

Documentatie

1. *Prezentare generala*

Este o aplicatie de frontend de prezentare, adauga si editare de joburi formata din 3 pagini: home, jobs si add jobs. In pagina home se afla 3 sectiuni: prima sectiune contine titlul aplicatiei, a doua sectiune contine doua clase de cautare a joburilor si una de adaugare a joburilor, iar a treia sectiune afiseaza joburile care au fost adaugate. In pagina jobs se afla o singura sectiune care afiseaza toate joburile care au fost adaugate pe site. Iar pagina "Add Jobs" contine un formular prin care se poate adauga un nou job.

2. *Structura aplicatiei*

Fisierul de baza este App.vue in care se adauga componentele de baza: Navbar si biblioteca RouterView pentru aducerea dinamica a celorlalte componente. In folderul src/views gasim toate componentele din paginile aplicatiei unde se incarca si componente din folderul src/components, iar in folderul router/index.js gasim definite route-le folosite in aplicatie.

3. *Functionalitatea aplicatiei*

In pagina "Home" se apeleaza fisierul "HomeView.vue" in care sunt apelate 3 componente din folderul components: Hero, HomeCards si JobListings. Hero afiseaza un titlu si un subtitlu intr-o sectiune cu un fundal verde. Componenta foloseste proprietati (props) pentru a primi titlul si subtitlul, oferind valori implicite daca nu sunt furnizate. HomeCards afiseaza doua carduri intr-o sectiune. Fiecare card ofera informatii si linkuri pentru dezvoltatori si angajatori, utilizand RouterLink pentru navigarea internă. Cardurile sunt afisate intr-o grila si au stiluri distincte pentru a atrage atentia utilizatorilor. JobListings afiseaza o lista de joburi intr-un format grila, folosind o cerere HTTP pentru a obtine datele joburilor si afisand un spinner pe durata încărcării. Componenta permite limitarea numărului de joburi afisate si include un buton optional pentru a naviga la o pagina cu toate joburile. Foloseste Vue Router pentru navigare, Axios pentru cereri HTTP si un spinner pentru a indica încărcarea datelor. Are setat un numar limit de joburi pe care le afiseaza si odata apelat butonul de "View All Jobs" atunci se seteaza un flag care determina dacă se afiseaza butonul de navigare către toate joburile (valoare implicită false).

In pagina "Jobs" se apeleaza fisierul "JobsView.vue" in care este apelata doar componenta "JobListings.vue" din folderul components. Totodata JobListings apeleaza JobListing care este containerul fiecarui job si care permite accesarea detaliilor fiecarui job prin butonul Read More care face un redirect catre pagina JobView prin RouterLink. JobView afiseaza detaliile unui job specificat si ofera functionalitati pentru editarea si stergerea jobului respectiv. Stergerea jobului se face accesand butonul "Delete Job" care apeleaza functia deleteJob si care afiseaza un dialog de confirmare, sterge jobul dacă utilizatorul confirmă si afiseaza notificări de succes sau eroare. Editarea jobului se face accesand butonul "Edit Job" si care face un redirect in fisierul EditJobView si care defineste un form pentru gestionarea datelor introduse in formular; un state pentru gestionarea starii jobului si starea de încărcare si un toast pentru afisarea notificărilor. Practic se afiseaza un nou formular cu datele jobului introduse in inputuri si la accesarea butonului "Update Job" se apeleaza functia handleSubmit pentru gestionarea trimiterii formularului si care creeaza un obiect updateJob cu datele din formular si face o solicitare PUT la server pentru a actualiza jobul. Dacă actualizarea are succes, afiseaza un mesaj de succes si redirectioneaza utilizatorul la pagina jobului actualizat. În caz de eroare, afiseaza un mesaj de eroare.

In pagina “AddJobs.vue” se apeleaza doar fisierul “AddJobView.vue” care contine un form in care se afla toate inputurile ce trebuiesc completate si odata apasat butonul “Add Jobs” se apeleaza la form functie handleSubmit care creeaza un nou job cu datele din formular si pe care-l trimite prin POST catre ‘api/jobs’ si unde intoarce un mesaj de succes in caz de succes, si un mesaj de eroare in caz de eroare.