

Facultatea de Automatică și Calculatoare

LIMBAJE ȘI TEHNOLOGII WEB

SongSphere

Student: Vieru Stefan

Grupa: SDTW 1B

Cuprins

1. Scopul lucrarii.....	3
2. Limbaje utilizate.....	3
3. Tehnologii utilizate.....	3
3.1 Java Spring Boot - backend.....	3
3.2 React - frontend.....	4
4. Exemple practice.....	6
5. Concluzii.....	10
6. Bibliografie.....	10

1. Scopul lucrării

Această lucrare are ca obiectiv dezvoltarea unei aplicații care să includă gestionarea unei platforme muzicale numită SongSphere.

2. Limbaje utilizate

La realizarea aplicației s-a folosit limbajul de programare Java.

Limba Java este un limbaj de programare orientat pe obiect, dezvoltat inițial de către Sun Microsystems și ulterior achiziționat de Oracle Corporation. Java a fost proiectat pentru a fi portabil, eficient în utilizare și orientat către scrierea de programe care rulează pe diverse dispozitive și platforme fără a necesita modificări semnificative.

Principalele avantaje ale acestui limbaj de programare sunt:

Platformă independentă: Codul sursă Java este compilat într-un format intermediar numit "bytecode", care poate fi apoi executat pe orice mașină virtuală Java (JVM), indiferent de arhitectura hardware sau sistemul de operare.

Sintaxă similară cu C/C++: Sintaxa Java este similară cu cea a limbajelor C și C++, dar adoptă un model de obiecte mai simplificat.

Gestionarea automată a memoriei: Java folosește garbage collection, ceea ce înseamnă că dezvoltatorii nu sunt nevoiți să gestioneze explicit procesele de alocare și dealocare a memoriei în cadrul codului lor. Gestionarea memoriei se face automat.

Bogată bibliotecă standard: Java vine cu o bibliotecă standard (Java Standard Edition - Java SE) care oferă numeroase funcționalități utile, cum ar fi manipularea datelor, gestionarea fișierelor, rețele, interfața grafică și multe altele.

3. Tehnologii utilizate

3.1 Java Spring Boot - backend

Spring Boot este un proiect din cadrul ecosistemului Spring Framework, care oferă un cadru simplificat și eficient pentru dezvoltarea rapidă a aplicațiilor Java. Iată câteva aspecte și avantaje importante ale Spring Boot:

Simplificarea Configurării: Spring Boot vine cu un sistem de configurare implicită, permițând dezvoltatorilor să înceapă rapid fără a fi nevoie să configureze manual o mulțime de detalii.

Gestionarea Dependințelor: Spring Boot facilitează gestionarea dependențelor prin intermediul Maven sau Gradle, oferind o listă predefinită de dependențe pentru diferite scopuri, cum ar fi dezvoltarea web, baze de date, securitate, etc.

Microservicii: Spring Boot este preferat în dezvoltarea microserviciilor, furnizând un cadru agil și eficient. Integrarea cu Spring Cloud adaugă funcționalități suplimentare pentru dezvoltarea aplicațiilor bazate pe microservicii.

Suport pentru Spring Ecosystem: Spring Boot se integrează perfect cu alte proiecte Spring, precum Spring Data, Spring Security, Spring Cloud, etc., facilitând dezvoltarea de aplicații enterprise complexe.

Încorporare a Serverului: Spring Boot include servere integrate, precum Tomcat, Jetty sau Undertow, eliminând necesitatea de a configura și utiliza un server extern. Astfel, aplicația poate fi ambalată și distribuită ca o arhivă simplă, fără a depinde de un server extern.

3.2 React - frontend

React.js, cunoscut mai frecvent sub numele de React, este o bibliotecă JavaScript gratuită și open-source. Aceasta este concepută pentru a construi interfețe de utilizator prin combinarea secțiunilor de cod (componente) pentru a crea site-uri web complete. Inițial dezvoltată de Facebook, acum este întreținută de Meta și comunitatea open-source.

Iata cateva avantaje ale acestei biblioteci:

Performanță Reactivă: React utilizează un concept denumit "Virtual DOM" pentru a optimiza actualizările interfeței utilizator. În loc să reîmprospăteze întreaga pagină la fiecare modificare, React operează asupra unei copii virtuale a DOM-ului, identificând eficient doar modificările necesare pentru actualizarea elementelor.

Componentizare Eficientă: React încurajează utilizarea componentelor - module izolate și reutilizabile de cod, ușor de asamblat pentru a crea interfețe complexe. Această abordare simplifică dezvoltarea, întreținerea și testarea aplicațiilor.

Unidirectional Data Flow: Datele într-o aplicație React se deplasează într-o singură direcție, facilitând monitorizarea și gestionarea eficientă a stării aplicației. Acest model conferă predictibilitate și claritate codului.

React Native pentru Mobilitate: Bazat pe React, React Native permite dezvoltarea aplicațiilor mobile pentru iOS și Android, folosind aceleași principii și componente, facilitând astfel crearea experiențelor mobile.

Comunitate Activă: React se bucură de o comunitate mare și dinamică de dezvoltatori, furnizând o multitudine de resurse, tutoriale și biblioteci terțe pentru a susține procesul de construire a aplicațiilor.

Integrare Ușoară și Flexibilitate: React poate fi integrat cu ușurință în proiecte existente, permițând dezvoltatorilor să adopte progresiv sau să îmbunătățească anumite secțiuni ale aplicației fără a revizui întregul cod.

React Hooks: Introduse în React 16.8, Hooks oferă o abordare mai simplă și concisă pentru gestionarea stării și ciclului de viață al componentelor, eliminând necesitatea utilizării claselor în componente.

Suport pentru Aplicații Complexe: React se potrivește construirii aplicațiilor cu stări complexe, cum ar fi aplicațiile single-page (SPA) sau cele enterprise, datorită gestionării eficiente a stării și datelor.

Rolul principal al React într-o aplicație este să gestioneze stratul de vizualizare al acelei aplicații, la fel cum face V (View) într-un model-view-controller (MVC), furnizând execuție de randare optimă și eficientă. Iată câteva dintre componentele cheie utilizate în dezvoltarea aplicațiilor React:

Componente: interfața utilizator este divizată în componente, care sunt bucatele izolate de cod reutilizabil. Acestea pot fi simple (funcționale) sau clasice (bazate pe clase). Componentele au stări și proprietăți care permit gestionarea datelor și interacțiunea între acestea.

JSX (JavaScript XML): React utilizează JSX, o sintaxă de extensie a JavaScript, care permite definirea structurii UI într-un mod similar cu XML sau HTML. Acest lucru face codul mai ușor de înțeles și de scris.

Props: proprietățile sunt utilizate pentru a transmite date de la o componentă părinte la o componentă copil. Acestea sunt considerate "read-only" și nu trebuie modificate în componenta copil.

State: starea reprezintă datele interne ale unei componente și poate fi modificată pe parcursul ciclului de viață al acesteia. Atunci când starea unei componente se schimbă, React re-renderizează componenta pentru a reflecta acele schimbări.

Hooks: hooks sunt funcții speciale introduse în React 16.8, care permit folosirea stării și a altor caracteristici React în componente funcționale. Câteva exemple: useState, useEffect, și useContext.

Routing: pentru gestionarea navigației într-o aplicație, React utilizează biblioteci precum React Router, care facilitează crearea rutelor și navigarea între diferite componente.

Redux: pentru gestionarea stării globale în aplicații mai mari, dezvoltatorii pot utiliza Redux, o bibliotecă care permite gestionarea stării într-un singur loc.

4. Exemple practice

Aplicatia SongSphere este utilizata ca si Interfata Utilizator pentru managementul unei baze de date cu artistii si melodii, pentru o platforma muzicala. Aceasta dispune de cateva modalitati de interactiune si vizualizare a continutului, realizate cu ajutorul rolurilor.

Rolurile asignate unui utilizator pot fi:

- USER
- ARTIST
- CONTENT_MANAGER
- ADMINISTRATOR_APP

In functie de rol, anumite drepturi si actiuni sunt permise fiecarui utilizator.

- USER: doar cereri de tip GET pe paginile cu artisti, melodii, si cereri PUT pe pagina de autentificare
- ARTIST: asemanator USER + cereri de tip CRUD pentru melodii
- CONTENT_MANAGER: asemanator ARTIST + cereri de tip CRUD pentru artisti
- ADMINISTRATOR_APP: acces la toate funcționalitățile aplicației

Aplicatia se foloseste de cateva tabele sql:

- artist
- record
- artist_portfolio – folosita pentru a lega artistii de melodii/albume
- role
- user
- user_role – folosita pentru asignarea rolului unui user

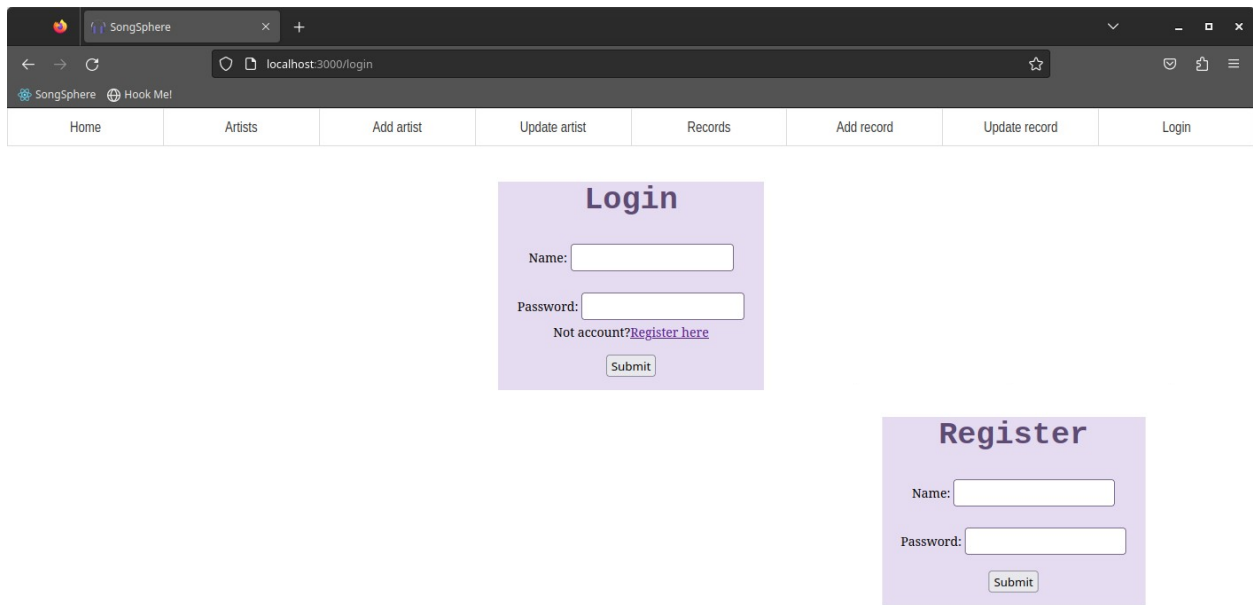


Figure 1: In stanga pagina de Autentificare, in partea dreapta pagina de Inregistrare

The image shows a web browser window with the address bar displaying 'localhost:3000/artistList'. The page features a navigation bar with links: Home, Artists, Add artist, Update artist, Records, Add record, Update record, and Logout. Below the navigation bar, there is a table listing artists. The table has four columns: Name, Active, Songs, and Action. The data rows are as follows:

Name	Active	Songs	Action
System of a Down	true	Link	Edit Delete
Jack Harlow	true	Link	Edit Delete
Roddy Ricch	true	Link	Edit Delete
Taylor Swift	true	Link	Edit Delete
Tate McRae	true	Link	Edit Delete

Figure 2: Pagina cu toti artistii

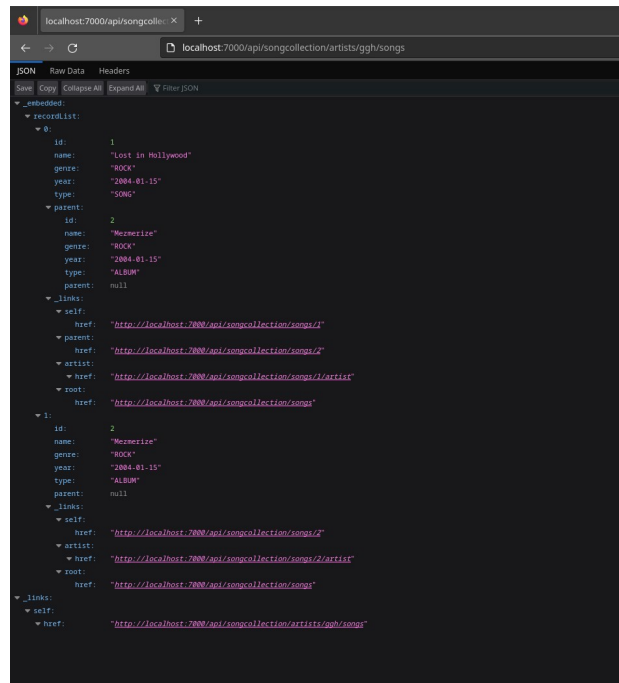


Figure 3: Lista melodiilor unui artist

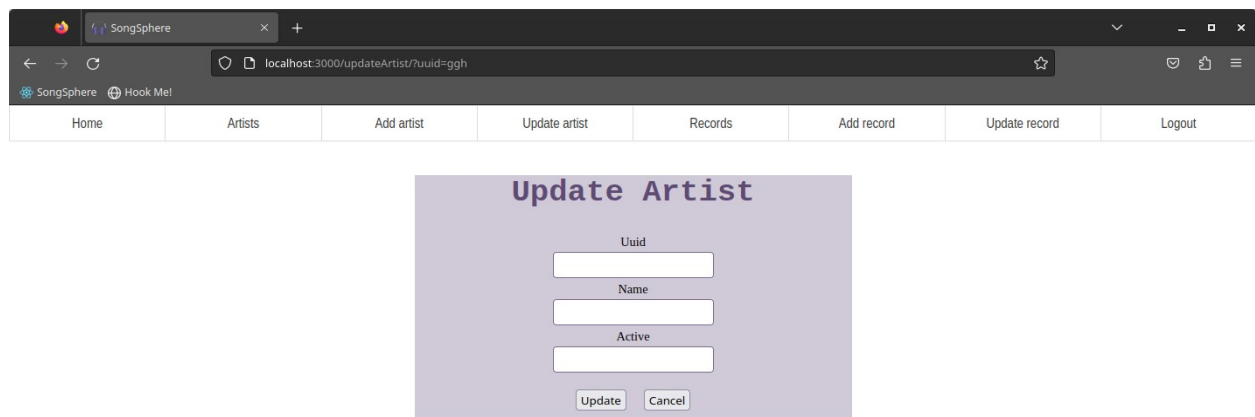


Figure 4: Pagina actualizare Artist

Name	Genre	Type	Year	Parent	Action
Lost in Hollywood	ROCK	SONG	2004-01-15	Mezmerize	Edit Delete
Mezmerize	ROCK	ALBUM	2004-01-15	None	Edit Delete

Figure 5: Pagina cu melodii

Add New Record

Name

Genre

Type

Year

Parent

Figure 6: Pagina adaugare melodie noua

The screenshot displays a web browser window with the title 'SongSphere' and the address bar showing 'localhost:3000/updateRecord'. The browser's navigation bar includes links for 'Home', 'Artists', 'Add artist', 'Update artist', 'Records', 'Add record', 'Update record', and 'Logout'. The main content area features a purple box titled 'Update Record' containing a form with the following fields: ID, Name, Genre, Type, Year, and Parent. At the bottom of the form are two buttons: 'Update' and 'Cancel'.

Figure 7: Pagina actualizare melodie

5. Concluzii

Utilizarea Spring Boot si ReactJS pentru realizarea aplicatiei s-a dovedit a fi o metoda eficienta si usor de intretinut, avand o posibilitate crescuta a adaugarii a noi functionalitati, dupa proiectarea aplicatiei.

6. Bibliografie

[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

<https://aws.amazon.com/what-is/java/>

[https://en.wikipedia.org/wiki/React_\(software\)](https://en.wikipedia.org/wiki/React_(software))

<https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>