

Universitatea Politehnica București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

## Programarea aplicativă a interfețelor

Aplicație de tip CRUD (CREATE, READ, UPDATE, DELETE)  
pentru organizarea unui sistem de asociere a artiștilor preferați  
cu genurile muzicale abordate de aceștia

Anghel-Neg Gabriela

433B

## I. Date Generale

Această aplicație este una de tip CRUD (CREATE, READ, UPDATE, DELETE) pentru organizarea unui sistem de asociere a artiștilor preferați cu genurile muzicale abordate de aceștia. Sistemul este compus din două tabele: muzicieni, stilmuzical. Primul tabel conține totalitatea artiștilor, numele lor de scenă și datele lor de naștere, iar al doilea conține câteva genuri muzicale, anul lor de apariție și originea.

## II. Tehnologii

Tehnologiile utilizate sunt:

- NodeJS (pentru interogarea bazei de date);
- phpmyadmin (pentru gestionarea bazei de date de către administrator);
- MySQL (baza de date a aplicației);
- HTML, CSS (framework Bootstrap 3.3.7), VueJS, Javascript (framework jQuery)(pentru afișarea datelor într-o formă vizuală pentru client).

## III. Arhitectura aplicației

Arhitectura aplicației este prezentată în figura de mai jos având 4 elemente: baza de date (prin MySQL), metoda de interogare a acesteia (prin NodeJS), aplicația vizuală (prin HTML, CSS, Javascript) și User-ul.

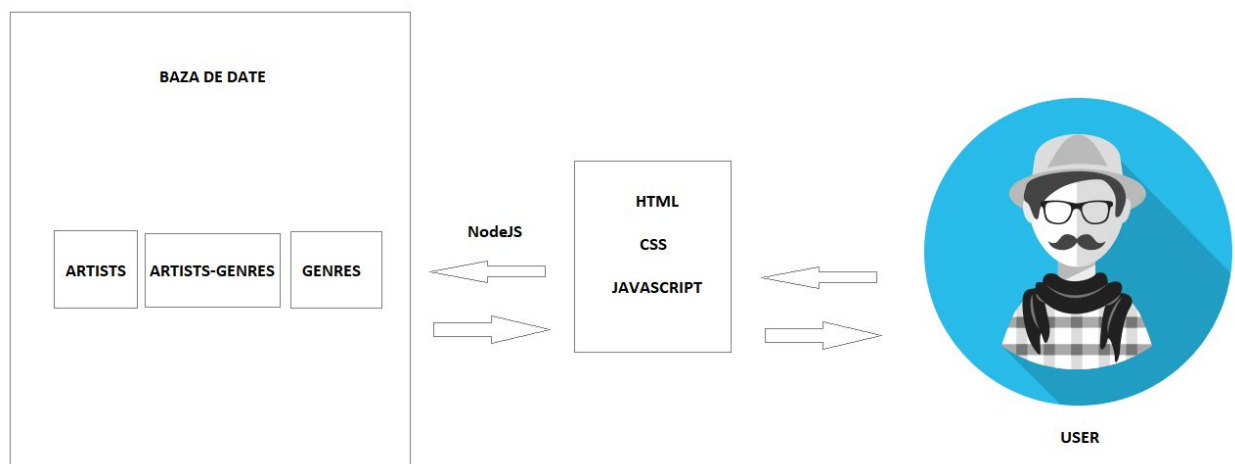


Figura 1.

## IV.Arhitectura bazei de date

Baza de date a fost compusă din 3 tabele: muzician, stilmuzical, muzician\_stil. Primul tabel, muzician conține field-urile: mId, nume, nume\_scena, an\_nastere, al doilea, stilmuzical conține field-urile: smId, nume, anAparitie, origine. Relația Many-To-Many este asigurată prin al treilea tabel muzician\_stil ce reține ce artist abordează anumite stiluri muzicale (un artist poate avea asociate mai multe stiluri muzicale, iar un stil muzical poate fi văzut la mai mulți artiști).

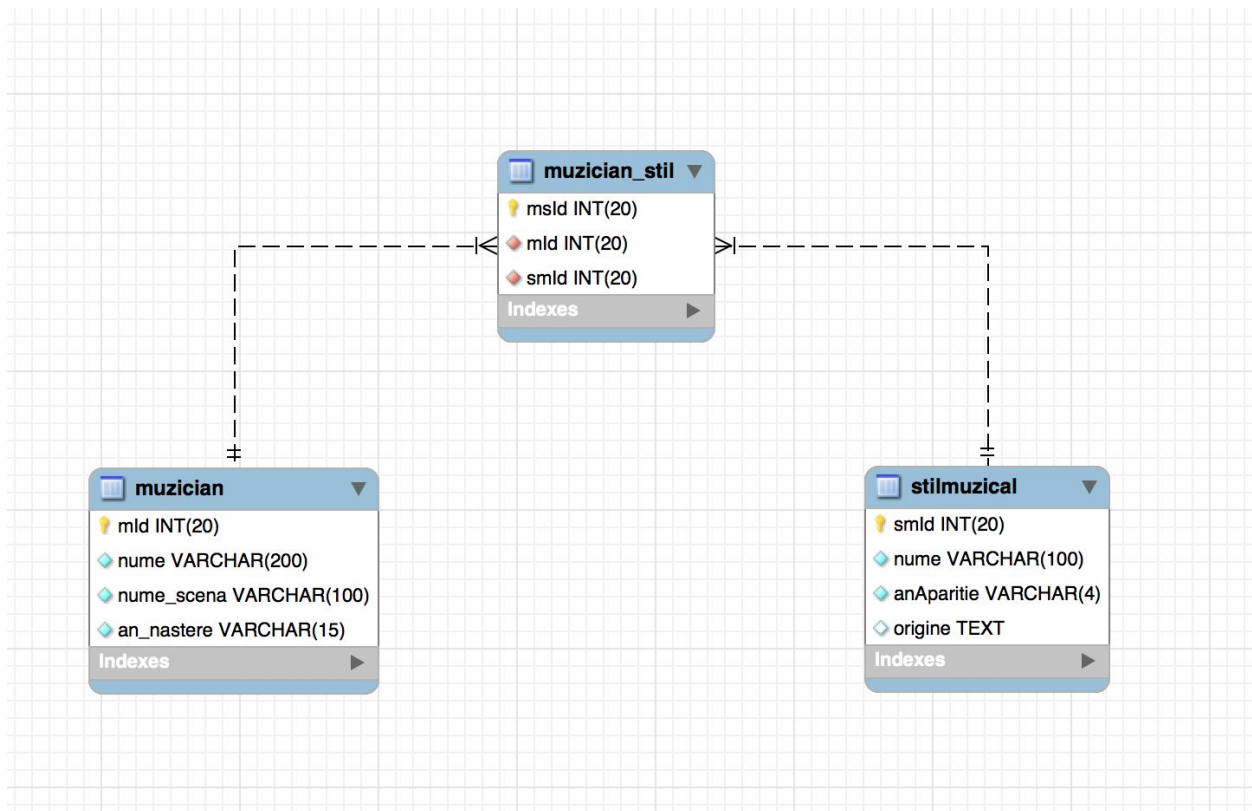


Figura 2.

## V.Funcții NodeJS & Prezentare Vizuală (HTML, CSS, JAVASCRIPT)

Funcțiile principale NodeJS folosite în această aplicație au fost cele de interogare pentru vizualizare, pentru creare, actualizare și ștergere.

Pentru toate aceste funcții a fost definit un fișier database.js ce conține credențialele de logare pentru baza de date, cât și funcția de deschidere a unei conexiuni către aceasta.

```
1  let mysql = require('mysql');
2  let db = mysql.createConnection({
3    host: 'localhost',
4    user: 'root',
5    password: 'admin',
6    database: 'database'
7  });
8
9  db.connect(function (err) {
10   if (err) {
11     console.error('error connecting: ' + err.stack);
12     return;
13   }
14
15   console.log('===== connected to database =====');
16 });
17
18 module.exports = db;
```

Figura 3.

“Node.js este o platformă construită pe motorul V8 al Chrome și se bazează pe un model event-driven și non-blocking I/O, folosind callback-uri. Este construită pe runtime-ul JavaScript de la Chrome, favorizează modularitatea, iar aplicațiile sale single-thread sunt scrise în JavaScript.”<sup>2</sup>

Cu ajutorul Nodejs am dezvoltat partea de server, iar cu tehnologia bootstrap am făcut elementele din pagină ( formular, tabele, butoane).

Partea de server se găsește în fișierele: controllers și routes.

Fișierul controllers este, de fapt, funcționalitatea din spatele rutelor.

Bootstrap este un framework gratuit și cu sursă deschisă (engleză: open-source) folosit în dezvoltarea site-urilor și a aplicațiilor web. Acesta conține modele HTML și CSS pentru tipografie, formularul, butoane, meniuri de navigare și multe alte componente. Pe lângă acestea există și extensii JavaScript ce aduc interactivitate paginilor create.

Bootstrap este folosit doar pentru crearea interfețelor. Bootstrap a fost dezvoltat în cadrul Twitter

ca un framework ce urmărea să prezinte un model de bază pentru crearea aplicațiilor interne.

“Node.js este un mediu de rulare cu sursă deschisă (open-source) ce permite crearea de aplicații de rețea scalabile în partea de server. Mediul de rulare este același motor de rulare JavaScript, același folosit de navigatorul de internet Google Chrome pe partea de client, V8 Javascript Runtime.

Modulele și librăriile Node.js oferă posibilitatea:

- scalării aplicațiilor pe măsură ce traficul crește;
- creării unor camere de conversații online sau a altor aplicații ce se bazează pe multe informații;
- manipulării directe a bazelor de date;
- accesării sistemului de fișiere;
- monitorizării traficului;
- încărcării rapide a fișierelor și afișarea unui progres real.

Dat fiind faptul că motorul JavaScript V8 și Node.js sunt scrise în C și C++ (dar și JavaScript), face ca Node.js să fie foarte rapid. Acesta este un lucru important atunci când o aplicație trebuie să fie scalabilă. Acest mediu permite operarea unui număr foarte mare de conexiuni simultane și asincrone. Aceste caracteristici permit scalabilitatea. Pe lângă asta există librării care pot fi personalizate pentru tratarea acestor probleme.”<sup>2</sup>

```

1  let musicDB = global.db;
2
3  let getAll = function () {
4    return new Promise((resolve, reject) => {
5      musicDB.query(`select ms.msId, m.nume as artist, m.nume_scena, m.an_nastere, sm.nume as stil, sm.anAparitie, sm.origine
6        from muzician_stil ms
7        left join muzician m
8          on ms.mId = m.mId
9        left join stilmuzical sm
10       on ms.smId = sm.smId`, (err, rows) => {
11         if (err) {
12           return reject(`Can't list`);
13         }
14         return resolve(rows);
15       });
16     });
17  }
18
19  let insertOne = function (link) {
20    return new Promise((resolve, reject) => {
21      musicDB.query(`insert into muzician_stil set ?`, link, (err, result) => {
22        if (err) {
23          return reject(`Can't insert`);
24        }
25        return resolve(result);
26      });
27    });
28  }
29
30
31  let deleteOne = function(linkId) {
32    return new Promise((resolve, reject) => {
33      musicDB.query(`delete from muzician_stil where msId = ${linkId}`, (err, result) => {
34        if (err) {
35          return reject(`Can't delete`);
36        }
37        return resolve(result);
38      });
39    });
40  }
41
42  module.exports = {
43    getAll,
44    insertOne,
45    deleteOne
46  }

```

Figura 4.

Interfața realizată este una simplă, intuitivă alcătuită din 2 butoane de adăugare în cele 2 tabele : muzicieni și stilmuzical.

MusicDB - Artists				
<div>Artists   Genres   Links</div>				
<div>New Artist</div>				
#	Nume	Nume de Scena	An Nastere	Action
3	Adele Laurie Blue Adkins	Adele	5/5/1988	<div>EditDelete</div>
4	James Solomon McDonald	Solomon Burke	21.03.1940	<div>EditDelete</div>
5	Aretha Louise Franklin	Aretha Franklin	March 25, 1942	<div>EditDelete</div>
6	Louis Daniel Armstrong	Louis Armstrong	August 4, 1901	<div>EditDelete</div>

Figura 5.

MusicDB - Genres				
<div>Artists   Genres   Links</div>				
<div>New Genre</div>				
#	Nume	An Aparitie	Origine	Action
1	Blue-eyed soul	1960	United States	<div>EditDelete</div>
2	Soul	1950	United States	<div>EditDelete</div>
3	Jazz	1920	New Orleans, United States	<div>EditDelete</div>

Figura 6.

Accesand unul din link-uri vom fi direcționați către meniul de adăugare a elementelor dorite, meniu în care vom avea acces și la vizualizarea elementelor componente din tabela respectivă.



The image shows a web form titled "MusicDB - Artists". At the top, there are three navigation links: "Artists" (highlighted in red), "Genres", and "Links". Below the links, there are three input fields: "Nume" (Name), "Nume de Scena" (Stage Name), and "An Nastere" (Year of Birth). Each field is represented by a text label followed by a rectangular input box. At the bottom of the form, there are two buttons: a green "Save" button and a blue "Cancel" button.

MusicDB - Artists

[Artists](#) [Genres](#) [Links](#)

Nume

Nume de Scena

An Nastere

Save Cancel

Figura 7.



Pe lângă cele 2 butoane interfața cuprinde și 2 câmpuri de selectare artiști și stiluri muzicale abordate de aceștia pentru a completa în tabela de asociere muzician\_stil.

### MusicDB - Artists & Genres

[Artists](#) [Genres](#) [Links](#)

Muzician

Stil

Save

Cancel

#	Muzician	Nume de Scena	An Nastere	Stil	Origine	An Aparitie	
3	Adele Laurie Blue Adkins	Adele	5/5/1988	Blue-eyed soul	United States	1960	<a href="#">Unlink</a>
5	Adele Laurie Blue Adkins	Adele	5/5/1988	Soul	United States	1950	<a href="#">Unlink</a>
4	James Solomon McDonald	Solomon Burke	21.03.1940	Soul	United States	1950	<a href="#">Unlink</a>
6	Aretha Louise Franklin	Aretha Franklin	March 25, 1942	Soul	United States	1950	<a href="#">Unlink</a>
7	Aretha Louise Franklin	Aretha Franklin	March 25, 1942	Jazz	New Orleans, United States	1920	<a href="#">Unlink</a>
8	Louis Daniel Armstrong	Louis Armstrong	August 4, 1901	Jazz	New Orleans, United States	1920	<a href="#">Unlink</a>

Figura 8.

Am făcut în folderul public folderul javascripts ce conține fișierele artists.js, genres.js și main.js ce asigură funcționalitatea celor html.

## **VI. Bibliografie**

1. Russell J.T. Dyer, "Learning MySQL and MariaDB", O'Reilly Media, United States of America, aprilie 2015;
2. <https://rolisz.ro/2011/08/02/introducere-in-node-js>
3. [https://www.w3schools.com/Php/php\\_mysql\\_intro.asp](https://www.w3schools.com/Php/php_mysql_intro.asp)
4. [https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp);