

---

---

PROIECT REALIZAT PENTRU DISCIPLINA

**Proiect 3**

---

---

*Universitatea Națională de Știință și Tehnologie Politehnică București*  
*Facultatea de Electronică, Telecomunicații și Tehnologia Informației*

LIBRARY MANAGEMENT SYSTEM

STUDENT

ADRIAN-GABRIEL DIACONU

**Profesor coordonator**

**Ș. I. Dr. Ing. Pupezescu Valentin**

2023 - 2024

---

## Contents

<b>1</b>	<b>Introducere</b>	<b>3</b>
1.1	Motivație . . . . .	3
1.2	Obiective . . . . .	3
1.3	Aplicabilitate . . . . .	3
<b>2</b>	<b>Tehnologii folosite</b>	<b>4</b>
2.1	Python . . . . .	4
2.2	Sistemul de gestiune a bazelor de date MySQL . . . . .	5
2.3	HTML . . . . .	5
2.4	Flask . . . . .	5
2.5	Flask-MySQLDB . . . . .	6
2.6	Werkzeug . . . . .	6
2.7	Flask-Login . . . . .	6
2.8	Flask-WTF și WTForms . . . . .	6
2.9	Flask-Talisman . . . . .	6
2.10	Bootstrap . . . . .	6
2.11	Jinja . . . . .	6
<b>3</b>	<b>Metoda Propusa</b>	<b>7</b>
3.1	Creearea bazei de date . . . . .	7
3.2	Pagina principala . . . . .	8
3.3	Autentificarea . . . . .	8
3.4	Vizualizarea Datelor . . . . .	12
3.5	Adăugarea de Date . . . . .	12
3.6	Modificarea datelor . . . . .	15
3.7	Ștergerea Datelor . . . . .	17
3.8	Procesul de rezervare . . . . .	18
<b>4</b>	<b>Concluzii</b>	<b>21</b>
4.1	Concluzii generale . . . . .	21
4.1.1	Dezvoltari ulterioare . . . . .	21

---

## List of Figures

2.1	Topul limbajelor de programare (2019) conform statistica . . . . .	4
2.2	Domeniile de utilizare ale Python (2018) conform cleveroad.com . . . . .	5
3.1	Tabela Authors . . . . .	7
3.2	Tabela Books . . . . .	7
3.3	Tabela Libraries . . . . .	7
3.4	Tabela BookAuthors . . . . .	7
3.5	Tabela BookReservations . . . . .	8
3.6	Tabela User . . . . .	8
3.7	Pagina de intampinare . . . . .	8
3.8	Pagina de Login si mesajul aferent introducerii datelor gresite . . . . .	10
3.9	Pagina de Inregistrare si mesajul aferent introducerii unui utilizator existent . .	11
3.10	Tabelul Books din pagina web . . . . .	12
3.11	Modal pentru adaugare . . . . .	14
3.12	Modalul specific modificarilor . . . . .	16
3.13	Mesajul de confirmare pentru stergerea datelor . . . . .	17
3.14	Mesajul aferent rolului userului pentru stergere . . . . .	18
3.15	Fereastra de unde se pot rezerva cartile . . . . .	18
3.16	Fereastra rezervarilor . . . . .	19

---

# 1 Introducere

## 1.1 Motivație

În epoca digitală, managementul bibliotecilor se confruntă cu provocări unice și dinamice. Proiectul nostru, dezvoltat în cadrul unui program universitar, răspunde la aceste provocări prin crearea unei aplicații inovative pentru gestionarea eficientă a bibliotecilor. Scopul principal al acestei aplicații este de a simplifica și eficientiza procesele de management ale unei biblioteci, transformând modul în care informațiile și resursele sunt accesate și gestionate.

## 1.2 Obiective

Acest proiect își propune următoarele obiective:

- **Simplificarea Proceselor de Gestionare a Bibliotecii:**

Crearea unei interfețe intuitive pentru a ușura procesul de catalogare, căutare și gestionare a cărților și a altor materiale.

- **Automatizarea Sistemului de Împrumut și Rezervare:**

Implementarea unui sistem automatizat pentru gestionarea împrumuturilor și rezervărilor de cărți, reducând astfel timpul necesar pentru aceste operațiuni și minimizând erorile umane.

- **Gestionarea Eficientă a Utilizatorilor:**

Dezvoltarea unui sistem de administrare a conturilor utilizatorilor, inclusiv înregistrare, autentificare și actualizarea profilurilor, pentru a asigura o experiență personalizată și sigură.

- **Securizarea Datelor și Tranzacțiilor:**

Asigurarea securității datelor utilizatorilor și a informațiilor tranzacțiilor prin implementarea unor protocoale de securitate robuste.

## 1.3 Aplicabilitate

În instituțiile academice și universități, aplicația poate fi folosită pentru a gestiona resursele educaționale și de cercetare, oferind studenților și cadrelor didactice un acces ușor la materialele necesare. Aceasta poate sprijini eforturile de digitalizare a bibliotecilor tradiționale, ajutând la transformarea colecțiilor fizice în resurse accesibile digital și la îmbunătățirea accesului la cunoaștere.

De asemenea, aplicația poate fi de mare ajutor pentru comunități și cluburi de carte, facilitând organizarea și gestionarea colecțiilor comune de cărți și sprijinind partajarea și discuțiile despre lecturi. Prin facilitarea accesului la

cărți și materiale de lectură, aplicația contribuie la democratizarea accesului la cunoaștere și la promovarea educației și culturii.

## 2 Tehnologii folosite

### 2.1 Python

Python este un limbaj de programare de înalt nivel, interpretat și orientat pe obiecte, cunoscut pentru sintaxa sa clară și ușor de înțeles. De-a lungul timpului, a devenit unul dintre cele mai populare limbaje de programare, fiind utilizat într-o varietate largă de domenii, de la dezvoltarea web la inteligența artificială. Python promovează citibilitatea codului și simplitatea limbajului. Acest aspect face să fie accesibil pentru programatori de diferite niveluri de experiență, facilitând dezvoltarea rapidă și eficientă a aplicațiilor.

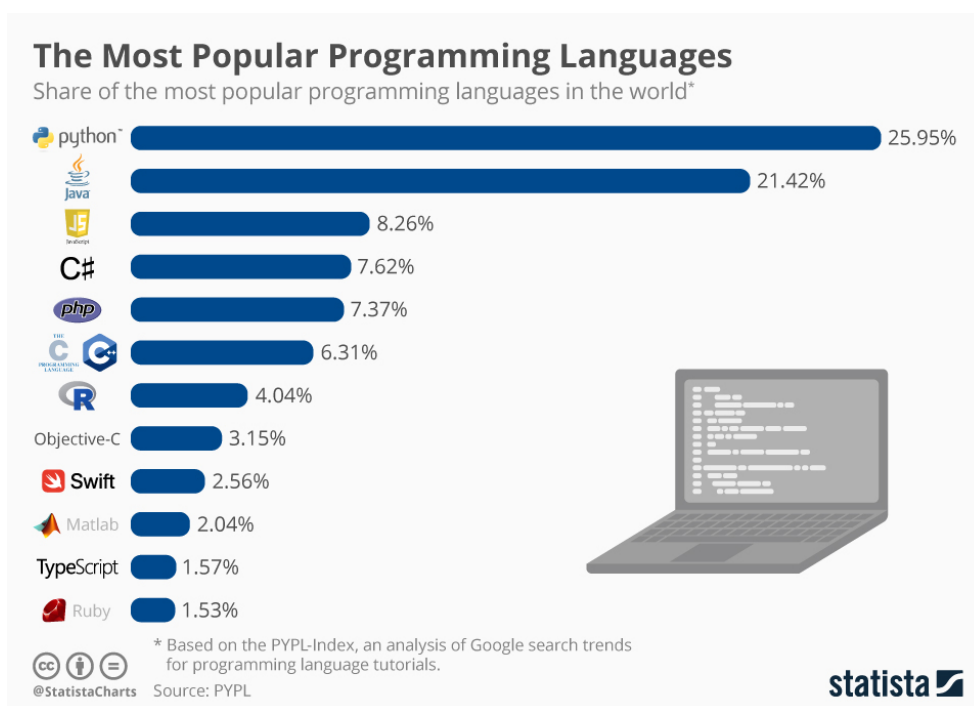


Figure 2.1: Topul limbajelor de programare (2019) conform statista

Unul dintre principalele avantaje ale Python este abundența de biblioteci disponibile. Pentru prelucrarea imaginilor, biblioteci precum OpenCV și Pillow sunt esențiale. În domeniul științei datelor, NumPy, Pandas și Matplotlib sunt printre cele mai utilizate. Aceste biblioteci extind puternic funcționalitățile limbajului și accelerează dezvoltarea de software. Python a devenit limbajul principal în domeniul inteligenței artificiale și machine learning. Framework-uri precum TensorFlow și PyTorch sunt dezvoltate în Python, facilitând construirea, antrenarea și implementarea modelelor de machine learning.

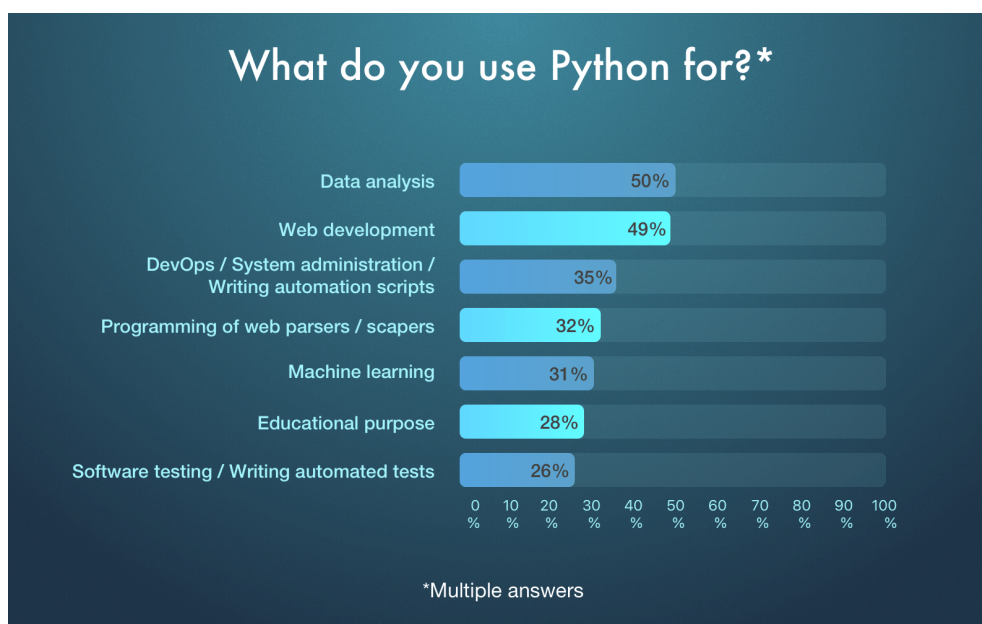


Figure 2.2: Domeniile de utilizare ale Python (2018) conform cleveroad.com

## 2.2 Sistemul de gestiune a bazelor de date MySQL

MySQL este un sistem de gestiune a bazelor de date relaționale (SGBDR), open-source, care folosește Structured Query Language (SQL) pentru gestionarea datelor. Este ales pentru flexibilitatea, fiabilitatea și performanța sa, fiind ideal atât pentru aplicații mici, cât și pentru sisteme mari și complexe. MySQL excelază în aplicații web datorită integrării sale ușoare cu limbaje de programare precum Java, oferind o soluție robustă pentru dezvoltarea web dinamică.

## 2.3 HTML

HTML (HyperText Markup Language) este limba de bază pentru crearea și structurarea paginilor web. Folosind o serie de elemente și atribute, HTML permite dezvoltatorilor să definească structura de conținut, cum ar fi paragrafe, titluri, linkuri și imagini, pe o pagină web. Această structură semantică nu numai că este esențială pentru prezentarea vizuală și organizarea conținutului în browser, dar joacă un rol crucial în accesibilitatea și optimizarea motorului de căutare (SEO). Fiind strâns integrată cu CSS pentru stilizare și JavaScript pentru funcționalități, HTML stă la baza dezvoltării web, permițând crearea de site-uri web interactive și atractive.

## 2.4 Flask

Un framework web ușor și flexibil pentru Python, utilizat pentru a construi aplicația web. Permite dezvoltarea rapidă și ușoară a interfețelor utilizator și a funcționalităților backend.

---

## 2.5 Flask-MySQLDB

O extensie Flask care facilitează interacțiunea între aplicația Flask și baza de date MySQL, asigurând un transfer eficient și sigur al datelor.

## 2.6 Werkzeug

O bibliotecă WSGI pentru Python, utilizată pentru gestionarea cererilor și răspunsurilor HTTP, oferind funcționalități de securitate cum ar fi hashing-ul parolelor.

## 2.7 Flask-Login

O extensie pentru gestionarea sesiunilor de utilizatori în Flask, facilitând procesele de autentificare, menținerea sesiunilor și protejarea rutelor accesibile doar utilizatorilor autentificați.

## 2.8 Flask-WTF și WTForms

Folosite pentru crearea și gestionarea formularelor web, facilitând colectarea și validarea datelor de intrare de la utilizatori într-un mod sigur și eficient.

## 2.9 Flask-Talisman

O extensie Flask pentru securitate, utilizată pentru a impune politici stricte de securitate în conținut, precum și pentru a forța utilizarea HTTPS în loc de HTTP.

## 2.10 Bootstrap

Un framework front-end pentru dezvoltarea unor interfețe web responsive și estetice, folosit pentru a crea un design user-friendly și accesibil pentru aplicație.

## 2.11 Jinja

Un motor de șabloane pentru Python, integrat cu Flask, utilizat pentru a genera paginile HTML dinamice ale aplicației, permițând o separare clară între logica de backend și prezentarea frontend.

### 3 Metoda Propusa

#### 3.1 Creerea bazei de date

Tema se bazează pe crearea unei baze de date ce are:

- Autori

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
AuthorID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
FirstName	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
LastName	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CountryOfOrigin	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 3.1: Tabela Authors

- Carti

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
BookID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Title	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
YearOfPublication	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Publisher	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Quantity	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 3.2: Tabela Books

- Biblioteca

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
LibraryID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Address	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 3.3: Tabela Libraries

Asocierile între tabele sunt următoarele:

- între tabela Autori și tabela Carti - raport de cardinalitate M:N.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
BookAuthorID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
BookID	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
AuthorID	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure 3.4: Tabela BookAuthors

- între tabela Carti și tabela Biblioteca – raport de cardinalitate M:N.



Table Name: <input type="text" value="bookreservations"/>		Schema: <b>adi</b>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
ReservationID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
BookID	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
UserID	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LibraryID	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ReservationDate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
DueDate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Figure 3.5: Tabela BookReservations

Table Name: <input type="text" value="users"/>		Schema: <b>adi</b>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
UserID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Username	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Password	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
FirstName	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LastName	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Telephone	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Address	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Email	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Role	ENUM('Admin', 'Regu...)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Figure 3.6: Tabela User

## 3.2 Pagina principala

În figura de mai sus putem vedea pagina de intampinare a site-ului. Conținutul este centrat pentru un aspect plăcut și pentru a face citirea informațiilor mai ușoară. Pagina principală conține informații generale despre proiect.

### Welcome to Our Library Management System

Creați o aplicație care să conțină o bază de date creată în MySQL și o interfață pentru aceasta. La crearea interfeței se va folosi tehnologia Python - Flask.

#### Detalii Proiect

- Profesor: Pupezescu Valentin
- Tip: Individual

#### Structura Bazei de Date

- Autorii(AutorID, NumeAutor, PrenumeAutor, TaraOrigine)
- Carti(CartelID, Denumire, AnAparitie, Editura)
- Biblioteca(BibliotecaID, Denumire, Adresa)

#### Asocieri între Tabele

- între tabela Autori și tabela Carti – raport de cardinalitate M:N.
- între tabela Carti și tabela Biblioteca – raport de cardinalitate M:N.

Interfața va trebui să permită utilizatorului să facă următoarele operații pe toate tabele: vizualizare, adăugare, modificare, ștergere. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celelalte tabele.

Login

Sign Up

Figure 3.7: Pagina de intampinare

## 3.3 Autentificarea

Prima cerință care trebuie abordată este implementarea accesului pe două niveluri. Am discutat mai devreme despre tabela cu utilizatori, în care avem lista cu toți utilizatorii, fiecare având un nume, o parolă și un rol. Cele două

---

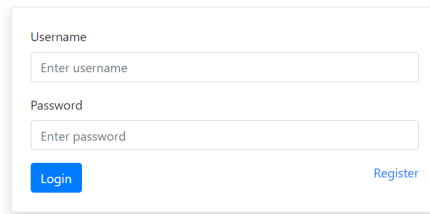
roluri corespund celor două niveluri, administrator și client.

Procesul de login folosește metode sigure de verificare a credențialelor, utilizând tehnologii precum Werkzeug pentru hashing-ul și verificarea parolelor. Aceasta garantează că doar utilizatorii autentici pot accesa conturile lor. Extensia Flask-Login este utilizată pentru a gestiona sesiunile de utilizatori, permițând utilizatorilor să rămână conectați și protejând rutele aplicației care necesită autentificare

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(cursorclass=
            cursors.DictCursor)
        cursor.execute("SELECT * FROM Users WHERE
            Username = %s", [username])
        user = cursor.fetchone()
        cursor.close()
        if user and check_password_hash(user['Password'], password):
            login_user(
                User(user['UserID'], user['Username'],
                    user['Email'], user['Role'], user['
                    FirstName'], user['LastName'],
                    user['Telephone'], user['Address'
                    ]))
            return redirect(url_for('home'))
        else:
            flash('Invalid username or password', '
                danger')
    return render_template('login.html')
```

Invalid username or password

## Login



The login form is a white box with a light gray border. It contains two input fields: 'Username' with a placeholder 'Enter username' and 'Password' with a placeholder 'Enter password'. Below the password field is a blue 'Login' button. To the right of the button is a blue 'Register' link.

Figure 3.8: Pagina de Login si mesajul aferent introducerii datelor gresite

Funcționalitatea de register permite utilizatorilor noi să creeze conturi în aplicație. Procesul de înregistrare implică completarea unui formular care colectează informații esențiale precum numele de utilizator, emailul și parola. WTForms asigură validarea acestor date, verificând atât completitudinea cât și corectitudinea informațiilor introduse. În plus, aplicația verifică dacă numele de utilizator sau adresa de email sunt deja folosite, prevenind astfel duplicatele și conflictele de cont. Securitatea datelor este o prioritate, așadar parolele sunt hash-uite folosind Werkzeug înainte de a fi stocate în baza de date, oferind o securitate sporită a informațiilor sensibile.

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        username = form.username.data
        email = form.email.data
        password = generate_password_hash(form.
            password.data)
        firstName = form.firstName.data
        lastName = form.lastName.data
        telephone = form.telephone.data
        address = form.address.data
        role = form.role.data

        cursor = mysql.connection.cursor(cursorclass=
            cursors.DictCursor)

        query = "INSERT INTO Users (Username, Password
            , FirstName, LastName, Telephone, Address,
            Email, Role) VALUES (%s, %s, %s, %s, %s, %s,
            %s, %s)"
        cursor.execute(query, (username, password,
```

```

        firstName, lastName, telephone, address,
        email, role))
mysql.connection.commit()
cursor.close()
return redirect(url_for('login'))
return render_template('register.html', form=form)

```

Aceste doua functii sunt apelate in back-end, iar datele sunt preluate dintr-un formular HTML. Validarea formularelor se face cu ajutorul bibliotecii Flask-WTForms

**Register**

Figure 3.9: Pagina de Inregistrare si mesajul aferent introducerii unui utilizator existent

## Cod pentru adaugarea Cartilor

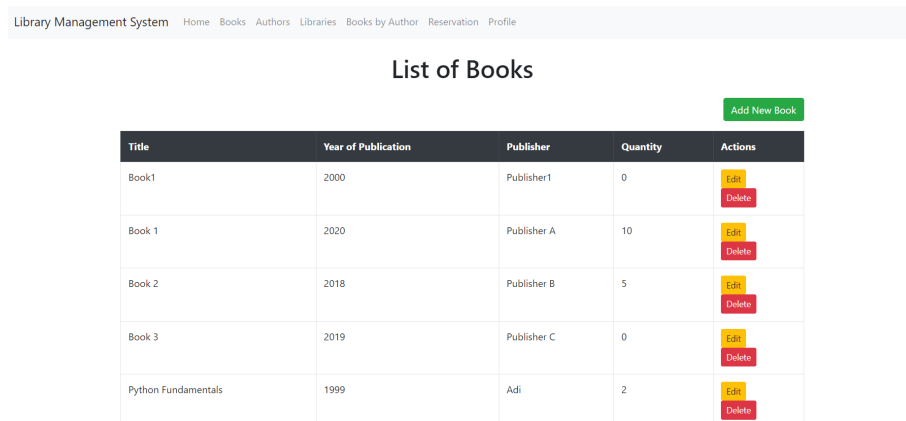
```

<div class="form-group">
{{ form.username.label(class="form-label") }}
{{ form.username(class="form-control" + (" is-invalid"
    if form.username.errors else "")) }}
{% if form.username.errors %}
    <div class="invalid-feedback">
        {{ form.username.errors[0] }}
    </div>
{% endif %}
</div>

```

### 3.4 Vizualizarea Datelor

Utilizatorii au capacitatea de a vizualiza diverse seturi de date din aplicație. Aceasta include lista cărților, autorilor, bibliotecilor și a altor resurse relevante. Procesul de vizualizare este realizat prin interogarea bazei de date MySQL și prezentarea rezultatelor într-o manieră ușor de citit și accesibilă, folosind interfața construită cu Flask și Bootstrap. De exemplu, la accesarea secțiunii de cărți, aplicația extrage și afișează detaliile tuturor cărților disponibile, precum titlu, autor și anul publicării.



Library Management System   Home   Books   Authors   Libraries   Books by Author   Reservation   Profile				
List of Books				
Add New Book				
Title	Year of Publication	Publisher	Quantity	Actions
Book1	2000	Publisher1	0	<a href="#">Edit</a> <a href="#">Delete</a>
Book 1	2020	Publisher A	10	<a href="#">Edit</a> <a href="#">Delete</a>
Book 2	2018	Publisher B	5	<a href="#">Edit</a> <a href="#">Delete</a>
Book 3	2019	Publisher C	0	<a href="#">Edit</a> <a href="#">Delete</a>
Python Fundamentals	1999	Adi	2	<a href="#">Edit</a> <a href="#">Delete</a>

Figure 3.10: Tabelul Books din pagina web

### Cod pentru adaugarea Cartilor

```
@app.route('/books')
@login_required
def books():
    form = BookForm()
    cursor = mysql.connection.cursor(cursorclass=
        cursors.DictCursor)
    cursor.execute("SELECT * FROM Books")
    books = cursor.fetchall()
    cursor.close()
    return render_template('books.html', books=books,
        form=form)
```

### 3.5 Adăugarea de Date

Adăugarea de date în aplicație este un proces vital care permite extinderea și actualizarea colecțiilor de resurse ale bibliotecii. Utilizatorii pot adăuga noi cărți, autori sau informații despre biblioteci folosind formularele web create cu

---

Flask-WTF și WTForms. Aceste formulare colectează datele necesare și, după validarea lor, informațiile sunt inserate în baza de date MySQL. De exemplu, când se adaugă o carte nouă, utilizatorii introduc detalii precum titlul, autorul și anul publicării, iar aplicația stochează aceste informații în baza de date.

## Cod pentru adaugarea Cartilor

```
@app.route('/add_author', methods=['POST'])
@login_required
def add_author():
    # Retrieve form data
    first_name = request.form['firstName']
    last_name = request.form['lastName']
    country = request.form['country']
    # Create a new author record
    cursor = mysql.connection.cursor(cursorclass=
        cursors.DictCursor)
    query = "INSERT INTO Authors (FirstName, LastName,
        CountryOfOrigin) VALUES (%s, %s, %s)"
    cursor.execute(query, (first_name, last_name,
        country))

    mysql.connection.commit()
    cursor.close()

    flash('New author added successfully!')

    return redirect(url_for('authors'))
```

Adăugarea se face prin intermediul unui modal.

Figure 3.11: Modal pentru adaugare

## Implementarea modalului

```
<div class="modal_fade" id="mymodal" tabindex="-1"
  role="dialog" aria-labelledby="exampleModalLabel"
  aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
  <h5 class="modal-title" id="exampleModalLabel">Add
    New Book</h5>
  <button type="button" class="close" data-dismiss="
    modal" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
<form method="POST" action="{ {url_for('add_book')} }"
  >
  <div class="modal-body">
```

```

    {{ form.hidden_tag() }}
    <div class="form-group">
        {{ form.title.label(class="form-label") }}
        {{ form.title(class="form-control",
            placeholder="Enter book") }}
    </div>
    <div class="form-group">
        {{ form.year.label(class="form-label") }}
        {{ form.year(class="form-control",
            placeholder="Enter book year") }}
    </div>
    <div class="form-group">
        {{ form.publisher.label(class="form-label")
        }}
        {{ form.publisher(class="form-control",
            placeholder="Enter book publisher") }}
    </div>
    <div class="form-group">
        {{ form.quantity.label(class="form-label")
        }}
        {{ form.quantity(class="form-control",
            placeholder="Enter book quantity") }}
    </div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary
        " data-dismiss="modal">Close</button>
    {{ form.submit(class="btn btn-primary") }}
</div>
</form>
</div>
</div>
</div>

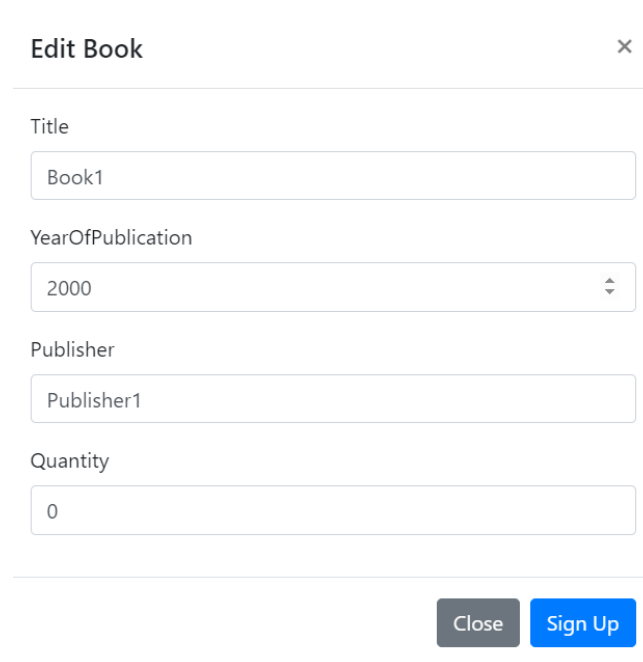
```

### 3.6 Modificarea datelor

Operațiunea de modificare a datelor în aplicația ta de gestionare a bibliotecii este o componentă esențială care asigură actualizarea și corectarea informațiilor existente în sistem. Aceasta permite utilizatorilor cu drepturi adecvate să actualizeze detalii despre cărți, autori, biblioteci și alte elemente. Pentru a asigura integritatea datelor, WTForms este folosit pentru validarea informațiilor intro-



duse în formular. Aceasta asigură că toate datele modificate sunt în formatul corect și respectă regulile de validare stabilite.



The image shows a web application modal titled "Edit Book". It contains a form with the following fields: "Title" (text input with "Book1"), "YearOfPublication" (dropdown menu with "2000"), "Publisher" (text input with "Publisher1"), and "Quantity" (text input with "0"). At the bottom right of the modal are two buttons: "Close" (grey) and "Sign Up" (blue).

Figure 3.12: Modalul specific modificarilor

## Cod pentru modificarea Cartilor

```
@app.route('/update_book/<int:id>', methods=['GET', 'POST'])
@login_required
def update_book(id):
    form = BookForm()
    cursor = mysql.connection.cursor(cursorclass=
        cursors.DictCursor)
    cursor.execute("SELECT * FROM Books")
    books = cursor.fetchall()
    cursor.close()
    if request.method == 'GET':
        cursor = mysql.connection.cursor(cursorclass=
            cursors.DictCursor)
        cursor.execute("SELECT * FROM Books WHERE
            BookID=%s", [id])
        book = cursor.fetchone()
        form.title.data = book['Title']
        form.year.data = book['YearOfPublication']
        form.publisher.data = book['Publisher']
        form.quantity.data = book['Quantity']
```

```

        cursor.close()
    elif form.validate_on_submit():
        title = form.title.data
        year = form.year.data
        publisher = form.publisher.data
        quantity = form.quantity.data
        cursor = mysql.connection.cursor(cursorclass=
            cursors.DictCursor)
        query = "UPDATE Books SET Title=_%s,
            YearOfPublication=_%s, Publisher=_%s,
            Quantity=%s WHERE BookID=_%s"
        cursor.execute(query, (title, year, publisher,
            quantity, id))
        mysql.connection.commit()
        cursor.close()
        flash('Book updated successfully!')
        return redirect(url_for('books'))
    return render_template('books.html', form=form,
        books=books)

```

### 3.7 Ștergerea Datelor

Ștergerea datelor este gestionată cu prudență pentru a asigura că informațiile importante nu sunt eliminate accidental. Această operațiune este de obicei rezervată utilizatorilor cu roluri administrative. Când un utilizator decide să șteargă o anumită intrare, cum ar fi o carte sau un autor, aplicația solicită confirmarea pentru a evita ștergeri accidentale. Odată confirmată, comanda de ștergere este executată, iar intrarea respectivă este eliminată permanent din baza de date.

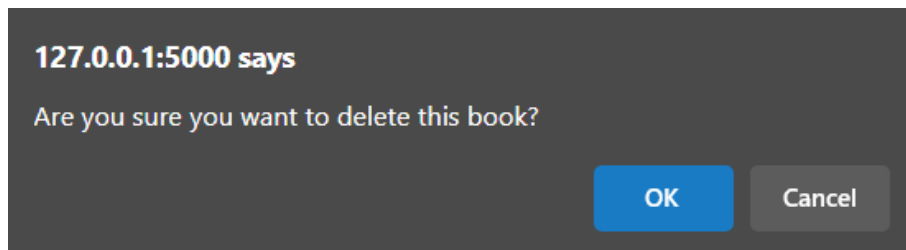


Figure 3.13: Mesajul de confirmare pentru ștergerea datelor

Only admin users can delete data.

## List of Books

Add New Book

Title	Year of Publication	Publisher	Quantity	Actions
Book1	2000	Publisher1	0	<a href="#">Edit</a> <a href="#">Delete</a>
Book 1	2020	Publisher A	10	<a href="#">Edit</a> <a href="#">Delete</a>
Book 2	2018	Publisher B	5	<a href="#">Edit</a> <a href="#">Delete</a>
Book 3	2019	Publisher C	0	<a href="#">Edit</a> <a href="#">Delete</a>
Python Fundamentals	1999	Adi	2	<a href="#">Edit</a> <a href="#">Delete</a>

Figure 3.14: Mesajul aferent rolului userului pentru stergere

### 3.8 Procesul de rezervare

Când un utilizator dorește să rezerve o carte sau o altă resursă, el navighează în secțiunea corespunzătoare din aplicație și selectează elementul dorit. Apoi, prin intermediul unei interfețe user-friendly, utilizatorul completează detaliile necesare pentru rezervare, cum ar fi data la care dorește să preia cartea. Aplicația procesează această cerere și înregistrează rezervarea în sistem, verificând disponibilitatea resursei și asigurându-se că rezervarea nu intră în conflict cu alte rezervări existente.

## List of Books and Authors

Add New Book-Author Relationship

Book Title	Author Name	Publisher	Year	Action
Book 2	John Smith	Publisher B	2018	<a href="#">Reserve</a> <a href="#">Edit</a> <a href="#">Delete</a>
Book 3	Alice Johnson	Publisher C	2019	<a href="#">Reserve</a> <a href="#">Edit</a> <a href="#">Delete</a>

Figure 3.15: Fereastra de unde se pot rezerva cartile

Un aspect important al aplicației este capacitatea fiecărui utilizator de a vedea propriile rezervări. După ce se autentifică, utilizatorul poate accesa o secțiune dedicată unde sunt listate toate rezervările sale curente. Această funcționalitate permite utilizatorilor să urmărească statusul fiecărei rezervări, inclusiv datele de preluare și returnare, și orice alte detalii relevante

## List of Reservations

Book Title	Library	Publisher	Reservation Date	Due Date	Action
Book 2	Library Aa	Publisher B	2024-01-15	2024-01-29	<a href="#">Cancel Reservation</a>
Book 2	Library Aa	Publisher B	2024-01-30	2024-02-13	<a href="#">Cancel Reservation</a>

Figure 3.16: Fereastră rezervărilor

### Codul specific Tranzacției

```
CREATE DEFINER='root'@'localhost' PROCEDURE `
  ReserveBook` (
    IN TBookID INT,
    IN TUserID INT,
    IN TLibraryID INT,
    IN TReservationDate DATE,
    IN TDueDate DATE,
    OUT TREZULTAT VARCHAR(50)
  )
BEGIN
  DECLARE TNrBooks INT;
  DECLARE mesaj VARCHAR(50);

  START TRANSACTION;

  SELECT Quantity INTO TNrBooks FROM Books WHERE
    BookID = TBookID;

  IF TNrBooks > 0 THEN
    UPDATE Books SET Quantity = Quantity - 1 WHERE
      BookID = TBookID;

    INSERT INTO BookReservations (BookID, UserID,
      LibraryID, ReservationDate, DueDate)
    VALUES (TBookID, TUserID, TLibraryID,
      TReservationDate, TDueDate);

    COMMIT;
```

---

```
        SET TREZULTAT = 'Reservation_successful';
ELSE
    ROLLBACK;
    SET TREZULTAT = 'Reservation_failed_-_No_
        copies_available';
END IF;
END
```

---

## 4 Concluzii

### 4.1 Concluzii generale

Acest proces m-a ajutat să înțeleg cum diferite componente software pot lucra împreună pentru a construi o soluție eficientă. Dezvoltarea funcționalităților de bază, cum ar fi vizualizarea, adăugarea, ștergerea și modificarea datelor, a fost un bun exercițiu de echilibru între funcționalitate și accesibilitate.

Implementarea sistemelor de login și register a fost, de asemenea, o parte importantă a proiectului, accentuând necesitatea protejării datelor utilizatorilor. Fiecare etapă din acest proces a subliniat cât de crucial este să construiești o aplicație nu doar eficientă, dar și securizată.

Dezvoltarea funcției de rezervare a fost o parte semnificativă a proiectului. A fost impresionant să văd cum funcționalitatea pe care am creat-o îmbunătățește experiența utilizatorilor, permițându-le să gestioneze eficient rezervările.

Pe lângă abilitățile tehnice, acest proiect mi-a oferit și înțelegerea importanței unui design orientat către utilizator. Am învățat că o aplicație trebuie să fie nu doar funcțională, ci și ușor de utilizat pentru toți utilizatorii. De asemenea, am învățat despre importanța feedbackului și a adaptării continue, modificând aplicația pentru a răspunde nevoilor în schimbare ale utilizatorilor și ale mediului de bibliotecă.

#### 4.1.1 Dezvoltari ulterioare

Un pas important în această direcție ar fi trecerea la un framework frontend modern și mai dinamic, precum React. React oferă un mod mai flexibil și eficient de a construi interfețe utilizator, permițând o experiență mai fluidă și mai interactivă. De exemplu, prin implementarea React, procesul de rezervare a cărților ar putea deveni mult mai intuitiv, cu actualizări în timp real și o interacțiune mai directă cu utilizatorul.

De asemenea, React ar putea facilita dezvoltarea unei versiuni mobile a aplicației, extinzând accesibilitatea și ușurința de utilizare. O interfață responsive și adaptabilă la diferite dispozitive ar face aplicația mult mai accesibilă pentru utilizatorii care preferă dispozitivele mobile.

---

## References

- [1] Flask Pallets Project. *Flask Official Documentation*. <https://flask.palletsprojects.com/>.
- [2] MySQL. *MySQL Official Documentation*. <https://www.mysql.com/>.
- [3] React. *React Official Documentation*. <https://reactjs.org/>.
- [4] MDN Web Docs. *Learn web development*. <https://developer.mozilla.org/>.
- [5] W3Schools Online Web Tutorials. *W3Schools*. <https://www.w3schools.com/>.
- [6] Bootstrap. *Bootstrap Official Documentation*. <https://getbootstrap.com/>.
- [7] Stack Overflow. <https://stackoverflow.com/>.
- [8] GitHub. <https://github.com/>.
- [9] Codecademy. *Learn to Code*. <https://www.codecademy.com/>.
- [10] Coursera. *Online Courses*. <https://www.coursera.org/>.