

# PROGRAMAREA INTERFETELOR PENTRU BAZE DE DATE

Profesor coordonator: **Ș. I. Dr. Ing. Pupezescu Valentin**

STUDENT: **Dinu Emilia Viviana**

GRUPA: **432A**

# CUPRINS

1. **Tema proiectului** pag. 3

# Descrierea sistemului de gestiune a bazelor de date MySQL……..pag. 3

* 1. Ce este MySQL……………………………………………………pag. 3
  2. Cum administreaza MySQL bazele de date…………………… .....pag. 3
  3. De ce este atat de util MySQL………………………………………pag. 4

1. **Tehnologia Flask utilizată în dezvoltarea aplicației**………………...pag. 4
   1. Ce este Flask………………………………………………………… pag. 4
   2. Cum funcționează Flask……………….…………………………pag.4
   3. Care sunt avantajele utilizării Flask…………………….pag. 4
2. **Limbajul HTML si utilizatea sa in aplicatii**……………………….pag. 5
   1. Ce este HTML…………………………………………………………pag. 5
   2. In ce scop utilizam limbajul HTML…………………………………pag. 5
3. **Descrierea aplicatiei**…………………………………………………pag. 5
   1. **Baza de date**………………………………………………………pag. 5
   2. **Diagrama logica a bazei de date (Diagrama ERD)**……………..pag. 7
   3. **Functionalitatea aplicatiei**………………………………………..pag. 8
      1. Arhitectura proiectului…………………………………………..pag. 8
      2. Implementarea functiilor…………………………………….. . .pag. 8
      3. Partea de front-end……………………………………………..pag. 14
4. **Concluzii**…………………………………………………………….pag. 14
5. **Bibliografie**………………………………………………………….pag. 15

# Tema proiectului

Tema proiectului se bazeaza pe dezvoltarea unei aplicatii ce contine o baza de date, creata in sistemul de gestionare a bazelor de date MySQL. Se pot utiliza diferite tehnologii: JSP, Hibernate, JPA, .NET, Python+Django, Python+Flask, etc.

Interfetele vor trebui sa permita utilizatorului sa execute urmatoarele operatii pe toate tabele: vizualizare, adaugare, modificare si stergere de date. Vizualizarea tabelelor de legatura va presupune vizualizarea datelor referite din celelalte tabele.

Pentru tema individuala primita, am ales 2 tehnologii diferite: JSP si Python+Flask. In aceasta prezentare ne vom orienta atentia catre tehnologia Flask. Asocierea pentru tabelele din baza de date este de M:N.

# Descrierea sistemului de gestiune a bazelor de date MySQL

* 1. **Ce este MySQL?**

MySQL este un sistem de gestionare a bazelor de date relationale open source care este utilizat in principal pentru aplicatiile online. MySQL poate crea și gestiona baze de date foarte utile (cum ar fi informatii despre angajati, inventar și multe altele), la fel ca alte sisteme, cum ar fi popularul Microsoft Access. In timp ce Microsoft Access, MySQL și alte sisteme de gestionare a bazelor de date servesc scopuri similare (de a gazdui datele), utilizarea difera foarte mult. [1]

MySQL este componenta integrata a platformelor [LAMP](https://ro.wikipedia.org/wiki/LAMP) sau WAMP (Linux/Windows- Apache- MySQL-PHP/Perl/Python). Popularitatea sa ca aplicatie web este strans legata de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. In multe carți de specialitate este precizat faptul ca MySQL este mult mai usor de invatat și folosit decat multe din aplicatiile de gestiune a bazelor de date, ca exemplu comanda de iesire fiind una simpla și evidenta: „exit” sau „quit”. [2]

# Cum administreaza MySQL bazele de date?

Pentru a administra bazele de date MySQL se poate folosi modul linie de comanda sau, prin descarcare de pe internet, o interfata grafica: [MySQL Administrator](https://ro.wikipedia.org/w/index.php?title=MySQL_Administrator&action=edit&redlink=1) și [MySQL Query](https://ro.wikipedia.org/w/index.php?title=MySQL_Query_Browser&action=edit&redlink=1) [Browser.](https://ro.wikipedia.org/w/index.php?title=MySQL_Query_Browser&action=edit&redlink=1) Un alt instrument de management al acestor baze de date este aplicatia gratuita, scrisa în PHP, [phpMyAdmin](https://ro.wikipedia.org/wiki/PhpMyAdmin). [2]

# De ce este atat de util MySQL?

Baza de date MySQL este folosita in principal ca mijloc de a stoca date pentru aplicatii mari, bazate pe web. Site-uri precum WordPress, iStock, GitHub, Facebook, NASA, Marina SUA, Tesla, Scholastic, Spotify, YouTube, Netflix, Glasses Direct, Symantec (si multe altele) folosesc baza de date MySQL ca mijloc de stocare a datelor pe din interiorul sau exteriorul site-urilor web și serviciilor interne. [1]

# Tehnologia Flask utilizată în dezvoltarea aplicației

* 1. **Ce este Flask?**

Flask este un micro-framework web pentru Python, care permite dezvoltarea rapidă a aplicațiilor web. Este minimalist și ușor de utilizat, oferind funcționalități esențiale, cum ar fi gestionarea rutelor HTTP, managementul sesiunilor, și integrarea cu bazele de date.

Flask nu impune o structură rigidă, permițând dezvoltatorilor să aleagă cum să își organizeze aplicația. Acesta se bazează pe biblioteci externe pentru funcționalități suplimentare, cum ar fi baze de date, validare de formulare și autentificare.[3]

# Cum functioneaza Flask?

Flask utilizează un model bazat pe rute HTTP. Fiecare rută corespunde unui URL și unui handler, care poate fi o funcție Python ce primește cererea HTTP și trimite un răspuns. De asemenea, Flask permite interacțiunea cu diverse baze de date, utilizând SQLAlchemy sau alte librării pentru manipularea datelor.

Principalele componente ale unei aplicații Flask sunt:

* **Rute**: pentru a asocia URL-urile cu funcțiile.
* **Template-uri**: pentru a genera interfețe dinamice (folosind Jinja2).
* **Formulare**: pentru colectarea datelor de la utilizatori.
* **Baze de date**: pentru gestionarea persistentei datelor.[3]

# Avantaje Flask?

Flask este alegerea ideală pentru proiecte mici și medii, datorită ușurinței sale de utilizare și flexibilității. Printre avantajele sale se numără:

* **Simplicitate și claritate**: Flask permite o învățare rapidă și dezvoltare rapidă.
* **Flexibilitate**: nu impune o structură strictă, lăsând dezvoltatorul să aleagă tool-urile necesare.
* **Extensibilitate**: Flask suportă multe extensii pentru a adăuga funcționalități suplimentare.
* **Documentație bine pusă la punct**: Flask are o documentație cuprinzătoare și exemple.[3]

# Limbajul HTML si utilitatea sa in aplicatii

* 1. **Ce este HTML?**

Unul din primele elemente fundamentale ale WWW ( World Wide Web ) este HTML ( Hypertext Markup Language ), care descrie formatul primar în care documentele sunt distribuite și văzute pe Web. Multe din trasaturile lui, cum ar fi independenta fata de platforma, structurarea formatării și legaturile hypertext, fac din el un foarte bun format pentru documentele Internet și Web.[4]

# In ce scop utilizam limbajul HTML?

Scopul HTML este mai degraba prezentarea informatiilor – paragrafe, fonturi, tabele ș.a.m.d. – decat descrierea semanticii documentului. In cadrul [dezvoltarii web](https://ro.wikipedia.org/wiki/Dezvoltare_web) de

tip [front-end,](https://ro.wikipedia.org/wiki/Front-end_%C8%99i_back-end) HTML este utilizat impreuna cu [CSS](https://ro.wikipedia.org/wiki/CSS) și [JavaScript.](https://ro.wikipedia.org/wiki/JavaScript)[2]

# Descrierea aplicatiei

* 1. **Baza de date**

Tema individuala se bazeaza pe crearea unei baze de date ce are 2 tabele in asociere M:N. Tabelele sunt:

**elevi** si **subiecte**. Pentru cele 2 tabele am ales cateva atribute caracteristice:

Pentru tabela **elevi**: am ales ca si cheie primara idelev. Celelalte atribute sunt: nume, prenume, clasa.

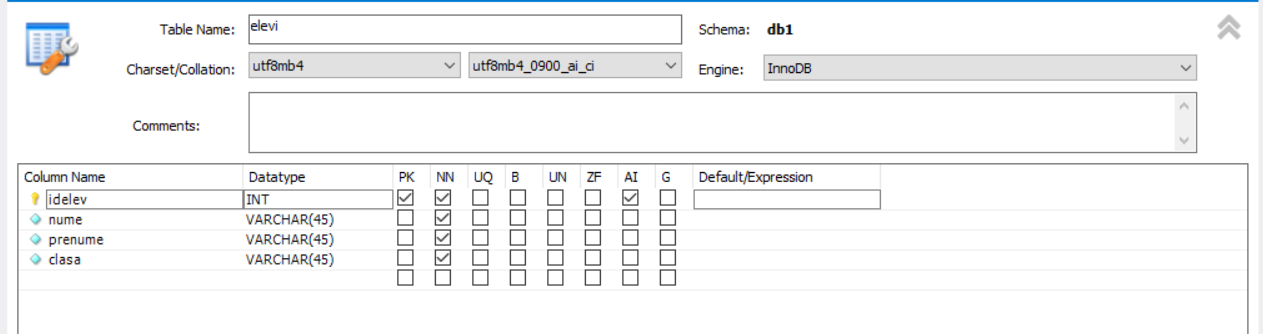


Figura 1.

Pentru tabela **subiecte**: am ales ca si cheie primara idsubiect. Celelalte atribute sunt:materie, nota.

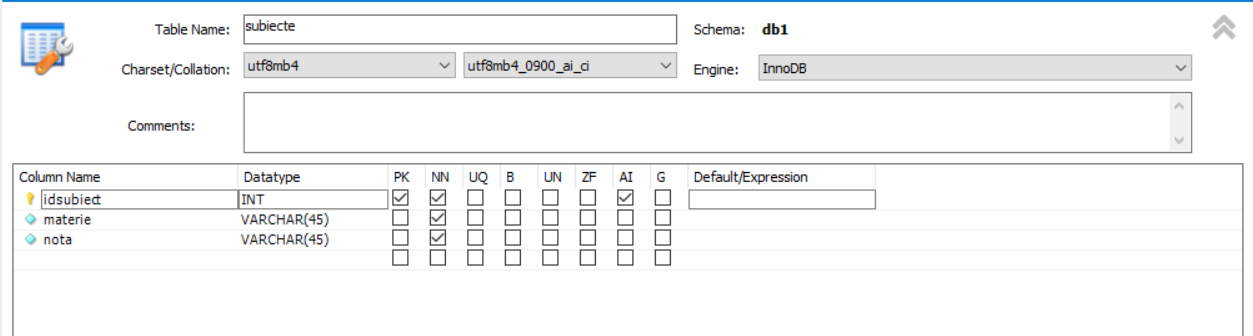


Figura 2.

# Ce este asocierea M:N?

Asocierea M:N (mai-multi-la-mai-multi) are ca si caracteristica faptul ca fiecarui element inregistrat intr-o tabela ii pot fi asociate mai multe elemente din cealalta tabela si invers. [5]

De exemplu in cazul nostru, un album poate fi asociat mai multor genuri muzicale, asa cum si unui gen muzical ii pot fi asociate mai multe albume.

# Ce reprezinta tabela de legatura?

Pentru a crea o relație mai-multi-la-mai-multi, trebuie sa se creeze o a treia tabela denumita deseori tabela de joctiune, care imparte relatia mai-multi-la-mai-multi în doua relatii unu-la-mai-multi. In cazul nostru, am ales ca si tabela de legatura tabela **profesori**.

In aceasta noua tabela, atributele ce au fost selectate ca si chei primare pentru tabelele anterioare vor deveni chei straine (FK) pentru tabela de legatura **profesori**.

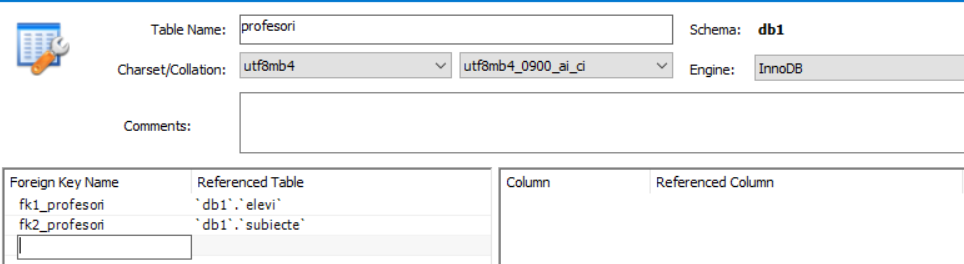


Figura 3.

Pentru aceasta tabela profesori cheile straine idelev si idsubiect vor constitui o cheie primara In rest am adaugat si: idprofesor, nume, prenume, materie, contact.

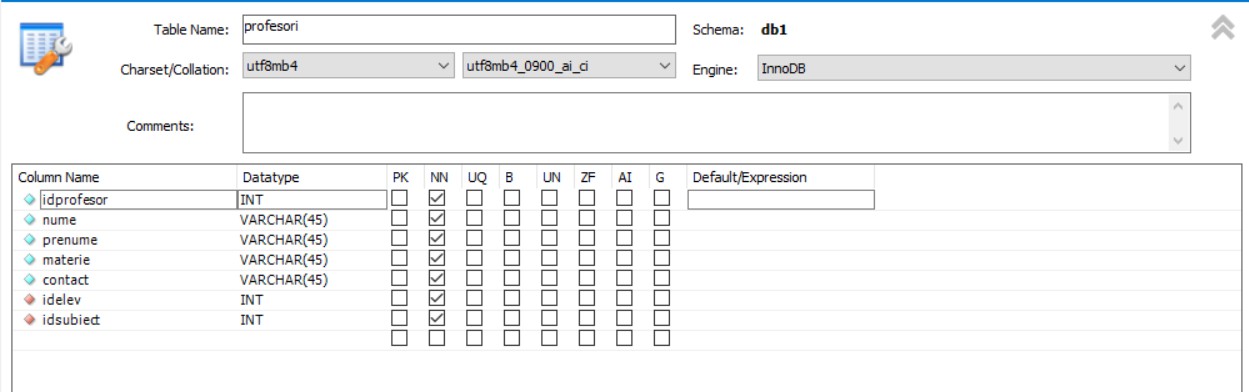


Figura 4.

# Diagrama logica a bazei de date (Diagrama ERD)

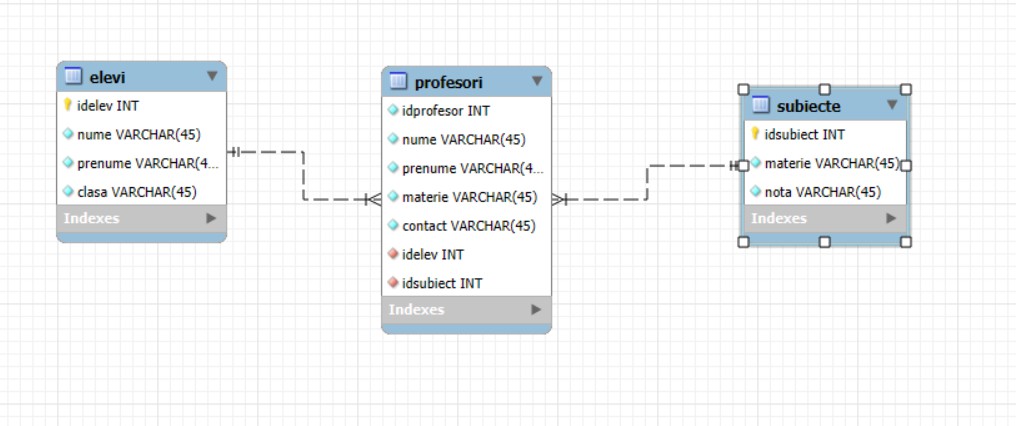
****

Figura 5.

**De ce este importanta diagrama logica a bazei de date?**

Aceasta ofera o reprezentare logica detaliata a datelor celor 3 tabele, a relatiilor dintre ele.

# Ce relatii exista intre cele trei tabele?

 Intre **elevi** si **profesori** exista o asociere de tip 1:N

 Intre **elevi** si **subiecte** exista o asociere de tip M:N

 Intre **subiecte** si **profesori** exista o asociere de tip 1:N.

# Functionalitatea aplicatiei

* + 1. **Arhitectura proiectului**

Proiectul realizat in tehnologia JSP are urmatoarea structura:

* + - * Un pachet DB ce continue clasa JavaBean, ce are rolul de a oferi toate functionalitatile principale ale interfetei bazei de date si anume: conectica, operatiile de afisare, adaugare, modificare, stergere.
      * Folder-ul webapp ce continue toate paginile JSP, ce au rolul de a importa functionalitatile din clasa JavaBean, fiind conectate intre ele si implicit conectate la pachetul DB, ce realizeaza partea dinamica a proiectului.

# Implementarea functiilor Cum se realizeaza conexiunea?

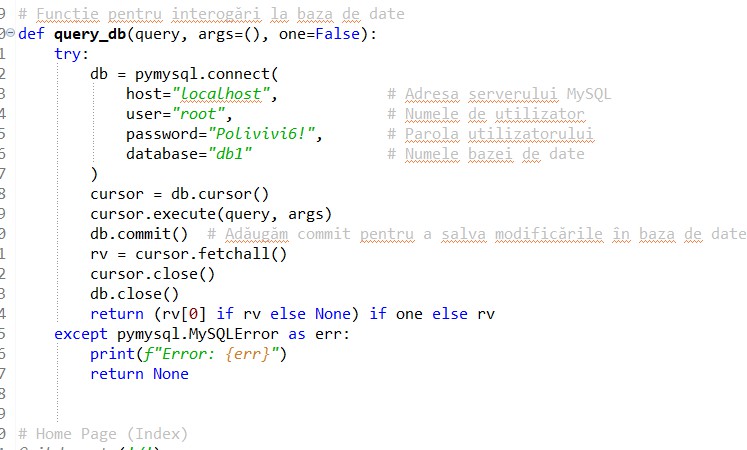
****

Figura 6.

Conexiuanea se face prin instructiunea db = pymysql.connect... unde vom nota informatiile de conectare ale bazei noastre de date.

Totodata close() pe obiectul query\_db() ce opreste conexiunea la baza de date.

# Cum se afiseaza datele?

In primul rand alegem ca si exemplu tabela Elevi. Obiectul din clasa app.py este tabela\_elevi() care returneaza un set de rezultate: linii de tip ‘elevi’

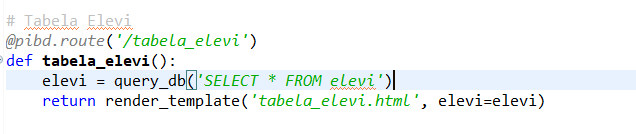


Figura 7.

Vom obtine prin apasarea butonului **Tabela Elevi** din cadrul index-ului urmatoarea pagina:

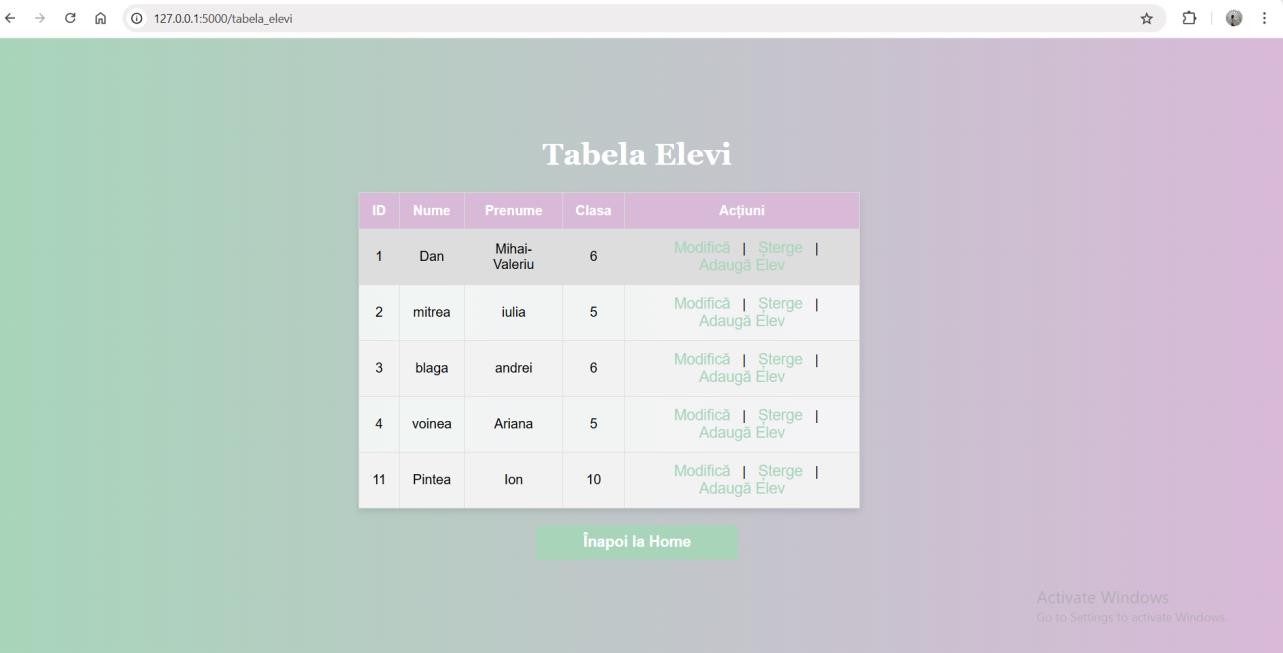


Figura 8.

# Cum se realizeaza stergerea datelor?

Luam drept exemplu stergerea datelor unui elev. Asadar ne indreptam atentia catre fisierul sterge\_elev.html.

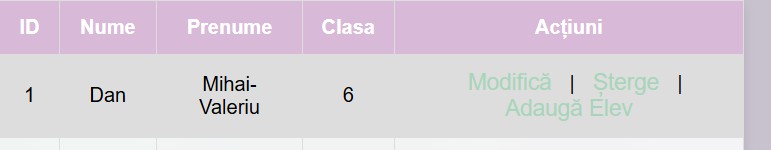


Figura 9.

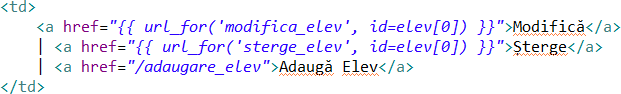
Daca alegem sa apasam butonul “Sterge elev” din cadrul paginii ce contine datele elevilor inregistrati, se va face legatura cu fisierul sterge\_elev.html.

Figura 10.

Se memoreaza id-ul elevului pe care vreau sa il sterg si se plaseaza intr-un query DELETE din functia sterge\_elev(idelev) din app.py:

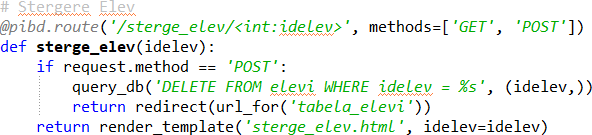


Figura 11.

La finalul operatiei, se va afisa urmatoarea pagina, unde confirmi stergerea si te redirectioneaza pe pagina de tabela elevi:

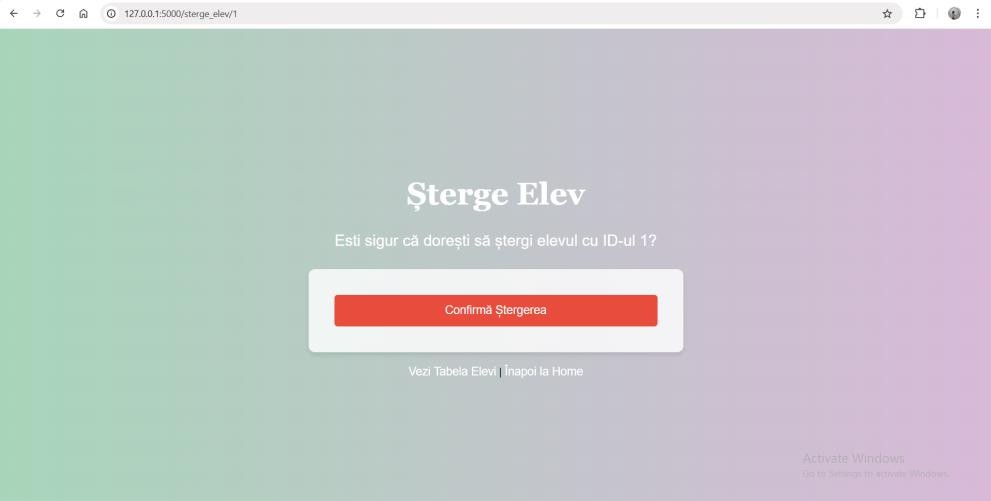


Figura 12.

# Cum se realizeaza adaugarea datelor?

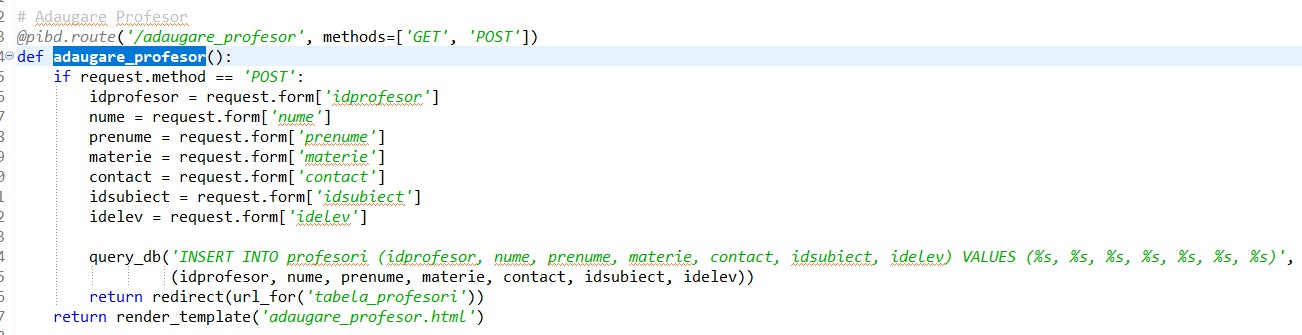
Pentru aceasta operatie vom lua ca exemplu tabela de legatura profesori. Deci ne concentram atentia asupra fisierului adaugare\_profesor();

Figura 13.

Se declara initial elementele pentru campurile pe care dorim sa le completam. Am realizat niste casete de tip INPUT pentru introducerea datelor ce se doresc adaugate in baza de date. Acestea vor fi adaugate numai daca in campurile vor fi in totalitate completate, in caz contrar, nu se va adauga nicio data.

Pagina in cadrul careia se adauga datele este:

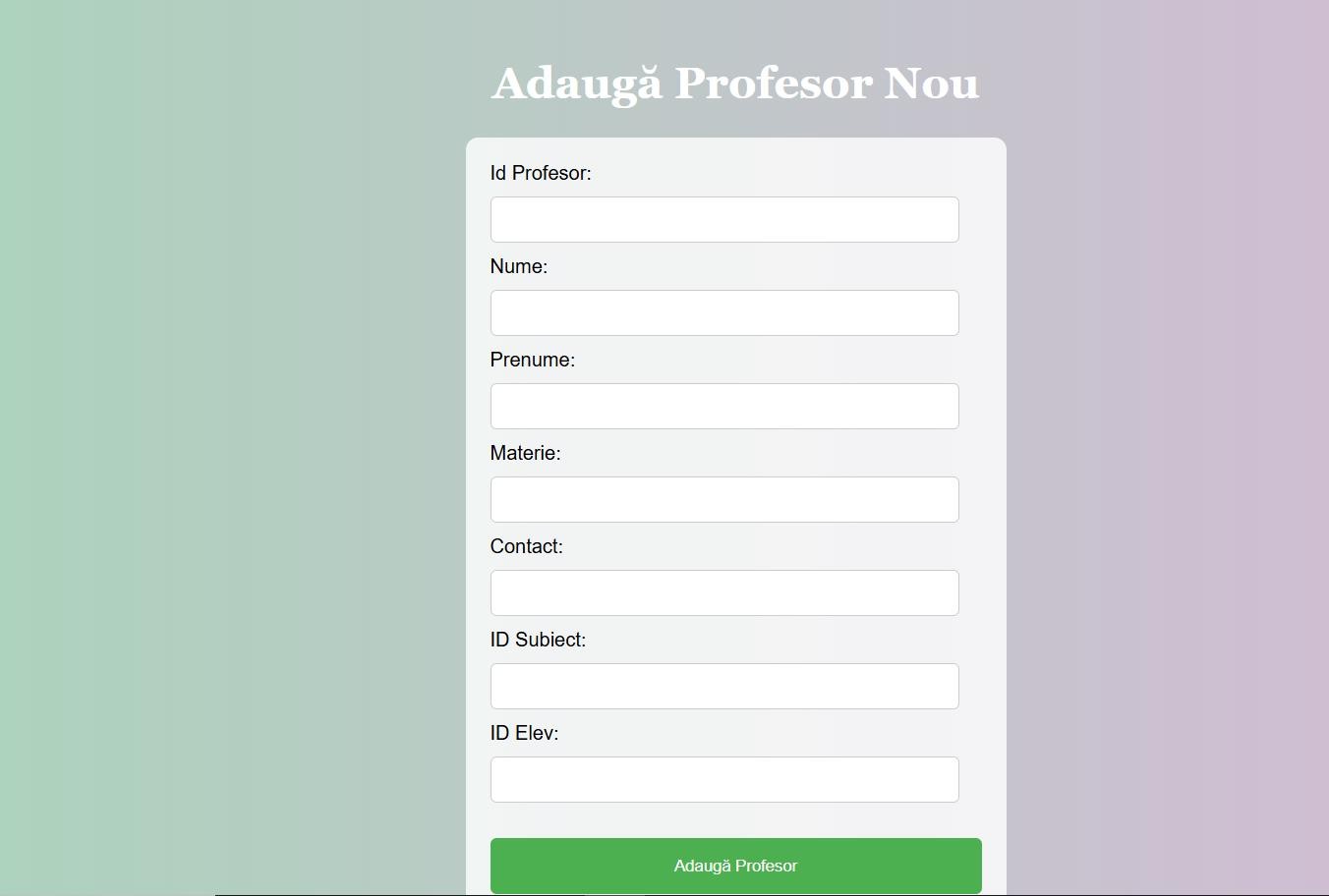
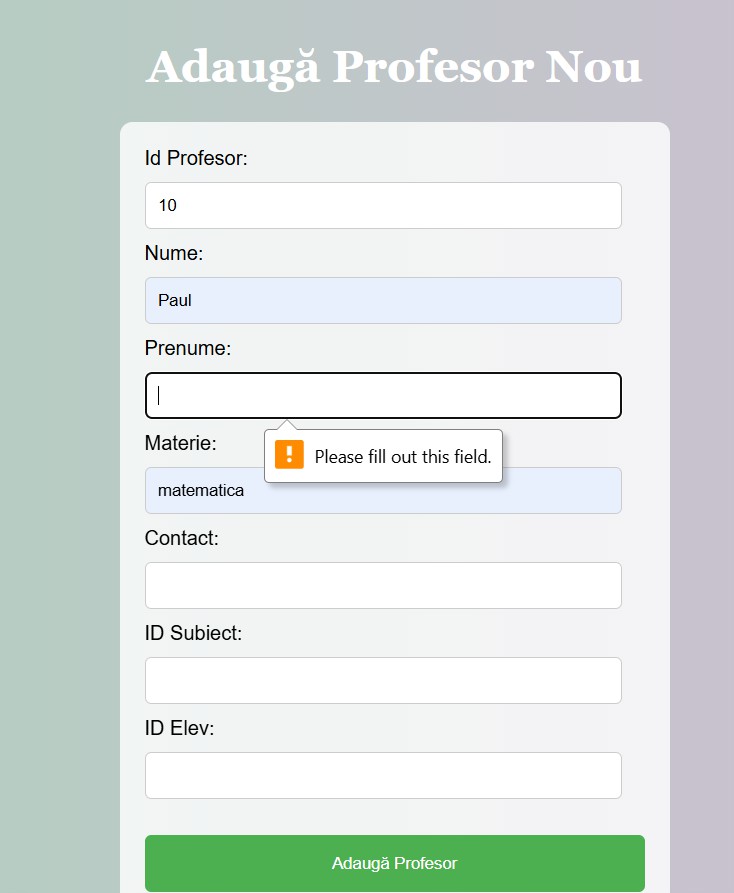


Figura 14

Odata apasat butonul “Adauga Profesor”, se va face legatura cu functia adauga\_profesor() din app.py.

In cazul in care un camp este lasat necompletat, atunci datele nu vor fi adaugate si va aparea urmatorul mesaj:

# Cum se realizeaza modificarea datelor?

Din fiecare tabela principala am ales sa am optiunea sa pot modifica anumite date. Asadar vom lua exemplu tabela elevi, deci ne vom indrepta atentia catre tabela elevi, unde am introdus, de exemplu, butonul pentru operatia de modificare:

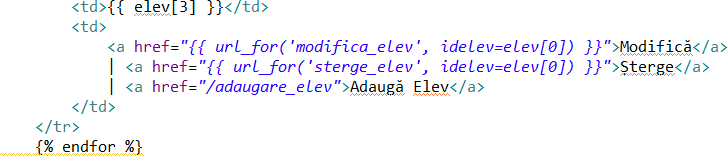


Figura 16.

Apasand butonul vom fi redirectionati catre fisierul modifica\_elev.html

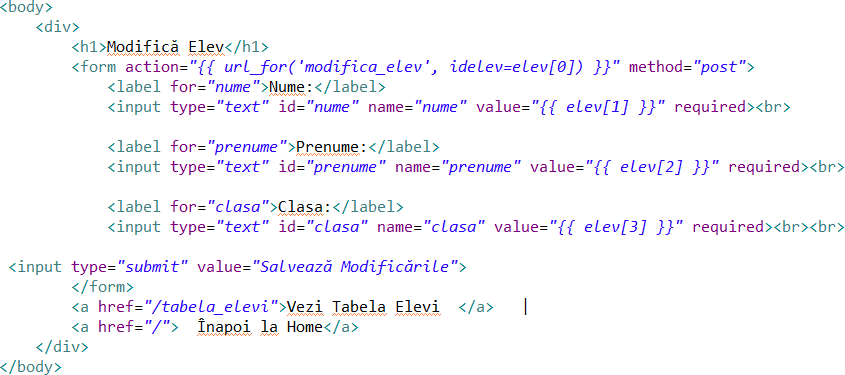


Figura 17.

In fisierul modifica\_elev.html, am creat campuri de tip INPUT pentru introducerea noilor date.

# Partea de front-end

Pentru un aspect mai placut al interfetei, am folosit limbajul HTML combinat cu CSS.

De exemplu am ales sa creez un aspect mai placut in ceea ce priveste butoanele, divs si background folosind instructiuni in blocul STYLE:



Figura 18.

Principalul rol al limbajului HTML folosit in cadrul acestei aplicatii este de a plasa componentele specifice design-ului paginii web la anumite coordonate si totodata pentru a seta dimensiunile corespunzatoare pentru div-uri ce compun pagina web.

# Concluzii

In prezent bazele de date sunt utilizate practic peste tot, fiecare companie, intreprindere mai ales cu caracter de producere, comercializare are nevoie si la sigur implimenteaza in sistemul lor o baza de date. Pentru o utilizare fiabila și corecta a unei baze de date este important sa se urmareasca realizarea unei arhitecturi ce ofera posibilitatea de a separa functionalitatile, si anume partea de back-end de partea de front- end.

Totodata, aplicatia realizata in tehnologia Flask si prezentata in cadrul acestui proiect, numita “pibdv2”, permite utilizatorului sa efectueze operatiile cerute pe baza de date creata in MySQL: afisare, adaugare, modificare si stergere.

# Bibliografie

* <https://www.nav.ro/blog/ce-este-mysql/> [1]
* <https://en.wikipedia.org/wiki/SQL> [2]
* Cursurile PIBD [5]
* <https://web.ceiti.md/lesson.php?id=1> [4]
* <https://docs.python.org/3/tutorial/> [3]