dotazů SQL.

repository/

- |—— CompanyRepository.java

- |—— ResumeRepository.java
- |—— RoleRepository.java
- |—— TechnologyRepository.java
- |—— UserFavoriteVacancyRepository.java
- |—— UserRepository.java
- |—— VacancyApplicationsRepository.java
- |—— VacancyRepository.java

## ■ services

Service jsou třídy, které obsahují obchodní logiku aplikace. Slouží jako spojovací článek mezi řadiči (kde se zpracovávají požadavky HTTP) a úložišti (kde se přistupuje k datům).

services/

- |—— AdminService.java
- |—— CompanyAuthenticationService.java
- |—— CompanyService.java
- |—— EmailService.java
- |—— JWTService.java
- |—— ResumeAnalysisService.java
- |—— ResumeService.java
- |—— TechnologyService.java
- |—— UserAuthenticationService.java
- |—— UserService.java
- |—— VacancyApplicationsService.java
- |—— VacancyFilterService.java
- |—— VacancyMatchingService.java
- |—— VacancyService.java

## ■ controllers

Kontroléry jsou vstupním bodem do aplikace, přes který se zpracovávají příchozí požadavky HTTP.

controllers/

- |—— AdminController.java
- |—— CompanyAuthenticationController.java
- |—— CompanyController.java
- |—— PublicController.java
- |—— UserAuthenticationController.java
- |—— UserController.java

## ■ dto

DTO (Data Transfer Object) se používá jako mezimodel, aby se zabránilo přímému předávání entit z databáze navenek (např. do odpovědi JSON nebo do frontendu).

dto/

- |—— CompanyDetailsDTO.java
- |—— CreateTechnologyDTO.java
- |—— CreateVacancyDTO.java
- |—— LoginCompanyDTO.java
- |—— LoginUserDTO.java
- |—— RegisterCompanyDTO.java
- |—— RegisterUserDTO.java
- |—— ResponsesToVacancyDTO.java
- |—— ResumeAnalysisResponseDTO.java
- |—— TechnologyDTO.java
- |—— UserApplicationsDTO.java
- |—— UserCoverLetterDTO.java

- └── UserDetailsDTO.java
- └── VacancyDetailsDTO.java
- └── VacancyFilterCriteriaDTO.java
- └── VerifyCompanyDTO.java
- └── VerifyUserDTO.java

## ■ config

místo pro konfigurační třídy, které spravují konfiguraci komponent, zabezpečení, CORS, serializace a dalších aspektů aplikace.

V tomto případě se jedná o vytvoření admina v databázi, konfiguraci šifrování a ověřování hesel, konfiguraci pro práci s cloudem, konfiguraci pro odesílání e-mailů, konfiguraci pro vytvoření JWT, konfiguraci pro vytvoření rolí, konfiguraci pro vytvoření zabezpečení aplikace.

config/

- └── AdminConfiguration.java (vytvoření admina v databázi)
- └── ApplicationConfiguration.java (konfiguraci šifrování a ověřování hesel)
- └── ClouinaryConfiguration.java (konfiguraci pro práci s cloudem)
- └── EmailConfiguration.java (konfiguraci pro odesílání e-mailů)
- └── JWTAuthenticationFilter.java (konfiguraci pro vytvoření JWT)
- └── RoleSeederConfiguration.java (konfiguraci pro vytvoření rolí)
- └── SecurityConfiguration.java (konfiguraci pro vytvoření zabezpečení aplikace)

## ■ utils

Tento balíček obsahuje pomocné třídy, které provádějí klíčové technické úlohy související se zpracováním životopisů a interakcí s modelem umělé inteligence:

utils/

- └── GeminiClient.java
- └── ResumeParser.java

- **Popis API endpointy**

**Public (/public)** Veřejné koncové body, které mohou používat nepřihlášení uživatelé

- **GET /public/all-vacancies** (příjem všech volných pracovních míst)
- **GET /public/all-vacancies/{id}** (získání konkrétní pracovní nabídky)
- **GET /public/all-technologies** (získání všech technologií)

**User Authentication(/auth/user)** Veřejný uživatelský kontrolér pro přihlášení k účtu nebo registraci

- **POST /auth/user/signup** (přihlášení k uživatelskému účtu)
- **POST /auth/user/login** (Registrace uživatelů)
- **POST /auth/user/verify** (Ověření uživatelského účtu)
- **POST /auth/user/resend** (Opakované odeslání e-mailu s ověřovacím kódem)

**User (/user)** Soukromé koncové body pro registrované uživatele

- **GET /user/my-account**(Získání údajů )
- **POST /user/uploadCV**(Nahrávání životopisů)
- **DELETE /user/deleteCV** (Odstranění životopisů)
- **GET /user/getCV** (Získání životopisů)
- **GET /user/getCvDetails** (Získání údajů ze životopisu)
- **POST /user/analyzeCV** (Analýza životopisů)
- **GET /user/getMatchingVacancies** (Získání vhodných volných pracovních míst)
- **POST /user/apply/{vacancyId}** (Odpověď na volné pracovní místo)
- **GET /user/my-applications** (Získávání odpovědí na volná pracovní místa)
- **POST /user/add-to-favorite/{vacancyId}** (Přidání práce do oblíbených)
- **DELETE /user/remove-from-favorite/{vacancyId}** (Odstranění volného místa z oblíbených položek)
- **GET /user/my-favorite-vacancies** (Získávání oblíbených pracovních míst)
- **PATCH /user/my-account/update-personal-info** (Úprava osobních údajů)

**Company Authentication(/auth/company)** Veřejný controller firmy pro přihlášení k účtu nebo registraci

- **POST /auth/company/signup** (přihlášení k firemnímu účtu)
- **POST /auth/company/login** (Registrace firmy)
- **POST /auth/company/verify** (Ověření účtu firmy)
- **POST /auth/company/resend** (Opakované odeslání e-mailu s ověřovacím kódem)



## **Company (/company)** Soukromé koncové body pro registrované firmy

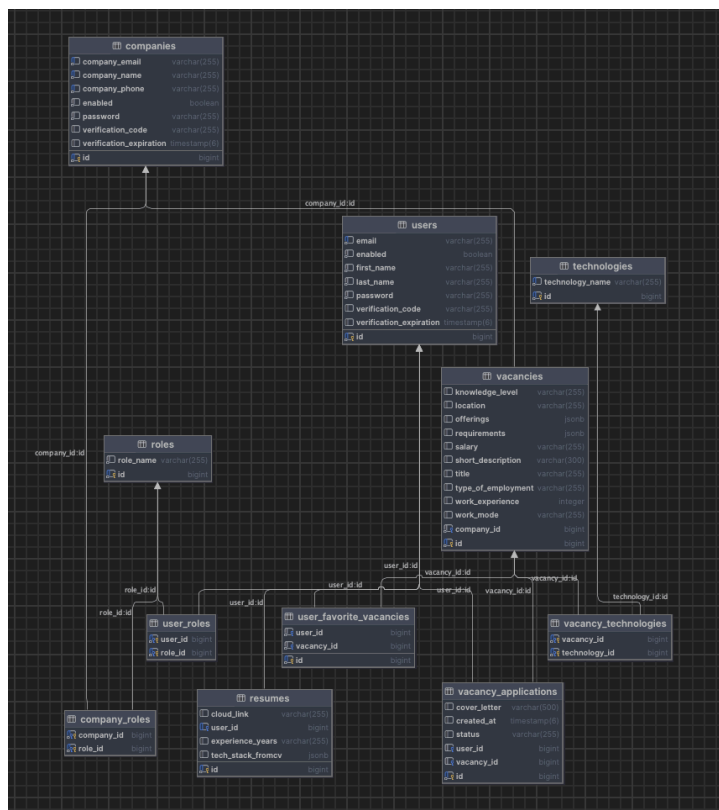
- **GET /company/my-company**(Získání údajů o firmě)
- **POST /company/create-vacancy**(vytváření pracovních míst)
- **GET /company/my-vacancies** (Získání vytvořených volných pracovních míst)
- **GET /company/my-vacancies/{vacancyId}/applications** (Získání odpovědí na konkrétní vytvořené volné pracovní místo)
- **PATCH /company/applications/{applicationId}/accept** (Přijetí kandidáta )
- **PATCH /company/applications/{applicationId}/reject** (Odmítnutí kandidáta )

## **Admin (/admin)** Soukromé koncové body pro admina

- **GET /admin/my-account**(Získání údajů admina)
- **POST /admin/add-technology**(Vytvoření technologie)
- **DELETE /admin/delete-technology/{id}** (Odstranění technologie)
- **DELETE /admin/delete-vacancy/{id}** (Odstranění volného pracovního místa)
- **GET /admin/all-users** (Získání všech uživatelů )
- **GET /admin/all-companies** (Získání všech společností )

## 6. Návrh databáze

- Diagram databáze



- Popis tabulek a relací

### 1. companies

- Uchovává informace o firmách.
- **Sloupce:** company\_email, company\_name, company\_phone, enabled, password, verification\_code, verification\_expiration, id (PK).

### 2. users

- Informace o uživatelích (Ukládá informace o uživateli).
- **Sloupce:** email, first\_name, last\_name, password, enabled, verification\_code, verification\_expiration, id (PK).

### 3. roles

- Různé role uživatelů (např. ROLE\_ADMIN, ROLE\_USER, ROLE\_COMPANY).
- **Sloupce:** role\_name, id (PK).

#### 4. user\_roles

- Relace mezi uživateli a jejich rolemi (M:N).
- **Sloupce:** user\_id (FK), role\_id (FK), id (PK).

#### 5. company\_roles

- Relace mezi firmami a rolemi.
- **Sloupce:** company\_id (FK), role\_id (FK), id (PK).

#### 6. resumes

- Uchovává životopisy uživatelů.
- **Sloupce:** cloud\_link, user\_id (FK), experience\_years, tech\_stack\_fromcv, id (PK).

#### 7. technologies

- Seznam technologií (např. Java, Python).
- **Sloupce:** technology\_name, id (PK).

#### 8. vacancies

- Nabídky pracovních pozic vytvořené firmami.
- **Sloupce:** knowledge\_level, location, offerings, requirements, salary, short\_description, title, type\_of\_employment, work\_experience, work\_mode, company\_id (FK), id (PK).

#### 9. vacancy\_technologies

- Vazba mezi nabídkami práce a technologiemi (M:N).
- **Sloupce:** vacancy\_id (FK), technology\_id (FK), id (PK).

#### 10. user\_favorite\_vacancies

- Oblíbené pracovní nabídky uživatelů.
- **Sloupce:** user\_id (FK), vacancy\_id (FK), id (PK).

## 11. vacancy\_applications

- Přihlášky uživatelů na pracovní nabídky.
- **Sloupce:** cover\_letter, created\_at, status, user\_id (FK), vacancy\_id (FK), id (PK).

- **Relace mezi tabulkami:**

- **users ↔ roles přes user\_roles (M:N).**
- **companies ↔ roles přes company\_roles (M:N).**
- **users ↔ resumes: 1:N (uživatel může mít více životopisů).**
- **companies ↔ vacancies: 1:N (firma může mít více nabídek).**
- **vacancies ↔ technologies přes vacancy\_technologies (M:N).**
- **users ↔ user\_favorite\_vacancies ↔ vacancies: M:N (uživatelé mohou mít oblíbené nabídky).**
- **users ↔ vacancy\_applications ↔ vacancies: M:N (uživatelé se mohou hlásit na více nabídek).**

## 7. Integrace

- K ukládání uživatelských souhrnů se na backendu používá rozhraní API Cloudinary.
- K analýze životopisů se používá rozhraní Google Gemini API.

## 8. Bezpečnost

- Pro účely zabezpečení se používají Tokeny JWT jsou potřebné k ověření, kdo je uživatel, a k tomu, aby mu byl umožněn přístup do správných částí aplikace. Když se uživatel přihlásí, server mu předá token. Tento token je připojen ke každému požadavku a server jej používá k tomu, aby pochopil, kdo požadavek odeslal a zda je k tomu oprávněn.

## 12. Přílohy

- Odkaz na návrh aplikace ve figmě

(<https://www.figma.com/design/CLQIBV5p9wejuXqliZ6CZE/JobFinder?node-id=0-1&t=mfdrxjRMjGg1PwOR-1>)