**Universitatea Națională de Știință și Tehnologie “Politehnica” București**

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

**Proiectarea Interfețelor pentru Baze de Date**

Profesor coordonator: Ș. l. dr. ing. Valentin Pupezescu

Student: Sava Rareș Andrei

Grupa 431A

1. **Tema proiectului**

Tema proiectului constă în dezvoltarea a două aplicaţii care să conţină o bază de date creată în sistemul de gestiune a bazelor de date MySql şi două interfeţe la aceasta, baza de date fiind comună. La crearea interfeţelor se vor folosi două tehnologii la alegere dintre JSP, Hibernate, JPA, .NET, Python, JavaFX sau Spring Boot. Interfeţele vor trebui să permită utilizatorului să execute următoarele operaţii pe toate tabele, inclusiv pe cele de legătură în cazul asocierilor de tip M:N: vizualizare, adăugare, modificare şi ştergere de date. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celelalte tabele.

Pentru tema individuală primită, am ales 2 tehnologii diferite: JSP si Hibernate. În această prezentare ne vom orienta atenția către tehnologia JSP.

Tema individuală impune includerea a două tabele principale, **studenti** și **cursuri.**  Întrucât asocierea pentru acestea este de M:N (many-to-many), am apelat la utilizarea unei tabele intermediare pe care am numit-o **proiect**, realizând astfel o aplicație de gestiune și de ținerea evidenței a proiectelor semestriale obligatorii pe care studenții trebuie să le realizeze într-o facultate.

1. **Descrierea sistemului de gestiune a bazelor de date MySQL**
   1. **Ce este MySQL?**

MySQL este un sistem de gestionare a bazelor de date relaționale open source care este

utilizat în principal pentru aplicațiile online. MySQL poate crea și gestiona baze de date foarte utile (cum ar fi informații despre angajați, inventar și multe altele), la fel ca alte sisteme, cum ar fi popularul Microsoft Access. În timp ce Microsoft Access, MySQL și alte sisteme de gestionare a bazelor de date servesc scopuri similare (de a găzdui datele), utilizarea diferă foarte mult. [1]

MySQL este componentă integrată a platformelor LAMP sau WAMP (Linux/Windows-

Apache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicație web este strâns legată de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. În multe carți de specialitate este precizat faptul că MySQL este mult mai ușor de învățat și folosit decât multe din aplicațiile de gestiune a bazelor de date, ca exemplu comanda de ieșire fiind una simplă și evidență: „exit” sau „quit”. [2]

* 1. **Cum administrează MySQL bazele de date?**

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau, prin descărcare de pe internet, o interfață grafică: MySQL Administrator și MySQL Query

Browser. Un alt instrument de management al acestor baze de date este aplicația gratuită,

scrisă în PHP, phpMyAdmin. [2]

* 1. **De ce este atât de util MySQL?**

Baza de date MySQL este folosită în principal ca mijloc de a stoca date pentru aplicații

mari, bazate pe web. Site-uri precum WordPress, iStock, GitHub, Facebook, NASA, Marina SUA, Tesla, Scholastic, Spotify, YouTube, Netflix, Glasses Direct, Symantec (și 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]

1. **Tehnologia JSP utilizată în dezvoltarea aplicației**
   1. **Ce este JSP?**

Java Server Pages este o simplă dar puternică tehnologie folosită pe partea de server pentru a genera conținut HTML dinamic. JSP este o extensie directă a Java Servlets și furnizează 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 să suporte mai multe tipuri de documente structurate, îndeosebi HTML și XML. În general, JSP-urile folosesc anumite informații pe care le trimit la server într-o cerere HTTP care interacționează cu datele existente pe acesta și crează dinamic un răspuns organizat într-un format standard (HTML, DHTML, XML, etc.) sau într-un format text sau neorganizat ce va fi trimis înapoi clientului. [2]

* 1. **Ce este SERVLET?**

Un servlet este un fișier java care poate solicita clientul, procesează-l și furnizează un fișier

HTML ca răspuns. Există mai multe servlete în interiorul containerului web. De asemenea, este posibilă maparea mai multor cereri la un servlet. Prin urmare, toate aceste configurații sunt incluse în acest fișier special, care este fișierul web.xml.[5]

* 1. **Care este diferența majoră între JSP și SERVLET?**

Principala diferență între JSP și Servlet este că JSP este un limbaj de scripting al paginii web care poate genera conținut web dinamic în timp ce servletul este un program Java care este deja compilat și folosit pentru a crea conținut dinamic de web. Într-o aplicație web obișnuită, clientul solicită o pagină Web de la server și serverul răspunde înapoi cu pagina necesară. Aceste pagini pot fi fie statice, fie dinamice. Conținutul unei pagini

statice este deja creat. Conținutul unei pagini dinamice este creat în timpul rulării. JSP și Servlet sunt două metode în Java pentru a crea pagini web dinamice. JSP este tradus și compilat într-un servlet de către containerul web. Pe de altă parte, un servlet este un program Java care este gestionat de containerul web.[5]

1. **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 trăsăturile lui, cum ar fi independența față de

platforma, structurarea formatării și legăturile hypertext, fac din el un foarte bun format

pentru documentele Internet și Web.[4]

* 1. **În ce scop utilizăm limbajul HTML?**

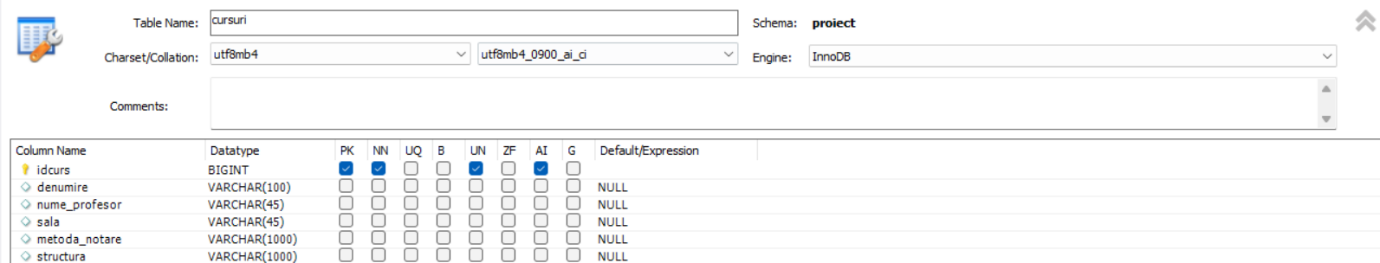
Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele

ș.a.m.d. – decât descrierea semanticii documentului. În cadrul dezvoltării web de

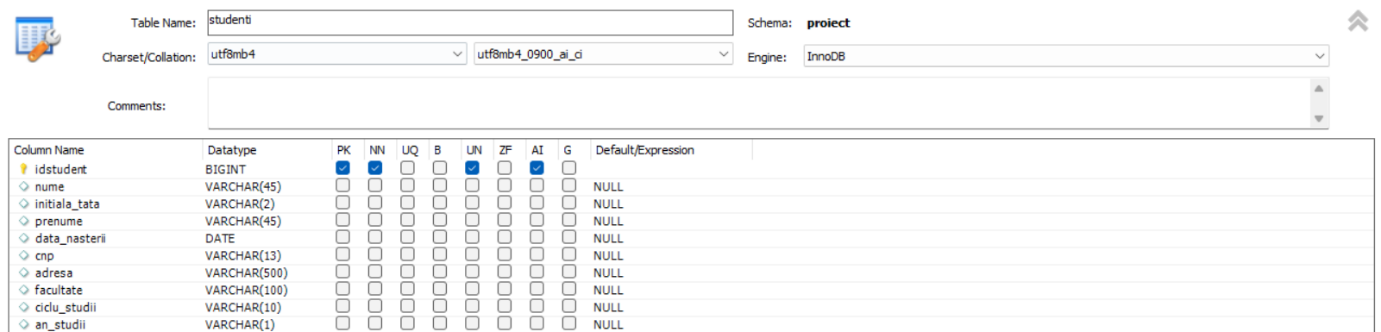
tip front-end, HTML este utilizat împreună cu CSS și JavaScript.[2]

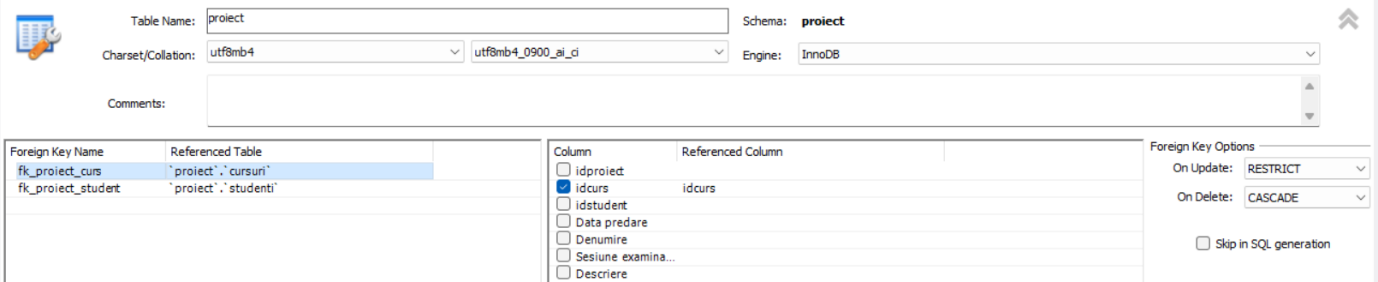
1. **Descrierea aplicației**
   1. **Baza de date**

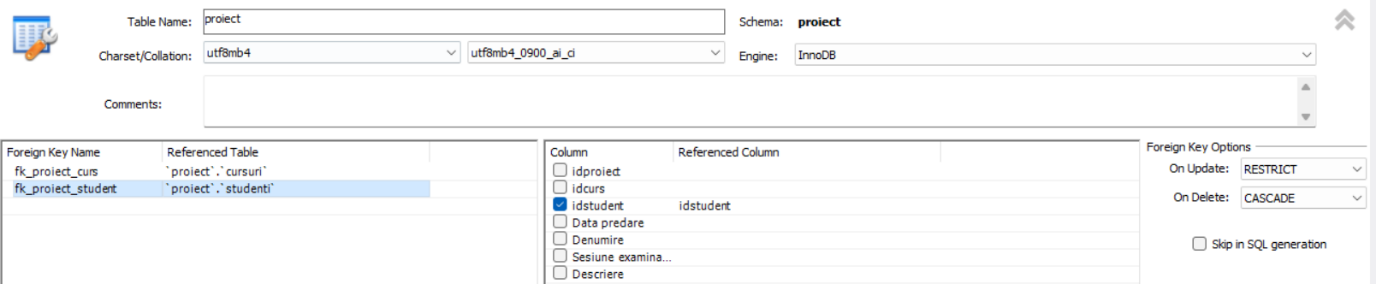
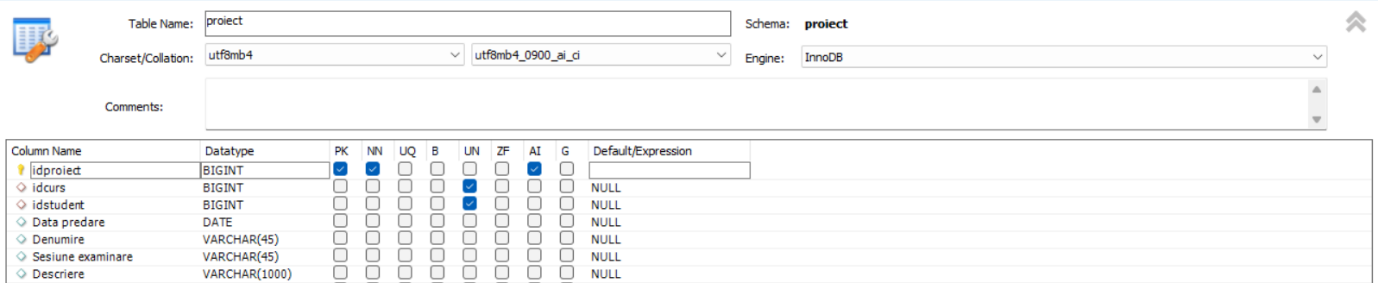
Pentru tabela **cursuri** am ales ca și cheie primară atributul **idcurs.** Restul atributelor sunt denumire, nume\_profesor, sala, metoda\_notare și structura.

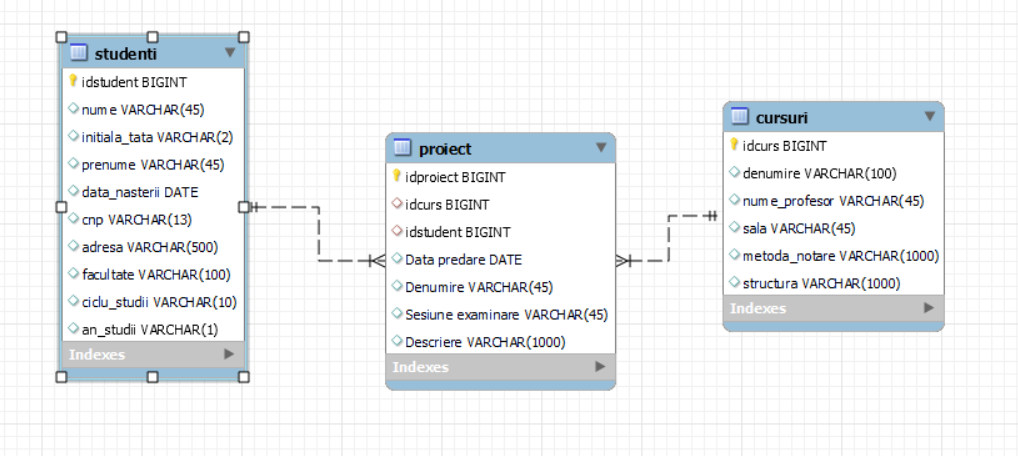


Pentru tabela **studenti** am ales ca și cheie primară atributul **idtudent.** Restul atributelor sunt nume, initiala\_tata, prenume, data\_nasterii, cnp, adresa, facultate, ciclu\_studii, an\_studii.



Pentru tabela **proiect** am ales ca și cheie primară atributul **idproiect**. Restul atributelor sunt Data Predare, Denumire, Sesiune examinare, Descriere, **iar cheile primare din tabelele anterioare devin chei străine (FK) pentru tabela curentă, de legătură.**

****

* 1. **Diagrama logică a bazei de date**
  2. **Funcționalitatea aplicației**

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.

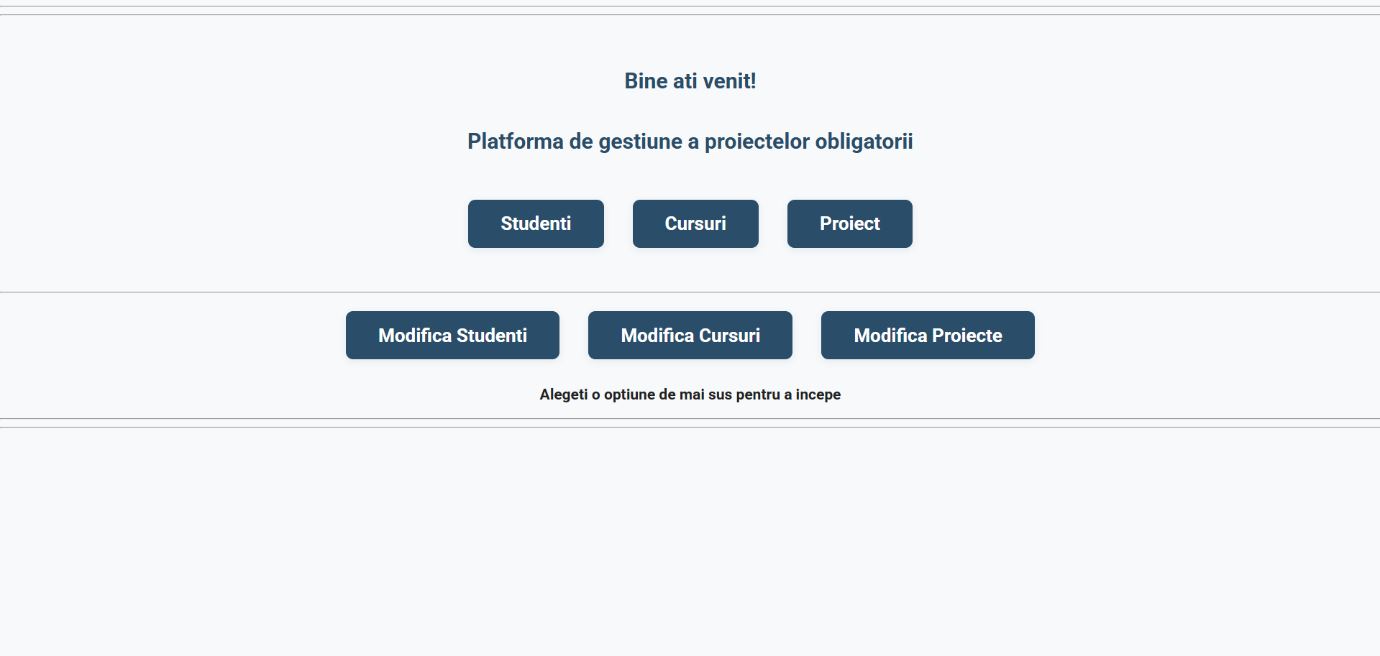
• Folder-ul webapp ce continue toate paginile JSP, ce au rolul de a importa functionalitatile

din clasa JavaBean, fiind conectate intre ele si implicit conectate la pachetul DB, ce

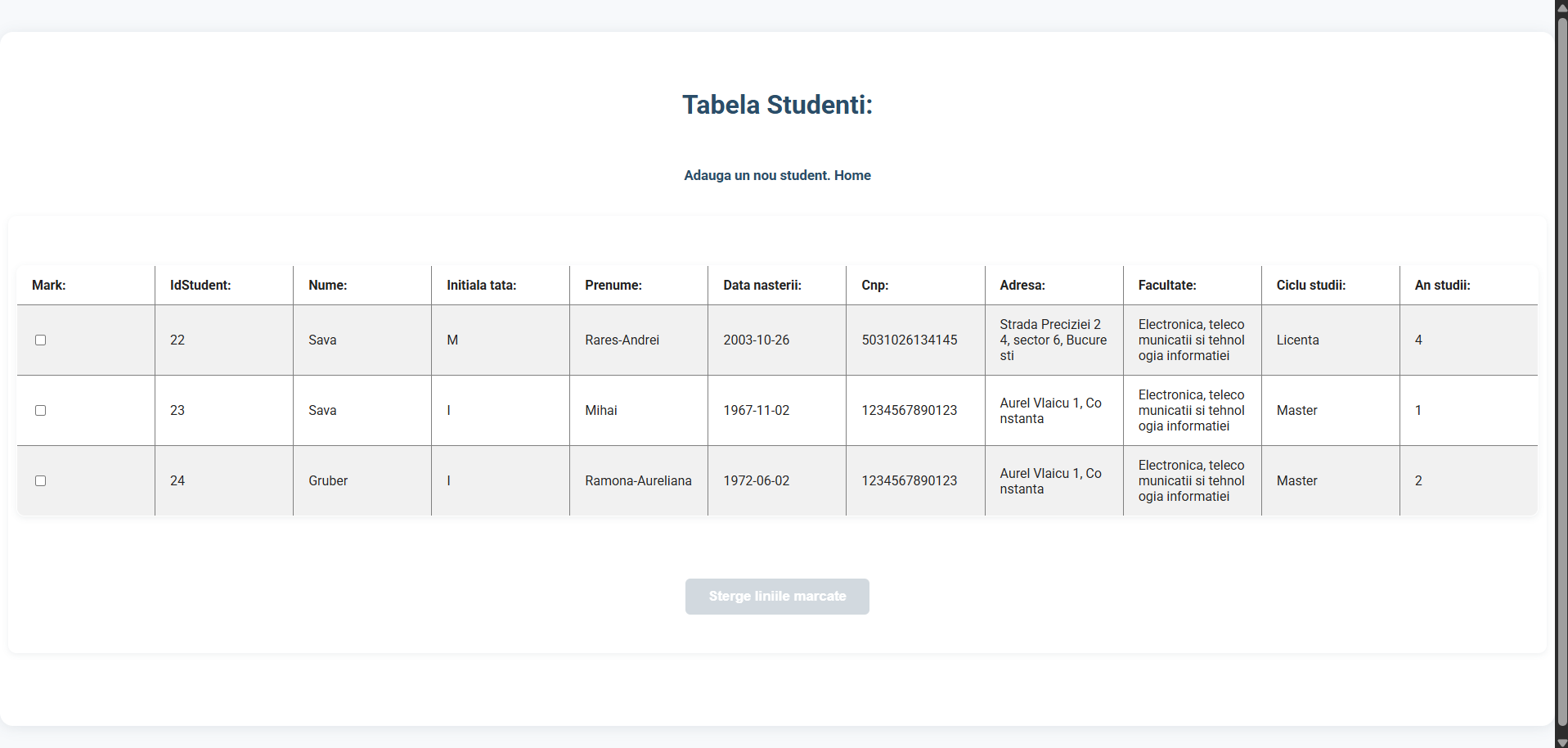
realizeaza partea dinamica a proiectului.

Am implementat mai multe metode car ene permit să facem operațiunile de care avem nevoie: vizualizare tabele (metodele vedeTabela si vedeProiect), inserare de date in tabele (metodele adaugaCurs, adaugaProiect si adaugaStudent), stergere date existente (stergeDateTabela), modificare date (modificaTabela), dar și metode ce ne permit întoarcerea unei linii folosind id-ul.

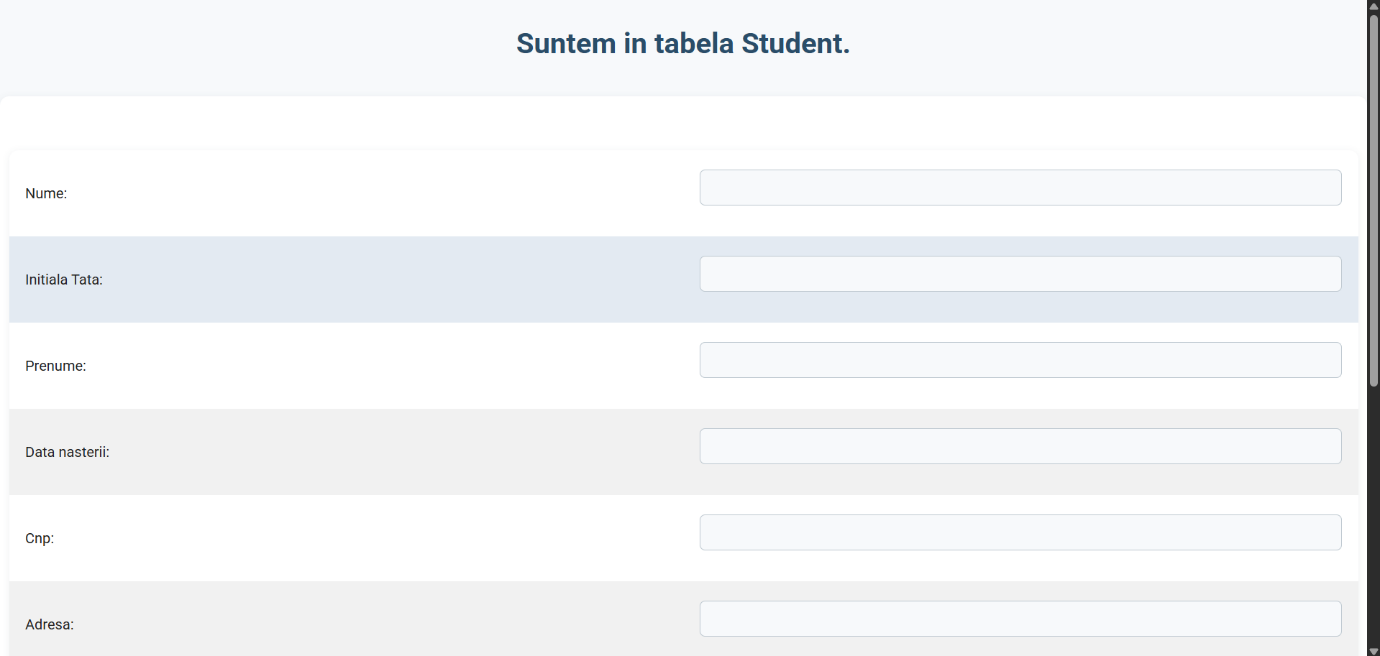
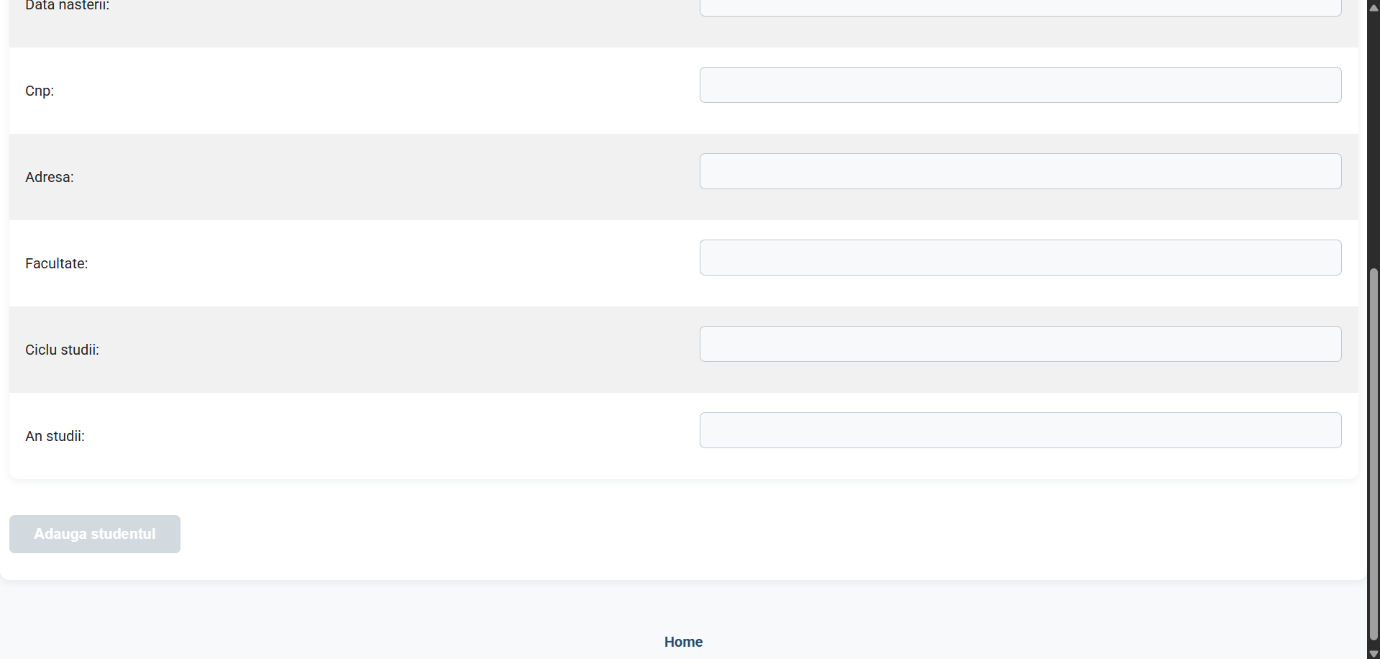
Pagina principală a interfeței (index.html) permite accesul spre paginile tabelelor unde putem vizualiza, adăuga, modifica sau șterge ce date dorim.

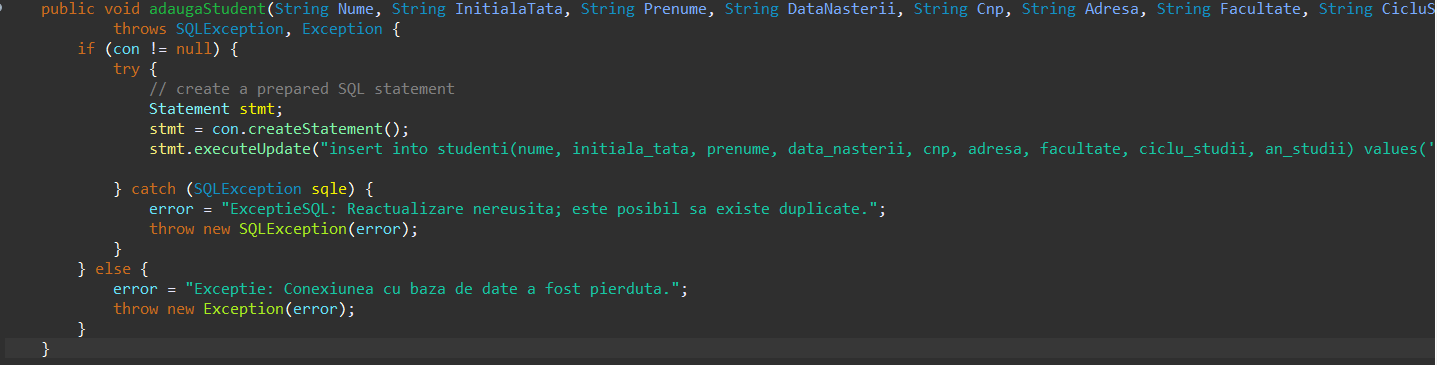


Din pagina principală, utilizatorul are posibilitatea de a intra pe 3 pagini aferente tabelelor **studenți**, **cursuri** și **proiecte**. Pe fiecare dintre aceste pagini poate vizualiza tabelele din baza de date, adăuga și șterge date. În “Modifica Studenti”, “Modifica Cursuri”, “Modifica Proiecte” se pot edita linii și se pot adăuga date noi.

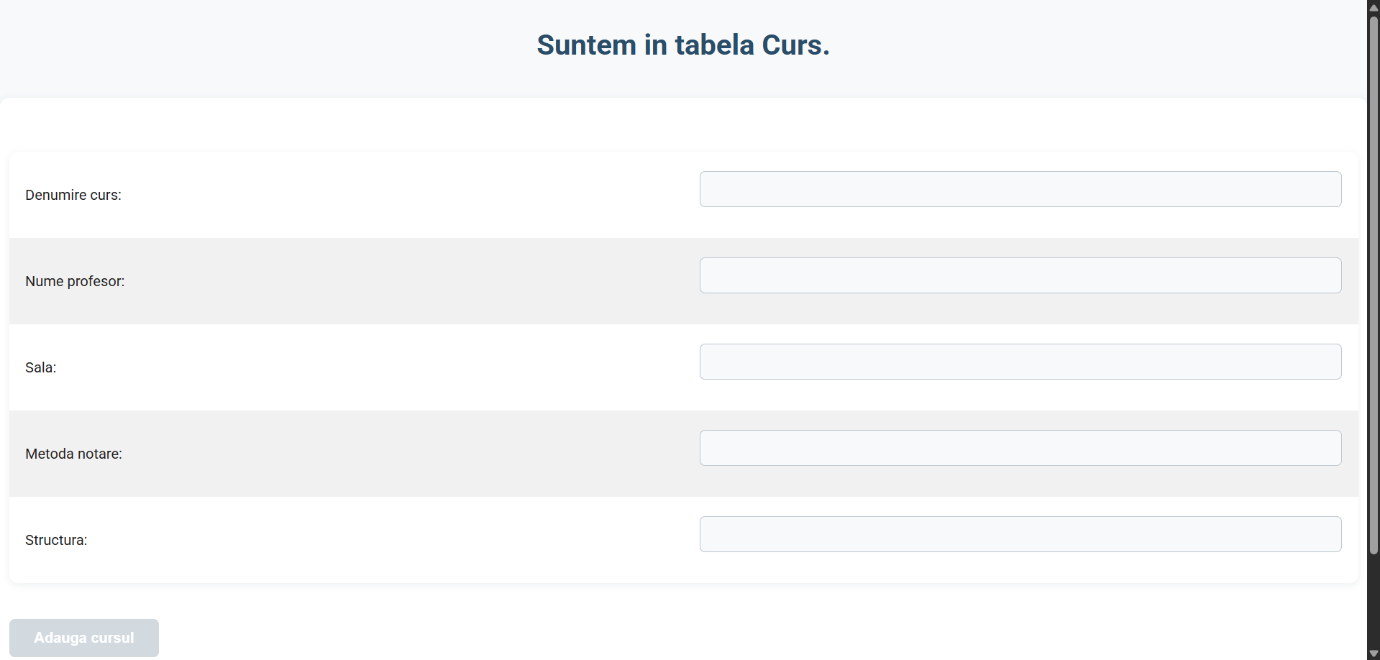
Funcționalitatea paginilor este similară pentru cele trei pagini, astfel: pentru a adăuga o înregistrare nouă în tabelă (student, curs sau proiect), utilizatorul trebuie să apese pe butonul corespunzător. Se va deschide o pagină în care sunt afișate datele din tabelele corespunzătoare.

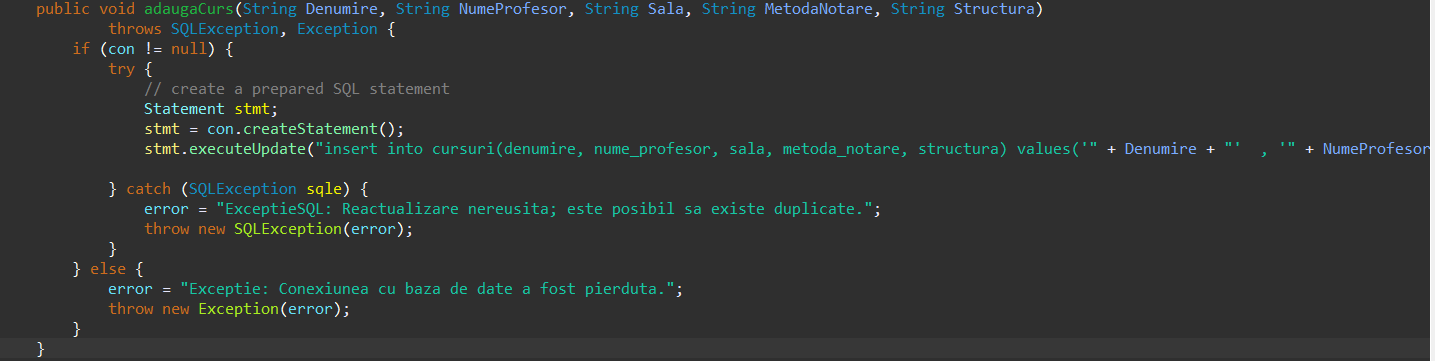
Dacă se dorește adăugarea unui nou student, utilizatorul dă click pe butonul “Adauga un nou student” și va fi trimis pe o nouă pagină cu un formular unde poate completa datele necesare.

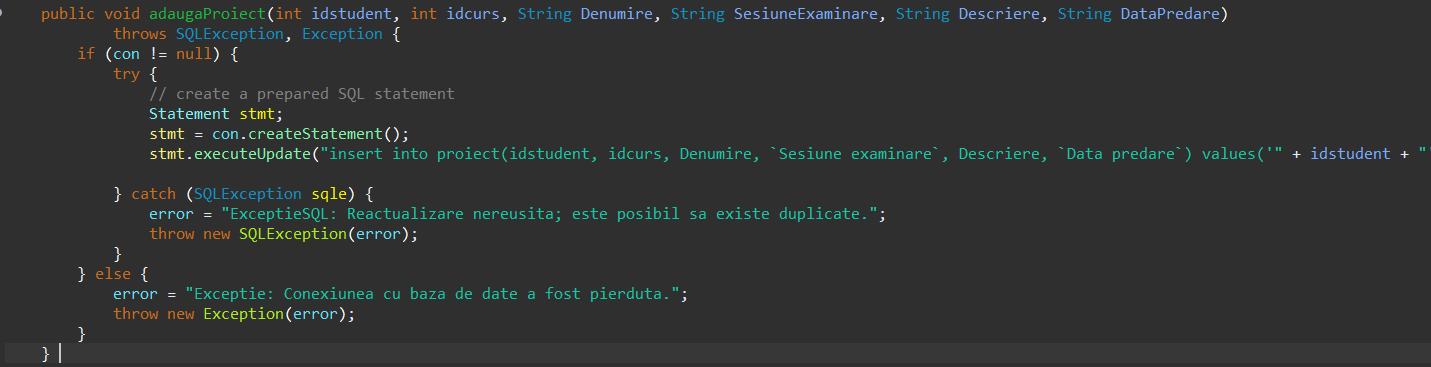
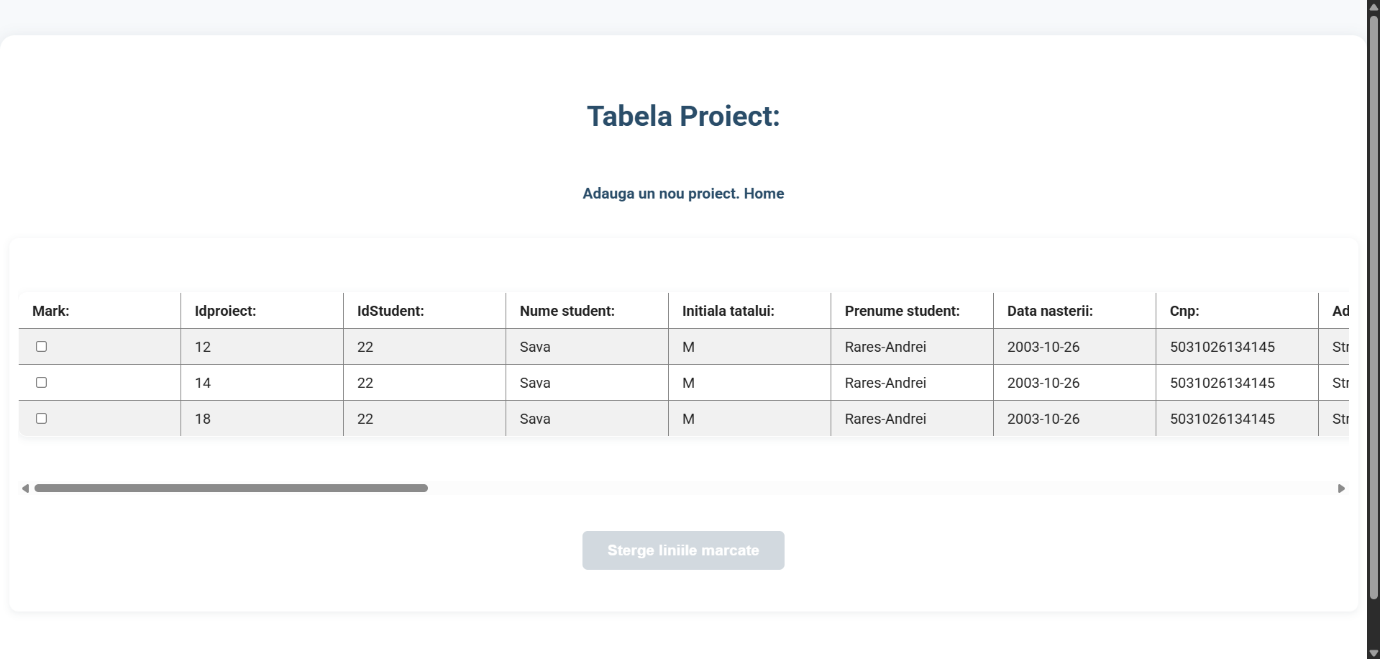






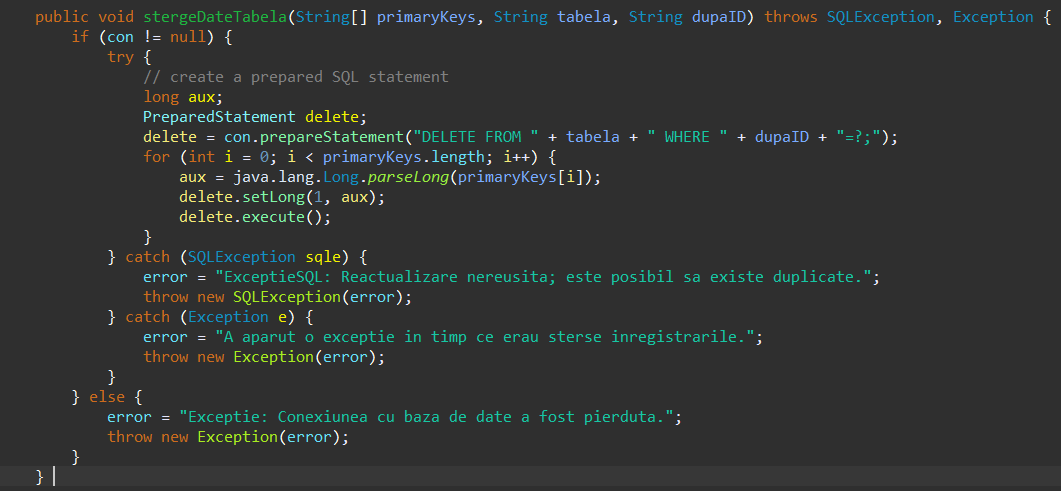
Dacă se dorește adăugarea unui nou curs, utilizatorul dă click pe butonul “Adauga un nou curs” și va fi trimis pe o nouă pagină cu un formular unde poate completa datele necesare.

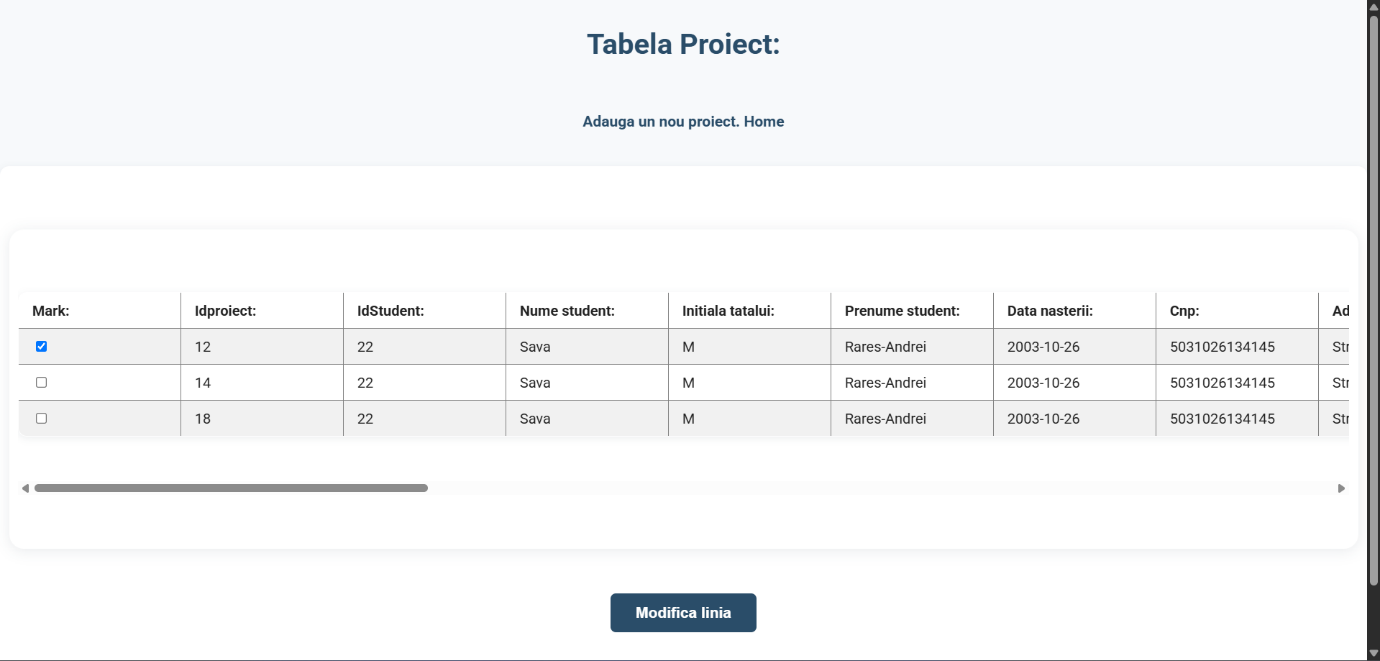


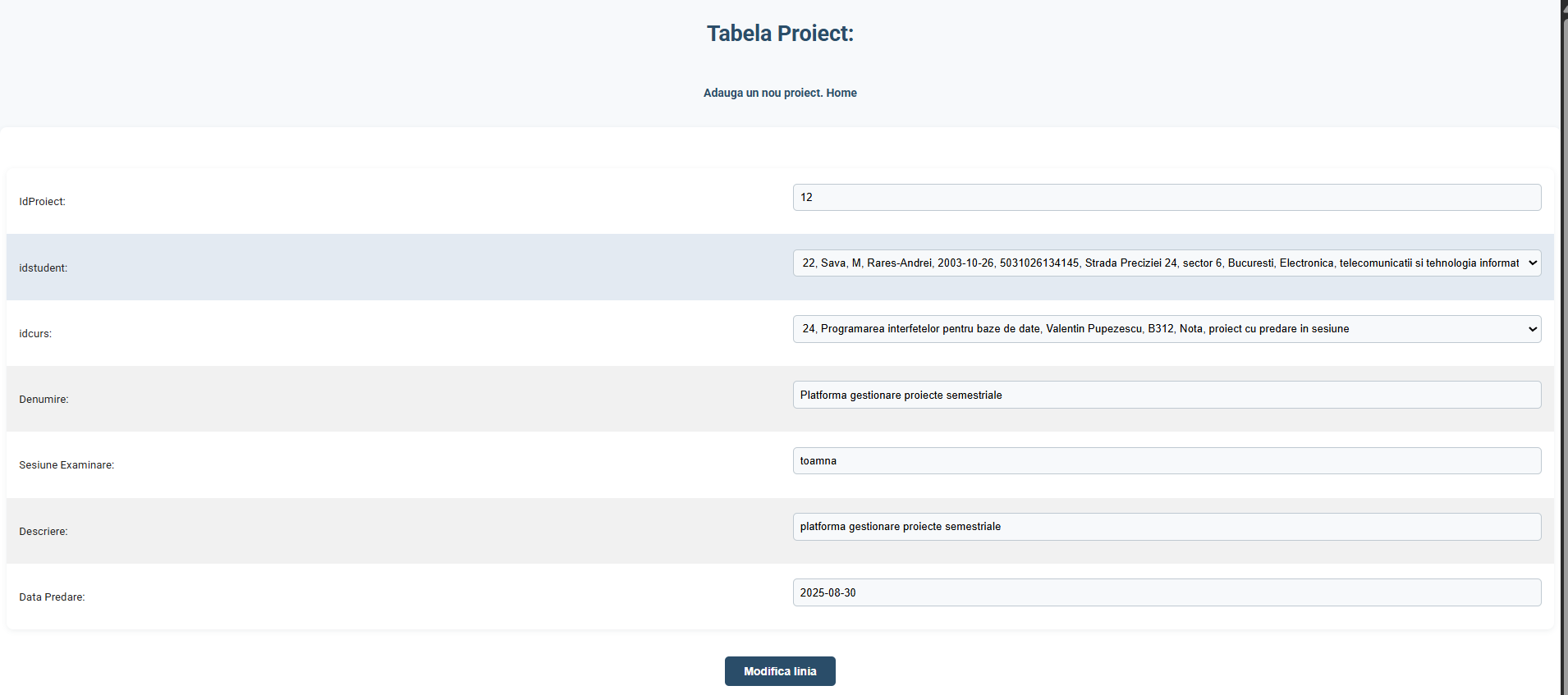
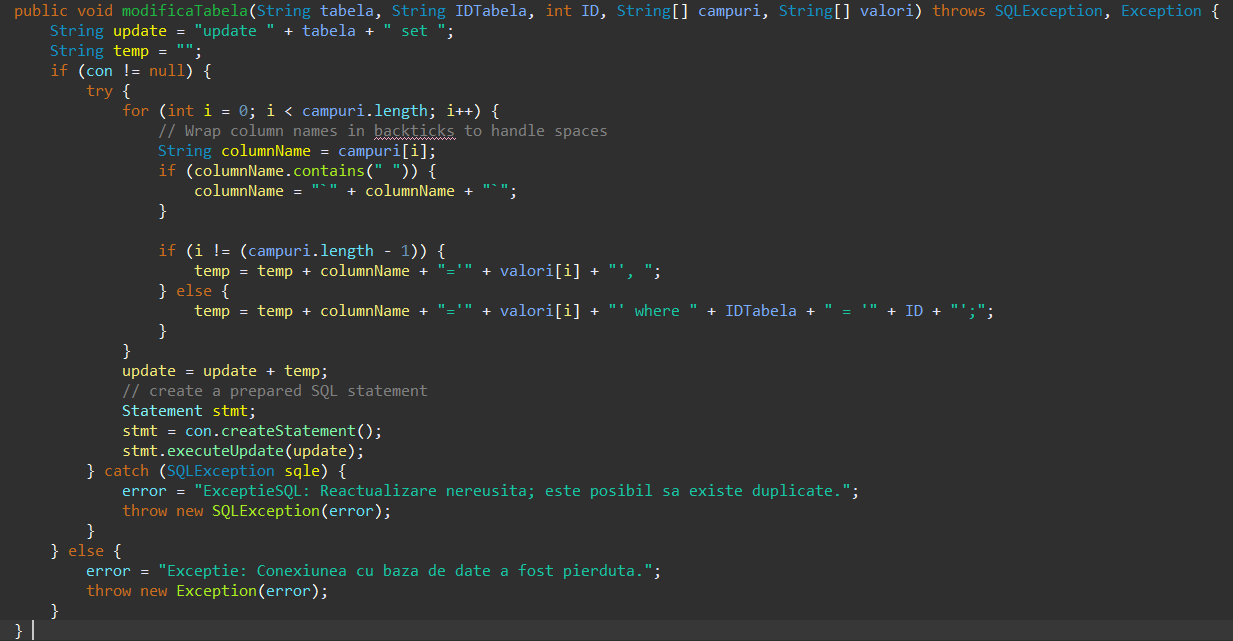
Dacă se dorește adăugarea unui nou proiect, utilizatorul dă click pe butonul “Adauga un nou proiect” și va fi trimis pe o nouă pagină cu un formular unde poate completa datele necesare. Singura diferență față de modul de inserare al datelor de până acum este că **trebuie sa selectăm studentul și cursul aferent proiectului nou introdus (studentul, autor, și cursul pentru care a fost realizat)**

Pentru a șterge câmpuri din tabele selectăm linia dorită și apăsăm pe butonul de ștergere “Sterge liniile marcate”





Pentru modificarea datelor din tabele, din pagina principală se poate da click pe cele trei butoane de modificare, se selectează entitatea, se apasă pe “Modifica linia”, se introduc schimbările si se apasă din nou pe “Modifica linia”.

1. **Concluzie**

Cursul m-a învățat să pot crea o interfață prietenoasă care permite gestionarea datelor dintr-o bază de date MySQL. Folosind interfața creată, secretariatul unei facultati poate să gestioneze ușor informațiile despre studenți, cursuri și proiectele aferente, fără a necesita un angajat specializat pentru administrarea datelor. Oricare angajat existent poate să efectueze aceste operații prin intermediul interfeței. Mai mult decât atât, datorită tehnologiilor JSP și MySQL, aplicația poate fi implementată pe un server, ceea ce permite accesul la aplicație din mai multe locații, de pe tot globul, atâta timp cât există o conexiune la internet. Astfel, nu mai este necesar ca un administrator de baze de date să se ocupe de gestionarea bazei de date folosind un program de management precum PhpMyAdmin, unde este necesar un nivel de cunoștințe SQL pentru a efectua operațiunile necesare.

1. **Bibliografie**

• https://www.nav.ro/blog/ce-este-mysql/ [1]

• https://ro.wikipedia.org/ [2]

• Cursuri PIBD [3]

• https://web.ceiti.md/lesson.php?id=1 [4]

• https://ro.sawakinome.com/ [5]