

UNIVERSITATEA POLITEHNICA BUCUREŞTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE



PROIECT DE DIPLOMĂ

FitCheck
Platformă web pentru un stil de viață sănătos

Gigi-Alexandru Dobre

Coordonator științific:
As. Drd. Ing. Daniel Călin Popeangă

Prof. Dr. Ing. Alexandru Boicea

BUCUREŞTI

2023

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT



DIPLOMA PROJECT

FitCheck
Web Platform for a healthy lifestyle

Gigi-Alexandru Dobre

Thesis advisor:
As. Drd. Ing. Daniel Călin Popeangă

Prof. Dr. Ing. Alexandru Boicea

BUCHAREST

2023

CUPRINS

Sinopsis	3
Abstract.....	3
Mulțumiri	4
1 Introducere	5
1.1 Context	5
1.2 Problema	5
1.3 Obiective	6
1.4 Structura lucrării.....	6
2 Analiza și specificarea cerințelor.....	8
3 Studiu de piață / Abordări existente.....	13
3.1 Analiza aplicațiilor de fitness existente.....	13
3.1.1 Fitness Blender.....	13
3.1.2 DAREBEE.....	15
3.1.3 Academy of Nutrition and Dietetics: eatright.org	16
3.2 Analiza tehnologiilor web.....	17
3.2.1 Backend – Node.js împreună cu ExpressJS	17
3.2.2 Backend – Java Spring Boot	17
3.2.3 Frontend – Angular	18
3.2.4 Frontend – React.....	18
3.2.5 Frontend – Bootstrap.....	18
4 Soluția propusă	19
4.1 Diagrama Use-Case	19
4.2 Tehnologii folosite	19
4.2.1 Backend – Node.js împreună cu ExpressJS	19
4.2.2 Frontend – Templating EJS-mate împreună cu Bootstrap.....	19
4.2.3 Bază de date – MongoDB.....	20
4.3 Diagrama de workflow	21
5 Detalii de implementare	22
5.1 Dependințe și model arhitectural	22
5.2 Funcționalități	23

5.2.1 Logare, înregistrare și profil	23
5.2.2 Pagina de pornire	24
5.2.3 Utilizator basic	25
5.2.4 Utilizator premium.....	28
6 Studiu de caz / Evaluarea rezultatelor.....	29
6.1 Teste ștergere și editare.....	29
6.1.1 Utilizatori.....	29
6.1.2 Exerciții.....	30
6.1.3 Rețete.....	31
6.2 Lighthouse teste de performanță	32
6.3 Feedback	33
7 Concluzii	34
8 Bibliografie	35
9 Anexe	36

SINOPSIS

FitCheck este o platformă web destinată tuturor pasionaților de sport și un stil de viață sănătos. Ideea a pornit de la perioada pandemiei, care ne-a obligat să renunțăm la obiceiurile noastre zilnice și prin prisma căreia sedentarismul ne-a acaparat complet. Astfel, atât persoanele care practicau sport înainte, cât și persoanele care erau sedentare, au simțit nevoie de a schimba situația în care se află. Publicul țintă este constituit atât din oameni cu mai puțină experiență, deoarece aplicația funcționează ca un antrenor personal, dar și din oameni experimentați care doresc să-și îmbunătățească cunoștințele despre exerciții, alimentație și să participe la competiții.

ABSTRACT

FitCheck is a web platform designed for all the sports and healthy lifestyle enthusiasts. The main idea originated from the pandemic period that forced us to give up our daily habits and through which a sedentary lifestyle took over. As a result, both those who were involved in sports before and those who were not felt the need to change their current situation. The target audience consists of individuals with less experience, as the application functions as a personal trainer, and also experienced individuals who want to improve their knowledge about exercises, food, and participate in competitions.

MULTUMIRI

As. Drd. Ing. Daniel Călin Popeangă pentru susținere și îndrumare în elaborarea proiectului de diplomă.

1 INTRODUCERE

1.1 Context

Datorită îmbunătățirii și dezvoltării constante a tehnologiei, avem oportunitatea de a crea și a expune publicului aplicațiile de care acesta are nevoie. Indiferent dacă este vorba despre o persoană introvertită sau exact opusul, platforma FitCheck are caracteristici potrivite oricărui tip de personalitate. Această aplicație oferă o varietate de funcționalități legate de sport și nutriție, printre care se numără:

- Exerciții în funcție de diferite grupe musculare
- Rețete de mâncăruri sănătoase, dar și gustoase
- Un element de diferențiere inovator și distractiv reprezentat de o competiție între utilizatorii aplicației, care implică arderea caloriilor
- Abonament premium pentru câștigătorii competițiilor, care le oferă posibilitatea creării unei rețele de prieteni

Motivul pentru care am ales o aplicație de fitness are la bază atât pasiunea mea pentru sport, cât și nevoia unui stil de viață sănătos, acum mai mult decât niciodată. De-a lungul timpului, sedentarismul și alimentația necorespunzătoare au dus la creșterea alarmantă a problemelor de sănătate precum obezitatea, bolile cardiovasculare și diabetul.

Având în vedere această nevoie și avansul tehnologic de care ne bucurăm, consider că este necesară combaterea problemelor de sănătate printr-o soluție accesibilă și ușor de utilizat, care motivează și ghidează oamenii în atingerea obiectivelor lor de fitness.

1.2 Problema

Problema are la bază stilul de viață sedentar al multor oameni, dar și alimentația nesănătoasă, ambele putând fi cauzate de programul încărcat și de varietatea de opțiuni fast-food, care sunt mai atrăgătoare și mai accesibile decât realizarea unui preparat cu ingrediente proaspete.

Cu toții am simțit impactul pe care pandemia l-a avut asupra noastră, fie prin munca de acasă, fie din cauză că am fost împiedicați să facem sport.

- Persoanele care făceau sporturi precum ciclism, alergat sau în cadrul unei săli de forță au fost obligate să renunțe la aceste obiceiuri și să găsească o alternativă acasă, în cel mai bun caz.
- Persoanele care nu făceau deloc sport, dar care beneficiau de deplasarea zilnică către muncă și înapoi, au fost obligate să renunțe la orice fel de mișcare.

1.3 Obiective

Platforma web “FitCheck” este destinată tuturor persoanelor, începând de la persoane cu experiență în domeniu până la persoane care se apucă pentru prima dată de exerciții fizice și doresc să facă o schimbare în alimentația lor.

Obiectivele principale ale proiectului sunt următoarele:

- Combaterea sedentarismului/obezității în rândul tuturor oamenilor, indiferent de vîrstă, prin diverse exerciții, concentrarea pe o anumită zonă musculară și felul în care se execută.
- Menținerea unei alimentații sănătoase prin rețete deosebite cu informații despre ingrediente, valori nutriționale și felul în care se pregătesc.
- Posibilitatea de a adăuga la favorite atât exerciții, cât și rețete care atrag atenția, având o experiență personalizată.
- Încurajarea utilizatorilor să participe în diferite competiții de tip maraton sau ardere de calorii, în urma cărora se pot câștiga premii.
- Recompensarea câștigătorilor cu abonamente premium și deblocarea funcționalității prin intermediul căreia fiecare utilizator își creează o rețea de persoane pe care le urmărește, având acces la exercițiile și rețelele lor favorite.
- Reducerea neajunsurilor altor aplicații, cum ar fi: dedicarea exclusivă exercițiilor sau rețetelor.

1.4 Structura lucrării

Lucrarea se concentrează pe următoarele 6 capitole, dezvoltând următoarele idei.

Capitolul 2 - Analiza și specificarea cerințelor

Acest capitol se bazează pe feedbackul primit în urma completării unui chestionar pe care l-am distribuit atât colegilor de facultate și din alte facultăți, cât și colegilor de muncă și prezintă o analiză a acestuia și a părerilor lor asupra funcționalităților dezvoltate în cadrul aplicației.

Capitolul 3 - Studiu de piață / Abordări existente

În cadrul acestui capitol, se pune accent pe soluțiile deja existente de piață, dar și pe tehnologiiile utilizate în crearea aplicațiilor web. Este realizată o analiză a soluțiilor și sunt evidențiate avantajele și dezavantajele în comparație cu aplicația FitCheck.

Capitolul 4 - Soluția propusă

Acest capitol va dezvolta cerințele necesare de sistem, tehnologiile folosite și arhitectura aplicației. Vor fi prezentate și diagramele use-case, workflow și entitate-relație.

Capitolul 5 - Detalii de implementare

Spre deosebire de capitolul anterior, o descriere amănunțită a tuturor funcționalităților aplicației este realizată.

Capitolul 6 - Studiu de caz / Evaluarea rezultatelor

Acet capitol analizează rezultatele diferitelor teste rulate pentru funcționalități, o serie de feedbackuri din partea celor care au observat aplicația și felul în care se comportă și o analiză Lighthouse, parte a setului de utilitare pentru dezvoltare oferite de Google.

Capitolul 7 - Concluzii

Ca parte a acestui capitol, se prezintă concluziile care reies în urma dezvoltării și analizei aplicației și viitoare planuri de îmbunătățire, menținere și extindere către o aplicație de mobil și smartwatch.

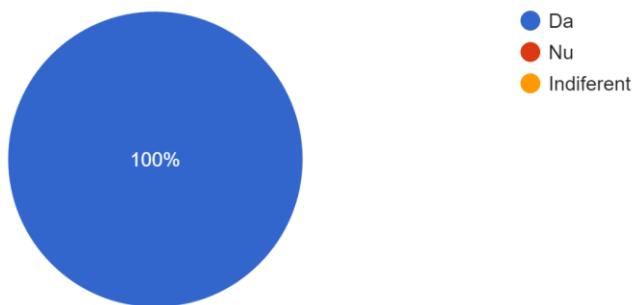
2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Acest capitol se concentrează asupra unei etape esențiale pentru a obține un produs care să fie optim, ușor de utilizat și captivant pentru potențialii clienți. Astfel, din dorința de a avea un stil de viață sănătos, fiind un utilizator al unor astfel de aplicații pentru fitness și din diverse discuții cu alții pasionați de sport, am conceput o serie de funcționalități care să înglobeze dorințele și nevoile tuturor. După rafinarea acestora, am întocmit un chestionar pe care l-am distribuit atât colegilor de facultate și din alte facultăți, cât și colegilor de muncă.

În prima parte a chestionarului, s-a dorit aflarea părerii oamenilor despre impactul sedentarismului și al obezității asupra sănătății, dar și efectul pe care pandemia l-a avut asupra accentuării acestuia.

Credeți că sedentarismul/obezitatea reprezintă o problemă cu un impact considerabil asupra sănătății?

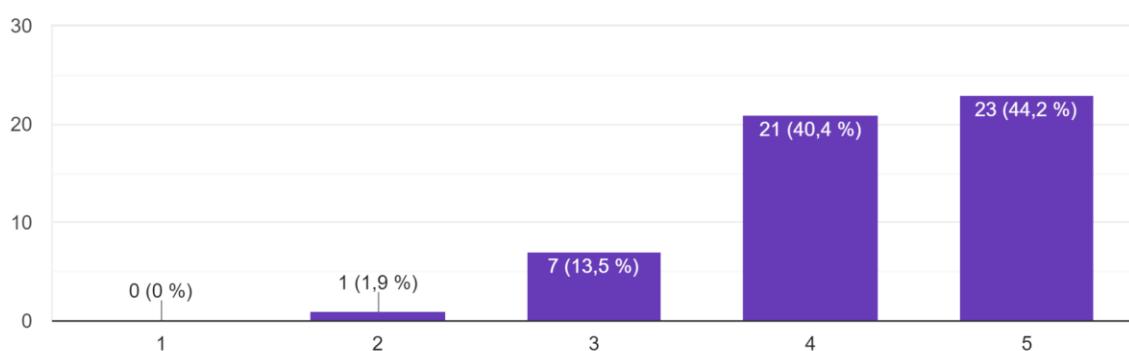
52 de răspunsuri



Figură 1 Întrebarea 1 – impactul sedentarismului/obezității asupra sănătății

Ce impact considerați că a avut pandemia asupra accentuării acestei probleme?

52 de răspunsuri

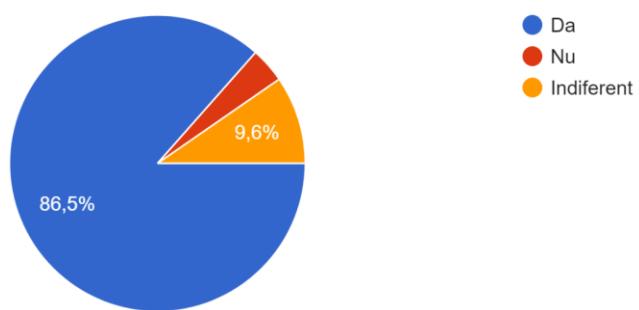


Figură 2 Întrebarea 2 – efectul pandemiei în accentuarea sedentarismului

După cum se poate observa, dintr-un eșantion de 52 de persoane, toate sunt de părere că această problemă are un impact considerabil, iar un procent de 84,6% crede că pandemia a avut un rol major în dezvoltarea ei.

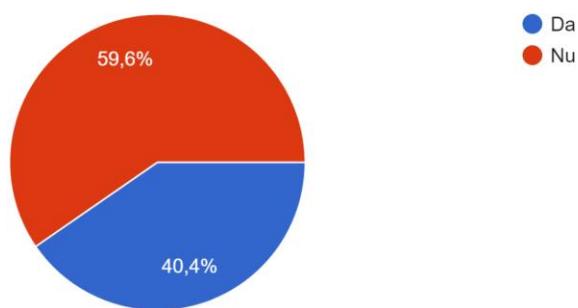
În continuare, obiectivul a fost aflarea opiniei respondenților despre aplicațiile dedicate unui stil de viață sănătos. Astfel, majoritatea consideră că acestea sunt benefice, indiferent dacă au mai folosit sau nu în trecut aplicații pentru fitness.

Credeti că ar fi utilă o aplicație dedicată îmbunătățirii stilului de viață al oamenilor?
52 de răspunsuri



Figură 3 Întrebarea 3 – utilitatea unei aplicații fitness

Ați mai folosit astfel de aplicații în trecut?
52 de răspunsuri

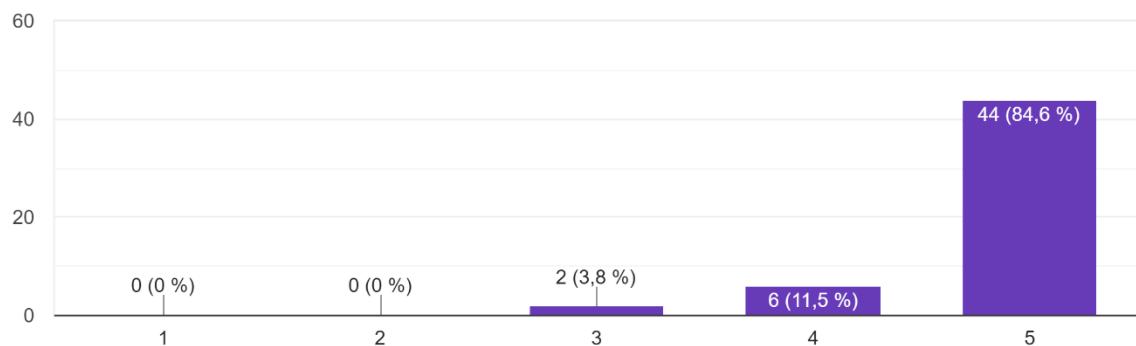


Figură 4 Întrebarea 4 – utilizarea aplicațiilor fitness în trecut

În ce-a de-a treia parte a chestionarului, s-a realizat prezentarea unor funcționalități cu scopul de a observa câte persoane le-ar găsi utile în cadrul aplicației. Astfel, s-a obținut o imagine de ansamblu a preferințelor și a priorităților anumitor funcționalități conform percepției potențialilor clienți.

În ce măsură considerați că aplicația trebuie să conțină atât exemple de exerciții fizice, cât și de rețete sănătoase?

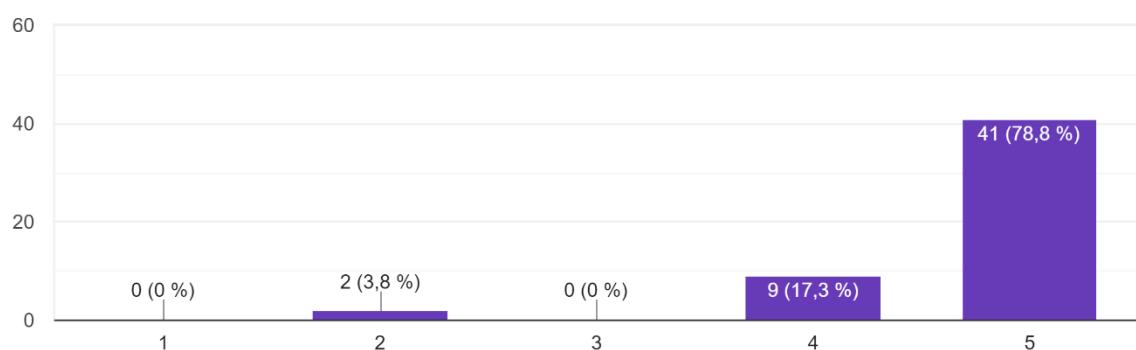
52 de răspunsuri



Figură 5 Întrebarea 5 – exerciții/rețete

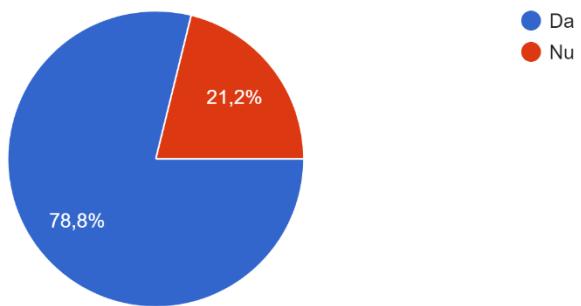
În ce măsură considerați că ar fi un plus să puteți salva la preferate atât exercițiile, cât și rețetele care v-au atras atenția?

52 de răspunsuri



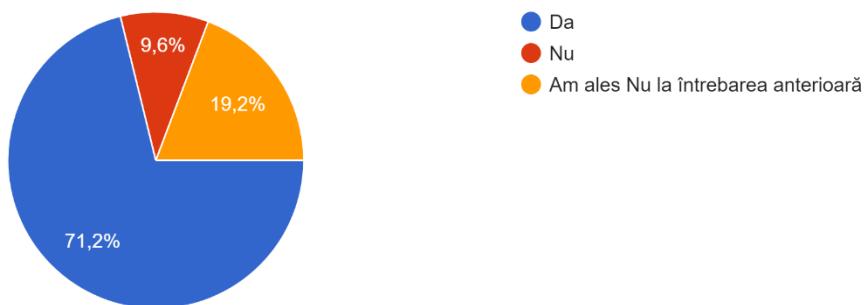
Figură 6 Întrebarea 6 – exerciții/rețete favorite

Credeți că o funcționalitate dedicată competițiilor ar aduce un plus aplicației?
52 de răspunsuri



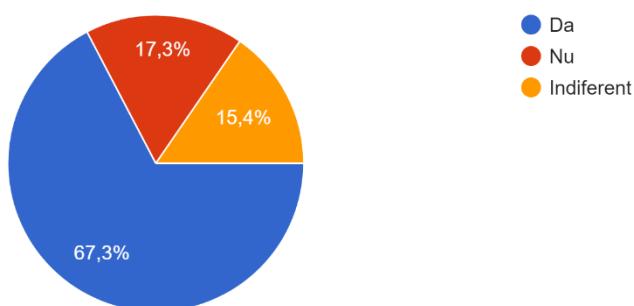
Figură 7 Întrebarea 7 – competiții

Dacă da, ați participa la competiții de tip maraton sau ardere de calorii pentru a câștiga diferite premii precum abonament/bunuri marca FitCheck?
52 de răspunsuri



Figură 8 Întrebarea 8 – tipuri de competiții și dorința de a participa la ele

Credeți că ați folosi o funcționalitate care vă permite să urmăriți activitatea altor utilizatori (de exemplu ce exerciții/rețete au salvat la favorite)?
52 de răspunsuri

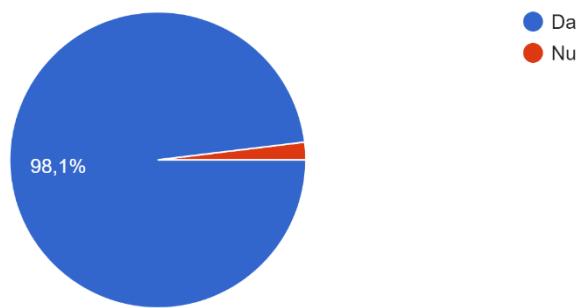


Figură 9 Întrebarea 9 – posibilitatea creării unei rețele de persoane pe care le urmărești

În urma voturilor respondenților, considerând că toate procentajele au fost peste 65%, am ajuns la concluzia că potențialii clienți și-ar dori să se realizeze implementarea tuturor funcționalităților, fără o ordine anume.

Ați recomanda o astfel de aplicație și altor persoane?

52 de răspunsuri



Figură 10 Întrebarea 10 – recomandarea aplicației

Într-un final, ultima întrebare arată procentajul persoanelor care ar recomanda această aplicație și altora și demonstrează un nivel ridicat de interes pentru o astfel de inițiativă.

3 STUDIU DE PIATĂ / ABORDĂRI EXISTENTE

În cadrul acestui capitol, se pune accent pe multiple aplicații de fitness care există pe piață, precum și asupra limitărilor lor, evidențiind oportunitățile de dezvoltare generate de lipsurile acestora și integrate în platforma web dezvoltată. În plus, se vor analiza și tehnologiile folosite în momentul de față pentru dezvoltarea web.

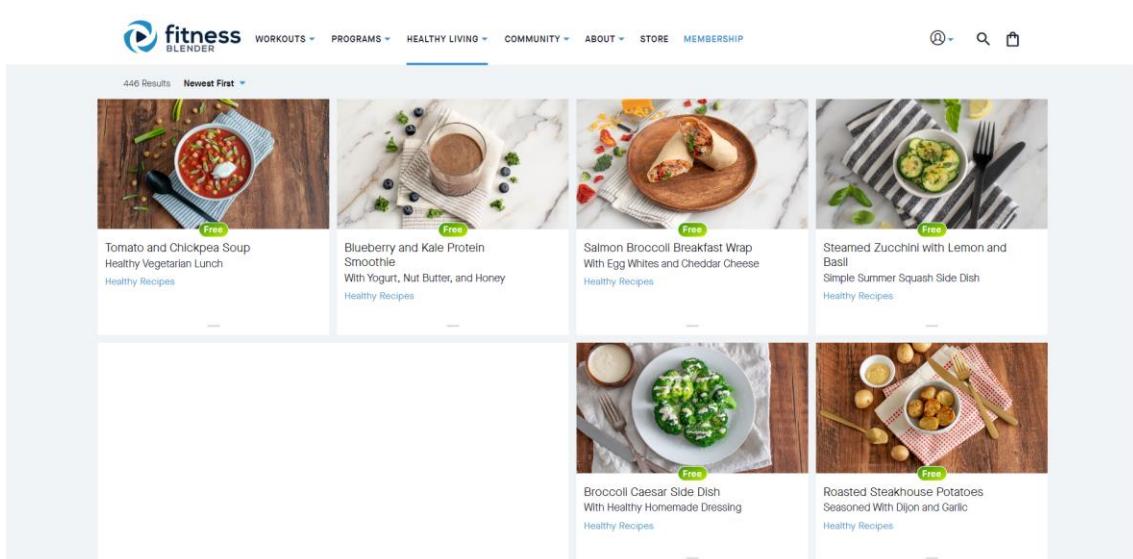
3.1 Analiza aplicațiilor de fitness existente

Un lucru observat în timpul cercetării pieței este că majoritatea aplicațiilor se axează doar pe exerciții sau doar pe rețete. O problemă care apare este aceea că utilizatorii, pentru a-și concepe un plan de antrenament bazat atât pe sport, cât și pe alimentație, au nevoie de mai mult decât de o singură aplicație. Poate fiecare dintre acestea are nevoie la rândul ei de un abonament pentru a putea avea acces la tot ce are mai bun de oferit. În mare parte a timpului, această abordare nu este practică, iar clientii pierd timp prețios încercând să se acomodeze cu diferitele stiluri de afișare și funcționare.

3.1.1 Fitness Blender

Aplicația Fitness Blender a fost lansată în 2010 de către soții Daniel și Kelli Segars, care formează o echipă de doi antrenori personali^[1]. Aceasta a devenit una dintre cele mai cunoscute platforme web pentru fitness din lume și se bazează atât pe antrenamente de diferite dificultăți, precum și pe rețete delicioase. Unul dintre minusurile acestui site sunt limitările aduse de un abonament pentru a putea experimenta site-ul la capacitatea sa maximă.

FitCheck nu împiedică utilizatorii să se bucure de toate exercițiile și rețetele disponibile, având astfel un avantaj în comparație cu Fitness Blender.



Figură 11 Rețete¹

¹ <https://www.fitnessblender.com/healthy-living/healthy-recipes?page=3>

Beginner Workouts

[View All Beginner Workouts](#)

No Jumping Remove-One Style Cardio Circuit
Bodyweight Challenge for Stamina and Mental Grit
40 Min • Total Body

Total Body Power Pilates Circuits
Core-Focused Routine for Long, Lean Muscles - Day 10
44 Min • Core, Total Body

Lower Body Giant Sets
Improve Leg Strength and Endurance - Day 1
39 Min • Lower Body

Power Cardio and Core
Bodyweight Only with Low-Impact Modifications - Day 3
39 Min • Total Body

Figură 12 Antrenamente începători[2]

HIIT Workouts

[View All HIIT Workouts](#)

Athletic-Inspired Cardio and Core Intervals
Bodyweight Drills for Speed, Power, and Agility
40 Min • Core, Lower Body

Upper Body Push Muscles with Kickboxing Finisher
Tabata Intervals to Sculpt the Shoulders, Chest and Triceps - Day 2
36 Min • Upper Body

Power Cardio and Core
Bodyweight Only with Low-Impact Modifications - Day 3
39 Min • Total Body

Cardio, Glutes, and Functional Core
Bored Easily Format with Bodyweight Exercises - Day 8
46 Min • Core, Lower Body, Total Body

Figură 13 Antrenamente de intensitate mare[2]

Strength Workouts

[View All Strength Workouts](#)

Compound Strength and Power Supersets
Bored Easily Full Body Routine with Isometric Burnout - Day 9
50 Min • Total Body

Upper Body Pull Strength with Bodyweight Burnout
Unilateral and Bilateral Exercises for the Back, Biceps, and Rear Deltoids - Day 4
36 Min • Upper Body

Total Body MetCon Circuits
Strength, Power and Cardio for Metabolic Conditioning - Day 6
45 Min • Total Body

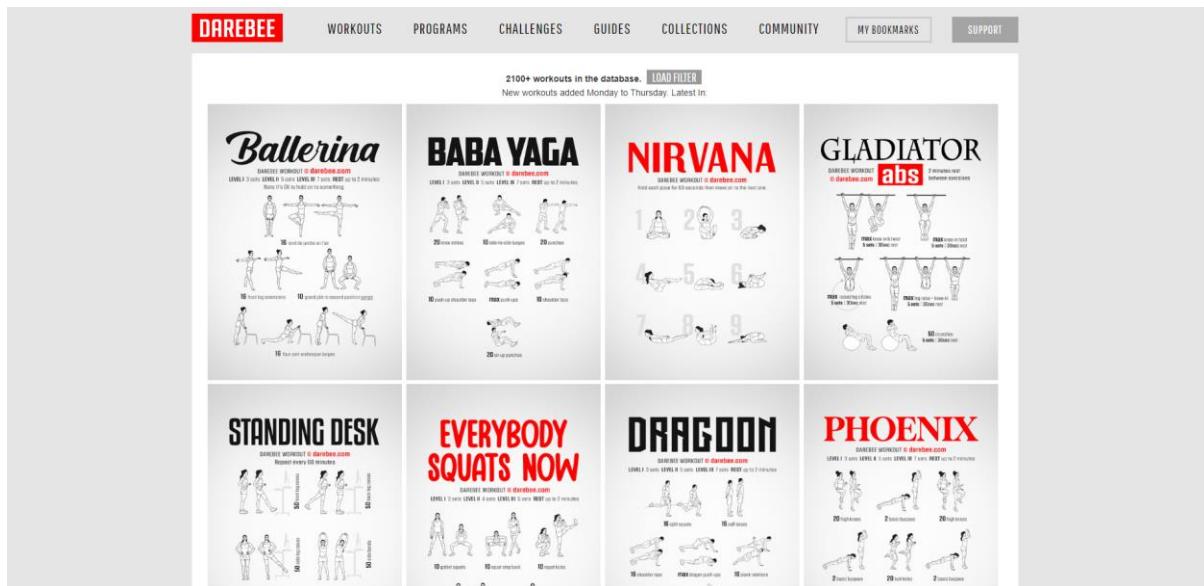
Upper Body Push Muscles with Kickboxing Finisher
Tabata Intervals to Sculpt the Shoulders, Chest and Triceps - Day 2
36 Min • Upper Body

Figură 14 Antrenamente de putere[2]

3.1.2 DAREBEE

Aplicația Darebee este destinată tuturor și este întreținută de un grup de voluntari pasionați de fitness, cu o experiență de peste 20 de ani, care cred că sportul ar trebui să fie accesibil indiferent de situația financiară. Toate antrenamentele sunt testate de către ei în ateliere și sunt concepute pentru a obține rezultate bune cu scopul de a-ți dezvolta atât corpul, cât și mintea[3].

Minusul pe care această aplicație îl are este faptul că pe lângă aceste antrenamente bine gândite, nu se axează și pe rețete și diete și nu are exerciții separate. Plusul este că oferă o multitudine de combinații de exerciții în comparație cu FitCheck, care te lasă să îți alegi singur exercițiile care îți se par interesante. În același timp este și un dezavantaj, câteodată fiind ideal să îți creezi propriul tău antrenament.

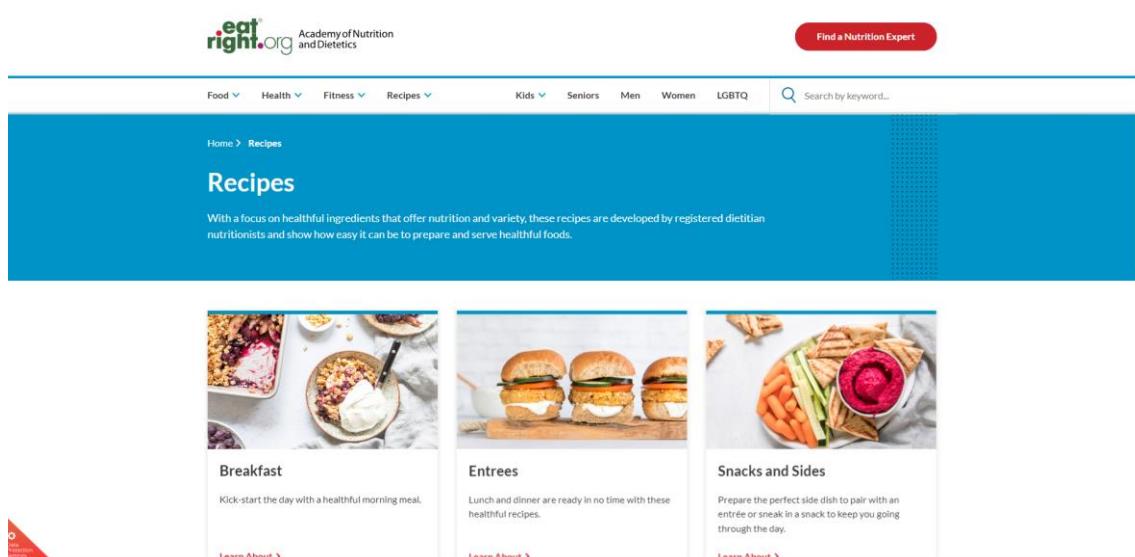


Figură 15 Antrenamente²

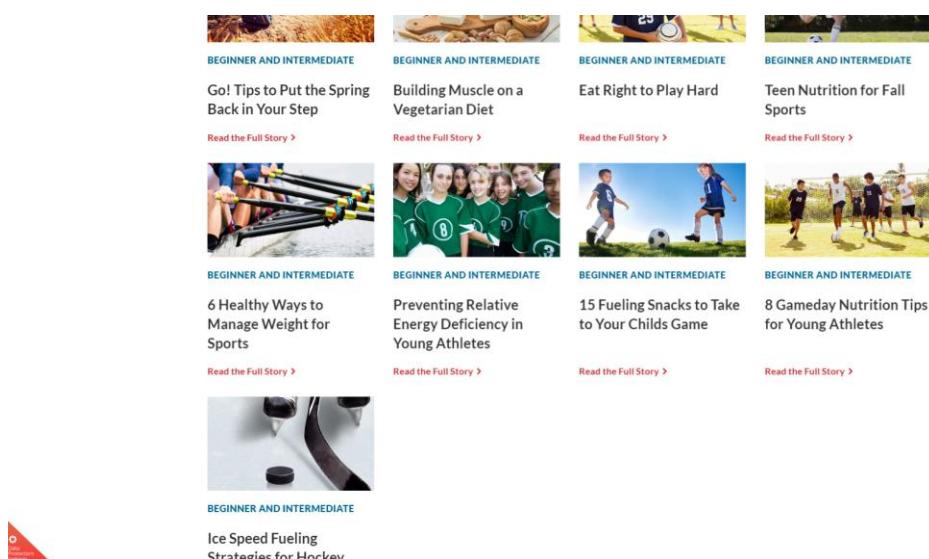
² <https://darebee.com/workouts.html>

3.1.3 Academy of Nutrition and Dietetics: eatright.org

Aplicația eatright.org a fost înființată de către Academy of Nutrition and Dietetics și este constituită din peste 112.000 de profesioniști în domeniul nutriției și al dietei. Misiunea lor este aceea de a promova un stil de viață sănătos prin intermediul mâncării[4]. Totodată, au dispuse pe pagina lor și idei de antrenamente, dar și indicații despre cum să te pregătești înainte de un maraton sau orice alt sport. Dezavantajul acestei platforme web constă în faptul că este greu de parcurs pentru a găsi ceea ce ai nevoie, iar dacă oferă informații despre exerciții, le oferă sub forma unui articol, fără instrucțiuni clare.



Figură 16 Rețete³



Figură 17 Articole despre sporturi⁴

³ <https://www.eatright.org/recipes>

⁴ <https://www.eatright.org/fitness/sports-and-athletic-performance/beginner-and-intermediate>

3.2 Analiza tehnologiilor web

Din punct de vedere al tehnologiilor cu ajutorul cărora se pot dezvolta aplicații web, există o multitudine de opțiuni din care se poate alege. Fiecare este performantă în felul ei, și se împart în două categorii: frontend și backend.

3.2.1 Backend – Node.js împreună cu ExpressJS

Node.js este un mediu de rulare ce oferă dezvoltatorului posibilitatea de a scrie și de a executa cod de JavaScript. Principalul avantaj pe care Node.js îl are este că se poate scrie atât codul pentru server, cât și cel pentru client în același limbaj de programare[5].

ExpressJS este un framework destul de simplu care aduce o suită de funcționalități în plus față de cele deja existente în Node.js[6]. Ceea ce scoate cu adevărat în evidență performanța acestuia este:

- Posibilitatea de defini rutarea folosind metodele HTTP definite în cadrul acestui framework⁵.
- Capabilitatea de interfațare folosind template engine⁶.
- Are un suport excelent pentru middleware, cu ajutorul căruia se pot adăuga noi funcționalități între cereri și răspunsuri, îmbunătățind gestionarea erorilor⁷.
- Fișiere statice pot fi servite prin folosirea unei funcții middleware predefinite, astfel scurtând calea către fișiere CSS, JavaScript și imagini⁸.

3.2.2 Backend – Java Spring Boot

Spring Boot este un framework al limbajului de programare Java și principala lui caracteristică este abilitatea de a crea microservicii, motiv pentru care este o tehnologie foarte puternică și scalabilă[7]. Printre avantaje se numără:

- Reducerea complexității procesului prin care o aplicație web este creată și pusă în funcțiune, punând accent pe un set de configurații deja definite, astfel încât dezvoltatorii să aibă în vedere mai mult logica platformei.
- Capabilitatea de a integra frameworkul Hibernate pentru bazele de date relaționale cu scopul de a lucra cu obiecte din Java în locul interogărilor SQL, dar și de a îmbunătăți persistența datelor⁹.
- Microserviciile prin intermediul cărora se pot crea mai multe componente independente care comunică între ele prin calluri de API.

⁵ <https://expressjs.com/en/guide/routing.html>

⁶ <https://expressjs.com/en/guide/using-template-engines.html>

⁷ <https://expressjs.com/en/guide/using-middleware.html>

⁸ <https://expressjs.com/en/starter/static-files.html>

⁹ <https://hibernate.org/orm/what-is-an-orm/>

3.2.3 Frontend – Angular

Angular este un framework foarte puternic pentru dezvoltarea frontendului unei aplicații web și în cadrul acestuia se folosește TypeScript, o versiune JavaScript care oferă verificări de sintaxă^[8]. Motivele pentru care s-ar folosi Angular sunt următoarele:

- Compilare ahead-of-time care permite o viteză ridicată de încărcare a componentelor în pagină deoarece codul de TypeScript și HTML este convertit în JavaScript în timpul buildului și folosit apoi de către browser¹⁰.
- Este dezvoltat și întreținut de cei de la Google, având un suport foarte bun.
- Sincronizarea dintre model și view este bidirectională, prin urmare orice modificare făcută asupra informațiilor din baza de date este propagată și către client și viceversa.

3.2.4 Frontend – React

React este o librărie de JavaScript și este una dintre cele mai populare tehnologii de frontend^[9] datorită numărului ridicat de repozitorii Github în care este folosit, aproape de 4 ori mai mare decât cel al lui Angular – 3.924.464¹¹ versus 1.076.422¹². Principalele puncte forte ale acesteia sunt:

- Este dezvoltată și întreținută de cei de la Meta, ceea ce înseamnă o stabilitate îndelungată și o comunitate mare de dezvoltatori.
- Fiecare componentă are starea ei proprie, iar informațiile pot să fie propagate atât de la parinți către copii, cât și invers. În principal se folosește doar prima variantă.
- Dispune de aşa numitul DOM virtual, care este mai eficient decât cel tradițional, fiind utilizat când vine vorba de modificări asupra componentelor, ca mai apoi să fie actualizat și cel real.

3.2.5 Frontend – Bootstrap

Bootstrap este un framework care oferă dezvoltatorilor posibilitatea de a crea aplicații web cu un design receptiv^[10]. A fost creat de o echipă de dezvoltatori din cadrul companiei Twitter, iar popularitatea acestuia a crescut datorită ușurinței cu care orice persoană care deține cunoștințe, nu neapărat avansate de HTML și CSS, poate să dezvolte frontendul pentru o platformă. Principalele avantaje ale folosirii acestui framework sunt:

- Datorită designului receptiv, funcționarea la parametri optimi ai aplicației este garantată pe toate dispozitivele.
- Dispune de o mulțime de componente deja implementate și personalizate precum: butoane, carusele de imagini, bara de navigație, modale, comprimarea textului.
- Oferă un grid predefinit din rânduri și coloane, fără ca dezvoltatorul să fie nevoie să construiască unul¹³.

¹⁰ <https://angular.io/guide/aot-compiler>

¹¹ <https://github.com/search?q=react>

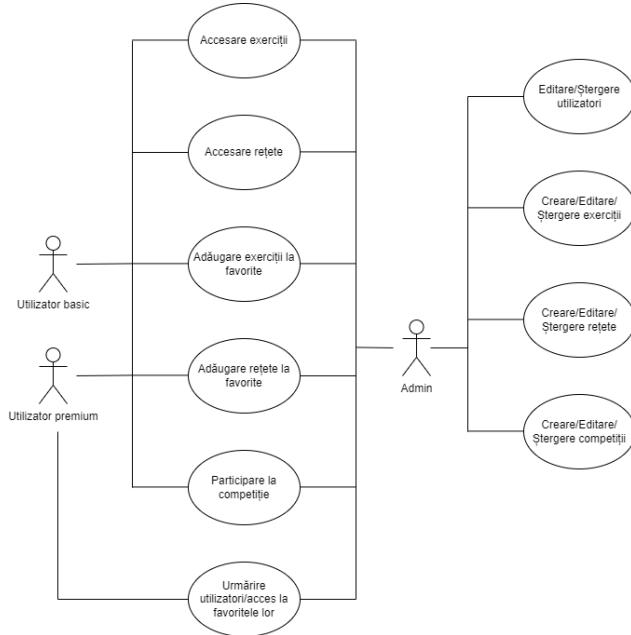
¹² <https://github.com/search?q=angular>

¹³ <https://getbootstrap.com/docs/5.3/layout/grid/>

4 SOLUȚIA PROPUȘĂ

Fiind o aplicație pentru web, nu există restricții pentru sistemul de operare. Totuși, este de evitat accesarea paginii din cadrul browserului Microsoft Edge. În afară de această restricție, nu contează browserul de pe care utilizatorul se conectează. Pentru implementare și învățarea tehnologiilor am folosit cursul “The Web Developer Bootcamp 2023”¹⁴.

4.1 Diagrama Use-Case



Figură 18 Diagrama Use-Case

Pentru a reprezenta acțiunile pe care fiecare dintre cele 3 tipuri de utilizatori le pot efectua, a fost aleasă o reprezentare printr-o diagramă use-case.

4.2 Tehnologii folosite

4.2.1 Backend – Node.js împreună cu ExpressJS

Motivul principal pentru care am ales să folosesc ExpressJS pentru a realiza partea de backend a aplicației este dezvoltarea acestuia, cât și a frontendului, în limbajul de programare JavaScript. În același timp, am beneficiat și de bibliotecile pe care acest framework le are de oferit și prin intermediul cărora am reușit să construiesc partea de controller a platformei.

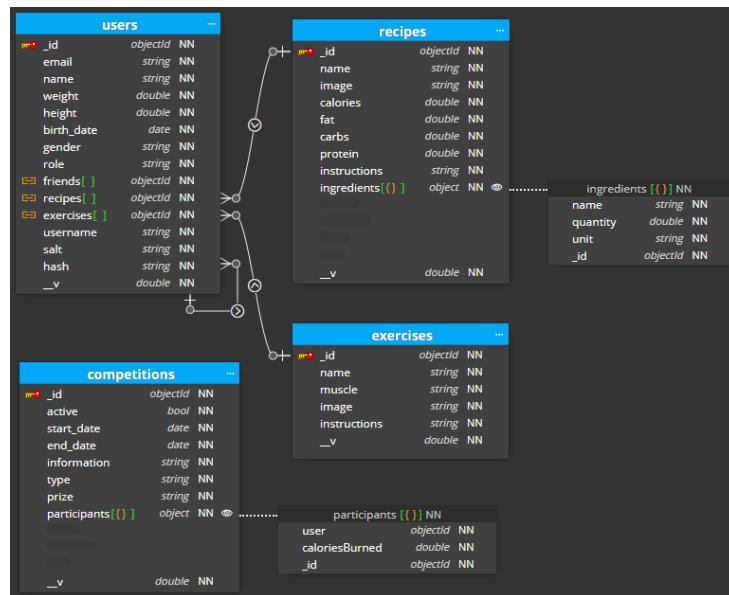
4.2.2 Frontend – Templating EJS-mate împreună cu Bootstrap

Pentru frontend, am ales un framework destul de popular și anume Bootstrap. Aceasta se bazează pe folosirea claselor predefinite pentru a crea elemente care au un design implementat deja de către cei de la Bootstrap. Pentru a introduce cod de JavaScript în html și a lua decizii pentru când sunt afișate sau câte elemente sunt afișate, a fost folosit ejs-mate, un modul de node pentru Express[11], astfel obținând partea de view a aplicației.

¹⁴ <https://www.udemy.com/course/the-web-developer-bootcamp/>

4.2.3 Bază de date – MongoDB

Baza de date folosită pentru implementarea aplicației este MongoDB[12]. Aceasta este folosită de foarte multe aplicații din toate domeniile, precum: SEGA, EA, SAP, Adobe, Google. Este o bază de date non-relațională cu performanțe extraordinare datorită conversiei din JSON în BSON înainte de stocare. Datele sunt stocate sub formă de colecții, asemănătoare unei baze de date SQL, având în plus flexibilitatea și posibilitatea unei structuri mult mai complexe, realizând partea de model a platformei.



Figură 19 Diagrama Entitate-Relație

Colecțiile folosite în cadrul aplicației sunt:

- Utilizatorul – conține informațiile necesare despre utilizator precum nume, greutate, înălțime, data nașterii, sex. Se salvează doar emailul și usernameul pe lângă celelalte detalii deoarece modulul pentru node numit passport a fost folosit pentru a cripta parola. Sunt stocate în baza de date și referințe către utilizatori, rețete și exerciții pentru funcționalitățile legate de urmărirea altor persoane care folosesc platforma și cele pentru adăugarea la favorite ale rețetelor și exercițiilor.
- Exercițiul – fiecare dintre exercițiile prezente în cadrul aplicației are o imagine cu zona musculară aferentă acestuia, tipul de mușchi lucrat și instrucțiuni detaliate despre cum se poate efectua.
- Rețeta – în cadrul rețetelor, sunt prezente imagini cu acestea, informații despre cum se prepară și despre valorile nutriționale, precum și o listă de ingrediente despre care sunt cunoscute numele, cantitatea și unitatea de măsură, adăugate la crearea unei rețete.
- Competiția – aceasta are o dată de început și una de sfârșit, detalii despre tipul de competiție, maraton sau ardere de calorii; adminul decide câștigătorul în urma unei analize asupra participantilor.

4.3 Diagrama de workflow



Figură 20 Diagrama de workflow

Pentru a înțelege pașii pe care un utilizator trebuie să îi urmeze pentru a se putea bucura de funcționalitățile platformei web, a fost realizată o diagramă a workflowului aplicației. Pentru a avea acces la acestea, fiecare utilizator va avea nevoie de un cont. Fie că are deja unul și se conectează la el sau nu are și își creează unul nou, acesta va fi redirecționat către pagina de pornire. Aici este prezent un carusel de imagini, din care se poate ajunge ușor la exerciții, rețete sau la competiție. Pe langă acestea, o altă modalitate de accesarea diferitei pagini din cadrul aplicației este bara de navigație prezentă în partea superioară a paginii.



Figură 21 Bara de navigație a adminului



Figură 22 Bara de navigație a utilizatorului basic



Figură 23 Bara de navigație a utilizatorului premium

5 DETALII DE IMPLEMENTARE

5.1 Dependințe și model arhitectural

Pentru a putea rula aplicația, următoarele module pentru node, cu versiunile aferente, trebuie să fie prezente pe mașina pe care se pornește serverul.



```
1 "dependencies": {
2   "connect-flash": "^0.1.1",
3   "connect-mongo": "^5.0.0",
4   "ejs": "^3.1.8",
5   "ejs-mate": "4.0.0",
6   "express": "4.18.2",
7   "express-session": "1.17.3",
8   "joi": "17.9.2",
9   "method-override": "3.0.0",
10  "mongoose": "6.10.0",
11  "passport": "0.6.0",
12  "passport-local": "1.0.0",
13  "passport-local-mongoose": "7.1.2"
14 }
```

Figură 24 Dependințe

Implementarea aplicației a fost realizată prin folosirea unui model arhitectural destul de întâlnit în dezvoltarea platformelor web numit Model-view-controller(MVC)[13].

- Componenta controller este responsabilă pentru comunicarea dintre celelalte două, astfel în cadrul acesteia este realizată toată logica aplicației.
- Componenta view se ocupă cu partea de interfață prezentată utilizatorului, precum și interacțiunea dintre acesta și partea de backend.
- Componenta model este reprezentată de felul în care informațiile sunt stocate în baza de date, acestea fiind modificate sau extrase de către controller și transmise către view.
- Pe lângă acestea, se mai adaugă și rutarea realizată cu ajutorul funcțiilor oferite de Express și sporește securitatea platformei, folosind o combinație de logică prezentă în controller și alte funcții definite cu scopul de a restricționa accesul la anumite endpointuri.

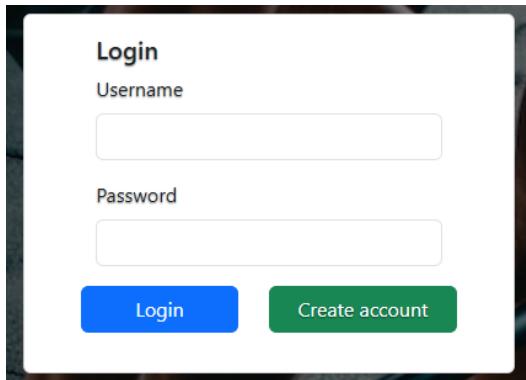


```
1 module.exports.isLoggedIn = (req, res, next) => {
2   if (!req.isAuthenticated()) {
3     req.flash('error', 'You must be logged in!');
4     return res.redirect('/login');
5   }
6   next();
7 }
8
9 module.exports.isAdmin = (req, res, next) => {
10  if (req.user.role !== 'Admin') {
11    req.flash('error', 'You must be an admin!');
12    return res.redirect('/');
13  }
14  next();
15 }
16
17 module.exports.isPremium = (req, res, next) => {
18  if (req.user.role === 'Basic') {
19    req.flash('error', 'You must be a premium user!');
20    return res.redirect('/');
21  }
22  next();
23 }
```

Figură 25 Funcții pentru restricționarea accesului

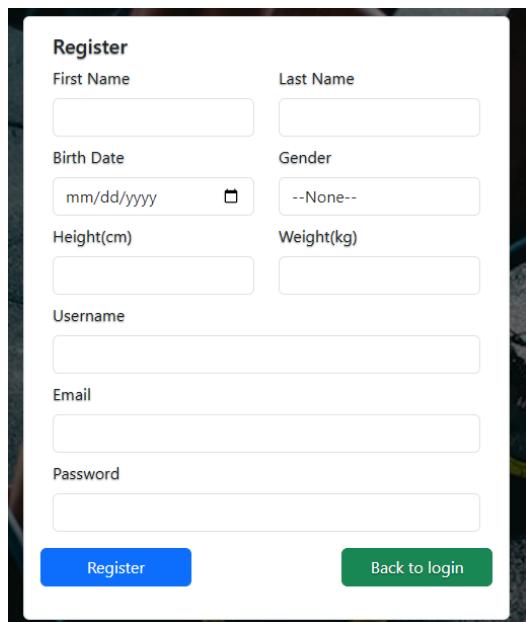
5.2 Funcționalități

5.2.1 Logare, înregistrare și profil



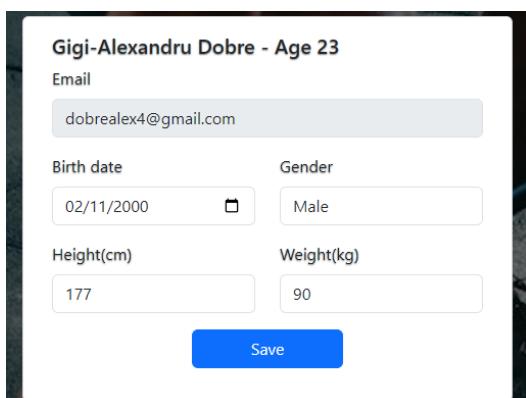
The login form consists of a white rectangular box with a black border. At the top center, it says "Login". Below that is a "Username" label with an input field. Below the input field is a "Password" label with an input field. At the bottom left is a blue "Login" button, and at the bottom right is a green "Create account" button.

Figură 26 Formular de logare



The registration form is a larger white rectangular box with a black border. It contains fields for "First Name" and "Last Name" (each with its own input field), "Birth Date" (with a date picker icon), "Gender" (with a dropdown menu showing "--None--"), "Height(cm)" and "Weight(kg)" (each with its own input field), "Username" (input field), "Email" (input field), and "Password" (input field). At the bottom left is a blue "Register" button, and at the bottom right is a green "Back to login" button.

Figură 27 Formular de înregistrare

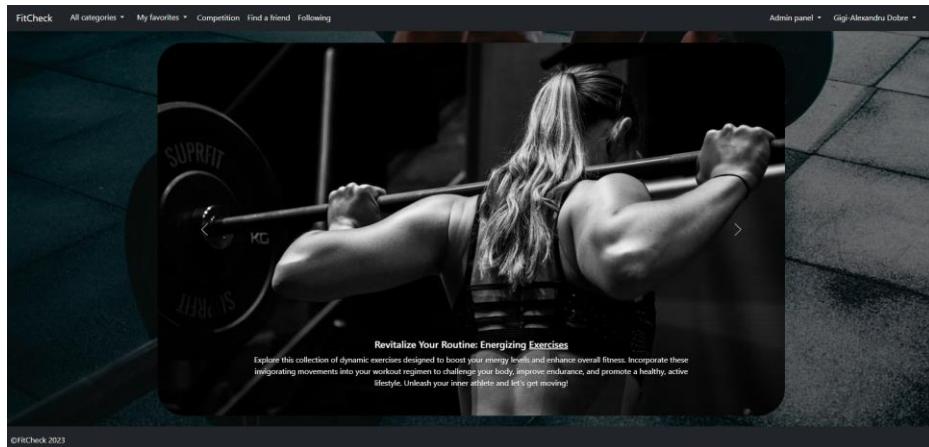


The profile edit form is a white rectangular box with a black border. At the top, it displays "Gigi-Alexandru Dobre - Age 23". It includes fields for "Email" (input field containing "dobrealex4@gmail.com"), "Birth date" (input field containing "02/11/2000" with a date picker icon), "Gender" (input field containing "Male"), "Height(cm)" (input field containing "177"), "Weight(kg)" (input field containing "90"), and a "Save" button at the bottom.

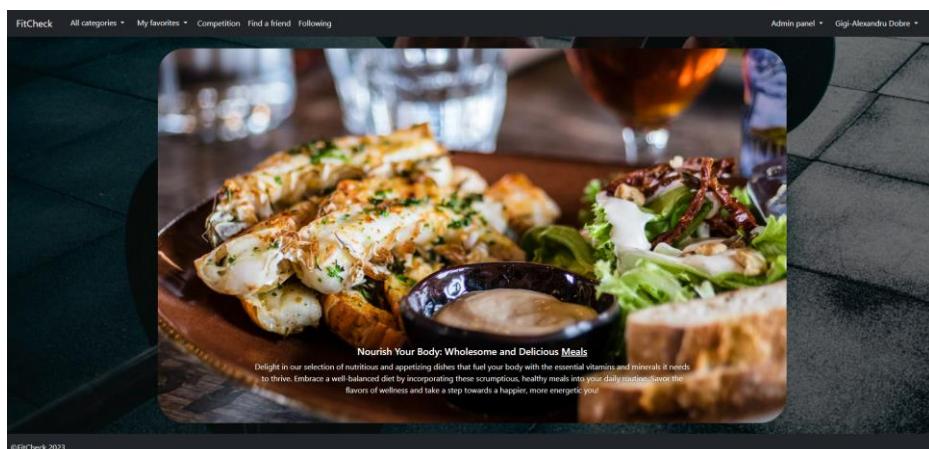
Figură 28 Formular de editare a profilului

5.2.2 Pagina de pornire

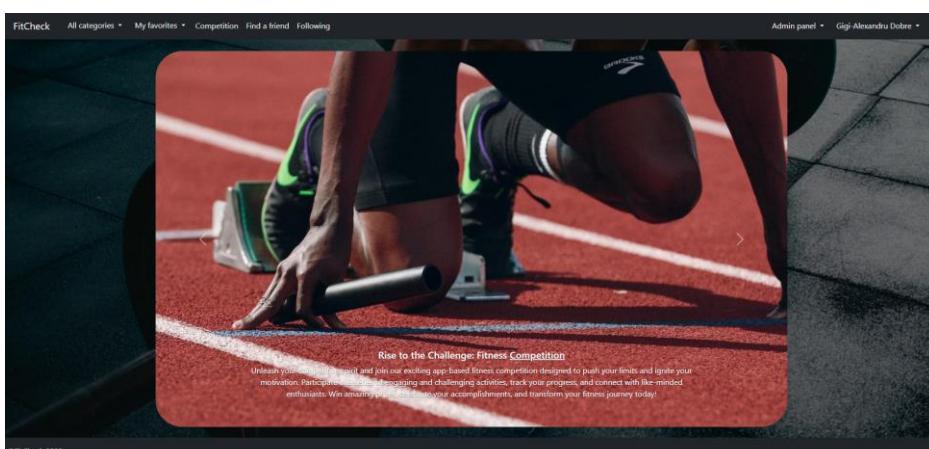
Fiecare dintre colecțiile prezentate anterior dispun de un fișier pentru controller, unul pentru rutare și unul pentru model. Din punctul de vedere al viewului, exercițiile, rețetele și competițiile au 4 fișiere: listă, listă admin, editare și creare. Pentru utilizator mai există și înregistrare, logare, profil, lista persoanelor pe care le urmărește și lista tuturor clienților.



Figură 29 Carusel exerciții

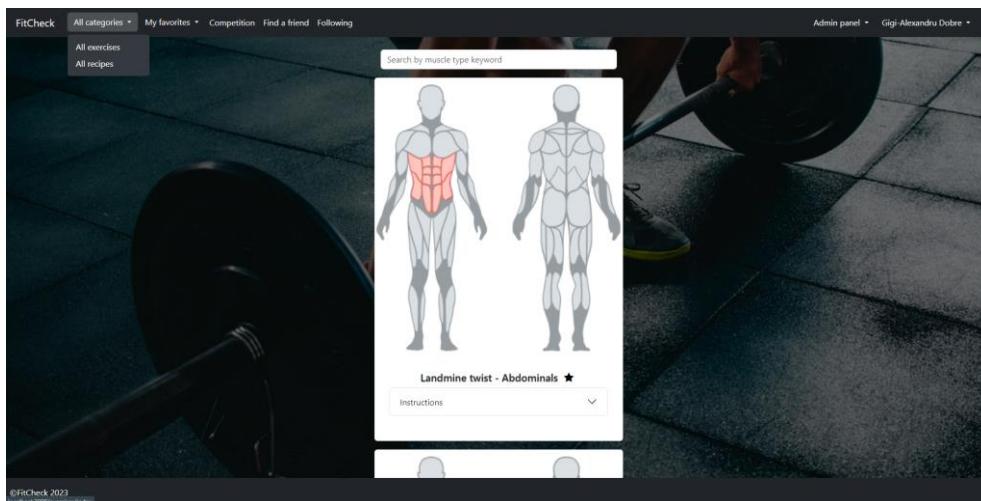


Figură 30 Carusel rețete



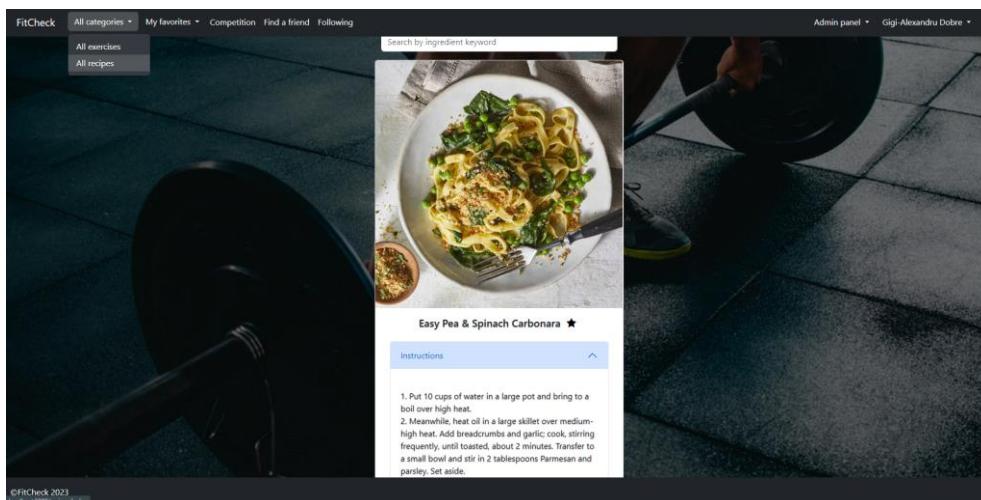
Figură 31 Carusel competiție

5.2.3 Utilizator basic



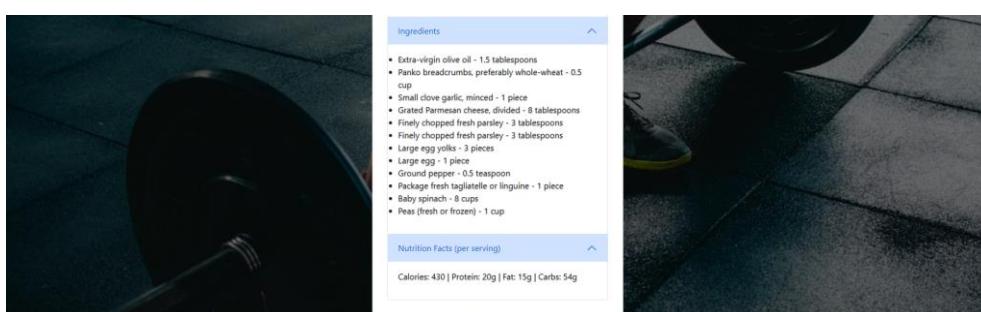
Figură 32 Exerciții

Pentru a facilita găsirea exercițiilor dorite în funcție de grupa musculară, se poate folosi butonul de căutare, unde, indiferent de folosirea majusculelor sau a minusculelor, se poate introduce un cuvânt cheie.

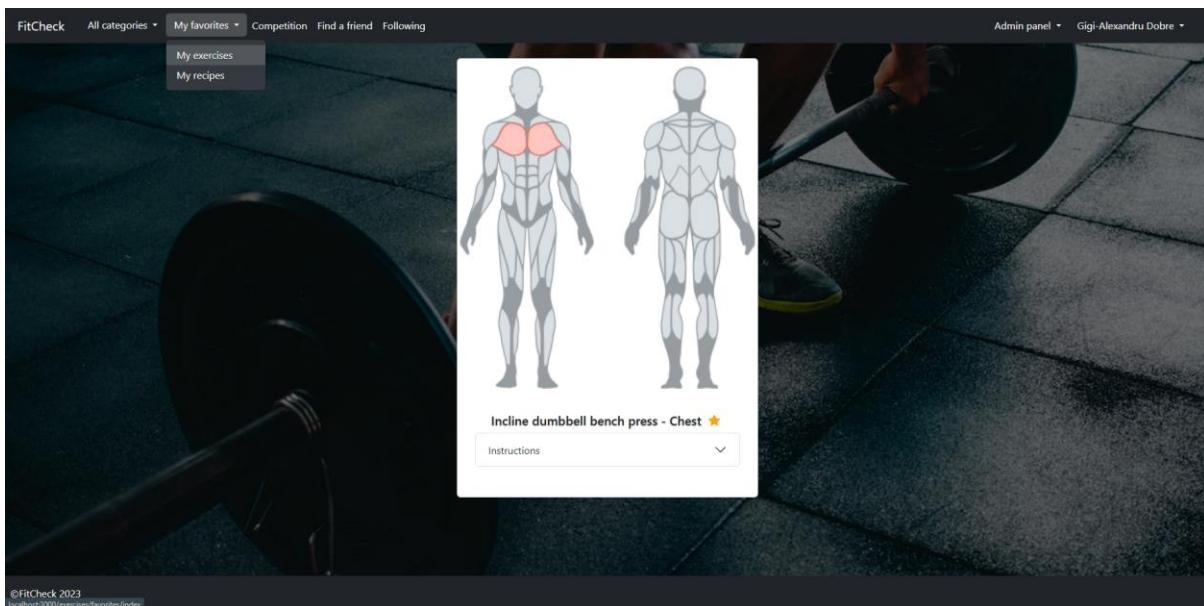


Figură 33 Rețete

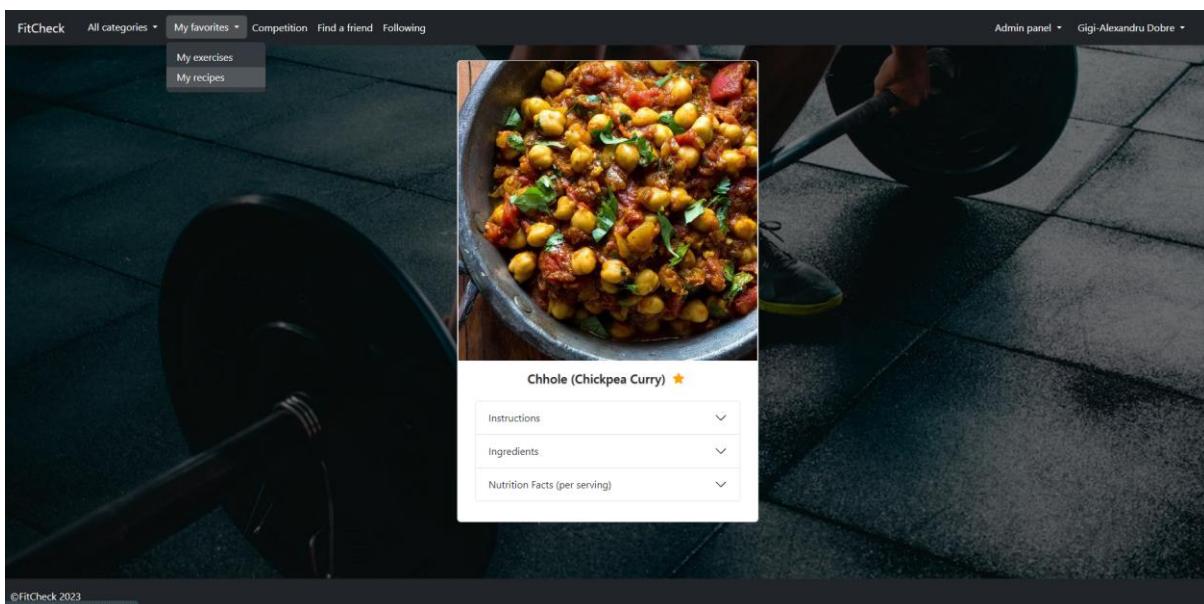
Același lucru se aplică și în cadrul rețetelor. Pe lângă instrucțiuni, în cadrul rețetelor sunt afișate și ingredientele și valorile nutriționale a unei porții.



Figură 34 Ingrediente și valorile nutriționale pentru rețete



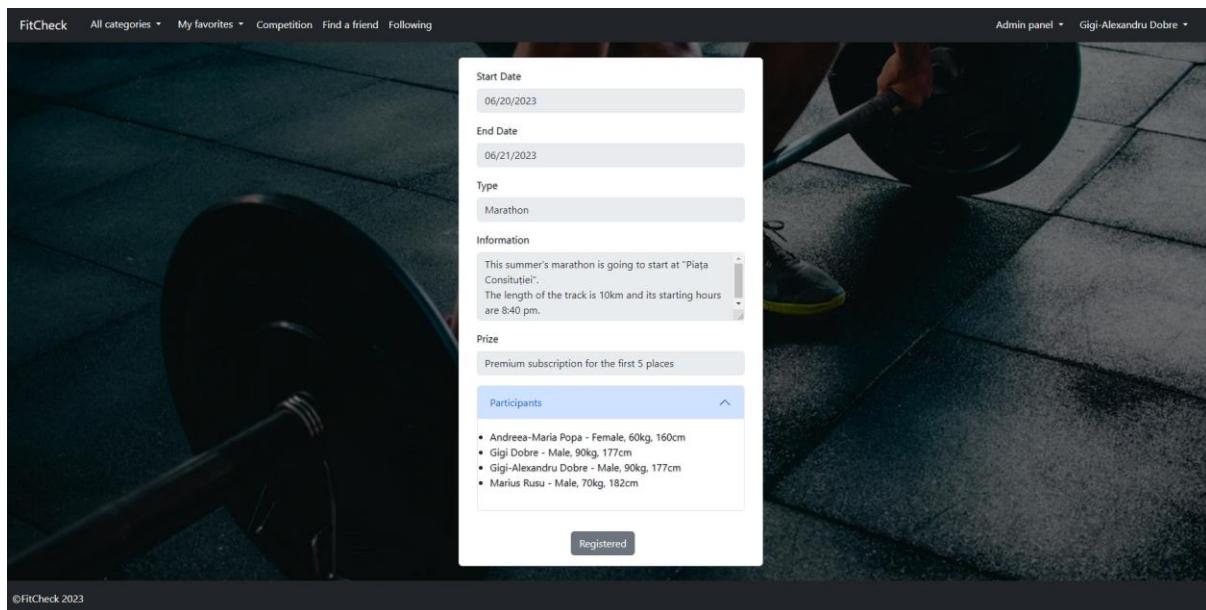
Figură 35 Exerciții favorite



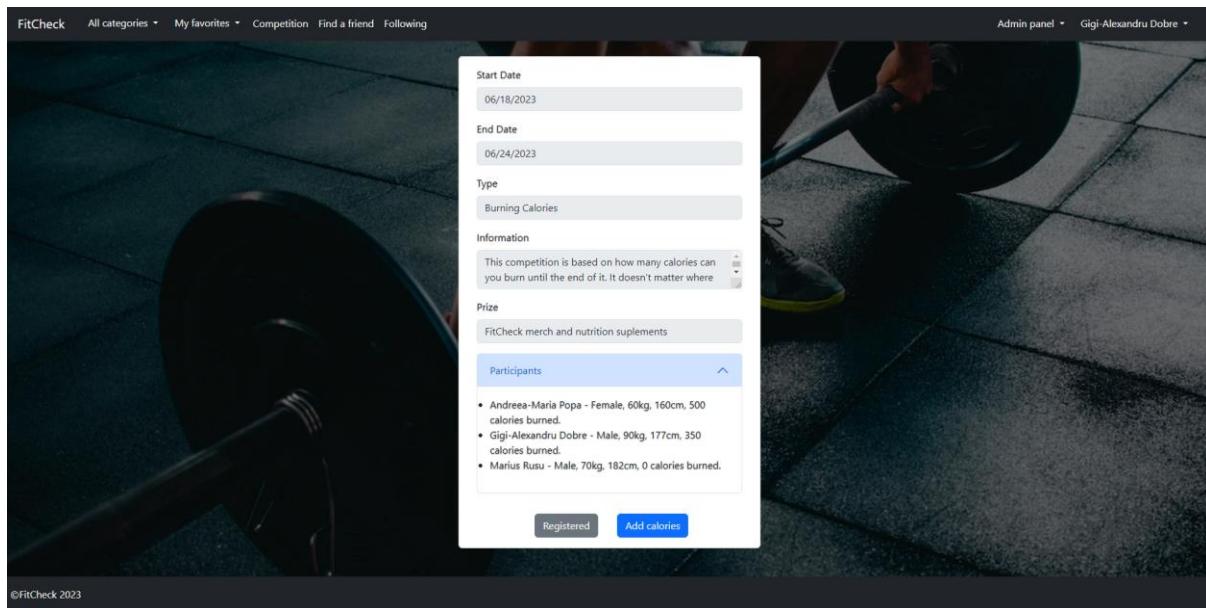
Figură 36 Rețete favorite

Pentru a salva la favorite exercițiile sau rețetele care i-au atras atenția utilizatorului și pe care ar dori să le aibă la îndemână atunci când le pune în practică, acesta trebuie să apese butonul în formă de stea pe paginile prezentate anterior. Pentru a evita scoaterea acestora din greșeală, în cazul în care este apăsat acest buton în cadrul favoritelor, elementul nu este șters de pe pagină până la următoarea reîmprospătare.

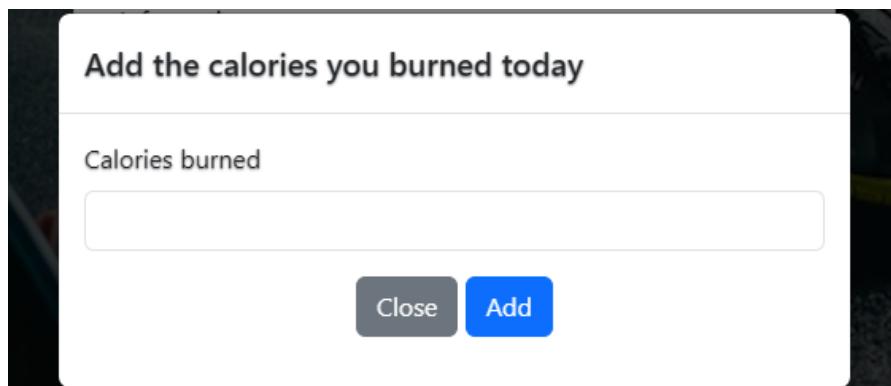
Un utilizator se poate înscrie până în momentul în care competiția începe. După aceea are posibilitatea de a ieși din competiție și de a-și adăuga calorii, în cazul în care tipul este arderea de calorii.



Figură 37 Competiție de tip maraton și lista cu participanți



Figură 38 Competiție de tip ardere de calorii și lista de participanți



Figură 39 Modal pentru adăugare de calorii

5.2.4 Utilizator premium

The screenshot shows the 'Following' section of the FitCheck app. At the top, there's a search bar labeled 'Search by name keyword'. Below it, three friend profiles are listed in cards:

- Gigi Dobre**
Email: alex@alex
Gender: Male
Height: 177 cm
Weight: 90 kg
Follow
- Marius Rusu**
Email: mariusur@gmail.com
Gender: Other
Height: 182 cm
Weight: 69 kg
Followed
- Andreea-Maria Popa**
Email: andreeapopa23@gmail.com
Gender: Female
Height: 160 cm

At the bottom left, there's a copyright notice: ©FitCheck 2023.

Figură 40 Găsește un prieten

The screenshot shows the profile page for Marius Rusu. At the top, his basic information is displayed:

- Marius Rusu**
- Email: mariusur@gmail.com
- Gender: Other
- Height: 182 cm
- Weight: 69 kg

Below this, there are two expandable sections:

- Favorite Exercises**
 - Landmine twist - Abdominals
 - Elbow plank - Abdominals
- Favorite Recipes**
 - Easy Pea & Spinach Carbonara - Extra-virgin olive oil, Panko breadcrumbs, preferably whole-wheat, Small clove garlic, minced, Grated Parmesan cheese, divided, Finely chopped fresh parsley, Large egg yolks, Large egg, Ground pepper, Package fresh tagliatelle or linguine, Baby spinach, Peas (fresh or frozen)

At the bottom left, there's a copyright notice: ©FitCheck 2023.

Figură 41 Persoanele urmărite

Opțiunea de a urmări alți utilizatori și de a vedea exercițiile și rețetele lor favorite este specifică celor cu abonament premium și este dedicată persoanelor pasionate care câștigă competiții, pentru a-și forma o rețea de prieteni. Acest lucru nu influențează dezvoltarea personală a unei persoane și în același timp protejează utilizatorii deoarece, pentru o mai bună filtrare și deosebire, este afișată adresa de email.

6 STUDIU DE CAZ / EVALUAREA REZULTATELOR

Evaluarea rezultatelor aplicației se împarte în 3 subcategorii: o sută de teste efectuate pe operațiile de ștergere și editare ale aplicației din punctul de vedere al adminului, excludând funcționalitățile demonstate în secțiunea detaliu de implementare, metrii de performanță din cadrul utilitarului Lighthouse de la Google și feedback primit din partea colegilor cărora le-am prezentat aplicația.

6.1 Teste ștergere și editare

6.1.1 Utilizatori

The screenshot shows the FitCheck application's Admin panel. At the top, there are navigation links: FitCheck, All categories, My favorites, Competition, Find a friend, Following, Admin panel (selected), and Gigi-Alexandru Dobre. On the left, there is a sidebar with the same navigation links. The main content area displays a list of users:

- Gigi-Alexandru Dobre - alexandrudb1
Email: dobrealx4@gmail.com Role: Admin
Edit Delete
- Gigi Dobre - alexandrudb1
Email: alex@alex Role: Premium
Edit Delete
- Marius Rusu - marius.rusu
Email: mariusur@gmail.com Role: Premium
Edit Delete
- Andreea-Maria Popa - andri.popa

At the bottom left, it says ©FitCheck 2023 localhost:1000/users/admin/index.

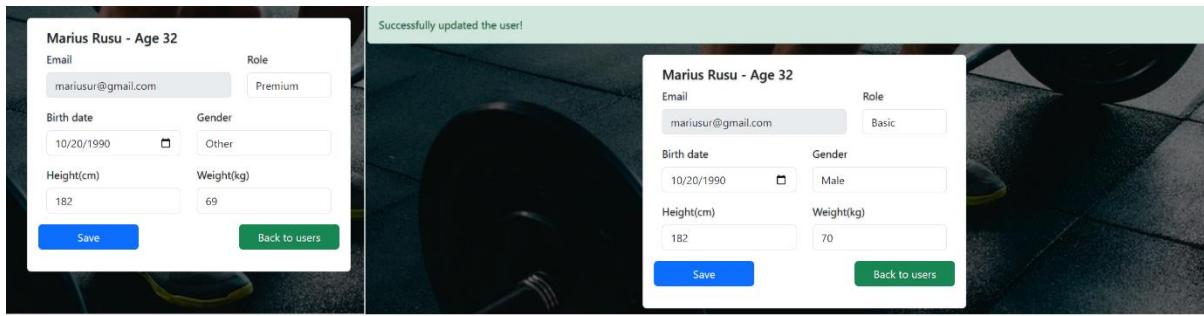
Figură 42 Listă utilizatori

The screenshot shows the FitCheck application's Admin panel after a user has been deleted. A success message 'Successfully deleted the user!' is displayed in a green box at the top. The main content area shows the remaining users:

- Gigi-Alexandru Dobre - alexandrudb1
Email: dobrealx4@gmail.com Role: Admin
Edit Delete
- Marius Rusu - marius.rusu
Email: mariusur@gmail.com Role: Premium
Edit Delete
- Andreea-Maria Popa - andri.popa
Email: andreeapopaz3@gmail.com Role: Basic
Edit Delete

At the bottom left, it says ©FitCheck 2023 localhost:1000/users/admin/index.

Figură 43 Ștergea utilizatorului alexandrudb1

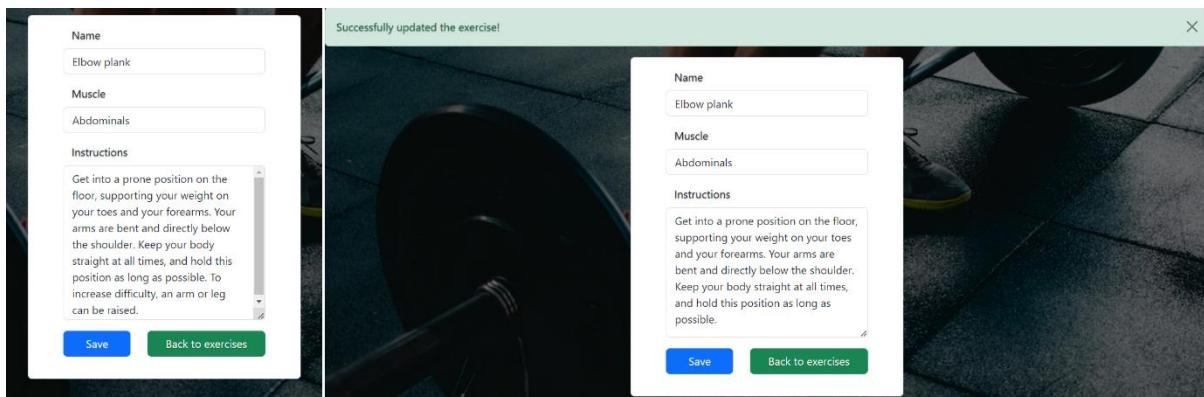


Figură 44 Înainte și după modificarea detaliilor unui utilizator

6.1.2 Exerciții

Figură 45 Listă exerciții

Figură 46 Ștergere exercițiu numit Landmine twist



Figură 47 Exercițiu înainte și după modificarea instrucțiunilor

6.1.3 Rețete

Easy Pea & Spinach Carbonara

1. Put 10 cups of water in a large pot and bring to a boil over high heat. 2. Meanwhile, heat oil in a large skillet over medium-high heat. Add breadcrumbs and garlic; cook, stirring frequently, until toasted, about 2 minutes. Transfer to a small bowl and stir in 2 tablespoons Parmesan and parsley. Set aside. 3. Whisk the remaining 6 tablespoons Parmesan, egg yolks, egg, pepper and salt in a medium bowl. 4. Cook pasta in the boiling water, stirring occasionally, for 1 minute. Add spinach and peas and cook until the pasta is tender, about 1 minute more. Reserve 1/4 cup of the cooking water. Drain and place in a large bowl. 5. Slowly whisk the reserved cooking water into the egg mixture. Gradually add the mixture to the pasta, tossing with tongs to combine. Serve topped with the reserved breadcrumb mixture.

- Extra-virgin olive oil - 1.5 tablespoons
- Panko breadcrumbs, preferably whole-wheat - 0.5 cup
- Small clove garlic, minced - 1 piece
- Grated Parmesan cheese, divided - 8 tablespoons
- Finely chopped fresh parsley - 3 tablespoons
- Finely chopped fresh parsley - 3 tablespoons
- Large egg yolks - 3 pieces
- Large egg - 1 piece
- Ground pepper - 0.5 teaspoon
- Package fresh tagliatelle or linguine - 1 piece
- Baby spinach - 8 cups
- Peas (fresh or frozen) - 1 cup

[Edit](#) [Delete](#)

Figură 48 Listă rețete

Successfully deleted the recipe!

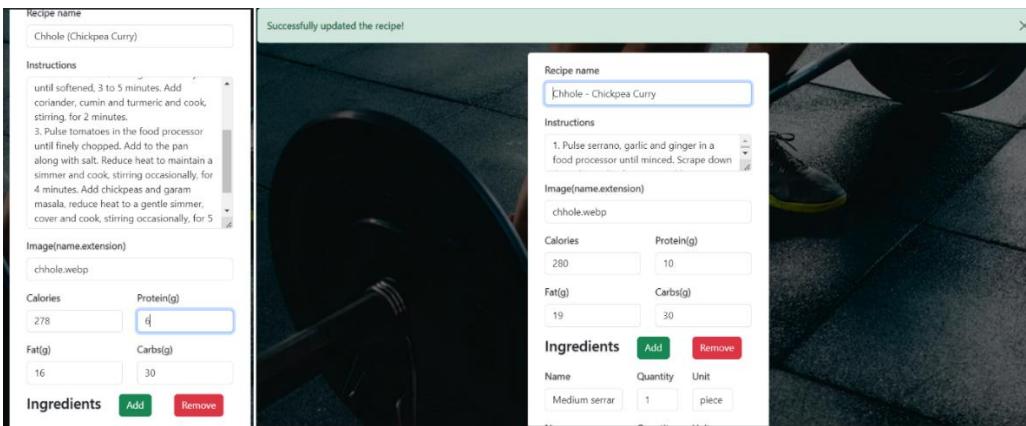
Search by ingredient keyword

Chhole (Chickpea Curry)

1. Pulse serrano, garlic and ginger in a food processor until minced. Scrape down the sides and pulse again. Add onion; pulse until finely chopped, but not watery. 2. Heat oil in a large saucpan over medium-high heat. Add the onion mixture and cook, stirring occasionally, until softened, 3 to 5 minutes. Add coriander, cumin and turmeric and cook, stirring, for 2 minutes. 3. Pulse tomatoes in the food processor until finely chopped. Add to the pan along with salt. Reduce heat to maintain a simmer and cook, stirring occasionally, for 4 minutes. Add chickpeas and garam masala, reduce heat to a gentle simmer, cover and cook, stirring occasionally, for 5 minutes more. Serve topped with cilantro, if desired.

- Medium serrano pepper - 1 piece
- Large cloves garlic - 4 pieces
- Fresh ginger, peeled and coarsely chopped - 1 piece
- Medium yellow onion, chopped (1-inch) - 1 piece
- Canola oil or grapeseed oil - 6 tablespoons
- Ground coriander - 2 teaspoons
- Ground cumin - 2 teaspoons
- Ground turmeric - 0.5 teaspoon
- No-salt-added canned diced tomatoes with their juice (from a 28-ounce can) - 2.25 cups

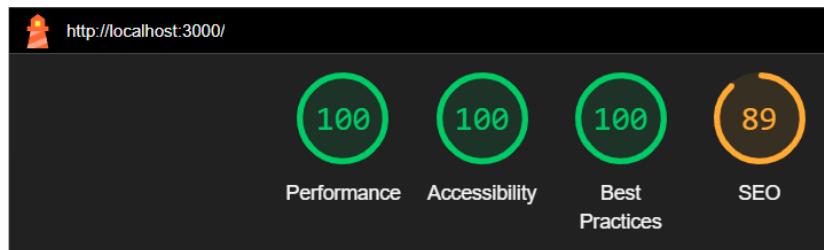
Figură 49 Ștergere rețetă numită Easy Pea & Spinach Carbonara



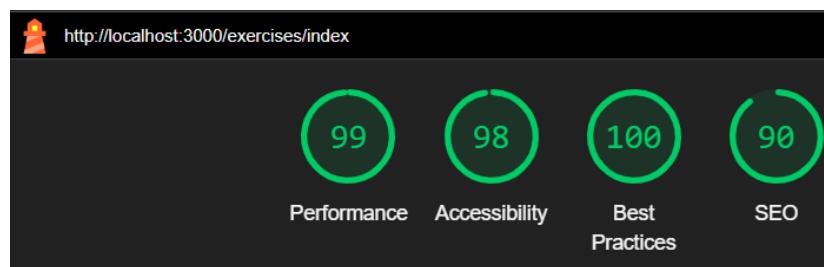
Figură 50 Rețetă înainte și după modificarea numelui și a valorilor nutriționale

6.2 Lighthouse teste de performanță

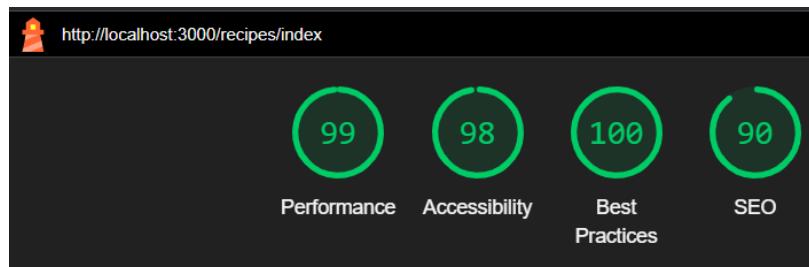
Pentru a măsura performanța, accesibilitatea, SEO-ul paginii, dar și dacă am respectat cele mai bune practici în realizarea aplicației, utilitarul celor de la Google numit Lighthouse a fost folosit pentru endpoint-urile cu importanță pentru utilizatori și cu încărcare mai mare decât celelalte.



Figură 51 Pagină de pornire



Figură 52 Listă de exerciții



Figură 53 Listă de rețete

6.3 Feedback

Pentru a constitui o listă de îmbunătățiri pe care aş putea să le aduc, am realizat o prezentare cu demo a aplicației și am prezentat-o colegilor de muncă, atât pasionați de sport, dar și care își doresc să înceapă un stil de viață mai sănătos. Astfel, aceștia au conceput următoarele sugestii:

- Atât pentru exerciții, cât și pentru rețete, să existe paginare, iar elementele să fie poziționate pe același rând.
- O nouă zonă în care să se sugereze rețete și exerciții de către utilizatori.
- Exercițiile să fie grupate în funcție de zonele musculare.
- Un meniu într-una dintre părțile paginii care servește drept feed de activitate.
- Un chat între utilizatori pentru a putea comunica și a se cunoaște mai bine.

7 CONCLUZII

În primă fază, am desfășurat o cercetare (sondaj de opinie) în rândul altor colegi de facultate, colegi de la alte facultăți și colegi de muncă cu privire la impactul sedentarismului și obezității și nevoile lor privind o eventuală aplicație de fitness și nutriție. Conform rezultatelor, toți au fost de acord ca sedentarismul este o problemă importantă, iar o mare parte dintre aceștia și-ar dori o aplicație fitness care să le satisfacă nevoile. Prin acest sondaj am urmărit să aflu nivelul de receptivitate a respondenților la diferite funcționalități pe care, ulterior, le-am și implementat.

După stabilirea funcționalităților, am realizat o analiză a tehnologiilor disponibile pe piață, cu un suport foarte bun din partea celor care le-au creat și a comunității de dezvoltatori. Astfel, am hotărât ca în cadrul aplicației să folosesc atât pentru backend, cât și pentru frontend, frameworkuri care se bazează pe JavaScript. Pentru a stoca toate informațiile necesare, am ales o bază de date non-relațională datorită capabilității acesteia de a folosi structuri complexe pentru colecțiile definite.

În continuare, după finalizarea implementării aplicației, am realizat o prezentare cu scopul de obține un feedback din partea potențialilor clienți. Am efectuat un demo al aplicației în urmă căruia am descoperit o serie de îmbunătățiri pe care le pot aduce platformei. Astfel, pe lângă dorința personală de integra aplicația de web cu o aplicație de mobil și smartwatch pentru a ușura accesul la funcționalitățile dezvoltate, am reușit să obțin indicații cu privire la aspectul aplicației și la noi caracteristici care pot spori numărul de utilizatori și facilită experiența lor.

Faptul că piața țintă este una în creștere constantă scoate în evidență nevoia dezvoltării unor astfel de platforme, motiv pentru care am decis să obțin un produs care să se concentreze pe cele mai importante aspecte ale unui antrenament bine realizat. Astfel, am reușit îmbinarea exercițiilor și rețetelor într-o manieră în care utilizatorii să aibă acces oricând la acestea cu o reprezentare diferită față de celelalte aplicații de fitness. Prin integrarea competițiilor cu diferite premii, consider că motivația oamenilor de a participa la acestea, dar și de a concura cu ei însăși pentru o viață mai sănătoasă ar crește, rezultatele fiind pozitive indiferent de poziția acestora în competiție.

Concluzionând, prin diversele sale funcționalități, aplicația FitCheck urmărește să-i ajute pe utilizatori să facă o schimbare în stilul lor de viață prin a le oferi o varietate de funcționalități care să eliminate limitările celorlalte.

8 BIBLIOGRAFIE

[1] Fitness Blender [Online]. Available:

<https://www.fitnessblender.com/page/about-fitness-blender> [Accessed 13 06 2023].

[2] Fitness Blender antrenamente [Online]. Available:

<https://www.fitnessblender.com/workouts> [Accessed 13 06 2023]

[3] DAREBEE [Online]. Available:

<https://darebee.com/about.html> [Accessed 14 06 2023]

[4] Academy of Nutrition and Dietetics: eatright.org [Online]. Available:

<https://www.eatright.org/about-the-academy> [Accessed 14 06 2023]

[5] Node.js [Online]. Available:

<https://nodejs.dev/en/learn/> [Accessed 14 06 2023]

[6] ExpressJS [Online]. Available:

<https://expressjs.com/> [Accessed 14 06 2023]

[7] Spring Boot [Online]. Available:

<https://spring.io/projects/spring-boot#overview> [Accessed 19 06 2023]

[8] Angular [Online]. Available:

<https://angular.io/guide/what-is-angular> [Accessed 19 06 2023]

[9] React [Online]. Available:

<https://react.dev/learn> [Accessed 20 06 2023]

[10] Bootstrap [Online]. Available:

<https://getbootstrap.com/docs/5.3/about/overview/> [Accessed 20 06 2023]

[11] ejs-mate [Online]. Available:

<https://www.npmjs.com/package/ejs-mate> [Accessed 20 06 2023]

[12] MongoDB [Online]. Available:

<https://www.mongodb.com/> [Accessed 20 06 2023]

[13] MVC [Online]. Available:

<https://developer.mozilla.org/en-US/docs/Glossary/MVC> [Accessed 20 06 2023]

9 ANEXE

```
1 const express = require('express');
2 const router = express.Router();
3 const users = require('../controllers/users');
4 const passport = require('passport');
5 const catchAsync = require('../utils/catchAsync');
6 const {isLoggedIn, isAdmin, isPremium} = require('../middleware');
7
8 router.route('/register')
9   .get(users.renderRegister)
10  .post(catchAsync(users.register));
11
12 router.route('/login')
13   .get(users.renderLogin)
14   .post(passport.authenticate('local', { failureFlash: true, failureRedirect: '/login' }), users.login);
15
16 router.route('/profile')
17   .get(isLoggedIn, users.renderProfile)
18   .put(isLoggedIn, catchAsync(users.updateProfile));
19
20 router.route('/users/admin/index')
21   .get(isLoggedIn, isAdmin, catchAsync(users.adminIndex));
22
23 router.route('/users/friends/index')
24   .get(isLoggedIn, isPremium, catchAsync(users.friendsIndex));
25
26 router.route('/users/following')
27   .get(isLoggedIn, isPremium, catchAsync(users.followingIndex));
28
29 router.route('/users/:id')
30   .get(isLoggedIn, isAdmin, catchAsync(users.renderEdit))
31   .put(isLoggedIn, isAdmin, catchAsync(users.editUser))
32   .delete(isLoggedIn, isAdmin, catchAsync(users.deleteUser));
33
34 router.route('/users/:id/addFavoriteExercise/:idEx')
35   .put(isLoggedIn, catchAsync(users.addFavoriteExercise));
36
37 router.route('/users/:id/addFavoriteRecipe/:idRec')
38   .put(isLoggedIn, catchAsync(users.addFavoriteRecipe));
39
40 router.route('/users/:id/followFriend/:idFriend')
41   .put(isLoggedIn, isPremium, catchAsync(users.followFriend));
42
43 router.route('/users/friends/index/:name')
44   .get(isLoggedIn, isPremium, catchAsync(users.friendIndex));
45
46 router.route('/users/admin/index/:name')
47   .get(isLoggedIn, isAdmin, catchAsync(users.nameAdminIndex));
48
49 router.get('/logout', users.logout);
50
51 module.exports = router;
```

Figură 54 Exemplu rutare

```

1 const Recipe = require('../models/recipe');
2 const User = require('../models/user');
3
4 module.exports.renderNew = (req, res) => {
5   res.render('recipes/new');
6 };
7
8 module.exports.createRecipe = async (req, res) => {
9   req.body.recipe.image = '/images/recipes/' + req.body.recipe.image + '.webp';
10  const exercise = new Recipe(req.body.recipe);
11  await exercise.save();
12  req.flash('success', 'Successfully created a new recipe!');
13  res.redirect('/');
14 }
15
16 module.exports.index = async(req, res) => {
17   const allRecipes = await Recipe.find({});
18   var favorites = false;
19   res.render('recipes/index', { allRecipes, favorites });
20 };
21
22 module.exports.favoriteIndex = async(req, res) => {
23   var allRecipes = await Recipe.find({ '_id': { $in: req.user.recipes.map(rec => rec.toString()) } });
24   var favorites = true;
25   res.render('recipes/index', { allRecipes, favorites });
26 };
27
28 module.exports.adminIndex = async(req, res) => {
29   const allRecipes = await Recipe.find({});
30   res.render('recipes/adminIndex', { allRecipes });
31 };
32
33 module.exports.deleteRecipe = async(req, res) => {
34   const { id } = req.params;
35   const allUsers = await User.find({});
36   for (let user of allUsers) {
37     var idx = user.recipes.findIndex(rec => rec.toString() === id);
38     if (idx != -1) {
39       user.recipes.splice(idx, 1);
40     }
41     await User.findByIdAndUpdate(user.id, { ...user });
42   }
43   await Recipe.findByIdAndDelete(id);
44   req.flash('success', 'Successfully deleted the recipe!');
45   res.redirect('/recipes/admin/index');
46 };
47
48 module.exports.renderEdit = async (req, res) => {
49   const { id } = req.params;
50   const recipe = await Recipe.findById(id);
51   if (!recipe) {
52     req.flash('error', 'Cannot find that recipe!');
53     return res.redirect('/recipes/admin/index');
54   }
55   res.render('recipes/edit', { recipe });
56 };
57
58 module.exports.editRecipe = async(req, res) => {
59   const { id } = req.params;
60   req.body.recipe.image = '/images/recipes/' + req.body.recipe.image;
61   await Recipe.findByIdAndUpdate(id, { ...req.body.recipe });
62   req.flash('success', 'Successfully updated the recipe!');
63   res.redirect('/recipes/${id}');
64 };
65
66 module.exports.ingredientIndex = async(req, res) => {
67   var { ingredient } = req.params;
68   var allRecipes = await Recipe.find({'ingredients.name': { $regex: ingredient, $options: 'i' }});
69   var favorites = false;
70   if (allRecipes.length == 0) {
71     req.flash('error', "There are no recipes containing this ingredient!");
72     return res.redirect('/recipes/index');
73   }
74   res.render('recipes/index', { allRecipes, favorites });
75 };
76
77 module.exports.ingredientAdminIndex = async(req, res) => {
78   var { ingredient } = req.params;
79   var allRecipes = await Recipe.find({'ingredients.name': { $regex: ingredient, $options: 'i' }});
80   if (allRecipes.length == 0) {
81     req.flash('error', "There are no recipes containing this ingredient!");
82     return res.redirect('/recipes/admin/index');
83   }
84   res.render('recipes/adminIndex', { allRecipes });
85 };
86

```

Figură 55 Exemplu controller

```

1  <% layout('layouts/boilerplate')%>
2  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
3  <% if (!favorites) { %>
4      <div class="row">
5          <form action="/exercises/index/" method="GET" class="mx-auto mb-3" onSubmit="submitConvertAction()" target="sub" style="width: 30rem;">
6              <input class="form-control" type="search" id="search_bar" placeholder="Search by muscle type keyword" aria-label="Search">
7          </form>
8      </div>
9  <% } %>
10 <% if (allExercises.length == 0) { %>
11     <div style="display: flex; justify-content: center; align-items: center; height: 70vh;">
12         <div style="width: 50rem;">
13             <div class="text-center" style="font-size: xx-large; color:whiteSmoke">
14                 <% if (!favorites) { %>
15                     Currently there are no exercises.
16                 <% } else { %>
17                     Currently there are no exercises added to favorites.
18                 <% } %>
19             </div>
20         </div>
21     </div>
22 <% } %>
23 <% for (let exercise of allExercises) { %>
24     <div class="row">
25         <div class="d-flex justify-content-center align-items-center card mb-3" style="width: 30rem;">
26             <img class="card-img-top mb-2" alt="<%= exercise.name %>" style="width: 80%; height: 80%; src="<%= exercise.image %>">
27             <div class="card-body" style="width: 100%">
28                 <div class="row mb-3 text-center">
29                     <h5 class="card-title">
30                         <%= exercise.name + " - " + exercise.muscle %>
31                     <form action="/users/<%= currentUser.id %>/addFavoriteExercise/<%= exercise.id %>" _method="PUT" method="POST" target="sub" style="display: inline;">
32                         <% if (currentUser.exercises.find(ex => ex.toString() === exercise.id) == undefined) { %>
33                             <button class="fa fa-star" aria-label="Button to save exercises to favorite"></button>
34                         <% } else { %>
35                             <button class="fa fa-star star-checked" aria-label="Button to save exercises to favorite"></button>
36                         <% } %>
37                     </form>
38                 </h5>
39                 <div class="accordion mb-3" id="accordionInstructions">
40                     <div class="accordion-item">
41                         <h2 class="accordion-header">
42                             <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
43                                 data-bs-target="#collapseInstructions<%= exercise.id %>" aria-expanded="false" aria-controls="collapseInstructions<%= exercise.id %>">
44                             Instructions
45                         </button>
46                     </h2>
47                     <div id="collapseInstructions<%= exercise.id %>" class="accordion-collapse collapse">
48                         <ul class="accordion-body">
49                             <p class="card-text"> <%= exercise.instructions %></p>
50                         </ul>
51                     </div>
52                 </div>
53             </div>
54         </div>
55     </div>
56 <% } %>
57 <iframe id="subframe" name="sub" style="display: none;"></iframe>
58
59 <script>
60     document.querySelectorAll('.fa-star').forEach(element => {
61         element.addEventListener('click', async function(e) {
62             if (e.target.classList.contains('star-checked')) {
63                 e.target.classList.remove('star-checked');
64             } else {
65                 e.target.classList.add('star-checked');
66             }
67         })
68     });
69
70     function submitConvertAction() {
71         location.href = document.querySelector('form').getAttribute('action') + this.search_bar.value;
72     }
73 </script>

```

Figură 56 Exemplu view