

Universitatea Politehnica din București

Aplicatie pentru gestiunea proiectelor

Jercan Alexandru-Constantin

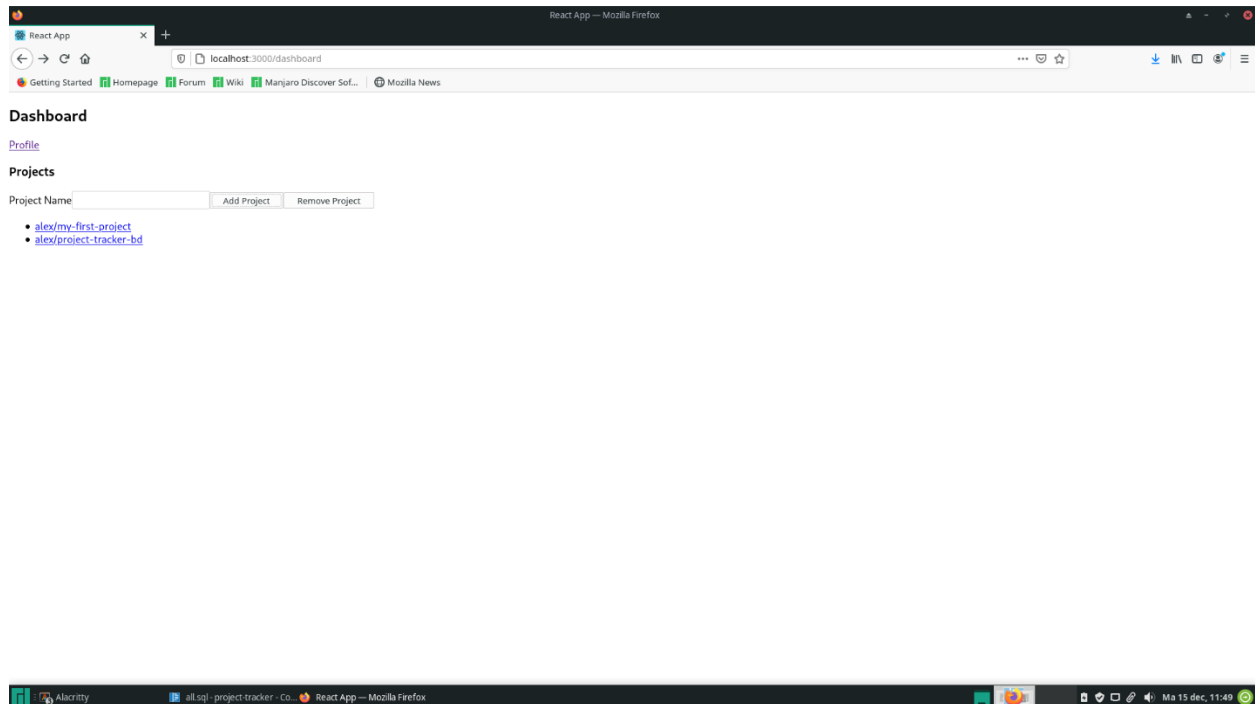
341C3

1/24/2021

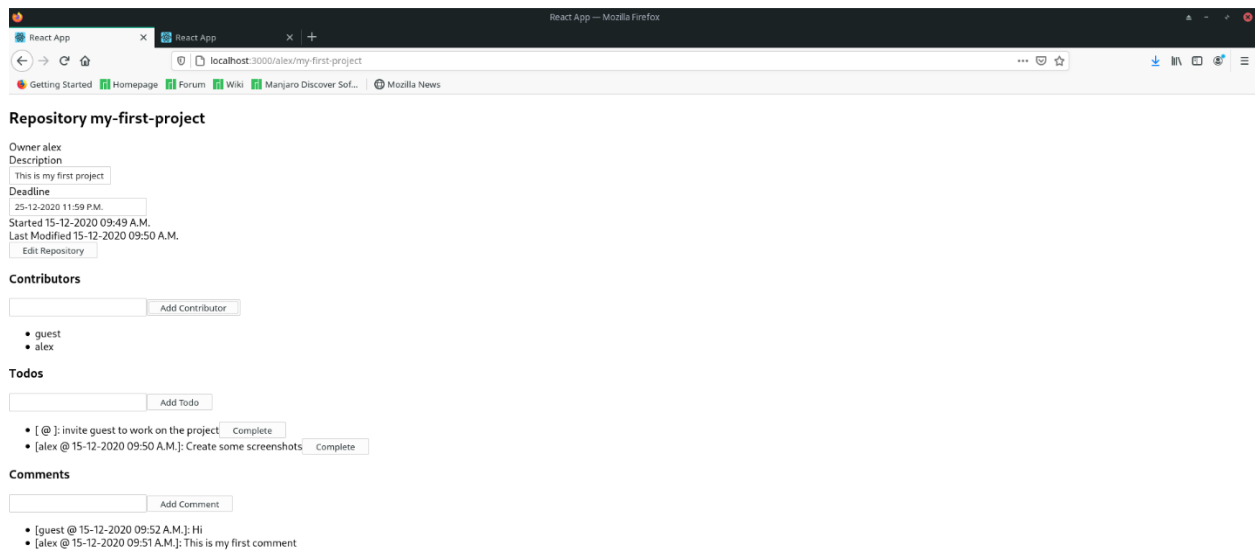
Capitolul 1. Descrierea proiectului

În viața unui student, programator,ținerea evidentei progresului unui proiect este esențială în completarea acestuia, așa că am decis ca proiectul creat de mine să se adreseze acestei probleme și am creat o aplicație pentru gestiunea proiectelor și a temelor. Principalele surse de inspirație pentru comportamentul și aspectul aplicației sunt github și site-ul de cursuri al facultății.

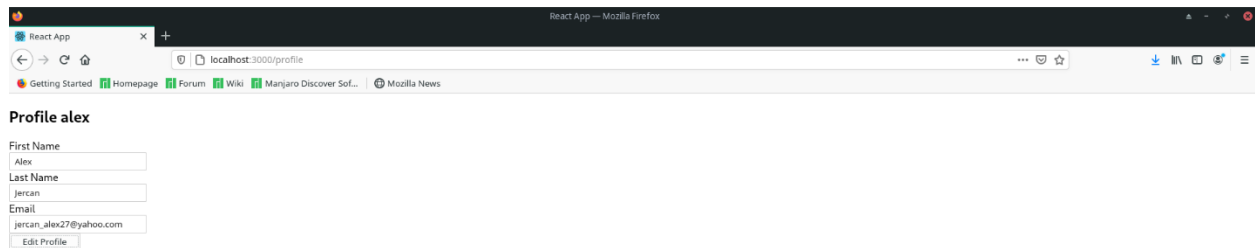
Una din problemele pe care le-am observat pe site-ul de cursuri o reprezintă cronologia, locul în care îți poți vedea și încărca temele, care afișează toate temele și testele, inclusiv cele de la alte grupe, făcând astfel destul de greu de observat temele tale. Aplicația mea rezolvă această problemă făcând foarte ușor de creat repository-uri care să reprezinte aceste teme.



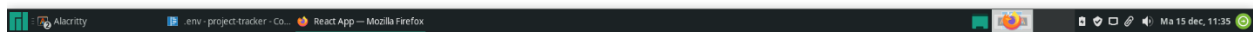
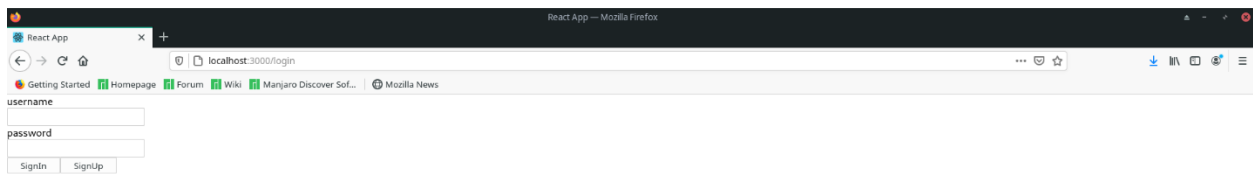
În această figură se poate observa pagina de start de unde îți poți adăuga sau șterge proiectele.



In aceasta figura se poate observa modul in care arata un proiect. Principalele caracteristici sunt adaugarea de comentarii, todo-uri, descriere, deadline, contributori (daca este un proiect in echipa). Oricine poate vedea un repository, dar numai contributorii pot modifica informatii. Poti accesa un repository prin nume_owner/nume_proiect, la fel ca pe github.



Am inclus si un mod de a putea edita profilul unui utilizator.



Pagina de login. Daca incerci sa accesezi site-ul fara sa fi logat, vei fi redirectat la aceasta pagina si apoi iti vei putea continua treaba.

Capitolul 2. Descrierea bazei de date

Pentru baza de date am creat 5 tabele, unul pentru a pastra utilizatorii, unul pentru a pastra proiectele, unul pentru comentarii, unul pentru todo-uri si unul pentru a realiza legatura dintre contributori si proiecte.

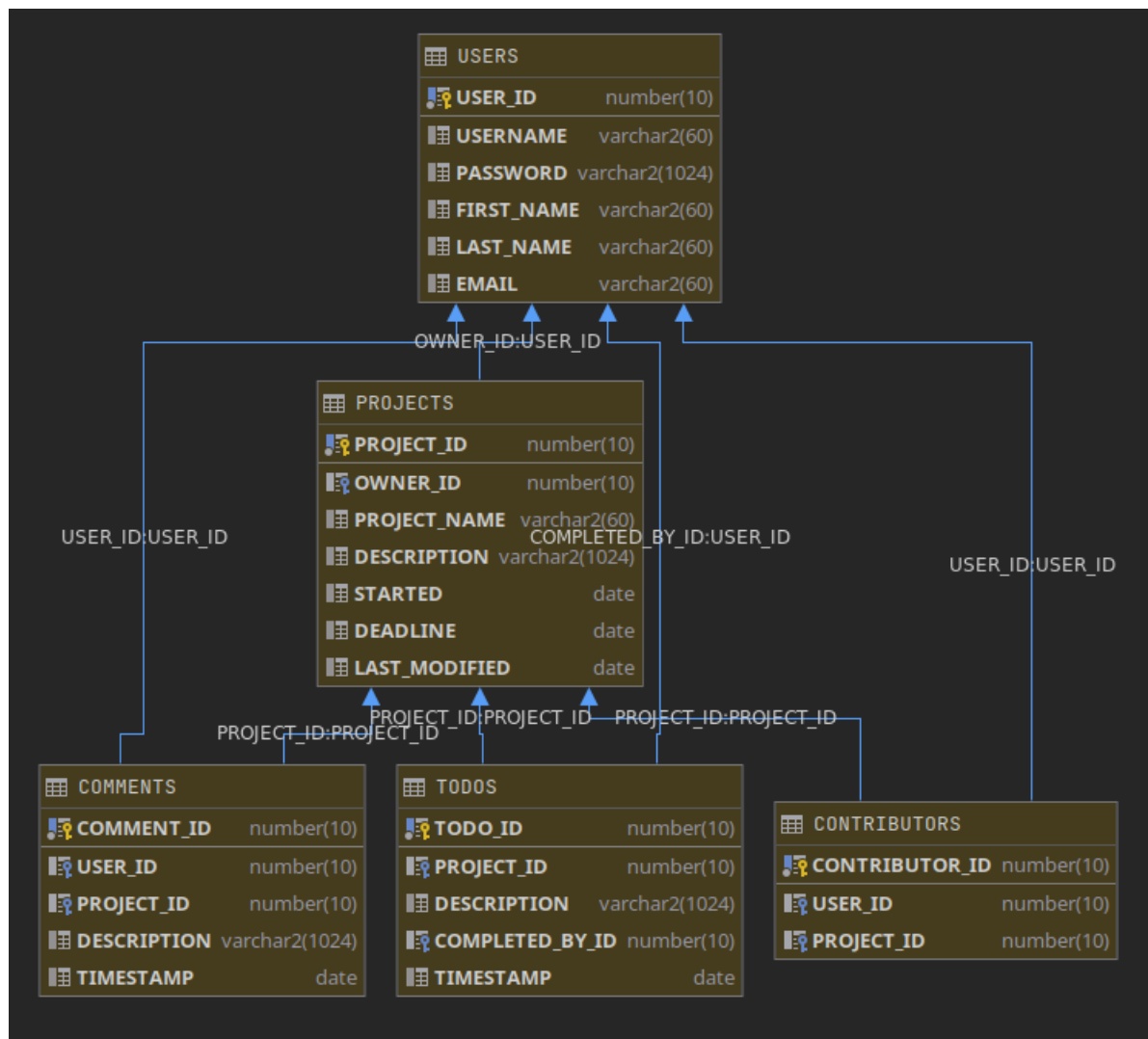
Tabelul de utilizatori contine o cheie primara `USER_ID`, doua coloane care nu pot fi modificate dupa signup, `username` si `password`, si trei coloane care pot fi editate de utilizator, `prenumele`, `numele` si `email-ul`.

Tabelul proiectelor contine o cheie primara `PROJECT_ID` si o cheie straina `OWNER_ID` care se refera la coloana `USER_ID` din tabelul `users` si reprezinta utilizatorul care a creat acest proiect. Coloanele `description` si `deadline` pot fi editate de contributori de pe pagina proiectului, `numele proiectului` nu mai poate fi schimbat, iar `last_modified` reprezinta data la care sa efectuat ultima modificare la informatiile proiectului.

Tabelul `comments` contine o cheie primara `COMMENT_ID` si doua chei straine, una reprezentand utilizatorul care a lasat un comentariu si cealalta reprezentand proiectul unde se afla comentariul. Coloana `timestamp` se genereaza automat in momentul care se trimite comentariul, iar `description` reprezinta corpul comentariului.

Tabelul `todos` are o cheie primara `TODO_ID` si doua chei straine, `project_id` pentru a stii la ce proiect se afla todo-ul respectiv si `completed_by` care reprezinta utilizatorul care a marcat todo-ul ca fiind completat. In mod asemanator cu tabelul `comments`, si acesta are o coloana pentru `timestamp` si pentru `continut`.

Tabelul `contributors` contine o cheie primara `CONTRIBUTOR_ID` si doua chei straine, `USER_ID` care reprezinta utilizatorul care contribuie la proiect si `PROJECT_ID` care reprezinta proiectul in sine.



Perechea username,project_name este unica, asta inseamna ca un proiect poate exista cu acelasi nume pentru mai multi utilizatori diferiti

Coloana username a tabelului user este unica, astfel nu pot exista mai multi utilizatori cu acelasi nume.

Am creat cate un trigger pentru fiecare tabel pentru a genera automat ID-urile.

Functii ajutatoare pentru a afla id pe baza username-ului si a proiectului, getUserId, getProjectId.

O functie care verifica daca un user este contributor doesContributorExistFunc.

Proceduri pentru a insera datele in baza de date insertUser, insertProject, insertContributor, insertComment, insertTodo. Pentru a edita datele, unde e posibil, editProfile, editRepository, completeTodo. Proceduri pentru a returna informatii, fie prin intermediul unui cursor pentru getProjects, getContributors, getComments si getTodos, fie pentru o intrare specifica precum getPassword care intoarce parola pentru un username dat, getProfile care intoarce informatii despre profilul unui utilizator, doesProjectExists care verifica daca un proiect exista sau nu, doesContributorExists care verifica daca

exista un contributor cu username-ul dat ca parametru. De asemenea am creat si proceduri care sterg date, deleteProject, care sterge in cascada informatia despre contributori, comentariile si todo-urile.

Pentru conexiunea cu baza de date am folosit oracle-db pentru nodejs. Variabilele de mediu ce trebuiesc setate sunt

ORACLEDB_USER=in cazul meu SYSTEM

ORACLEDB_PASSWORD=in cazul meu parola

ORACLEDB_CONNECTIONSTRING=in cazul meu localhost/orclpdb1

ORACLEDB_POOL_MIN=numarul minim de conexiuni concurente

ORACLEDB_POOL_MAX=nr maxim

In plus am creat un wrapper pentru oracle-db care contine o functie care se conecteaza la baza de date, face o cerere si apoi elibereaza conexiunea, pentru a avea o comunicare eficienta.

Capitolul 3. Concluzii

Consider ca aplicatia isi atinge obiectivul, acela de a oferi posibilitatea gestiunii proiectelor cu o interfata cu utilizatorul cat mai simpla. Nu este pe departe la fel de buna ca github/cs curs pentru proiectele mari, dar pentru task-uri mici (cum ar fi teme la laborator sau pentru elevii de la liceu/generala etc) poate fi mult mai usor de utilizat.

Capitolul 4. Bibliografie

oracledb - [oracledb - npm \(npmjs.com\)](https://www.npmjs.com/package/oracledb)

react - [Getting Started – React \(reactjs.org\)](https://reactjs.org/docs/getting-started.html)