Proiect Metode Dezvoltare Software

PetSitting App

Enache Ioana Diana Beatrice

Glazov Maria Liana

Suciu Delia Bianca

Scăunașu Anne Marie

Cuprins

[Motivație 3](#_Toc513809245)

[Introducere 5](#_Toc694552183)

[User stories 6](#_Toc147063915)

[Design și baza de date 7](#_Toc1455078164)

[Source control cu git 11](#_Toc411322471)

[Aplicația noastră 12](#_Toc1115001575)

[Codul sursă 13](#_Toc856865320)

[Asistența gpt 17](#_Toc1601008623)

[Produsul final 21](#_Toc370105605)

# Motivație

Motivația unei aplicații web pentru pet sitteri poate fi explicată prin mai multe beneficii și nevoi pe care le satisface atât pentru proprietarii de animale de companie, cât și pentru pet sitterii și pet walkerii:

1. Conveniență și accesibilitate:

Pentru proprietarii de animale de companie: Aplicația le oferă posibilitatea de a găsi rapid și ușor un pet sitter sau un pet walker în zona lor. În loc să caute recomandări sau să posteze anunțuri, pot accesa o platformă centralizată unde găsesc profesioniști verificați.

Pentru pet sitteri și pet walkeri: Le oferă o platformă unde își pot promova serviciile și pot găsi clienți noi fără a investi în marketing individual.

2. Flexibilitate și opțiuni multiple:

Proprietarii de animale de companie pot alege dintr-o varietate de pet sitteri și pet walkeri, în funcție de preferințe, disponibilitate și buget. Pot compara prețuri și servicii pentru a găsi cea mai bună opțiune.

3. Economii de timp:

Căutarea unui pet sitter sau pet walker poate fi o sarcină consumatoare de timp. O aplicație centralizată simplifică acest proces, economisind timp pentru utilizatori.

4. Ușurință în utilizare:

O interfață intuitivă și funcționalități ușor de utilizat fac ca întregul proces de căutare și angajare a unui pet sitter sau pet walker să fie simplu și eficient.

5. Construirea unei comunități:

Aplicația poate ajuta la crearea unei comunități de iubitori de animale care să împărtășească sfaturi, experiențe și recomandări, contribuind astfel la îmbunătățirea serviciilor și la crearea unui sentiment de apartenență.

# Introducere

Proiectul se găsește la <https://github.com/AnneScaunasu/MDS_project>

Acesta este o platformă web pentru servicii de petsitting, dezvoltată folosind framework-urile Spring și Angular. Scopul principal al platformei este de a conecta proprietarii de animale de companie cu petsitteri de încredere care oferă diverse servicii, cum ar fi plimbarea câinilor, îngrijirea animalelor la domiciliu sau la petsitter, și multe altele.

Publicul țintă pentru această aplicație este format din proprietari de animale de companie care caută servicii de petsitting și petsitteri care doresc să ofere servicii.

# User stories

As a Pet Owner, I want to create an account so that I can access the services provided by the platform.

As a Pet Owner, I want to search for petsitters by location and service type so that I can find suitable sitters near me.

As a Pet Owner, I want to book a petsitter so that I can ensure my pet is taken care of when I am unavailable.

As a Petsitter, I want to create an account so that I can offer my services on the platform.

As a Petsitter, I want to see and respond to reviews so that I can engage with my clients and improve my service.

As an Administrator, I want to monitor platform activity so that I can generate reports and insights.

# Design și baza de date

1. Logo

Design minimalist și ușor de reținut, reprezentativ pentru ceea ce oferă și transmite un sentiment de grijă și bunăstare.



2. Nume: Paws&Playdates

Numele este clar și are relevanța, indicând clar cu ce se ocupa aplicația, utilizarea o aliterației (două cuvinte care încep cu litera ‘p’), oferă un caracter catchy si memorabil, evocă imagini pozitive legate de animale și joacă, aspect modern, simplu și original, versatil (poate incorpora multiple servicii de tip pet sitting, pet walking, etc.)

3. Paleta de culori

Paleta de culori aleasă pentru acest proiect conține o varietate echilibrată de culori, asigurând astfel un echilibru vizual plăcut. Turcoazul și coralul sunt culori luminoase care duc cu gândul la joacă și energie, în timp ce nuanțele calde de crem conferă un sentiment de grijă și atenție. Culoarea mov adaugă un accent de profesionalism și seriozitate. Paleta oferă o mare flexibilitate în designul UI, permițând combinarea diferitelor culori și asigurând suficiente culori deschise, mai închise, de accent și de fundal.



3. Baza de date

Baza de date a fost implemantată folosind MySQL.

Avem următoarea diagramă:



Tabelele folosite în dezvoltarea aplicației sunt:

1. Users (Utilizatori):

◦ Atribute: user\_id, username, password, email, phone\_number, location, registration\_date

◦ Descriere: Reprezintă utilizatorii aplicației, fie că sunt proprietari de animale de companie, fie pet sitteri sau pet walkeri.

2. Pets (Animale de companie):

◦ Atribute: pet\_id, user\_id, pet\_name, species, breed, age, special\_needs, vaccination\_status, notes

◦ Descriere: Conține informații despre animalele de companie ale utilizatorilor.

◦ Relație: Fiecare animal aparține unui utilizator (user\_id).

3. PetService (Servicii pentru animale):

◦ Atribute: service\_id, service\_type, description, rate\_per\_hour, availability, location

◦ Descriere: Definiția serviciilor oferite, cum ar fi pet sitting sau pet walking.

◦ Relație: Poate fi oferit de utilizatori.

4. Bookings (Rezervări):

◦ Atribute: booking\_id, user\_id, service\_id, pet\_id, booking\_date, start\_time, end\_time, total\_cost, status

◦ Descriere: Reprezintă rezervările făcute de utilizatori pentru servicii specifice.

◦ Relații:

▪ Fiecare rezervare este asociată cu un utilizator (user\_id).

▪ Fiecare rezervare este legată de un serviciu (service\_id).

▪ Fiecare rezervare implică un animal de companie (pet\_id).

5. Payments (Plăți):

◦ Atribute: payment\_id, booking\_id, amount, payment\_date, payment\_method, payment\_status

◦ Descriere: Detalii despre plățile efectuate pentru rezervări.

◦ Relație: Fiecare plată este asociată cu o rezervare (booking\_id).

6. Reviews (Recenzii):

◦ Atribute: review\_id, booking\_id, rating, review\_text, review\_date

◦ Descriere: Recenziile utilizatorilor asupra serviciilor primite.

◦ Relație: Fiecare recenzie este legată de o rezervare (booking\_id).

Relații între entități:

• Users și Pets: Un utilizator poate avea mai multe animale de companie

• Users și Bookings: Un utilizator poate face mai multe rezervări

• Pets și Bookings: Un animal de companie poate fi implicat în mai multe rezervări

• PetService și Bookings: Un serviciu poate avea mai multe rezervări

• Bookings și Payments: Fiecare rezervare are o plată asociată

• Bookings și Reviews: Fiecare rezervare poate avea o recenzie asociată

Înainte de a implementa aplicația propriu zisă, am creat un mock up folosind figma, pentru a ne forma o idee asupra cum vrem să arate <https://www.figma.com/design/1sOXxDfHtYRrqGNnTYXgCG/Pet-Sitters?node-id=73-4>

# Source control cu git

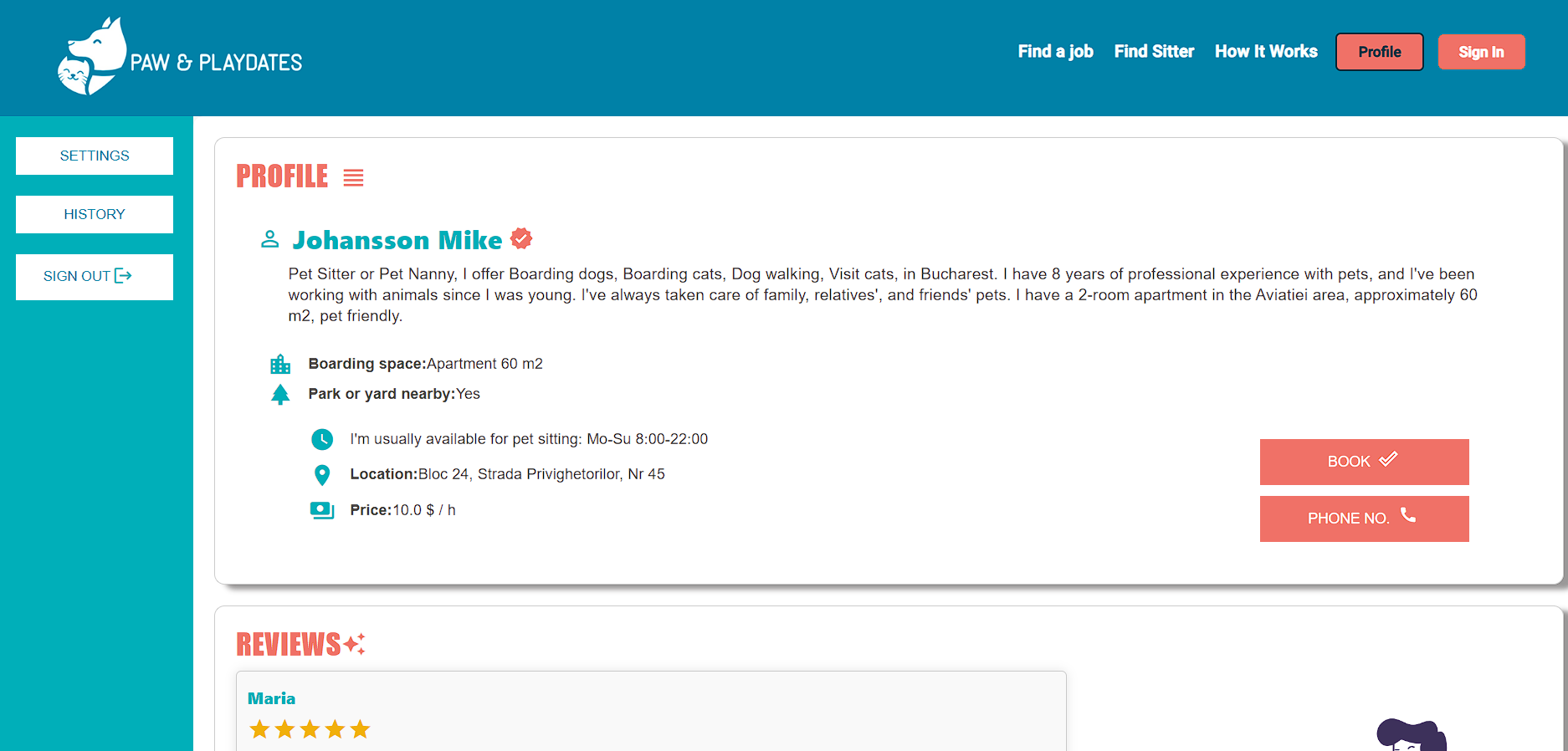
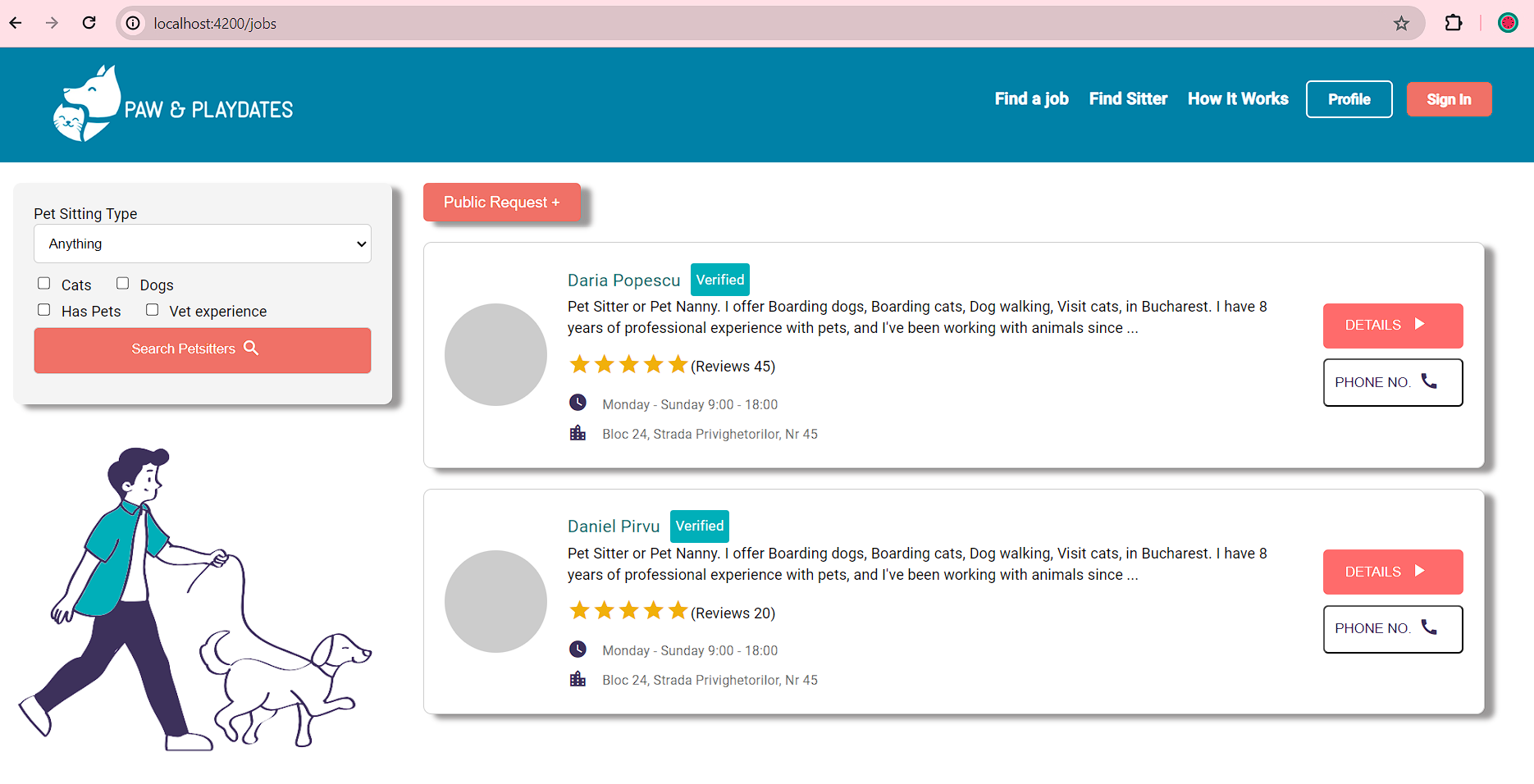
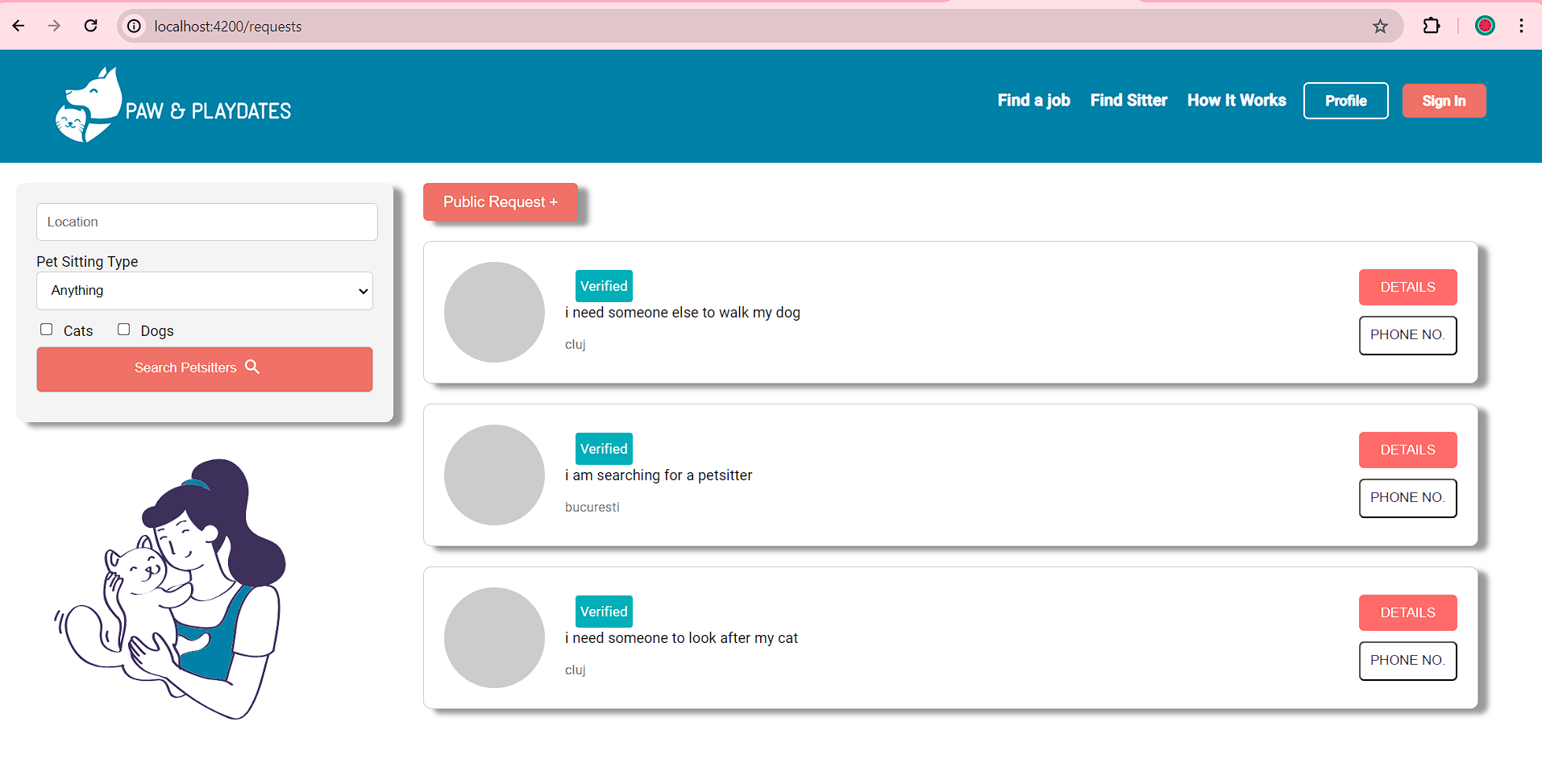
Pentru o muncă în echipă mai eficientă am folosit github.

Branch-ul main este privat, ca urmare de fiecare dată când au fost implementate lucruri noi a fost nevoie crearea unui alt branch, efectuarea de push si commit în branch-ul respectiv, și apoi efectuarea unui pull request pentru a aduce modificările în main. Pull requestul trebuie să fie aprobat de un coechipier înainte de a putea face merge.

# Aplicația noastră

Avem 4 ferestre între care utilizatorul poate naviga: home, jobs, requests și profile.

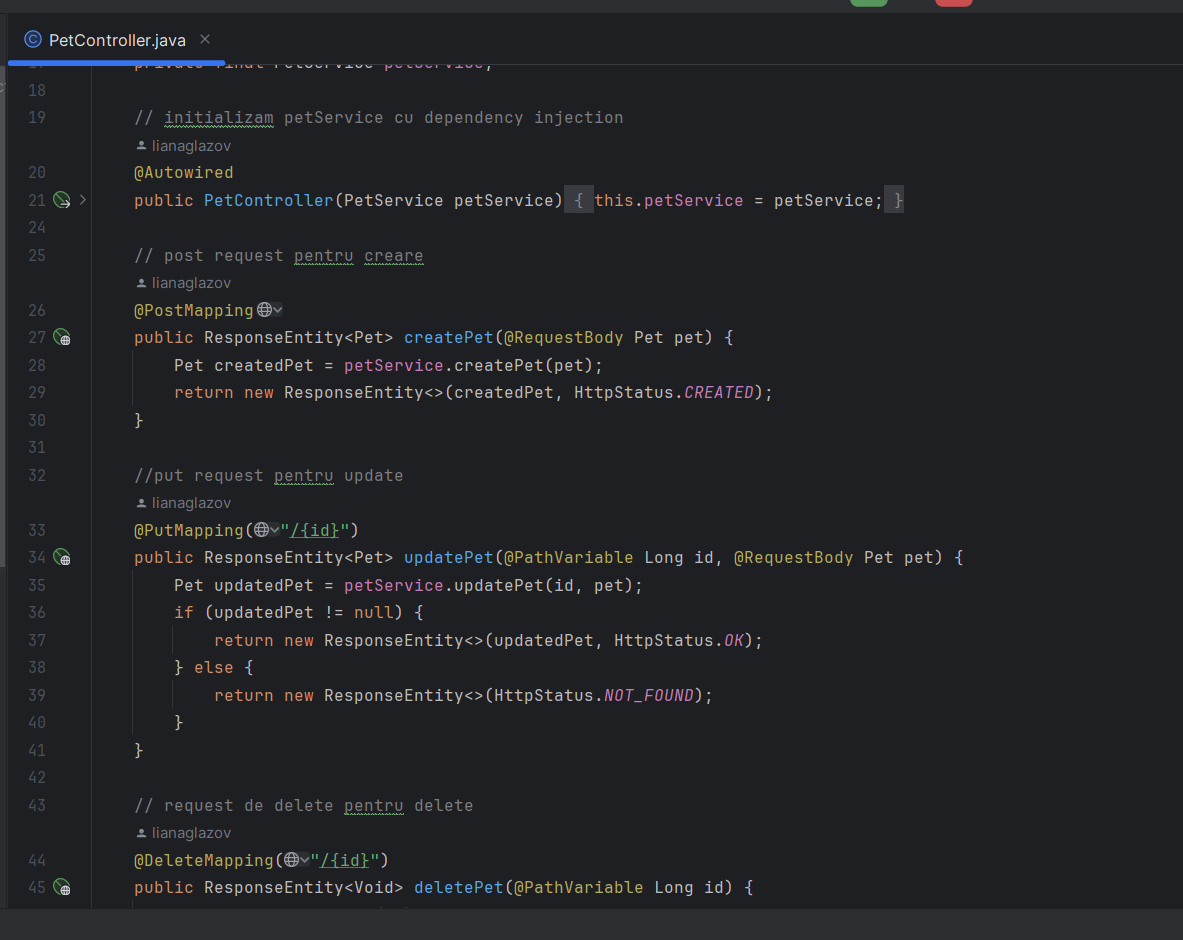
Fereastra “Home” este pagina principală a aplicației, unde utilizatorii pot găsi informații generale și relevante despre serviciile și caracteristicile aplicației, “Jobs” este dedicată listării și detaliilor despre toți petsitterii înregistrați în aplicație, “Requests” este destinată vizualizării și gestionării cererilor de pet sitting, iar “Profile” este secțiunea dedicată fiecărui utilizator înregistrat, unde aceștia își pot accesa și gestiona informațiile personale.

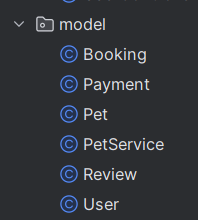
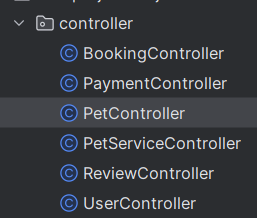


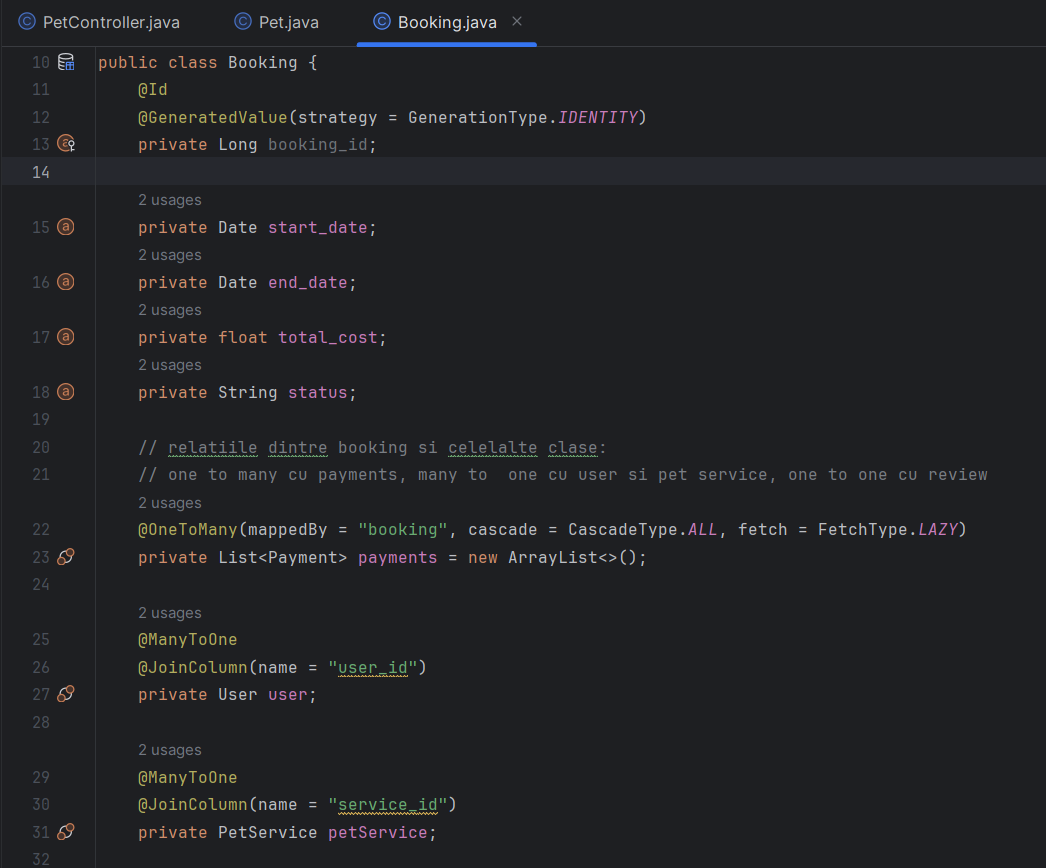
# Codul sursă

Partea de backend a fost scrisă în limbajul java, am creat clase pentru fiecare entitate din baza de date, și am efectuat operații de crud pe acestea.

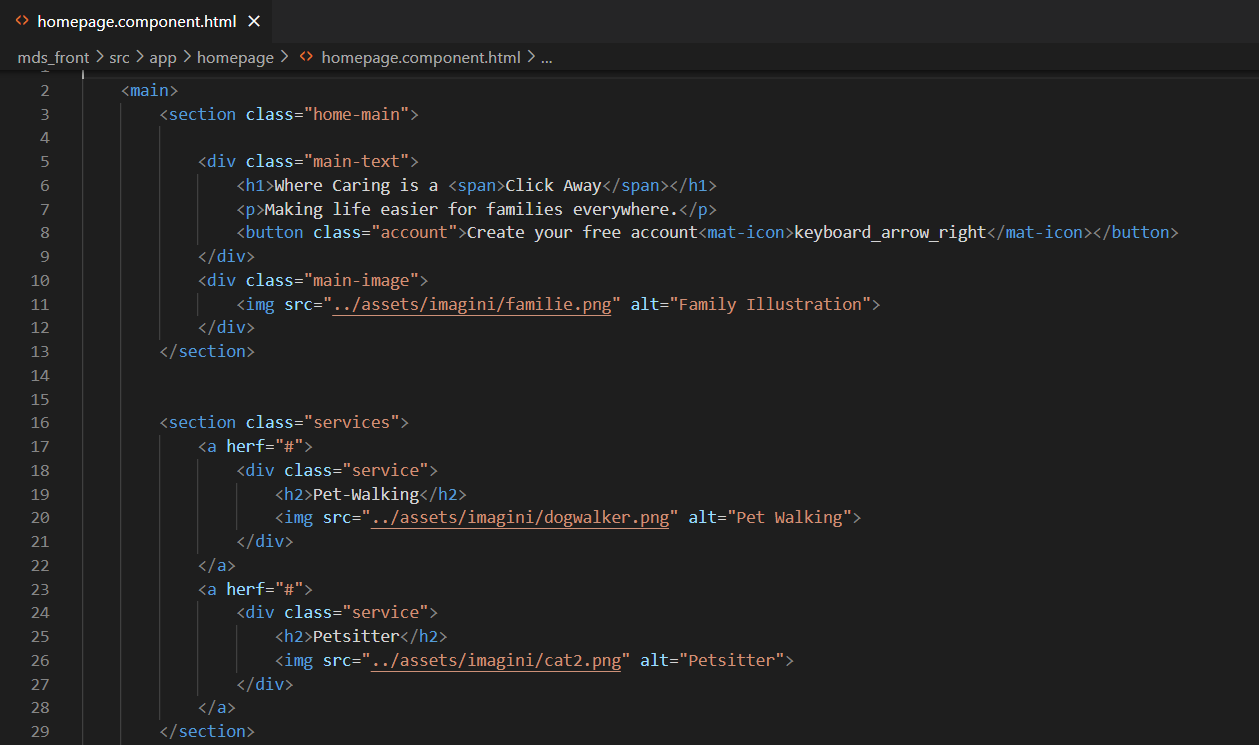
Framework-ul spring ne-a oferit un set de class patterns pentru ușurință în dezvoltare și uniformitate în lucru.



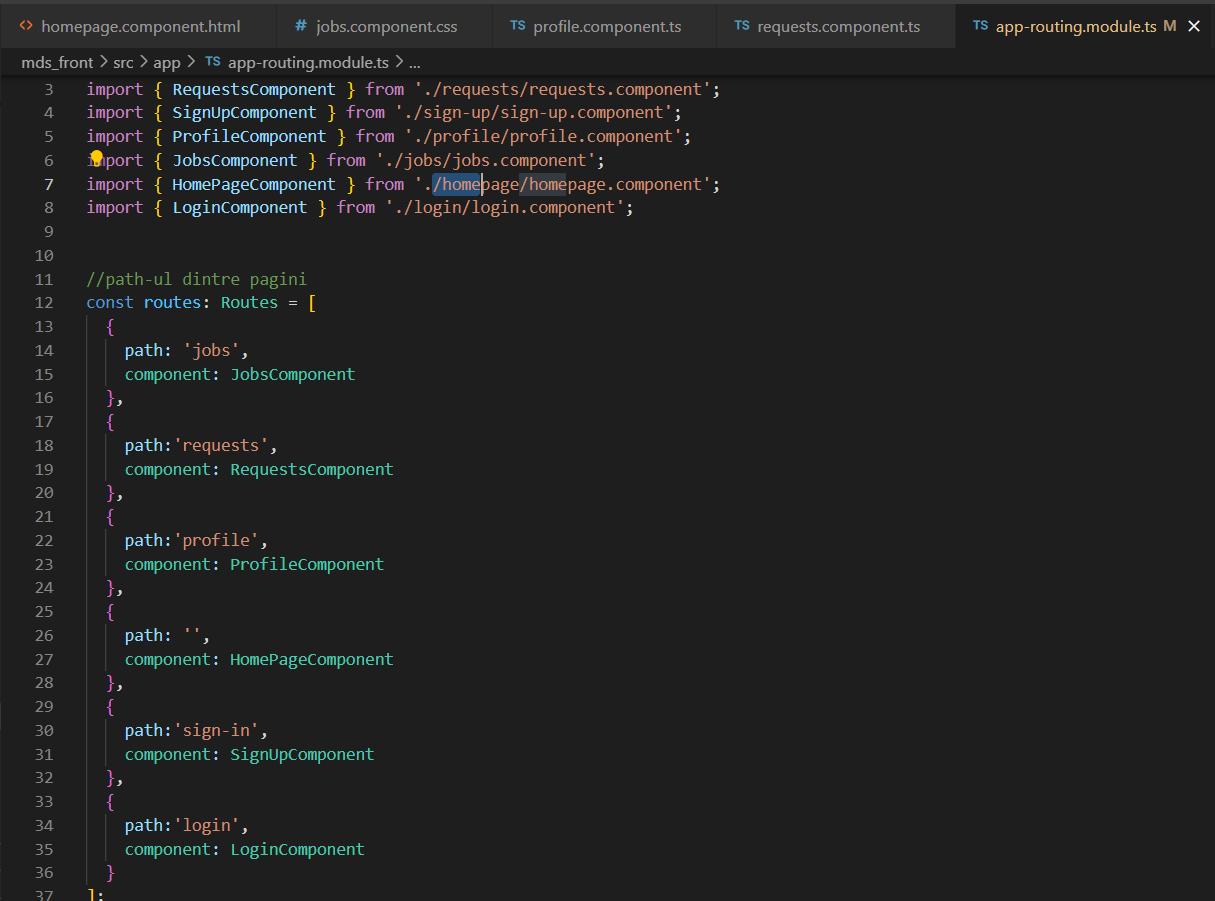
 



Pe partea de front, aplicația a fost dezvoltată folosind html, css și typescript.

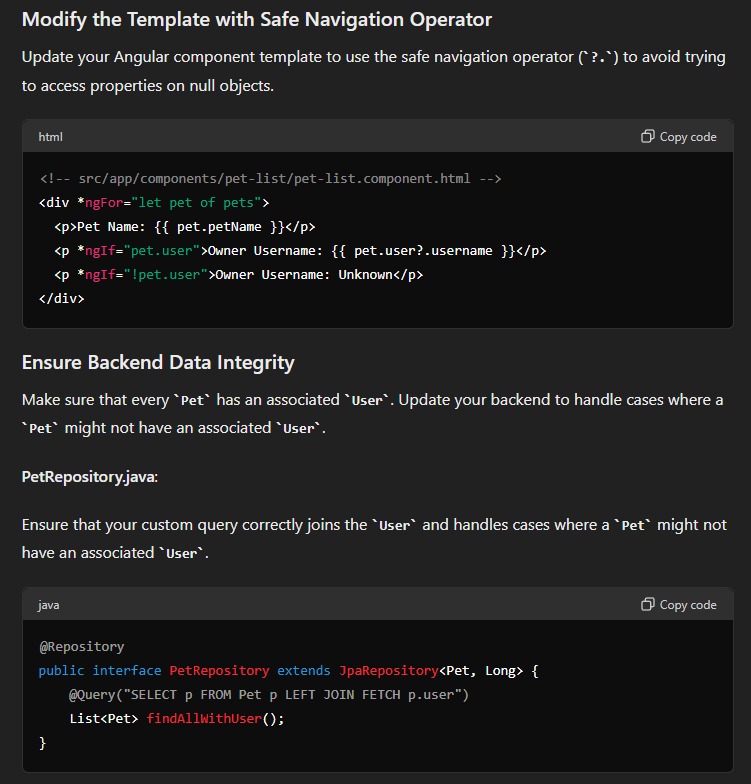


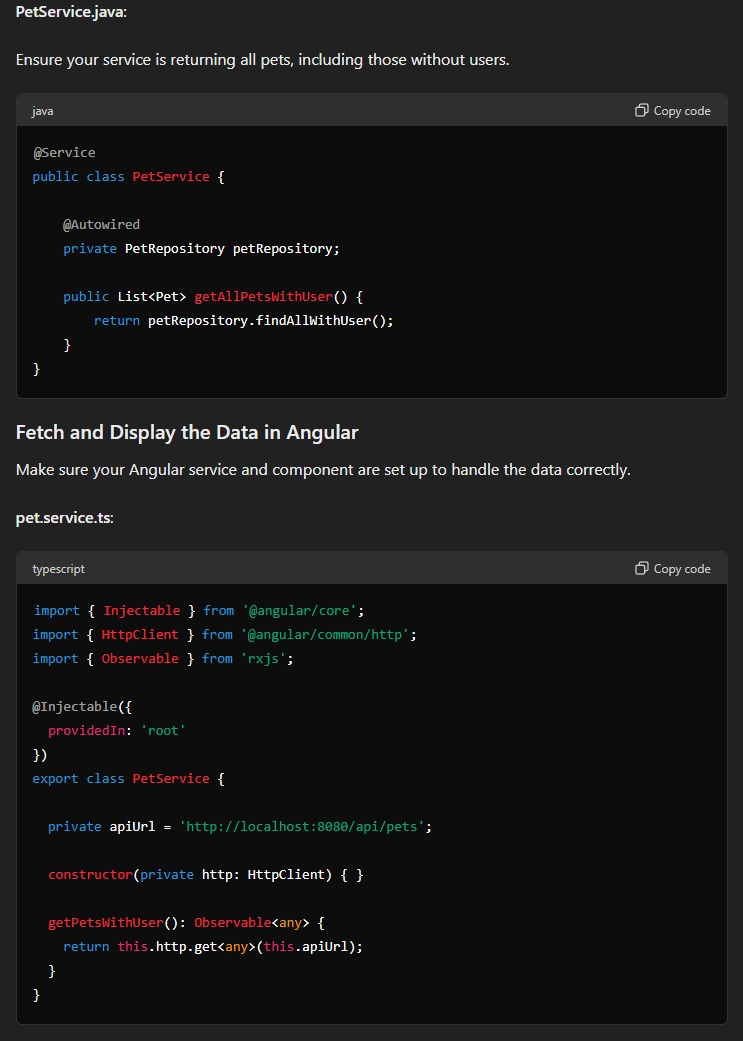




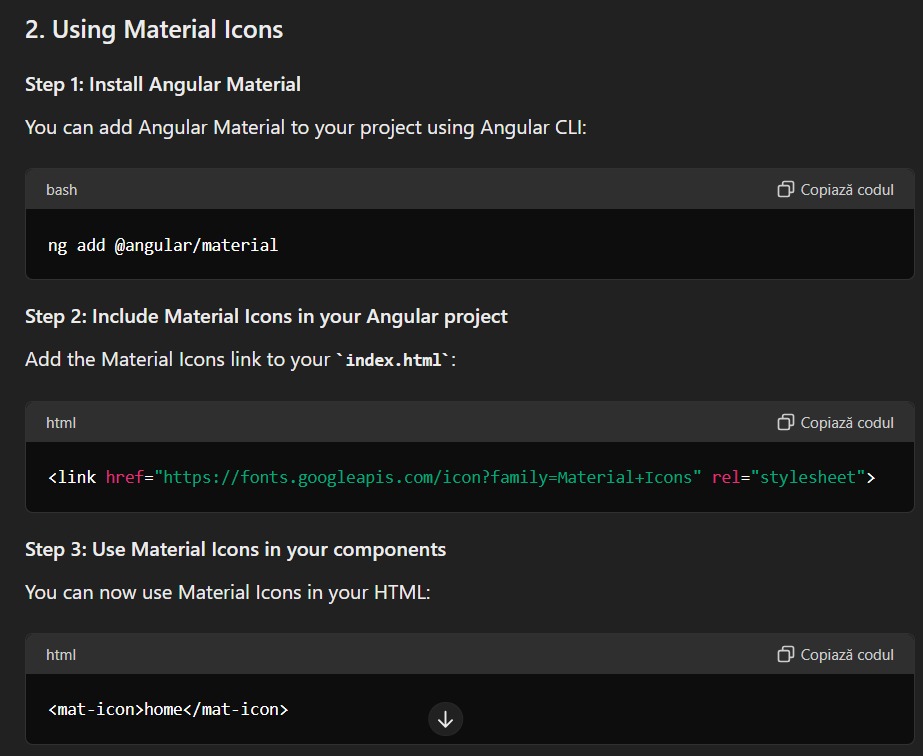
# Asistența gpt

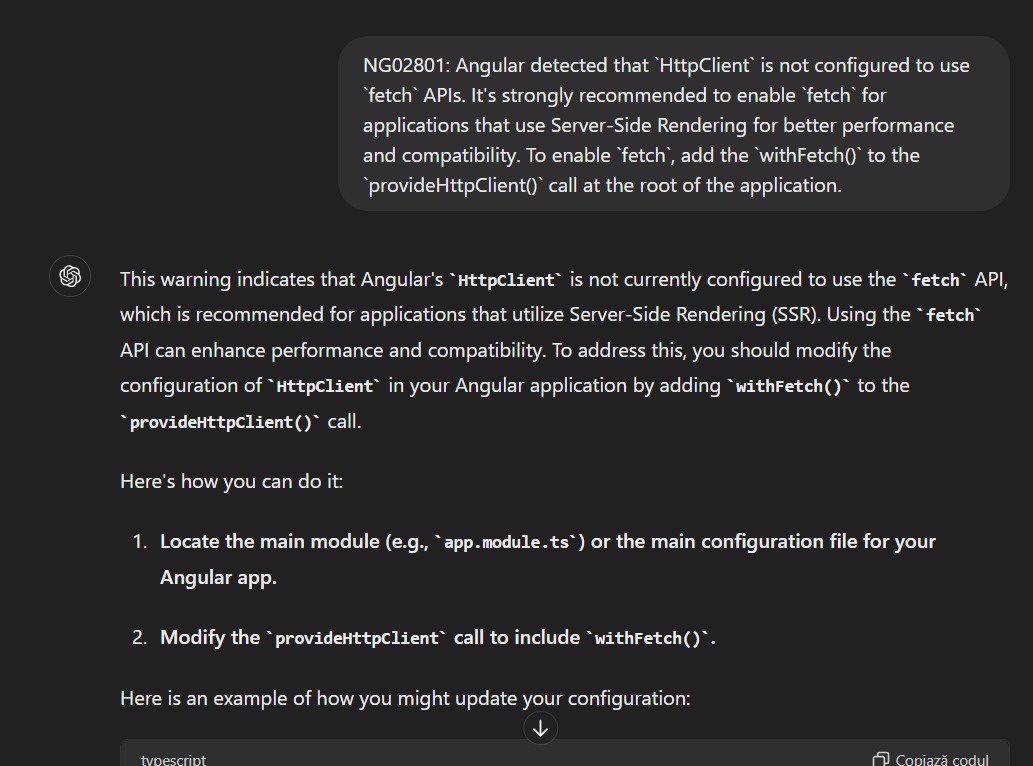
Pe parcursul dezvoltării proiectului, din dorința de a ne eficientiza munca, sau din cauza unor probleme întâlnite în cod, care păreau că nu au o soluție, am cerut ajutor uneltelor de inteligență artificială.





Am folosit ChatGPT pentru a mă ghida in folosirea unui element de Angular care mi-a permis sa creez o bucla prin care am creat un întreg div, pentru fiecare record din tabela "requests" a bazei de date. Se extrag record-urile din tabela "requests", si trecând prin backend, sunt direcționate către front-end, iar in fișierul de typescript sunt preluate si inserate in bucla care creează noi div-uri in pagina de requests.





Erorile sunt o mare bătaie de cap care apare în viața de zi cu zi a unei persoane care lucrează în dezvoltare software, iar programele de AI pot să găsească eficient soluții pentru a le remedia, ușurând procesul de debugging.

# Produsul final

Deși rezultatul final nu a corespuns complet așteptărilor noastre inițiale și nu am reușit să implementăm toate funcționalitățile dorite, munca depusă în acest proiect a fost intensă și implicată. Am întâmpinat provocări și obstacole pe parcursul dezvoltării, dar am depus eforturi considerabile pentru a realiza cât mai mult posibil în resursele și timpul disponibile.

Am învățat multe din acest proiect, inclusiv cum să navigăm prin provocări și să gestionăm resursele în mod eficient. Experiența acumulată ne va ghida în proiectele viitoare și ne va ajuta să creștem și să ne îmbunătățim abilitățile. În final, eforturile noastre au fost o lecție despre perseverență, colaborare și adaptabilitate în fața schimbărilor și a provocărilor în dezvoltarea de software.