DOCUMENTAȚIA PROIECTULUI

EventsApp

STUDENT

Buta Catalin

**Cuprins**

1. Prezentarea proiectului, ce isi propune, ce probleme rezolva

2. Tehnologiile folosite

3. Baza de date: diagrama bazei de date + scurta prezentare a tabelelor si a relatiilor dintre ele

4. Prezentarea API-ului: screenshot la Swagger pentru a putea vedea toate endpoint-ruile + scurta descriere a CRUD-urilor

5. Prezentare despre cum poate fi utilizata aplicatia: tipuri de utilizatori, ce vede fiecare, autentificare etc

6. Concluzii si contributii (cum v-ati impartit task-urile, ce ati invatat in urma acestui proiect etc)

7. Link GIT catre codul proiectului

# Prezentarea proiectului, ce isi propune, ce probleme rezolva

EventsApp este o aplicatie de creare de evenimente pe diferite categorii de interes la care se pot inscrie mai multe persoane. Aceste functionalitati sunt implementate intr-o maniera simpla.

Probleme rezolvate:

* Proces simplu de inregistrare si autentificare in aplicatie
* Operatii CRUD pentru crearea, vizualizarea, editarea si stergerea evenimentelor
* Posibilitatea proprietarilor evenimentelor de a adauga/sterge participanti.

# Tehnologiile folosite

**Limbaje de programare:** C#

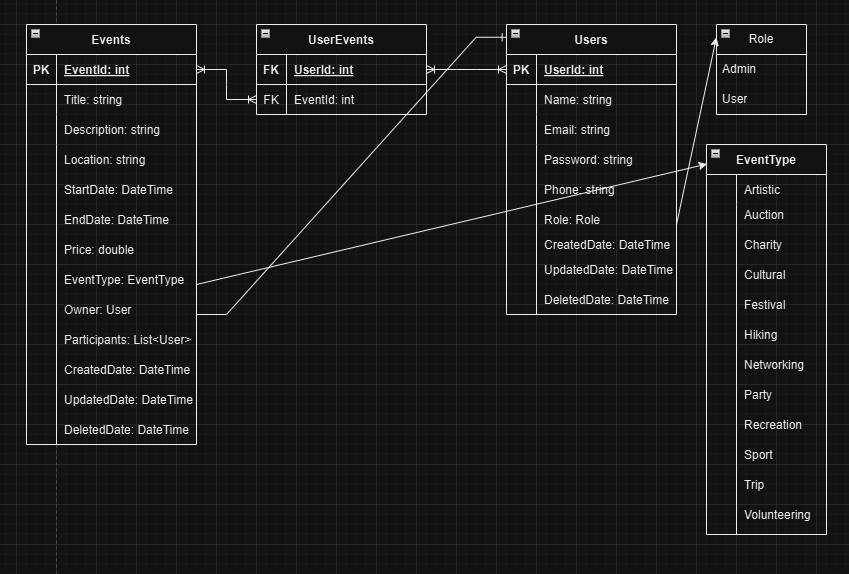
**Platforme de dezvoltare:** .NET

**Frameworks:** ASP.NET Core

**Baze de date:** SQLServer

**Sistem de versionare:** Git

# Baza de date

****

Entitati:

* User - proprietati: UserId, Name, Email, PasswordSalt, PasswordHash, Phone, Role (User, Admin), CreatedDate, UpdatedDate, DeletedDate
* Event – proprietati: EventId, Title, Description, Location, StartDate, EndDate, Price, EventType (Artistic, Auction, Charity, Cultural, Festival, Hiking, Networking, Party, Recreation, Sport, Trip, Voolunteering), Owner (User), Participants (List<UserEvent>)

Relatii intre tabele:

* Event <-> User (din perspectiva proprietarului)

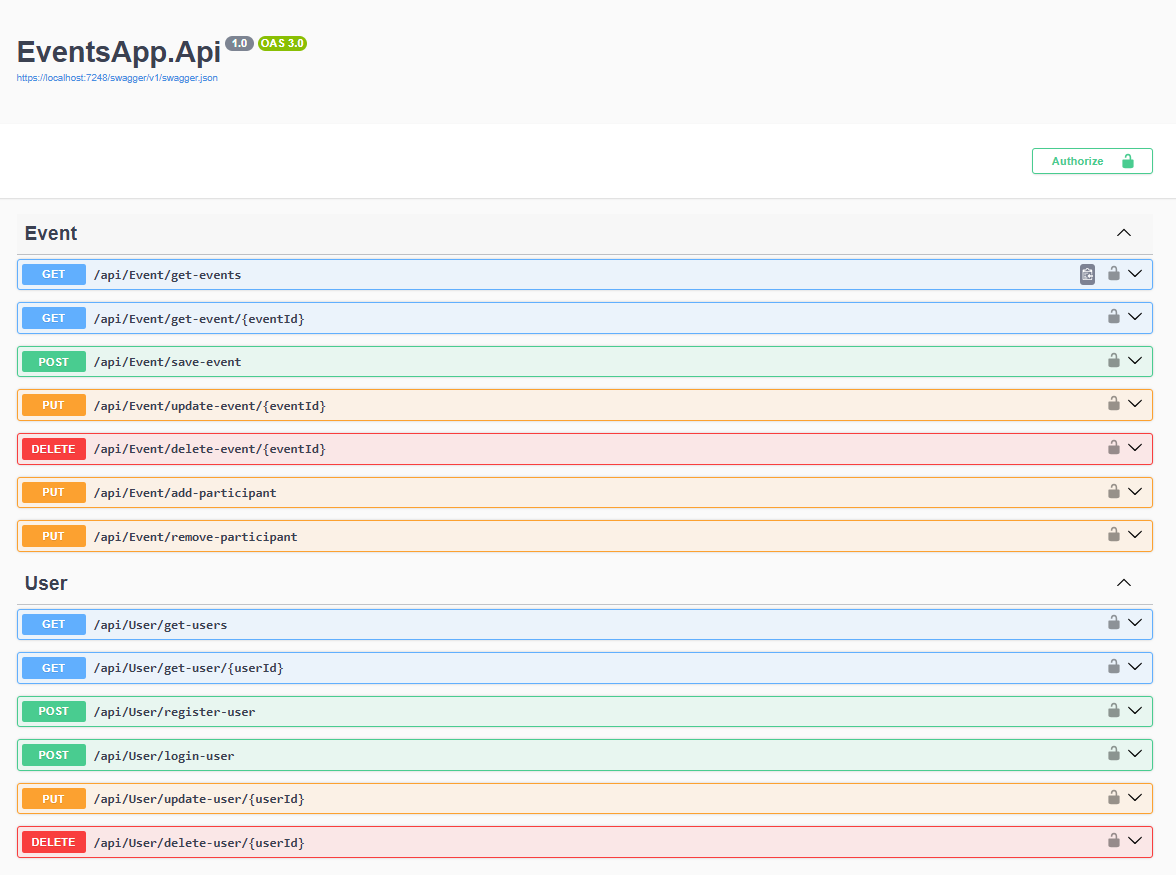
Un eveniment are un singur proprietar, iar acelasi user poate fi proprietar pentru mai multe evenimente. => relatie ManyToOne

* Event <-> User (din perspectiva participantilor)

La un eveniment pot participa mai multe persoane, iar o persoana poate participa la mai multe evenimente => relatie ManyToMany

Prin transformarea clasei de model UserEvent de catre Entity Framework Core s-a creat tabelul de legatura UserEvents in baza de date. Acest tabel contine 2 chei straine (una este legata de tabelul User si cealalta de tabelul Event).

# Prezentarea API-ului



Descriere:

Event:

* GET /api/Event/get-events: Obtine toate evenimentele.
* GET /api/Event/get-event/{eventId}: Obtine un eveniment specific pe baza unui ID.
* POST /api/Event/save-event: Salveaza un nou eveniment.
* PUT /api/Event/update-event/{eventId}: Modifica un eveniment existent pe baza ID-ului.
* DELETE /api/Event/delete-event/{eventId}: Sterge un eveniment pe baza ID-ului.
* PUT /api/Event/add-participant: Adauga un participant la un eveniment.
* PUT /api/Event/remove-participant: Elimina un participant de la un eveniment.

User:

* GET /api/User/get-users: Obtine toti utilizatorii.
* GET /api/User/get-user/{userId}: Obtine un utilizator specific pe baza unui ID.
* POST /api/User/register-user: Inregistreaza un nou utilizator.
* POST /api/User/login-user: Autentifica un utilizator.
* PUT /api/User/update-user/{userId}: Actualizeaza un utilizator pe baza ID-ului.
* DELETE /api/User/delete-user/{userId}: Sterge un utilizator pe baza ID-ului.

# Utilizarea aplicatiei

1. La pornirea aplicatiei se deschide in browser Swagger-UI care contine documentatia REST API-ului.
2. Pentru a inregistra un user nou, trimitem un HTTP POST request la /api/User/register-user. Detaliile de inregistrare sunt puse in request body.
3. Dupa inregistrare ne autentificam pentru a obtine un JSON Web Token necesar pentru a autoriza restul request-urilor din aplicatie.
4. Pentru toate celelalte request-uri in afara de register-user si login-user trebuie sa fim autorizati (pe request header trebuie sa existe cheia “Authorization” cu un Json Web Token ca si valoare.
5. Userii autentificati nu pot modifica decat propriile lor date. In caz contrar se arunca exceptie. Similar pentru evenimente, userii autentificati nu-si pot modifica decat propriile evenimente.
6. Doar adminii pot sterge useri sau evenimente din aplicatie.
7. Userii autentificati pot adauga sau sterge participanti doar de la propriile evenimente.

# Concluzii si contributii

Lucrand la acest proiect, am invatat ce se intampla pe partea de server a unei aplicatii. Am aflat cum sa fiu flexibil cand structurez datele trimise pe request si cele primite pe response cu ajutorul DTO si cum functioneaza autentificarea si autorizarea cu Json Web Token.

# Link Git

https://github.com/CatalinButa/EventsApp