

Proiect Baze de Date

Tema proiectului: Aplicatie pentru evidenta consultatiilor la o clinica medicala.

Proiectul contine 8 tabele, excluzand tabelele de legatura. Acesta are 4 relatii 1:1, 3 relatii 1:N si o relatie N:N.

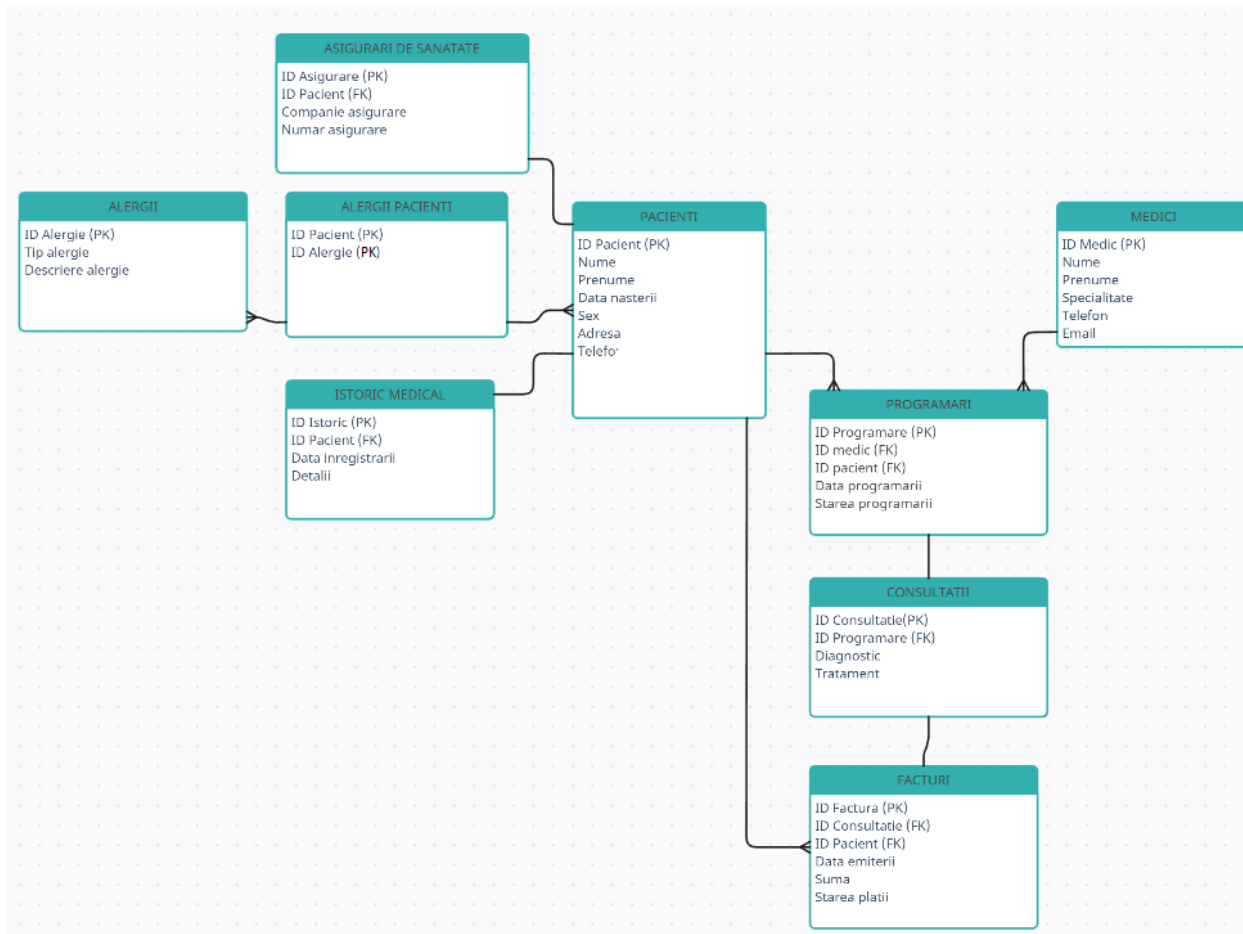
Tabelele sunt urmatoarele:

1. Asigurari de sanatate (stocheaza detalii despre asigurarile de sanatate ale pacientilor)
2. Alergii pacienti (face legatura intre tabela Alergii si tabela Pacienti, pentru a stoca detalii cu privire la alergiile pacientilor)
3. Alergii (stocheaza detalii cu privire la alergii)
4. Istoric medical (stocheaza detalii despre istoricul medical al pacientilor)
5. Pacienti (stocheaza detalii despre pacienti)
6. Medici (stocheaza detalii despre medici)
7. Programari (inregistreaza programarile facute pentru consultatii)
8. Consultatii (pastreaza detalii despre consultatiile medicale)
9. Facturi (stocheaza informatii cu privire la facturile consultatiilor)

Exista urmatoarele **relatii** intre tabele:

1. N:N intre Alergii – Pacienti
2. 1:N intre Pacienti – Programari
3. 1:N intre Medici – Programari
4. 1:N intre Pacienti – Facturi
5. 1:1 intre Pacienti – Istoric Medical
6. 1:1 intre Pacienti – Asigurari de sanatate
7. 1:1 intre Programari – Consultatii
8. 1:1 intre Consultatii – Facturi

Diagrama tabelor este urmatoarea:



Aplicatia poate functiona in 2 regimuri:

1. Pacient

Daca ne logam cu un mail si o parola in pagina de login vom fi redirectionati pe pagina de pacient. De acolo putem face mai multe lucruri, pe care le voi prezenta mai jos:

Logarea se face prin urmatoarea interogare:

```
"select * from Pacienti where Email='$email' and Password='$password'"
```

In momentul in care email-ul si parola sunt diferite de 'admin', suntem redirectionati pe pagina Profile.

Personal Info

FULL NAME
Barbu Andrei

BIRTHDAY
2002-07-26 UPDATE

GENDER
M UPDATE

ADDRESS
Str Christigii UPDATE

NUMAR ASIGURARE
1234567892 UPDATE

Aici observam mai multe campuri ce pot fi modificate. Daca modificam un camp si apasam pe butonul UPDATE, acesta va fi modificat in baza de date. Interogările folosite sunt urmatoarele:

```
"UPDATE pacienti SET `Data nasterii`=' $newAge' WHERE Email='$email'"
"UPDATE pacienti SET `Sex`=' $newGender' WHERE Email='$email'"
"UPDATE pacienti SET `Adresa`=' $newAddress' WHERE Email='$email'"
"UPDATE `asigurari de sanatate` a
    JOIN `pacienti` p ON a.`ID Pacient` = p.`ID Pacient`
    SET a.`Numar asigurare` = '$newAsig'
    WHERE p.`Email` = '$email';"
```

Interogarea pentru numarul de asigurare reprezinta prima interogare simpla din aplicatie.

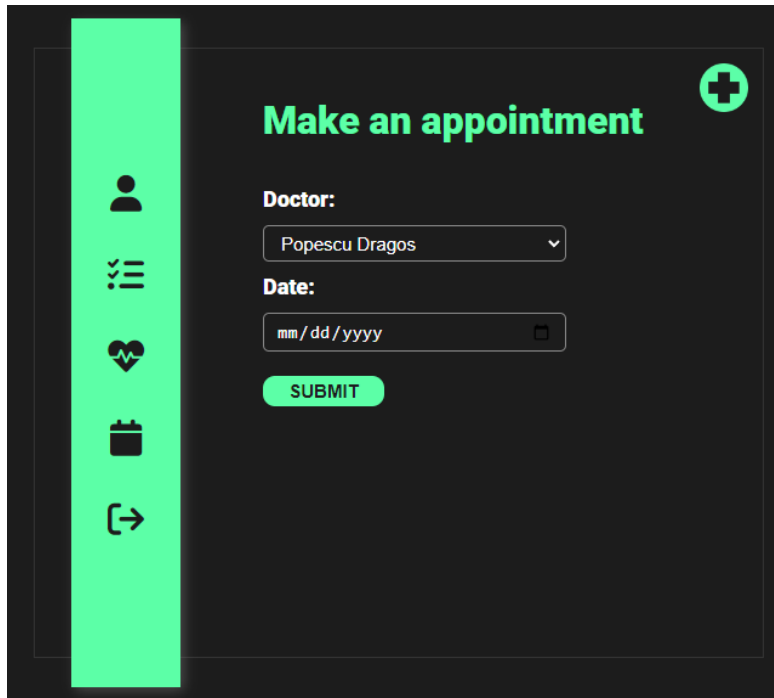
A doua interogare simpla este folosita pentru a salva intr-o variabila numarul de asigurare al pacientului:

```
"SELECT a.`Numar asigurare`
FROM `asigurari de sanatate` a
JOIN `pacienti` p ON a.`ID Pacient` = p.`ID Pacient`
WHERE p.`Email` = '$email'" )
```

Update-ul este facut pe doua tabele: Pacienti si Asigurari de sanatate.

Barbu Andrei-Catalin
331AA

Daca apasam pe urmatoarea iconita din meniu, vom fi redirectionati pe pagina de programari. In aceasta pagina putem selecta un doctor si o data pentru a face o programare. Dupa ce apasam SUBMIT, aceasta este introdusa in baza de date.



In aceasta pagina se face primul insert din aplicatie, in tabela programari:

```
"insert into programari (`ID Medic`, `ID Pacient`, `Data programarii`, `Starea programarii`) values ('$idMedic', '$idPacient', '$appointmentDate', 'In asteptare')"
```

Urmatoarea iconita ne redirectioneaza pe pagina de alergii. Aici putem vedea alergiile pe care le are pacientul, dar si putem adauga sau sterge alergii.

Pentru a afisa alergiile am folosit a treia interogare simpla a aplicatiei:

```
"
SELECT a.`ID Alergie`, a.`Tip Alergie`, a.`Descriere alergie`
FROM pacienti p
JOIN `alergii pacienti` ap ON p.`ID Pacient` = ap.`ID Pacient`
JOIN alergii a ON ap.`ID Alergie` = a.`ID Alergie`
WHERE p.`Email` = '$email';
"
```

Atributele Tip Alergie si Descriere Alergie urmeaza sa fie afisate in tabel:

The screenshot shows a web application interface for managing allergies. On the left is a dark sidebar with icons for a person, a list, a heart with a pulse line, a calendar, and a right-pointing arrow. The main content area has a light blue header with the title 'Allergies' and a green plus icon. Below the header is a table with two columns: 'Tip Alergie' and 'Descriere Alergie'. The table contains two rows: 'Alimentara' with 'Alergie la alune' and 'Medicamentoasa' with 'Alergie la Nurofen'. Below the table are two sections: 'Adauga alergie:' and 'Sterge alergie:'. Each section has a dropdown menu with the text '-- select an option --' and a corresponding button ('ADD' and 'DELETE' respectively).

Tip Alergie	Descriere Alergie
Alimentara	Alergie la alune
Medicamentoasa	Alergie la Nurofen

Adauga alergie:
-- select an option --
ADD

Sterge alergie:
-- select an option --
DELETE

Putem adauga o alergie din o lista cu alergii deja prestabilite de cabinet. Pentru asta, selectam o alergie din drop-down si apasam pe butonul ADD. Al doilea Insert a fost facut pe tabela Alergii in urmatorul mod:

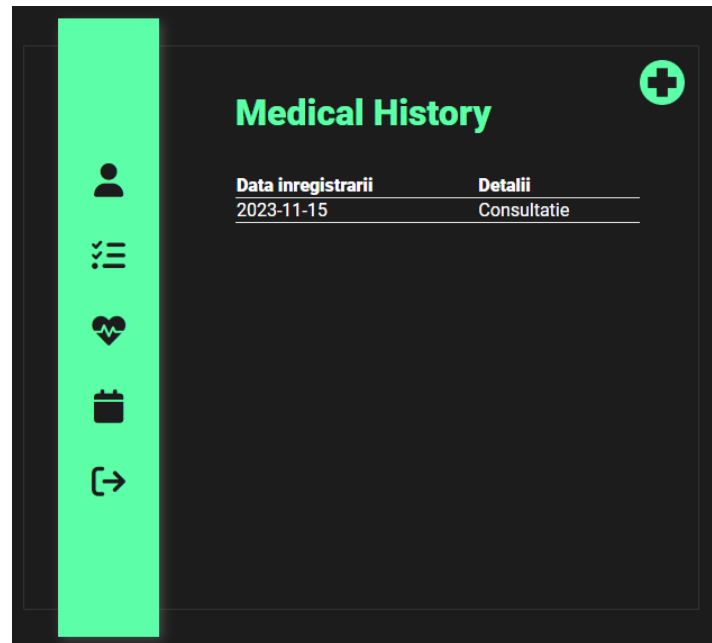
```
"insert into `alergii_pacienti` (`ID Alergie`, `ID Pacient`) values  
($idAlergie, $idPacient)"
```

In al doilea drop-down gasim alergiile pe care pacientul deja le are. Daca selectam una si apasam pe butonul DELETE, se va realiza primul Delete al aplicatiei, in urmatorul mod:

```
"delete from `alergii_pacienti` where `ID Pacient`=' $idPacient' and `ID  
Alergie`=' $idAlergie'"
```

Barbu Andrei-Catalin
331AA

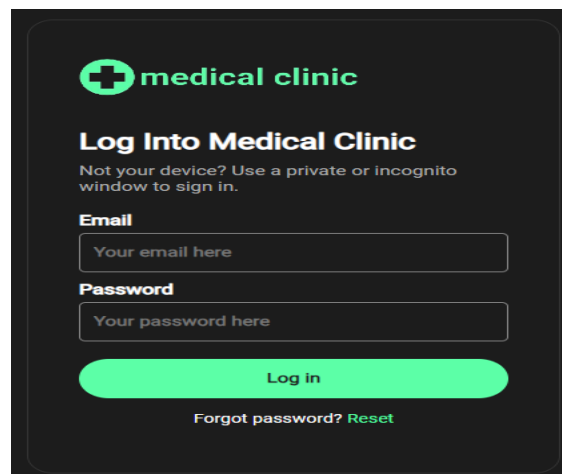
Urmatoarea iconita ne duce pe pagina istoricului medical. Aceasta afiseaza un tabel cu istoricul medical al pacientului la clinica.



Pentru a face asta, vom folosi a patra interogare simpla a aplicatiei:

```
"
SELECT im.`Data inregistrarii`, im.`Detalii`
FROM `istoric medical` im
JOIN `pacienti` p ON im.`ID Pacient` = p.`ID Pacient`
WHERE p.`Email` = '$email';
"
```

Ultima iconita ne va duce inapoi pe pagina de login, unde ne putem loga din nou ca si pacient sau putem scrie 'admin' in ambele campuri pentru pagina de administrator.



2. Administrator

Daca ne logam cu credentialele 'admin' si 'admin' vom fi redirectionati pe pagina dashboard:



Nume și Prenume Pacient	Nume și Prenume Medic	Data programării	Starea programării
Marin Andreea	Rapeanu Tudor	2024-12-21	Confirmat
Popescu Andrei	Nancu Iustin	2024-02-01	Confirmat
Marin Andreea	Barbu Codrin	2024-01-27	In asteptare
Marin Andreea	Popescu Dragos	2024-01-20	In asteptare
Marin Andreea	Popescu Dragos	2024-01-12	In asteptare
Caragea Matei	Barbu Codrin	2024-01-05	Confirmat
Barbu Andrei	Popescu Dragos	2023-11-25	In asteptare
Caragea Matei	Gavrila Viorel	2023-11-21	In asteptare

Aici vedem o lista cu toate programarile clinicii. Pentru a reusi sa afisez toate aceste date, am folosit urmatoarea interogare simpla a aplicatiei:

```
"
SELECT pr.`ID Programare`,
       CONCAT(pa.`Nume`, ' ', pa.`Prenume`) AS 'Nume si prenume pacient',
       CONCAT(m.`Nume`, ' ', m.`Prenume`) AS 'Nume si prenume medic',
       pr.`Data programarii`,
       pr.`Starea Programarii`
FROM programari pr
JOIN pacienti pa ON pr.`ID Pacient` = pa.`ID Pacient`
JOIN medici m ON pr.`ID Medic` = m.`ID Medic`
ORDER BY pr.`Data programarii` DESC;
"
```

Urmatoarea iconita ne va duce pe pagina de facturi a clinicii. Aceasta contine un tabel cu toate facturile, cat si optiunea de a sterge toate facturile unui pacient:

Nume și Prenume Pacient	Data Emiterii	Suma	Starea plății
Caragea Matei	2024-01-01	2432	Confirmat
Popescu Andrei	2023-12-25	4000	Confirmat
Marin Andreea	2021-11-21	432	In asteptare

Sterge facturile pacientului:

-- select an option --

DELETE

Pentru a afisa detaliile facturilor, am folosit a sasea interogare simpla a aplicatiei:

```
SELECT CONCAT(p.`Nume`, ' ', p.`Prenume`) AS 'Nume si Prenume Pacient',  
        f.`Data emiterii`,  
        f.`Suma`,  
        f.`Starea platii`  
FROM pacienti p  
JOIN facturi f ON p.`ID Pacient` = f.`ID Pacient`  
ORDER BY f.`Data emiterii` DESC;
```

Pentru a efectua al doilea delete al aplicatiei, acesta va folosi o interogare complexa si va sterge datele din tabela facturi.

```
DELETE FROM facturi  
WHERE `ID Pacient` IN (  
    SELECT `ID Pacient` FROM pacienti WHERE Nume =  
'$numePacient' AND Prenume = '$prenumePacient'  
)
```


Urmatoarea iconita ne duce pe pagina pentru Detalii Pacienti. Aceasta foloseste un drop-down pentru a selecta un pacient. Dupa ce apasam pe butonul submit, aplicatia arata istoricul medical, facturile si asigurarea de sanatate a pacientului.

Detalii Pacienti

Selecteaza pacient:

-- select an option --

SELECT

Istoric Medical pentru Caragea Matei

Data înregistrării	Detalii
2023-11-09	Operatie

Facturi pentru Caragea Matei

Data emiterii	Suma	Starea platii
2024-01-01	2432	Confirmat

Asigurări de Sănătate pentru Caragea Matei

Companie Asigurare	Numar Asigurare
Regina Maria	3412342123

Toate aceste date sunt obtinute cu ajutorul urmatoarelor trei interogari complexe ale aplicatiei:

```
"
SELECT `Data inregistrarii`, `Detalii`
FROM `istoric medical`
WHERE `ID Pacient` = (SELECT `ID Pacient` FROM pacienti WHERE Nume =
'$numePacient' AND Prenume = '$prenumePacient')
"
```

```
"
SELECT `Data emiterii`, `Suma`, `Starea platii`
FROM `facturi`
WHERE `ID Pacient` = (SELECT `ID Pacient` FROM pacienti WHERE Nume =
'$numePacient' AND Prenume = '$prenumePacient')
"
```

```
"
SELECT `Companie Asigurare`, `Numar Asigurare`
FROM `asigurari de sanatate`
WHERE `ID Pacient` = (SELECT `ID Pacient` FROM pacienti WHERE Nume =
'$numePacient' AND Prenume = '$prenumePacient')
"
```

In total, aplicatia contine:

- UPDATE pentru 2 tabele
- INSERT pentru 2 tabele
- DELETE pentru 2 tabele
- 6 interogari simple
- 4 interogari complexe

Bibliografie:

1. <https://www.w3schools.com/sql/default.asp>
2. <https://curs.upb.ro/2023/course/view.php?id=8628#section-15>
3. <https://www.w3schools.com/php/default.asp>