

Colegiul Național “B.P. Hasdeu” Buzău

**Lucrare pentru obținerea atestatului profesional la
INFORMATICĂ**

Gestiunea unui cabinet medical

**Profesor coordonator,
Olăroiu Nicolae**

**Realizator,
Mărunțiș Ștefan-Andrei
clasa 12A**

mai 2021

Cuprins

1. Prezentarea generală a temei.....	3
2. Resurse hardware și software necesare.....	4
2.1 Scurtă descriere a limbajului PHP	4
2.2 Scurtă prezentarea a serverului de baze de date MySQL	6
2.3 Resurse hardware necesare rulării proiectului.....	7
3. Realizarea aplicației	8
3.1 Structura bazei de date	8
3.2 Descrierea interfeței	12
4. Utilizarea aplicației	27
5. Extinderi posibile ale aplicației.....	31
6. Bibliografie	32

1. Prezentarea generală a temei

Gestiunea eficientă a unui volum mare de date a devenit o necesitate în ultimii ani, întrucât permite creșterea considerabilă a profitului în multe domenii de activitate. Există însă situații în care existența unui program de gestiune a bazei de date poate avea un impact mai mare asupra vieții noastre.

În sistemul medical o situație critică poate fi redresată cu informații corecte la momentul potrivit. Spre exemplu, un medic poate salva viața unui pacient dacă primește informația că acel pacient este alergic la un medicament pe care doctorul voia să i-l administreze. O aplicație de gestiune a bazei de date poate preveni ușor asemenea situații.

În plus, instituțiile medicale nu sunt mereu echipate cu tehnică performantă sau cu personal experimentat în utilizarea calculatoarelor. De aceea, am ales să fac proiectul sub forma unei aplicații web care poate rula pe orice calculator care are instalat un browser, fără necesitatea de a instala în prealabil alte programe. De asemenea, aplicația este echipată cu o interfață ușor de folosit și sugestivă, astfel că oricine poate lucra cu ea.

Aplicația este capabilă să gestioneze toate datele necesare unui cabinet medical, îndeplinind pe deplin nevoile instituției. Așadar, datorită simplității și versatilității, este ideală pentru un cabinet medical mic sau mediu.

2. Resurse hardware și software necesare

2.1 Scurtă descriere a limbajului PHP

PHP este un limbaj de programare. Numele PHP provine din limba engleză și este un acronim recursiv : **Php: Hypertext Preprocessor**. Folosit inițial pentru a produce pagini web dinamice, este folosit pe scară largă în dezvoltarea paginilor și aplicațiilor web. Se folosește în principal înglobat în codul HTML, dar începând de la versiunea 4.3.0 se poate folosi și în mod „linie de comandă” (CLI), permițând crearea de aplicații independente. Este unul din cele mai importante limbaje de programare web open-source și server-side, existând versiuni disponibile pentru majoritatea web serverelor și pentru toate sistemele de operare.

PHP-ul este unul din cele mai folosite limbaje de programare server-side, conform unui studiu efectuat de Netcraft în aprilie 2002, apărând pe 9 din cele 37 milioane de domenii cercetate în studiu. Popularitatea de care se bucură acest limbaj de programare se datorează următoarelor caracteristici :

- Familiaritatea : sintaxa limbajului este foarte ușoară combinând sintaxele unora din cele mai populare limbaje Perl sau C;
- Simplitatea : sintaxa limbajului este destul de liberă. Nu este nevoie de includere de biblioteci sau de directive de compilare, codul PHP inclus într-un document executându-se între marcasele speciale;
- Eficiența : PHP-ul se folosește de mecanisme de alocare a resurselor, foarte necesare unui mediu multiutilizator, așa cum este web-ul;
- Securitate : PHP-ul pune la dispoziția programatorului un set flexibil și eficient de măsuri de siguranță;
- Flexibilitate : fiind apărut din necesitatea dezvoltării web-ului, PHP a fost modularizat pentru a ține pasul cu dezvoltarea diferitelor tehnologii. Nefiind legat de un anumit server web, PHP-ul a fost integrat pentru numeroasele servere web existente: Apache, IIS, Zeus, etc.;

- **Gratuitate** : este probabil cea mai importantă caracteristică a PHP-ului. Dezvoltarea PHP-ului sub licența open-source a determinat adaptarea rapidă a PHP-ului la nevoile web-ului, eficientizarea și securizarea codului.

2.2 Scurtă prezentarea a serverului de baze de date MySQL

MySQL este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeză MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD (Sistem de Gestiune a Bazelor de Date) open-source la ora actuală, fiind o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).

Deși este folosit foarte des împreună cu limbajul de programare PHP, cu MySQL se pot construi aplicații în orice limbaj major. Există multe scheme API disponibile pentru MySQL ce permit scrierea aplicațiilor în numeroase limbaje de programare pentru accesarea bazelor de date MySQL, cum ar fi: C, C++, C#, Java, Perl, PHP, Python, FreeBasic, etc., fiecare dintre acestea folosind un API specific.

Licența GNU GPL nu permite încorporarea MySQL în softuri comerciale; cei care doresc să facă acest lucru pot achiziționa, contra cost, o licență comercială de la compania producătoare, MySQL AB.

MySQL este componentă integrată a platformelor LAMP sau WAMP (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicație web este strâns legată de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. În multe cărți de specialitate este precizat faptul că MySQL este mult mai ușor de învățat și folosit decât multe din aplicațiile de gestiune a bazelor de date, ca exemplu comanda de ieșire fiind una simplă și evidentă: „exit” sau „quit”.

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau, prin descărcare de pe internet, o interfață grafică: MySQL Administrator și MySQL Query Browser. Un alt instrument de management al acestor baze de date este aplicația gratuită, scrisă în PHP, phpMyAdmin.

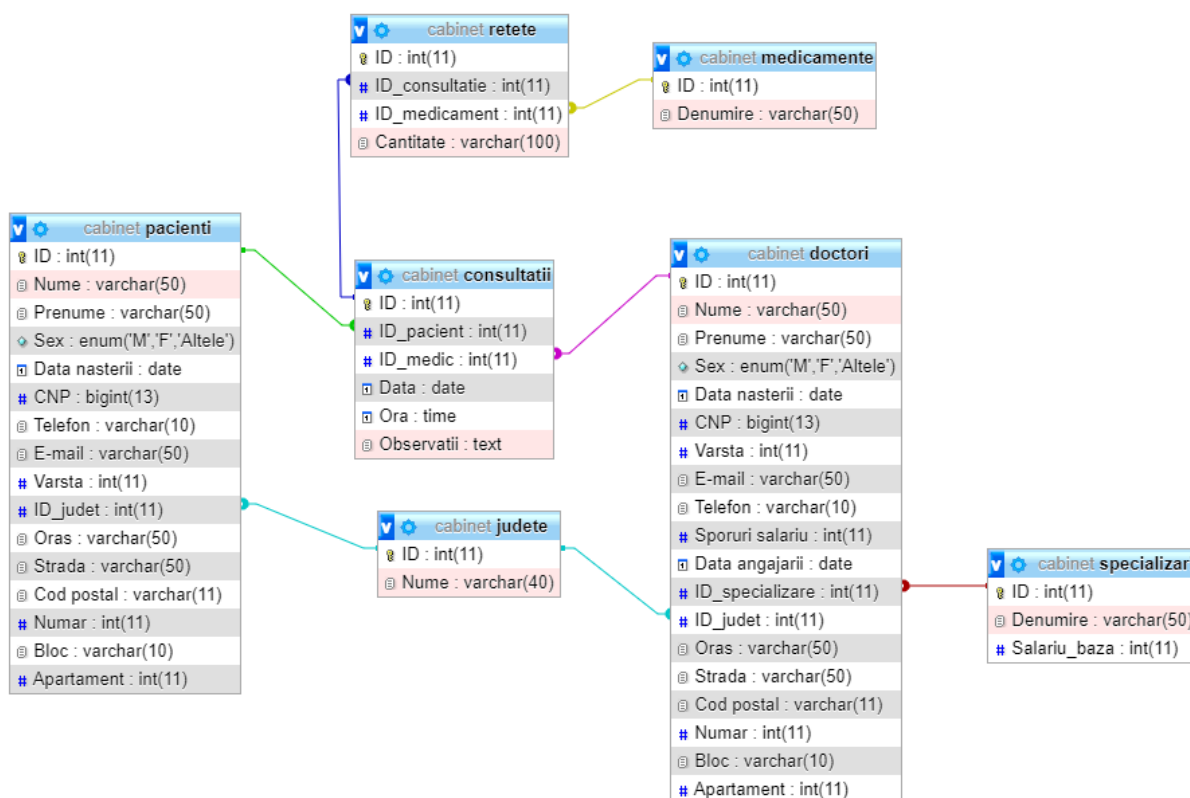
MySQL poate fi rulat pe multe dintre platformele software existente: AIX, FreeBSD, GNU/Linux, Mac OS X, NetBSD, Solaris, SunOS, Windows 9x/NT/2000/XP/Vista.

2.3 Resurse hardware necesare rulării proiectului

- **Procesor (CPU):** Intel Core i7-4510U @ 2.00 GHz 2.60 GHz
- **Memorie RAM:** 8GB
- **Placă video (GPU):** NVIDIA GeForce 840M
- **Stocare:** 1TB HDD

3. Realizarea aplicației

3.1 Structura bazei de date



Baza de date conține mai multe tabele cu date despre pacienți, doctori, medicamente precum și tabele suplimentare care rezolvă relațiile dintre acestea. Relațiile dintre tabele sunt vizibile pe schemă.


#	Nume	Tip	Colaționare	Atribute	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	Nume	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
3	Prenume	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
4	Sex	enum('M', 'F', 'Altele')	utf8mb4_general_ci		Nu	Altele		
5	Data nasterii	date			Nu	Niciuna		
6	CNP	bigint(13)			Nu	Niciuna		
7	Telefon	varchar(10)	utf8mb4_general_ci		Nu	Niciuna		
8	E-mail	varchar(50)	utf8mb4_general_ci		Da	NULL		
9	Varsta	int(11)			Nu	Niciuna		
10	ID_judet 	int(11)			Nu	Niciuna		
11	Oras	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
12	Strada	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
13	Cod postal	varchar(11)	utf8mb4_general_ci		Nu	Niciuna		
14	Numar	varchar(11)	utf8mb4_general_ci		Da	NULL		
15	Bloc	varchar(10)	utf8mb4_general_ci		Da	NULL		
16	Apartament	varchar(11)	utf8mb4_general_ci		Da	NULL		

Tabela **pacienți** stochează date de identificare despre clienții cabinetului medical. Acestea includ detalii precum **nume**, **prenume**, **sex**, **data nașterii**, **CNP**, **Telefon**, **E-mail** (câmp opțional). Câmpul vârstă este completat automat la introducerea unui pacient prin intermediul interfeței. Datele de identificare sunt urmate de adresa pacientului; **ID_judet** este o cheie străină către tabela **judete**; **oras**, **stradă** și **cod poștal** sunt câmpuri text obligatorii, iar câmpurile **număr**, **bloc** și **apartament** sunt câmpuri text opționale, având în vedere că o persoană poate locui la casa „12 Bis” (caz în care numărul nu este un set de cifre, iar blocul și apartamentul nu există).


#	Nume	Tip	Colaționare	Atribute	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	Nume	varchar(40)	utf8mb4_general_ci		Nu	Niciuna		

Tabela **judete** este un nomenclator care stochează denumirile celor 41 de județe din România, respectiv a municipiului București. Tabelele **pacienți** și **doctori** fac referință la această tabelă prin intermediul cheilor primare cu numele **ID_judet**.




#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	Nume	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
3	Prenume	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
4	Sex	enum('M', 'F', 'Altele')	utf8mb4_general_ci		Nu	Altele		
5	Data nasterii	date			Nu	Niciuna		
6	CNP	bigint(13)			Nu	Niciuna		
7	Varsta	int(11)			Nu	Niciuna		
8	E-mail	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
9	Telefon	varchar(10)	utf8mb4_general_ci		Nu	Niciuna		
10	ID_specializare 	int(11)			Nu	Niciuna		
11	Sporuri salariu	int(11)			Nu	0		
12	Data angajarii	date			Nu	Niciuna		
13	ID_judet 	int(11)			Nu	Niciuna		
14	Oras	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
15	Strada	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
16	Cod postal	varchar(11)	utf8mb4_general_ci		Nu	Niciuna		
17	Numar	varchar(11)	utf8mb4_general_ci		Da	NULL		
18	Bloc	varchar(10)	utf8mb4_general_ci		Da	NULL		
19	Apartament	varchar(11)	utf8mb4_general_ci		Da	NULL		

Tabela **doctori** este asemănătoare cu tabela **pacienți**, reținând date de identificare și adresa doctorilor. În plus față de tabela **pacienți**, câmpul **E-mail** este obligatoriu. De asemenea, această tabelă reține date despre specializarea doctorului în câmpul **ID_specializare** care este o cheie străină către tabela **specializări**, **sporurile pentru salariu** și **data angajării**.


#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	Denumire	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		
3	Salariu_baza	int(11)			Nu	Niciuna		

Tabela **specializări** este un nomenclator care reține date despre specializările medicilor, anume **denumirea** specializării și **salariul de bază**.

#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	Denumire	varchar(50)	utf8mb4_general_ci		Nu	Niciuna		

Tabela **medicamente** conține câmpul **Denumire** care reține denumirea medicamentului.




#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	ID_pacient 	int(11)			Nu	Niciuna		
3	ID_medec 	int(11)			Nu	Niciuna		
4	Data	date			Nu	Niciuna		
5	Ora	time			Nu	Niciuna		
6	Observatii	text	utf8mb4_general_ci		Da	NULL		

Tabela **consultații** rezolvă relația „many-to-many” dintre tabelele **pacienți** și **doctori**. În plus, aceasta reține **data** și **ora** examinării, precum și eventuale **observații** ale doctorului.




#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar
1	ID 	int(11)			Nu	Niciuna		AUTO_INCREMENT
2	ID_consultatie 	int(11)			Nu	Niciuna		
3	ID_medicament 	int(11)			Nu	Niciuna		
4	Cantitate	varchar(100)	utf8mb4_general_ci		Nu	Niciuna		

Tabela **rețete** rezolvă relația „many-to-many” dintre tabelele **consultații** și **medicamente**. În urma unei consultații, medicul prescrie pacientului o rețetă care conține mai multe medicamente și cantitatea care trebuie administrată periodic. Astfel, câmpul **ID_consultație** este o cheie străină către tabela **Consultații** și prin intermediul acesteia se pot afla pacientul și doctorul care au participat la consultație. Câmpul **ID_medicament** este o cheie străină către tabela **medicamente**, iar câmpul **cantitate** conține detalii despre administrarea medicamentului.

3.2 Descrierea interfeței



Meniul principal al programului conține câte un meniu pentru fiecare tabelă din baza de date, cu excepția tabelii **rețete**, care este alipită tabelii **consultații** în cadrul interfeței. Fiecare meniu (și paginile corespunzătoare) are alocat câte o culoare pentru a ajuta utilizatorul să recunoască ușor tabela cu care lucrează. În plus, interfața se adaptează dimensiunii ecranului pe care este rulată, așa cum se poate observa mai jos. Cele 2 butoane din fiecare meniu permit vizualizarea datelor din tabelă, respectiv adăugarea unei intrări noi.



Cele 5 interfețe sunt asemănătoare, astfel că voi prezenta tabelele **consultații** și **rețete**. Pentru a accesa acest meniu trebui doar să apăsăm unul dintre cele două butoane galbene, spre exemplu "Vezi consultații" va afișa următoarea pagină:

Tabela consultații									
<div> Acasă + Adaugă consultație </div> <div> Filtrează: Toti doctorii Toti pacienții <div> Y Filtrează Șterge filtre </div> </div>									
ID	Specializare	Doctor	Pacient	Data și ora (zi,luna,an, ora:min)	Observații	Acțiuni consultație	Medicament	Cantitate	Acțiuni rețete
1	Medic neurolog	Coca Melania	Popescu Cecilia	04.04.2021, 21:30	Pacientul este sanatos		Bioforte 3000	3 pastile/saptamana	
							Vitamina C 3000mg	1/2 pastile/zi	
							Paracetamol	3 pastile/zi timp de 1 saptamana	
							+ Adaugă rețetă		
2	Medic stomatolog	Fossil Mirel	Ioana Prioteasa	17.02.2021, 21:34	Pacientul necesita verificare		Faringosept	1 pastila/zi	
							Vitamina C 3000mg	2 pastile/saptamana	
							+ Adaugă rețetă		
4	Medic stomatolog	Fossil Mirel	Ioana Prioteasa	26.02.2021, 04:15	Pacientul este pe moarte; vesnica pomenire		Supramax articulații	3 pastile/zi	
							Bioforte 3000	3 pastile/zi	
							+ Adaugă rețetă		

În susul paginii sunt două butoane care permit revenirea la meniul principal (butonul "Acasă"), respectiv adăugarea unei consultații noi în baza de date. Sub ele se află un mic meniu care permite filtrarea datelor afișate în funcție de medic și de pacient. Tabelul din centrul paginii afișează datele din tabelele **consultații** și **rețete**, evidențiind relațiile dintre ele. Cele două butoane din coloana "Acțiuni consultație" permit editarea și stergerea datelor unei consultații, iar în mod similar butoanele din coloana "Acțiuni rețete" permit prelucrarea datelor unei rețete. Butonul "Adaugă rețetă" din interiorul tabelului permite adăugarea unei rețete în relație cu consultația de pe acel rând. În josul paginii apar două butoane identice butoanelor din susul paginii.

Apăsând butonul "Adaugă consultație" (din meniul principal sau din pagina "Vezi consultații") va deschide următorul formular:

Adaugă consultație

Detalii consultație


Doctor*

1 - Tache Daniela - Medic oftalmolog ▾


Pacient*

9 - Popescu Cecilia ▾

Data*


mm/dd/yyyy 


Ora*


--:-- -- 

Observații

Câmpurile marcate cu * sunt obligatorii.

 Acasă

 Vezi consultații

 Trimite

Câmpurile "Doctor" și "Pacient" vor deschide un dropdown care afișează doctorii, respectiv pacienții din baza de date și permit selectarea unuia. Prestabilit este selectată intrarea cu ID-ul minim.

Doctor*

2 - Fossil Mirel - Medic stomatolog ▾

Pacient*

11 - Dan Silviu ▾

Data*

11 - Dan Silviu

Ora*

16 - Ioana Prioteasa

Observații

1 - Tache Daniela - Medic oftalmolog

2 - Fossil Mirel - Medic stomatolog

4 - Radio Cornelius - Medic cardiolog

5 - Coca Melania - Medic neurolog

9 - Popescu Cecilia

16 - Ioana Prioteasa

19 - Lupu Cerasela

37 - Esperantia Piscopesco

Câmpurile "Data" și "Ora" deschid un meniu specializat care permite selectarea datelor corespunzătoare. Structura acestui meniu variază în funcție de sistemul de operare și de browser-ul folosit.

The left screenshot shows the 'Data*' field with a calendar for April 2021. The date 27 is selected. The right screenshot shows the 'Ora*' field with a time picker set to 06:46 PM.

Câmpul "Observații" este o casetă de text opțională care poate fi redimensionată. În final, cele 3 butoane de sub formular permit revenirea la meniul principal, vizualizarea datelor din tabela prezentată precedent sau trimiterea datelor pentru a fi validate și adăugate în baza de date. În funcție de validitatea datelor introduse, pagina va afișa diverse mesaje de eroare. Spre exemplu, mai jos este afișat mesajul de eroare pentru necompletarea unui câmp obligatoriu:

Detalii consultație

Doctor* 1 - Tache Daniela - Medic oftalmolog

Pacient* 9 - Popescu Cecilia

Data* 04/03/2021

Ora* --:-- --

Observații

Acest câmp trebuie completat!

Câmpurile marcate cu * sunt obligatorii.

Acasă Vezi consultații Trimite

Dacă utilizatorul modifică structura formularului (folosind, spre exemplu, "Inspect element"), pagina va afișa un mesaj de eroare:



Dacă datele sunt introduse corect și adăugarea lor în baza de date se efectuează cu succes, se afișează un mesaj de succes, care permite navigarea către meniu, pagina care afișează datele din tabelă sau adăugarea unei rețete la această consultație.



Dacă apăsăm butonul "Adaugă rețetă" de pe această pagină vom fi redirecționați către alt formular care permite adăugarea unei rețete consultației pe care tocmai am inserat-o în baza de date; câmpul consultație este completat automat. Urmează mai multe ferestre care permit adăugarea simultană a mai multor medicamente la aceeași rețetă. Checkbox-ul "Trimite rețetă" trebuie bifat pentru a trimite medicamentul respectiv către baza de date, altfel intrarea respectivă va fi ignorată.

Adaugă rețetă

Consultație

Consultație* 23 - Medic stomatolog Fossil Mirel - Lupu Cerasela - 17.07.2020, 22:42

Conținut rețetă 1

Medicament* 1 - Paracetamol

Cantitate*

Trimite rețetă ☒

Conținut rețetă 2

Medicament* 1 - Paracetamol

Cantitate*

Trimite rețetă ☐

Conținut rețetă 3

Medicament* 1 - Paracetamol

Cantitate*

Trimite rețetă ☐

Conținut rețetă 4

Medicament* 1 - Paracetamol

Cantitate*

Trimite rețetă ☐

Câmpurile marcate cu * sunt obligatorii, dacă ați bifat **Trimite rețetă**.

[Acasă](#) [Vezi consultații](#) [✓ Trimite](#)

Completarea inadecvată a câmpurilor va genera un mesaj de eroare la trimiterea formularului:

Conținut rețetă 1

Medicament* 3 - Vitamina C 3000mg

Cantitate*

Acest câmp trebuie completat!

Trimite rețetă ☒

La trimiterea datelor din formular sunt afișate diverse mesaje, care comunică utilizatorului succesul sau refuzul adăugării datelor.

Adaugă consultație

✗ Atenție! Rețeta 1 nu a putut fi adăugată!

⚠ Atenție! Acest medicament este deja prescris pacientului!

⚠ Rețeta 3 a fost ignorată.

✓ Rețeta 4 a fost adăugată cu succes.

🏠 Acasă

👁 Vezi consultații

Dacă revenim la pagina care afișează datele din tabelă și selectăm butonul pentru modificarea datelor unei consultații se deschide o pagină care arată astfel:

Modifica consultație

Detalii consultație

Doctor* 5 - Coca Melania - Medic neurolog ▼

Pacient* 9 - Popescu Cecilia ▼

Data* 04/04/2021 📅

Ora* 09:30 PM ⌚

Observații Pacientul este sanatos

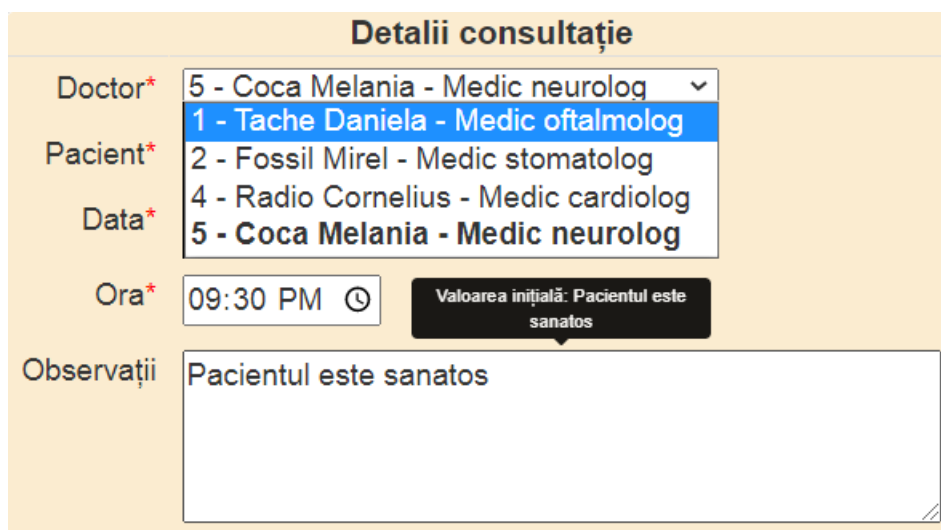
Câmpurile marcate cu * sunt obligatorii.

🏠 Acasă

👁 Vezi consultații

✓ Trimite

Datele introduse inițial în formular sunt evidențiate cu bold pentru casetele de tip dropdown respectiv cu un tooltip pentru celelalte casete:



Detalii consultație

Doctor* 5 - Coca Melania - Medic neurolog
1 - Tache Daniela - Medic oftalmolog

Pacient* 2 - Fossil Mirel - Medic stomatolog

Data* 4 - Radio Cornelius - Medic cardiolog
5 - **Coca Melania - Medic neurolog**

Ora* 09:30 PM Valoarea inițială: Pacientul este sanatos

Observații Pacientul este sanatos

La selectarea butonului care șterge o intrare din baza de date apare o pagină care permite utilizatorului să revoce acțiunea:



Șterge consultație

Sigur vrei să ștergi intrarea
1 - Medic neurolog Coca Melania - Popescu Cecilia - 04.04.2021, 21:30
din tabela consultații?

Vezi consultații Șterge

La apăsarea butonului "Șterge" apare un mesaj care spune utilizatorului dacă ștergerea s-a făcut cu succes. În general, mesajul de eroare apare dacă intrarea este pusă în relație cu alte tabele.



Șterge consultație **Șterge consultație**

✓ Consultația a fost ștearsă cu succes.

Acasă Vezi consultații

✗ Atenție! Ceva nu a mers!

Acasă Vezi consultații

3.3 Descrierea modulelor de cod utilizate

Toate paginile au un head comun care permite utilizarea de scripturi Java, librăria grafică Bootstrap și formaterile din fișierul "Interfata.css". Spre exemplu, head-ul fișierului acasă este:

```
<head>
  <title>Acasă</title>
  <link rel="icon" href="https://cdn3.iconfinder.com/data/icons/health/100/hospital-512.png">
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <link rel="stylesheet" href="Interfata.css">
  <style>
    * {
      text-align: center;
    }
  </style>
</head>
```

Unele pagini pot conține și formateri ale interfeței suplimentare în tag-ul style, meniul fiind un exemplu. Titlul paginii este încadrat într-un "well", un element de pagină care ocupă partea de sus:

```
<div class="well" id="title">
  Bine ați venit!<br/>Aici puteți lucra cu baza de date pentru cabinetul medical.
</div>
```

Meniurile pentru cele 5 tabele sunt încadrate într-o grilă Bootstrap, care oferă adaptabilitate paginii pe diverse rezoluții:

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-1"></div>
    <div class="col-md-2">
      <div class="panel panel-cyan">
        <div class="panel-heading cyan-heading">Pacienți</div>
        <div class="panel-body">
          <div class="btn-group-vertical">
            <a href="VeziPacienti.php" class="btn btn-cyan"><span class="glyphicon glyphicon-eye-open"></span> Vezi pacienți</a>
            <a href="AduagaPacient.php" class="btn btn-cyan"><span class="glyphicon glyphicon-plus"></span> Aduagă pacient</a>
          </div>
        </div>
        <div class="panel-footer">
          Pentru a lucra cu tabela pacienți, apăsați mai sus.
        </div>
      </div>
    </div>
  </div>
```

Fișierul "Interfata.css" conține cod CSS care formatează stilul. Spre exemplu, bucata de cod de mai jos setează imaginea de fundal:

```
body { /* Seteaza imaginea de fundal */
  background-image: url('Assets/Fundal2.jpg');
  background-repeat: repeat;
}
```

```

/* Culoarea YELLOW pentru CONSULTATII si RETETE */

.thead-yellow {
    background-color: #f4ab25;
    text-align: center;
    font-weight: bold;
}

#title-yellow {
    font-size: 3em;
    text-align: center;
    color: #cc9600;
}

.panel-yellow {
    border: 1px solid #cc9600;
}

.yellow-heading {
    background-color: #fbecd0;
    color: #cc9600;
}

.bg-yellow {
    background-color: #fbecd0;
}

.btn-yellow {
    background-color: #e69500;
    color: white;
}

```

Urmează câte un set de formatări pentru fiecare tabelă, în funcție de culoare. Clasele definite în exemplul din stânga setează diverse nuanțe de galben pentru elementele din paginile tabelor **consultații** și **rețete**. Spre exemplu, "thead-yellow" setează culoarea de fundal și fontul pentru primul rând din tabele, etc.

```

<?php
    $host = "localhost";
    $username = "root";
    $password = "";
    $database = "cabinet";
?>

```

În fișierul "constante.php" se găsesc datele de conectare la baza de date, pentru a putea fi ușor modificate de către utilizator:.

Afișarea datelor din tabele necesită parcurgerea mai multor etape. Mai întâi se încearcă conectarea la baza de date:

```

require 'constante.php';

$connection = mysqli_connect($host, $username, $password) or
    die ("Nu am putut sa ma conectez la serverul mysql");

mysqli_select_db($connection, $database) or die (mysqli_error($connection));

```

În cazul unui refuz apare un mesaj de eroare.

Ulterior se verifică dacă tabela de afișat conține date (exemplul de mai jos este pentru tabela **pacienți**). În cazul în care tabela este goală se afișează un mesaj de eroare:

```
$sql = "SELECT COUNT(id) AS cnt FROM pacienti";
$r = mysqli_query($connection, $sql);
$row = mysqli_fetch_array($r);

if ($row["cnt"] == 0)
{
    echo "<div class='alert alert-danger' style='text-align: center;'>
        <span class='glyphicon glyphicon-remove'></span> <strong>Atenție!</strong> Nu există date în tabelă.
    </div>";
    echo '<a href="Acasa.php" class="btn btn-cyan"><span class="glyphicon glyphicon-home"></span> Acasă</a>';
    exit;
}
```

Altfel, se creează o interogare SQL care extrage din baza de date câmpurile de afișat în tabel:

```
$sql = "SELECT p.ID, p.Nume, p.Prenume, p.Sex, p.`Data nasterii`, p.CNP, p.Telefon, p.`E-mail`, p.Varsta,
        j.Nume AS Judet, p.Oras, p.Strada, p.`Cod postal`, p.Numar, p.Bloc, p.Apartament FROM pacienti AS p
        LEFT JOIN judete AS j ON p.ID_judet=j.ID ORDER BY p.ID ASC";
$r = mysqli_query($connection, $sql);
```

În continuare este structurat capul tabelului în care se afișează datele:

```
<table class="table">
  <thead class="thead-cyan">
    <tr>
      <td class="id">ID</td>
      <td>Nume</td>
      <td>Prenume</td>
      <td>Sex</td>
      <td>Data nașterii<br/>(zi.luna.an)</td>
      <td>CNP</td>
      <td>Telefon</td>
      <td>E-mail</td>
      <td>Vârstă</td>
      <td>Judet</td>
      <td>Oras</td>
      <td>Adresă</td>
      <td>Cod poștal</td>
      <td class="actions">Acțiuni</td>
    </tr>
  </thead>
```

Pentru a afișa datele propriu-zise este rulat un script PHP care formatează și scrie pe ecran pacenții, rând cu rând:

```
<tbody>
<?php
    $i=1;
    while ($row = mysqli_fetch_array($r))
    {
        if ($i%2==0)
            echo "<tr class='bg-cyan'>";
        else
            echo "<tr bgcolor='#ffffff'>";
        $i++;
        echo "<td>" . $row["ID"] . "</td>";
        echo "<td>" . $row["Nume"] . "</td>";
        echo "<td>" . $row["Prenume"] . "</td>";
        echo "<td>" . $row["Sex"] . "</td>";
        $data=date_create($row["Data nasterii"]);
        echo "<td>" . date_format($data,"d.m.Y") . "</td>";
        echo "<td>" . $row["CNP"] . "</td>";
        echo "<td>" . $row["Telefon"] . "</td>";
        echo "<td>" . $row["E-mail"] . "</td>";
        echo "<td>" . $row["Varsta"] . "</td>";
        echo "<td>" . $row["Judet"] . "</td>";
        echo "<td>" . $row["Oras"] . "</td>";
        $adresa=$row["Strada"];
        if ($row["Numar"])
            $adresa.=" Nr " . $row["Numar"];
        if ($row["Bloc"])
            $adresa.=" Bloc " . $row["Bloc"];
        if ($row["Apartament"])
            $adresa.=" Ap " . $row["Apartament"];
        echo "<td>" . $adresa . "</td>";
        echo "<td>" . $row["Cod postal"] . "</td>";
        echo "<td><a class='glyphicon glyphicon-edit glyph-cyan' href = 'ModificaPacient.php?ID=" . $row["ID"] . "'></a>";
        echo "<td><a class='glyphicon glyphicon-trash glyph-cyan' href = 'StergePacient.php?ID=" . $row["ID"] . "'></a></td>";
        echo "</tr>";
    }
?>
</tbody>
```

Câmpurile din formularul de adăugare sunt încadrate într-un tabel, iar pentru fiecare se rulează câte un script PHP care permite păstrarea datelor introduse dacă formularul este trimis completat necorespunzător:

```
<?php
    $in = "<input type = 'text' name = 'Nume'";
    if (isset($_SESSION["valoareNume"]))
        $in .= " value = '" . $_SESSION["valoareNume"] . "'";
    $in .= ">";
    echo $in;
    if (isset($_SESSION["errorNume"]))
    {
        echo "</td><td class='error'>";
        switch($_SESSION["errorNume"])
        {
            case 1: echo "Hackere!"; break;
            case 2: echo "Acest câmp trebuie completat!"; break;
            case 3: echo "Introdu un număr valid!"; break;
            case 4: echo "Selectează un câmp valid!"; break;
            case 5: echo "Acest câmp permite numai litere!"; break;
        }
    }
?>
```

Validarea datelor și umplerea vectorului sesiune este făcută de funcția de mai jos, cu doi parametri: primul este denumirea variabilei de validat, iar al doilea este tipul variabilei.

```
function validateInput($key, $type = 0)
{
    $errorKeyName = "error" . ucfirst($key);
    $valueKeyName = "valoare" . ucfirst($key);

    unset($_SESSION[$errorKeyName]);
    $_SESSION[$valueKeyName] = $_POST[$key];

    if (!isset($_POST[$key]))
    {
        $_SESSION[$errorKeyName] = 1;
        return 0;
    }

    if (empty(trim($_POST[$key])) && $type!=-1)
    {
        $_SESSION[$errorKeyName] = 2;
        return 0;
    }

    if ($type == 1)
    {
        if (!is_numeric($_POST[$key]))
        {
            $_SESSION[$errorKeyName] = 3;
            return 0;
        }
    }

    if ($type == 2)
    {
        if ($_POST[$key]!="M" && $_POST[$key]!="F"
            && $_POST[$key]!="Altele")
        {
            $_SESSION[$errorKeyName] = 4;
            return 0;
        }
    }

    if ($type == 3)
    {
        if (!ctype_alpha($_POST[$key]))
        {
            $_SESSION[$errorKeyName] = 5;
            return 0;
        }
    }

    return 1;
}
```

Fiecare câmp este verificat și userul este redirecționat înapoi la formular dacă apar erori:

```
$ok = 1;
$ok = $ok & validateInput("Nume",3);
$ok = $ok & validateInput("Prenume",3);
$ok = $ok & validateInput("Sex",2);
$ok = $ok & validateInput("Data_nasterii");
$ok = $ok & validateInput("CNP",1);
$ok = $ok & validateInput("Telefon",1);
$ok = $ok & validateInput("E_mail",-1);
$ok = $ok & validateInput("ID_judet",1);
$ok = $ok & validateInput("Oras");
$ok = $ok & validateInput("Strada");
$ok = $ok & validateInput("Cod_postal",1);
$ok = $ok & validateInput("Numar",-1);
$ok = $ok & validateInput("Bloc",-1);
$ok = $ok & validateInput("Apartament",-1);

if ($ok == 0) //Daca am erori, redirectez browserul catre formular
{
    header("Location: AdaugaPacient.php");
    exit;
}
```


Apoi datele introduse din formular sunt prelucrate pentru a proteja sistemul de eventuali hackeri:

```
$ok = 1;
$ok = $ok & validateInput("Nume",3);|
$ok = $ok & validateInput("Prenume",3);
$ok = $ok & validateInput("Sex",2);
$ok = $ok & validateInput("Data_nasterii");
$ok = $ok & validateInput("CNP",1);
$ok = $ok & validateInput("Telefon",1);
$ok = $ok & validateInput("E_mail",-1);
$ok = $ok & validateInput("ID_judet",1);
$ok = $ok & validateInput("Oras");
$ok = $ok & validateInput("Strada");
$ok = $ok & validateInput("Cod_postal",1);
$ok = $ok & validateInput("Numar",-1);
$ok = $ok & validateInput("Bloc",-1);
$ok = $ok & validateInput("Apartament",-1);

if ($ok == 0)          //Daca am erori, redirectez browserul catre formular
{
    header("Location: AdaugaPacient.php");
    exit;
}
```

Vârsta este generată automat folosind următoarea funcție:

```
function varsta($Data_nasterii)
{
    $fo = new DateTimeZone('Europe/Bucharest');
    $dn = date_create_from_format('Y-m-d', $Data_nasterii, $fo);
    $azi=new DateTime('now');
    $varsta=date_diff($dn, $azi);
    return $varsta->format("%y");
}
```

La final, dacă toate datele sunt validate cu succes, se generează interogarea care adaugă datele:

```
$sql = "INSERT INTO `pacienti` (`Nume`, `Prenume`, `Sex`, `Data nasterii`, `CNP`, `Telefon`,
`E-mail`, `Varsta`, `ID_judet`, `Oras`, `Strada`, `Cod postal`, `Numar`, `Bloc`, `Apartament`)
VALUES ('".$Nume."', ' ".$Prenume."', ' ".$Sex."', ' ".$Data_nasterii."', ' ".$CNP."', ' ".$Telefon."', ' ";
if ($E_mail)
    $sql.=" $E_mail;
$sql.=" ' ".$Varsta."', ' ".$ID_judet."', ' ".$Oras."', ' ".$Strada."', ' ".$Cod_postal."', ' ";
if ($Numar)
    $sql.=" $Numar;
$sql.=" ' ";
if ($Bloc)
    $sql.=" $Bloc;
$sql.=" ' ";
if ($Apartament)
    $sql.=" $Apartament;
$sql.=" ' )";
$r = mysqli_query($connection, $sql);
```

În funcție de rezultatul interogării, se afișează un mesaj de eroare sau de succes:

```
if ($r)
{
    echo "<div class='alert alert-success' style='text-align: center;'"
        "<span class='glyphicon glyphicon-ok'></span> Pacientul a fost adăugat cu succes."
        "</div>";
}
else
{
    echo "<div class='alert alert-danger' style='text-align: center;'"
        "<span class='glyphicon glyphicon-remove'></span> <strong>Atenție!</strong> Ceva nu a mers!"
        "</div>";
}
```

În mod asemănător sunt prelucrate datele și în formularele care permit modificarea sau ștergerea datelor din tabelă:

```
$sql = "UPDATE `pacienti` SET `Nume`='".$Nume."', `Prenume`='".$Prenume."', `Sex`='".$Sex."',
`Data nasterii`='".$Data_nasterii."', `CNP`='".$CNP."', `Telefon`='".$Telefon."',
`E-mail`='".$E_mail."', `Varsta`='".$Varsta."', `ID_judet`='".$ID_judet."', `Oras`='".$Oras."',
`Strada`='".$Strada."', `Cod postal`='".$Cod_postal."', `Numar`='".$Numar."', `Bloc`='".$Bloc."',
`Apartament`='".$Apartament.'" WHERE ID='".$ID.'";";
$r = mysqli_query($connection, $sql);
```

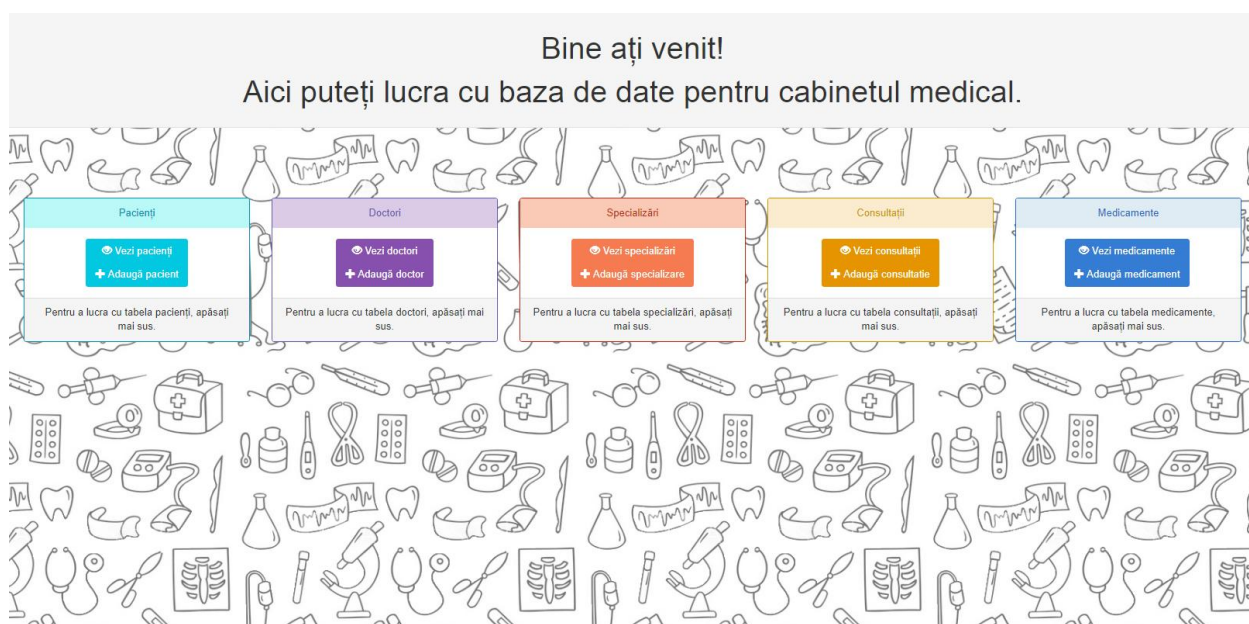
```
$sql = "DELETE FROM pacienti WHERE ID='".$$_GET["ID"].'";";
$r = mysqli_query($connection, $sql);
```

Afișarea câmpurilor dropdown din diverse formulare necesită interogarea bazei de date și apoi afișarea opțiunilor printr-un script PHP. Spre exemplu, pentru afișarea județelor se folosește următoarea secvență de cod:

```
$sql = "SELECT ID, Nume FROM judete ORDER BY ID ASC;";
$r = mysqli_query($connection, $sql);
```

```
<tr>
    <td class="legenda">Județ<star>*</star></td>
    <td>
        <select name="ID_judet">
            <?php
                while ($row = mysqli_fetch_array($r))
                {
                    echo "<option value='".$row["ID"]."'";
                    if (isset($_SESSION["valoareID_judet"]))
                        if ($_SESSION["valoareID_judet"]==$row["ID"])
                            echo ' selected';
                    echo ">".$row["ID"]." - ".$row["Nume"]."</option>";
                }
            ?>
        </select>
    </td>
</tr>
```

4. Utilizarea aplicației



Utilizatorul alege din meniul principal tabela cu care dorește să lucreze și apasă butonul corespunzător.

Indiferent de tabela aleasă, poate fi selectată fie pagina care afișează datele, fie pagina care conține formularul de adăugare a unei intrări noi. Spre exemplu, butonul "Vezi pacienți" va scrie pe ecran detaliile pacienților din baza de date sub formă de tabel:

De aici, utilizatorul poate adăuga un pacient cu butonul "Adaugă pacient" (care se găsește și în meniul principal). Browser-ul va fi redirecționat către pagina care conține formularul de adăugare:

Adaugă pacient

Date indentificare

Nume*

Prenume*

Sex* ☐ M ☐ F ☒ Altele

Data nașterii*

CNP*

Telefon*

E-mail

Adresa

Județ*

Oraș*

Stradă*

Cod poștal*

Număr

Bloc

Apartament

Câmpurile marcate cu * sunt obligatorii.

[Acasă](#)
[Vezi pacienți](#)
[Trimit](#)

Tipul de dată acceptat de fiecare câmp este semnalat intuitiv, prin tipul de casetă disponibil. Utilizatorul are opțiunea de a completa formularul și de a trimite datele pentru verificare sau de a reveni la meniul principal ori la pagina care afișează datele din tabelă. Dacă încearcă să trimită formularul incomplet, vor apărea diverse mesaje de eroare care îi comunică utilizatorului ce trebuie modificat pentru a continua:

Adaugă pacient

Date indentificare

Nume*

Prenume*

Sex* ☒ M ☐ F ☐ Altele

Data nașterii*

CNP* Acest câmp trebuie completat!

Telefon* Acest câmp trebuie completat!

E-mail

Adresa

Județ*

Oraș* Acest câmp trebuie completat!

Stradă* Acest câmp trebuie completat!

Cod poștal* Introdu un număr valid!

Număr

Bloc

Apartament

Câmpurile marcate cu * sunt obligatorii.

[Acasă](#)
[Vezi pacienți](#)
[Trimit](#)

Dacă toate datele sunt completate corespunzător, va apărea următorul mesaj de succes:

Adaugă pacient

✓ Pacientul a fost adăugat cu succes.

[Acasă](#)
[Vezi pacienți](#)

De pe această pagină, utilizatorul poate reveni la tabelul cu pacienți pentru a vedea intrarea nou adăugată:

54	Ion	Ion	M	09.04.2021	1234567891234	0712345678	ionion@yahoo.com	0	Arad	Ionesti	Str Ion, Nr 100, Bloc 101, Ap 123456	12		
----	-----	-----	---	------------	---------------	------------	------------------	---	------	---------	--------------------------------------	----	--	--

[Acasă](#) [Adaugă pacient](#)

Folosind primul buton din partea dreaptă, utilizatorul poate edita intrarea din baza de date. Interfața este asemănătoare cu cea din formularul de adăugare, însă este completată automat cu datele pacientului de modificat și valorile inițiale sunt evidențiate:

Date

Valoarea inițială: Ion

Nume*

Prenume*

Sex* ☒ M
☐ F
☐ Altele

Data nașterii*

CNP*

Telefon*

E-mail

Adresa

Județ*

Oraș*

Stradă*

Cod poștal*

Număr

Bloc

Apartament

Câmpurile marcate cu * sunt obligatorii.

[Acasă](#) [Vezi pacienți](#) [Trimite](#)

La trimiterea formularului, datele introduse sunt din nou verificate și utilizatorul va fi îndrumat de program pentru a completa corespunzător câmpurile.

Dacă utilizatorul apasă butonul în formă de coș de gunoi din tabelul care afișează pacienții, va fi redirecționat către o pagină care cere verificarea acțiunii:

Șterge pacient

Sigur vrei să ștergi intrarea
54 - Ion Ion
din tabela pacienți?

[Vezi pacienți](#) [Șterge](#)

Ștergerea poate fi revocată prin revenirea la pagina cu tabelul, sau confirmată apăsând butonul "Șterge", caz în care utilizatorul va fi redirecționat către o pagină care conține unul din următoarele mesaje:



De cele mai multe ori, mesajul de eroare apare din cauza unei relații între intrarea care se încearcă a fi ștearsă și o altă tabelă (spre exemplu, un pacient care este programat pentru o consultație nu poate fi ștears decât dacă mai întâi este ștearsă consultația respectivă).

În mod asemănător se poate lucra cu toate celelalte tabele, iar utilizatorul va recunoaște ușor entitățile cu care lucrează după culoarea interfeței.

5. Extinderi posibile ale aplicației

Aplicația poate fi dezvoltată pe viitor pe mai multe planuri, atât prin îmbunătățirea modului de funcționare, cât și prin ea unor elemente de interfață noi.

În ceea ce privește funcționarea aplicației, ea poate fi extinsă pentru a permite stocarea în baza de date a mai multor entități (spre exemplu, date despre angajații care nu sunt doctori, cum ar fi recepționiștii). O altă extindere poate fi restricționarea accesului la date în funcție de rolul utilizatorului (administrator, doctor, recepționist). În plus, aplicația poate fi modularizată și mai mult pentru a simplifica aspectul codului.

Îmbunătățiri se pot aduce și la nivelul interfeței. Se poate adăuga o opțiune care permite sortarea tabelelor în funcție de o coloană sau posibilitatea de a filtra datele din orice tabelă, în mod asemănător tabelului consultații. Mai mult, pot fi adăugate criterii noi pentru filtrare, cum ar fi data, vârsta, starea pacientului, etc.

6. Bibliografie

<https://ro.wikipedia.org/wiki/PHP>

<https://ro.wikipedia.org/wiki/MySQL>