Aplicatie pentru urmarirea istoricului medical

intr-un cabinet veterinar

1. Definirea cerintelor

Eu am avut de facut o aplicatie pentru a tine evidenta istoricului medical intr-un cabinet veterinar si am ales sa se numeasca “PetVet”. In etapa de proiectare a aplicatiei am identificat cateva componente pe care trebuie sa le contina aplicatia:

* Medicii ce lucreaza la acest cabinet veterinar;
* Medicamentele pe care le pune la dispozitie cabinetul veterinar;
* Pacientii ce viziteaza acest cabinet, impreuna cu proprietarii lor daca este cazul;
* Interventiile pe care le poate face un medic;
* Fisa medicala a unui pacient, ce tine evidenta unei vizite la cabinet si din ce cauza;

Pe baza acestor componente am identificat si niste functionalitati pe care trebuie sa le indeplineasca aplicatia:

* Aplicatia trebuie sa poata lasa un utilizator(medicul) sa se inregistreze in baza de date si dupa sa poata sa se logheze in aplicatie;
* Pe langa utilizatorii obisnuiti(userii) ar trebui sa existe un utilizator admin ce poate accesa paginile de administrare si insera, modifica sau sterge componente din baza de date;
* Odata logat, medicul poate accesa datele despre el, despre pacienti si despre componentele ce tin de medicamente si interventii;
* Medicul sa poata accesa informatii utile cabinetului veterinar, cum ar fi pacientii ce au fost in stare de urgenta ;
* Medicul sa poata introduce date relevante pentru aplicatie cum ar fi date personale sau date despre pacienti;

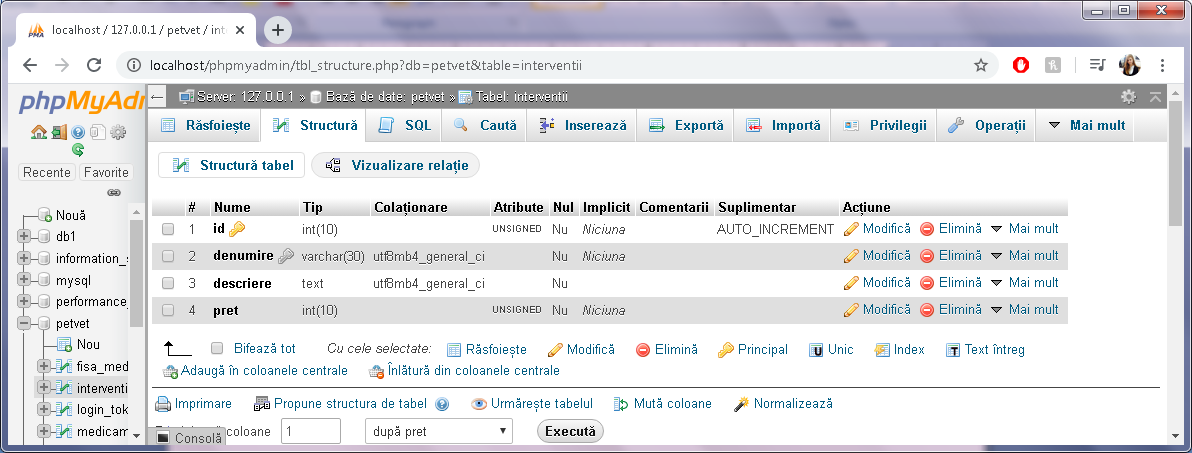
1. Proiectarea bazei de date

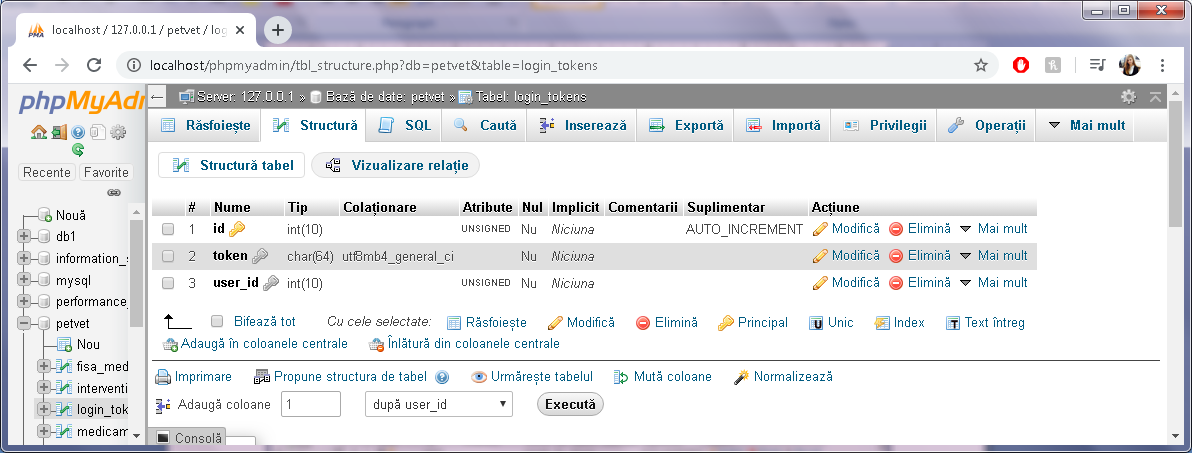
In total am folosit 8 tabele: 6 tabele ce reprezinta componentele, 1 tabel ce retrezinta relatia dintre 2 tabele si inca un tabel ce tine cont de utilizatorul logat numit login\_tokens.

* **login\_tokens**: in acest tabel se retin niste token-uri/ cookie-uri unice si sunt active cat timp un utilizator este logat. Un utilizator logat inseamna un token, dupa delogare tokenul va fi sters din tabel. Acest tabel este folosit doar pentru a tine minte ce utilizator este logat deoarece acesta poate sta logat timp de 7 zile, dupa care va fi delogat automat.
* **medici**: acest tabel retine datele personale ale medicilor inregistrati: nume, prenume, cnp, username-ul cu care se logheaza si parola si specializarea acestuia;
* **pacienti**: acest tabel retine datele despre pacienti: nume, rasa, specia si id-ul proprietarului corespunzator proprietarului lui;
* **proprietari**: acest tabel retine datele personale despre proprietari: nume, prenume, cnp;
* **medicamente**: retine datele despre fiecare medicament: denumirea, descrierea, cate bucati sunt si specia destinata;
* **interventiii**: retine datele despre fiecare interventie ce poate fi efectuata de catre medici: denumire, descriere si pret;
* **medici\_interventii**: tine evidenta interventiilor si a medicilor, mai exact ne spune ce interventii stiu medicii sa realizeze;
* **fisa\_medicala**: reprezinta istoricul medical al pacientilor si retine datele: data la care s-a efectuat fisa, pacientul, ce medicament i s-a administrat, de ce interventie a avut parte, si de ce afectiune sufera

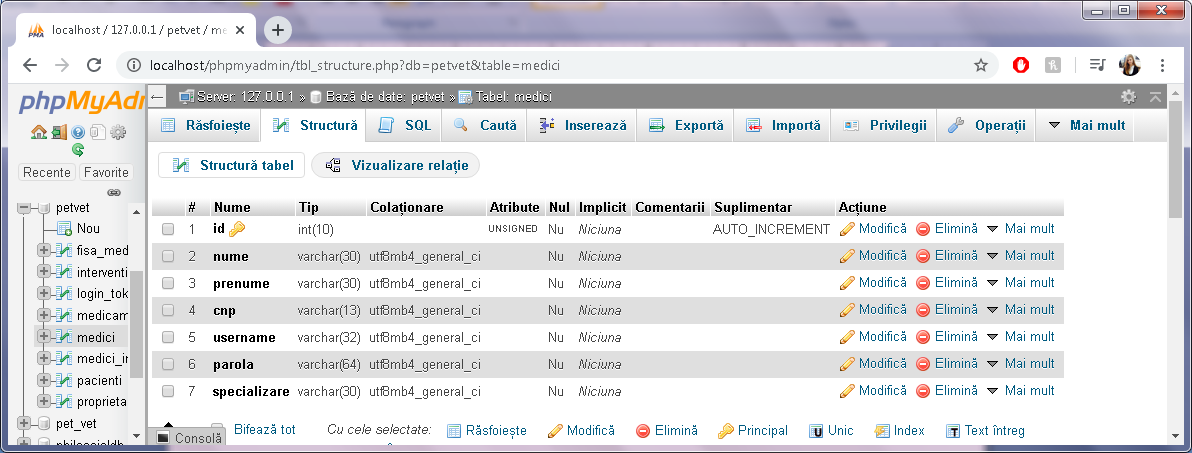
Tabelele au forma:

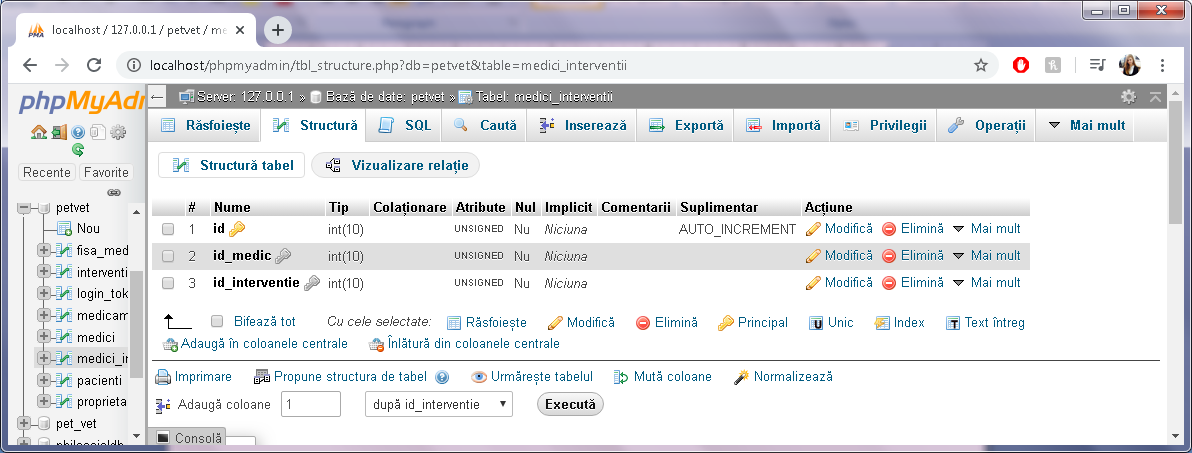


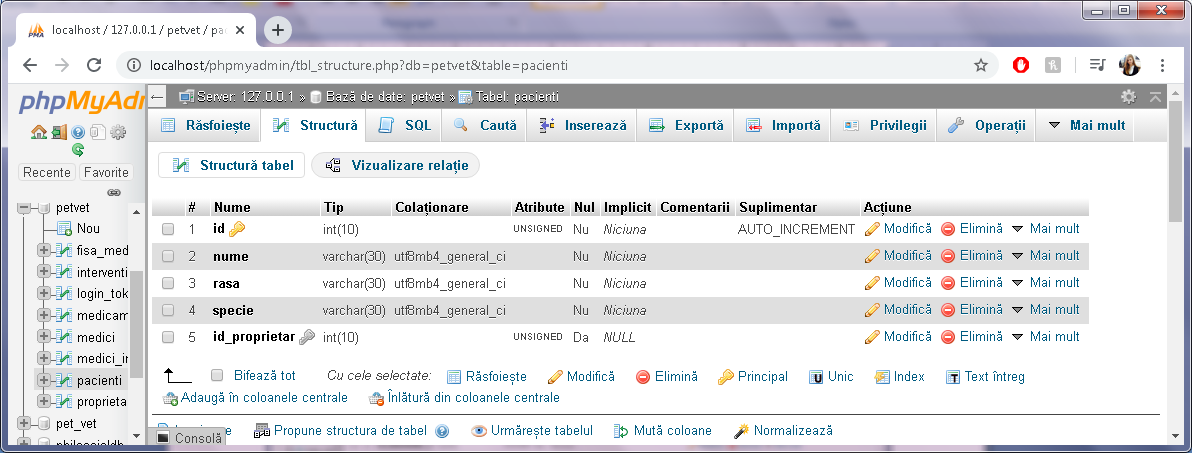


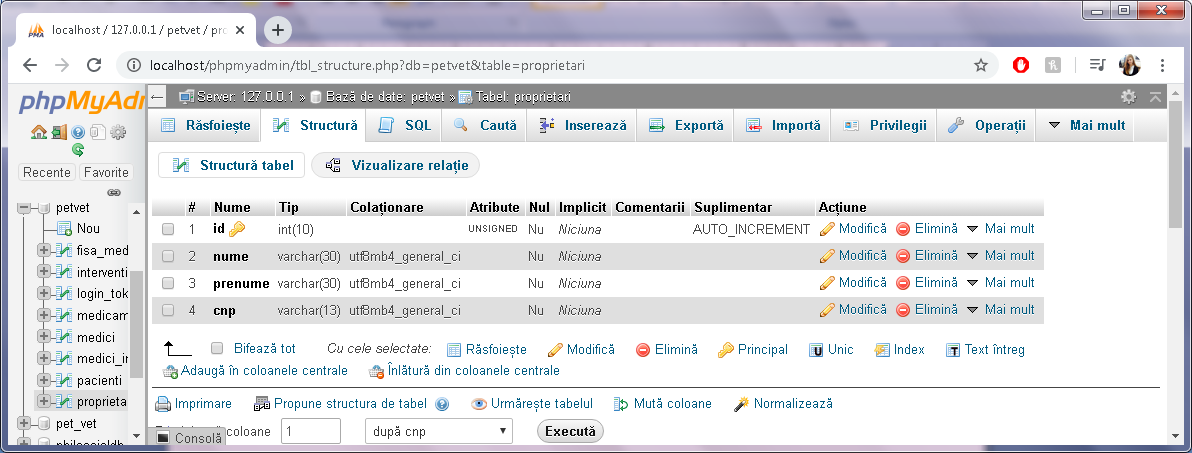




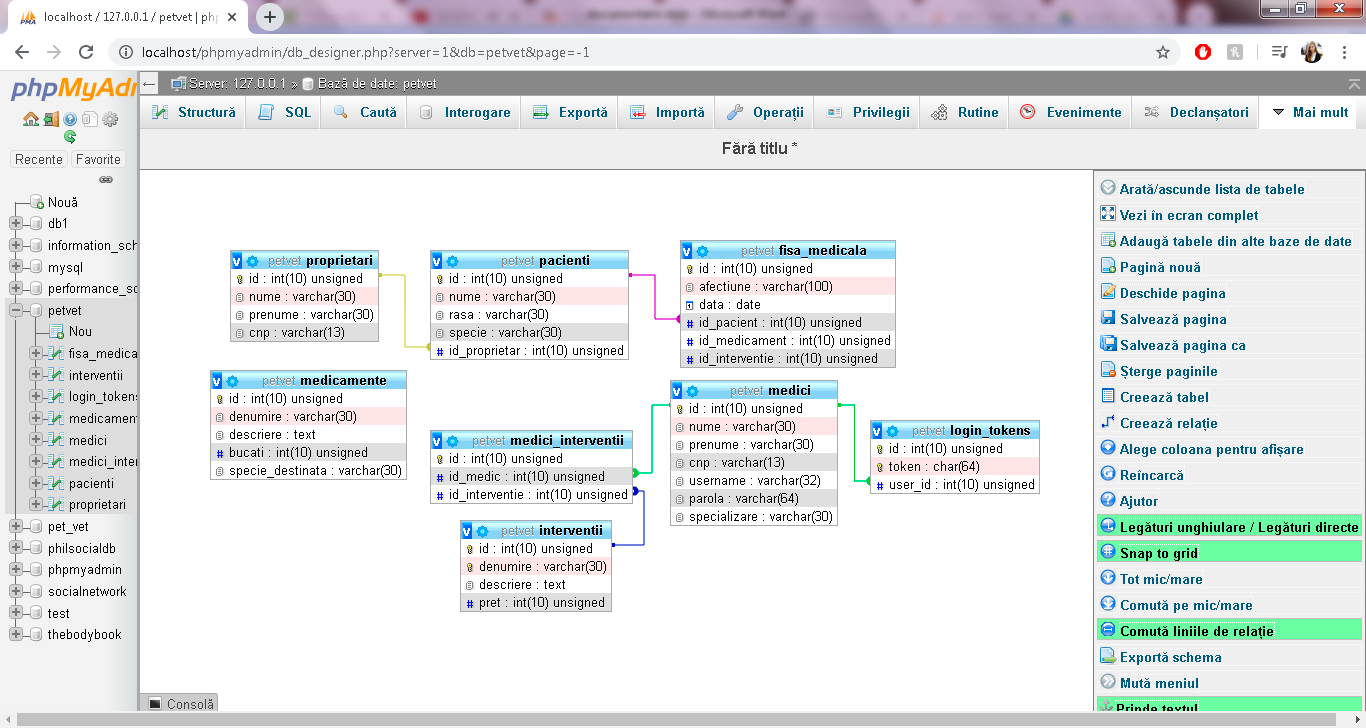








Si au urmatoarele relatii:



**One-to-Many**: Proprietar(cheia primara: id) – Pacient(cheia straina: id\_proprietar)

-un pacient(animal de companie) are un singur proprietar, dar un proprietar poate sa detina mai multe animale de companie;

Pacient(cheia primara: id) – Fisa Medicala(cheia straina: id\_pacient)

-fisa medicala este specifica unui singur macient, dar un pacient poate avea mai multe fise medicale deoarece poate vizita de mai multe ori cabinetul medical;

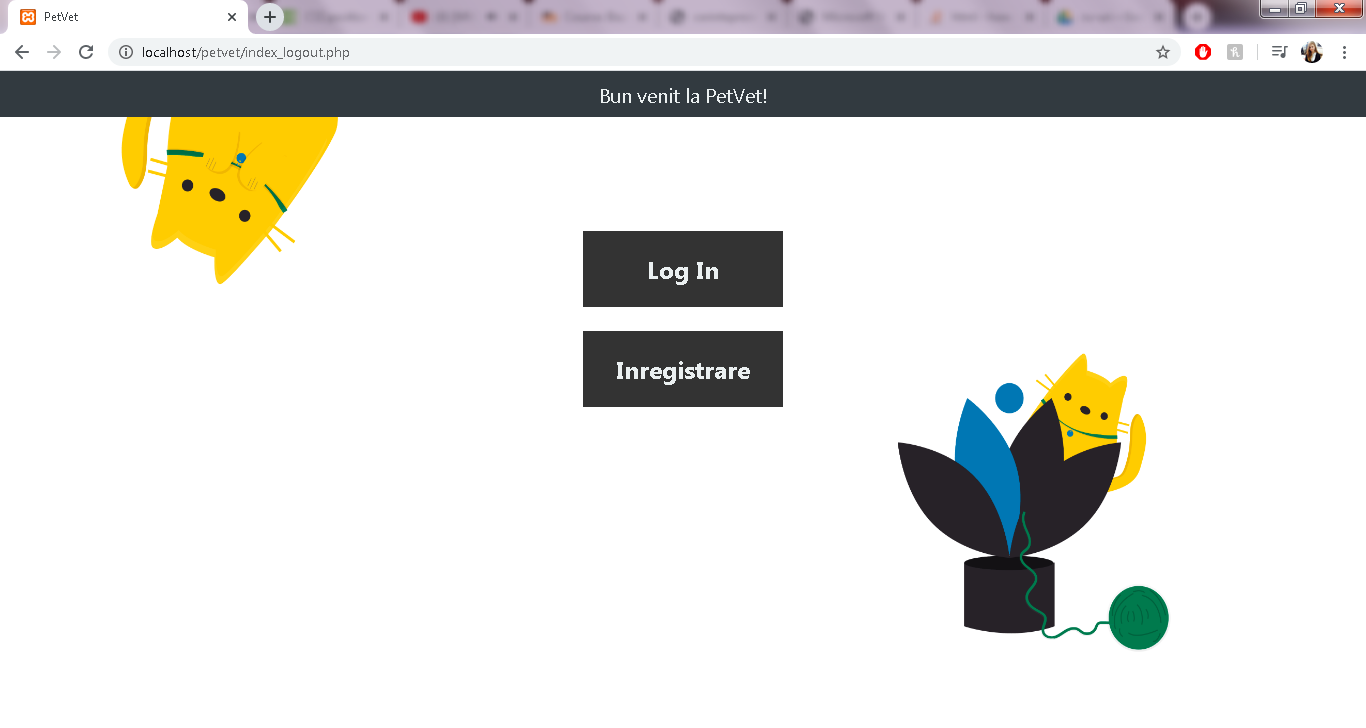
**Many-to-Many**: Medici – Interventii(cheie staina compusa: id\_medic, id\_interventie)

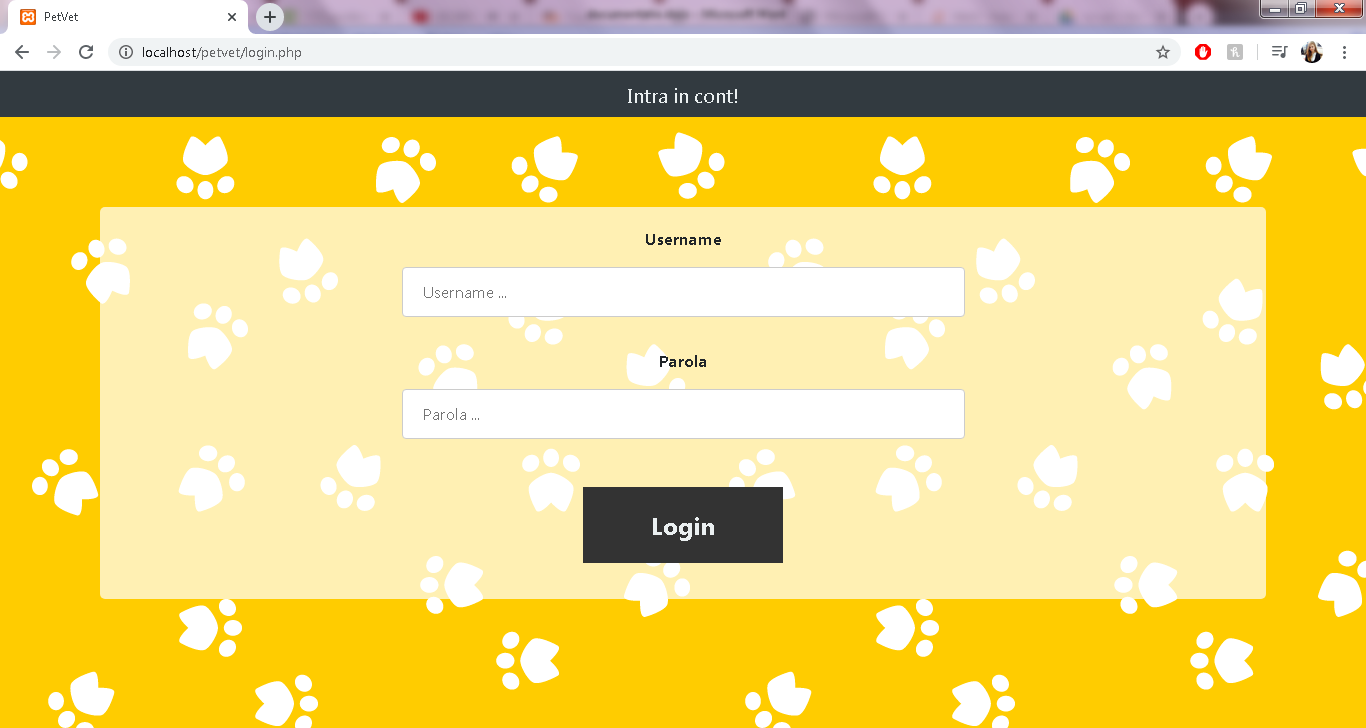
-un medic poate efectua mai multe interventii, dar aceleasi interventii pot fi efectuate de mai multi medici;

1. Proiectarea interfetei vizuale si implementarea acesteia intr-un mediu de programare

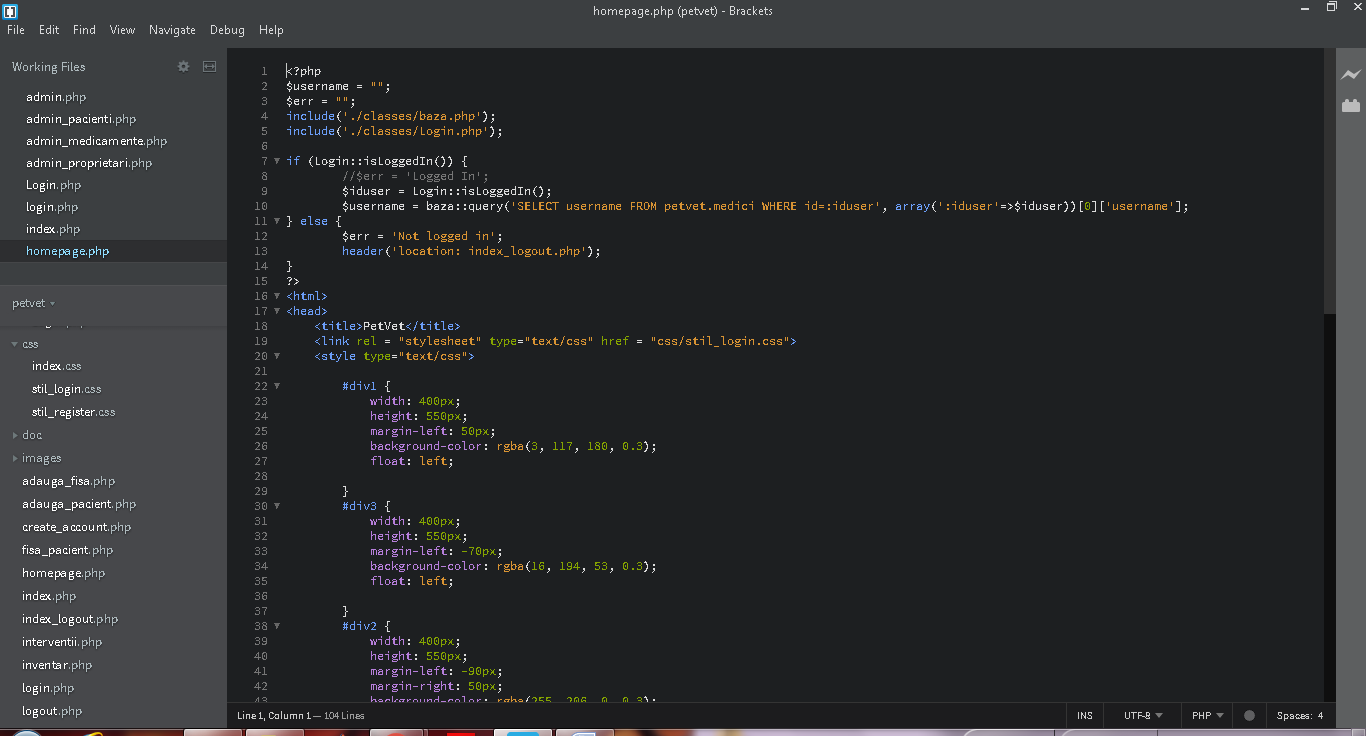
Am ales ca mediu de programare sa folosesc php + html + css (pentru interfata). Am 3 fisiere css pe care le folosesc in mai multe documente php, iar unde este nevoie de lucruri in plus cum ar fi: butoane diferite atunci in header intre tagurile <style></style> introduc componentele css necesare. Legat de partea de programare folosesc in acelasi timp php si html, exceptand baza.php unde fac conexiunea cu baza de date si Login.php unde verific tokenul de logare si returnez userul logat in aplicatie. In aceste doua fisiere din urma am doar cod php. Pentru restul fisierelor, chiar daca pare mai incurcat codul pentru ca este php combinat cu html, am gasit o ordine a scrierii codului astfel incat sa aiba sens.

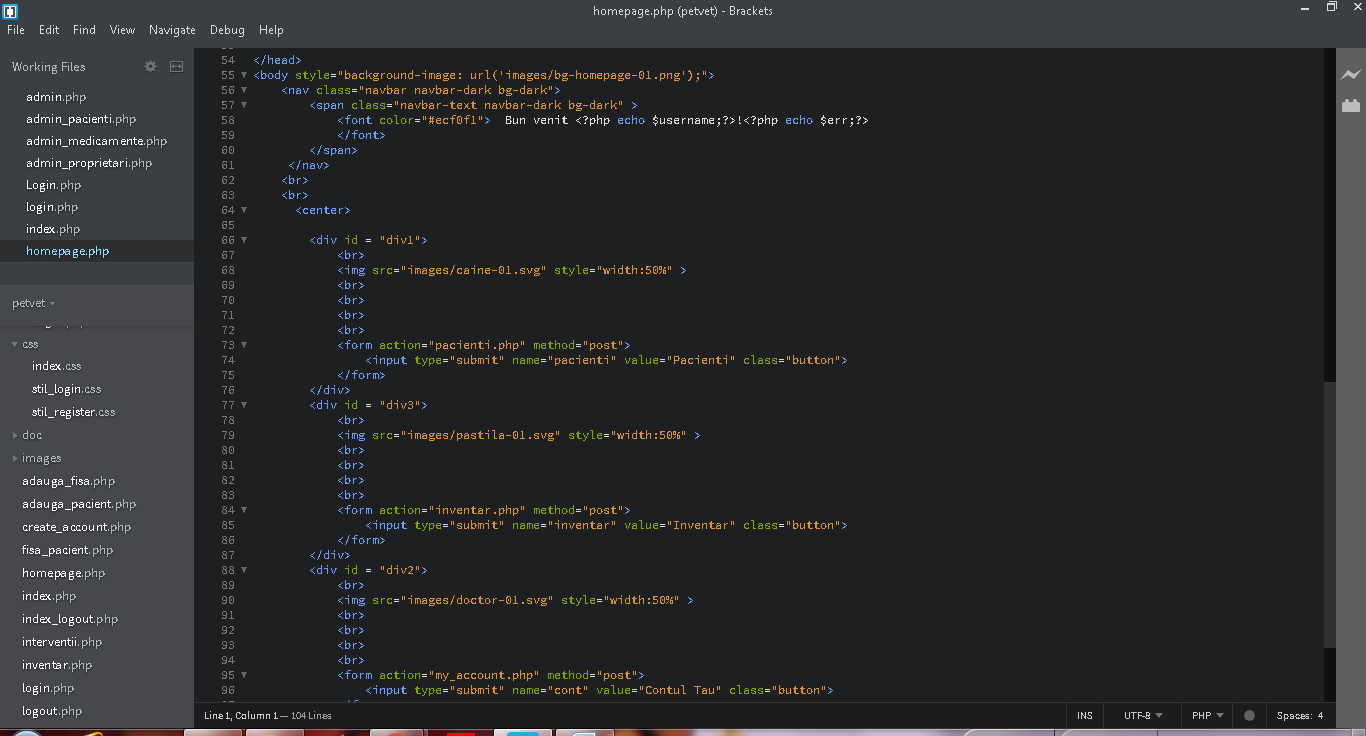
**Exemple din interfata:**

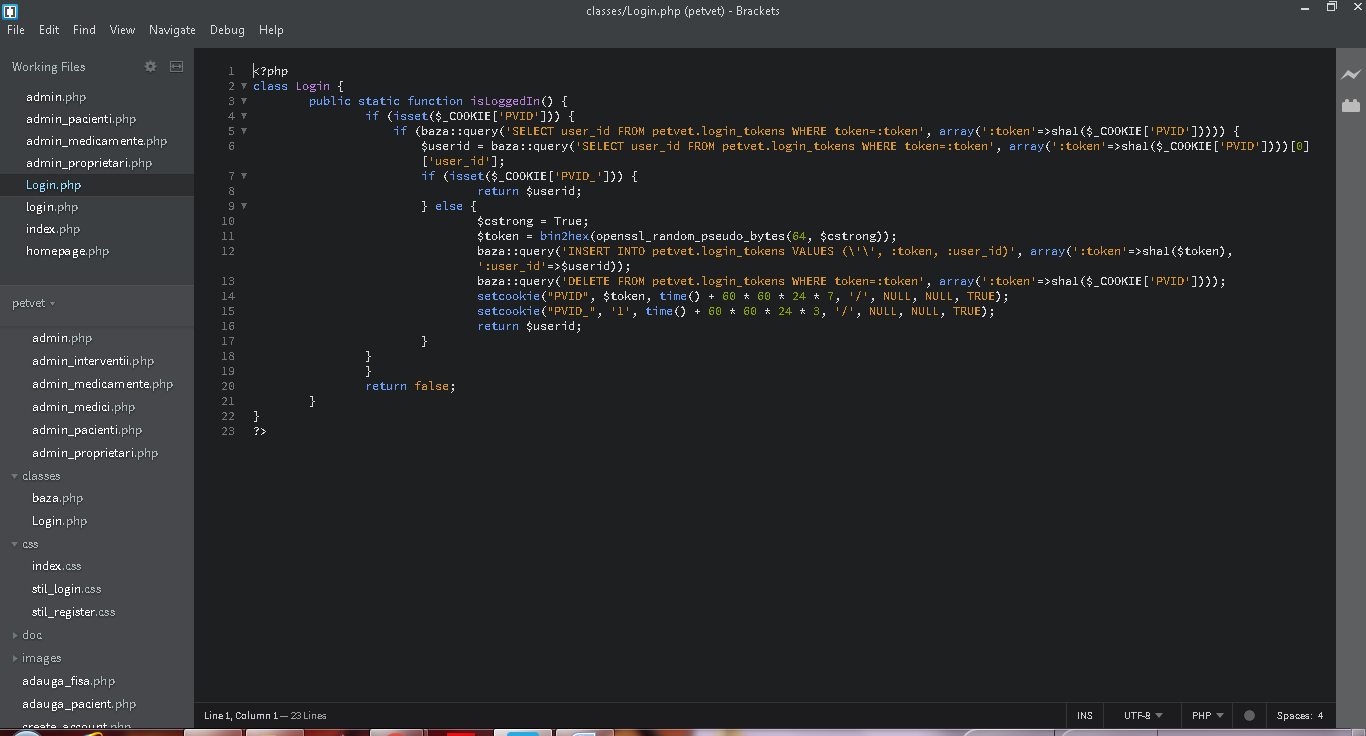




**Exemple din cod:**

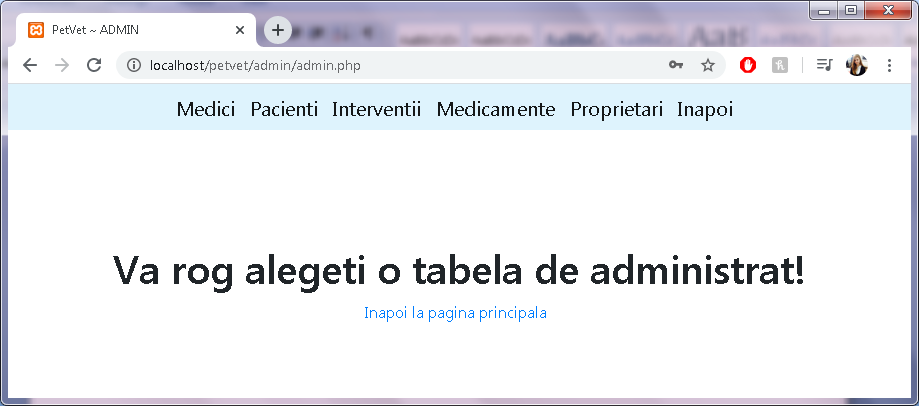
****

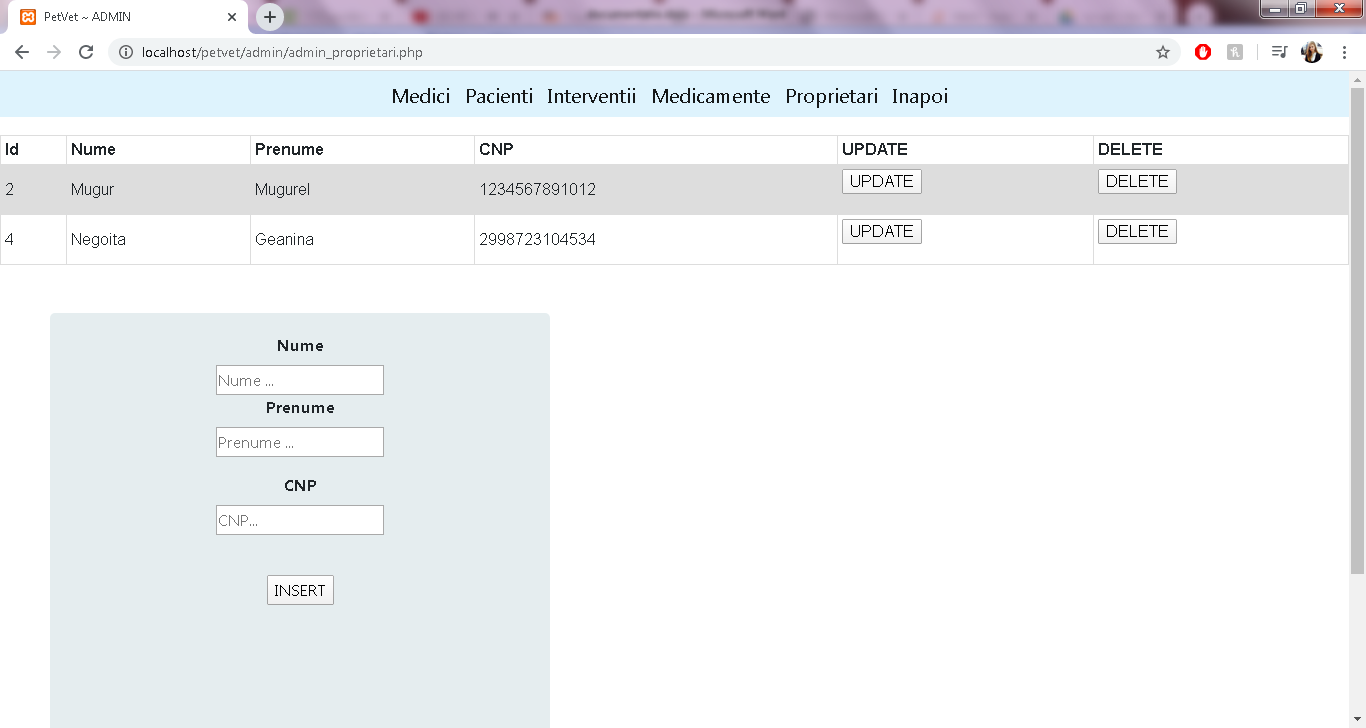
****

****

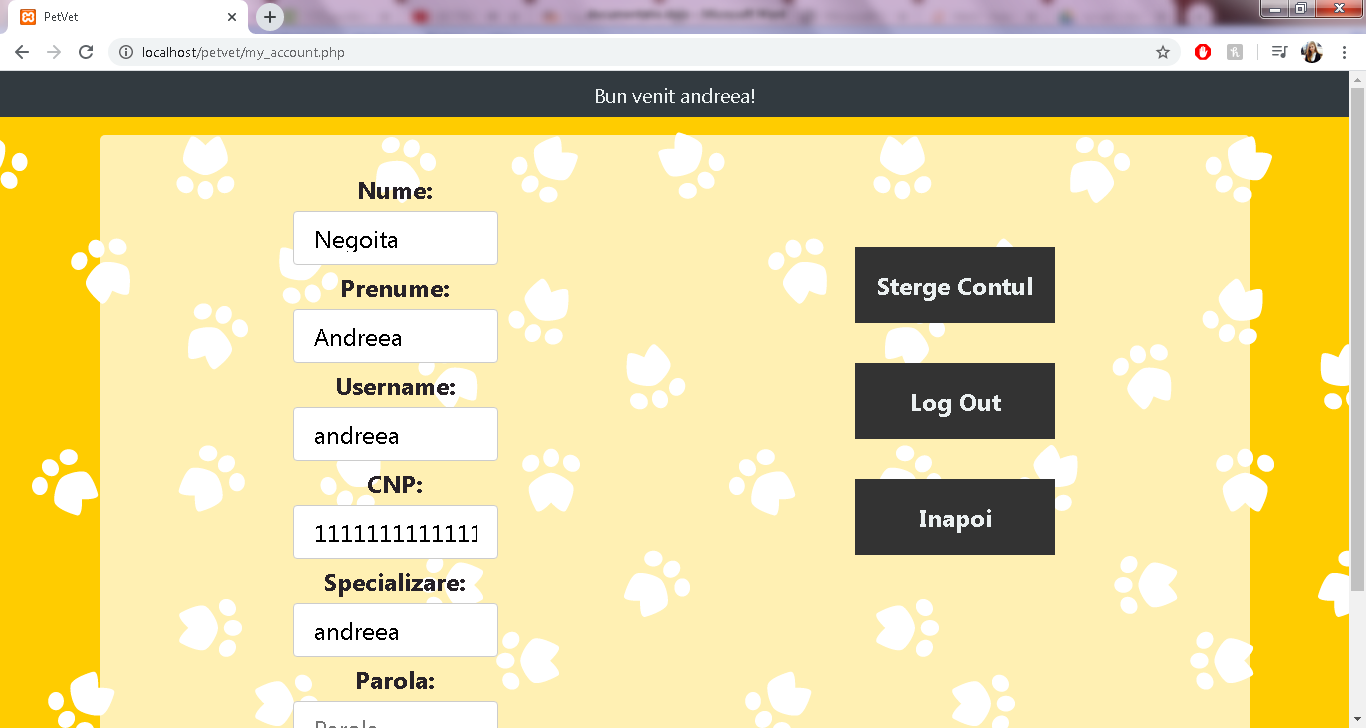
1. **Corelarea interfetei vizuale cu baza de date**

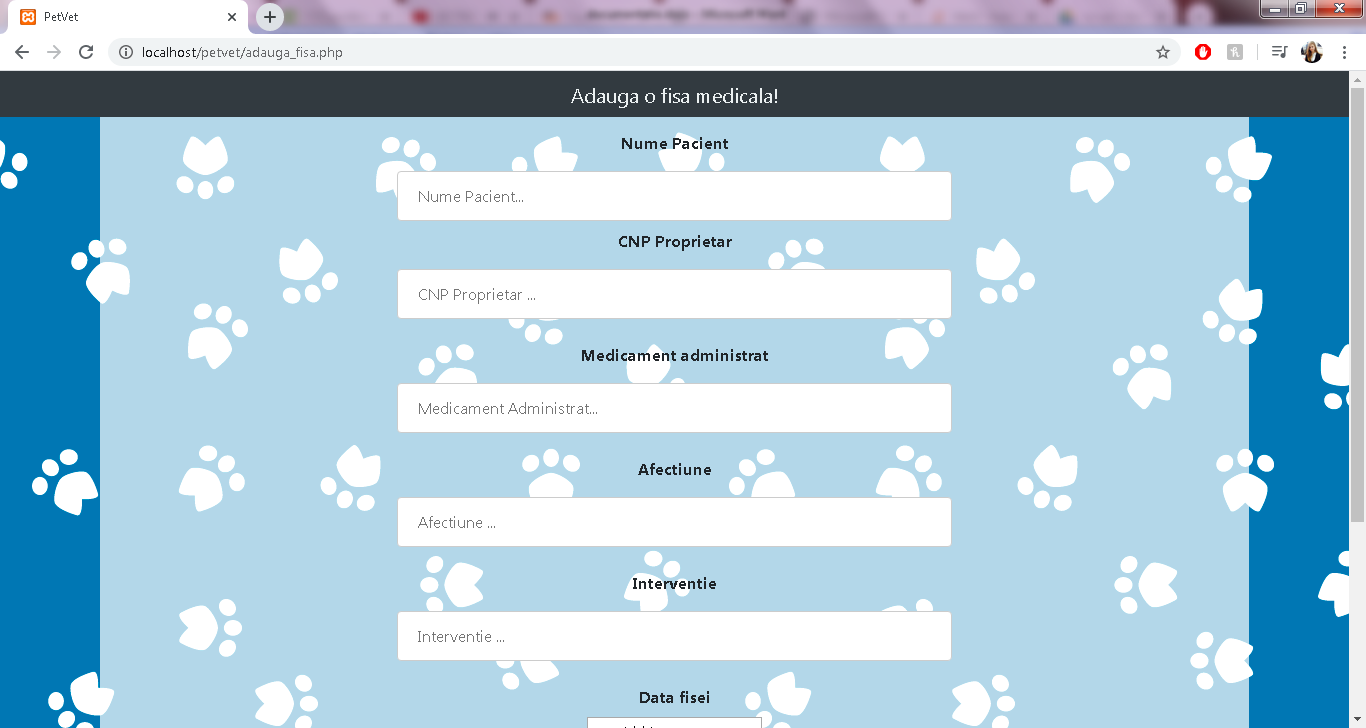
In mare parte pentru insert, update, delete se ocupa administratorul(username: admin2020 parola: parola2020) care face aceste operatii pentru 5 tabele:





Dar si medicul poate face insert, update, delete pentru datele proprii sau sa adauge pacienti si fise medicale sau sa adauge in tabelul de legatura cu interventiile:





In componenta aplicatiei am folosit multe interogari, dar le voi mentiona pe cele mai importante:

Interogari simple:

1. Afisez pe ecran toate interventiile utilizatorului curent logat

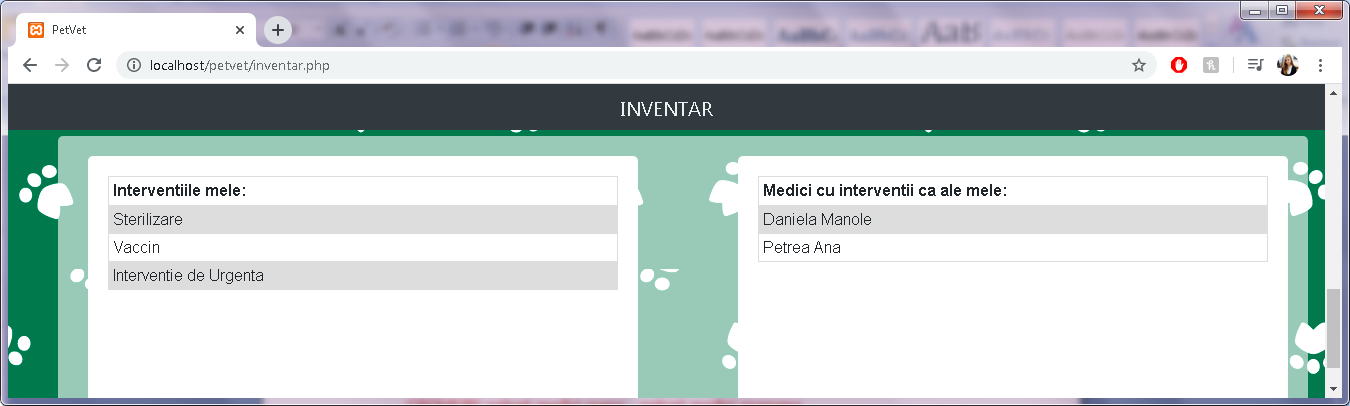
Se gaseste in fisierul inventar.php la linia 194 iar pe ecran apare atunci cand utilizatorul intra in pagina inventarului.

SELECT \* FROM petvet.interventii INNER JOIN petvet.medici\_interventii ON petvet.interventii.id = petvet.medici\_interventii.id\_interventie WHERE petvet.medici\_interventii.id\_medic = :idmedic ;

1. Afisez pe ecran medicii care au interventiile ca ale mele

Se gaseste in fisierul inventar.php la linia 211, iar pe ecran apare atunci cand utilizatorul intra in pagina inventarului.

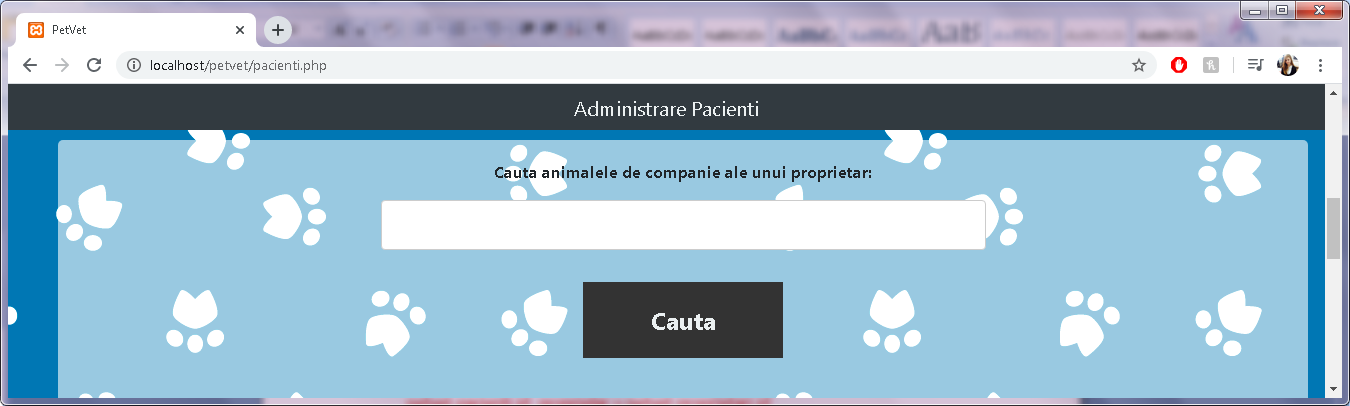
SELECT DISTINCT petvet.medici.nume, petvet.medici.prenume FROM petvet.medici INNER JOIN petvet.medici\_interventii ON petvet.medici.id = petvet.medici\_interventii.id\_medic WHERE petvet.medici.id != :idmedic AND petvet.medici\_interventii.id\_interventie IN (SELECT petvet.interventii.id FROM petvet.interventii INNER JOIN petvet.medici\_interventii iON petvet.interventii.id =petvet.i.id\_interventie WHERE petvet.i.id\_medic =:idmedic)

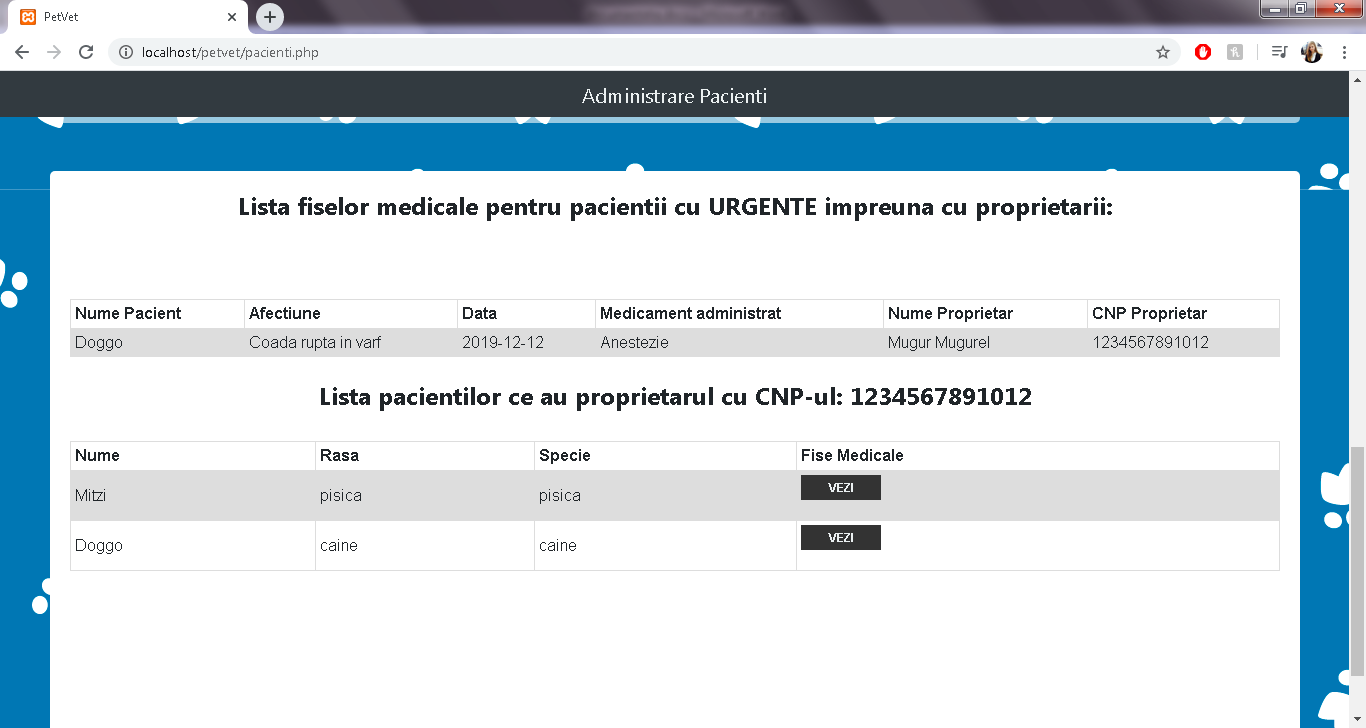


1. In functie de CNP-ul unui proprietar, pot sa afisez ce animale detine

Se gaseste in fisierul pacienti.php la linia 270 si apare pe ecran formularul pentru introducerea cnp-ului proprietarului in pagina despre pacienti.

SELECT petvet.pacienti.id, petvet.pacienti.nume, petvet.pacienti.rasa, petvet.pacienti.specie FROM petvet.pacienti INNER JOIN petvet.proprietari ON petvet.pacienti.id\_proprietar = petvet.proprietari.id WHERE petvet.proprietari.id = :idproprietar





1. Cati bani a achitat un proprietar de-a lungul timpului platind interventiile

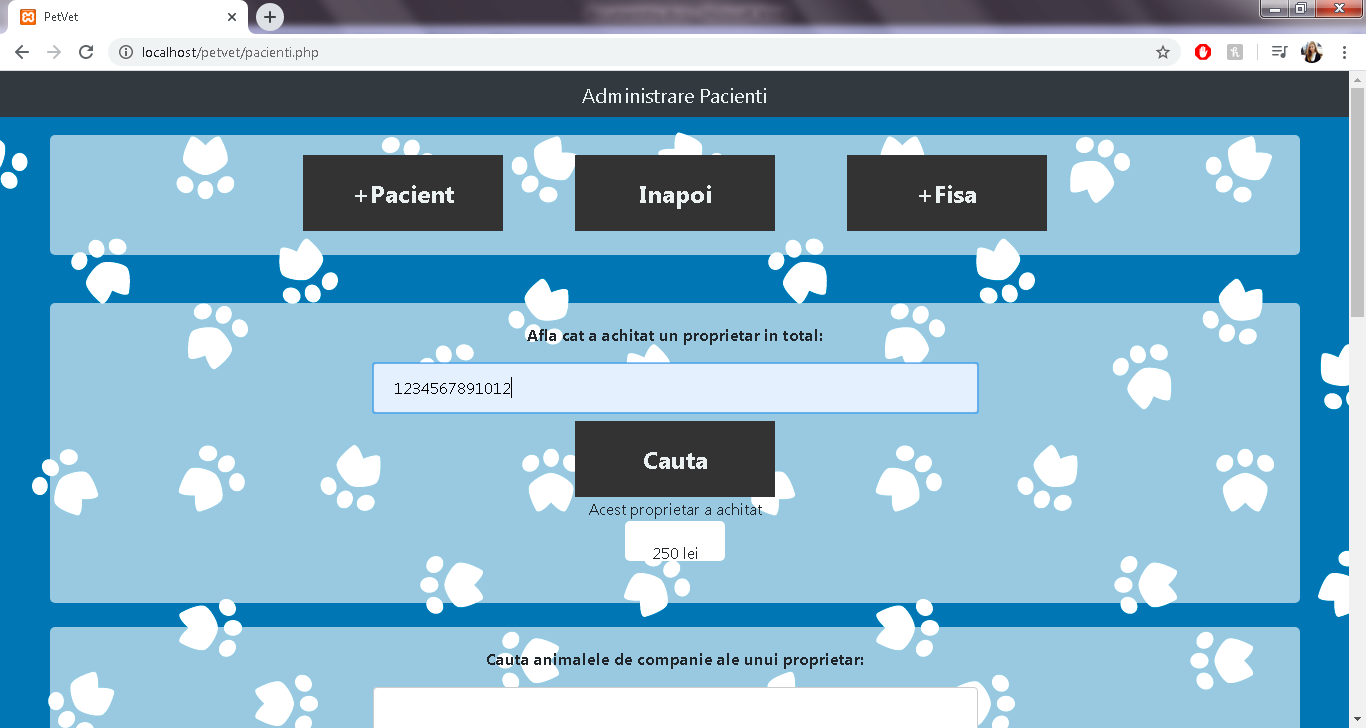
Se gaseste in fisierul pacienti.php la linia 27, iar formularul pentru introducerea proprietarului se gaseste in pagina despre pacienti.

SELECT SUM(petvet.interventii.pret) FROM petvet.interventii INNER JOIN petvet.fisa\_medicala ON petvet.interventii.id = petvet.fisa\_medicala.id\_interventie

INNER JOIN petvet.pacienti ON petvet.fisa\_medicala.id\_pacient = petvet.pacienti.id

INNER JOIN petvet.proprietari ON petvet.pacienti.id\_proprietar = petvet.proprietari.id

WHERE petvet.proprietari.id = :idproprietar



1. Ce pacienti au fost in stare de urgenta, sa se afiseze fisele lor medicale

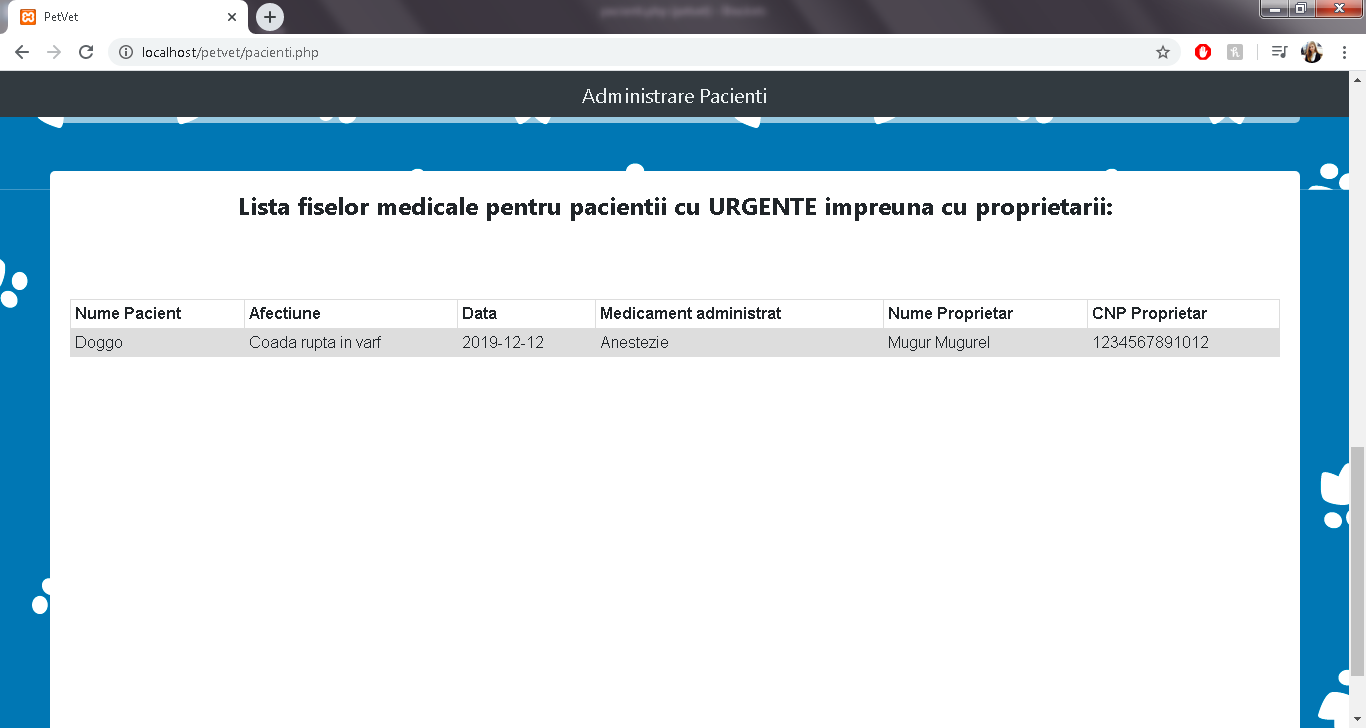
Se gaseste in fisierul pacienti.php la linia 215, iar datele rezultate se afla in pagina despre pacienti.

SELECT \* FROM petvet.fisa\_medicala INNER JOIN petvet.interventii ON petvet.fisa\_medicala.id\_interventie = petvet.interventii.id WHERE petvet.interventii.denumire=:denumire

1. Afisarea proprietarilor pacientilor cu urgente

Se gaseste in fisierul pacienti.php la linia 236, iar datele rezultate se afla in pagina despre pacienti.

SELECT petvet.proprietari.nume FROM petvet.proprietari INNER JOIN petvet.pacienti ON petvet.proprietari.id = petvet.pacienti.id\_proprietar WHERE petvet.pacienti.id=:id



Interogari complexe

1. Afisarea fiselor medicale unde pastilele sunt pe terminate(sub 30 de bucati)

Se gaseste in fisierul pacienti.php la linia 304. In urma apasarii butonului asociat din pagina despre pacienti, mai jos se va afisa lista fiselor medicale rezultate interogarii.

SELECT \* FROM petvet.fisa\_medicala

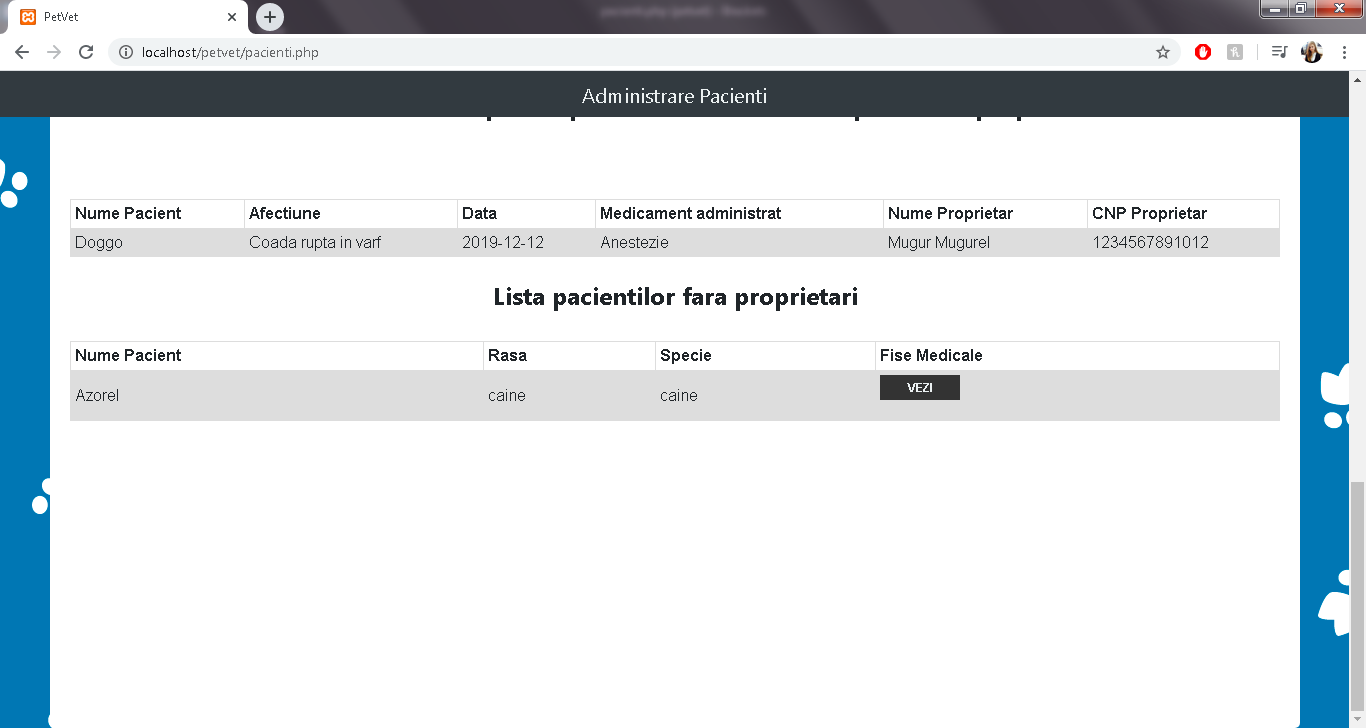
WHERE 30 > (SELECT bucati FROM petvet.medicamente WHERE petvet.medicamente.id = petvet.fisa\_medicala.id\_medicament)

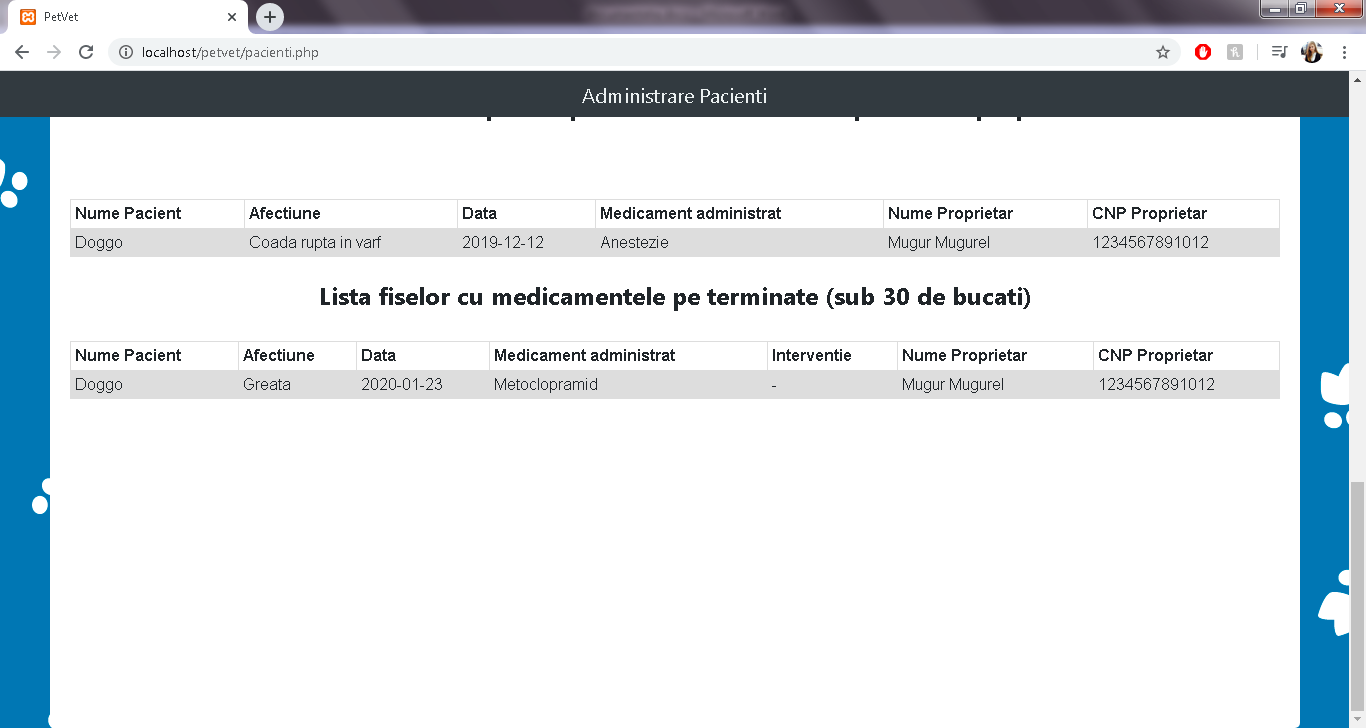
1. Afisarea pacientilor fara proprietar

Se gaseste in fisierul pacienti.php la linia 357. In urma apasarii butonului asociat din pagina despre pacienti, mai jos se va afisa lista fiselor medicale rezultate interogarii.

SELECT \* FROM petvet.pacienti WHERE id NOT IN (SELECT petvet.pacienti.id FROM petvet.pacienti INNER JOIN petvet.proprietari ON petvet.pacienti.id\_proprietar = petvet.proprietari.id



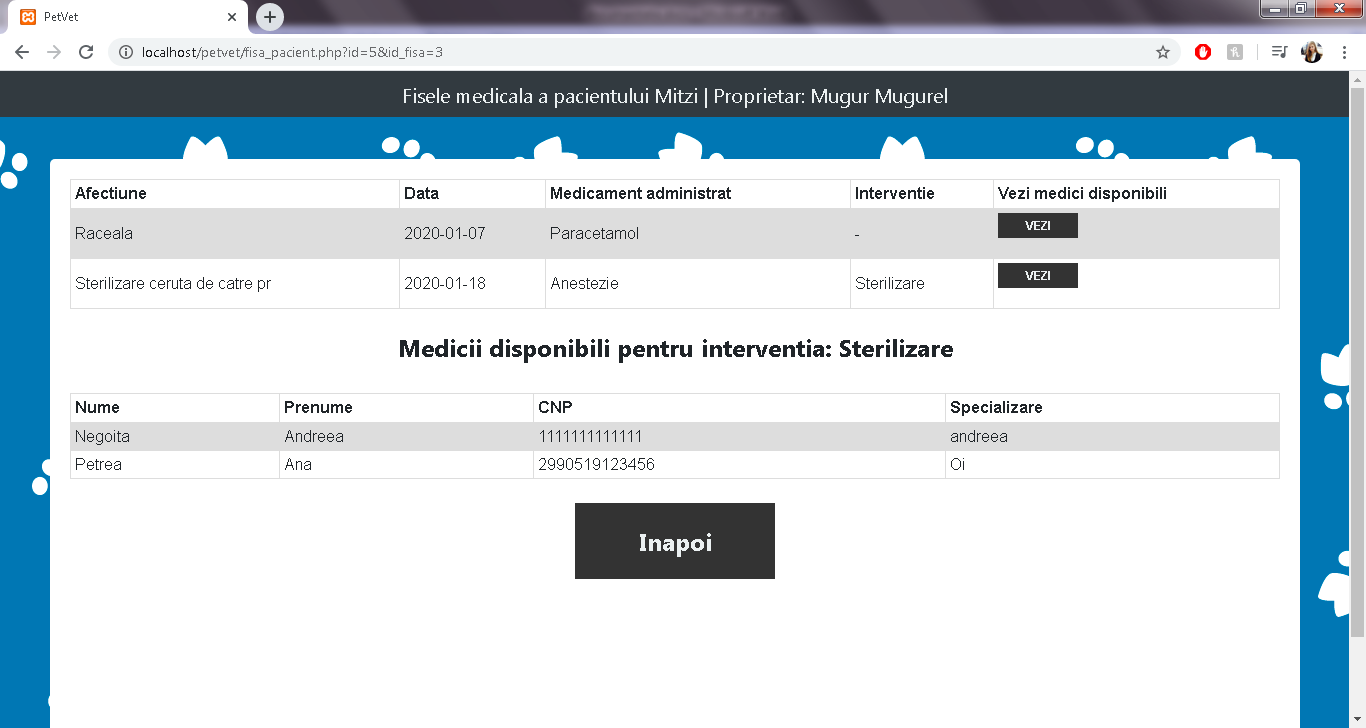




1. Afisarea medicilor responsabili de a face interventia dintr-o fisa medicala

Se gaseste in pagina fisa\_pacient.php la linia 193. Pentru a afisa medicii responsabili de o interventie de procedeaza astfel: in pagina pacienti se cauta animalele de companie ale unui proprietar. Pentru fiecare animal putem intra in pagina despre fisele lui medicale. In dreptul unei fise medicale avem un buton care ne arata ce medici se ocupa de interventia respectiva fisei acestui pacient.

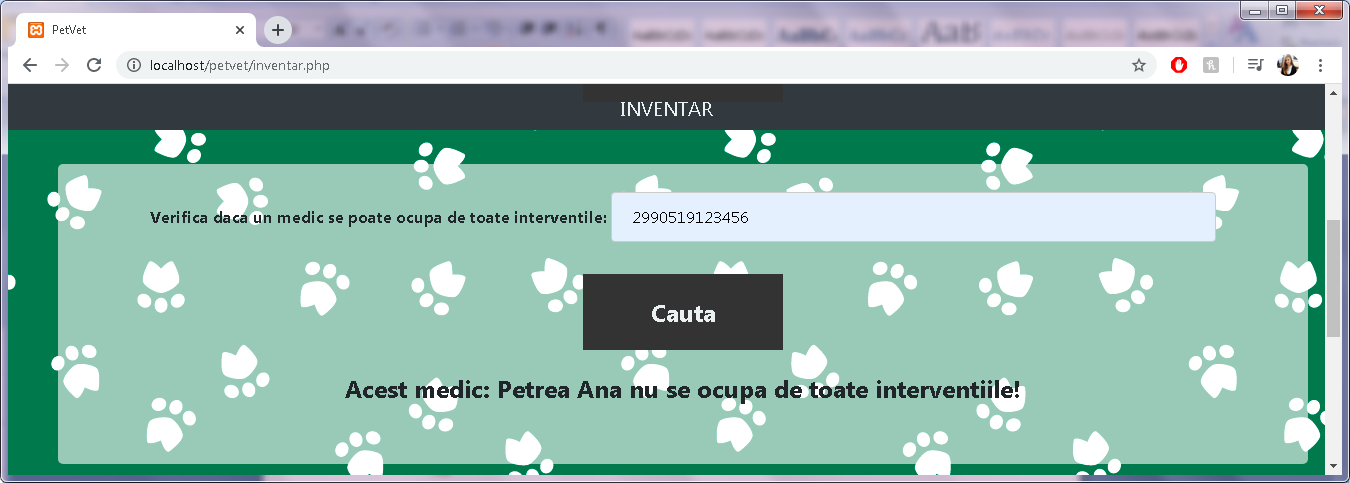
SELECT \* FROM petvet.medici WHERE id IN (SELECT id\_medic FROM petvet.medici\_interventii WHERE id\_interventie = (SELECT id\_interventie FROM petvet.fisa\_medicala WHERE id = :id)



1. In functie de CNP-ul unui medic putem afisa daca acesta este responsabil de toate interventiile sau nu

Se gaseste in fisierul inventar.php la linia 169, iar pe ecran se gaseste in pagina inventarului. Se introduce cnp-ul medicului iar butonul de cautare ne arata daca medicul respectiv se ocupa de toate interventiile sau nu.

SELECT \* FROM petvet.medici WHERE (SELECT COUNT(\*) FROM petvet.interventii) = (SELECT COUNT(petvet.medici\_interventii.id\_interventie) FROM petvet.medici\_interventii WHERE petvet.medici\_interventii.id\_medic = :id



Fisierele responsabile de partea de administrare se gasesc in folderul admin, iar fisierele responsabile de conectarea la baza de date si partea de login se afla in folderul classes. In folderul css se gasesc documentele css importate in cele php.

Automat aplicatia deschide pagina index.php, daca utilizatorul este deja logat suntem redirectionati in homepage.php, iar daca nu este logat ajungem in pagina index\_nelogat.php. In paginile de administratie se ajunge doar daca ne logam cu datele: username = admin2020 si parola = parola2020. In celelalte pagini se ajunge dupa logare prin homepage: pacienti.php, inventar.php si my\_account.php . De aici in functie de preferinte se trece in restul de pagini. In fiecare pagina se verifica daca utilizatorul este logat, pentra ca in cazul unei erori sa afiseze paginile pentru persoane nelogate.

Daca doriti sa intrati in conturile utilizatorilor deja inregistrati de mine puteti scrie atat la username cat si la parola valoarea username-ului din tabele. Acest lucru nu este o regula, doar ca am ales sa introduc astfel datele pentru a nu tine minte parolele. Daca doriti sa va inregistrati ca medic puteti sa puneti orice parola doriti, acesta va fi criptata pentru a fi mai sigura.