UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DOMENIUL: Calculatoare și tehnologia informației

**Aplicație web - gestionarea**

**unei baze de date pentru o**

**clinică medicală privată**

Proiect la disciplina

Baze de date

**Student:**

Ceică Sergiu 1310A

**Profesor coordonator:**

Cătălin Mironeanu

Capitolul 1 – Introducere

In zilele noastre, importanta de a avea o evidenta digitala a informatiilor si un acces rapid la citirea si scrierea acestora este o prioritate in calitatea serviciilor oferite de orice institutie primeste si ofera informatii in fiecare zi. Abundenta de hartii, semnaturi si rapoarte pot fi realizate acum mult mai rapid si mai eficient. Cu doar cateva click-uri putem realiza documente generate automat fara greseli.

Aplicatia isi propune sa gestioneze baza de date a unei clinici medicale private utilizand tehnologii web care fac interfata cu utilizatorul mult mai intuitiva, limitand de asemenea posibilitatea introducerii de date eronate. Informatiile sunt stocate intr-o baza de date locala, aplicatia neavand astfel nevoie de un server remote. De asemenea aceasta poate fi accesata fara conexiune la internet, introducand in browser adresa: <http://127.0.0.1:5000> dupa rularea scriptului.

Verificarea corectitudinii datelor este verificata automat de catre script, putand fi adaugate oricand noi constrangeri pentru acest lucru. Aplicatia poate fi folosita cu usurinta de oricine, facand gestiunea bazei de date mult mai simpla pentru personalul medical.

Capitolul 2 – Aplicații necesare

Aplicatia necesita instalarea urmatoarelor aplicatii:

**Python 3.7** sau mai nou. Aplicatia este realizata in limbajul de programare Python , de aceea este obligatoriu ca acesta sa fie instalat pe dispozitivul pe care se doreste rularea scriptului.

Se poate descarca din: <https://www.python.org/downloads/>

**Biblioteci folosite:**

**Flask 2.2.2**: Flask este un API (Application Programming Interface) Python ce permite construirea de aplicații web. A fost dezvoltat de Armin Ronacher (un programator austriac si un speaker cunoscut la conferintele în domeniul software).

Aceasta biblioteca poate fi instalata utilizand comanda: pip install Flask .

**Flask\_Login 0.6.2**: Flask-Login oferă gestionarea sesiunilor utilizatorilor pentru Flask. Se ocupă de sarcinile obișnuite de conectare, deconectare și amintire a sesiunilor utilizatorilor dvs. pe perioade lungi de timp.

**Flask\_sqlalchemy 3.0.2**: Flask-SQLAlchemy este o extensie pentru Flask care adaugă suport pentru SQLAlchemy aplicației. Simplifică utilizarea SQLAlchemy cu Flask prin configurarea obiectelor și modelelor comune pentru utilizarea acestor obiecte, cum ar fi o sesiune legată de fiecare cerere web, modele și motoare.

**Werkzeug 2.2.2**: Werkzeug este o bibliotecă utilitară pentru limbajul de programare Python , cu alte cuvinte un set de instrumente pentru aplicații WSGI ( Web Server Gateway Interface ) și este licențiat sub licență BSD . Werkzeug poate realiza obiecte software pentru funcții de solicitare, răspuns și funcții utilitare.

Capitolul 3 – Interfata cu utilizatorul

Pentru a proteja datele, aplicatia necesita crearea unui cont care sa permita accesul la informatii. Pagina de Login este prima care se deschide la accesarea aplicatiei, ea arata ca in imaginea de mai jos. Logarea utilizatorilor se face cu ajutorul adresei de e-mail si a unei parole.

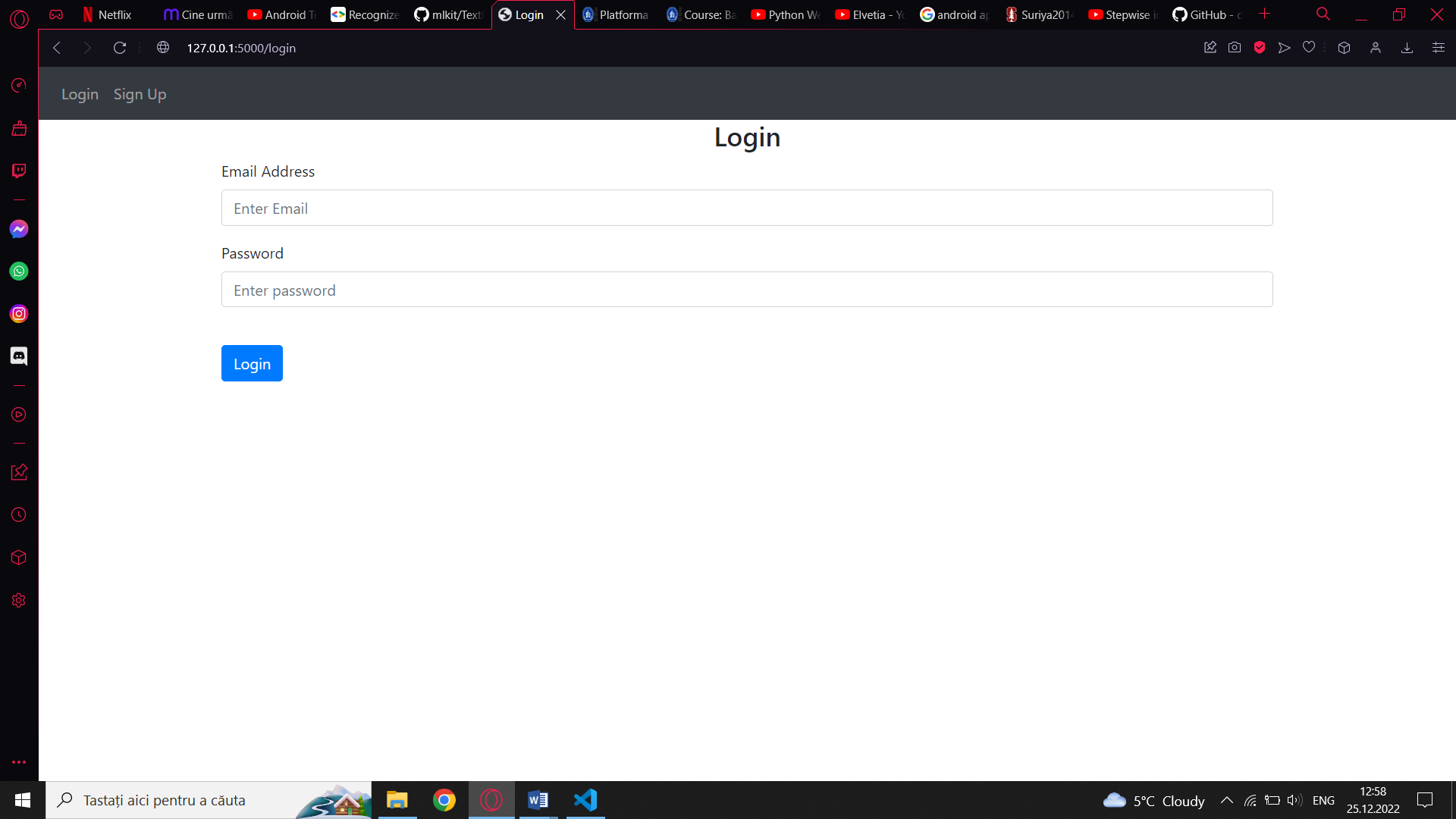


Fig 1 – Pagina Login

In cazul in care nu aveti un cont, crearea se face accesand butonul Sign Up din bara de navigatie. Acesta va redirectiona utilizatorul pe pagian de Sign Up unde, dupa completarea corecta a tuturor campurilor, va avea acces la baza de date. La completarea incorecta, aplicatia trimite un mesaj de eroare in care se sugereaza o corectie pentru completarea campurilor.

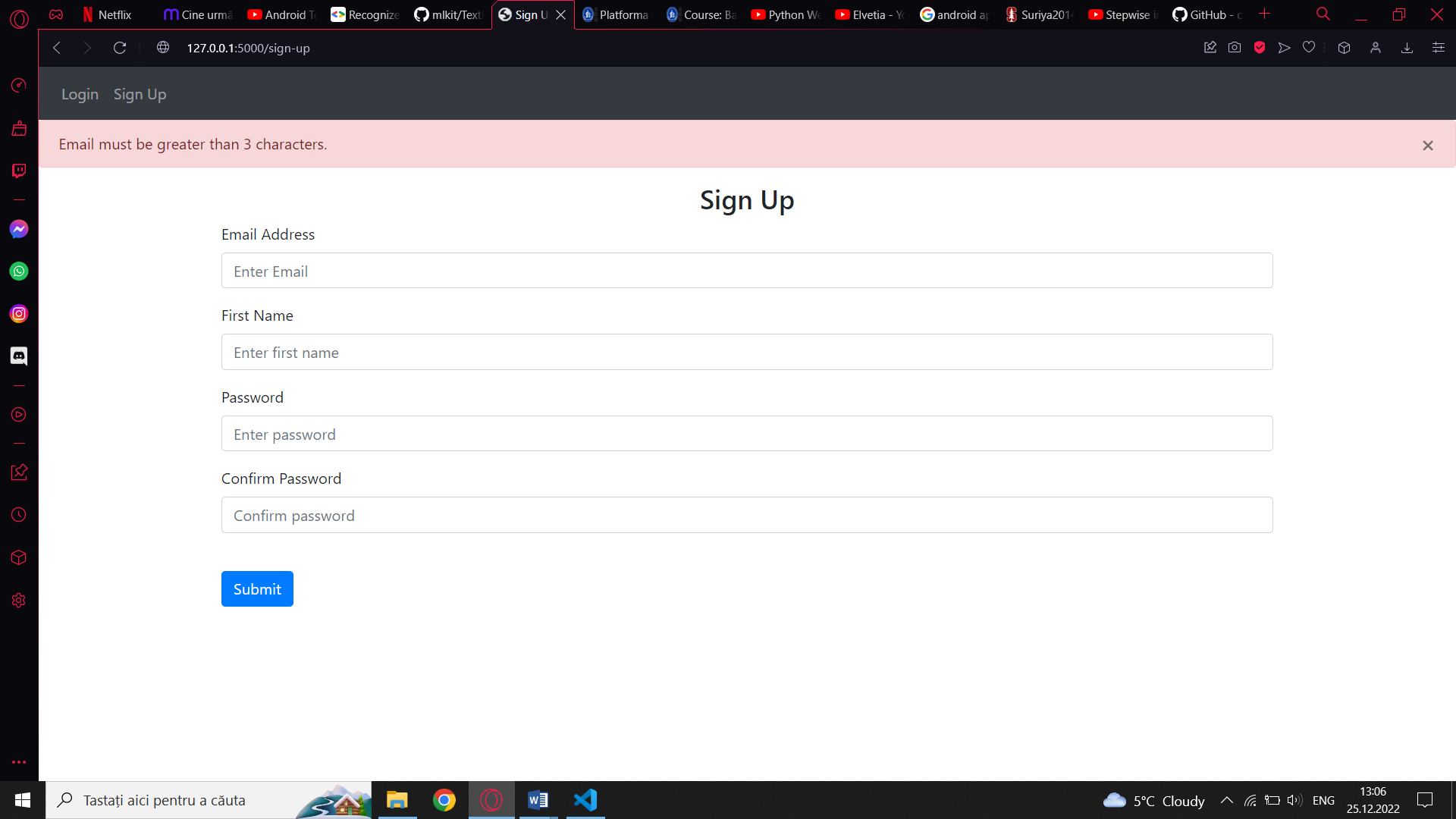


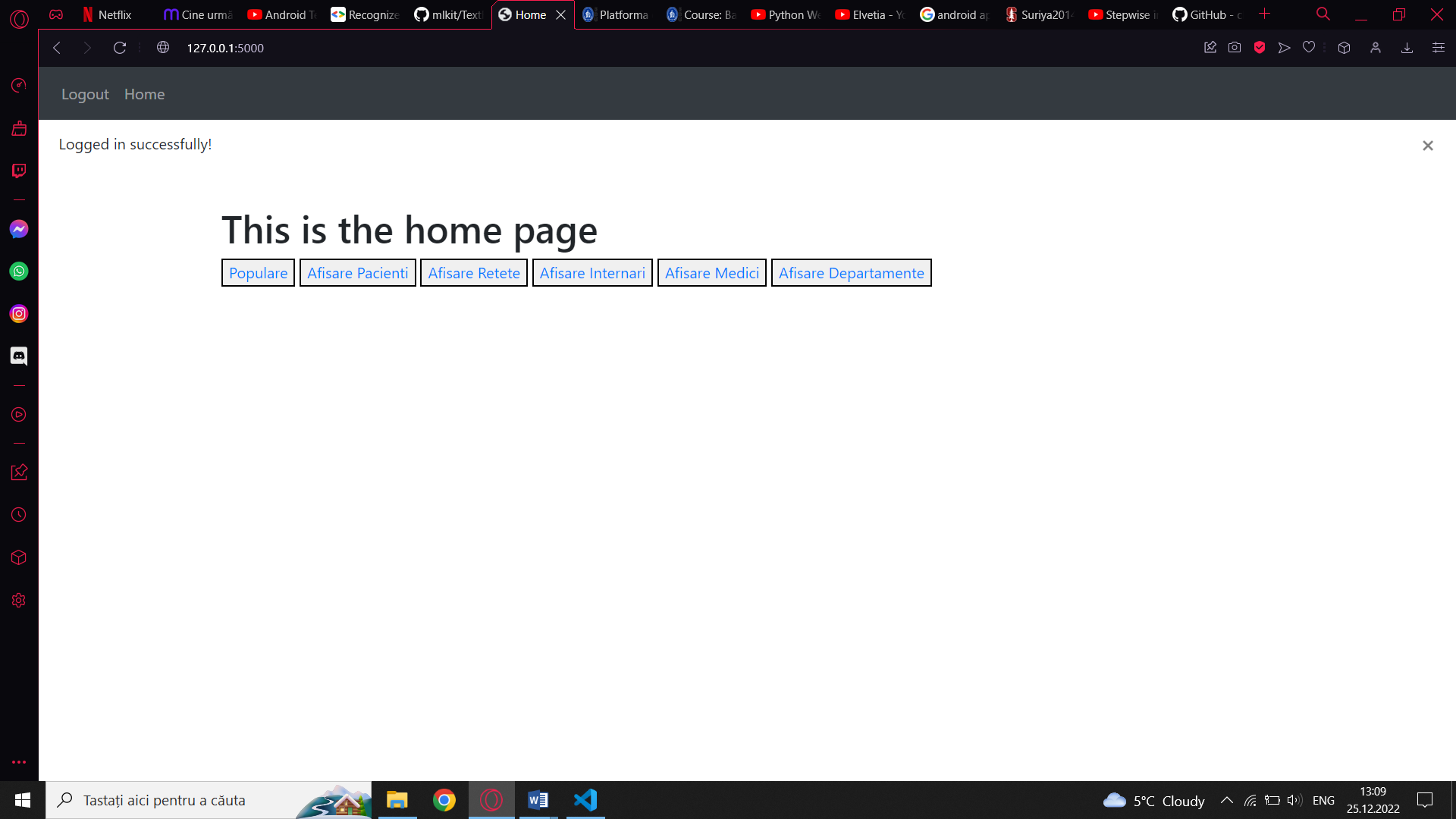
Fig 2 – Pagina Sign Up (A fost apasat submit fara a completa campul pentru Email)

Dupa introducerea corecta a datelor in pagina de Login, aplicatia redirectioneaza utilizatorul pe pagina ,,Acasa” a site-ului, afisand mesajul ,,Login successfully!”. In ceasta pagina, sunt disponibile urmatoarele actiuni:

-butonul ,,Populare” populeaza baza de date cu cateva date (se poate folosi o singura data).

-butonul ,,Afisare pacienti” afiseaza pacientii clinicii formatand informatiile din baza de date intr-un mod in care utilizatorul le poate citi usor.

-butonul ,,Afisare Retete” afiseaza datele din tabela Retete s.a.m.d



Exemplu: Dupa apasarea butonului ,,Afisare internari”, utilizatorul este redirectionat pe pagina ce afiseaza datele din tabela Internari, cat si date din celelalte tabele prin Join-uri.

Se observa ca in tabela internari avem 4 randuri.

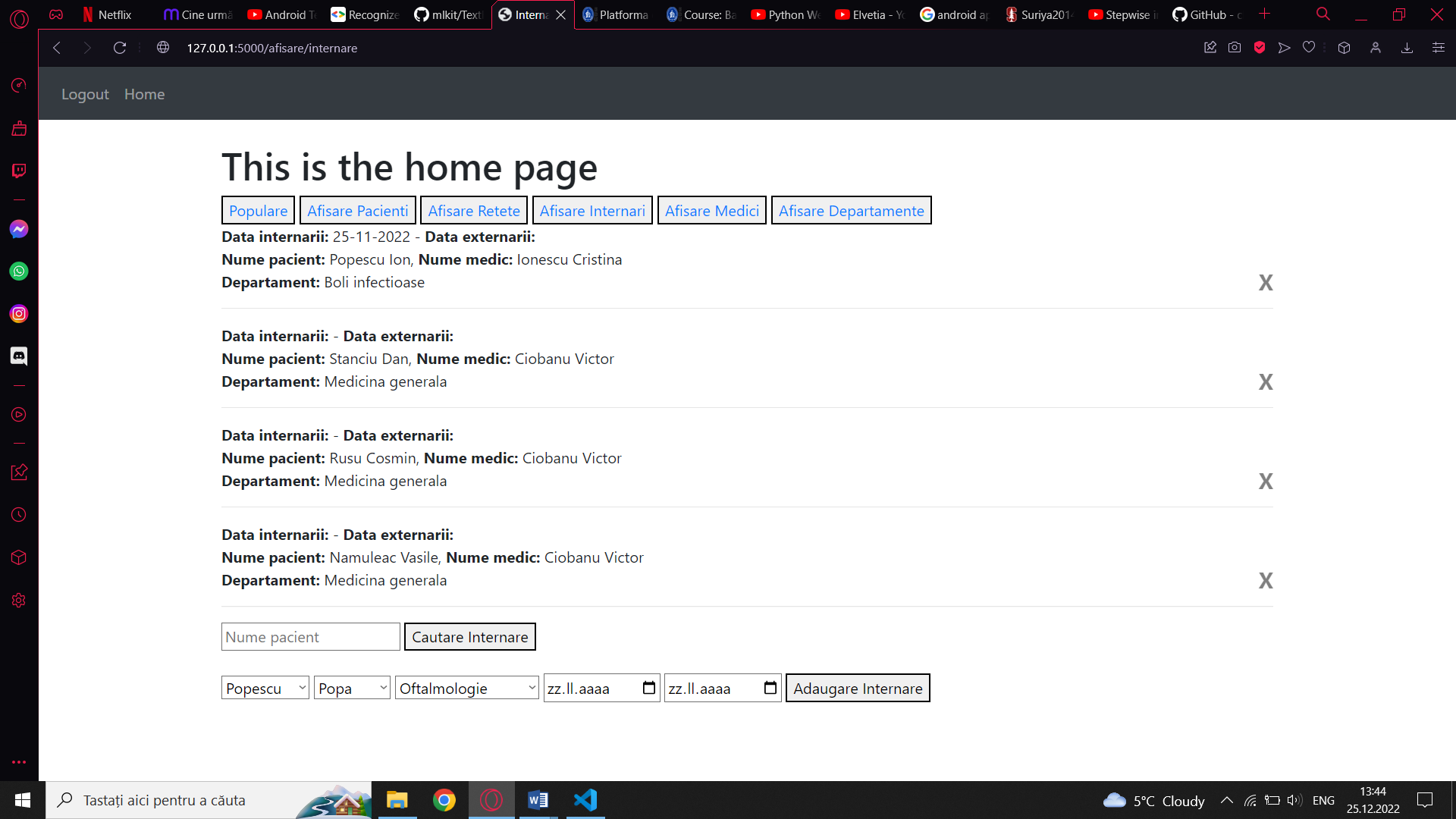
Campurile neimportante pentru utilizator nu sunt afisare (spre exemplu id-ul interogarii), in schimb sunt afisare campuri de interes din celelalte tabele precum numele si prenumele pacientului.

,,X”-ul din dreapta fiecarui rand reprezinta un buton care are ca rezultat stergerea acelui rand din baza de date. Este disponibil pentru fiecare rand din fiecare tabela.

In josul paginii, avem optiunea de a cauta o internare cu ajutorul campului de input ,,Nume pacient”, ceea ce inseamna ca, cautarea unei internari se va face cu ajutorul numelui pacientului.

De asemenea mai avem optiunea de a adauga un rand in tabela internari cu ajutorul imput-urilor din stanga butonului ,,Adaugare internare”. Cheile straine sunt alese cu ajutorul unui dropdown-menu pentru fiecare dintre acestea, acest lucru previne greselile pe care le poate inroduce utilizatorul in scriere, cat si situatia in care cheia straina introdusa de utilizator nu se gaseste in celelalte tabele.

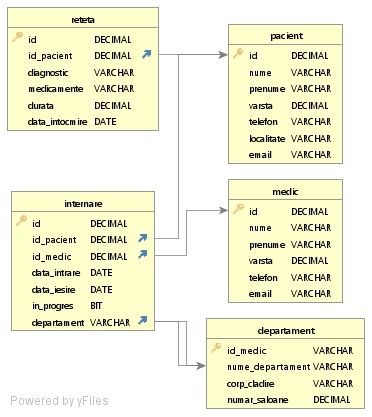
La adaugarea unui nou rand in tabela, se genereaza automat un id unic pentru aceasta (cheia primara) in partea de Back-End. De asemenea aplicatia are si un algoritm prin care, la stergerea unei interogari, urmatoarea adaugata va lua cheia primara ramasa libera, facand astfel aranjarea optima a datelor.



Operatiile mentionate anterior se vor regasi pe pagina fiecarui tabel.

Capitolul 4 – Baza de date

Baza de date a aplicatiei este alcatuita din 5 tabele: Pacienti, Retete, Internari, Medici si Departamente.



Relatiile dintre tabele pot fi observate in diagrama de mai sus, majoritatea fiind de tipul ,,one to many”. Structura bazei de date este relativ simpla pentru a facilita o accesare rapida si eficienta a datelor.

Operatiile care sunt implementate in aplicatie sunt:

* SELECT – cu ajutorul acestei comenzi, se pot vizualiza datele dintr-o tabela sau alta, este executa la apasarea butoanelor de afisare din pagina principala.
* INSERT – comanda care adauga noi randuri in baza de date, folosita la apasarea butonului de adaugare din fiecare pagina dupa completarea campurilor aferente.
* DELETE – comanda care sterge randuri din tabele, executa la apasarea butonului ,,X”
* UPDATE

Capitolul 5 – Concluzii

Obiectivul de a avea o gestiune eficienta si rapida a informatiilor esentiale dintr-o clinica medicala a fost realizat cu succes in aplicatia prezentata. Am oferit utilizatorului o interfata intuitiva, care limiteaza probabilitatea de introducere a datelor eronate aproape de 0 procente, macar din partea pe care o realizeaza algoritmul. In viitor, aplicatia poate fi extinsa usor datorita manierei in care a fost scris codul, astfel ca va putea primi mentenanta si update-uri regulat, si de asemenea permite adaugarea de noi facilitati care sa duca gestiunea datelor la un alt nivel.

REFERINTE:

* <https://www.youtube.com/watch?v=dam0GPOAvVI&t=7122s&ab_channel=TechWithTim>
* <https://www.w3schools.com/sql/>
* <https://docs.python.org/3/library/sqlite3.html>
* <https://www.youtube.com/watch?v=pd-0G0MigUA&ab_channel=CoreySchafer>