**A circular logo with text and numbers

Description automatically generatedUNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE POLITEHNICA BUCUREȘTI**

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Proiect (2)**

**Programarea Interfețelor pentru Baze de Date**

Student: Năstase Maria-Magdalena

Grupa: 434D

Profesor coordonator

Ș.l. Dr. Ing. Pupezescu Valentin

București 2025

Cuprins

1. [Cerințe 3](#_Toc188912620)

[2.Tehnologii utilizate 3](#_Toc188912804)

[***2.1*** ***MySQL*** 3](#_Toc188912805)

[***2.2*** ***Pyhton*** 3](#_Toc188912806)

[***2.3*** ***Python Flask*** 4](#_Toc188912807)

[3.Descrierea aplicației 5](#_Toc188912808)

[***3.1*** ***Baza de date*** 5](#_Toc188912809)

[***3.2*** ***Diagrama ERD*** 5](#_Toc188912809)

[***3.3*** ***Codul*** 7](#_Toc188909945)

*3.3.1* ***Conectarea la baza de date ...................................................................................................................7***

***3.3.2 Gestionarea tabelelor..............................................................................................................................7***

[4. Interfața TK 1](#_Toc188909946)0

**5. Concluzii**...............................................................................................................................16

**1.Cerințe**

Să se creeze 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 (ex: JSP, Hibernate, JPA, .NET, Python etc.). Baza de date va fi compusă din tabelele: Sculptures și Sculptors, cu asociere de tip M:N.

Interfețele realizate trebuie să permită utilizatorului să execute următoarele operații pe toate tabelele (CRUD): vizualizare, adăugare, modificare și ștergere de date. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celalte tabele.

Pentru tema primită, se aleg interfețe diferite: tkinter si flask din Python. În prezentul document, se va face referire la tehnologia JSP (Flask).

## **2.Tehnologii utilizate**

### ***MySQL***

MySQL este un sistem de gestiune 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.

### ***Pyhton***

Python este un limbaj de programare foarte popular, oferind posibilitatea programării structurate, dar și orientate pe obiect, și incluzând și elemente din paradigma funcțională. Este un limbaj de scripting, ceea ce înseamnă că este interpretat și nu compilat, economisind mult timp în procesul de dezvoltare și debugging. Limbajul oferă o multitudine de funcționalități într-o sintaxă ușor de învățat. Are module, clase, excepții, tipuri dinamice și garbage collection. API-ul Python oferă module pentru o gamă foarte mare de funcționalități, de la cele de bază pentru lucrul cu șiruri și fișiere, până la cele pentru lucrul cu procese, threaduri, sockets, serializare etc. Multe dintre aceste module oferă o interfață foarte similară programării la nivel de sistem din C, astfel încât funcțiile Python au același nume si aproximativ aceiași parametrii ca cei pentru funcțiile din C pentru apeluri de sistem.[[1]](#footnote-1)

### ***Python Flask***

Flask este un framework web micro dezvoltat pentru Python, cunoscut pentru simplitatea și flexibilitatea sa. Este utilizat frecvent pentru a crea aplicații web rapide și eficiente, fiind o alegere populară în comunitatea Python. Flask este un framework minimalist, ceea ce înseamnă că oferă doar funcționalitățile esențiale pentru dezvoltarea unei aplicații web, permițând dezvoltatorilor să extindă funcționalitatea cu biblioteci și module externe.

Flask este implementat pe baza bibliotecii Werkzeug pentru gestionarea cererilor HTTP și a bibliotecii Jinja2 pentru procesarea șabloanelor HTML. Această combinație permite dezvoltatorilor să creeze aplicații web dinamice, unde logica backend poate fi conectată ușor cu interfața frontend.

Flask urmează o arhitectură simplă, care oferă flexibilitate maximă. Spre deosebire de alte framework-uri monolitice, Flask permite integrarea și utilizarea diverselor extensii pentru baze de date, autentificare, migrarea schemelor și multe altele. Acest lucru face ca Flask să fie ideal pentru proiecte mici și medii, dar și pentru prototipuri rapide.

Deoarece Flask nu impune structuri rigide sau configurații complexe, este o soluție ușor de utilizat pentru dezvoltatorii care doresc să construiască aplicații web personalizate într-un timp scurt.

## **3.Descrierea aplicației**

### ***Baza de date***

Tema dată presupune asocierea M:N a două tabele, „Sculptures” și „Sculptors”.

Asocierea M:N (Many-to-Many) are ca și caracteristică faptul că fiecărui element înregistrat într-o tabelă îi pot fi asociate mai multe elemente din cealaltă tabelă și invers.

De exemplu, în acest caz, un sculptor poate fi asociat mai multor sculpturi, așa cum și unei sculpturi îi pot fi asociate mai mulți sculptori.

Aplicația are ca scop principal administrarea datelor legate de:

* **Sculptori**: informații precum numele, prenumele, anul nașterii, anul decesului, naționalitatea și stilul artistic.
* **Sculpturi**: titlu, material, înălțime, greutate, anul creației și locația muzeului.
* **Asocieri** între sculptori și sculpturi.

O imagine care conține text, captură de ecran, număr, Font

Descriere generată automat

Figura 1. Tabela Sculptors

O imagine care conține text, software, număr, Font

Descriere generată automat

Figura 2. Tabela Sculptures

Pentru o asociere M:N este necesară crearea unei relații de legătură, în acest caz, „Sculpture\_Sculptor”, o treia tabelă de joncțiune. Scopul acesteia este de a diviza asocierea many-to-many în două asocieri de tip 1:N.

În această nouă tabelă, atributele ce au fost selectate drept chei primare pentru tabelele de mai sus, vor deveni chei străine pentru tabela de legătură.

O imagine care conține text, captură de ecran, Font, linie

Descriere generată automat

Figura 3. Tabela Sculpture\_Sculptor

***3.2 Diagrama ERD***

O imagine care conține text, captură de ecran, diagramă, Font

Descriere generată automatDiagrama ERD oferă o reprezentare vizuală complexă care ilustrează entitățile, relațiile și constrângerile acestora, oferind o perspectivă detaliată a structurii și interconexiunilor din cadrul bazei de date.

Asocierile dintre tabele:

* Între „Sculptors” și „Sculptures” este o asociere M:N;
* Între „Sculpture\_Sculptor” și „Sculptures” este o asociere 1:N;
* Între „Sculptors” și „Sculpture\_Sculptor” este o asociere 1:N.

### ***3.3 Codul***

Conectarea la o bază de date MySQL în Python poate fi realizată folosind diverse module, cum ar fi mysql-connector-python, pymysql sau sqlalchemy. Aceste module facilitează interacțiunea cu baza de date MySQL prin intermediul Python, permițând accesul la date și manipularea acestora.

În paralel, dezvoltarea interfețelor grafice în Python se poate realiza cu ajutorul bibliotecii Tkinter, care oferă un set de instrumente pentru construirea aplicațiilor cu interfață grafică.

**3.3.1 Conectarea la baza de date**

* Funcția connect\_to\_database() este utilizată pentru stabilirea conexiunii cu baza de date MySQL.
* Mesajele de eroare sunt afișate utilizatorului în cazul în care conexiunea eșuează.

O imagine care conține text, captură de ecran, Font

Descriere generată automat

* + 1. **Gestionarea tabelelor**

**Gestionare sculptori:**

* **Adăugare**: add\_sculptor() permite utilizatorului să introducă un sculptor nou.
* **Editare**: edit\_sculptor() oferă funcționalitatea de a modifica detalii despre un sculptor existent.
* **Ștergere**: delete\_sculptor() permite eliminarea unui sculptor selectat din bază.

**Gestionare sculpturi:**

* Funcțiile sunt similare celor pentru sculptori: add\_sculpture(), edit\_sculpture() și delete\_sculpture().

**Gestionare asocierile dintre sculptori și sculpturi:**

* **Adăugare**: add\_association() permite crearea unei noi asocieri între un sculptor și o sculptură.
* **Editare**: edit\_association() modifică o asociere existentă.
* **Ștergere**: delete\_association() șterge relația selectată.

O imagine care conține text, captură de ecran

Descriere generată automat

Funcția add\_sculptor() din Flask procesează cererile POST pentru a adăuga un sculptor în baza de date. Datele trimise prin formular sunt preluate, validate, și inserate în tabela Sculptors folosind o interogare SQL. În caz de succes, modificările sunt salvate, iar în caz de eroare, se returnează un mesaj JSON cu detalii.

O imagine care conține text, captură de ecran

Descriere generată automat Funcția delete\_sculptor() șterge un sculptor din baza de date, verifică sculpturile asociate acestuia și elimină acele sculpturi care nu mai sunt legate de alți sculptori. După ștergere, reindexează ID-urile în tabela Sculptors pentru a menține ordinea.

O imagine care conține captură de ecran, text

Descriere generată automat

Funcția edit\_sculptor() actualizează informațiile unui sculptor specificat prin ID. Datele sunt primite printr-un formular (request.form) și utilizate într-o interogare SQL UPDATE pentru a modifica înregistrarea în tabela Sculptors. După actualizare, modificările sunt salvate în baza de date.

**Funcții similare sunt folosite și pentru tabelele Sculptures și Sculpture-Sculptors.**

* 1. **Interfața Flask**
  2. O imagine care conține text, captură de ecran, software

     Descriere generată automat**Afișarea datelor**

Fragmentul care permite afișarea și interacțiunea cu datele referitoare la sculptori este funcția get\_sculptors(), împreună cu un tabel HTML generat dinamic. Aceasta preia datele din tabela Sculptors și le afișează într-o pagină web, permițând utilizatorilor să vizualizeze informațiile și să interacționeze cu acestea pentru operațiuni suplimentare, cum ar fi ștergerea sau modificarea.

* Afișarea datelor: Funcția get\_sculptors() execută o interogare SQL (SELECT \* FROM Sculptors) și construiește un tabel HTML în care sunt afișate datele preluate, precum ID-ul sculptorului, numele, prenumele, anul nașterii, anul decesului, naționalitatea și stilul artistic.
* Interacțiunea cu datele: Fiecare rând din tabel include butoane pentru operațiuni precum Modifică și Șterge, care permit utilizatorului să editeze sau să șteargă înregistrarea selectată prin funcțiile respective, bazându-se pe ID-ul sculptorului.

Această implementare oferă o interfață ușor de utilizat pentru gestionarea datelor sculptorilor din baza de date.

**O imagine care conține statuie, sculptură, Sculptură in piatră, artă

Descriere generată automat**

Pagina Home

**O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat**

Pagina sculptori

**4.2 Adăugarea datelor**

**O imagine care conține text, captură de ecran, software

Descriere generată automat**

**O imagine care conține text, captură de ecran, Font, proiectare

Descriere generată automat**

Fragmentul de cod reprezintă logica pentru adăugarea unui sculptor în baza de date prin intermediul unei aplicații Flask. Mai exact:

1. Preluarea datelor:
   * Datele introduse de utilizator în formularul HTML sunt colectate folosind request.form și stocate în variabila data.
2. Construirea interogării SQL:
   * O comandă INSERT INTO Sculptors este pregătită pentru a adăuga datele noi în baza de date, specificând coloanele (Nume, Prenume, AnNastere, etc.) și valorile corespunzătoare.
3. Executarea și confirmarea:
   * Interogarea SQL este executată folosind cursorul bazei de date.
   * După executare, modificările sunt salvate cu connection.commit().
   * Dacă adăugarea are succes, utilizatorului i se returnează o pagină HTML care confirmă adăugarea sculptorului.
4. Gestionarea erorilor:
   * Dacă conexiunea la baza de date nu poate fi stabilită, se returnează un mesaj JSON de eroare cu codul HTTP 500.

Această funcție este responsabilă de gestionarea logicii backend pentru adăugarea unui sculptor în tabela Sculptors și asigură informarea utilizatorului cu privire la succesul operațiunii.

**4.3 Modificarea datelor**

**O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat** Modificarea datelor se realizează prin selectarea unei înregistrări și apăsarea butonului de **Modifică** din interfață. Fragmentul de cod care preia datele din câmpurile introduse de utilizator și le utilizează pentru actualizarea sculptorului selectat în baza de date este ilustrat mai sus.

**4.4 Ștergerea datelor**

**O imagine care conține text, Font, captură de ecran, siglă

Descriere generată automat** Ștergerea datelor se realizează prin selectarea liniei și apăsarea butonului de Șterge din pagină.

**5. Concluzii**

Proiectul dezvoltat utilizează tehnologiile MySQL și Python Flask pentru a crea o aplicație web robustă și ușor de utilizat pentru gestionarea datelor despre sculptori, sculpturi și asocierile dintre acestea. Cu ajutorul acestei aplicații, utilizatorii pot administra eficient baza de date, având posibilitatea de a efectua operațiuni de creare, citire, actualizare și ștergere a informațiilor.

Interfața web dezvoltată cu Flask oferă utilizatorilor o experiență intuitivă și accesibilă. Designul simplu și structura flexibilă permit navigarea facilă și interacțiunea directă cu datele stocate în baza de date MySQL, prin intermediul unor formulare și tabele HTML dinamice.

Prin intermediul operațiunilor CRUD (creare, citire, actualizare, ștergere), utilizatorii pot gestiona cu ușurință informațiile din baza de date, având posibilitatea de a adăuga noi înregistrări, de a vizualiza și edita datele existente, precum și de a șterge informațiile care nu mai sunt necesare.

Unul dintre avantajele majore ale acestei aplicații constă în utilitatea sa pentru organizarea și administrarea informațiilor despre sculptori și sculpturi. Aplicația oferă o platformă centralizată pentru gestionarea acestor date, facilitând procesul de evidență și monitorizare, contribuind astfel la eficientizarea activităților administrative.

În concluzie, prin combinarea tehnologiilor MySQL și Python Flask, proiectul oferă o soluție completă și simplă pentru gestionarea informațiilor despre sculptori și sculpturi. Interfața intuitivă și funcționalitatea extinsă fac din această aplicație un instrument esențial pentru utilizatori în administrarea datelor.

**Bibliografie**

[**https://flask.palletsprojects.com/en/stable/**](https://flask.palletsprojects.com/en/stable/)

[**https://www.geeksforgeeks.org/flask-tutorial/**](https://www.geeksforgeeks.org/flask-tutorial/)

[**https://www.restack.io/p/flask-docs-quickstart**](https://www.restack.io/p/flask-docs-quickstart)

[**https://medium.com/@knowledgelibary/python-flask-documentation-760b718d4246**](https://medium.com/@knowledgelibary/python-flask-documentation-760b718d4246)

1. https://ocw.cs.pub.ro/courses/asc/laboratoare/01 [↑](#footnote-ref-1)