

Librărie online

Student: Teocan Patricia

Grupa: 30234

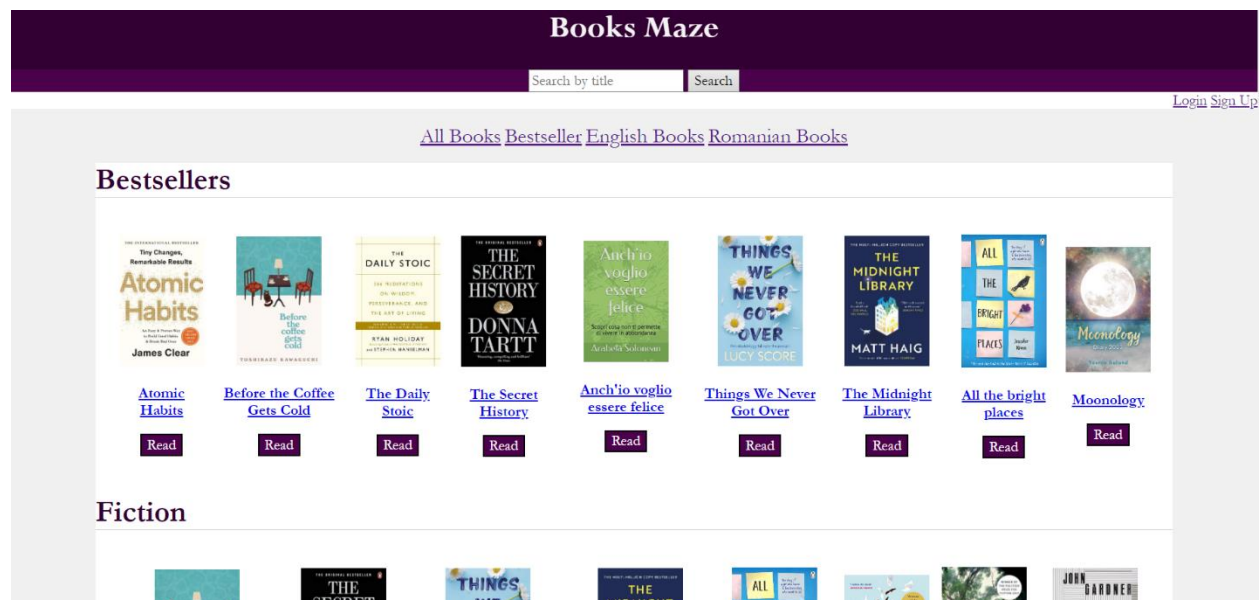
Cuprins

1. Introducere	3
2. Tehnologii.....	5
3. Deployment diagram	5
4. Class diagram	6
5. Use cases.....	6
6. Anexă	7
7. Bibliografie	12

1. Introducere

Librăria online descrisă este o pagină web creată pentru a veni în ajutorul cititorilor. Această aplicație are la baza funcționării o bază de date pentru a menține și accesa cu ușurință date despre utilizatori, administratori și produsele gestionate. Modul de utilizare se împarte în două mari categorii fiecare beneficiind de diverse oportunități: client și administrator. După cum îi spune și numele, administratorul se ocupă de partea de management al aplicației și a datelor despre produse. Rolul administratorului este de a gestiona stocurile librăriei, de a adăuga și șterge cărți. Pe partea de utilizator sunt mai multe beneficii. Clienții pot să achiziționeze cărți.. Pentru a-i fi mai ușor clientului în căutarea cărții dorite, sau a uneia cât mai potrivite pentru el, poate filtra rezultatele căutării în funcție de limbă sau există o secțiune în care sunt afișate cărțile ce au câștigat titlul de bestseller.

Pagina web este realizată în așa fel încât să fie cât mai ușor de înțeles și de utilizat pentru client și toate opțiunile de care beneficiază să fie la vedere. În primă instanță, la deschiderea aplicației apare pagina principală, unde se pot vedea cărțile, clasificate după gen, fără a fi autentificat.



Pentru a intra în cont există secțiunea de înregistrare sau de logare, însă pentru a putea diferenția partea de utilizatori de cea de administrație, la înregistrare sau la cea de intrare în cont, trebuie specificat rolul prin alegerea unei opțiuni “client”, “admin”. Tot aici, sunt afișate câteva cărți din baza de date, clasificate după gen, așezate pe rânduri. Utilizatorul poate alege să vadă



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

cărțile sortate după limbă, română sau engleză, sau pe cele de tip bestseller. Accesând aceste pagini, vor fi afișate mai multe informații despre cărți decât pe pagina principală. În plus, poate căuta cărți în funcție de titlu, folosindu-se de butonul și câmpul din partea de sus.

Login

Name	<input type="text" value="admin"/>
Password	<input type="password" value="....."/>
Role	<input type="text" value="Enter admin or user"/>
<input type="button" value="Login"/> <input type="button" value="Register"/>	

Register

Name	<input type="text" value="Enter your name"/>
Username	<input type="text" value="admin"/>
Phone	<input type="text" value="Enter phone number"/>
Password	<input type="password" value="....."/>
<input type="button" value="Sign Up"/>	
<input type="button" value="Cancel"/>	

Dacă utilizatorul nu este conectat și apasă pe butonul pentru a citi o carte, este redirecționat mai întâi spre pagina de logare. Dacă acesta nu are un cont înregistrat, își poate crea unul în secțiune de Register. În cazul în care utilizatorul a intrat în cont, există opțiunea de a-și vedea datele, apăsând butonul Account din partea de sus, de lângă Log Out, care îl deconectează, redirecționându-l pe pagina principală.

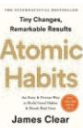

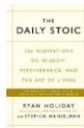






Books Maze

[Account](#) [Log Out](#)

[All Books](#) [Bestseller](#) [English Books](#) [Romanian Books](#)

Welcome to your account, Patricia Teocan - user!

Bestsellers

 Atomic Habits <input type="button" value="Read"/>	 Before the Coffee Gets Cold <input type="button" value="Read"/>	 The Daily Stoic <input type="button" value="Read"/>	 The Secret History <input type="button" value="Read"/>	 Anch'io voglio essere felice <input type="button" value="Read"/>	 Things We Never Got Over <input type="button" value="Read"/>	 The Midnight Library <input type="button" value="Read"/>	 All the bright places <input type="button" value="Read"/>	 Moonology <input type="button" value="Read"/>
---	---	---	--	--	---	--	---	---

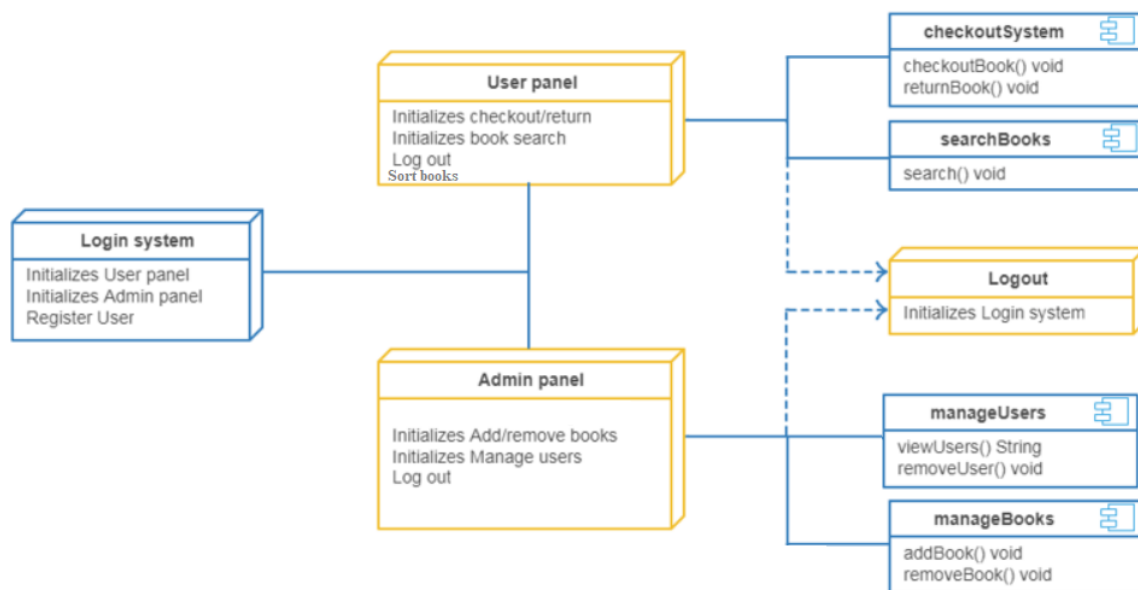
Fiction

2. Tehnologii

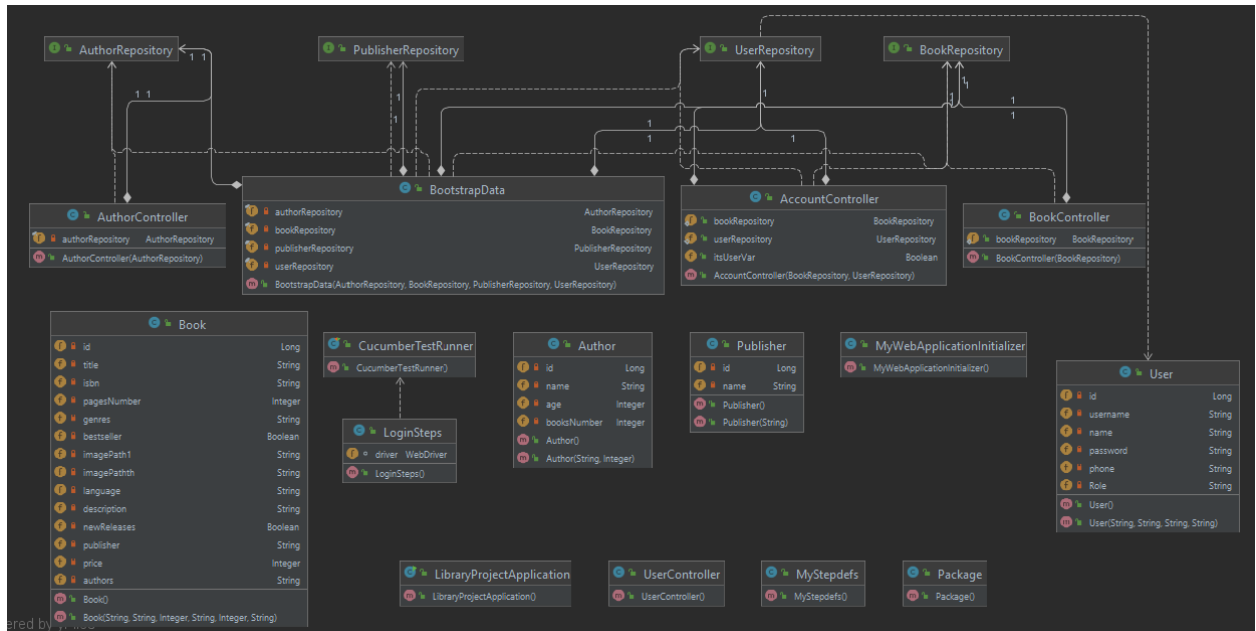
Aplicația este dezvoltată în IntelliJ IDEA. Codul paginii web al librăriei online este descris într-un proiect de tip MAVEN în limbajul de programare orientat pe obiecte Java utilizând framework-ul Spring BOOT respectând formatul MODEL VIEW CONTROLLER (MVC). Din cadrul framework-ului folosim baza de date JPA H2. Pentru operațiile asupra bazei de date, precum INSERT, DELETE, UPDATE și altele se folosește tehnologia Spring Hibernate. Pentru realizarea părții de view, se va folosi codul în HTML cu THYMELEAF ce lucrează împreună cu Java.

Pentru testarea funcționalității aplicației, am folosit perechea SELENIUM – CUCUMBER ce verifică datele introduse în aplicație de către utilizator, folosind pagini WEB din pachetul CHROME.

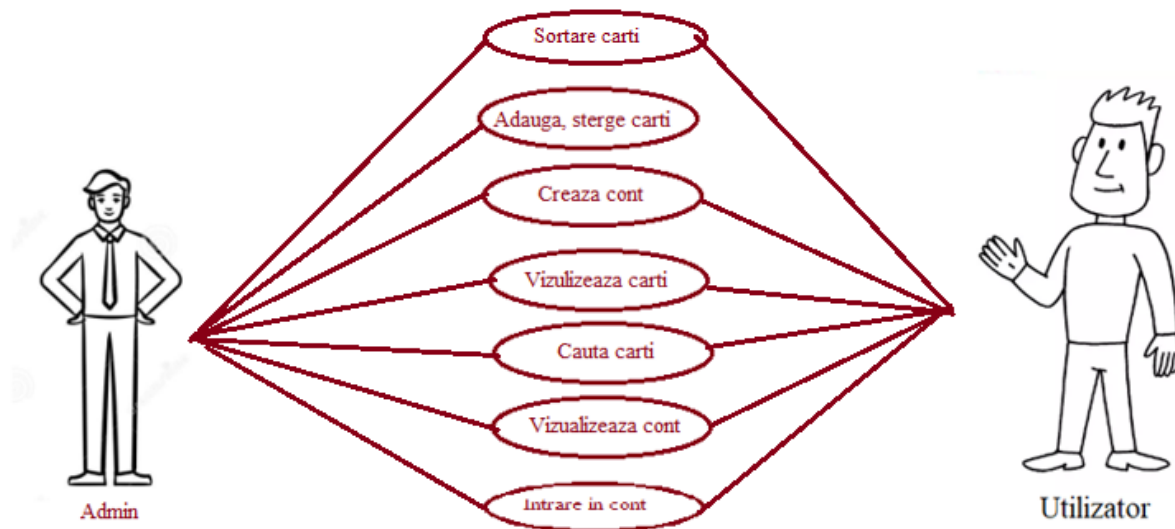
3. Deployment diagram



4. Class diagram



5. Use cases





6. Anexă

Clasa Book

```
@Entity
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String title;
    private String isbn="";
    private Integer pagesNumber;
    private String genres;
    private Boolean bestseller = Boolean.FALSE;
    private String imagePath1="";
    private String imagePathth="";
    private String language;
    private String description = " ";
    private Boolean newReleases = Boolean.FALSE;
    private String publisher;
    private Integer price;
    private String authors;
    public Book() { }
    public Book(String title, String isbn, Integer pagesNumber, String genres, Integer price, String language)
    {
        this.title = title;
        this.isbn = isbn;
        this.pagesNumber = pagesNumber;
        this.genres = genres;
        this.price = price;
        this.language = language;
    }
    public Long getId() {return id; }
    public String getTitle() {return title; }
    public String getIsbn() {return isbn; }
    public String getAuthors() {return authors; }
    public Integer getPagesNumber() {return pagesNumber; }
    public String getGenres() { return genres; }
    public String getPublisher() {return publisher; }
    public Integer getPrice() { return price; }
    public Boolean getbestseller(){return bestseller;}
    public Boolean getnewReleases(){return newReleases;}
    public String getImagePath1(){return imagePath1;}
    public String getImagePathth(){return imagePathth;}
    public String getLanguage(){return language;}
    public String getDescription(){return description;}

    public void setId(Long id) {id = id; }
    public void setTitle(String title) { this.title = title; }
    public void setIsbn(String isbn) {this.isbn = isbn; }
    public void setAuthors(String authors) {this.authors = authors; }
    public void setPagesNumber(Integer pagesNumber) {this.pagesNumber = pagesNumber; }
    public void setGenres(String genres) {this.genres = genres; public void setPublisher(String publisher)
    {this.publisher = publisher; }
    public void setPrice(Integer price) { this.price = price; }
    public void setBestseller(Boolean bestseller){this.bestseller = bestseller;}
    public void setnewReleases(Boolean newReleases){this.newReleases = newReleases;}
    public void setLanguage(String language){this.language = language;}
    public void setImagePathth(String imagePathth){this.imagePathth= imagePathth;}
    public void setImagePath1(String imagePath1){this.imagePath1= imagePath1;}
    public void setDescription(String description){this.description= description;}
    @Override
    public String toString() {
        return "Book{" + "id=" + id + ", title='" + title + '\'' + ", isbn='" + isbn + '\'' + " " + "'; }
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Book book = (Book) o;
        return id != null ? id.equals(book.id) : book.id == null;
    }
    @Override
    public int hashCode() return id != null ? id.hashCode() : 0; }
}
```



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Controllerul clasei Book

```
@Controller
public class BookController {
    public static BookRepository bookRepository;
    public BookController(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }
    @RequestMapping() // pt link
    public static String getBooksMain(Model model){
        model.addAttribute("books", bookRepository.findAll());
        ArrayList<Book> books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> ! (x.getGenres().equals("Fiction")));
        model.addAttribute("booksFiction", books);

        books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> x.getbestseller() == Boolean.FALSE);
        model.addAttribute("booksBestseller", books);

        books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> ! (x.getGenres().equals("Personal Development")));
        model.addAttribute("booksPersonalDevelopment", books);

        books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> ! (x.getGenres().equals("Universal Literature")));
        model.addAttribute("booksUniversalLiterature", books);
        return "main";
    }
    @RequestMapping("/books") // pt link
    public String getBooks(Model model){
        model.addAttribute("books", bookRepository.findAll());
        return "books/list";
    }
    @RequestMapping("/bestseller")
    public String getTrendingBooks(Model model){
        ArrayList<Book> books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> x.getbestseller() == Boolean.FALSE);
        model.addAttribute("books", books);
        return "books/bestseller";
    }
    @RequestMapping("/englishBooks")
    public String getEnglishBooks(Model model){
        ArrayList<Book> books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> ! (x.getLanguage().equals("English")));
        model.addAttribute("books", books);
        return "books/englishBooks";
    }
    @RequestMapping("/romanianBooks")
    public String getRomanianBooks(Model model){
        ArrayList<Book> books = (ArrayList<Book>) bookRepository.findAll();
        books.removeIf(x-> ! (x.getLanguage().equals("Romanian")));
        model.addAttribute("books", books);
        return "books/romanianBooks";
    }
}
```

Introducere cărți în baza de date

```
Book atomicHabits = new Book("Atomic Habits", "12345", 320, "Universal literature",40,
"English");
atomicHabits.setImagePath1("../atomic.jpg");
atomicHabits.setDescription("People think that when you want to change your life, you need to
think big. But world-renowned habits expert James Clear has discovered another way.He knows that
real change comes from the compound effect of hundreds of small decisions.");
atomicHabits.setBestseller(Boolean.TRUE);

jamesClear.increaseBooksNumber();
atomicHabits.setAuthors("James Clear");
```




UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

```
atomicHabits.setPublisher("Cornerstone");
bookRepository.save(atomicHabits);

Book beforetheCoffeeGetsCold = new Book("Before the Coffee Gets Cold","93274982", 224, "Fiction",
55, "English");
beforetheCoffeeGetsCold.setImagePath1("../beforeTheCoffee.jpg");
beforetheCoffeeGetsCold.setDescription("In a small back alley in Tokyo, there is a cafe which has
been serving carefully brewed coffee for more than one hundred years. But this coffee shop offers
its customers a unique experience: the chance to travel back in time.");
beforetheCoffeeGetsCold.setBestseller(Boolean.TRUE);
ToshikazuKawaguchi.increaseBooksNumber();
beforetheCoffeeGetsCold.setAuthors("Toshikazu Kawaguchi");
beforetheCoffeeGetsCold.setPublisher("PanMacmillan");
bookRepository.save(beforetheCoffeeGetsCold);

Book theDailyStoic = new Book("The Daily Stoic","12312", 416, "Personal Development",
55,"English");
theDailyStoic.setImagePath1("../dailyStoic.jpg");
theDailyStoic.setDescription("'A generous gift of guidance on modern living culled from a canon
of wisdom hatched long ago.' - Maria Popova, editor of Brain Pickings\n");
theDailyStoic.setBestseller(Boolean.TRUE);
RyanHoliday.increaseBooksNumber();
theDailyStoic.setAuthors("Ryan Holiday");
theDailyStoic.setPublisher("PanMacmillan");
bookRepository.save(theDailyStoic);

Book theSecretHistory = new Book("The Secret History","827627", 640, "Fiction", 61,"Romanian");
theSecretHistory.setImagePath1("../secretHistory.jpg");
theSecretHistory.setDescription("'Everything, somehow, fit together; some sly and benevolent
Providence was revealing itself by degrees and I felt myself trembling on the brink of a fabulous
discovery, as though any morning it was all going to come together.");
theSecretHistory.setBestseller(Boolean.TRUE);
DonnaTartt.increaseBooksNumber();
theSecretHistory.setAuthors("Donna Tartt");
theSecretHistory.setPublisher("ProfileBooksLtd");
bookRepository.save(theSecretHistory);

Book essereFelice = new Book("Anch'io voglio essere felice", "43322", 164, "Personal
Development", 64, "English");
essereFelice.setBestseller(Boolean.TRUE);
essereFelice.setDescription("Il libro parla del viaggio che ho intrapreso per raggiungere la
felicità in tutte le aree della vita, parla in modo chiaro e semplice, con esempi della vita di
tutti giorni.");
essereFelice.setImagePath1("../essereFelice.jpg");
ArabelaSoloean.increaseBooksNumber();
essereFelice.setAuthors("Arabela Soloean");
essereFelice.setPublisher("ProfileBooksLtd");
bookRepository.save(essereFelice);
```

Pagina principală realizată cu HTML și THYMELEAF

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns:font-
family="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8"/>
    <title>Books Maze</title>
</head>
<body style="background-color:#f0f0f0; font-family: Garamond; font-size:110%; ">
<div>
    <div id="top" class="bar" style="background: #330033">
        <a class="logo"> <b> <h1 style="color: whitesmoke; font-size:220%;font-family:
Perpetua; margin-top: 0px" width="70" height="70" align="center">Books Maze</h1> </b> </a>
        <div style="background: #4b014b; font-size: 100%; font-family: Garamond; color:
whitesmoke"; align="center" >
            <form action="#" th:action="@{/searchBooks}" th:object="${books}" th:method="POST">
                <input type="text" name="title" placeholder="Search by title" aria-
```



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

```
label="SearchByTitle" autocomplete="off" style="font-size: 100%; font-family: Garamond;">
    <input type="submit" value="Search" class="search-bar-submit" aria-label="Search
submit" style="font-size: 100%; font-family: Garamond">
    </form>
</div>
</div>
</div>
<div id="headerBar" align="center">
    <div class="search-component"; align="right" style="background: white">
        <a class="btn primary" href="/account/login" >Login</a>
        <a class="btn primary" href="/account/create" >Sign Up</a>
    </div>
</div>

<p align="center">
    <a class="btn" href="/books" style="font-size:140%">All Books</a>
    <a class="btn" href="/bestseller" style="font-size:140%">Bestseller</a>
    <a class="btn" href="/englishBooks" style="font-size:140%">English Books</a>
    <a class="btn" href="/romanianBooks" style="font-size:140%">Romanian Books</a>
</p>
<table id="bestsellers" style="margin-left: 100px; margin-right: 100px; background-color: white;
border-top: 1px solid lightgray; padding: 15px;" align="center" cellspacing="10" >
    <div style="margin-left: 100px; margin-right: 100px; background-color:white; font-family:
Garamond; color: #330033; font-size:200%;"> <b> Bestsellers </b> </div>
    <th th:each="book : ${booksBestseller}" >
        <p>  </p>

        <a class="btn" th:href="${book.title}" th:text="${book.title}" >"</a>
        <p>
        <form action="/loginRead" method="post">
            <input type="submit" value="Read" class="search-bar-submit" aria-label="Search
submit" style="font-size: 100%; font-family: Garamond;background-color:#4b014b;color:white ">
            </form>
        </p>
    </th>

</table>
<table id="fiction" style="margin-left: 100px; margin-right: 100px; background-color: white;
border-top: 1px solid lightgray; padding: 15px;" align="center" cellspacing="10" >
    <div style="margin-left: 100px; margin-right: 100px; background-color:white; font-family:
Garamond; color: #330033; font-size:200%;"> <b>Fiction </b> </div>
    <th th:each="book : ${booksFiction}" >
        <p> 
        <p th:text="${book.title}"></p>
        <form action="/loginRead" method="post">
            <input type="submit" value="Read" class="search-bar-submit" aria-label="Search
submit" style="font-size: 100%; font-family: Garamond;background-color:#4b014b;color:white ">
            </form>
        </th>
</table>
<table id="personalDevelopment" style="margin-left: 100px; margin-right: 100px; background-color:
white; border-top: 1px solid lightgray; padding: 15px;" align="center" cellspacing="10" >
    <div style="margin-left: 100px; margin-right: 100px; background-color:white; font-family:
Garamond; color: #330033; font-size:200%;">
        <b>Personal Development</b>
    </div>
    <th th:each="book : ${booksPersonalDevelopment}" >
        <p> 
        <p th:text="${book.title}">
        </p>

        <form action="/loginRead" method="post">
            <input type="submit" value="Read" class="search-bar-submit" aria-label="Search
submit" style="font-size: 100%; font-family: Garamond;background-color:#4b014b;color:white ">
            </form>
        </th>
</table>
<table id="booksUniversalLiterature" style="margin-left: 100px; margin-right: 100px; background-
color: white; border-top: 1px solid lightgray; padding: 15px;" align="center" cellspacing="10" >
    <div style="margin-left: 100px; margin-right: 100px; background-color:white; font-family:
Garamond; color: #330033; font-size:200%;">
```



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

```
<b>
    Universal Literature
</b>
</div>
<th th:each="book : ${booksUniversalLiterature}" >
    <p> 
    <p th:text="${book.title}">
    </p>
    <form action="/loginRead" method="post">
        <input type="submit" value="Read" class="search-bar-submit" aria-label="Search
submit" style="font-size: 100%; font-family: Garamond;background-color:#4b014b;color:white ">
    </form>
    </th>
</table>
</body>
</html>
```

Pagina de logare realizată cu HTML și THYMELEAF

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org" xmlns:padding="http://www.w3.org/1999/xhtml"
    xmlns:width="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8">
    <title>Register</title>
</head>
<body style="background-color:#f0f0f0; font-family: Garamond; font-size:110%; ">
<div>
    <div id="top" class="bar" style="background: #330033">
        <a class="logo">
            <b>
                <h1 style="color: whitesmoke; font-size:220%;font-family: Perpetua; margin-top:
0px" width="70" height="70" align="center">Register</h1>
            </b>
        </a>
    </div>
</div>
<div align="center" style="background: white; margin-left: 450px; margin-right: 450px; margin-
bottom: 150px; margin-top: 150px" >
    <form action="#" th:action="@{/saveUser}" th:object="${users}" th:method="POST">

        <p>
            <label style="width: 100px; margin-left: 0px"><b>Name</b></label>
            <input type="text" placeholder="Enter your name" name="name" required>
        </p>
        <p>
            <label><b>Username</b></label>
            <input type="text" placeholder="Enter Username" name="username" required>
        </p>
        <p>
            <label><b>Phone</b></label>
            <input type="text" placeholder="Enter phone number" name="phone" required>
        </p>
        <p>
            <label ><b>Password</b></label>
            <input type="password" placeholder="Enter Password" name="password" required>
        <p>
            <button type="submit">Sign Up</button>
    </form>
</form>
<form action=".." method="post">
    <input type="submit" value="Cancel">
</form></div>
```



```
</div>  
</body>  
</html>
```

7. Bibliografie

- <https://www.thymeleaf.org/doc/articles/standardurlysyntax.html>
- <https://www.baeldung.com/spring-boot-crud-thymeleaf>
- <https://www.w3schools.com/html>
- <https://html.com>
- <https://spring.io/projects/spring-boot>
- [https://www.udemy.com/course/automate-tests-using-selenium-webdriver-with-java-cucumber/?utm_source=adwords&utm_medium=udemyads&utm_campaign=LongTail-New la.EN cc.ROWMTA-B&utm_content=deal4584&utm_term= . ag_98771201205 . ad_533999956744 . kw . de c . dm . pl . ti dsa-1007766171312 . li_9040263 . pd . &matchtype=&gclid=CjwKCAiA8OmdBhAgEiwAShr403p7eNYsY7X-qaWusDEo_KDKUziciPExo5MVdmBoNlg88307IYO-xoCo6AQAvD_BwE](https://www.udemy.com/course/automate-tests-using-selenium-webdriver-with-java-cucumber/?utm_source=adwords&utm_medium=udemyads&utm_campaign=LongTail-New%20la.EN%20cc.ROWMTA-B&utm_content=deal4584&utm_term=.ag_98771201205.ad_533999956744.kw.de.c.dm.pl.ti.dsa-1007766171312.li_9040263.pd.&matchtype=&gclid=CjwKCAiA8OmdBhAgEiwAShr403p7eNYsY7X-qaWusDEo_KDKUziciPExo5MVdmBoNlg88307IYO-xoCo6AQAvD_BwE)
- <https://www.blazemeter.com/blog/selenium-vs-cucumber#:~:text=Selenium%20is%20used%20for%20automating,terms%20of%20functionalities%20and%20usage>
-