

# 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………………………………………………………
  2. Cum administreaza MySQL bazele de date………………………….
  3. De ce este atat de util MySQL………………………………………..

1. **Tehnologia JSP utilizata in dezvoltarea aplicatiei**………………...pag. 4
   1. Ce este JSP……………………………………………………………
   2. Ce este SERVLET……………….…………………………………….
   3. Care este diferenta majora dintre JSP si SERVLET…………………….
2. **Limbajul HTML si utilizatea sa in aplicatii**……………………….pag. 5
   1. Ce este HTML…………………………………………………………
   2. In ce scop utilizam limbajul HTML……………………………………
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. 15
4. **Concluzii**…………………………………………………………….pag. 15
5. **Bibliografie**………………………………………………………….pag. 16

# 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 Hibernate. In aceasta prezentare ne vom orienta atentia catre tehnologia JSP. 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 JSP utilizata in dezvoltarea aplicatiei

* 1. **Ce este JSP?**

Java Server Pages este o simpla dar puternica tehnologie folosita pe partea de server pentru a genera continut HTML dinamic. JSP este o extensie directa a Java Servlets si furnizeaza o modalitate de a separa partea de procesare de cea de prezentare. Motorul JSP este doar un alt Servlet, mapat la extensia \*.jsp. [5]

Paginile JSP sunt create sa suporte mai multe tipuri de documente structurate, indeosebi HTML si XML. În general, JSP-urile folosesc anumite informatii pe care le trimit la server intr-o cerere HTTP care interactioneaza cu datele existente pe acesta si creaza dinamic un raspuns organizat intr-un format standard (HTML, DHTML, XML, etc.) sau intr-un format text sau neorganizat ce va fi trimis înapoi clientului. [2]

# Ce este SERVLET?

Un servlet este un fișier java care poate solicita clientul, proceseaza-l și furnizează un fisier HTML ca raspuns. Exista mai multe servlete in interiorul containerului web. De asemenea, este posibila maparea mai multor cereri la un servlet. Prin urmare, toate aceste configuratii sunt incluse in acest fisier special, care este fisierul web.xml.[5]

# Care este diferenta majora intre JSP si SERVLET?

Principala diferenta intre JSP și Servlet este ca JSP este un limbaj de scripting al paginii web care poate genera continut web dinamic în timp ce servletul este un program Java care este deja compilat și folosit pentru a crea conținut dinamic de web.

Intr-o aplicatie web obisnuita, clientul solicita o pagina Web de la server și serverul raspunde inapoi cu pagina necesara. Aceste pagini pot fi fie statice, fie dinamice. Continutul unei pagini statice este deja creat. Continutul unei pagini dinamice este creat in timpul rularii. JSP și Servlet sunt două metode în Java pentru a crea pagini web dinamice. JSP este tradus si compilat intr-un servlet de catre containerul web. Pe de alta parte, un servlet este un program Java care este gestionat de containerul web.[5]

# 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.

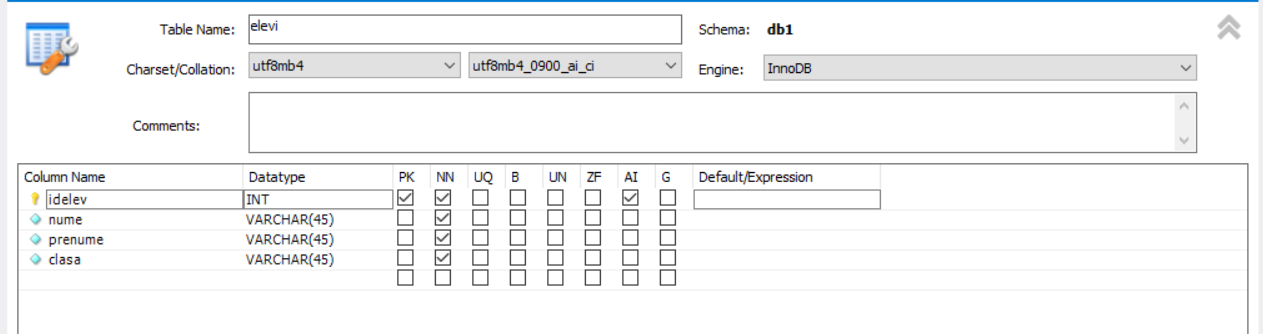


Figura 1.

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

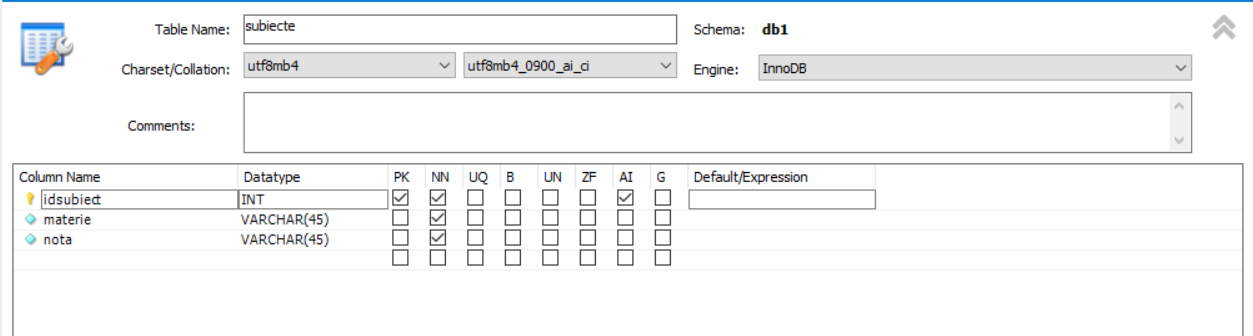


Figura 2.

# 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. [3]

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**.

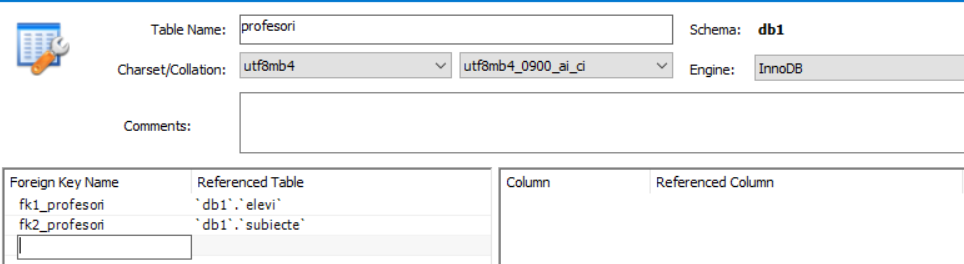


Figura 3.

Pentru aceasta tabela atributelor sunt cheile straine idelev si idsubiect si: idprofesor, nume, prenume, materie, contact.

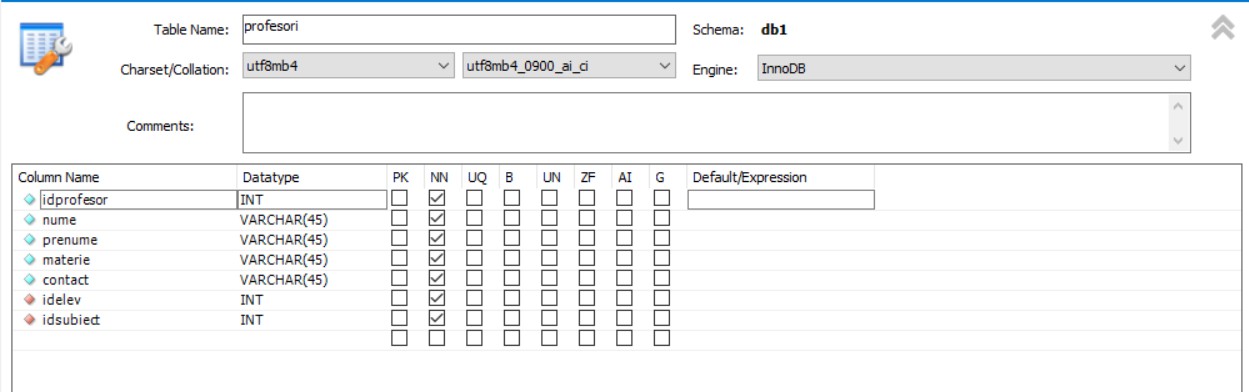


Figura 4.

# Diagrama logica a bazei de date (Diagrama ERD)

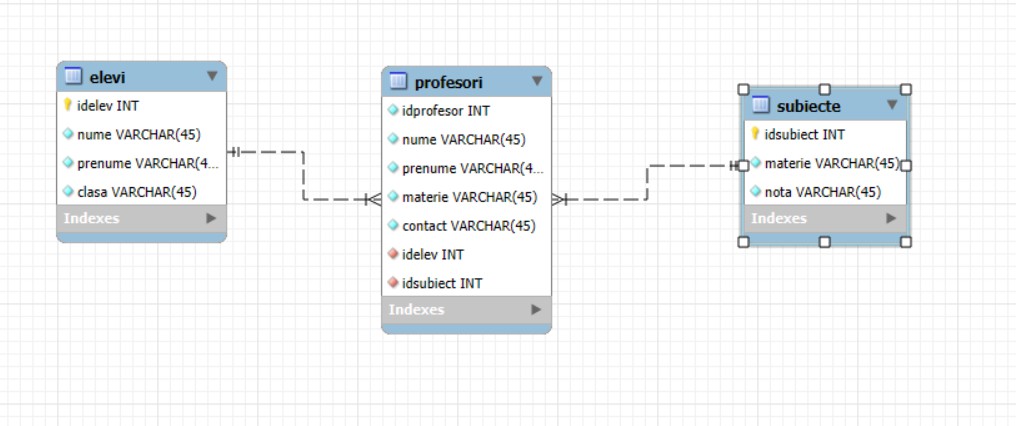
****

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?

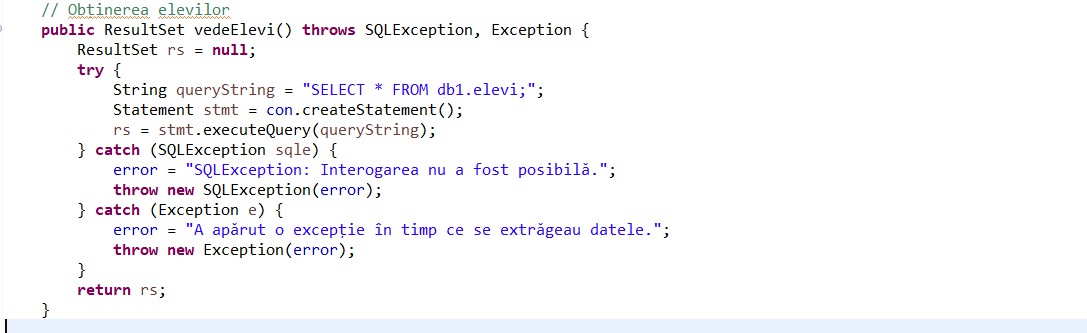
****

Figura 6.

Obiectul ResultSet va stoca rezultatul interogării SQL. Este inițializat la null, iar după executarea interogării va conține toate rândurile returnate. Avem un SELECT pentru a prelua toate coloanele din tabela elevi. Obiectul Statement este utilizat pentru a executa interogări SQL pe conexiunea deschisă la baza de date (con). Metoda executeQuery() execută interogarea și returnează un obiect ResultSet, care conține rezultatele.

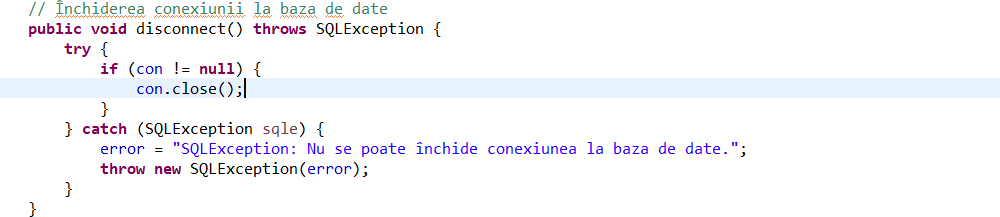
Totodata, exista si o functie disconnect() ce realizeaza close() pe obiectul de conexiune con:

Figura 7.

# Cum se afiseaza datele?

In primul rand alegem ca si exemplu tabela Elevi. Functia din clasa JavaBean este vedeElevi() care returneaza un set de rezultate: linii de tip ‘elevi’, ce le implementeaza intr-un obiect de tip results. Acesta este declarat initial null.

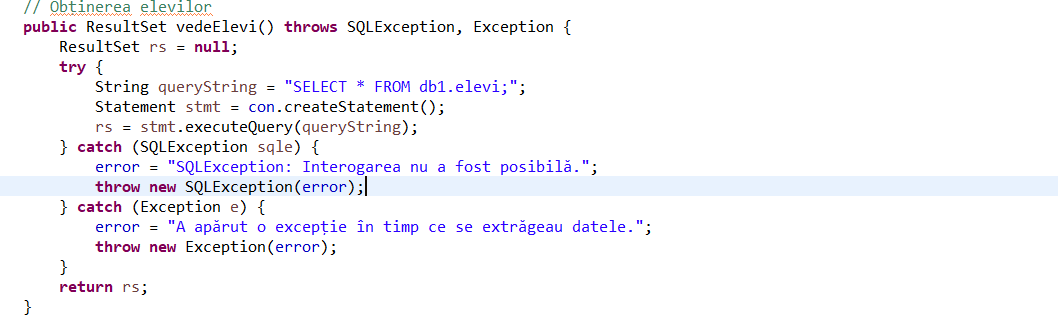


Figura 8.

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

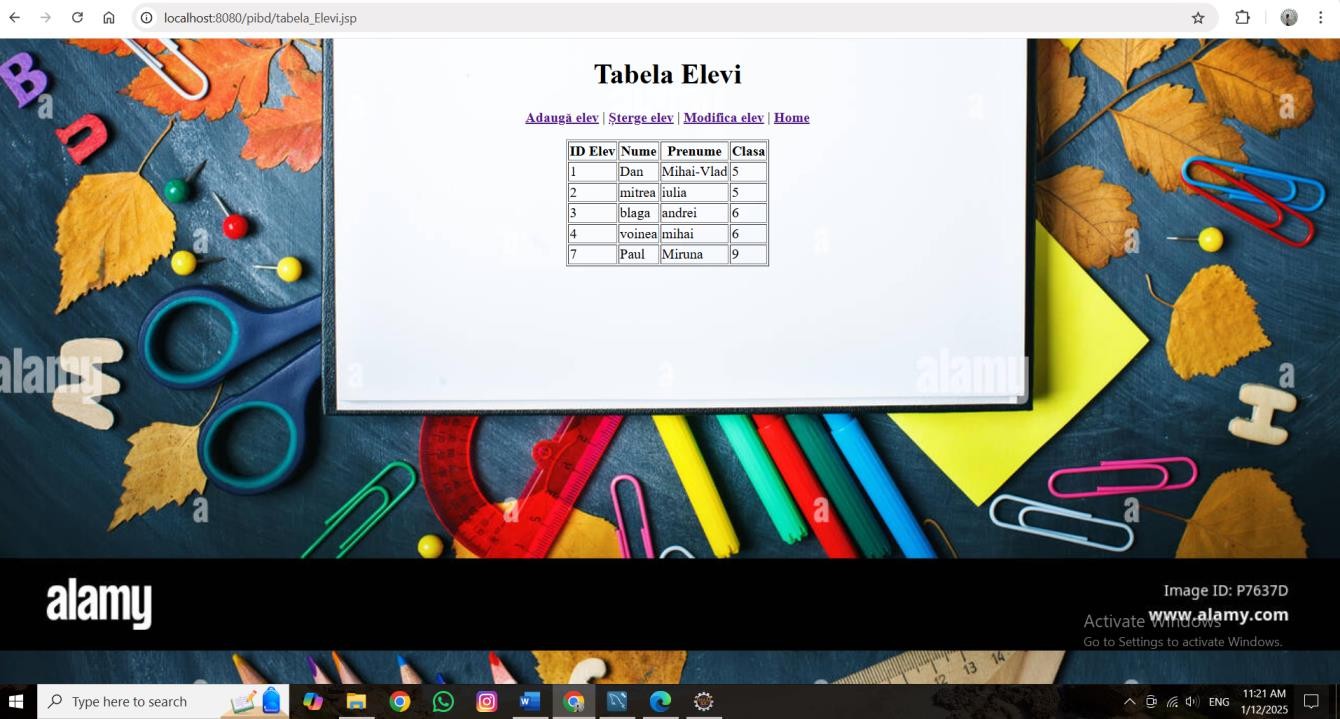


Figura 9.

# Cum se realizeaza stergerea datelor?

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

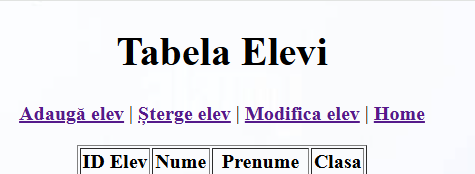


Figura 10.

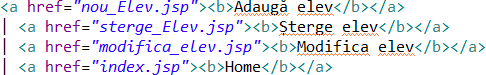
Daca alegem sa apasam butonul “Sterge elev” din cadrul paginii ce contine datele elevilor inregistrati, se va face legatura cu fisierul sterge\_Elev.jsp.

Figura 11.

Se memoreaza id-ul elevului pe care vreau sa il sterg si se plaseaza intr un query DELETE din functia stergeElev(idelev) din JavaBean:

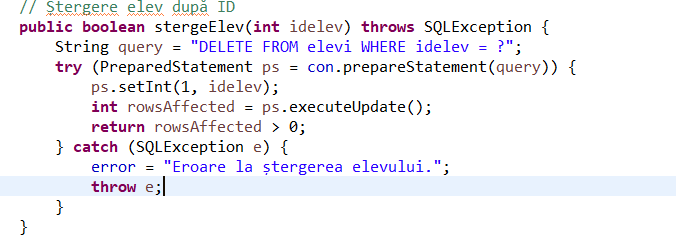


Figura 12.

La finalul operatiei, se va afisa urmatoarea pagina:

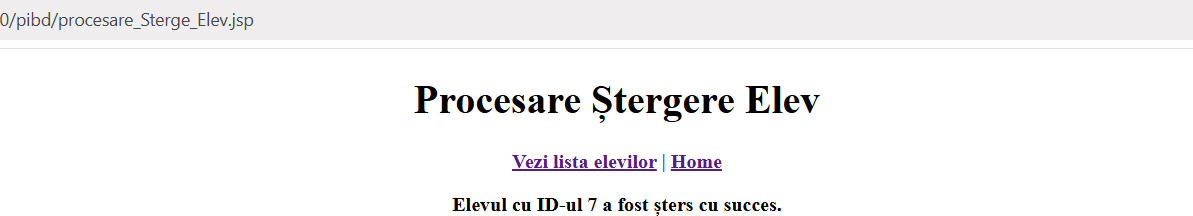


Figura 13

# Cum se realizeaza adaugarea datelor?

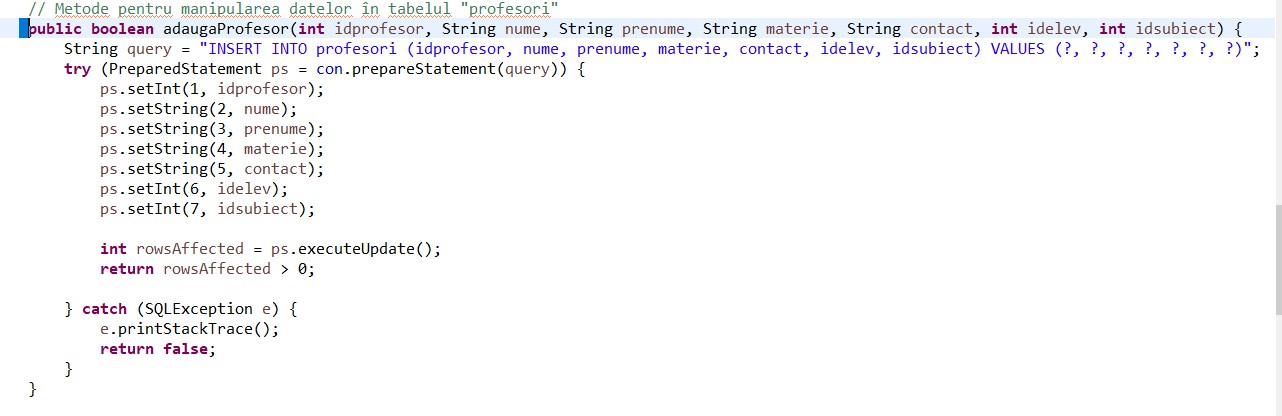
Pentru aceasta operatie vom lua ca exemplu tabela de legatura profesori. Deci ne concentram atentia asupra fisierului adaugareProfesor();

Figura 14

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:

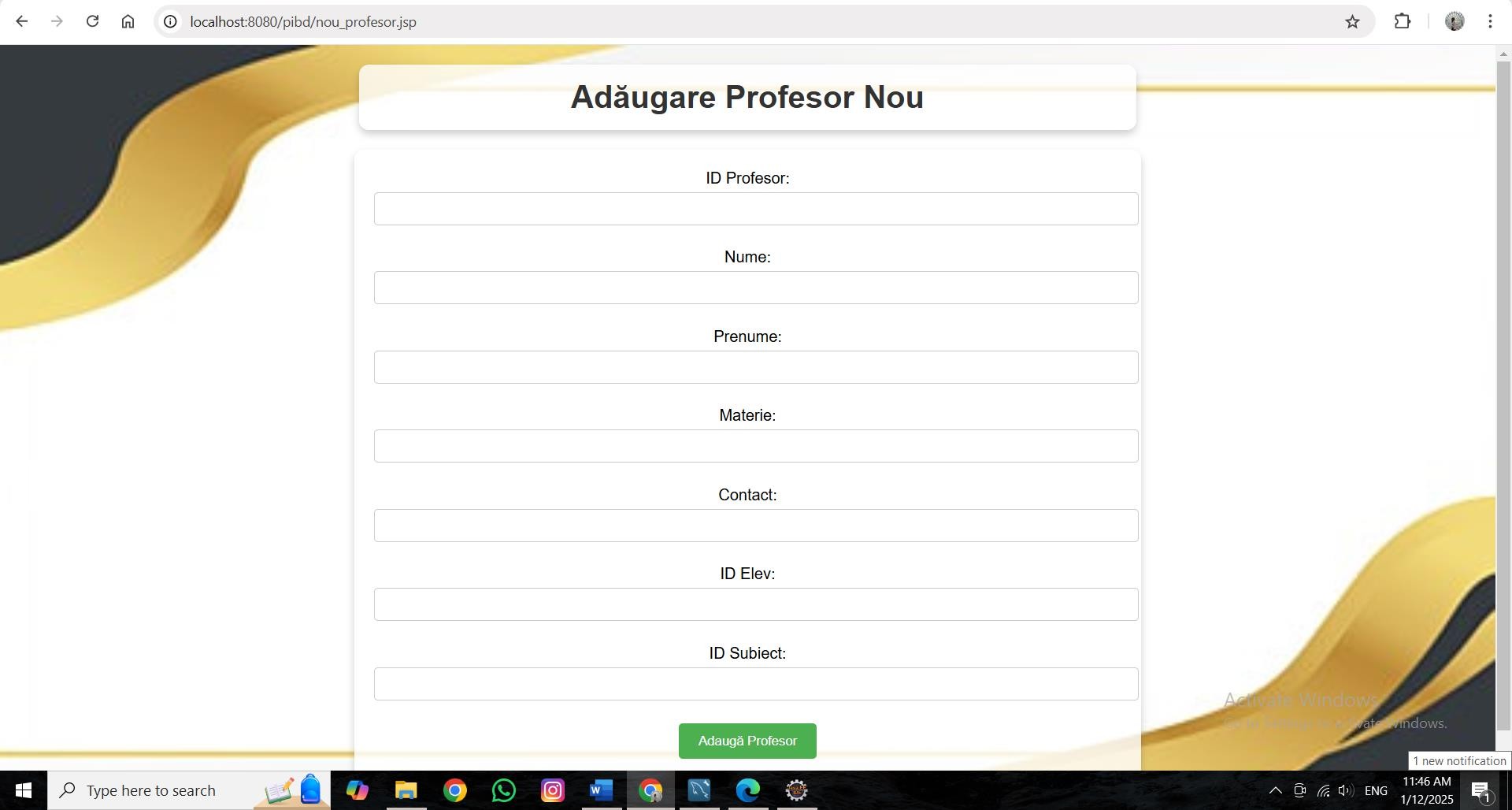


Figura 15

O data apasat butonul “Adauga Profesor”, se va face legatura cu functia adaugaProfesor(…) din clasa JavaBean. In cazul in care un camp este lasat necompletat, atunci datele nu vor fi adaugate si va aparea urmatorul mesaj:

Figura 16



# 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:

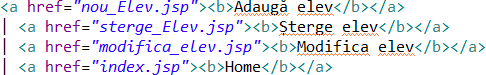


Figura 17

Apasand butonul vom fi redirectionati catre fisierul JSP modifica\_elev.jsp



Figura 18



Figura 19

In fisierul modifica\_elev, am creat campuri de tip INPUT pentru introducerea noilor date. Acestea au fost create in cadrul formularului ce este legat de fisierul procesare\_Modifica\_Elev.jsp, unde se realizeaza apelul functiei modificaElev().

# 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:

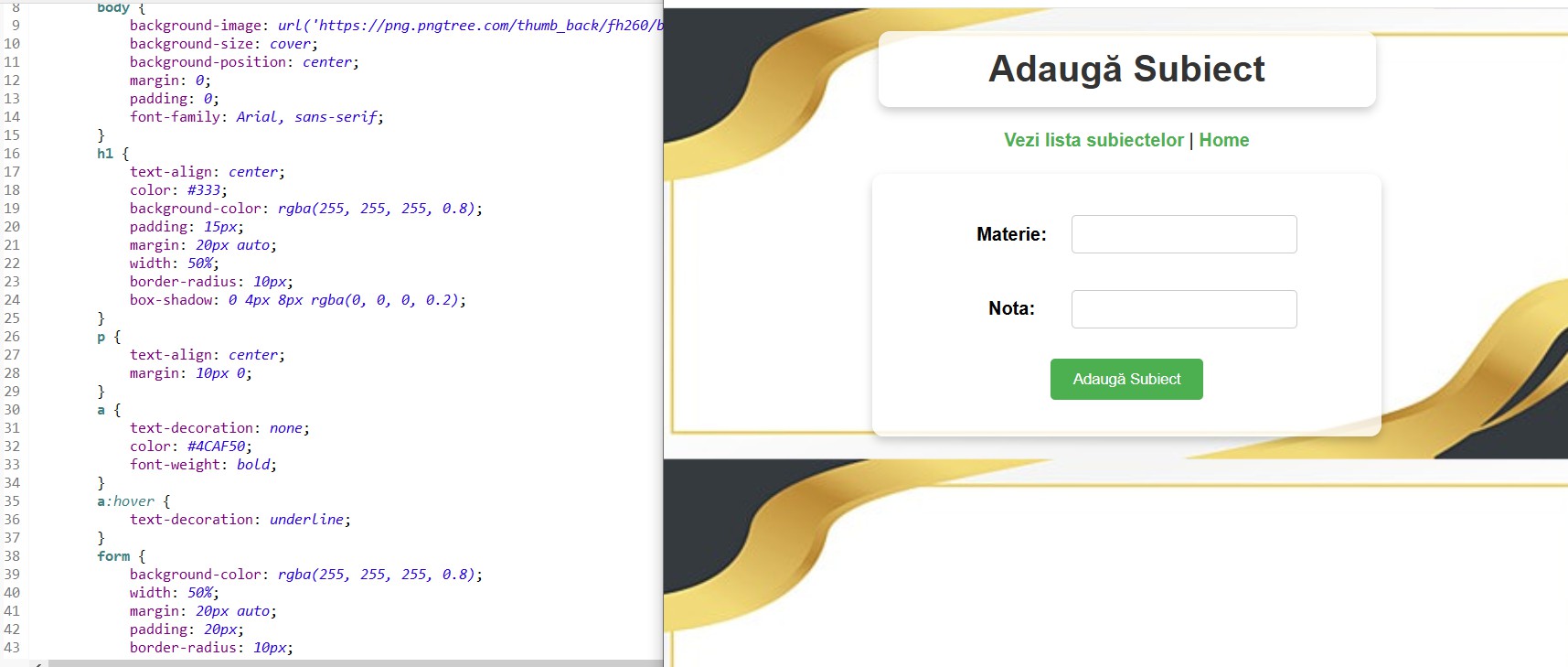


Figura 20

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 JSP si prezentata in cadrul acestui proiect, numita “pibd”, 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 [3]
* <https://web.ceiti.md/lesson.php?id=1> [4]
* <https://ro.sawakinome.com/> [5]