

# DOCUMENTAȚIA PROIECTULUI

Project Manager API

STUDENȚI

Roman Onisim

Sîrboiu Filip

# Cuprins

<b>Prezentare generala proiectului.....</b>	<b>4</b>
<b>Tehnologii folosite.....</b>	<b>6</b>
ASP.NET Core.....	6
ASP.NET Core este un framework open-source dezvoltat de Microsoft pentru construirea de aplicații web moderne, conectate la internet, inclusiv API-uri RESTful.....	6
Entity Framework Core.....	6
PostgreSQL.....	6
PostgreSQL este un sistem de gestionare a bazelor de date relaționale open-source, cunoscut pentru fiabilitatea, robustețea și caracteristicile sale avansate. A fost ales ca SGBD pentru acest proiect datorită:.....	6
○ Suport pentru standarde SQL: PostgreSQL oferă suport extins pentru standardele SQL, permițând utilizarea de funcționalități avansate și interogări complexe.....	6
○ Extensibilitate: PostgreSQL permite adăugarea de funcții definite de utilizator, tipuri de date personalizate și alte extensii.....	6
○ Integritate și securitate: Oferă mecanisme robuste de integritate referențială și controale avansate de acces și permisiuni.....	6
Swagger.....	7
Visual Studio (+ Visual Studio Code).....	7
Visual Studio este un IDE (Integrated Development Environment) dezvoltat de Microsoft, folosit pentru dezvoltarea aplicațiilor .NET.....	7
Visual Studio Live Share.....	7
<b>Baza de date.....</b>	<b>8</b>
Tabelele bazei de date.....	9
Users.....	9
Projects.....	9
Tasks.....	9
Relațiile dintre tabele.....	10
<b>Prezentarea API-ului.....</b>	<b>11</b>
Swagger Screenshots.....	11
Descrierea CRUD-urilor.....	12
ProjectController.....	12
TaskController.....	12
UserController.....	13
<b>Prezentare a utilizării aplicației.....</b>	<b>14</b>

Universitatea Transilvania din Braşov  
Facultatea de Matematică şi Informatică

Tipuri de utilizatori.....	14
1. Admin:.....	14
2. Moderator:.....	14
3. Utilizator:.....	14
Autentificare.....	14
1. Înregistrare (Register):.....	14
2. Autentificare (Login):.....	14
Vizualizare şi gestionare date.....	15
Admin.....	15
Moderator.....	15
Utilizator.....	15
Exemplu de utilizare.....	15
1. Înregistrarea unui nou utilizator:.....	15
2. Autentificarea:.....	15
3. Gestionarea proiectelor şi task-urilor:.....	16
4. Actualizarea şi ştergerea:.....	16
<b>Concluzii si contributii.....</b>	<b>17</b>
Contributii.....	17
Ce am învăţat în urma acestui proiect.....	17
<b>Link către codul proiectului.....</b>	<b>18</b>

## Prezentare generala proiectului

Proiectul nostru, intitulat **Project Manager API**, este conceput pentru a oferi o soluție completă și eficientă pentru gestionarea proiectelor și a sarcinilor asociate acestora. API-ul nostru permite utilizatorilor să gestioneze toate aspectele unui proiect, de la crearea acestuia până la alocarea sarcinilor și urmărirea progresului.

Obiectivul principal al **Project Manager API** este de a facilita procesul de management al proiectelor printr-o interfață programatică robustă și ușor de utilizat. Prin intermediul acestui API, utilizatorii pot efectua operațiuni CRUD (Create, Read, Update, Delete) pe proiecte și sarcini, asigurându-se astfel că toate informațiile relevante sunt centralizate și accesibile într-un mod structurat.

API-ul este proiectat să fie scalabil și flexibil, astfel încât să poată fi integrat cu ușurință în diverse aplicații sau platforme existente. De asemenea, API-ul oferă suport pentru autentificare și autorizare, asigurând astfel securitatea și integritatea datelor.

Printre problemele rezolvate de acest proiect se numără:

- Lipsa de centralizare a proiectelor
- Dificultăți în urmărirea progresului
- Comunicare și colaborare ineficientă
- Securitatea datelor

## Tehnologii folosite

În dezvoltarea **Project Manager API**, am utilizat un set variat de tehnologii moderne pentru a asigura eficienţa, scalabilitatea şi securitatea aplicaţiei noastre. Aceste tehnologii ne-au permis să creăm o soluţie robustă şi uşor de întreţinut, capabilă să răspundă nevoilor diverse ale utilizatorilor.

### ASP.NET Core

**ASP.NET Core** este un framework open-source dezvoltat de Microsoft pentru construirea de aplicaţii web moderne, conectate la internet, inclusiv API-uri RESTful.

### Entity Framework Core

**Entity Framework Core (EF Core)** este un ORM (Object-Relational Mapper) pentru .NET, care permite dezvoltatorilor să lucreze cu o bază de date utilizând obiecte .NET. EF Core a fost utilizat în acest proiect pentru a simplifica interacţiunea cu baza de date PostgreSQL.

### PostgreSQL

**PostgreSQL** este un sistem de gestionare a bazelor de date relaţionale open-source, cunoscut pentru fiabilitatea, robusteţea şi caracteristicile sale avansate. A fost ales ca SGBD pentru acest proiect datorită:

- **Suport pentru standarde SQL:** PostgreSQL oferă suport extins pentru standardele SQL, permiţând utilizarea de funcţionalităţi avansate şi interogări complexe.
- **Extensibilitate:** PostgreSQL permite adăugarea de funcţii definite de utilizator, tipuri de date personalizate şi alte extensii.
- **Integritate şi securitate:** Oferă mecanisme robuste de integritate referenţială şi controale avansate de acces şi permisiuni.

## Swagger

**Swagger** este un set de instrumente open-source pentru documentarea şi testarea API-urilor. Am utilizat Swagger pentru a genera documentaţia interactivă a API-ului nostru, oferind următoarele avantaje:

- **Interfaţă interactivă:** Swagger UI oferă o interfaţă web interactivă unde utilizatorii pot explora şi testa endpoint-urile API-ului fără a fi nevoie de un client separat.
- **Documentaţie automată:** Anotările din cod sunt utilizate pentru a genera automat documentaţia API-ului, asigurând astfel consistenţa şi actualizarea continuă a acesteia.
- **Facilitarea testării:** Swagger permite testarea rapidă a endpoint-urilor API direct din interfaţa web, ceea ce este util pentru dezvoltatori şi testerii.

## Visual Studio (+ Visual Studio Code)

**Visual Studio** este un IDE (Integrated Development Environment) dezvoltat de Microsoft, folosit pentru dezvoltarea aplicaţiilor .NET.

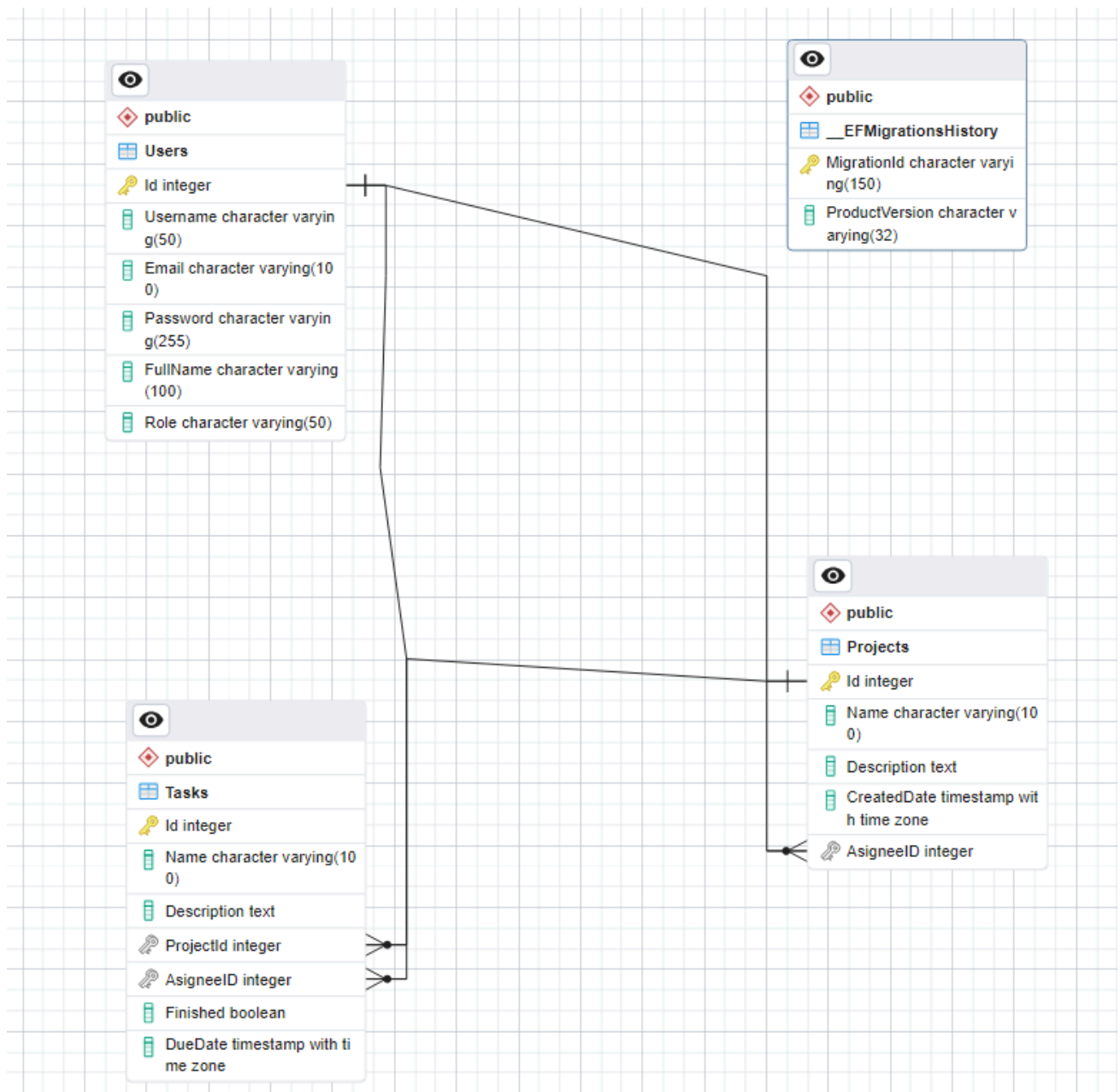
## Visual Studio Live Share

**Visual Studio Live Share** este o extensie pentru Visual Studio şi Visual Studio Code care permite colaborarea în timp real între dezvoltatori. Aceasta a fost esenţială pentru proiectul nostru, deoarece:

- **Colaborare în timp real:** Permite mai multor dezvoltatori să editeze acelaşi fişier simultan, să ruleze şi să debugeze cod împreună.
- **Partajare sesiunii de cod:** Dezvoltatorii pot partaja sesiunea lor de cod cu alţii fără a fi nevoie să îşi împărtăşească întregul mediu de dezvoltare.
- **Suport pentru chat şi apeluri audio:** Integrarea cu servicii de comunicare permite dezvoltatorilor să discute şi să colaboreze direct în timpul sesiunilor de Live Share.

## Baza de date

Diagrama bazei de date este prezentată mai jos şi include tabelele principale utilizate în **Project Manager API**: **Users**, **Projects**, **Tasks** şi **\_\_EFMigrationsHistory**.



## Tabelele bazei de date

### Users

- **Id:** Cheia primară, identificator unic pentru fiecare utilizator.
- **Username:** Numele de utilizator, de tip `character varying(50)`.
- **Email:** Adresa de e-mail a utilizatorului, de tip `character varying(100)`.
- **Password:** Parola utilizatorului, de tip `character varying(255)`.
- **FullName:** Numele complet al utilizatorului, de tip `character varying(100)`.
- **Role:** Rolul utilizatorului în sistem, de tip `character varying(50)`.

### Projects

- **Id:** Cheia primară, identificator unic pentru fiecare proiect.
- **Name:** Numele proiectului, de tip `character varying(100)`.
- **Description:** Descrierea proiectului, de tip `text`.
- **CreatedDate:** Data creării proiectului, de tip `timestamp with time zone`.
- **AssigneeID:** Cheia străină către tabela `Users`, reprezentând utilizatorul responsabil de proiect.

### Tasks

- **Id:** Cheia primară, identificator unic pentru fiecare task.
- **Name:** Numele task-ului, de tip `character varying(100)`.
- **Description:** Descrierea task-ului, de tip `text`.
- **ProjectId:** Cheia străină către tabela `Projects`, reprezentând proiectul căruia îi aparține task-ul.
- **AssigneeID:** Cheia străină către tabela `Users`, reprezentând utilizatorul responsabil de task.
- **Finished:** Stare de finalizare a task-ului, de tip `boolean`.
- **DueDate:** Data limită pentru finalizarea task-ului, de tip `timestamp with time zone`.



## Relațiile dintre tabele

- Relația între **Users** și **Projects** este realizată prin coloana **AsigneeID** din tabela **Projects**, care se referă la coloana **Id** din tabela **Users**. Aceasta relație indică faptul că fiecare proiect are un utilizator responsabil (asignee). Acesta este gândit ca fiind responsabilul proiectului.
- Relația între **Users** și **Tasks** este realizată prin coloana **AsigneeID** din tabela **Tasks**, care se referă la coloana **Id** din tabela **Users**. Aceasta relație indică faptul că fiecare task are un utilizator responsabil (asignee). Acesta este gândit ca fiind responsabilul task-ului.
  - Principiul acesta este gândit ca la Jira - un task poate avea doar un responsabil. Dacă un task trebuie să aibă mai mulți responsabili, atunci este semn că task-ul este prea mare și trebuie spart în mai multe task-uri, care vor avea fiecare câte un responsabil.
- Relația între **Projects** și **Tasks** este realizată prin coloana **ProjectId** din tabela **Tasks**, care se referă la coloana **Id** din tabela **Projects**. Aceasta relație indică faptul că fiecare task aparține unui proiect specific.

## Prezentarea API-ului

API-ul nostru oferă o serie de endpoint-uri pentru gestionarea utilizatorilor, proiectelor şi task-urilor. Mai jos este prezentată structura principalelor controlere şi operaţiunile CRUD disponibile.

### Swagger Screenshots

Project			^
GET	/api/Project/getProjectById		🔒 ▼
GET	/api/Project/getAllProjectsByUserId		🔒 ▼
GET	/api/Project/getAllProjects		🔒 ▼
POST	/api/Project/addProject		🔒 ▼
PUT	/api/Project/updateProject		🔒 ▼
DELETE	/api/Project/deleteProject		🔒 ▼
Task			^
GET	/api/Task/getTaskById		🔒 ▼
GET	/api/Task/getTasksByProjectId		🔒 ▼
POST	/api/Task/addTask		🔒 ▼
PUT	/api/Task/updateTask		🔒 ▼
DELETE	/api/Task/deleteTask		🔒 ▼
User			^
POST	/api/User/register		🔒 ▼
POST	/api/User/login		🔒 ▼
GET	/api/User/getUserById		🔒 ▼
GET	/api/User/getAllUsers		🔒 ▼
PUT	/api/User/updateUser		🔒 ▼
DELETE	/api/User/deleteUser		🔒 ▼



## Descrierea CRUD-urilor

### ProjectController

- **GET /api/project/getProjectById:** Obține detaliile unui proiect specific prin ID.
- **GET /api/project/getAllProjectsByUserId:** Obține toate proiectele asociate unui utilizator specific.
- **GET /api/project/getAllProjects:** Obține toate proiectele, cu opțiuni de sortare.
- **POST /api/project/addProject:** Adaugă un proiect nou. Doar utilizatorii cu rolurile Admin și Moderator pot accesa acest endpoint.
- **PUT /api/project/updateProject:** Actualizează un proiect existent. Doar utilizatorii cu rolurile Admin și Moderator pot accesa acest endpoint.
- **DELETE /api/project/deleteProject:** Șterge un proiect existent. Doar utilizatorii cu rolul Admin pot accesa acest endpoint.

### TaskController

- **GET /api/task/getTaskById:** Obține detaliile unui task specific prin ID.
- **GET /api/task/getTasksByProjectId:** Obține toate task-urile asociate unui proiect specific.

- **POST /api/task/addTask:** Adaugă un task nou.
- **PUT /api/task/updateTask:** Actualizează un task existent. Doar utilizatorii cu rolurile Admin şi Moderator pot accesa acest endpoint.
- **DELETE /api/task/deleteTask:** Şterge un task existent. Doar utilizatorii cu rolul Admin pot accesa acest endpoint.

## UserController

- **POST /api/user/register:** Înregistrează un nou utilizator. Endpoint-ul este accesibil fără autentificare.
- **POST /api/user/login:** Autentifică un utilizator şi returnează un token JWT. Endpoint-ul este accesibil fără autentificare.
- **GET /api/user/getUserById:** Obţine detaliile unui utilizator specific prin ID. Doar utilizatorii cu rolurile Admin şi Moderator pot accesa acest endpoint.
- **GET /api/user/getAllUsers:** Obţine o listă cu toţi utilizatorii. Doar utilizatorii cu rolurile Admin şi Moderator pot accesa acest endpoint.
- **PUT /api/user/updateUser:** Actualizează detaliile unui utilizator existent. Doar utilizatorii cu rolurile Admin şi Moderator pot accesa acest endpoint.
- **DELETE /api/user/deleteUser:** Şterge un utilizator existent. Doar utilizatorii cu rolul Admin pot accesa acest endpoint.

# Prezentare a utilizarii aplicatiei

## Tipuri de utilizatori

Aplicația **Project Manager API** suportă trei tipuri principale de utilizatori, fiecare cu niveluri diferite de acces și permisiuni:

### 1. Admin:

- Are cele mai înalte permisiuni.
- Poate adăuga, actualiza și șterge utilizatori.
- Poate gestiona toate proiectele și task-urile (creare, citire, actualizare, ștergere).
- Poate vedea toate datele și face modificări în întreaga aplicație.

### 2. Moderator:

- Are permisiuni limitate comparativ cu Admin-ul.
- Poate adăuga și actualiza proiecte și task-uri.
- Poate vedea toate proiectele și task-urile.
- Nu poate șterge utilizatori sau alte entități, decât cele specificate.

### 3. Utilizator:

- Are permisiuni de bază.
- Poate vizualiza proiectele și task-urile asociate contului său.
- Poate crea și gestiona proiecte și task-uri proprii.
- Nu are acces la funcționalitățile administrative.

## Autentificare

Autentificarea în **Project Manager API** se face prin intermediul unui sistem de login care utilizează token-uri JWT (JSON Web Token). Procesul de autentificare include:

### 1. Înregistrare (Register):

- Endpoint: **POST /api/user/register**
- Permite unui nou utilizator să își creeze un cont.
- Detalii necesare: Username, Email, Parolă, FullName, Rol (opțional, implicit este User).

### 2. Autentificare (Login):

- Endpoint: **POST /api/user/login**
- Permite unui utilizator existent să se autentifice.
- Detalii necesare: Username, Parolă.

- Returnează un token JWT care este utilizat pentru accesul la endpoint-urile securizate.

## Vizualizare şi gestionare date

### Admin

- **Dashboard complet:** Adminul are acces la toate datele din sistem, inclusiv utilizatori, proiecte şi task-uri.
- **Gestionarea utilizatorilor:** Poate adăuga, actualiza şi şterge utilizatori.
- **Gestionarea proiectelor şi task-urilor:** Poate crea, citi, actualiza şi şterge orice proiect sau task.

### Moderator

- **Vizualizare proiecte şi task-uri:** Poate vedea toate proiectele şi task-urile.
- **Crearea şi actualizarea proiectelor şi task-urilor:** Poate adăuga şi actualiza proiecte şi task-uri.
- **Limitări:** Nu poate şterge utilizatori şi are permisiuni administrative limitate.

### Utilizator

- **Vizualizare proiecte şi task-uri proprii:** Poate vedea doar proiectele şi task-urile asociate contului său.
- **Crearea şi gestionarea proiectelor şi task-urilor proprii:** Poate adăuga şi actualiza proiecte şi task-uri proprii.
- **Limitări:** Nu are acces la datele şi funcţionalităţile administrative.

## Exemplu de utilizare

### 1. Înregistrarea unui nou utilizator:

- Utilizatorul completează detaliile necesare (Username, Email, Parolă, FullName) şi trimite cererea către endpoint-ul **POST /api/user/register**.
- Utilizatorul primeşte un mesaj de confirmare a înregistrării.

### 2. Autentificarea:

- Utilizatorul introduce Username şi Parolă şi trimite cererea către endpoint-ul **POST /api/user/login**.
- Utilizatorul primeşte un token JWT care va fi folosit pentru autentificare în cererile ulterioare.

**Available authorizations** ✕

**Bearer (apiKey)**

JWT Authorization header using the Bearer scheme. Example: 'Bearer <token\_placeholder>'

Name: Authorization

In: header

Value:

Authorize Close

### 3. Gestionarea proiectelor şi task-urilor:

- Utilizatorul autentificat poate crea proiecte şi task-uri folosind endpoint-urile **POST** `/api/project/addProject` şi **POST** `/api/task/addTask`.
- Utilizatorul poate vizualiza proiectele şi task-urile asociate contului său folosind endpoint-urile **GET** `/api/project/getAllProjects` şi **GET** `/api/task/getTasksByProjectId`.

### 4. Actualizarea şi ştergerea:

- Utilizatorul poate actualiza proiectele şi task-urile existente folosind endpoint-urile **PUT** `/api/project/updateProject` şi **PUT** `/api/task/updateTask`.
- Adminul poate şterge utilizatori, proiecte şi task-uri folosind endpoint-urile **DELETE** `/api/user/deleteUser`, **DELETE** `/api/project/deleteProject` şi **DELETE** `/api/task/deleteTask`.

# Concluzii si contributii

## Contributii

Proiectul **Project Manager API** a fost realizat în colaborare strânsă, utilizând Visual Studio Live Share + Discord pentru a facilita munca în echipă. În loc să împărţim strict task-urile între noi, am lucrat împreună la fiecare aspect al proiectului, colaborând în timp real la scrierea codului, testare şi rezolvarea problemelor.

## Ce am învăţat în urma acestui proiect

Acest proiect ne-a oferit oportunitatea de a ne aprofunda cunoştinţele şi abilităţile în mai multe domenii esenţiale pentru dezvoltarea de aplicaţii web şi API-uri. Printre cele mai importante lecţii învăţate se numără:

- **Colaborare eficientă:** Utilizarea Visual Studio Live Share ne-a permis să lucrăm în timp real, să discutăm idei şi să rezolvăm probleme împreună. Aceasta ne-a ajutat să îmbunătăţim abilităţile de colaborare şi comunicare în echipă.
- **Autentificare şi autorizare:** Implementarea unui sistem de autentificare bazat pe JWT ne-a oferit o înţelegere profundă a securităţii aplicaţiilor web, gestionarea sesiunilor şi protejarea resurselor API.
- **Entity Framework Core:** Am învăţat să utilizăm EF Core pentru a mapă entităţile la tabelele din baza de date, să realizăm migraţii şi să scriem interogări LINQ eficiente. Aceasta ne-a oferit o perspectivă asupra modului de lucru cu ORM-uri în aplicaţiile .NET.
- **Documentare şi testare API:** Utilizarea Swagger pentru documentarea şi testarea API-ului ne-a arătat importanţa unei documentaţii clare şi interactive pentru utilizatorii finali şi dezvoltatori. Am învăţat să adnotăm corect endpoint-urile şi să testăm interacţiunile API.
- **Managementul proiectelor:** Am avut ocazia să planificăm şi să gestionăm un proiect complet, de la definirea cerinţelor şi arhitecturii, până la implementare, testare şi documentare. Acest proces ne-a îmbunătăţit abilităţile de organizare şi gestionare a timpului.



## Link către codul proiectului

<https://github.com/FilipS14/ProjectManagerAPI>