UNIVERSITATEA „POLITEHNICA” DIN BUCUREȘTI

Facultatea de Electronică Telecomunicații și Tehnologia Informației

PROIECT P.I.B.D.

Programarea unei aplicații în tehnologie JSP

ce operează o bază de date în tehnologie MySQL

Tițu Alex-Costin

Grupa 433A

București 2022

Cuprins

1. Structura aplicației și informații generale ........... pag. 3-5
2. Descrierea aplicației ............................................ pag. 5-7
3. Interfața aplicației ............................................... pag. 7-9
4. Funcțiile de modificare și ștergere .................... pag. 9-15
5. Funcția de adăugare ....................................... pag. 16-17
6. Funcțiile de vizualizare și modificare profil ..... pag. 17-20
7. Concluzii ............................................................... pag. 20
8. Bibliografie ........................................................... pag. 21

Game Zone

1. Structura aplicației și informații generale

1.1 Cerința proiectului și tema alocată

Creați două aplicații care să conțină o bază de date creată în sistemul de gestiune a bazelor de date MySQL și două interfețe la aceasta (baza de date este comună). La crearea interfețelor se vor folosi două tehnologii (la alegere - ex.: PHP, JSP, Hibernate, JPA, .NET etc.).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Număr temă** | **Grupă** | **Nume, prenume** | **Tabela 1** | **Tip asociere** | **Tabela 2** |
| 26 | 433A | Tițu Alex-Costin | Jocuri | M:N | Jucători |

1.2 Descrierea tehnologiilor utilizate

În realizarea aplicației de baze de date am utilizat un **sistem de gestiune al bazelor de date relațional**, mai exact **MySQL**, împreună cu **Java Server Pages (JSP)** pentru a face legătura între interfața accesibilă utilizatorului și baza de date în sine.

Un **sistem de gestiune al bazelor de date** este un sistem computerizat de menținere a evidenței uneia numite activități, folosind baze de date**. O bază de date relațională** este compus dintr-o mulțime finită de relații ce sunt reprezentate sub formă de tabele. O relație este unică într-o bază de date și se definește prin intermediul atributelor sale (coloanele tabelei). Fiecare atribut reprezintă tipul de entitate sau asociere e care îl descrie relația respectivă. Acestea au o denumire și un domeniu de definiție (tip de dată). Atributele iau o singură valoare (scalare).[1]

Fiecare relație are o cheie primară (PK) drept atribut, cheie unică și ireductibilă ce diferențiază fiecare element al relației de toate celelalte. O cheie primară poate avea mai multe chei candidat (superchei ce pot fi folosite drept cheie primară). O supercheie este o submulțime de atribute ale unei relații care prezintă proprietatea de unicitate. Prin urmare, o cheie primară poate fi naturală ( atribut simplu sau compus al relației cu proprietățile de mai sus, în mod natural. Ex: CNP-ul) sau artificială (atribut, de obicei simplu, ce nu reprezintă o proprietate a tipului de entitate).[2]

Pentru a face legături între relații se utilizează o cheie străină (Foreign Keys), ce constă în introducerea cheii private a tabelei ce află în relație de 1 drept atribut în tabela ce se află în relație de M (one-to-many). Pentru a face o legătură many to many se utilizează o tabelă ce conține ambele chei private ale celor două tabele ce se află în această relație. Foarte important este ca atributul reprezentat de cheia străină și cheia primară pe care o descrie să aibă domenii compatibile (ex: ambele să fie unsigned BIGINT dacă PK este astfel).

Baza de date relațională este vizualizată și modificată prin interogări de tipul SELECT, INSERT, UPDATE și DELETE. Pentru a putea vizualiza într-un mod practic o relație de legătură dintr-o relaționare many-to-many se utiliează funcția de inner join pentru vizualizarea și a datelor referite.

Pentru menținerea integrității referențiale am ales ștergerea și actualizarea în cascadă atunci când un joc/jucător este eliminat/modificat din baza de date. Astfel, orice alte elemente ce conțin id-ul cheii străine identic cu id-ul cheii primare obiectului ce urmează să fie șters/modificat, va fi și acestea eliminat/modificat în baza de date.

Integritatea referențială este proprietatea bazei de date prin care orice cheie străină fie are o valoare care se regăsește printre valorile cheii candidate referite (străine), fie are valoarea NULL.[3]

Pentru realizarea interfeței dintre utilizator și bază de date s-a utilizat tehnologia **Java (sau Jakarta) Server Pages (JSP)**, o tehnologie de programare înclinată către partea de server ce permite conceperea de aplicații Web platform-independent și dinamice.[4] Prin urmare, cu ajutorul acestei tehnologii, pot fi accesate și prelucrate date în mod dinamic în interiorul aplicației Web dezvoltate. Această tehnologie se folosește de un fișier Java tip class în care sunt definite toate operațiile ce pot fi făcute asupra bazei de date (connect, disconnect, select, update etc.) denumit JavaBeans (Model). Din acest fișier vor fi instanțiat funcțiile în interiorul paginilor Web scrise în tehnologia JSP pentru a oferi dinamicitatea percizată anterior. Fișierele JSP sunt o îmbinare dintre Servlet-uri (operații pe parte de server) și Pagina Web (View) respectivă. Pentru a implementa și rula fișiere JSP este necesar un server web compatibil, cu un servlet container, precum Apache Tomcat (în cazul meu).[5]

Mai jos voi prelua o imagine sugestivă despre cum funcționează tehnologia descrisă anterior:

A screenshot of a computer

Description automatically generated with low confidence

**Fig-1.0 „The JSP Model 2 architecture” (Source: Wikipedia)[6]**

Partea de design al interfeței aplicației (front-end) constă în elemente de HTML, CSS, Bootstrap, JavaScript.

Pentru realizarea interfeței site-ului m-am folosit și de aplicația **Bootstrap Studio** ce implementează framework-ul open-source **Bootstrap**, bazat pe template-uri bazate pe **CSS** și **JavaScript** pentru tipografie, formulare, butoane, navigație, și alte componente de interfațare.[7]

**Cascading Style Sheets (CSS)** este un limbaj style sheet (pagini stilizate) utilizat pentru descriere și formatarea prezentării (design-ul) unei pagini scrise într-un limbaj de tip markup (de marcare) precum **HTML**. [8]

**HyperText Markup Language**, sau **HTML** este limbajul markup standard pentru documentele concepute pentru a fi deschise într-un browser web. Scopul limbajului este mai degrabă prezentarea informațiilor decât descrierea semanticii documentului. În cadrul dezvoltării web de tip front-end, HTML este utilizat împreună cu CSS și JavaScript.[9]

2. Descrierea aplicației

Game Zone reprezintă un site ce permite înregistrarea datelor personale a mai multor utilizatori, datele jocurilor video pe care aceștia le joacă și profilul utilizatorilor în cadrul jocului respectiv. Acest lucru le permite utilizatorilor să își compare rezultatele și performanța atinsă față de mulți alți utilizatori ai platformei.

Toate informațiile sunt stocate într-o bază de date realizată în tehnologie MySQL denumită „titualex” de structura următoare:

A picture containing graphical user interface

Description automatically generated

**Fig-1.1 Diagrama logică a bazei de date generată în MySQL**

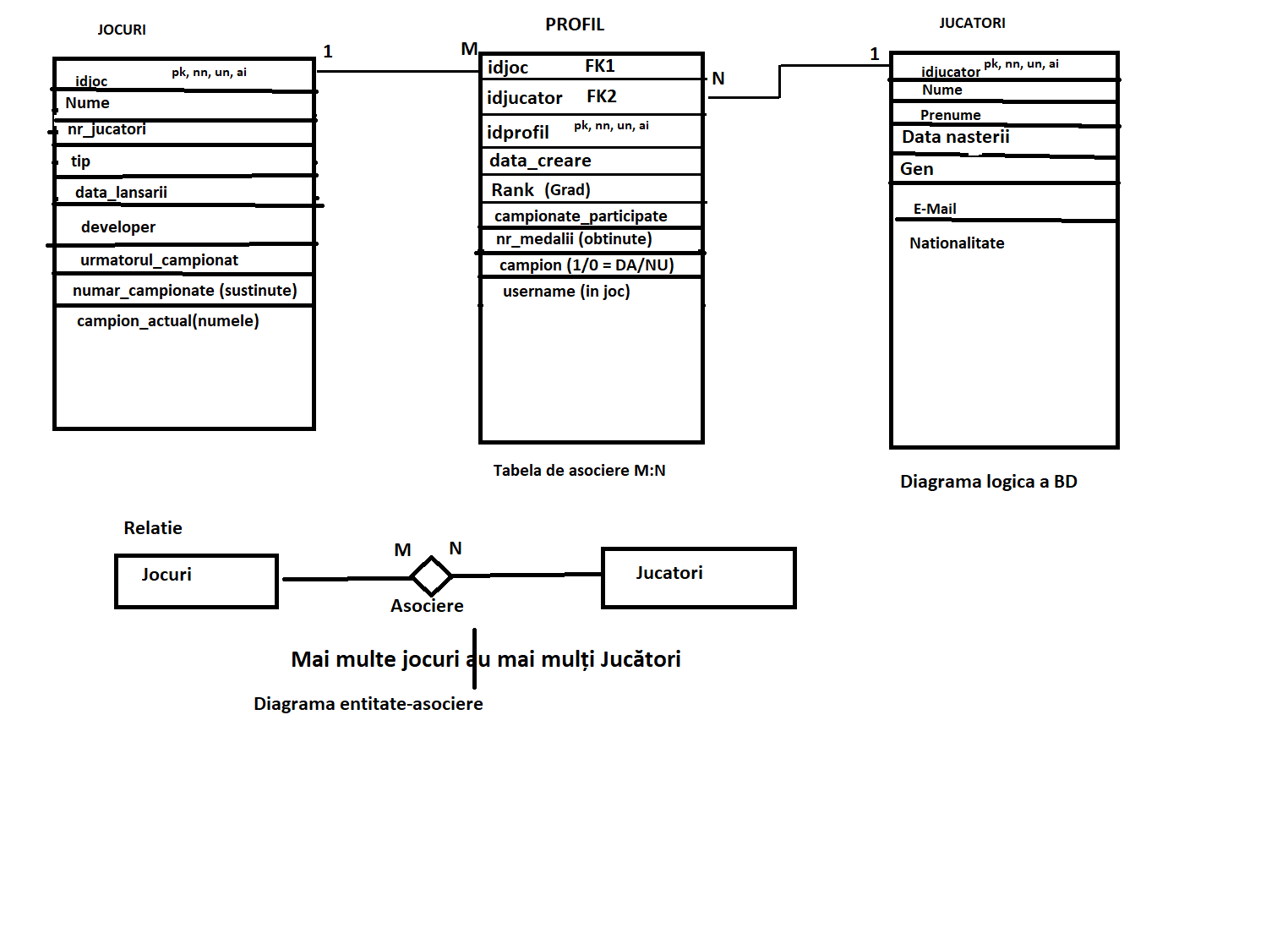
În cadrul tabelei jocuri se va stoca numele jocului, numărul de jucători, tipul jocului, data lansării și cine a produs jocul. Pe lângă aceste informații se vor mai stoca informațiile utile despre campionate: când va avea loc următorul, câte s-au ținut și cine este ultimul campion.

În tabela jucători se stochează informațiile personale ale jucătorilor precum numele, prenumele, email-ul, data nașterii, genul și naționalitatea.

Numele tabelelor sunt date sugestiv, iar relația dintre *jocuri și jucători* este many to many sau M:N (mai mulți jucători, pot juca mai multe jocuri). Prin urmare, pentru a realiza această legătură am utilizat tabela profil ce permite ca fiecare utilizator să aibă mai multe profiluri la mai multe jocuri și invers, în cazul jocurilor.

În tabela profil se păstrează id-ul jocului pe care jucătorul respectiv îl joacă și id-ul jucătorului (pentru relația M:N), data la care a fost creat profilul, gradul (rank-ul) în jocul respectiv, dacă deține un astfel de sistem jocul și numele de utilizator din joc. Pe lângă aceste informații, mai sunt memorate numărul de campionate la care a participat, numărul de medalii obținute și dacă este campion al ultimului campionat ținut.

Mai departe, voi prezenta o schiță a ideii de la care am plecat în implementarea bazei de date.



**Fig-1.2 Schițarea bazei de date**

Au fost alese elemente ce sunt obligatorii de completat la fiecare intrare nouă în baza de date (element NOT NULL) pentru a menține un minim de informație oferit utilizatorilor despre jocul, jucătorul sau profilul respectiv. Acestea sunt notate cum rombul albastru plin în diagrama generată de aplicația MySQL.

Elementele notate cu romb roșu gol reprezintă o cheie străină (în tabela profil), iar celalalte elemente notate cu romb albastru gol reprezintă datele, fără vreo constrângere.

În dreapta fiecărui element, este scris și domeniul pe care este definit ca tip de dată.

**VARCHAR(45)** reprezintă un șir de lungime de maxim 45 de caractere. **BIGINT** reprezintă timpul de date întreg definit pe 8 octeți. **DATE** stochează formatul de dată YYYY-MM-DD (**Y**ear, **M**onth, **D**ay). **TINYINT** stochează doar valoarea 0 sau 1, precum un tip de date Boolean.

Toate cheile primare sunt definite ca BIGINT, NN (nenule), UN (fără semn) și AI (autoincrementare).

Cheile străine sunt definite doar ca fiind BIGINT și UN, pentru a coincide cu domeniul de definiție al cheilor primare pe care le reprezintă. Ele fiind chei străine ce trebuie să corespundă altor intrări în tabela jocuri sau jucători, nu trebuie autoincrementate sau definite ca nenule, pentru a se putea alege exact ce joc și jucător reprezintă profilul respectiv. Mai jos voi expune diagrama UML a aplicației realizată în tehnologia JSP.

Graphical user interface, text, application, email

Description automatically generated

**Fig-1.3 Diagrama UML a proiectului realizat în tehnologie JSP**

3. Interfața aplicației

Interfața are ca pagină de pornire o scurtă prezentare a scopului site-ului împreună cu câteva imagini sugestive. Navigația se face fie din partea de sus, fie din partea de jos a paginii.

Graphical user interface, application, Teams

Description automatically generated

**Fig-2.0 Pagina principală: Navigație și prezentare (partea superioară)**

Graphical user interface

Description automatically generated

**Fig-2.1 Pagina principală: Navigație și prezentare (partea inferioară)**

În acest mod se va naviga pe tot cuprinsul site-ului: cu ajutorul celor două zone împreună cu alte butoane ajutătoare. Prin apăsarea butoanelor Games, Gamers, Profiles, se va putea vizualiza una din cele trei tabele menționate anterior, respectiv jocuri, jucători și profiluri. Prin click pe drop-down-ul Register, se poate alege unul din cele trei formulare de înregistrare de noi date în cele trei tabele. Prin apăsarea logo-ului din partea stângă, sus, în orice moment al utilizării aplicației, se revine la pagina principală.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig-2.2 Navigația din partea superioară (nav) a paginii împreună cu textul și imaginile sugestive**

A computer screen capture

Description automatically generated with medium confidence

**Fig-2.3 Navigația din partea inferioară (footer) a paginii împreună cu textul sugestiv**

Codurile de mai sus reprezintă elementele de navigație scrisă în tehnologiile front-end menționate.

1. Funcțiile de modificare și ștergere

Mai departe voi prezenta afișarea jocurilor înregistrate pe platformă, împreună cu modificarea și ștergerea de jocuri din baza de date. Similar, va funcționa și în cazul jucătorilor, fiind aduse doar mici modificări de interfață. Aceste operațiuni sunt SELECT, UPDATE și DELETE în MySQL.

A screenshot of a video game

Description automatically generated

**Fig-3.0 Afișarea datelor din tabela jocuri, comenzile de ștergere/modificare**

Pentru a modifica o linie trebuie mai întâi bifată opțiunea de modificare (lucru care dezactivează automat opțiunea de ștergere) și apăsat butonul „Modifica datele” pentru a trece la selectarea unei linii din tabel în scopul modificării.

Pentru a șterge o linie trebuie mai întâi bifată opțiunea de Ștergere (lucru care dezactivează automat opțiunea de modificare) și apăsat butonul „Șterge liniile marcate” pentru a șterge liniile ce au fost selectate în cadrul paginii curente.

Text

Description automatically generated

**Fig-3.1 Funcția de ascundere a butoanelor de ștergere, modificare**

Dacă nu este selectată nicio linie (atât la modificare, cât și la ștergere), la apăsarea butonului, utilizatorul este redirecționat la o pagină unde este avertizat că este obligatorie selectarea a cel puțin unei linii pentru a face operațiunea respectivă.

În cazul modificărilor liniilor din tabel, trebuie selectată numai și numai o linie din tabela respectivă, altfel pagina va oferi spre editare, doar prima linie ce apare bifată în tabel în urma căutării bifării/bifărilor efectuate.

A screenshot of a computer

Description automatically generated with medium confidence

**Fig-3.2 Avertizare obținută în urma executării operațiunii de ștergere fără a bifa cel puțin o linie.**

A screenshot of a video game

Description automatically generated

**Fig-3.3 Alegerea liniei 4 spre editare, după apăsarea butonului „Modifica datele”**

Graphical user interface

Description automatically generated

**Fig-3.4 Interfața de editare a liniei selectate din tabel. (cu modificări efectuate)**

Campionul actual se poate selecta doar dintre utilizatorii înregistrați în platformă. Aceștia vor fi înregistrați în tabelă cu numele și prenumele. În cazul în care există linii necompletate (blank), site-ul va pune „N/A”, în cazul textelor, „0” în cazul numerelor și „0000-0-0” în cazul unei date. (în cazul tuturor paginilor de modificare de date se procedează la fel)

Graphical user interface, website

Description automatically generated

**Fig-3.5 Mesajul ce oferă rezultatul operației de editare.**

Atât în cazul ștergerii cât și în cazul modificării liniilor din tabel, se vor oferi în urma operațiilor rezultatul acestora sub forma unui mesaj precum cel de mai sus.

Mai jos va fi prezentat codul aferent operațiunilor de modificare și ștergere pentru tabela jocuri(aproape identic fiind și la jucători).

Se observă în următoarele imagini reprezentarea sub formă tabelară a informațiilor și utilizarea elementelor de JSP pentru afișarea informațiilor conținute în baza de date cu ajutorul Java Bean. De asemenea apare și acțiunea de selectare pentru modificare sau ștergere a unei linii din tabel, sub formă de form ce va apela la rândul lui alt fișier JSP pentru efectuarea operației.

De fiecare dată când sunt efectuate operații asupra bazei de date, programul se va conecta la aceasta prin intermediul funcției connect() din Java Bean, iar apoi se va deconecta de la aceasta cu ajutorul funcției disconnect().

Text

Description automatically generated

**Fig-3.6 Selectare liniei/liniilor spre ștergere și afișarea tabelei (similar cu modificare)**

Text

Description automatically generated

**Fig-3.7.1 Interfața de editare a liniei selectate din tabela jocuri**

Text

Description automatically generated

**Fig-3.7.2 Interfața de editare a liniei selectate din tabela jocuri**

Text

Description automatically generated

**Fig-3.8.1 Efectuarea modificărilor în baza de date**

Au fost puse condiții pentru situațiile în care utilizatorul lasă din greșeală formularul necompletat. Astfel aplicația va pune valori prestabilite în aceste situații pentru a preveni apariția erorilor.

Text

Description automatically generated

**Fig-3.8.2 Trimiterea modificărilor către baza de date**

Text

Description automatically generated

**Fig-3.9.1 Ștergerea liniei din baza de date**

Text

Description automatically generated

**Fig-3.9.2 Trimiterea comenzii de ștergere către baza de date**

Ultimele 4 imagini prezintă contactarea la baza de date în privința efectuării operației ce a fost inițiată (ștergere/modificare) și compunerea instanție MySQL ce va spune sub ce formă să se efectueze modificarea.

1. Funcția de adăugare

Mai departe voi prezenta funcționalitatea de **adăugare** de noi elemente la baza de date, mai exact de adăugare a unui jucător nou. (similar și în cazul jocului)

Din cele două drop-downs se pot alege genul (masculin, feminin sau altul) și țara (dintr-o listă ce conține toate țările).

Graphical user interface, application

Description automatically generated

**Fig-3.1 Interfața de adăugare a unui jucător nou**

Text

Description automatically generated

**Fig-3.2 Adăugarea unui nou jucător**

Text

Description automatically generated

**Fig-3.3 Trimiterea comenzilor de adăugare de jocuri sau jucători**

Din nou, sunt verificate înainte de efectuarea adăugării unui element că au fost completate toate câmpurile. Dacă acestea au date lipsă, aplicația va pune de la sine date predefinite. Ultima imagine prezintă compunerea instanțelor MySQL.

6. Funcțiile de vizualizare și modificare profil

Vizualizarea profilurilor înregistrate de utilizatori, alături de datele acestora și datele jocului pe care îl joacă se face cu ajutorul operației de inner join în baza de date MySQL.

Coloana campion, ce reprezintă dacă utilizatorul este ultimul câștigător al ultimului campionat ținut pentru jocul respectiv este reprezentat de o valoarea Boolean (Adevărat = 1 iar Fals = 0). Prin urmare, 1 va reprezenta că este campion, iar 0 că nu este campion.

Liniile de cod pentru interfața aplicației și efectuarea operațiilor sunt similare cu cele prezentate anterior, singura diferență fiind apariția cheilor străine și împrumutarea elementelor din tabelele legate la tabela profil, precum numele și prenumele jucătorului, numele jocului etc.

Graphical user interface

Description automatically generated

**Fig-4.1.1 Vizualizarea Profilurilor înregistrare**

Text

Description automatically generated

**Fig-4.1.2 Vizualizarea Profilurilor înregistrare**

Graphical user interface, application, Teams

Description automatically generated

**Fig-4.2.1 Modificarea Profilului aferent liniei selectate**

Text

Description automatically generated

**Fig-4.2.2 Modificarea Profilului (1)**

Text

Description automatically generated

**Fig-4.2.3 Modificarea Profilului (2)**

Text

Description automatically generated

**Fig-4.2.3 Comunicarea spre adăugarea unui Profil nou**

Funcția utilizată pentru a trimite un statement de modificare a tabelei profil este aceeași cu funcția utilizată și în cazul modificărilor tabelelor jocuri sau jucători.

Preluarea datelor din tabelele legate la tabela profil se face sub forma de result sets (seturi de rezultate) prin interogarea acestora cu ajutorul funcției vedeTabela().

1. Concluzii

Realizarea unei aplicații web în tehnologie JSP reprezintă o alternativă foarte practică pentru proiecte scurte și rapide, sau proiecte de mici dimensiuni. De asemenea ar putea fi un bun punct de start pentru a pune rapid în practică o idee, fără a complica prea mult conceptul funcționalității aplicației. Acest lucru se datorează implementării funcționalității dinamice a aplicației mai mult în fișierele web JSP ale aplicației, decât în clase Java (ex.: Hibernate).

MySQL este un SGBD intuitiv cum multiple funcționalități, ce poate fi utilizat atât în proiecte de scară mică, dar și mare.

8. Bibliografie

1. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
2. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
3. *Prezentare Curs 2 „Baze de date relaţionale” -* S.l. Dr. Ing. Valentin Pupezescu – anul universitar 2021-2022
4. *https://en.wikipedia.org/wiki/Jakarta\_Server\_Pages*
5. *https://en.wikipedia.org/wiki/Jakarta\_Server\_Pages*

1. *<https://en.wikipedia.org/wiki/Jakarta_Server_Pages#/media/File:JSP_Model_2.svg>* - Fig-1.6 „The JSP Model 2 architecture”
2. *https://en.wikipedia.org/wiki/Bootstrap\_(front-end\_framework)*
3. *https://en.wikipedia.org/wiki/CSS*
4. *https://en.wikipedia.org/wiki/HTML*