Universitatea “POLITEHNICA” din Bucuresti

Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

**Proiect**

**Programarea Interfetelor pentru Baze de Date**

**-Tehnologie Hibernate-**

Prof. coordonator: Student: Chindea Eduard-Cristian

Dr. Ing. Pupezescu Valentin Grupa: 431B

***Cerinta***

Creaţi 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 este comună). La crearea interfeţelor se vor folosi două tehnologii(la alegere - ex.: JSP, Hibernate, JPA, .NET, Python etc.).

***Intruducere***

Tehnologia folosita pentru crearea bazei de date este MySQL.

MySQL este un sistem de genstiune a bazelor de date relationale, produs de compania suedeza MySQL AB si distrubuita sub Licenta Publica Generala GNU. Este cel mai popular SGBD open-source la ora actuala, fiind o componenta cheie a stivei LAMP (Linux, Apache, MySQL, PHP).

***Implementare***

Baza de date a fost realizata cu ajutorul MySQL Workbench, fiind alcatuita din doua tabele: angajati si reviewuri. Pentru a realiza o asociere intre aceste doua tabele de tip M:N, a fost creata o tabela intermediara numita centralizare.

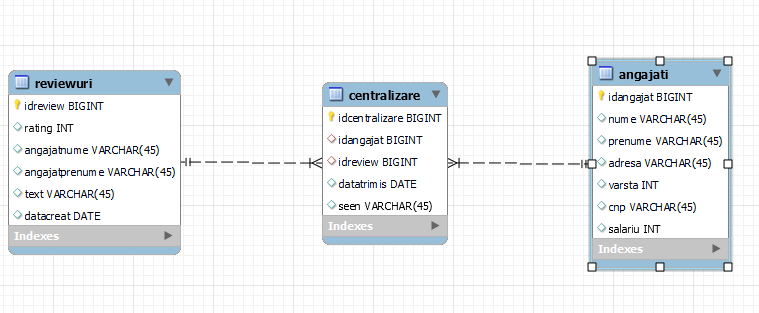
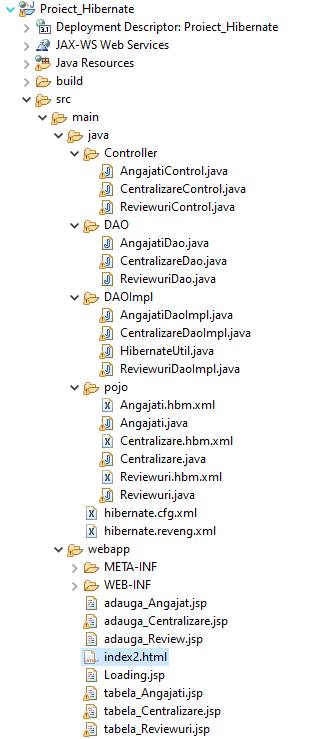


Diagrama EER a tabelelor si continuturilor acestora

**Tehnologia Hibernate**

Hibernate ORM este un instrument de mapare obiect-relational pentru limbajul de programare Java. Acesta ofera un cadru pentru cartografierea unui model de domeniu orientat pe obiecte intr-o baza de date relationala. Hibernate se ocupa de problemele de nepotrivire a impedantei obiect-relationale prin inlocuirea accesurilor directe si permanente ale bazei de date cu functii de manipulare a obiectelor la nivel inalt.

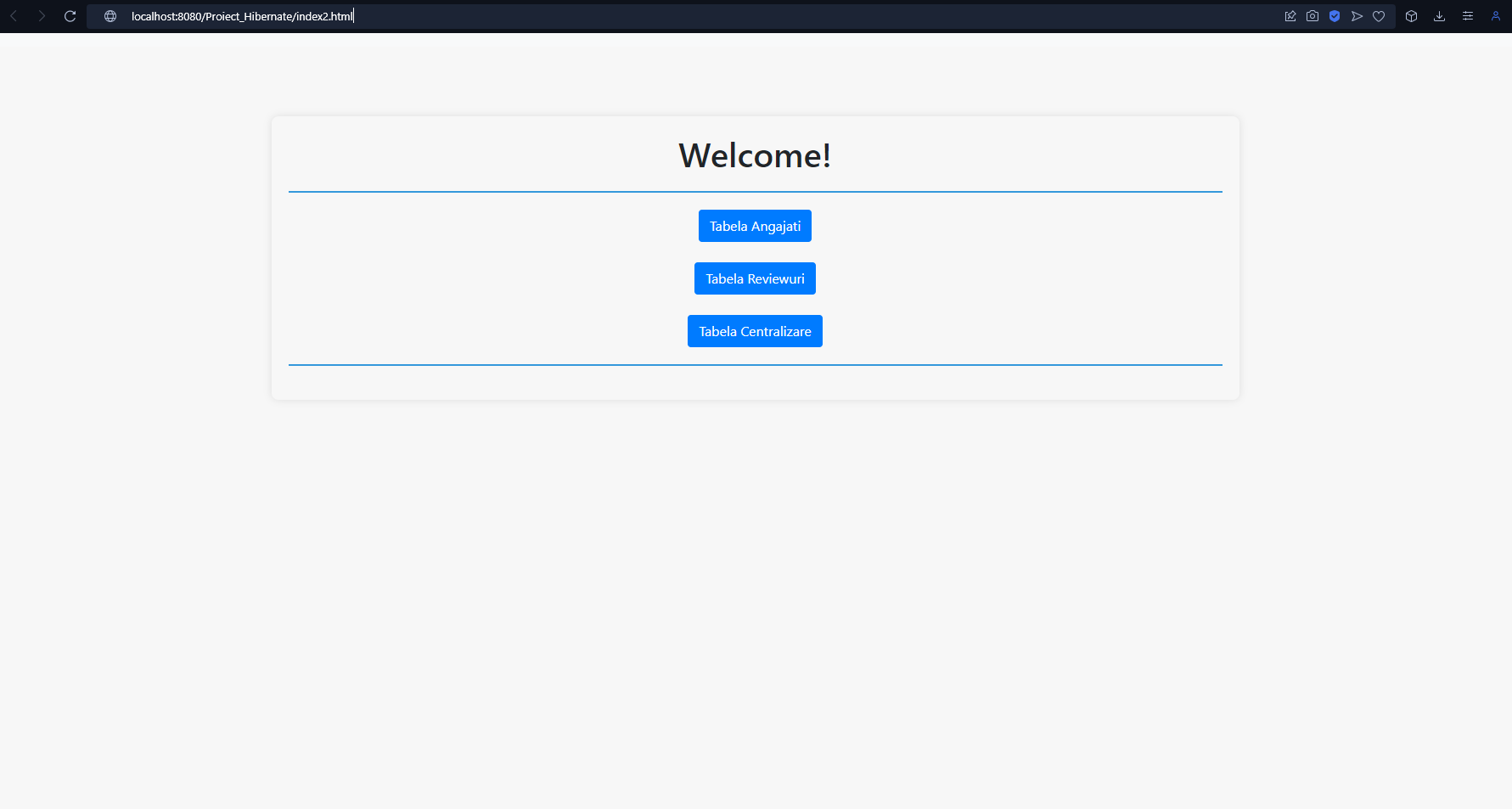


Se poate observa existenta a 3 clase aferente celor 3 tabele din baza de date MySQL. De asemenea avem 3 interfete, cate una pentru fiecare tabela: AngajatiDao, CentralizareDao si ReviewuriDao. Acestea contin metodele de adaugare, afisare, modificare si stergere a campurilor din baza de date. Interfetele sunt implementate de clasele DaoImpl.

In final, avem pentru fiecare tabela cate o clasa de control. Acestea deriva din clasa abstracta HttpServlet din pachetul jakarta.servlet.http.HttpServlet. Cu ajutorul acestei clase putem crea un Servlet HTTP care poate fi folosit pentru un website. Clasa HibernateUtil este folosita pentru generarea campului de tip StaticSessionFactory cu ajutorul caruia putem efectua operatii pe baza de date.

***Explicarea codului***

Pagina principala a interfetei, index2.html, permite accesul catre paginile tabelelor, unde putem vizualiza, adauga sau modifica datele existente.



Pagina principala a interfetei

Din pagina principala, utilizatorul are posibilitatea de a intra pe una din cele 3 pagini aferente tabelelor. Functionalitatea celor 3 pagini este similara: pentru a modifica tabelul (adaugare, modificare, stergere), utilizatorul are la dispozitie 3 butoane pentru fiecare metoda.

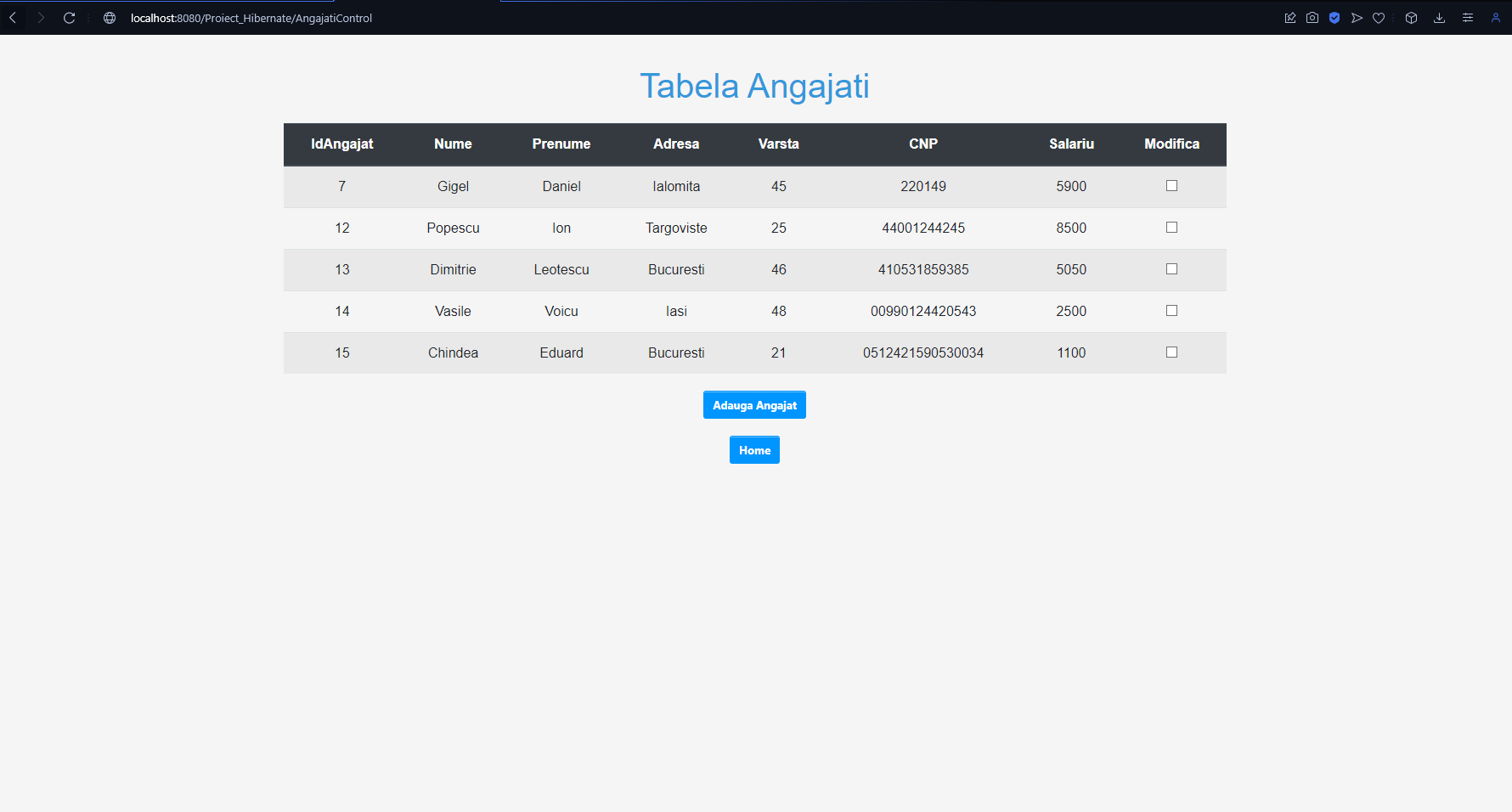


Tabela Angajati

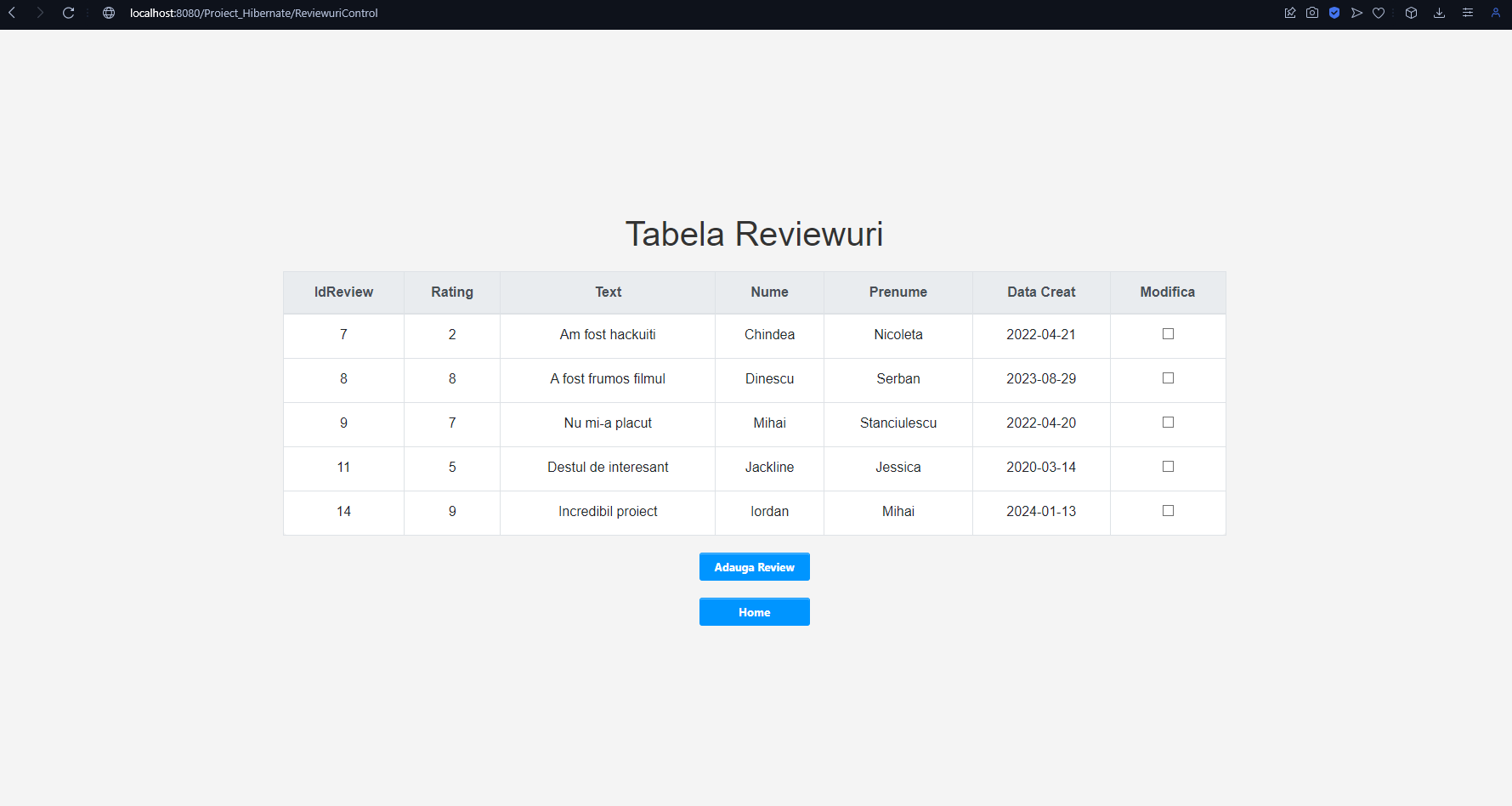


Tabela Reviewuri

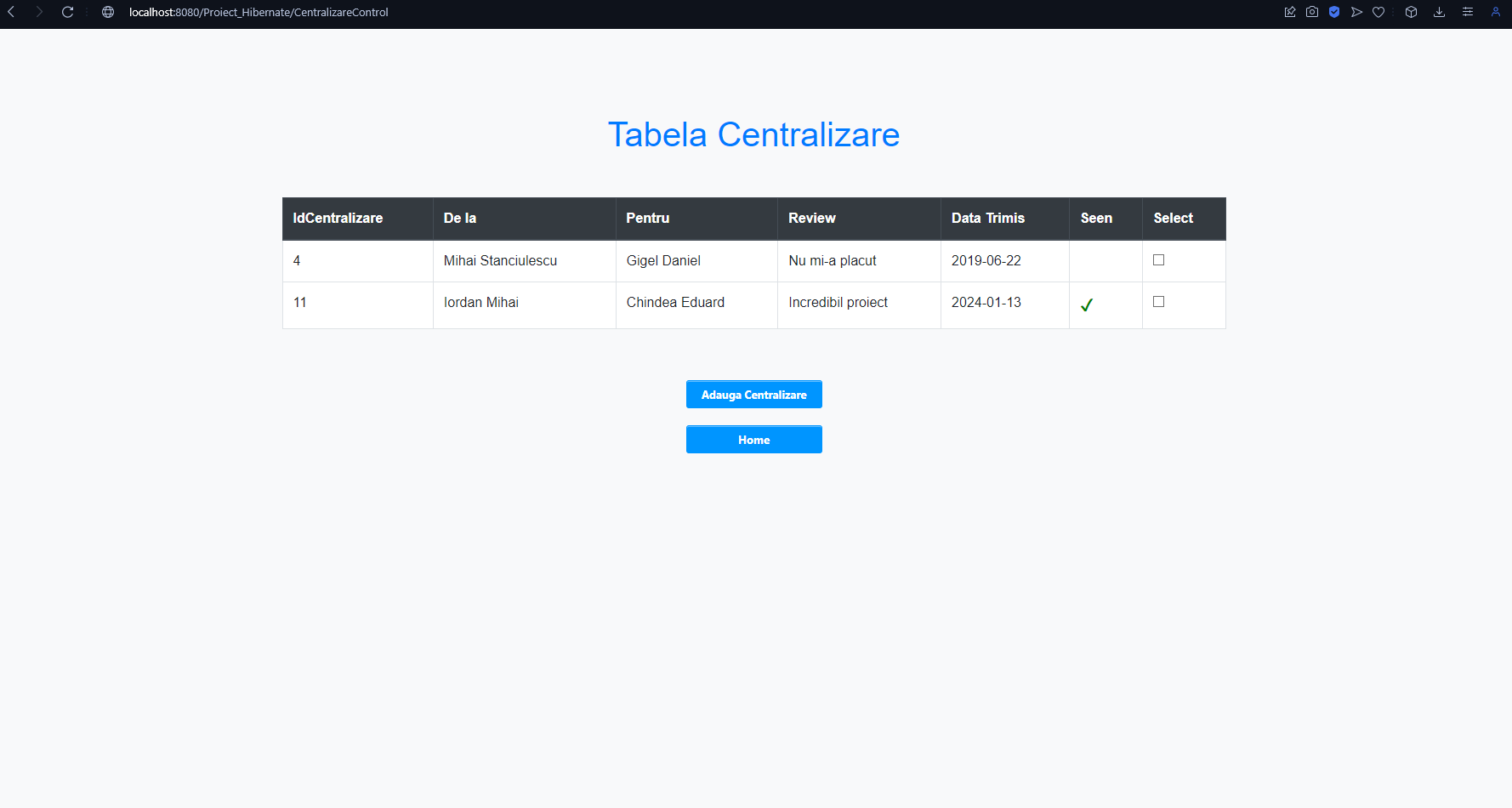
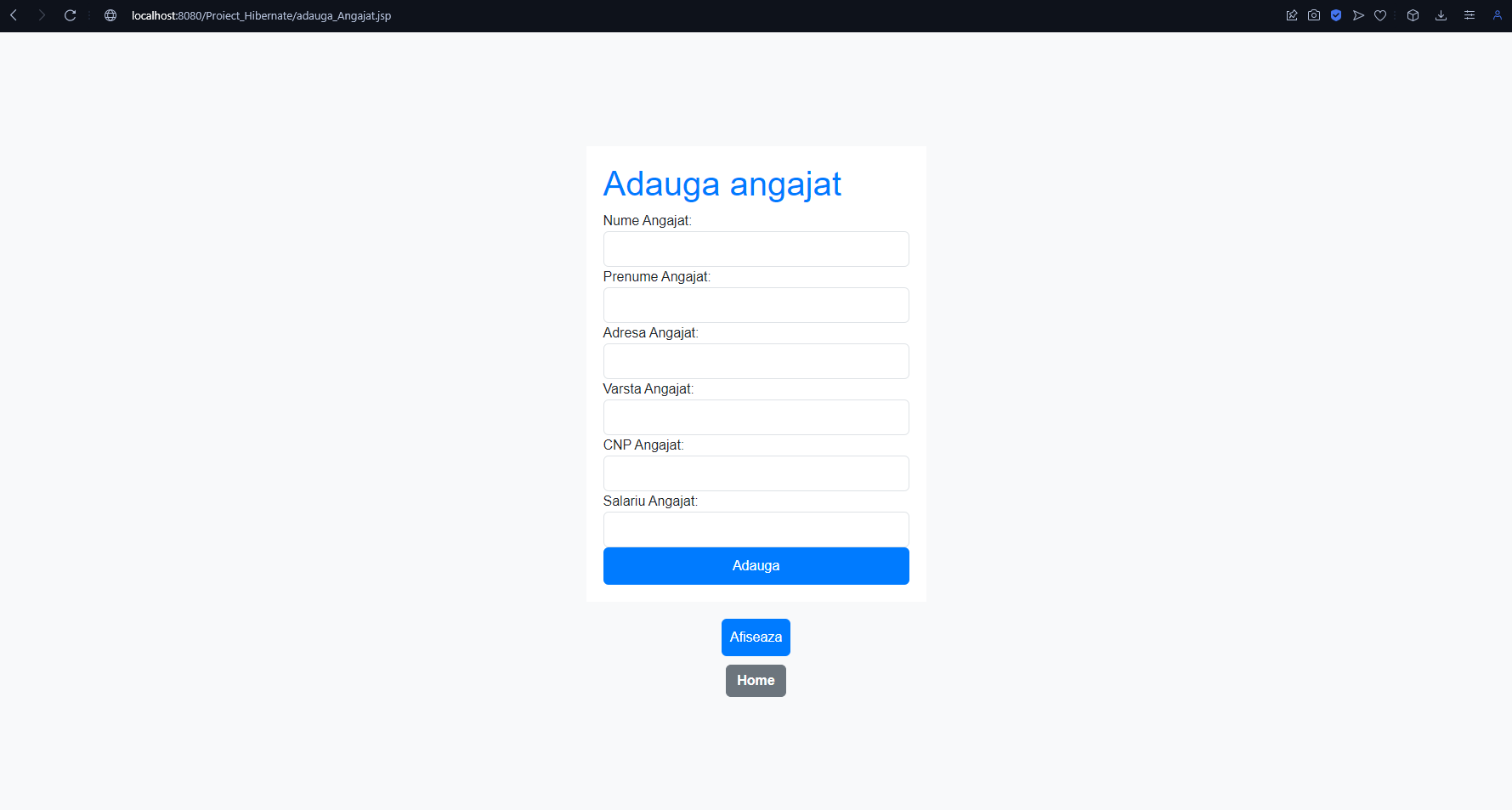


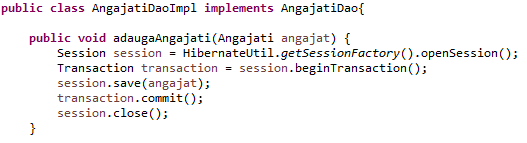
Tabela Centralizare

Pentru adaugarea unui nou angajat, se da click pe butonul “Adauga Angajati”, care va deschide o pagina cu un formulare pe care untilizatorul il completeaza cu datele necesare.



Adaugarea in tabel a unui angajat

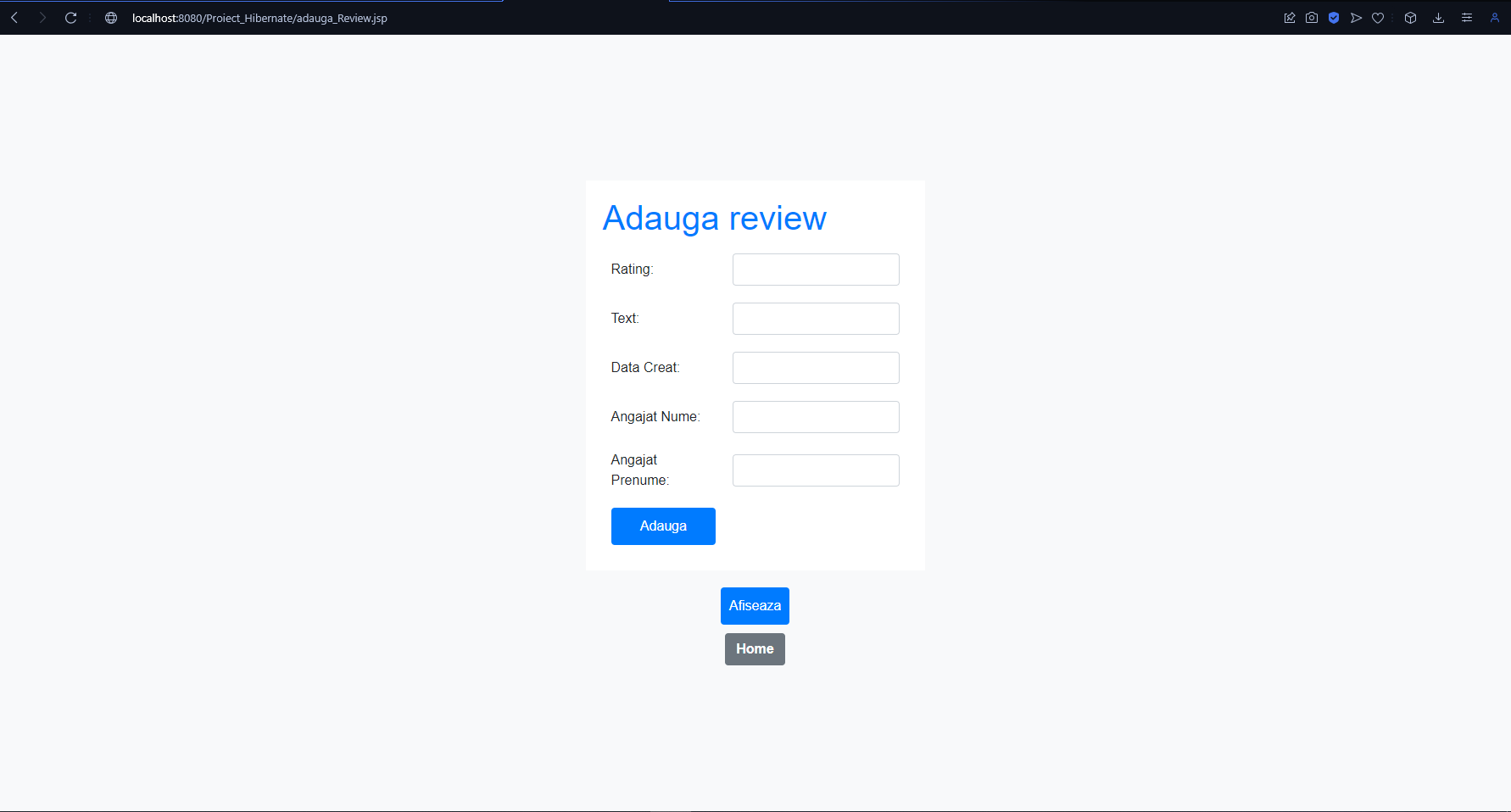




Codurile necesare adaugarii unui angajat

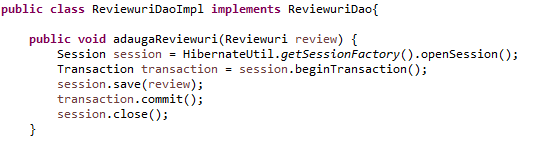
Codul din clasa AngajatiControl ne permite sa preluam datele din formularul htlm folosind metoda request.getParameter(). Folosind datele, cream un obiect de tip angajat si il adaugam in baza de date prin intermediul clasei AngajatiDaoImpl. In aceasta clasa adaugam, cu ajutorul metodei adaugaAngajati si campului Session din HibernateUtil, un nou angajat cu ajutorul instructiunii session.save(angajat).

Similar, in cazul tabelei cu reviewuri avem clasele ReviewuriControl si ReviewuriDao pentru adaugarea unui review, respectiv CentralizareControl si CentralizareDao pentru a adauga o centralizare.



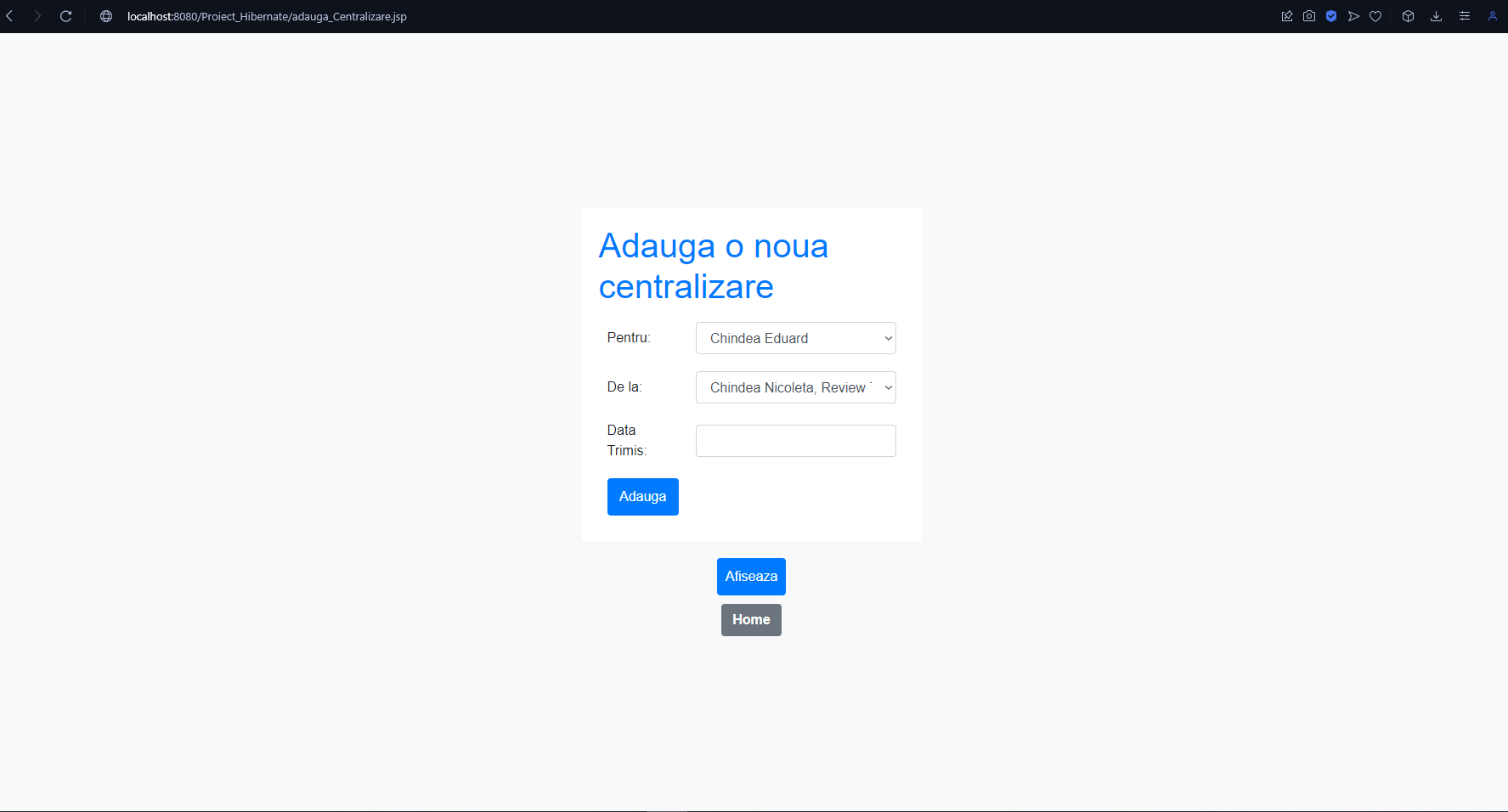
Pagina pentru adaugarea unui review





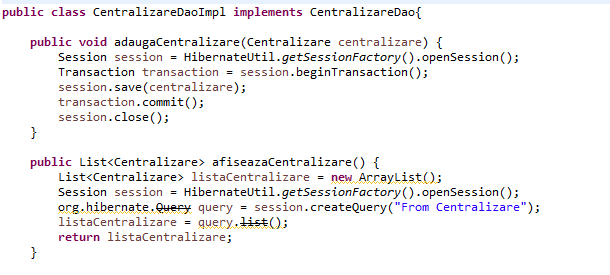
Codurile necesare adaugarii unui review

In cazul tabelei Centralizare, codul difera prin faptul ca aceasta tabela este una de legatura, in care trebuie sa selectam angajati si reviewuri deja existente din celelalte tabele



Pagina pentru adaugarea unei centralizari



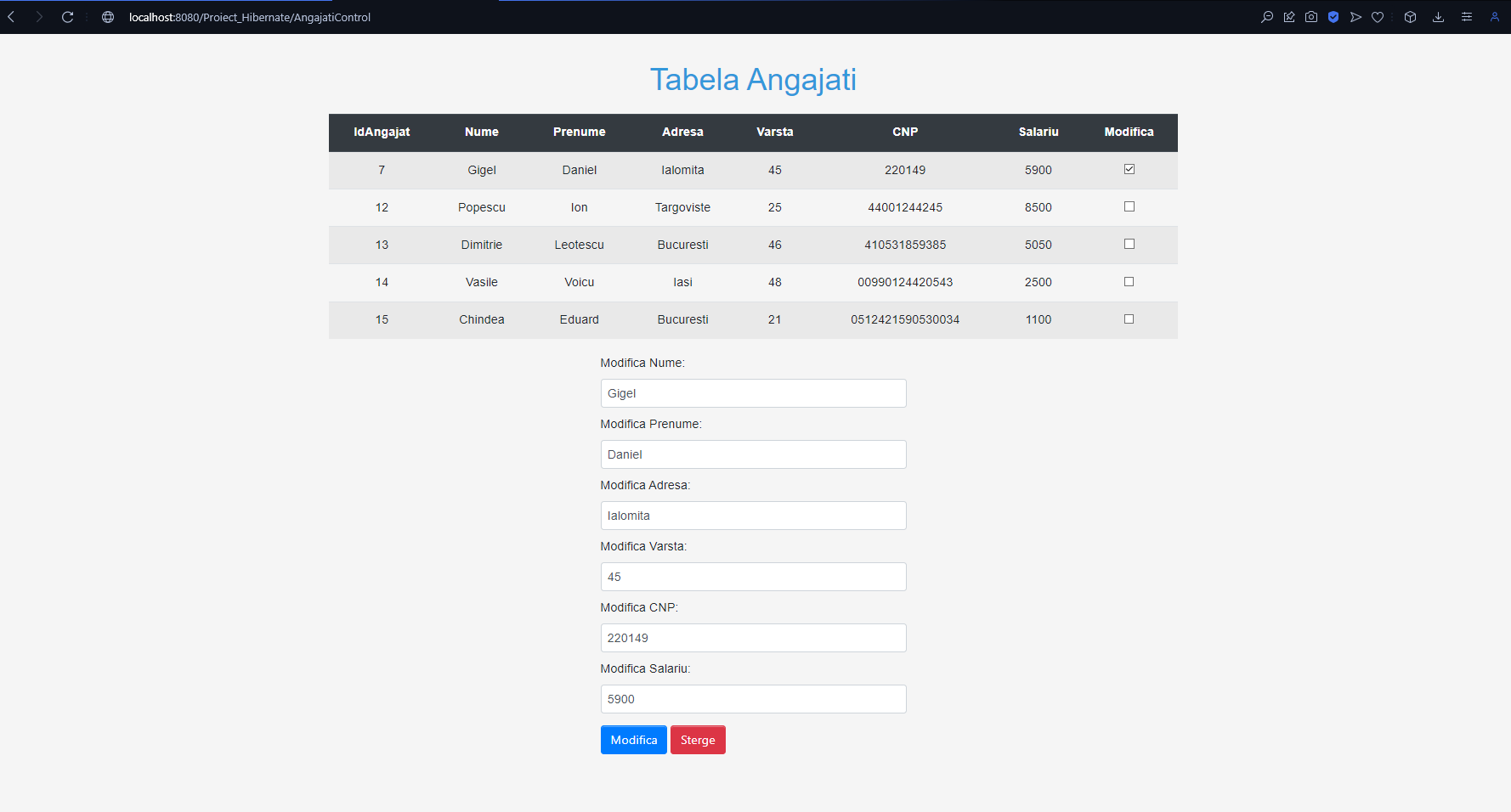


Codurile pentru adaugarea unei centralizari

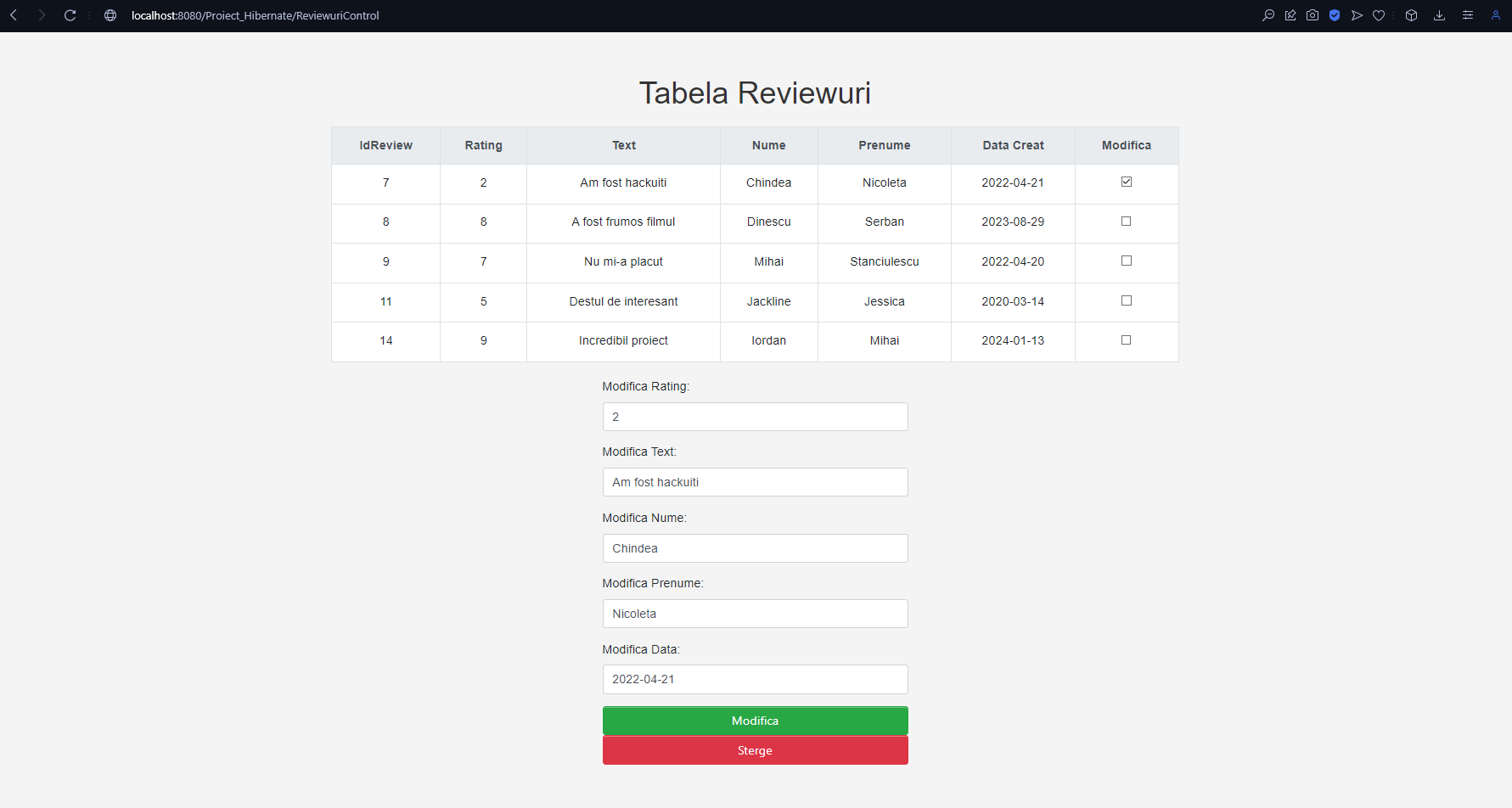
Deoarece tabela de centralizare este o tabela de legatura, in functia doPost din CentralizareControl facem rost de angajati si reviewurile asociate folosindu-ne de metoda session.get() si de id-ul pentru angajati, respectiv reviewuri.

In cazul centralizarii, cat si al reviewurilor, avem parametrii datatrimis/datacreat, care sunt de tip Date cu formatul YYYY-MM-DD. In formularele de adaugare pentru aceste tabele, acesti parametrii sunt adaugati ca string, facand conversia in clasele de control.

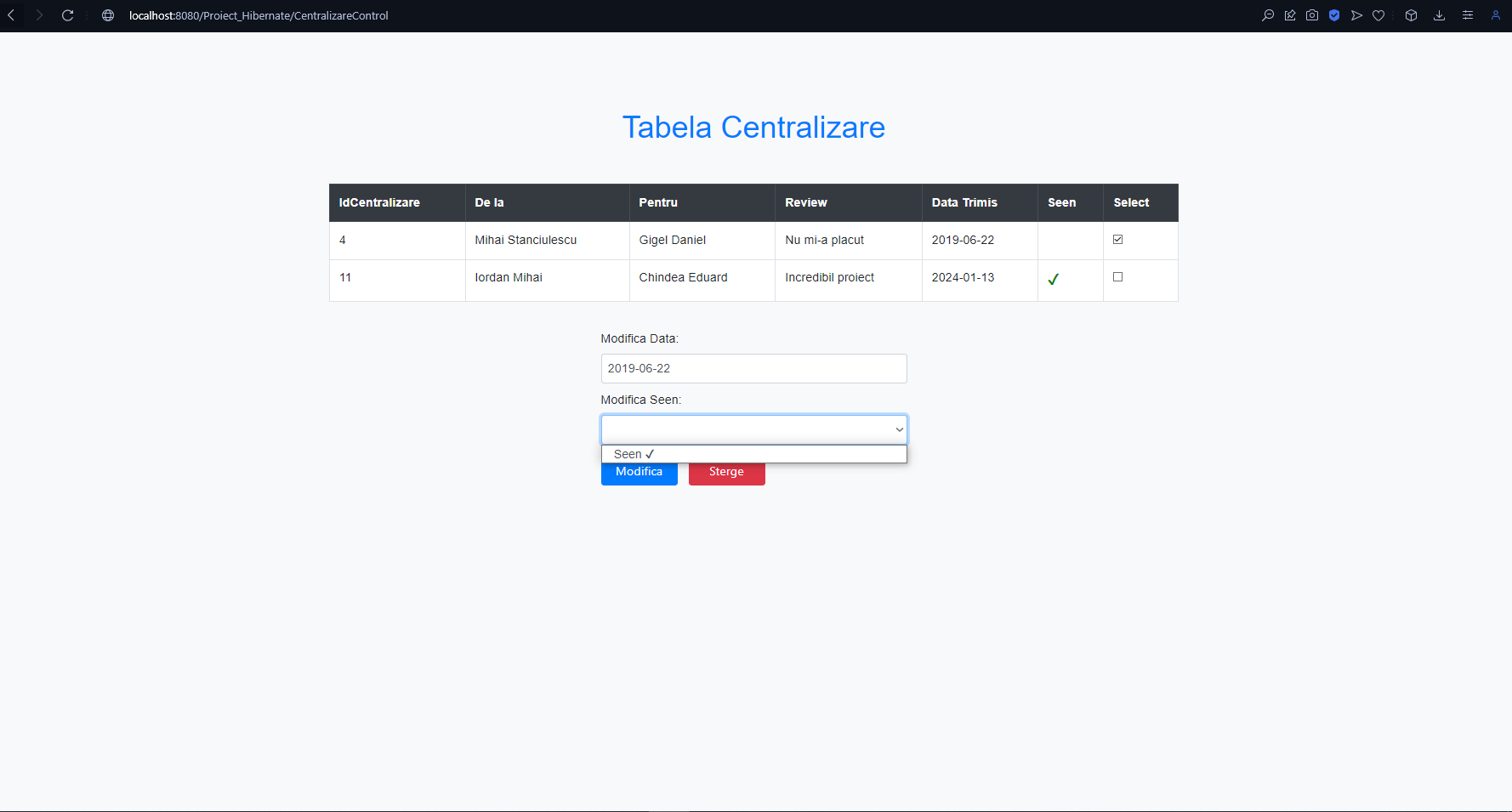
In cazul stergerii sau modificarii in orice tabel, se selecteaza checkbox-ul din dreptul liniei cu care dorim sa lucram.



Modificarea si stergerea din tabela Angajati

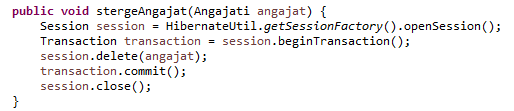


Modificarea si stergerea din tabela Reviewuri



Modificarea si stergerea din tabela Centralizare

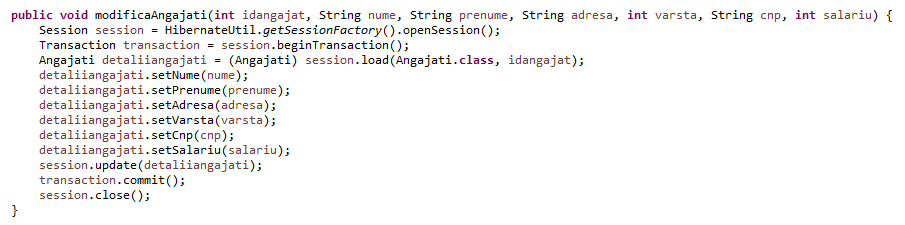
Pentru a sterge un angajat, selectam checkbox-ul din dreptul lui si apasam pe butonul Sterge.



Codurile necesare stergerii unui angajat

Pentru stergerea unui angajat, se intra in if-ul din metoda doPost. Aici, id-ul angajatului se converteste in integer (din string), dupa care se sterge din tabela angajatul cu id-ul obtinut, prin apelarea metodei stergeAngajat, in care se foloseste campul session din clasa HibernateUtil cu functia session.delete(angajat). In mod similar, se sterge si un review sau o centralizare.

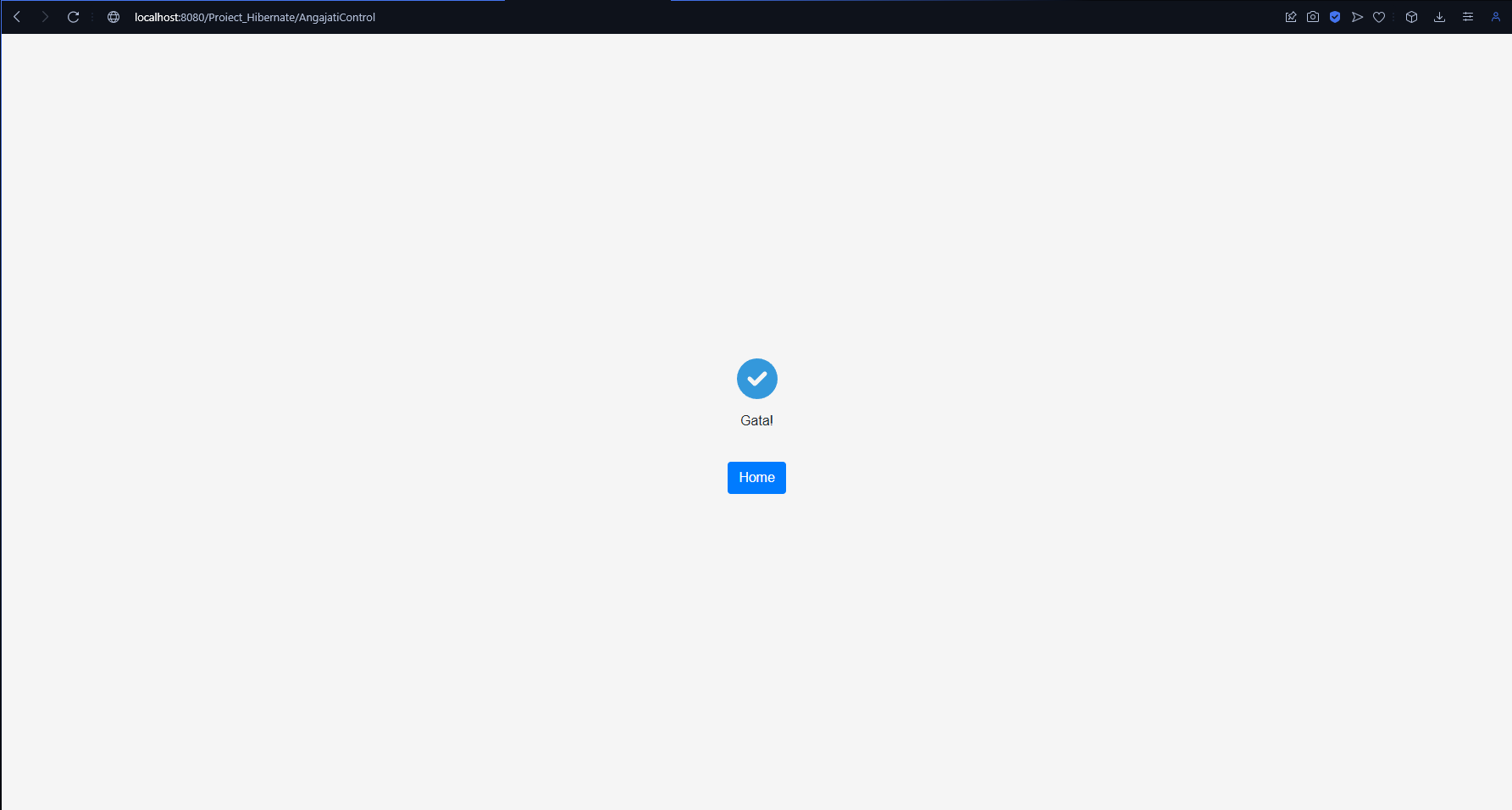
In cazul modificarii unui camp, dupa bifarea checkbox-ului pentru a selecta un angajat, apare un formular unde se poate actualiza orice camp doreste utilizatorul.



Codul pentru modificarea tabelei de angajati

Ca si la stergere, se foloseste if-ul din metoda doPost, prin care se preiau campurile ce se doresc a fi modificate si se apeleaza metoda modificaAngajati din AngajatiDaoImpl. Astfel, se preiau date angajatului selectat si se modifica toate campurile cu datele nou introduse, dupa care, folosind session.update(detaliiangajati) se actualizeaza baza de date cu noile data. In mod similar se modifica datele si in cazul reviewurilor si centralizarilor.

In cazul modificarii (adaugare, modificare, stergere) oricarui tabel, utilizatorul este trimis dupa apasearea butonului specific catre o pagina de load, pentru a se confirma ca schimbarile au avut loc, dupa care se poate intoarce la pagina principala apasand butonul Home.



Pagina de loading

Pentru inbunatarirea aspectului paginilor am folosit elemente de programare HTML, CSS si Bootstrap.

***Bibliografie***

- Curs de Programare a Interfetelor pentru Baze de Date (V. Pupezescu)

- <https://hibernate.org/orm/>

- <https://en.wikipedia.org/wiki/MySQL>

- <https://stackoverflow.com/>

- <https://chat.openai.com>

- <https://getcssscan.com/css-buttons-examples>