Sisteme informatice pentru monitorizarea mediului

Grupa 6503

**JobSearch**

<https://jobsearch-n4zw.onrender.com>

**Student:**

**Diaconu Alexandru-Gabriel**

**Cuprins**

[***1.*** ***Prezentarea temei*** 2](#_Toc199344239)

[***2.*** ***Justificarea temei*** 2](#_Toc199344240)

[***3.*** ***Tehnologii folosite*** 2](#_Toc199344241)

[***4.*** ***Arhitectura proiectului*** 4](#_Toc199344242)

[***5.*** ***Descrierea funcționalității*** 5](#_Toc199344243)

[***5.1*** ***Inregistrare si autentificare*** 5](#_Toc199344244)

[***5.2*** ***Upload și analiză CV*** 6](#_Toc199344245)

[***5.3*** ***Recomandare și filtrare joburi*** 7](#_Toc199344246)

[***5.4*** ***Profil utilizator și management cont*** 8](#_Toc199344247)

[***5.5*** ***Resetarea parolei prin link de resetare*** 9](#_Toc199344248)

[**Bibliografie** 10](#_Toc199344249)

[***Anexe*** 11](#_Toc199344250)

# ***Prezentarea temei***

JobSearch este o platforma web dedicata persoanelor care se afla in cautarea unui loc de munca, avand ca scop eficientizarea procesului de identificare a locurilor de munca relevante. Utilizatorii isi pot crea cont, pot sa incarce CV-ul in format .pdf. Aplicatia foloseste inteligenta artificiala pentru a extrage automat cuvinte cheie din continutul CV-ului, iar pe baza acestora si a preferintelor selectate de utilizator (precum locatia, nivelul de experienta si salariul dorit), platforma interogheaza surse externe de joburi, in cazul de fata Jooble, si returneaza cele mai potrivite anunturi, ordonate dupa scorul de potrivire.

# ***Justificarea temei***

In contextul actual al pietei muncii, procesul de cautare a unui loc de munca devine complex si mai ales consumator de timp. Majoritatea platformelor existente ofera rezultate fara a tine cont cu adevarat de profilul saau particularitaile CV-ului acestuia. Astfel utilizatorii sunt nevoiti sa parcurga manual sute, chiar mii de anunturi, ceea ce duce la o experienta obositoare si ineficienta.

Prin dezvoltarea aplicatiei JobSearch, am urmarit sa raspund acestei probleme, oferind o solutie care face procesul de cautare mai usor. Utilizarea inteligentei permite o potrivire mai precisa intre profilul candidatului si cerintele angajatorului.

Platforma poate ajuta atat tinerii absolventi cat si profesionistii cu experienta sa isi gaseasca maai usor un loc de munca potrivit.

# ***Tehnologii folosite***

Frontend:

* Vite

Vite provine din limba franceza si inseamna rapid. Este un tool pentru build care urmareste sa ofere o experienta de developare mai rapida si mi eficienta pentru proiectele web moderne. [1]

* React

React este biblioteca principala folosita pentru dezvoltarea interfetei. React te lasa sa creezi propria interafata folosind piese individuale numite componente. [2]

* TailwindCSS

Este un framework CSS. Spre deosebire de alte framework-uri, precum Bootstrap , acesta nu oferă o serie de clase predefinite pentru elemente precum butoane sau tabele. În schimb, creează o listă de clase CSS „utilitare” care pot fi folosite pentru a stiliza fiecare element prin combinare și potrivire. [3]

Backend:

* Node.js si Express

Backend-ul este construit cu Node.js si Express, oferind un API robust pentru toate operatiunile aplicatiei care includ: autentificare, procesarea CV-ului, gestionarea utilizatorilor, comunicarea cu API-uri externe.

* JWT

A fost folosit pentru autentificarea si securizarea sesiunilor. Astfel fiecare utilizator primeste un token la logare, folosit pentru a accesa rute protejate.

* Multer

A fost utilizat pentru upload-ul CV-urilor in format PDF.

* Nodemailer

Pentru trimiterea emailurilor de resetare a parolei.

* Bycrypt.js

Aceasta librarie a fost folosita pentru securizarea parolelor stocate in baza de date.

Baza de date

* Supabase

S-a utilizat supabase, fiind o baza de date Postgres, pentru stocarea utilizatorilor, a CV-urilor si a datelor asociate acestora. Supabase este o alternativa Firebase care este open source. [4]

Api-uri externe

* Jooble Api

Jooble Api a fost cel mai relevant Api, deoarece ofera un acces la un volum mare de oferte de munca, oferind o documentatie bine structurata si un raspuns rapid in format JSON. Singura limitare este ca ofera maxim 1000 de rezultate per Api call.

* Google Gemini API

Pentru extragerea automata a cuvintelor cheie din CV, am folosit Google Gemini API. Acesta analizeaza textul din CV si returneaza un set de cuvinte cheie relevante, fiecare cuvant avand asociat un scor.

* Logo.dev si Clearbit

Au fost utilizate pentru a afisa logo-ul companiei al fiecarui job. Clearbit a fost folosit ca alternativa in cazul in care Logo.dev nu gaseste logo pentru companie.

Cloud/Hosting

* Render.com

Proiectul este hostat pe Render.com, ceea ce asigura accesul public de oriunde.

# ***Arhitectura proiectului***

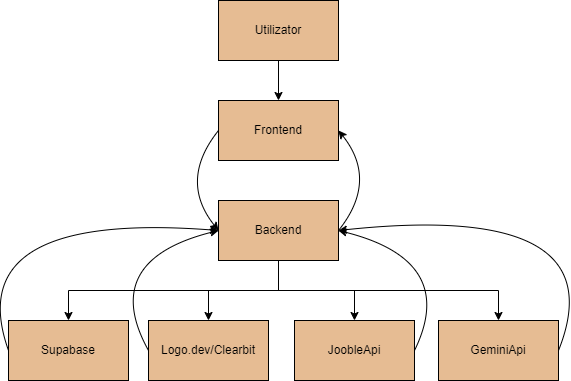


Fig.1 Arhitectura

Aplictia JobSearch este conceputa pe un model client server, separandu-se partea de interfata cu utilizatorul de logica de business si gestionarea datelor. Utilizatorul interactioneaza cu interfata, realizata cu React, care este responsabila de trimiterea cererilor catre backend si afisarea datelor. Backendul este cel care comunica cu alte servicii externe. Astfel supabase este utilizat pentru stocarea datelor utilizatorilor, JoobleApi ofera date despre joburi existente, GeminiApi este utilizat pentru extragerea cuvintelor relevante din CV, Logo.dev/Clearbit furnizeaza logo-urile companiilor care au publicat joburile. Aceasta structura ofera o experienta fluida utilizatorilor finalo, backendul fiind capabil sa gestioneze cereri complexe.

# ***Descrierea funcționalității***

## ***Inregistrare si autentificare***

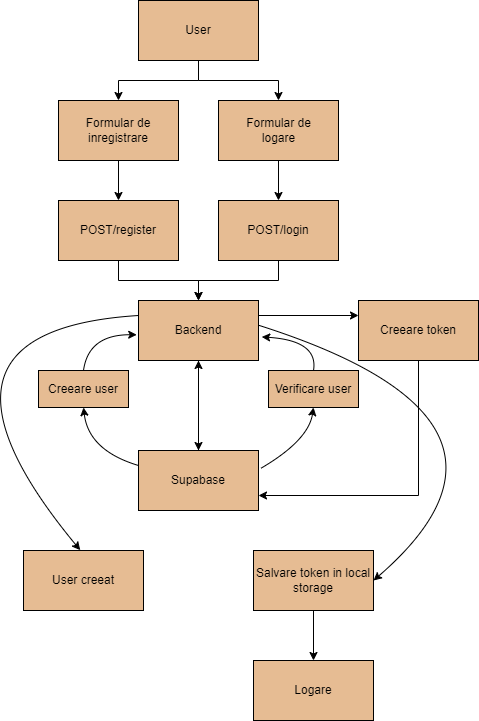


Fig. 2 Diagrama inregistrare si autentificare

Userul intra pe formularul pe care doreste. Daca doreste sa sa inregistreze isi introduce datele in campurile specifice si face cerere de inregistrare. In acel moment se face o cerere catre backend pe /register si acesta salveaza datele in baza de date. Daca s-a reusit inserarea in baza de date utilizatorul este anuntat ca s-a inregistrat cu succes. Daca doreste sa se logheze isi introduce adresa de email si parola folosite la inregistrare si face cererea de logare. In acel moment se face cerere catre backend pe /login si se verifica in supabase daca exista utilizatorul. Daca exista backendul creeaza un token pe care il insereaza in baza de date. Daca inserarea s-a facut cu succes atunci backendul trimite catre utilizator tokenul. Cand utilizatorul primeste token-ul pe partea de frontend, se salveaza tokenul in local storage-ul browserului.

## ***Upload și analiză CV***

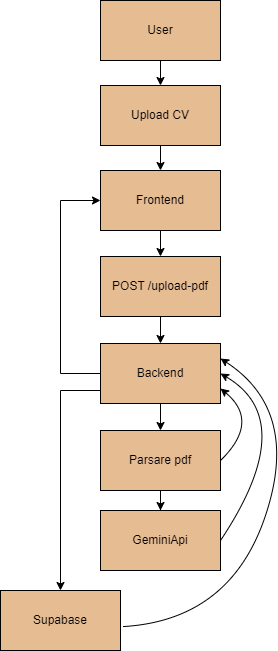


Fig. 3 Upload și analiză CV

User-ul face upload la CV in frontend, iar atunci se apeleaza backend-ul pe /upload-pdf. In backend mai intai se verifica tokenul primit si dupa care se preia fisierul pdf si se extrage textul din el. Acest text urmeaza a fi salvat in baza de date. Apoi aceste informatii sunt procesate cu ajutorul Gemini Api care extrage cuvintele relevante si le atribuie un scor de relevanta de la 1 la 100. In final datele extrase si CV-ul vor fi salvate in baza de date. Daca apare vreo eroare intre timp, spre exemplu Gemini Api nu raspunde sau nu se poate parsa fisierul, sau nu se pot insera in baza de date backendul anunta utilizatorul ca o eroare a avut loc.

## ***Recomandare și filtrare joburi***

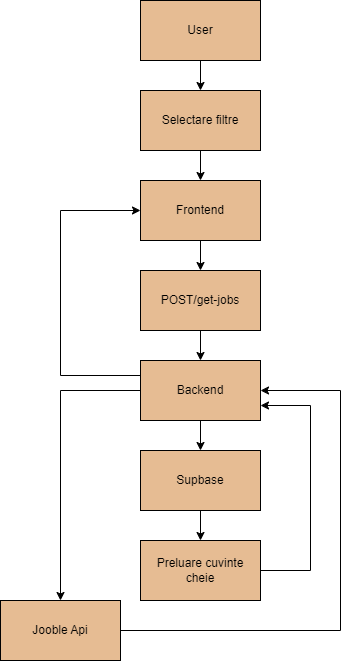


Fig. 4 Recomandare și filtrare joburi

User-ul face selectia filtrelor dupa care frontendul care cerere la backend pe /get-jobs. In backend se face mai intai verificarea tokenului dupa care se cer de la baza de date cuvintele cheie pentru a face cerere la Jooble Api. Raspunul primit de la Jooble Api, impreuna cu cuvintele relevante cu scor, este dupa aceea trimis la frontend. In frontend se calculeaza scorul de potrivire a fiecarei descriere cu cuvintele cheie. Sistemul de scor functioneaza astfel: prima data este scorul total la care se adauga puncte daca cuvantul cheie se afla In titlu si text. Daca locatia selectata este inclusa in locatia jobului se adauga puncte in plus la scorul total. De asemenea exista scorul pentru senioritate. Se verifica daca filtrul pentru senioritate se afla in text sau in titlu se adauga puncte la bonusul de senioritate. Apoi se normalizeaza scorul total la care se adauga scorul de senioritate. Se afiseaza in ordine descrescatoare scorul.

## ***Profil utilizator și management cont***

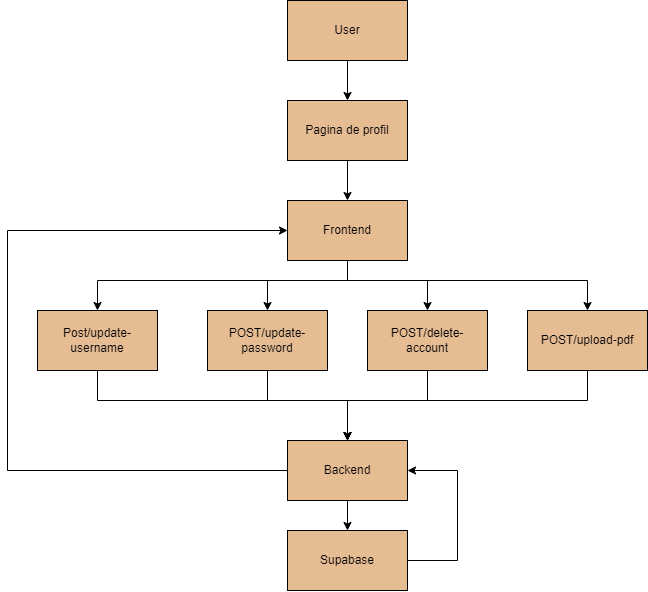


Fig. 5 Profil utilizator și management cont

Dupa autentificare utilizatorul poate aceesa pagina de profil, unde interactioneaza cu interfata care ii permita sa isi schimbe usernameul, sa isi schimba parola, sa isi incarce alt CV sau sa isi stearga contul. Toate aceste cereri de tip POST sunt trimise catre backend. Mai departe se face cerere la supabase pentru a face modificarile necesare in baza de date. Dupa procesare supabase trimite raspuns backend-ului care la randul lui trimite catre frontend mesaj care actualizeaza interfata utilizatorului in functie de succesul operatiunii.

## ***Resetarea parolei prin link de resetare***

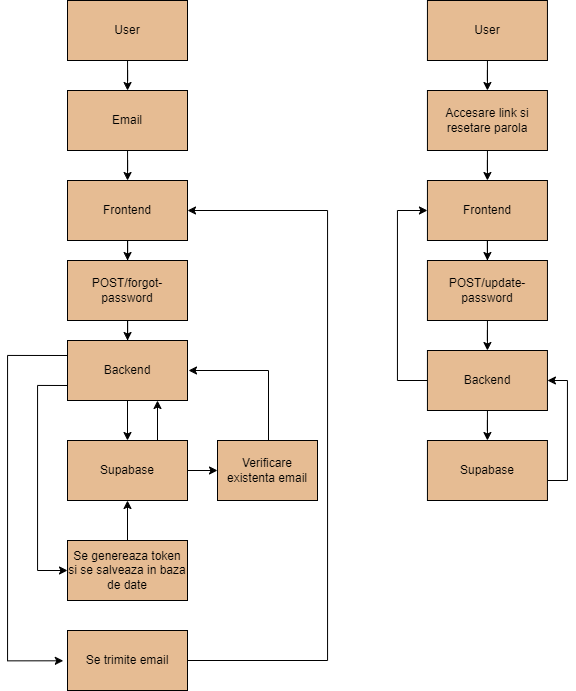


Fig. 6 Resetarea parolei prin link de resetare

Pentru resetarea parolei utilizatorul introduce adresa de email corespunzatoare contului pentru care vrea sa schimbe parola. Din frontend se face cerere catre backend pe /forgot-password. In backend se verifica mai intai daca adresa de email exista in baza de date. Daca nu exista nu se va intampla nimic. Daca exista se genereaza un token pentru resetare valabil 15 minute care este salvat in baza de date si apoi se introduce in likul de resetare ca parametru. Dupa aceasta se trimite email-ul. In ambele cazuri utilizatorul va fi anunta ca s-a trimis email doar daca exista adresa de email in baza de date. Dupa ce a primit email-ul asociat utilizatorul il va accesa, isi va introduce noua parola dorita si va face o cerere catre backend pe /update-password. Parola apoi se va salva in baza de date si utilizatorul va fi redirectionat catre pagina de logare.

# **Bibliografie**

[1] <https://vite.dev/guide/>

[2] <https://react.dev/>

[3] <https://en.wikipedia.org/wiki/Tailwind_CSS>

# [4] <https://supabase.com/>

# ***Anexe***

Codul sursa se gaseste pe linkul: <https://github.com/DiaconuGabriel/JobSearch>