

*Academia de Studii Economice București*  
*Facultatea de Cibernetică, Statistică și Informatică Economică*

## ***Aplicație web pentru gestionarea rezolvării bug-urilor***

Aplicație realizată de:

**Barbu Alexandra-Maria**

**Boangiu Alexandra-Mihaela**

**Bratu Andrei-Bogdan**

**Cornea Alexandru-Nicolae**

**Tudose Bogdan**

Profesor îndrumător:

**asociat drd. Cartas Cosmin Gabriel**

Grupa **1075**

*An universitar 2019-2020*

# Cuprins

1. Introducere	3
1.1. Definire problemă	3
1.2. Definire obiective	3
2. Descriere scenariu utilizare	3
3. Implementare funcționalități back-end	4
3.1. Creare și rulare server/client	4
3.2. Metodele HTTP <i>POST</i> și <i>GET</i>	5
3.3. Stocare persistentă	7
3.4. Rutare ( <i>Single Page Application</i> )	9
4. Implementare funcționalități front-end	10
5. Concluzii	13
Bibliografie	13

# 1. Introducere

## 1.1. Definire problemă

În zilele noastre, dezvoltarea aplicațiilor a devenit o necesitate pentru majoritatea domeniilor existente. Fie că ne raportăm la domeniul economic, domeniul producției sau domeniul comercial, acestea sunt omniprezente în cadrul oricărei întreprinderi, fapt care ridică standardele dezvoltatorilor de aplicații.

Indiscutabil, domeniul IT vine zi de zi cu noi și noi provocări pentru programatori în ceea ce privește soluțiile oferite sau aplicațiile proiectate. După cum deja știm bine din experiența proiectelor pe care le-am realizat până acum, implementarea unei aplicații perfect funcționale, care să ia în calcul toate aspectele posibile este extrem de dificilă, însă nu și imposibilă. De aceea, propunem o aplicație care să vină în ajutorul programatorilor în ceea ce privește testarea proiectelor realizate de ei, primind feedback-uri și task-uri de îndeplinit din partea altor persoane, tocmai pentru a obține cea mai bună versiune a aplicației aflată în discuție.

## 1.2. Definire obiective

Aplicația noastră își propune să gestioneze bug-urile într-o aplicație. Ideea acesteia a plecat de la dorința și totodată necesitatea de a elimina vulnerabilitățile descoperite în fazele finale ale proiectelor. Astfel, scopul principal devine eliminarea disfuncționalităților în cadrul soluțiilor și facilitarea organizării membrilor proiectelor și a persoanelor care au rolul de a le testa, pentru a putea oferi utilizatorilor finali o experiență de înaltă calitate.

Mai mult decât atât, aplicația vine cu o interfață user-friendly, extrem de eficientă, care le permite utilizatorilor să înregistreze informațiile-cheie despre proiecte și bug-uri într-un timp foarte scurt. Arhitectura acesteia este de tip *Single Page Application*, lucru implementat cu ajutorul mecanismului de rutare. Funcționalitățile incorect implementate vor fi monitorizate și se va încerca să fie gestionate și rezolvate rapid în funcție de prioritatea și de gradul de severitate aferente.

# 2. Descriere scenariu utilizare

Utilizatorul se poate conecta la aplicație cu un cont bazat pe o adresă de email și, bineînțeles, cu o parolă.

În cazul în care acesta face parte dintr-un proiect, deținând calitatea de *MP*, el poate înregistra un proiect pentru a fi monitorizat în cadrul aplicației, repository-ul proiectului și echipa de proiect din care face parte fiind singurele date pe care trebuie să le furnizeze. Astfel, *MP*-ul poate adăuga, respectiv modifica un proiect.

În cazul în care acesta nu este asignat niciunui proiect, el se poate adăuga ca *TST* la proiect, putând, astfel, să înregistreze un bug în aplicație. Bug-ul conține aspecte referitoare la severitate, prioritate de rezolvare, descriere și un link la commit-ul la care se referă. Astfel, *TST*-ul doar poate adăuga un bug.

În calitate de *MP*, pot fi vizualizate bug-urile înregistrate în aplicație aferente proiectelor din care acesta face parte. Pe lângă acest aspect, își poate alocă rezolvarea unui bug.

### 3. Implementare funcționalități back-end

Back-end-ul este realizat cu ajutorul unei interfețe *REST*, în *Node.js*.

### 3.1. Creare și rulare server/client

Rularea serverului este prezentată mai jos. Comenzile folosite au fost `cd webosses_gr1075/server`, respectiv `npm start`.

```
AWS Cloud9 File Edit View Run Tools Window Support View Run
Go to Anything (Ctrl #) Welcome npm -fp-172-31-28 x +
Environment
▼ webtech_barbaclouds:
  ► assignment_4
  ► bug_tracking
  ► lab07_1075
  ► lab08_1075
  ► lab09_1075
  ► lab10_1075
  ► lab11_1075
  ► lab13_1075
  ► product-app
  ▼ webosses_gr1075
    ► my-app
    ▼ server
      ► config
      ► controller
      ► Database
      ► node_modules
      ► routes
      ► service
      ► package-lock.json
      ► package.json
      ► server.js
      Documentable Faza 2.x
      README.md
  ► webtech_libs
    ► README.md

git repository:
keywords:
author:
license: (ISC)
About to write to /home/ubuntu/environment/webosses_gr1075/server/package.json:
{
  "name": "server",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "start": "nodemon server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "sequelize": "^2.0.2",
    "sequelize": "^5.21.3"
  },
  "devDependencies": {
    "nodemon": "^2.0.2"
  },
  "description": ""
}

Is this OK? (yes) yes
ubuntu:/environment/webosses_gr1075/server (master) $ npm start

> server@1.0.0 start /home/ubuntu/environment/webosses_gr1075/server
> nodemon server.js

[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting server.js

Server started on port 8080
Executing (default): SELECT 1+1 AS result
Executing (default): CREATE TABLE IF NOT EXISTS `PROJECTS` (`denumireProject` VARCHAR(255) NOT NULL, `linkRepository` VARCHAR(255) NOT NULL, `ername` REFERENCES `USERS` (`username`) ON DELETE SET NULL ON UPDATE CASCADE) ENGINE=InnoDB;
Executing (default): CREATE TABLE IF NOT EXISTS `USERS` (`username` VARCHAR(255) NOT NULL, `password` VARCHAR(255) NOT NULL, `email` VARCHAR(255) NOT NULL, `priority` VARCHAR(255) NOT NULL, `privilege` VARCHAR(255) NOT NULL, `role` VARCHAR(255) NOT NULL, PRIMARY KEY (`denumirebug`), FOREIGN KEY (`projectIdDenumireProject`) REFERENCES `PROJECTS` (`denumireProject`) ON DELETE SET NULL ON UPDATE CASCADE) ENGINE=InnoDB;
Database connection success!
Executing (default): SHOW INDEX FROM `PROJECTS`;
Executing (default): SHOW INDEX FROM `USERS`;
Executing (default): SHOW INDEX FROM `BUGS`;
```

Rularea clientului este prezentată mai jos. Comenzile folosite au fost `cd webosses_gr1075/my-app`, respectiv `npm start`.

```

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P) Welcome x npm -p-172-31-28 x
ubuntu:/environment $ cd webtech_labs/my-app
ubuntu:/environment/webtech_labs/my-app (master) $ npm i axios
> node-sass@4.13.0 install /home/ubuntu/environment/webtech_labs/my-app/node_modules/node-sass
> node scripts/install.js
Cached binary found at /home/ubuntu/.npm/node-sass/4.13.0/linux-x64-binding.node
> core-js@2.6.10 postinstall /home/ubuntu/environment/webtech_labs/my-app/node_modules/core-js
> node postinstall || echo "ignore"
> core-js@4.7 postinstall /home/ubuntu/environment/webtech_labs/my-app/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"
> core-js-pure@1.4.7 postinstall /home/ubuntu/environment/webtech_labs/my-app/node_modules/core-js-pure
> node -e "try{require('./postinstall')}catch(e){}"
> node-sass@4.13.0 postinstall /home/ubuntu/environment/webtech_labs/my-app/node_modules/node-sass
> node scripts/build.js
Binary found at /home/ubuntu/environment/webtech_labs/my-app/node_modules/node-sass/vendor/linux-x64-binding.node
Testing binary
Binary is fine
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/jest-haste-map/node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/chokidar/node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules/fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
+ axios@0.19.1
added 1786 packages from 852 contributors and audited 905704 packages in 41.183s
found 1 low severity vulnerability
run npm audit fix to fix them, or 'npm audit' for details
ubuntu:/environment/webtech_labs/my-app (master) $ npm start
> my-app@0.1.0 start /home/ubuntu/environment/webtech_labs/my-app
> PORT=3000 react-scripts start

```

### 3.2. Metodele HTTP *POST* și *GET*

La început, până am implementat și interfața cu utilizatorul, pentru a adăuga useri, proiecte și bug-uri în baza de date, am folosit aplicația *POSTMAN* (request-ul **POST**), iar pentru a verifica că adăugarea a fost făcută cu succes, am folosit tot aplicația *POSTMAN* (request-ul **GET**). Acum, toate aceste lucruri pot fi realizate, respectiv vizibile chiar din pagina aplicației.

#### ▪ Tabela *USERS*

##### ➤ GET

```

GET http://3.17.177.8:8080/api/user

Params Authorization Headers (7) Body Pre-request Script Tests Settings
Query Params
KEY VALUE
Key Value

ody Cookies Headers (7) Test Results
Pretty Raw Preview Visualize BETA JSON
[
  {
    "username": "abcde@abcd.com",
    "name": "numeabc",
    "prenume": "prenumeabc",
    "password": "Ikejhgwbjfw",
    "statut": "MPP"
  },
  {
    "username": "abckbd@email.com",
    "name": "numeabc",
    "prenume": "prenumeabc",
    "password": "Uhgjfndbfjsd",
    "statut": "TST"
  },
  {
    "username": "username1@user.com",
    "name": "username1",
    "prenume": "preusername1",
    "password": "Passworduser1",
    "statut": "MPP"
  }
]

```

## ▪ Tabela *PROJECTS*

➤ GET

GET http://3.17.177.8:8080/api/user POST http://3.135.204.60:8080/api/u... POST http://3.135.204.60:8080/api/p... GET http://3.17.177.8:8080/api/proj... GET http://3.1

Untitled Request

GET http://3.17.177.8:8080/api/project

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize BETA JSON

```

1  [
2    {
3      "denumireProiect": "2twertwr",
4      "linkRepository": "reponoufor123",
5      "denumireEchipa": "echipanouafor123",
6      "USERUsername": "abcde@abcd.com"
7    },
8    {
9      "denumireProiect": "dsfasdf",
10     "linkRepository": "sdfafsaf",
11     "denumireEchipa": "asdfasdfa",
12     "USERUsername": "username1@user.com"
13   },
14   {
15     "denumireProiect": "ngiergeritg",
16     "linkRepository": "reponou1",
17     "denumireEchipa": "echipanoua1",
18     "USERUsername": "abcde@abcd.com"
19   },
20   {
21     "denumireProiect": "proiect123",
22     "linkRepository": "repo123",
23     "denumireEchipa": "echipa123",
24     "USERUsername": "abckbd@email.com"
25   },
26   {
27     "denumireProiect": "proiect2nou",
28     "linkRepository": "repo2nou",
29     "denumireEchipa": "echipa2nou",
30     "USERUsername": "abckbd@email.com"
31   },
32   {
33     "denumireProiect": "proiectnou1",
34     "linkRepository": "djsknfakdfn",
35     "denumireEchipa": "sdckjfnkfjnqefw",
36     "USERUsername": "abcde@abcd.com"
37   },
38   {
39     "denumireProiect": "proiectnou2",
40     "linkRepository": "reponou2",
41     "denumireEchipa": "echipanoua2",

```

## ▪ Tabela *BUGS*

➤ GET

```
GET http://3.17.177.8:8080/api/user • GET http://3.17.177.8:8080/api/bug • POST http://3.135.204.60:8080/api/u... • POST http://3.135.204.60:8080/api/p... • GET http://3.17.177.8:8080/api/proj... • + • • • No Environment

GET http://3.17.177.8:8080/api/bug Send Save

Pretty Raw Preview Visualize BETA JSON

166 {
167   "descriere": "dasdfafa",
168   "link": "fwefefw",
169   "PROJECTDenumireProject": "projectuser1"
170 },
171 {
172   "denumireBug": "Bug-ul 3 inregistrat de user-ul abcde@abcd.com pentru proiectul : proiect123",
173   "severitate": "Ignorable",
174   "prioritate": "LOW",
175   "descriere": "fvsdfvfvs",
176   "link": "wengeradvsdsva",
177   "PROJECTDenumireProject": "proiect123"
178 },
179 {
180   "denumireBug": "Bug-ul 3 inregistrat de user-ul username1@user.com pentru proiectul : proiectuser1",
181   "severitate": "Ignorable",
182   "prioritate": "LOW",
183   "descriere": "dasdfafa",
184   "link": "fwefefw",
185   "PROJECTDenumireProject": "projectuser1"
186 },
187 {
188   "denumireBug": "Bug-ul 4 inregistrat de user-ul username1@user.com pentru proiectul : proiectuser1",
189   "severitate": "Ignorable",
190   "prioritate": "LOW",
191   "descriere": "dasdfafa",
192   "link": "fwefefw",
193   "PROJECTDenumireProject": "projectuser1"
194 },
195 {
196   "denumireBug": "Bug-ul 5 inregistrat de user-ul abcde@abcd.com pentru proiectul : proiectnou1",
197   "severitate": "Relative",
198   "prioritate": "MEDIUM",
199   "descriere": "Bugpentruproiectnou1",
200   "link": "http://bugnou1",
201   "PROJECTDenumireProject": "proiectnou1"
202 },
203 {
204   "denumireBug": "Bug-ul 5 inregistrat de user-ul username1@user.com pentru proiectul : proiectuser1",
205   "severitate": "Ignorable",
206   "prioritate": "LOW",
207   "descriere": "dasdfafa",
208   "link": "fwefefw",
209   "PROJECTDenumireProject": "projectuser1"
210 },
211 {
212   "denumireBug": "den1",
213   "severitate": "Important",
214   "prioritate": "HIGH",
215   "descriere": "descrierea bugului1",
216   "link": "http://lgjnsfkjns",
217   "PROJECTDenumireProject": "proiect2nou"
218 }
```

### 3.3. Stocare persistentă

Baza de date folosită este una relațională, accesul la aceasta făcându-se prin intermediul unui ORM, și anume *Sequelize*.

În cele ce urmează, vom evidenția structura tabelelor folosite, precum și validările pe câmpurile acestora.

#### **Tabela *USERS***

Tabela *USERS* conține datele personale ale utilizatorilor, fie aceștia membri de proiect (MP) sau tester (TST) – această diferențiere se face în câmpul *statut*. Aceste informații vor fi stocate în baza de date, în momentul în care utilizatorul se înregistrează în aplicație, completând formularul de *Register*. Apoi, acesta se poate autentifica în formularul de *Login*, pentru a putea utiliza aplicația.

Această tabelă se află în legătură 1-m cu tabela *PROJECTS*.

COLUMNS	TYPE
USERNAME ( <i>PK</i> )	STRING
NUME	STRING
PRENUME	STRING
PASSWORD	STRING
STATUT	STRING

De asemenea, tabela are validatori pe următoarele câmpuri:

- ✓ username: trebuie să fie obligatoriu completat, să aibă între 4 și 50 caractere și să respecte formatul unei adrese de email
- ✓ nume: trebuie să fie obligatoriu completat și să conțină caractere alpha
- ✓ prenume: trebuie să fie obligatoriu completat, să conțină caractere alpha și să aibă între 2 și 15 caractere
- ✓ password: trebuie să fie obligatoriu completat, să aibă prima literă majusculă și să aibă minim 5 caractere

### **Tabela *PROJECTS***

Tabela *PROJECTS* conține informațiile despre proiect, și anume denumirea acestuia, link-ul repository-ului în care se găsește respectivul proiect și echipa de proiect asignată.

Această tabelă se află în legătură 1-m cu tabelele *USERS* și *BUGS*.

COLUMNS	TYPE
DENUMIREPROIECT ( <i>PK</i> )	STRING
LINKREPOSITORY	STRING
DENUMIREECHIPA	STRING

De asemenea, tabela are validatori pe următoarele câmpuri:

- ✓ denumireProiect: trebuie să fie obligatoriu completat și să aibă între 3 și 50 caractere
- ✓ linkRepository: trebuie să fie obligatoriu completat și să aibă între 3 și 60 caractere



- ✓ denumireEchipa: trebuie să fie obligatoriu completat și să aibă între 3 și 50 caractere

### **Tabela BUGS**

Tabela *BUGS* conține informațiile despre bug-urile înregistrate de către testeri, și anume denumirea acestuia, severitatea, prioritatea, descrierea fiecăruia dintre bug-uri și link-ul repository-ului în care a fost încărcat.

Această tabelă se află în legătură 1-m cu tabela *PROJECTS*.

COLUMNS	TYPE
DENUMIREBUG ( <i>PK</i> )	STRING
SEVERITATE	STRING
PRIORITATE	STRING
DESCRIERE	STRING
LINK	STRING

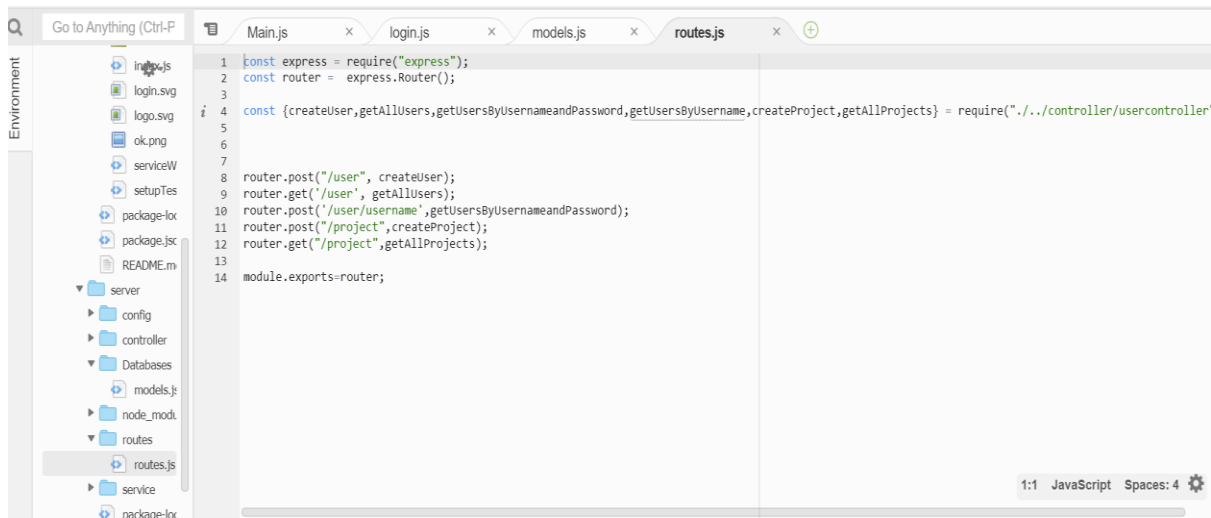
De asemenea, tabela are validatori pe următoarele câmpuri:

- ✓ denumireBug: trebuie să fie obligatoriu completat
- ✓ severitate: trebuie să fie obligatoriu completat
- ✓ prioritate: trebuie să fie obligatoriu completat și să aibă între 3 și 8 caractere
- ✓ descriere: trebuie să fie obligatoriu completat și să aibă între 5 și 20 caractere
- ✓ link: trebuie să fie obligatoriu completat și să aibă între 5 și 60 caractere

### **3.4. Rutare (*Single Page Application*)**

Aplicația web este realizată în așa manieră încât interacționează cu utilizatorul, rescriind în mod dinamic pagina curentă în loc de a încărca noi pagini de pe server. Această abordare evită întreruperea experienței utilizatorului între paginile succesive, făcând ca aplicația să se comporte mai mult ca o aplicație desktop.

O mică parte din implementare este prezentată în cele ce urmează.



## 4. Implementare funcționalități front-end

Framework-ul folosit în realizarea front-end-ului este *React.js*. React este o bibliotecă JavaScript pentru construirea interfețelor utilizatorului. Aceasta este folosită ca bază în dezvoltarea aplicațiilor mobile sau a aplicațiilor single-page.

- Formular Login

Login

Username

Email Address

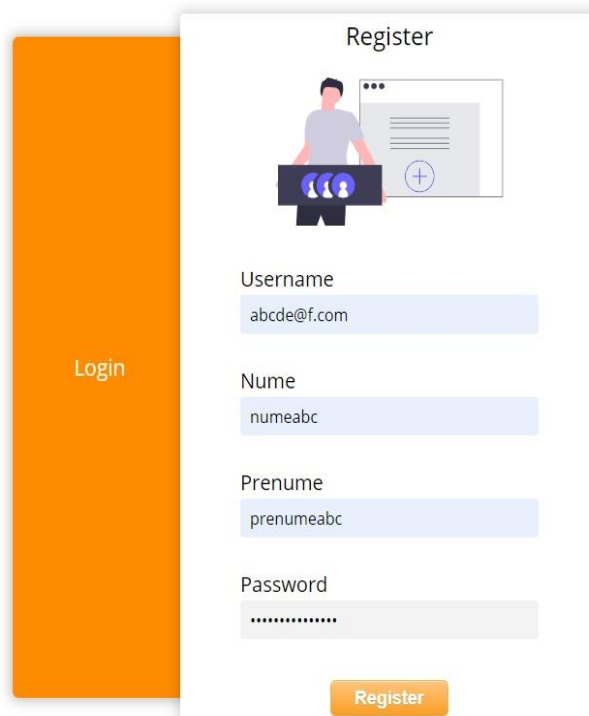
Password

Password

Login

Register

- Formular Register



The image shows a 'Register' form overlaying a large orange 'Login' button. The 'Register' form has a title 'Register' and an illustration of a person at a computer. It contains the following fields:

- Username:** abcde@f.com
- Nume:** numeabc
- Prenume:** prenumeabc
- Password:** (masked with dots)

At the bottom of the form is an orange 'Register' button.

- Tester Interface

---



The 'Tester Interface' contains three main sections:

- Inregistrare Proiect** (Project Registration)
- Adaugare ca tester** (Add as tester)
- Inregistrare Bug** (Register Bug)

Below the first two buttons is a button labeled **Vizualizare Proiecte Disponibile** (View Available Projects). This leads to a table of projects:

2hwertwr
dsfasdf
ngiergeritg
proiect123
proiect2nou
proiectnou1
proiectnou2
proiectu1
proiectu134
proiectuser1
proiectwefw
proj1
rwrtwefwergwegwew

Below the table is a field: Denumire proiect: proiectuser1, followed by a green checkmark icon.

The **Inregistrare Bug** section includes:

- Proiect: **proiectuser1**
- Descriere:
- Link:
- Severity: **LOW** (dropdown menu)
- Priority: **Ignorable** (dropdown menu)
- Inregistreaza Bug** (orange button)

---

- Project Manager Interface

Inregistrare Proiect

Adaugare ca tester


Vizualizare Bug-uri inregistrate

Modificare Proiect

Denumire proiect:

Repository:

Denumire echipa:



Lista proiecte pentru user-ul : abcde@abcd.com

2twertwr
nglgergitg
proiectnou1
proiectnou2
proiectu134
rwrwetwergwergwegwer

**Modificare proiect**

Denumire proiect: **2twertwr**

Repository:

Denumire echipa:

Modificare Proiect

Lista bug-uri pentru MP-ul : abcde@abcd.com

Bug-ul 1 inregistrat de user-ul username1@user.com pentru proiectul : 2twertwr
Bug-ul 1 inregistrat de user-ul username1@user.com pentru proiectul : nglergitg
Bug-ul 2 inregistrat de user-ul username1@user.com pentru proiectul : nglergitg
Bug-ul 5 inregistrat de user-ul abcde@abcd.com pentru proiectul : proiectnou1
Bug-ul 1 inregistrat de user-ul username1@user.com pentru proiectul : proiectu134
Bug-ul 1 inregistrat de user-ul username1@user.com pentru proiectul : rwrwetwergwergwegwer

O parte din codul necesar pentru proiectarea front-end-ului este reprezentat în imaginea următoare:

```

250 App.js | register.js | login.js | Main.js | server.js | models.js | index.js | routes.js | users.js | users.js | project.js | bug.js | App.scss | index.css | style.css |
251
252 <div style={{ [marginLeft: "50px", marginLeft: "150px"]}}><div><div style={{ height: "50px" }}>Inregistrare Bug</div></div>
253 </div>
254 </div>
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314

```

## 5. Concluzii

Aplicația web proiectată reprezintă un pas foarte important în dezvoltarea unei platforme și mai complexe, care să urmărească rezolvarea fiecărui aspect ce poate apărea în viața de zi cu zi.

## Bibliografie

- ❖ <https://developer.mozilla.org/en-US/>
- ❖ <https://sequelize.org/>