

Documentatie: “Gestiune a activitatilor dintr-un spital”



Cuprins

1. ENUNT TEMA:.....	3
2. TEHNOLOGIA FOLOSITA PENTRU IMPLEMENTARE:	4
3. DESCRIEREA INTERFETEI CU UTILIZATORUL:	5
4. STRUCTURA BAZEI DE DATE:	16
5. DESCRIEREA SCRIPTURILOR:.....	17
6. CUM POATE FI DEZVOLTA APLICATIA IN VIITOR:.....	20
7. BIBLIOGRAFIE:.....	21

1. Enunt tema:

Să se scrie o aplicație de „Gestiune a activităților dintr-un spital” în tehnologie servlets/JSP cu interfață grafică. Aplicația va gestiona o bază de date cu următoarele tabele:

Medici – cod identificare unic, Nume și prenume, Specializare medicală, cod secție unde activează;

Secții spital- cod identificare unic, Nume secție, număr de paturi, cod specializare medicală

Pacienți – cod identificare unic, Nume și prenume, cod diagnostic, cod secție internare, cod medic responsabil

Fișe observații – cod identificare unic, cod pacient asociat, cod diagnostic pacient, cod medic alocat, cod secție internare, data internare, data externare

Diagnostic – cod identificare unic, nume diagnostic, cod secție asociată, cod specializare medicală asociată

Baza de date folosită în cadrul aplicației va fi de tip MySQL.

Aplicația va permite:

- Pe cont **administrator** - introducerea datelor în tabele, afișarea și modificarea datelor din tabele precum și stergerea acestora;
 - Pe cont de tip **registratura** internari cu nume și parolă date – creare fișă de observații cu date pacient, data internare, diagnostic internare, secție alocată, medic alocat
 - Pe cont de tip **medic** cu nume și parolă date – completare date de tip diagnostic revizuit în spital, data externare (dacă pacientul este externat).
-

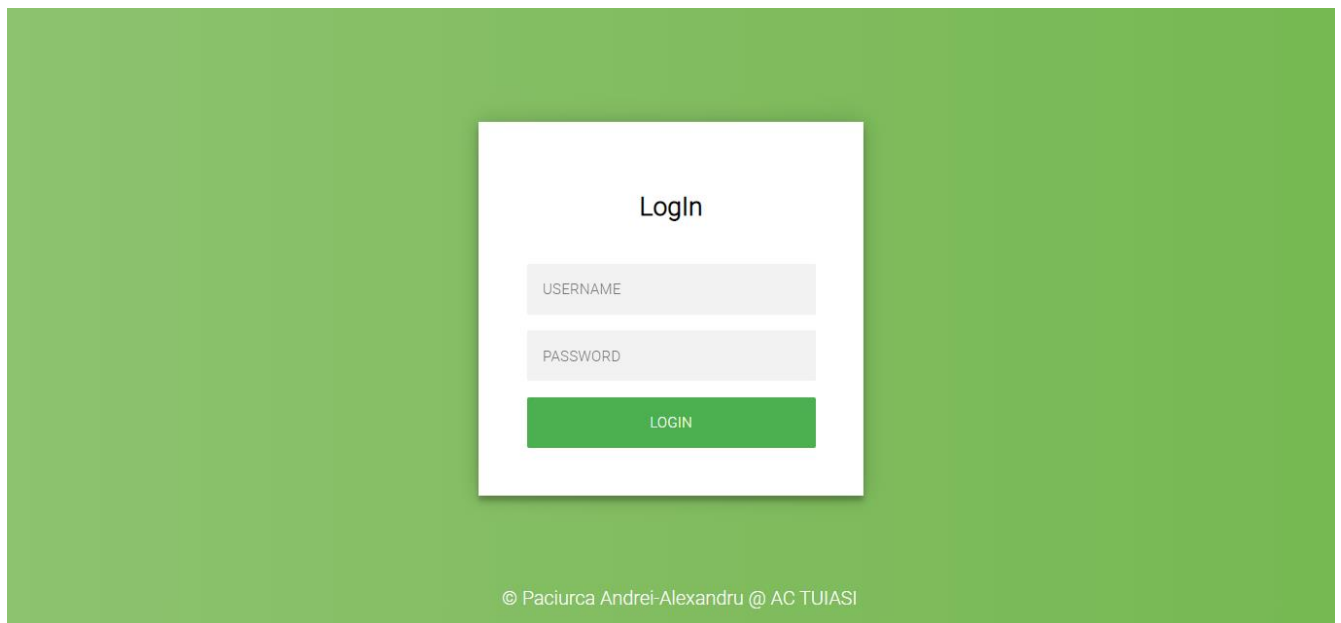
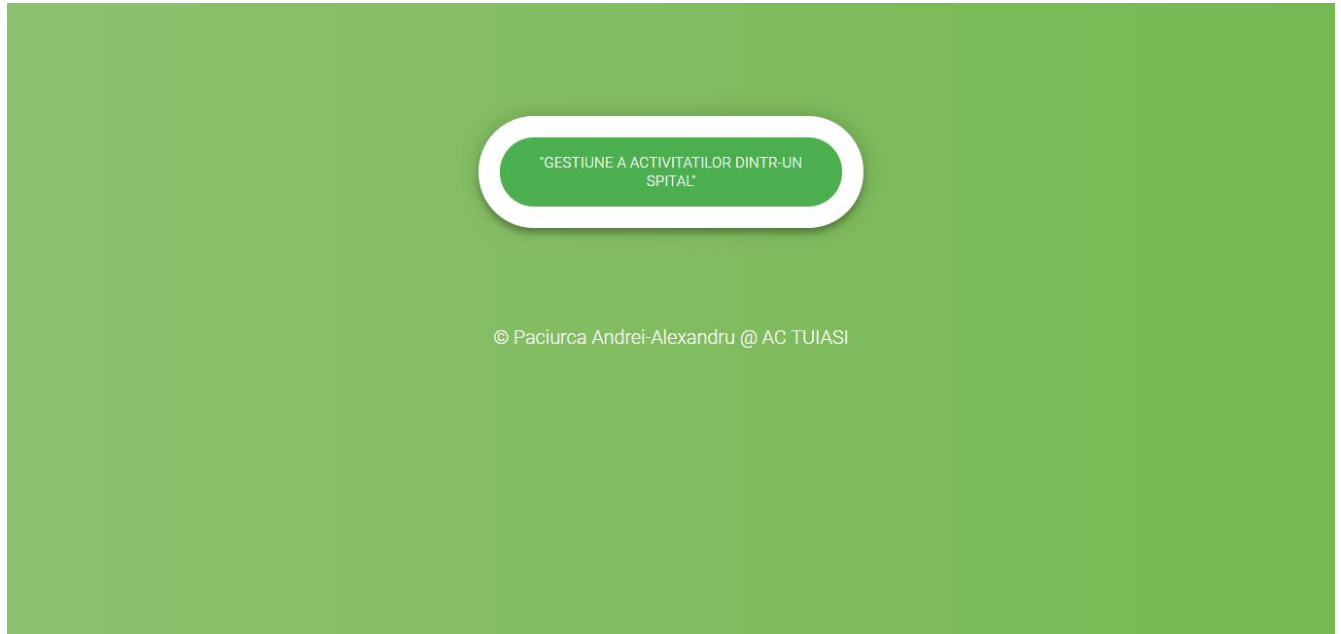
2. Tehnologia folosita pentru implementare:

In elaborarea acestui proiect s-au utilizat urmatoarele tehnologii si programe: Apache Tomcat, WAMP, MySQL DB, **Java Server Pages**, Visual Studio Code, HTML, CSS, Eclipse IDE.

- Apache Tomcat – reprezinta o aplicatie ce implementeaza un server local (dezvoltat pur in Java) pentru rularea, executarea, afisarea si interactiunea a multor tehnologii web si server side.
- WAMP (**Windows Apache MySQL & PHP**) – reprezinta o suita de aplicatii ce incorporeaza mai multe tehnologii; in cadrul acestui proiect s-a utilizat doar pentru partea de crearea si conectare la baza de date „spital” utilizat pentru gestiunea si stocarea datelor din aplicatie.
- MySQL – reprezinta tipul de baza de date (cel mai popular si usor de gestionat) pe care a fost construita aplicatia.
- **Java Server Pages** – reprezinta o „subtehnologie” din cadrul JDK pentru a lucra si interactiona cu partea de server side si client side.
- Visual Studio Code – este un text editor (open-source) dezvoltat de Microsoft ce suporta toate limbajele de programare importante si prezinta numeroase plugin-uri utile (am folosit Live Editor pentru conceperea in timp real a elementelor statice de tip HTML si CSS)
- HTML (**HyperText Markup Language**) - un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afisate intr-un browser.
- CSS (**Cascading Style Sheets**) - este un standard pentru formatarea elementelor unui document HTML.
- Eclipse IDE – mediul principal de dezvoltare pentru aplicatia curenta ce inglobeaza toate uneltele folosite mai sus (run on server, JSP+HTML+CSS, conectarea cu Apache si baza de date MySQL).

3. Descrierea interfetei cu utilizatorul:

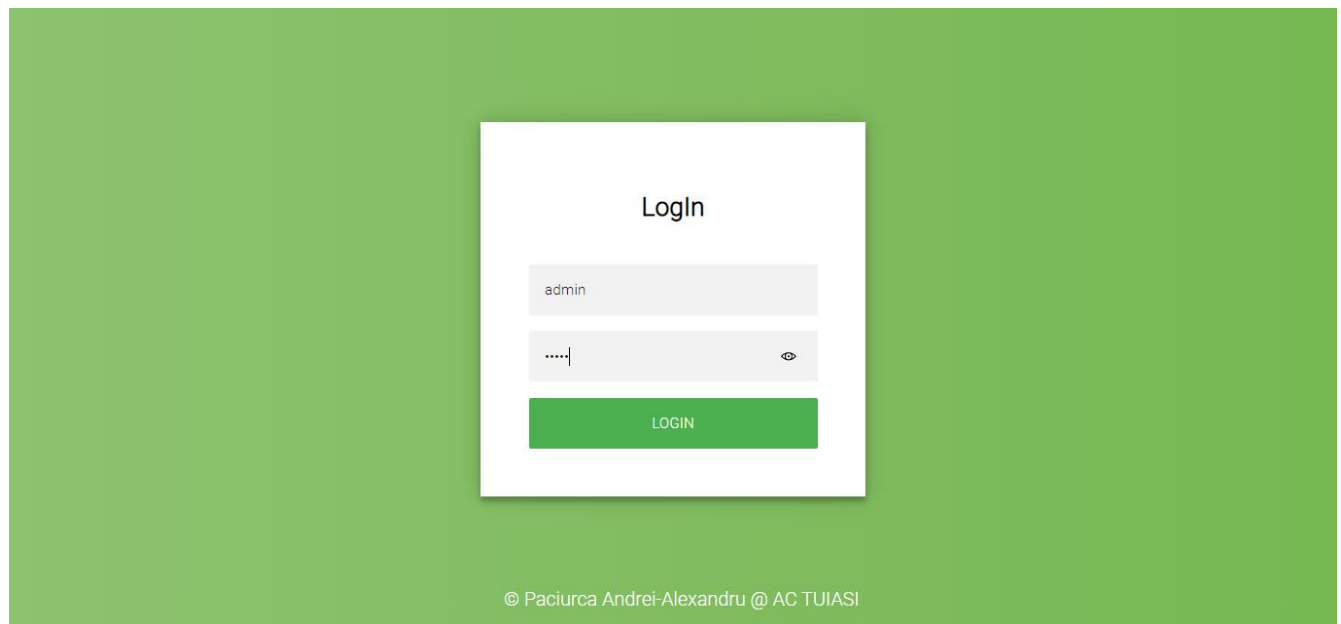
- **Partile comune:** pagina de index/welcome si pagina/formularul de logare.



La prima accesare a site-ului utilizatorul este intampinat de pagina *index.jsp* care contine un singur buton ce nu are un rol functional, altul decat al unui simplu *redirect* spre *formularul de login* pentru a putea intra in aplicatia propriu-zisa.

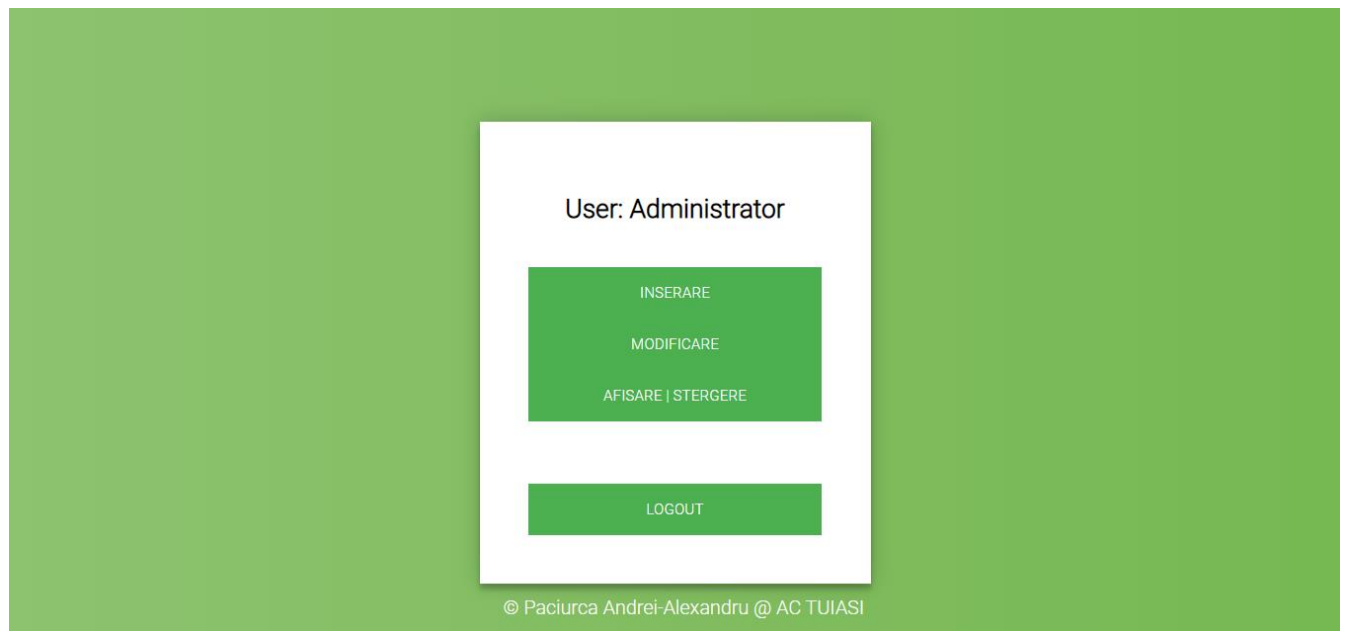
Odata apasat butonul ce contine denumirea aplicatiei, de pe pagina de index, utilizatorul va fi redirectionat spre pagina ce contine cateva elemente simple si necesare de HTML: un *div* (utilizat cu scopul de a formata prin CSS mai multe elemente deodata), un *formular* cu doua campuri de tip input (unul de tip *text*, iar celalalt de tip *password*) si un input de tip *buton* submit cu denumirea „*LOGIN*”.

- Utilizatorul **administrator** (USERNAME = *admin*; PASSWORD = *admin*):



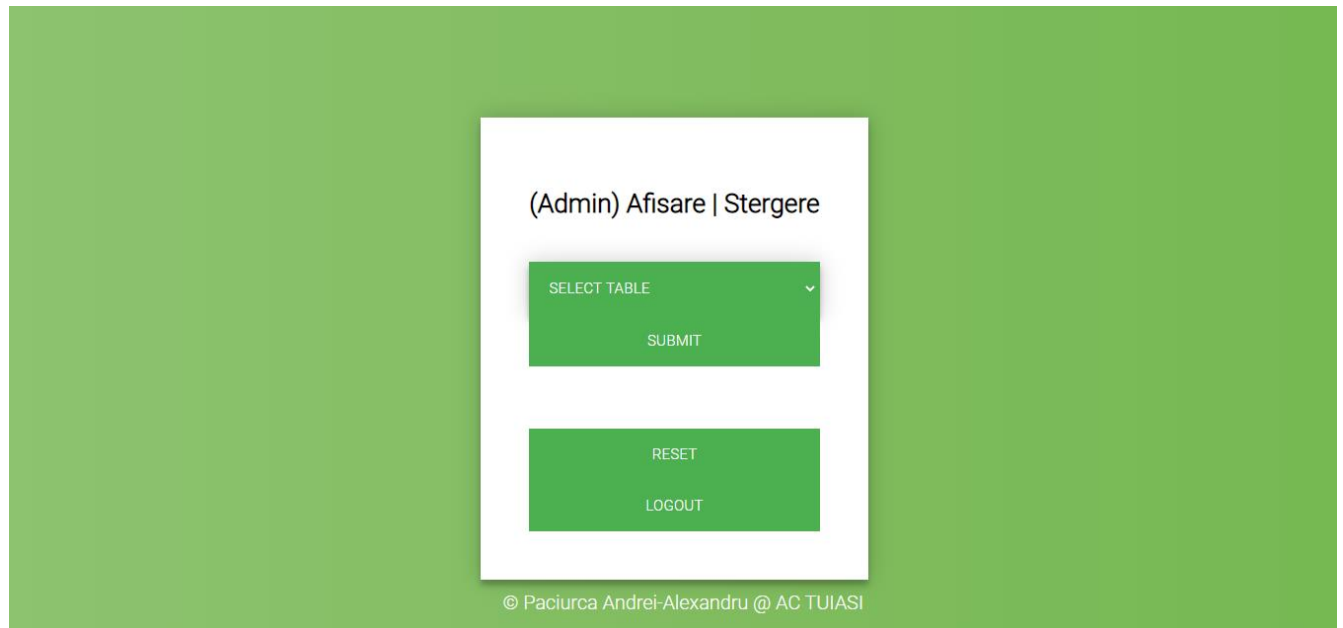
The screenshot shows a login form titled "Login" centered on a green background. The form is a white box containing two input fields: the first is a text field with "admin" entered, and the second is a password field with "...." and a toggle eye icon. Below these is a green "LOGIN" button. At the bottom of the green background, there is a copyright notice: "© Paciorca Andrei-Alexandru @ AC TUIASI".

Meniul principal/Optiunile contului de *administrator*:



The screenshot shows a menu titled "User: Administrator" centered on a green background. The menu is a white box containing three green buttons stacked vertically: "INSERARE", "MODIFICARE", and "AFISARE | STERGERE". Below these is a separate green button labeled "LOGOUT". At the bottom of the green background, there is a copyright notice: "© Paciorca Andrei-Alexandru @ AC TUIASI".

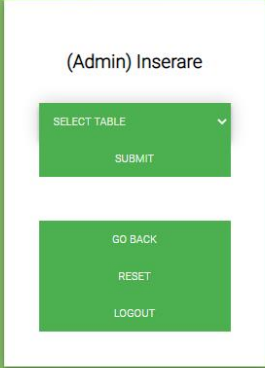
Optiunea de Afisare | Stergere corespunzatoare contului *administrator*:



Se va selecta din meniul de tip „drop-down” tabelul dorit, iar apoi utilizatorul va fi redirectionat spre o alta pagina ce se ocupa cu afisarea/stergerea propriu-zisa a randurilor din tabel:



Optiunea de Inserare corespunzatoare contului de *administrator*:



(Admin) Inserare

SELECT TABLE ▼

SUBMIT

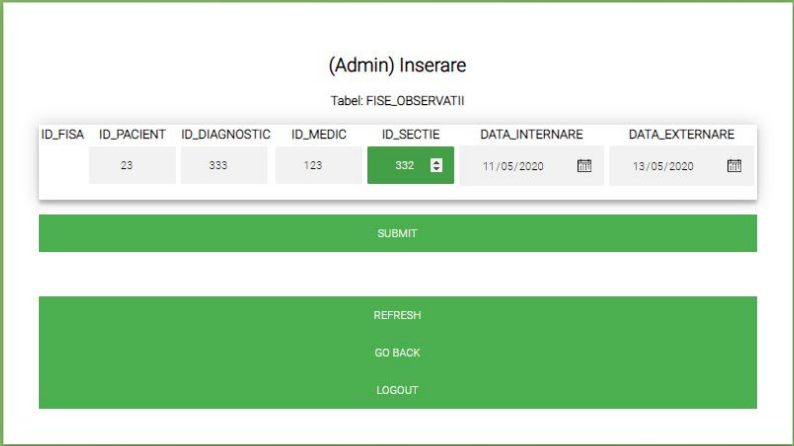
GO BACK

RESET

LOGOUT

© Paciorca Andrei-Alexandru @ AC TUIASI

Dupa selectarea tabelului dorit, utilizatorul va fi intampinat de urmatoare structura ce este bazata pe un element de tip `<table>` si multe elemente de tip `<input>` ce se adapteaza in functie de tabel (input-uri de tip: hidden, number, datetime-local, text, radio, checkbox):



(Admin) Inserare

Tabel: FISE_OBSERVATII

ID_FISA	ID_PACIENT	ID_DIAGNOSTIC	ID_MEDIC	ID_SECTIE	DATA_INTERNARE	DATA_EXTERNARE
	23	333	123	332	11/05/2020	13/05/2020

SUBMIT

REFRESH

GO BACK

LOGOUT

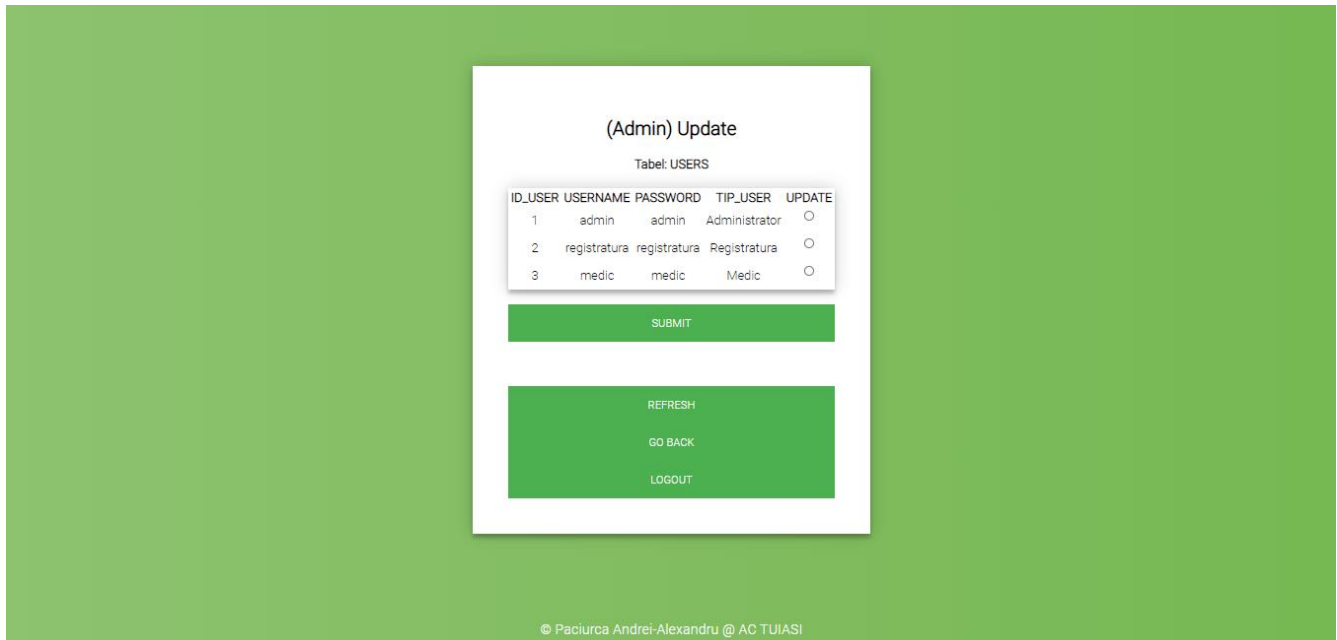
Optiunea de Modificare corespunzatoare contului de *administrator*:



The screenshot shows a web form titled "(Admin) Modificare" centered on a green background. The form is a white rectangle containing a dropdown menu labeled "SELECT TABLE" with a downward arrow. Below the dropdown is a green "SUBMIT" button. Further down is another green button labeled "GO BACK". At the bottom of the form are two more green buttons labeled "RESET" and "LOGOUT".

© Paciurca Andrei-Alexandru @ AC TUIASI

Dupa selectarea tabelului dorit, utilizatorul va fi pus sa decida (printr-un grup de butoane radio) ce rand din tabel doreste sa modifice:



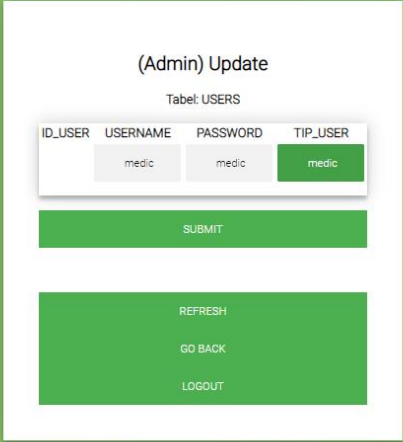
The screenshot shows a web form titled "(Admin) Update" centered on a green background. The form is a white rectangle. Below the title is the text "Tabel: USERS". Underneath is a table with the following data:

ID_USER	USERNAME	PASSWORD	TIP_USER	UPDATE
1	admin	admin	Administrator	<input type="radio"/>
2	registratura	registratura	Registratura	<input type="radio"/>
3	medic	medic	Medic	<input type="radio"/>

Below the table is a green "SUBMIT" button. At the bottom of the form are three green buttons labeled "REFRESH", "GO BACK", and "LOGOUT".

© Paciurca Andrei-Alexandru @ AC TUIASI

Dupa ce a fost ales randul ce urmeaza a fi modificat utilizatorului va fi intampinat de o noua pagina ce contine input-urile corespunzatoare pentru modificare cat si valorile anterioare. Acesta (user-ul) poate decide daca nu modifica nimic, daca modifica o parte din campuri, sau pe toate (cu exceptia ID-ului randului respectiv din motive de pastrare a integritatii bazei de date).



(Admin) Update

Tabel: USERS

ID_USER	USERNAME	PASSWORD	TIP_USER
	medic	medic	medic

SUBMIT

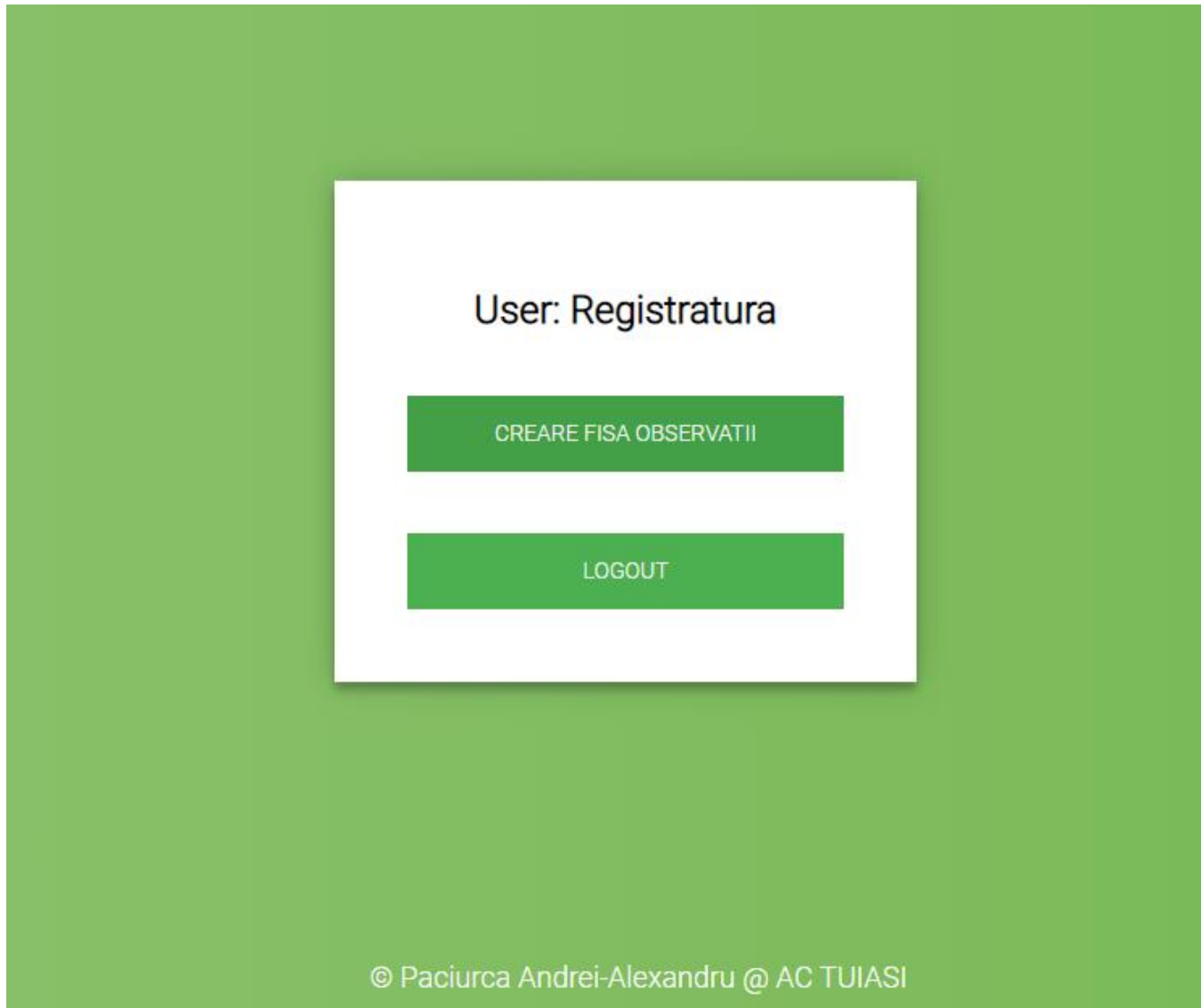
REFRESH

GO BACK

LOGOUT

© Paciurca Andrei-Alexandru @ AC TUIASI

- Utilizator **registratura** (USERNAME: *registratura*; PASSWORD: *registratura*):



Singura optiune disponibila user-ului de registratura, conform enuntului, este de a crea o noua fisa de observatie.

Imediat dupa apasarea butonului „CREARE FISA OBSERVATII” acest va fi redirectionat pe pagina de insemnarea a unei noi fise. Mai exista sectia „DATE UTILE” unde sunt afisate tabelele „Medici” si „Pacienti” menite sa-l ajute pe utilizatorul *registratura* (totodata pentru pastrarea unei organizari bune si a unei integritati a bazei de date).

(Registratura) Creare Fisa Observatii

Tabel: FISE_OBSERVATII

ID_FISA	ID_PACIENT	ID_DIAGNOSTIC	ID_MEDIC	ID_SECTIE	DATA_INTERNARE	DATA_EXTERNARE
					22/05/2020 17:19:35,367	22/05/2020 17:19:35,367

SUBMIT

DATE UTILE:

ID_MEDIC	NUME	PRENUME	ID_SPECIALIZARE	ID_SECTIE
1	Miron	Ioan	1	1
2	Maizel	Alexandru	2	2
3	Trifan	Radu-Constantin	4	4
4	Codreanu	Paula-Irina	3	3

ID_PACIENT	NUME	PRENUME	ID_DIAGNOSTIC	ID_SECTIE
1	Paciurca	Andrei-Alexandru	2	2
2	Iacinshi	Vlad	1	1
3	Arama	Andra	1	1
4	Dinga	Malina	1	1
5	Turcu	Andreea	3	3
6	Mihaila	Octavian-Dumitru	2	2

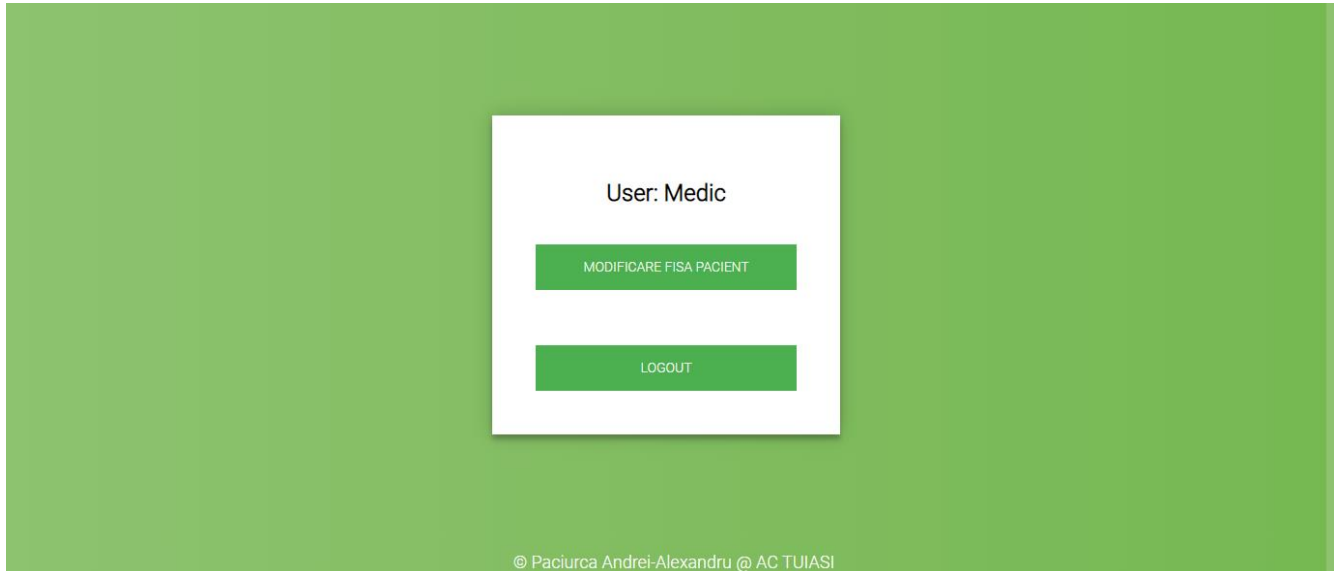
REFRESH

GO BACK

LOGOUT

© Paciurca Andrei-Alexandru @ AC TUIASI

- Utilizator **medic** (USERNAME: *medic*; PASSWORD: *medic*):



Singura optiune disponibila, conform enuntului, estea aceea de a revizui diagnosticul pacientului si de modificare a datelor de internare/externare (adica update pe campurile unui pacient specific)

Dupa apasarea butonului, se va afisa tot continutul tabelului „*Fise_Observatii*” si posibilitatea de a selecta randul/pacientul pentru modificare/revizuire.

(Medic) Update

Tabel: FISE_OBSERVATII

ID_FISA	ID_PACIENT	ID_DIAGNOSTIC	ID_MEDIC	ID_SECTIE	DATA_INTERNARE	DATA_EXTERNARE	UPDATE
1	1	2	2	2	2020-05-20 08:09:08.0	2020-05-23 14:12:08.0	<input type="radio"/>
2	2	1	1	1	2020-05-22 14:13:40.0	2020-05-22 14:13:40.0	<input type="radio"/>
3	3	1	1	1	2020-05-22 14:14:18.0	2020-05-22 14:14:18.0	<input type="radio"/>
4	3	1	1	1	2020-05-22 14:14:31.0	2020-05-22 14:14:31.0	<input type="radio"/>

SUBMIT

REFRESH

GO BACK

LOGOUT

Student: Paciurca Andrei-Alexandru
Grupa: 1304A

Dupa ce a fost selectat randul/pacientul ce urmeaza a fi modificat, medicul poate face update doar pe coloanele ID_DIAGNOSTIC, DATA_EXTERNARE (am inclus si eu DATA_INTERNARE):



(Medic) Update

Tabel: FISE_OBSERVATII

ID_DIAGNOSTIC	DATA_INTERNARE	DATA_EXTERNARE
1	2020-05-22 1	2020-05-22 1

SUBMIT

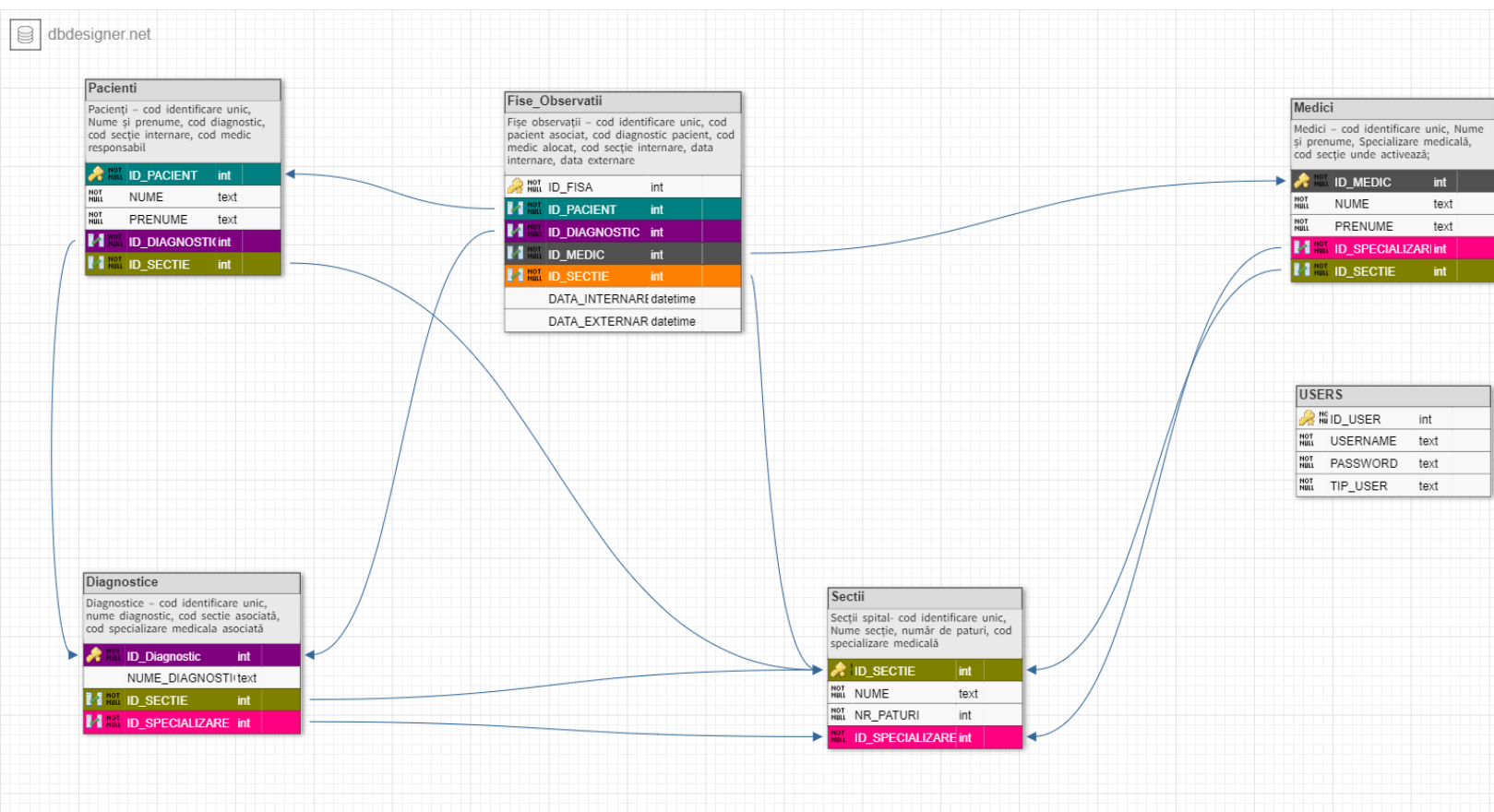
REFRESH

GO BACK

LOGOUT

© Paciurca Andrei-Alexandru @ AC TUIASI

4. Structura bazei de date:



5. Descrierea scripturilor:

- Pentru optiunea de DELETE s-a folosit urmatoare bucata script:

```

1.  ResultSet rs = st.executeQuery("select * from " + session.getAttribute("table"));
2.      ResultSetMetaData rsmd = rs.getMetaData();
3.
4.      String ID = rsmd.getColumnName(1);
5.
6.      String[] select = request.getParameterValues("idCheckBox");
7.      if (select != null && select.length != 0) {
8.          for (String delete : select){
9.              st.executeUpdate("DELETE from "+session.getAttribute("table")+ " where "+ID+"="+
delete);
10.         }
11.     }

```

- Pentru *inserarea* din administrator s-a folosit:

```

1.  ResultSet rs = st.executeQuery("select * from " + session.getAttribute("tableInsert"));
2.      ResultSetMetaData rsmd = rs.getMetaData();
3.
4.      String columns = " ( ";
5.      for (int i = 1; i <= rsmd.getColumnCount(); i++) {
6.          if(i<rsmd.getColumnCount())
7.              columns = columns + rsmd.getColumnName(i).toString() + " , ";
8.          else
9.              columns = columns + rsmd.getColumnName(i).toString() + " ) ";
10.     }
11.
12.      String values = " ( ";
13.      String[] toInsert = request.getParameterValues("insertInput"); //date coloane pt UPDATE
14.      if (toInsert != null && toInsert.length != 0) {
15.          for (int i = 1; i <= rsmd.getColumnCount(); i++) {
16.              if(i<rsmd.getColumnCount())
17.                  values += "'" + toInsert[i-1]+ "'" + " , ";
18.              else
19.                  values += "'" + toInsert[i-1]+ "'" + " ) ";
20.          }
21.      }
22.
23.      int update = st.executeUpdate("INSERT INTO "+session.getAttribute("tableInsert")+ " "+columns+"
VALUES "+values);
24.      if(update > 0){
25.          response.sendRedirect("admin_inserare.jsp");
26.      }
27.      else{
28.          response.sendRedirect("logout.jsp");
29.      }

```

- Pentru *modificarea*/update-ul datelor din tabele de pe contul de *administrator* s-a folosit:

```
1. ResultSet rs = st.executeQuery("select * from " + session.getAttribute("tableUpdate").toString()  
   );  
2.   ResultSetMetaData rsmd = rs.getMetaData();  
3.  
4.   String[] toInsert = request.getParameterValues("updateInput");  
5.  
6.   for (int i = 1; i <= rsmd.getColumnCount(); i++) {  
7.       int update = st.executeUpdate("UPDATE "+session.getAttribute("tableUpdate").toString()+" SET  
       "+rsmd.getColumnName(i)+" = ' "+toInsert[i-  
       1]+"' WHERE "+rsmd.getColumnName(1)+"="+toInsert[0]);  
8.  
9.       if(update <= 0){  
10.           response.sendRedirect("logout.jsp");  
11.       }  
12.   }  
13. response.sendRedirect("admin_select_update.jsp");
```

- Pentru update-ul ce se produce la apasarea butonului „Modificare fisa pacient” din cadrul contului de *medic*:

```
1.   String[] toInsert = request.getParameterValues("updateInput");  
2.  
3.       int update = 0;  
4.       update = st.executeUpdate("UPDATE fise_observatii SET " +  
5.       "ID_DIAGNOSTIC = ' "+toInsert[0]+"' , "+  
6.       "DATA_INTERNARE = ' "+toInsert[1]+"' , "+  
7.       "DATA_EXTERNARE = ' "+toInsert[2]+"' "+  
8.       " WHERE "+rsmd.getColumnName(1)+"="+session.getAttribute("IDrow"));  
9.  
10.      if(update <= 0)  
11.          response.sendRedirect("logout.jsp");  
12.  
13.      response.sendRedirect("medic_homepage.jsp");
```

- Pentru inserarea corespunzatoare butonului „Creare fisa observatie” de pe contul de *registratura*:

```

1. String columns = " ( ";
2.     for (int i = 1; i <= rsmd.getColumnCount(); i++) {
3.         if(i<rsmd.getColumnCount())
4.             columns = columns + rsmd.getColumnName(i).toString() + " , ";
5.         else
6.             columns = columns + rsmd.getColumnName(i).toString() + " ) ";
7.     }
8.
9.     String values = " ( ";
10.    String[] toInsert = request.getParameterValues("insertRegInput");
11.    if (toInsert != null && toInsert.length != 0) {
12.        for (int i = 1; i <= rsmd.getColumnCount(); i++) {
13.            if(i<rsmd.getColumnCount())
14.                values += "'" + toInsert[i-1] + "' + " , ";
15.            else
16.                values += "'" + toInsert[i-1] + "' + " ) ";
17.        }
18.    }
19.
20.    int update = st.executeUpdate("INSERT INTO fise_observatii "+columns+" VALUES "+values);
21.
22.    if(update > 0){
23.        response.sendRedirect("registratura_homepage.jsp");
24.    }
25.    else{
26.        response.sendRedirect("logout.jsp");
27.    }

```

- Conectarea la baza de date *spital*:

```

1. Class.forName("com.mysql.jdbc.Driver");
2. Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/spital","root", "");
3. Statement st = con.createStatement();

```

- Exemplu de afisare sub forma tabelara:

```

1. <table>
2.     <tr>
3.         <% for (int i = 1; i <= columnCount; i++)
4.             out.println("<th>"+rsmd.getColumnName(i)+"</th>");
5.             out.println("<th>DELETE</th>"); %>
6.     </tr>
7.     <% String valueRowID = "";
8.         while(rs.next()){
9.             out.println("<tr>");
10.            for (int i = 1; i <= columnCount; i++) {
11.                out.println("<td>"+rs.getString(i)+"</td>");
12.                if(i==1) //prima coloana -> cheia primara a tabelului: ID_(text)
13.                    valueRowID = rs.getString(i);
14.            }
15.            out.println("<td><input type='checkbox' name='idCheckBox' value='"+valueRowID+"' /><td>");
16.            out.println("</tr>");
17.        } %>
18. </table>

```

6. Cum poate fi dezvoltata aplicatia in viitor:

- Mentiuni importante:

Aceasta aplicatie are scop didactic si pur demonstrativ. Nu este perfecta din niciun punct de vedere. Pe tot parcursul dezvoltarii acesteia, aplicatia a fost testata in browserele: **Google Chrome** (v 83), **Microsoft Edge** (v 83, chromium based) si browserul intern inclus in aplicatia Eclipse IDE.

- Aplicatia poate fi dezvoltata in urmatoarele directii:

- Securitate: se pot face o multitudine de aspecte pe partea de securitate a aplicatiei. In primul rand s-ar putea stoca conturile (cu user si parola) intr-o baza de date separata, cel mai probabil criptate cu metoda/algoritmul SHA-256 si eliminarea posibilitatii de tip atac SQL-Injection.
- Robustete/organizare: s-ar putea considera reorganizarea fisierelor pe foldere si subfoldere, alegerea unui tipar de numire/redenumire care sa ramana consecvent, generalizarea functiilor de afisare, insert, delete, update, etc; cu alte cuvinte aplicarea principiilor de OOP cu accent pe abstractizare si incapsulare.
- Reorganizarea bazei de date: o baza de date bine gandita si organizata corect de la bun inceput (treaba unui administrator si designer de baze de date) aplicatia ramane consecventa si usor de manipulat prin cod indiferent la scara la care este folosita (locala, nationala, globala); ma gandesc in mod special la introducerea de relatii cat mai complexe si logice intre tabele, chei externe, chei primare multiple, triggeri si elemente de limbaj PL-SQL.

- Concluzii:

Putem afirma faptul ca exista un vast pachet de tehnologii destinate dezvoltarii de aplicatii web, si de aceea alegerea lor se va face in functie de complexitatea aplicatiei ce urmeaza a fi dezvoltata, in functie de fonduri si nu in ultimul rand in functie de programator.

Prin intermediul acestui proiect am reusit sa-mi insusesc mai bine cunostiintele in limbajul de programare Java, sa stiu sa lucrez cu fisiere de tip JSP, HTML, CSS, SQL.

7. Bibliografie:

- Cea mai importanta sursa de inspiratie: <https://stackoverflow.com/>
- Resursele si tutorialele puse la dispozitie de W3School: <https://www.w3schools.com/>
- Pagina de start a aplicatiei: <http://localhost:8080/Paciurca-Andrei-Alexandru/>