

# Proiect Baze de Date

## Aplicație pentru evidența unor spectacole de stand up

### 1. Definirea cerintelor

Lista funcționalităților:

*1. Gestiunea Entităților (CRUD):* Aplicația permite operații de creare, citire, actualizare și ștergere din partea adminilor pentru entitățile principale: Spectacole, Artiști, Locații și Organizatori.

*2. Autentificare și Inregistrare:* Utilizatorii noi se pot înregistra prin completarea unui formular, contul fiind creat automat cu rolul de utilizator standard. Login-ul se bazează pe sesiuni HTTP, iar interfața se adaptează dinamic în funcție de rol (admin/user).

*3. Achiziție Bilete:* Spectatorii pot vizualiza evenimentele disponibile și pot achiziționa bilete, primind un cod unic de rezervare (UUID) generat de sistem.

*4. Validarea Datelor:* Sistemul include validări pentru a asigura integritatea datelor (ex: numarul de telefon să aibă 10 cifre, verificarea tipurilor numerice pentru preț și capacitate).

*5. Sortare Dinamică:* Tabelele de date permit sortarea ascendentă sau DESCENDENTĂ după orice coloană (Data, Preț, Nume, Locație), facilitând navigarea rapidă prin liste de date.

*6. Căutare și Filtrare:* Motor de căutare integrat care permite filtrarea rezultatelor după cuvinte cheie (Titlu, Locație, Organizator).

*7. Statistici în Timp Real:* Afisarea automată a ultimelor bilete vândute și a următoarelor evenimente programate în paginile de resurse

*8. Rapoarte Complexă:*

- Calculul vânzărilor totale per spectacol, filtrabil dinamic după perioadă.
- Generarea topului organizatorilor în funcție de încasările generate.
- Identificarea celui mai fidel spectator.
- Calculul procentual al gradului de ocupare al locațiilor, combinând date din trei tabele diferite.

## 2. Proiectarea bazei de date

### *Utilizator*

ID\_Utilizator – INT, PK, AI

Username – VARCHAR(50)

Parola – VARCHAR(255)

Rol – ENUM('admin', 'user')

ID\_Spectator – INT, FK

### *Spectator*

ID\_Spectator – INT, PK, AI

Nume\_Spectator – VARCHAR(100)

Email\_Spectator – VARCHAR(100)

Telefon\_Spectator – VARCHAR(15)

### *Artist*

ID\_Artist – INT, PK, AI

Nume\_Artist – VARCHAR(100)

Prenume\_Artist – VARCHAR(100)

Varsta – INT, UNSIGNED

Nationalitate – VARCHAR(50)

Experienta\_Ani – INT, UNSIGNED

### *Spectacol*

ID\_Spectacol – INT, PK, AI

Titlu – VARCHAR(100)

Data\_Spectacol – DATE

Ora – TIME

Pret\_Bilet – DECIMAL(6,2)

ID\_Locatie – INT, FK

ID\_Organizator – INT, FK

### *Locatie*

ID\_Locatie – INT, PK, AI

Nume\_Locatie – VARCHAR(100)

Adresa – VARCHAR(150)

Oras – VARCHAR(50)

Capacitate – INT

### *Organizator*

ID\_Organizator – INT, PK, AI

Nume\_Organizator – VARCHAR(100)

Email\_Organizator – VARCHAR(100)

Telefon\_Organizator – VARCHAR(15)

### Bilet

ID\_Bilet – INT, PK, AI  
 ID\_Spectacol – INT, FK  
 ID\_Spectator – INT, FK  
 Data\_Cumparare – DATETIME  
 Cod\_Bilet – VARCHAR(20), UNIQUE

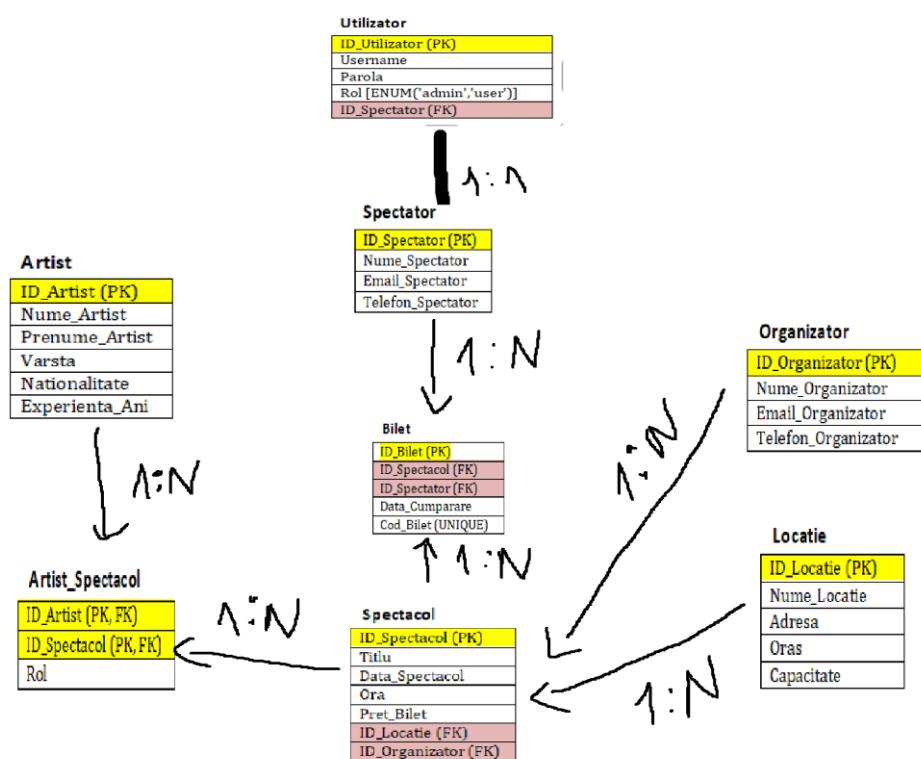
### Artist\_Spectacol

ID\_Artist – INT, PK/FK  
 ID\_Spectacol – INT,PK/ FK  
 Rol – VARCHAR(50)

## 3. Relațiile dintre tabele

Spectator – Utilizator: 1:1  
 Locatie – Spectacol : 1:N  
 Organizator – Spectacol : 1:N  
 Spectacol – Bilet : 1:N  
 Spectator – Bilet : 1:N  
 Artist – Spectacol : N:N

## Diagrama



## 4. Constragerile de integritate impuse

-Chei primare pentru fiecare tabel

-Chei străine pentru relațiile dintre tabele

-Validarea tipurilor de date(DECIMAL pentru prețuri, DATETIME pentru planificări și VARCHAR pentru text)

-Restricții pe valori:

\*Unicitate (UNIQUE): Pe câmpurile Username, Email, Telefon și Cod\_Bilet.

\*Domeniu (CHECK): Coloana Rol acceptă doar valorile 'admin' sau 'user'.

\*Validări logice (Backend): Telefonul trebuie să aibă exact 10 cifre, parola minim 5 caractere, iar email-ul să conțină @gmail.com.

## 5. Functionarea Aplicatiei

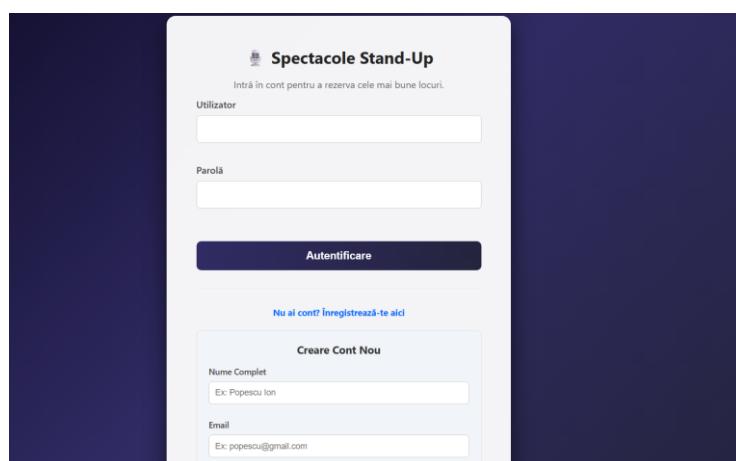
Aplicația este concepută pentru a gestiona evidența spectacolelor de stand-up, permitând administratorilor să gestioneze resursele și spectatorilor să achiziționeze bilete. Funcționarea acesteia implică următoarele componente și procese principale:

### A. Autentificare si înregistrare și conectare la Baza de Date

Aplicația se deschide pe pagina de login, unde utilizatorul se poate autentifica pe baza *username-ului* și *parolei*. Se verifică autentificarea prin conexiunea cu baza de date. Sistemul validează datele (regex pentru telefon, unicitate email/username) și inserează sincronizat înregistrările în două tabele: Spectator (datele de contact) și Utilizator (credențialele de acces).

În cazul în care utilizatorul nu are un cont, acesta își poate crea unul accesând butonul „Nu ai cont? Înregistrează-te aici”. La apasarea butonului utilizatorul își poate crea un cont completând informațiile din formularul ce apare.

Odată autentificat, utilizatorul primește un rol (admin sau user) care dictează vizibilitatea butoanelor de editare/ștergere în interfață.



## B. Administrare și gestiune date

Administratorul poate efectua operațiuni CRUD (Create, Read, Update, Delete) asupra entităților Spectacol, Artist, Locatie și Organizator.

**Integritate:** La ștergerea unui spectacol, sistemul gestionează automat dependențele, eliminând biletele asociate pentru a menține consistența bazei de date.

**Validări:** Orice adăugare de spectacol trece prin filtre logice (ex: prețul trebuie să fie pozitiv, data să fie validă).

Suplimentar, câteva pagini includ widget-uri informative în partea inferioară (ex: "Următoarele evenimente găzduite" sau "Ultimele colaborări"), generate prin interogări SQL specifice.

## C. Experiența spectatorului

Utilizatorii standard au acces la vizualizarea ofertei de spectacole.

**Mecanismul de căutare:** Interfața permite filtrarea rapidă a evenimentelor după titlu, locație sau organizator, folosind interogări SQL parametrizate cu clauza LIKE.

**Achiziția de bilete:** La apăsarea butonului "Cumpără", sistemul generează un UUID unic (Universally Unique Identifier) pentru bilet și îl salvează în baza de date cu marcajul temporal curent (CURRENT\_TIMESTAMP), simulând o tranzacție reală.

## D. Raportare și analiză

Această componentă demonstrează complexitatea interogărilor SQL implementate:

*Interogări Simple:* Pentru afișarea tabelelor curente, aplicația folosește JOIN-uri multiple pentru a prezenta date lizibile (ex: Numele Locației în loc de ID\_Locatie).

*Interogări Complexe:* Secțiunea "Raport" rulează interogări avansate care utilizează:

\*Subcereri în clauza FROM pentru calculul vânzărilor totale.

\*Subcereri corelate pentru determinarea gradului de ocupare al sălilor.

\*Funcții Agregat (COUNT, SUM) și grupări pentru topul organizatorilor și identificarea celui mai fidel spectator.

## 6. Interogari simple cu JOIN

- **Spectacole și detalii organizare (query 1):** SELECT cu LEFT JOIN pe 3 tabele (Spectacol, Locatie, Organizator) și sortare dinamică.
- **Bilete și detalii spectatori (query 2):** SELECT cu LEFT JOIN pe 3 tabele (Bilet, Spectacol, Spectator) cu filtrare parametrizată.
- **Widget Admin: Ultimele bilete vândute (query 3):** SELECT cu JOIN pe 3 tabele (Bilet, Spectacol, Spectator) folosind clauza TOP 3 și sortare descrescătoare.
- **Locații și evenimente viitoare (query 4):** JOIN între Locatie și Spectacol cu clauza TOP 5 și funcție de dată (GETDATE).
- **Organizatori și istoric evenimente (query 5):** JOIN între Organizator și Spectacol cu TOP 5 și ORDER BY descrescător.
- **Listă control rapidă: Spectacole recente (query 6):** SELECT simplu cu un JOIN între Spectacol și Locatie, limitat la ultimele 5 înregistrări.

## 7. Interogari complexe

- **Vânzări totale și sume per spectacol (complex query 1):** Tabelă derivată (Subcerere în FROM), COUNT, SUM, GROUP BY și filtrare pe interval de date (BETWEEN).
- **Top organizatori după încasări (complex query 2):** Subcerere în clauza WHERE (IN), funcție agregat SUM, GROUP BY și ORDER BY.
- **Identificarea celui mai fidel spectator (complex query 3):** Subcerere scalară în SELECT, Subcerere cu TOP 1 în WHERE și GROUP BY.
- **Gradul de ocupare procentual al locațiilor (complex query 4):** Subcereri corelate în lista SELECT, funcția COUNT și operații aritmetice.