Healthy Lifestyle Web Application

Documentatie

Am implementat o aplicatie web pentru urmarirea nutritiei si a activitatii fizice. In fiecare zi utilizatorii pot inregistra ce au mancat pentru a vedea cate calorii, carbohidrati, grasimi si proteine au consumat si ce activitati fizice au facut pentru a vedea cate calorii au ars. Tehnologiile utilizate sunt ASP.NET Core si React, iar baza de date folosita este Microsoft SQL Server.

1. User stories – pe trello

Authentication, Compute calories, Log food, Log physical activity, Add physical exercise, Choose physical exercises, Physical exercises recommendations, Add new food, Profile configuration, Add goal, Make new food public request, Delete users, Progress report

Un utilizator se poate autentifica si poate utiliza functionalitatile aplicatiei doar atunci cand este logat.

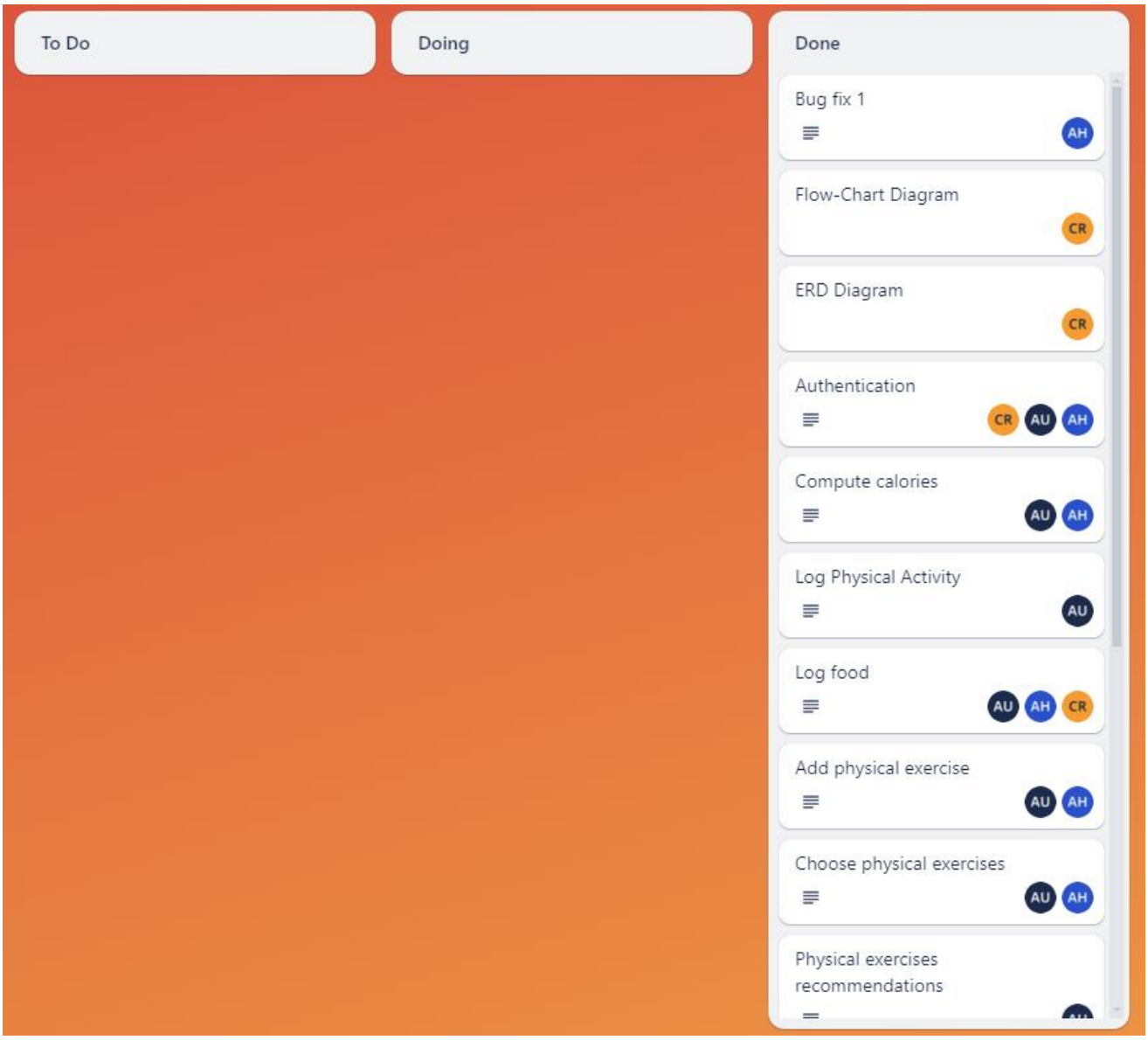
Utilizatorul poate sa si creeze un profil in care introduce date despre el (zi de nastere, greutate, inaltime) si scopul sau (ex: mentinere greutate).

In fiecare zi utilizatorul isi poate adauga mancarea consumata si activitatile fizica.

Utilizatorul poate vedea si un grafic cu progresul facut.

Utilizatorul poate crea o noua mancare si poate face cerere sa fie publice, care trebuie aprobata de un admin.

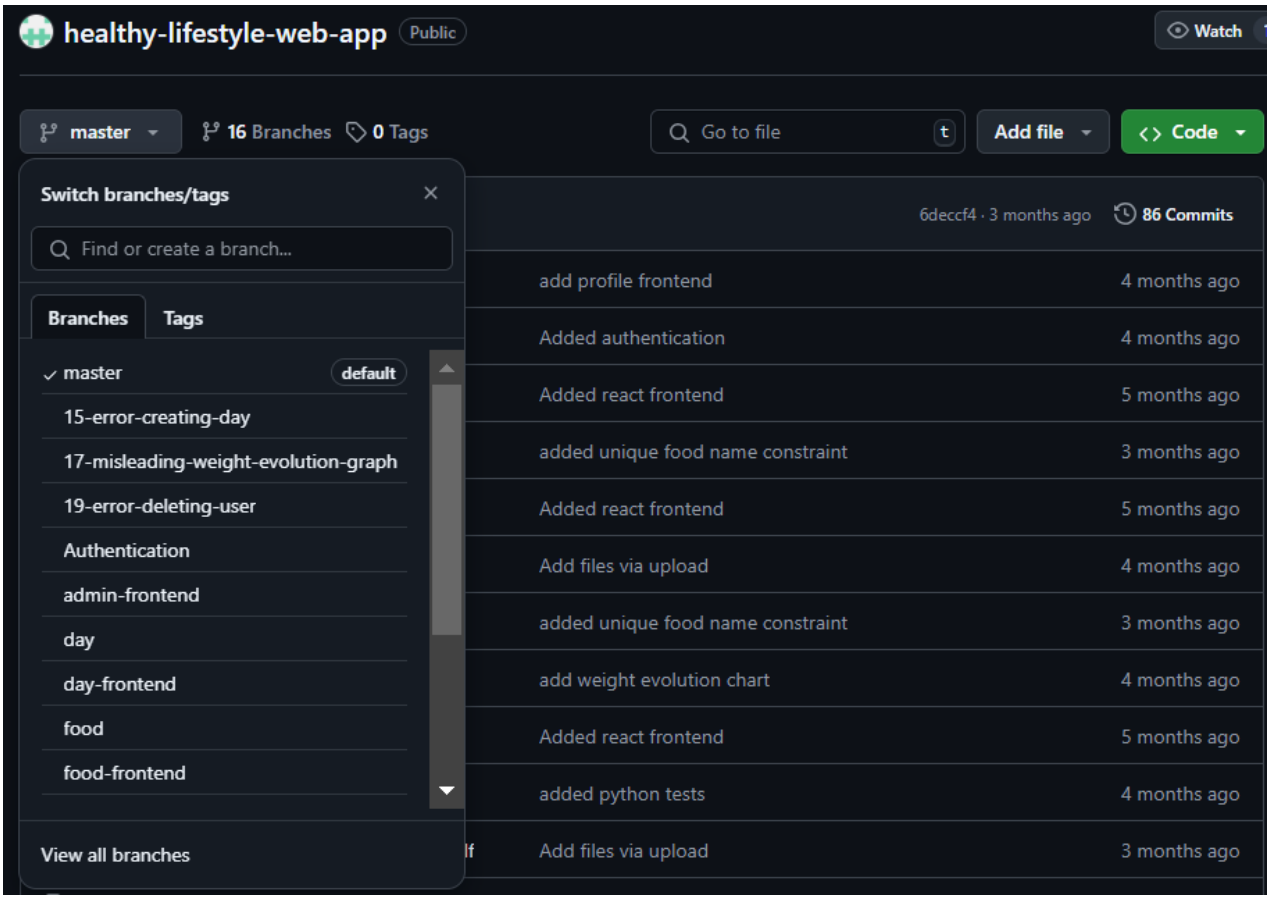
Adminii au acces la o pagina in plus in care pot vedea toate datele din aplicatie (utilizatori, mancare, activitati, request-uri), pot sterge utilizatori, aproba/ nega request-uri, adauga mancare sau activitati fizice, promova un utilizator la rolul de admin.


To Do
Doing
Done
Bug fix 1
AH
Flow-Chart Diagram
CR
ERD Diagram
CR
Authentication
CR AU AH
Compute calories
AU AH
Log Physical Activity
AU
Log food
AU AH CR
Add physical exercise
AU AH
Choose physical exercises
AU AH
Physical exercises recommendations

2. Source control cu git

Proiectul este pus pe github.

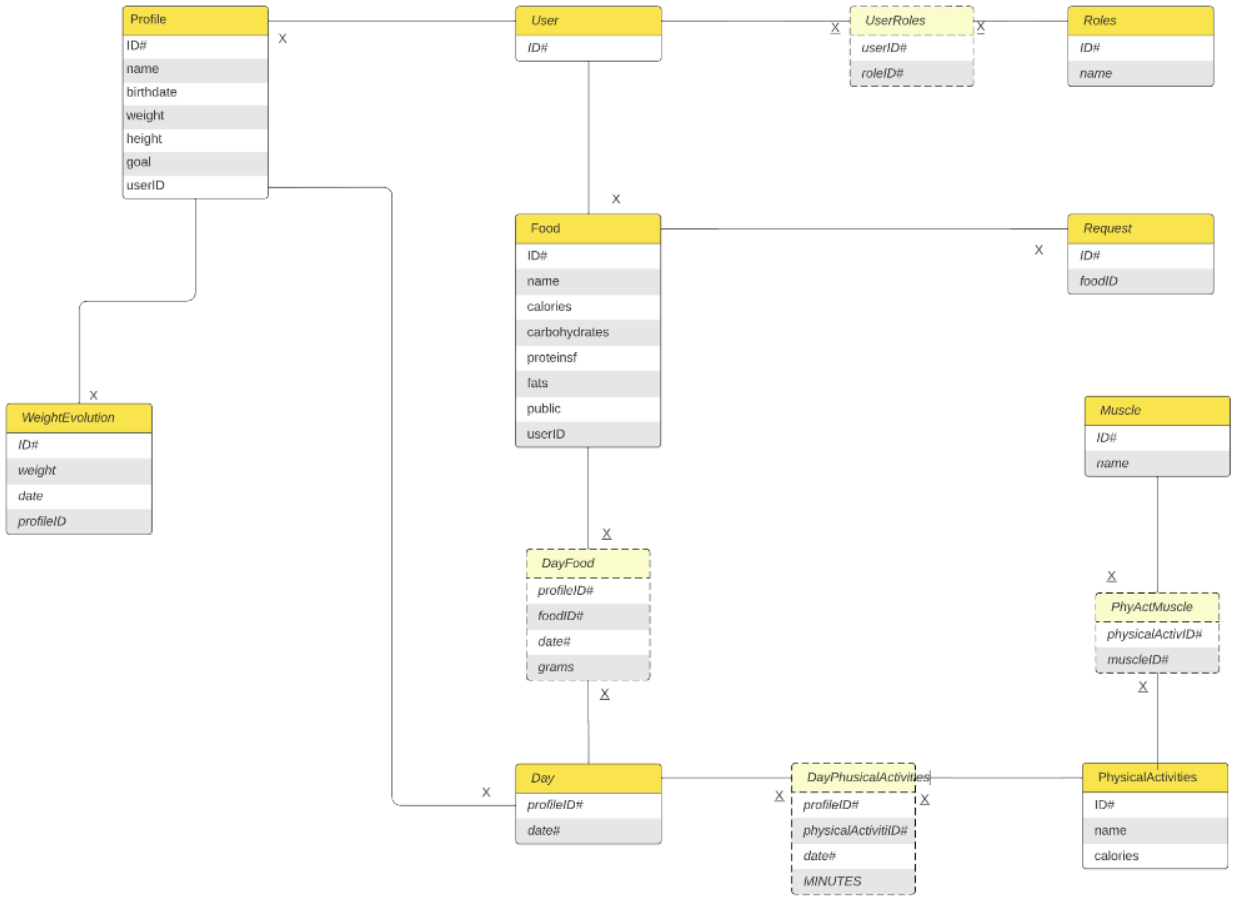
https://github.com/ana0101/healthy-lifestyle-web-app



Am impartit proiectul in parti (ex. profilul utilizatorului, mancare, request-uri) si le-am dezvoltat in branch-uri separate pentru care am realizat ulterior pull request pentru a fi unite in master.

3. Diagrame

Am realizat diagrama conceptuala a bazei de date si o diagrama flowchart care descrie utilizarea aplicatiei din perspectiva unui utilizator. Diagramele se afla pe github: https://github.com/ana0101/healthy-lifestyle-web-app/tree/master/diagrams



4. Teste automate

Am creat 5 teste automate folosind pytest care verifica anumite endpoint-uri atat in caz de succes cat si la esec.

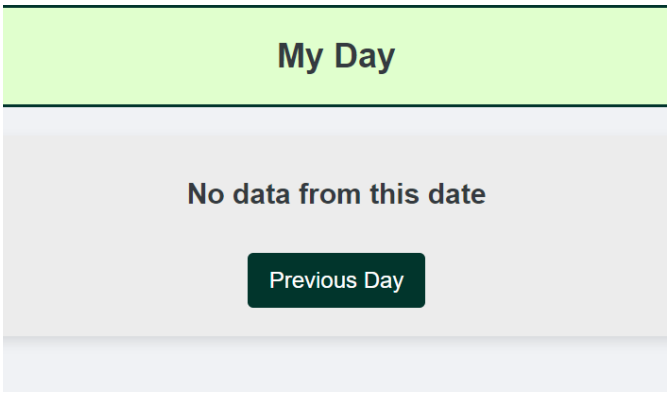
- test_admin_users – verifica autentificarea si operatiile pe care le poate efectua un administrator in legatura cu alti utilizatori (sters, promovat)
- test_food – verifica endpoint-urile care tin de mancare
- test_physical_activity – verifica endpoint-urile care tin de activitati fizice
- test_physical_activity_muscle – verifica endpoint-urile care tin de muschi si impreunarea muschilor cu activitati fizice
- test_profile – verifica endpoint-urile care tin de profilul utilizatorului (creare, modificare, stergere)

```
rootdir: D:\Universitate\MDS\Proiect\python_tests
collected 32 items

test_admin_users\test_admin_users.py .....                                [ 15%]
test_food\test_food.py ....                                               [ 28%]
test_physical_activity\test_physical_activity.py .......                  [ 50%]
test_physical_activity_muscle\test_physical_activity_muscle.py .......    [ 71%]
test_profile\test_profile.py .........                                    [100%]

============================================================ 32 passed in 3.09s ============================================================
```

5. Raportare bug si rezolvare cu pull request

- A fost raportat bug-ul error-creating-day, atunci cand un utilizator isi crea profilul dupa ce isi crea contul nu se genera o zi pentru cea curenta (zilele sunt create la 12 noaptea). A fost reparat astfel: ziua este creata cand este creat profilul.

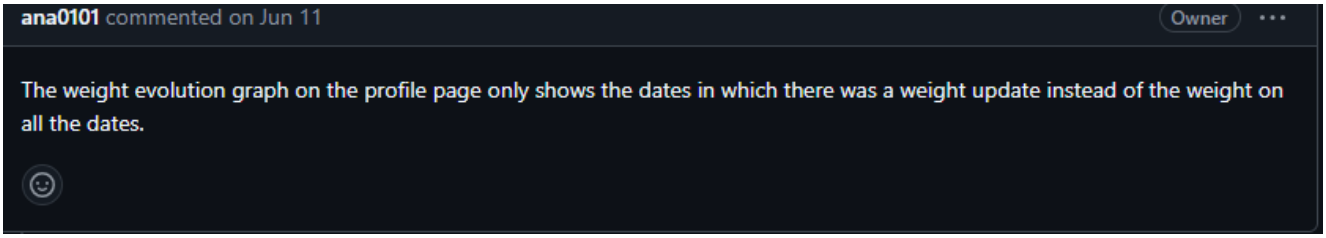
Imediat dupa crearea contului si a profilului:



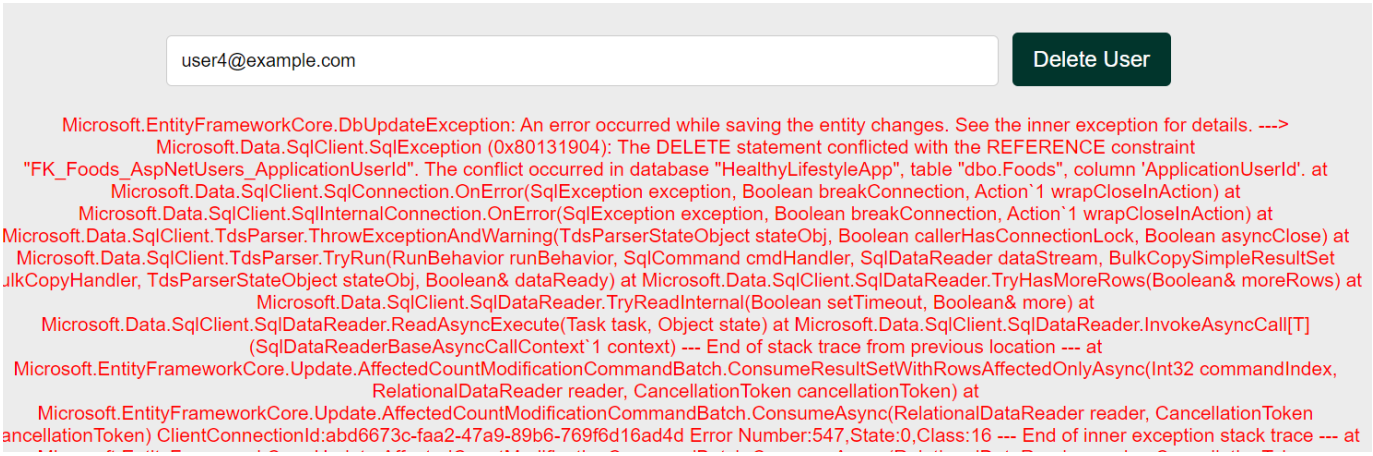
In functia pentru crearea profilului am creat si ziua:

```
if (await _profileRepository.PostAsync(profile))
{
    if (await _weightEvolutionRepository.PostAsync(profile.Id, profile.Weight))
    {
        if (await _createDaysService.CreateDayForNewUser(profile))
        {
            return Ok("Profile created successfully");
        }
        return BadRequest("Error creating day");
    }
    return BadRequest("Error creating weight evolution");
}
return BadRequest("Profile already exists");
```

- A fost raportat bug-ul misleading-weight-graph-evolution, graficul era eronat (pe axa Ox erau inregistrate doar zilele in care greutate se modificare), a fost reparat.



- A fost raportat bug-ul error-deleting-user, un admin nu putea sterge un utilizator care a creat o mancare noua. A fost reparat stergand inainte mancarurile create de acel utilizator plus schimbarea id-ului utilizatorului in null atunci cand mancarea este facuta publica (in cazul in care alti utilizatori ar fi adaugat acea mancare la zilele lor, stergerea utilizatorului creator ar fi sters si acele mancaruri stricand astfel zilele acelora).



In functia care sterge utilizatori am adaugat:

```
// Get the user's foods and delete them before deleting the user
var foods = await _applicationContext.Foods.Where(f => f.ApplicationUserId == user.Id).ToListAsync();
if (foods != null && foods.Any())
{
    _applicationContext.Foods.RemoveRange(foods);
}
```

6. Comentarii in cod

Avem comentarii in cod care descriu ce fac anumite functii si bucati de cod.

```
// While the request is being processed
// If the request fails then the profile doesn't exist
// or if the user clicked on the edit profile button =>
// user is seeing the profile form
if (!profileInfo || editProfile) {
    return <div>{renderForm(profileInfo)}</div>;
}
```

```
// Creates a day for the current date
4 references
public async Task<bool> PostDayAsync(Entities.Profile profile)
{
    // Calculate the amount of calories that should be consumed in a day
    // based on goal and weight (ignoring physical activity calories)
    Goal goal = profile.Goal;
    double we = profile.Weight;

    int calories = (int)(we * 38);

    if (goal == Goal.Lose)
    {
        calories = (int)(0.8 * calories);
    } else if (goal == Goal.Gain) {
        calories = (int)(1.2 * calories);
    }

    Day day = new(profile.Id, DateOnly.FromDateTime(DateTime.Today), calories);

    try
    {
        _context.Days.Add(day);
```