



# Nutrivia

de Asandei Ștefan-Alexandru

Clasa a X-a

**Colegiul Național Iași**

Profesor: Pădurariu Emanuela-Tatiana



# Cuprins

<b>1. Introducere</b>	
1.1 <b>Prezentare generală</b> .....	<b>2</b>
1.2 <b>Catalogul de produse</b> .....	<b>3</b>
1.3 <b>Pagina “For you”</b> .....	<b>4</b>
1.4 <b>Forum</b> .....	<b>5</b>
1.5 <b>Panou administrator</b> .....	<b>5</b>
<b>2. Inginerie web</b>	
2.1 <b>Frontend</b> .....	<b>6</b>
2.1.1 <b>Design</b> .....	<b>7</b>
2.1.2 <b>Accesibilitate</b> .....	<b>8</b>
2.1.3 <b>Experiența de utilizare</b> .....	<b>9</b>
2.2 <b>Backend</b>	
2.2.1 <b>Arhitectură</b> .....	<b>10</b>
2.2.2 <b>Baza de date</b> .....	<b>11</b>
<b>3. Aspecte tehnice</b>	
3.1 <b>Securitate</b> .....	<b>12</b>
3.2 <b>Testare</b> .....	<b>13</b>
3.3 <b>AI Generativ</b> .....	<b>13</b>
3.4 <b>Sistemul de provocări</b> .....	<b>14</b>
3.5 <b>Serializare</b> .....	<b>15</b>
<b>4. Bibliografie</b> .....	<b>16</b>

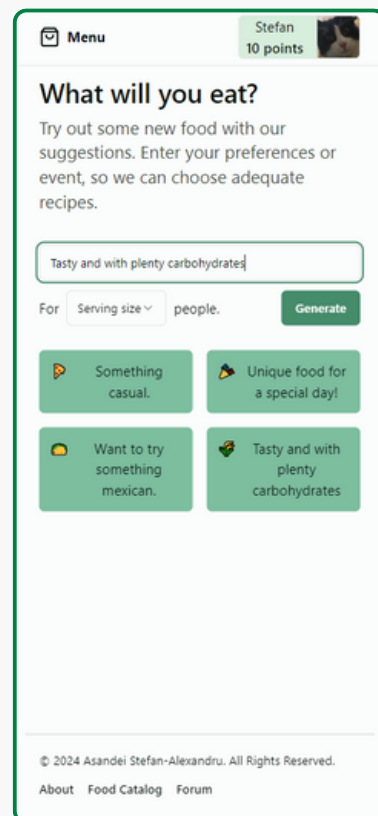
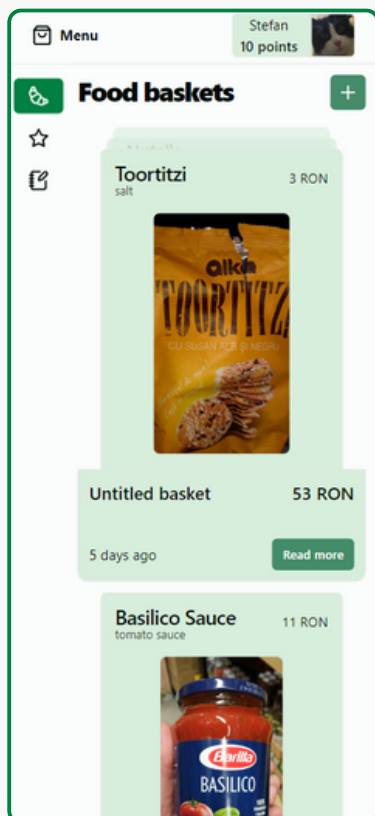


## 1.1 Prezentare generală

Nutrivia este o aplicație web ce încurajează utilizatorii să aibă o alimentație sănătoasă. Acest lucru este realizat prin colecția de produse cu informații nutriționale complete, mijloace de motivare și unelte de planificare a meselor. Aplicația se adresează oricăror persoane ce doresc să aibă control și să fie informate asupra a ceea ce consumă.

Paginile au fost gândite cu următoarele scopuri:

- Catalogul de produse: interfața completă cu toate informațiile nutriționale disponibile
- For you: planificarea meselor sănătoase, cât și monitorizarea progresului
- Forum: locul unde utilizatorii pot împărtăși rețete, pot vorbi despre experiența lor și își pot seta ținte comune

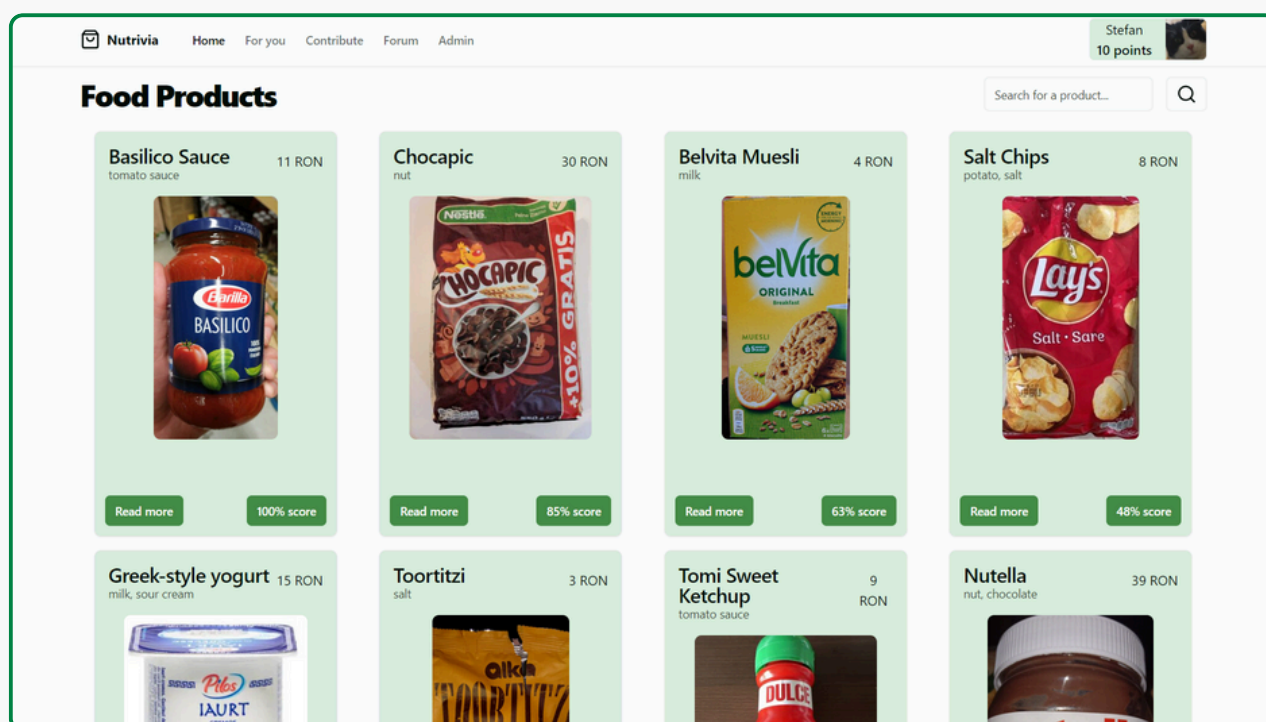


## 1.2 Catalogul de produse

Pe această pagină se găsesc toate produsele disponibile în aplicație. Acestea conțin informații nutriționale complete, disponibile împreună cu un scor nutrițional calculat pe baza calității nutriționale a produsului și a review-urilor lăsate de utilizatori.

Produsele pot fi adăugate de administrator, dintr-un panou specific, sau de utilizatori prin intermediul unei pagini de contribuție. Cererile trimise trebuie aprobate de administrator.

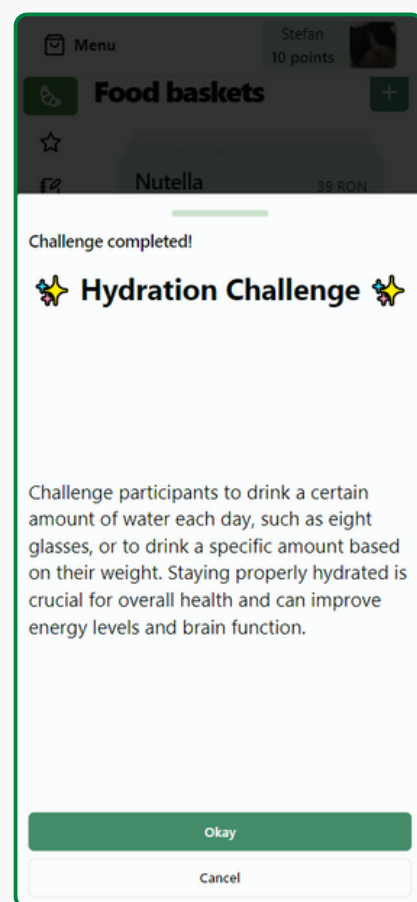
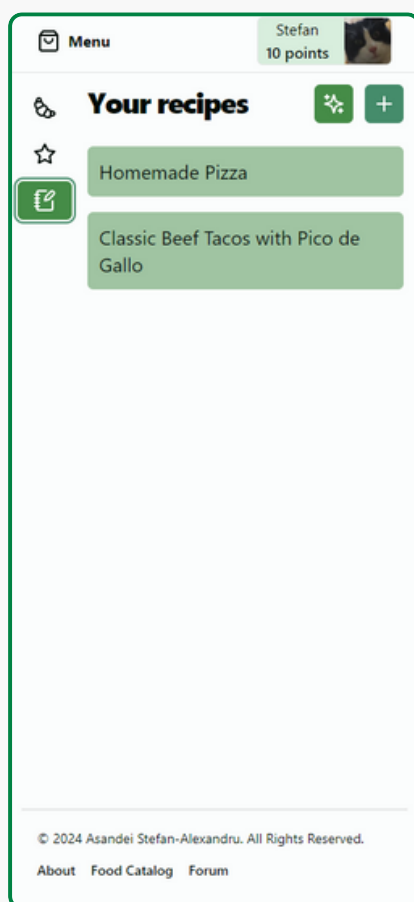
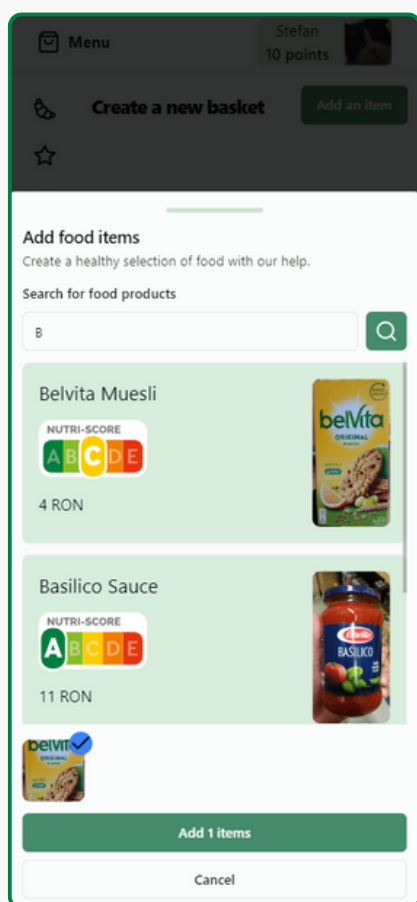
În funcție de numărul de puncte ale unui utilizator, un discount va fi aplicat produselor, pentru a încuraja consumul sănătos. Punctele sunt acordate în urma provocărilor, gândite să "gamifice" experiența de utilizare.



## 1.3 Pagina "For You"

Această pagină este împărțită în trei secțiuni:

- Coșuri de alimente: utilizatorii își pot planifica mesele creând coșuri, aceștia vor primi recomandări cu produse complementare pentru o masă echilibrată.
- Vizualizare progres: un calendar afișează progresul pe zile, semnificând provocările completate, încurajând utilizatorul să continue obiceiul.
- Rețetele mele: utilizatorii își pot organiza propriile rețete sau pot primi recomandări din partea unui sistem inteligent ce ia în considerare preferințele și alergiile lor.

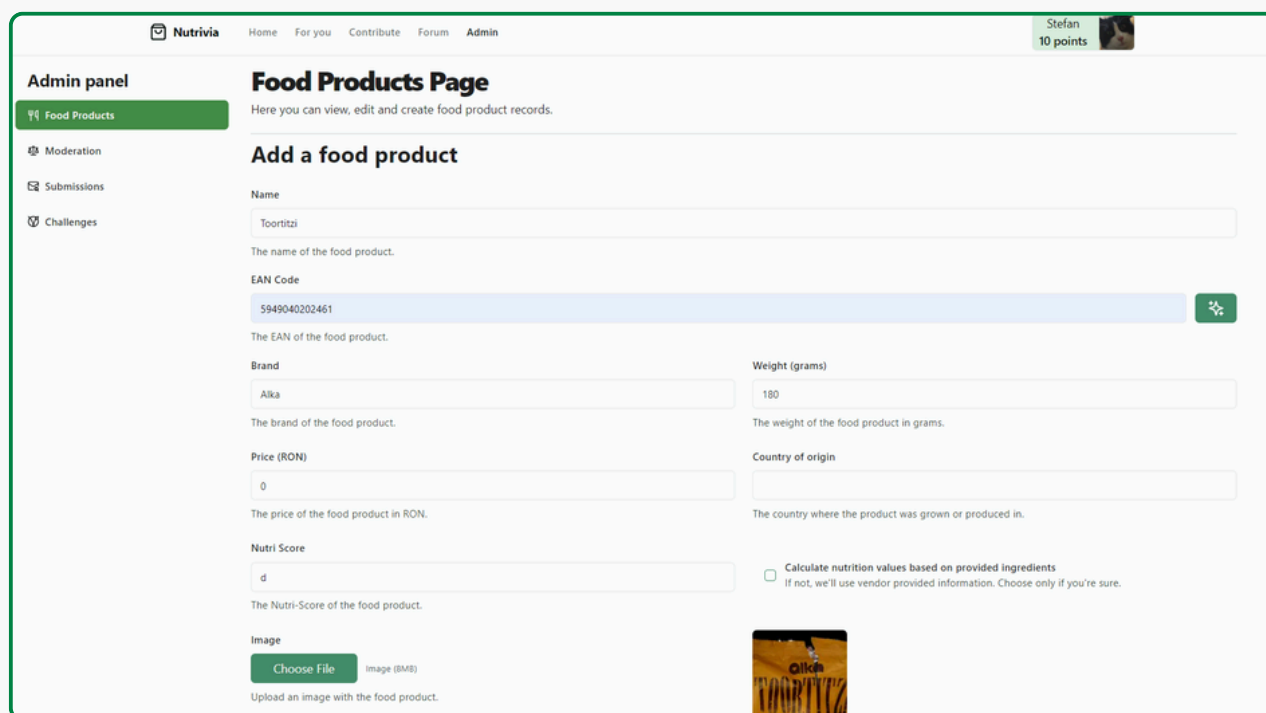


## 1.4 Forum & gamification features

Forumul oferă utilizatorilor un loc unde să interacționeze, oferind un element de comunitate. Pot fi inițiate provocări de grup, unde utilizatorii pot participa împreună, crescând șansele de a completa provocarea.

## 1.5 Panou administrator

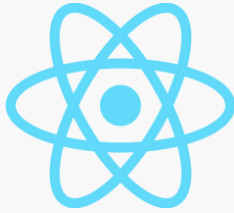
Această pagină este împărțită în mai multe secțiuni pentru gestionarea entităților disponibile (alimente, ingrediente, provocări) și moderarea conținutului (postări, comentarii). Formularele de introducere a unor noi produse/alimente beneficiază de auto-completare inteligentă (pe baza numelui sau a codului EAN) pentru a asigura corectitudinea datelor. În spate, sunt folosite API-urile OpenFoodFacts și USDA.



The screenshot displays the 'Admin panel' of the Nutrivia application. The left sidebar contains navigation links: 'Food Products' (selected), 'Moderation', 'Submissions', and 'Challenges'. The main content area is titled 'Food Products Page' and includes a description: 'Here you can view, edit and create food product records.' Below this is the 'Add a food product' form. The form fields include: 'Name' (Toortitz), 'EAN Code' (5949040202461), 'Brand' (Alka), 'Weight (grams)' (180), 'Price (RON)' (0), 'Country of origin' (empty), 'Nutri Score' (d), and 'Image' (Choose File). A checkbox for 'Calculate nutrition values based on provided ingredients' is also present. The form is designed to collect detailed information about food products for the application.

# Frontend

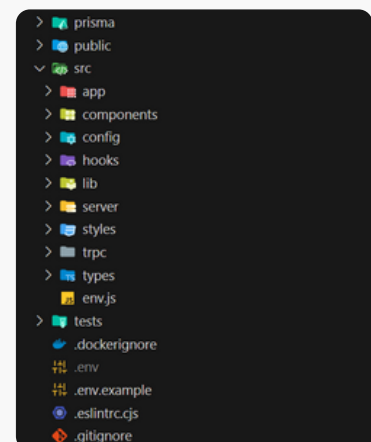
Frontend-ul folosește următoarele tehnologii:



- TypeScript: limbajul de programare folosit, oferă un sistem complet de typing pentru structurile de date ce asigură structura corectă a informației din comunicarea dintre componente și client - server
- ReactJS: librăria pentru UI, asigură utilități pentru a face interfața cât mai interactivă
- Tailwind: o serie de clase CSS utilitare pentru a impune un style system consistent și a unifica codul pentru UI în componente TSX fără fișiere externe CSS
- RadixUI: o colecție de componente funcționale accesibile, fără stiluri preaplicate.
- Redux: librărie pentru gestionarea stării globale.

Alte librării:

- dayjs: utilități pentru lucrul cu timp relativ
- marked: convertire markdown în HTML
- DomPurify: sanitizare
- Vault: componentă pentru drawer
- Sonner: componentă pentru toasts

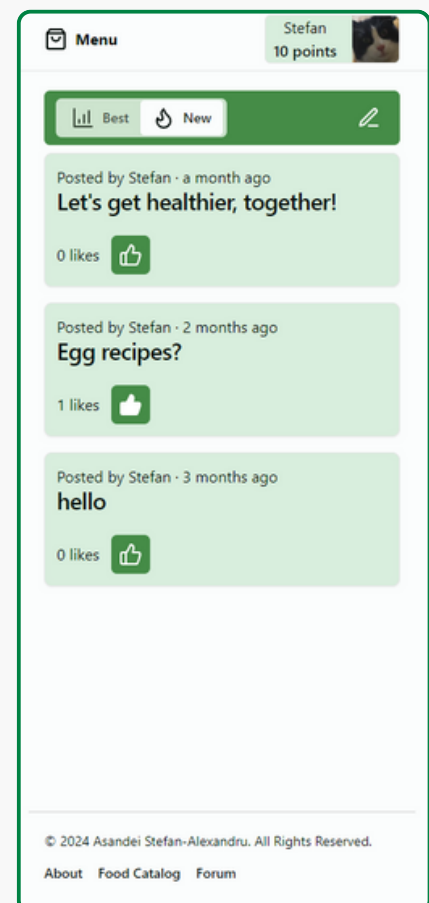
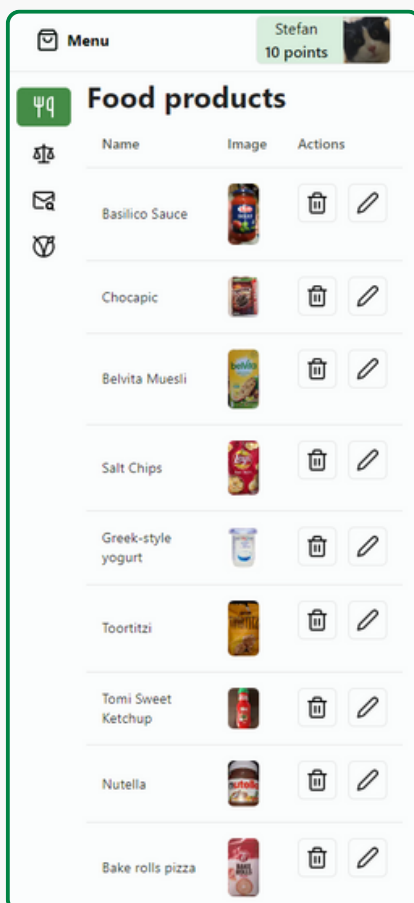




## 2.1.1 Design

Am prioritizat designul pentru a oferi o experiență de utilizare îmbunătățită și o interfață plăcută.

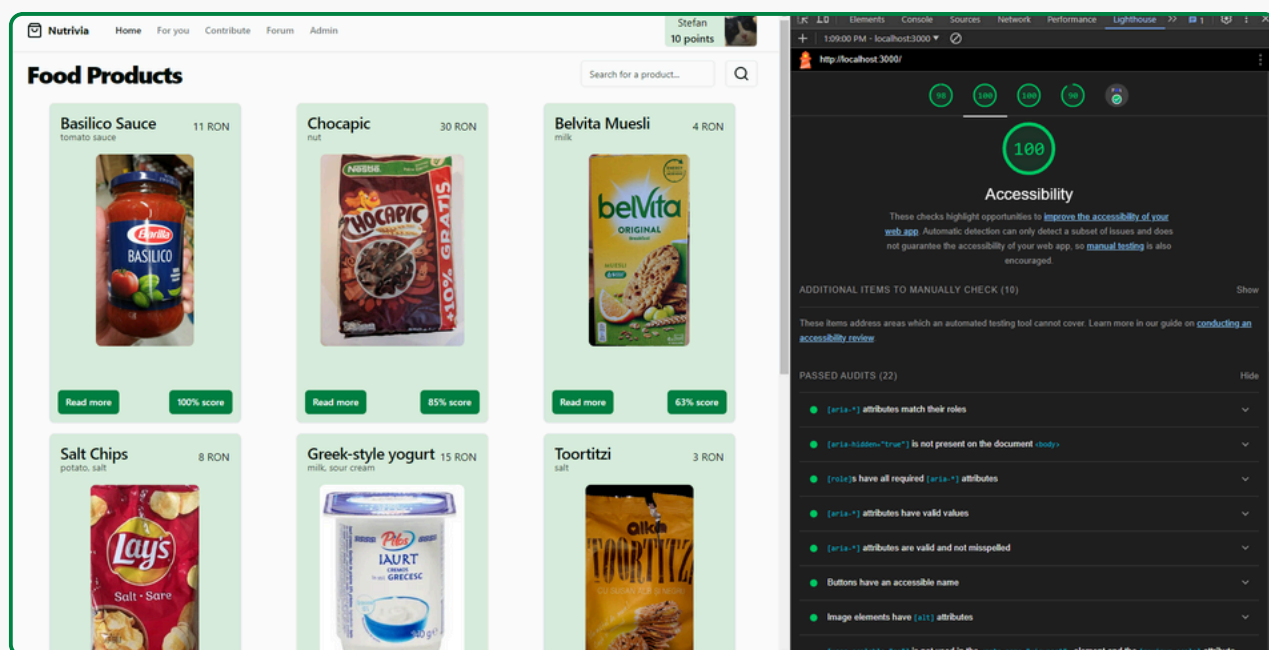
- Unitate și echilibru vizual: Am integrat o paletă de culori și un sistem de tipografie coerente pentru a crea o compoziție echilibrată și o experiență vizuală plăcută.
- Concordanța interfeței: Interfața este intuitivă și ușor de navigat, adaptată nevoilor specifice ale utilizatorilor.
- Ergonomie: Am simplificat navigarea și am integrat căutarea internă și recomandările de conținut pentru a asigura o experiență confortabilă și eficientă pentru utilizatori.



## 2.1.2 Accesibilitate

Pentru a asigura o experiență accesibilă pentru toți utilizatorii, am integrat în proiect biblioteca Radix UI. Această bibliotecă respectă îndrumările practice de autorizare WAI-ARIA și este testată într-o gamă largă de browsere moderne și tehnologii de asistență utilizate frecvent.

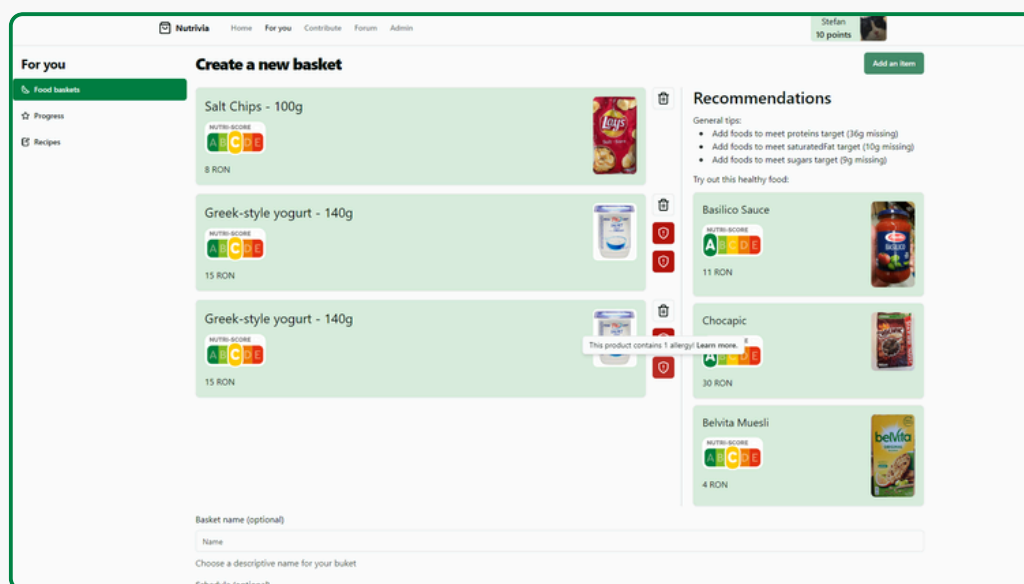
În plus, am luat în considerare contrastul în modurile de lumină și întuneric pentru a asigura o citibilitate optimă și o experiență plăcută pentru toți utilizatorii, indiferent de preferințele lor de afișare.



## 2.1.3 Experiența de utilizare (UX)

Am optimizat experiența de utilizare a aplicației, urmărind următoarele aspecte:

- **Ergonomie:** Am acordat atenție aspectelor de ergonomie, cum ar fi implementarea unei căutări intuitive și a altor funcționalități care facilitează navigarea și interacțiunea cu aplicația.
- **PWA (Aplicație Web Progresivă):** Am implementat funcționalitatea PWA pentru a oferi acces offline (pentru conținutul prezent în cache) și posibilitatea de instalare a aplicației pe dispozitivele utilizatorilor. De asemenea, oferă acces pentru a trimite notificări de sistem.
- **Design Responsiv:** Am dezvoltat o interfață care se adaptează la diferite dispozitive și dimensiuni de ecran, asigurând o experiență uniformă și plăcută pentru utilizatori, indiferent de dispozitivul pe care îl folosesc. (exemplu: un Modal pe desktop va fi un Drawer pe telefoane)

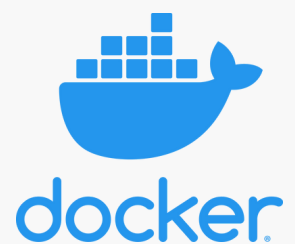


## 2.2.1 Arhitectură

Backend-ul utilizeaza urmatoarele tehnologii:

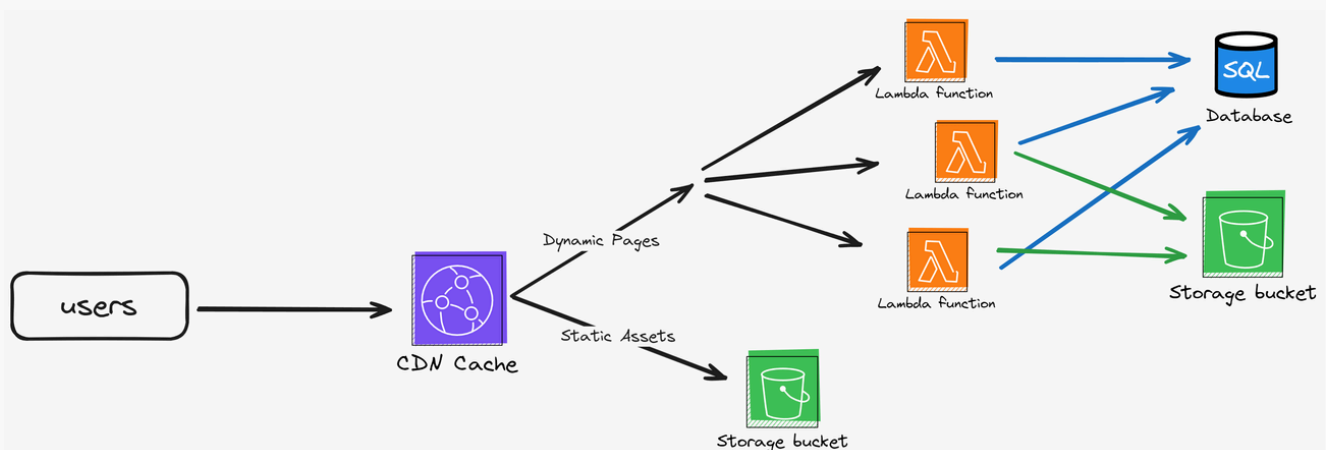
- NextJS
- tRPC
- Prisma
- Ollama
- Mistral 7B
- Docker
- Alpine Linux
- PostgreSQL

# NEXT.js



Backend-ul utilizează o arhitectură serverless pentru scalabilitate și costuri reduse. Fiecare pagină este randată de o funcție Lambda, utilizatorul primind din start conținutul HTML funcțional. Interogările către baza de date sunt efectuate tot de către funcțiile Lambda, ceea ce reduce timpul de așteptare și ascunde procesul pentru utilizatorii rău intenționați.

Pentru cost redus este folosit un serviciu wrapper AWS S3 (UploadThing), ce oferă un free tier. Hosting-ul este făcut pe Vercel. O diagramă cu infrastructura:



## 2.2.1 Baza de date

Pentru a gestiona eficient datele, am optat pentru utilizarea PostgreSQL împreună cu Prisma. Această combinație oferă o soluție robustă și scalabilă pentru stocarea și manipularea datelor. Query-urile repetate sunt stocate în cache, evitând interogarea inutilă. În cadrul schemei bazei de date, pentru a modela legăturile dintre entități au fost folosite tipurile de relații:

- Many to Many: Utilizatori – Alergii, Produse – Ingrediente, etc.
- One to One: Submisie – Produs, etc.
- One to Many: Cos – Produse, Postare – Comentarii

Câteva exemple de interogări, pentru completarea unei provocări și funcțiile necesare afișării unei notificări cu acest lucru:

```
getCompleted: protectedProcedure.query(async ({ ctx }) => {
  return await ctx.db.trackedChallenge.findMany({
    where: { userId: ctx.session.user.id },
    include: { challenge: true },
  });
}),

getQueued: publicProcedure.query(async ({ ctx }) => {
  if (ctx.session == null) return null;

  return await ctx.db.completedChallenge.findFirst({
    where: { userId: ctx.session.user.id },
    include: { challenge: true },
  });
}),

markSeen: protectedProcedure
  .input(z.object({ challengeId: z.string().uuid() }))
  .mutation(async ({ ctx, input }) => {
    return await ctx.db.completedChallenge.deleteMany({
      where: {
        userId: ctx.session.user.id,
        challengesId: input.challengeId,
      },
    });
  }),
```

```
complete: publicProcedure
  .input(
    z.object({
      challengeId: z.string().uuid(),
      userId: z.string().optional(),
    })
  )
  .mutation(async ({ ctx, input }) => {
    const challenge = await ctx.db.challenges.findFirst({
      where: { id: input.challengeId },
      select: { doneBy: true },
    });

    // ...

    // mark the challenge as completed
    await ctx.db.challenges.update({
      where: { id: input.challengeId },
      data: { doneBy: { connect: { id: userId } } },
    });

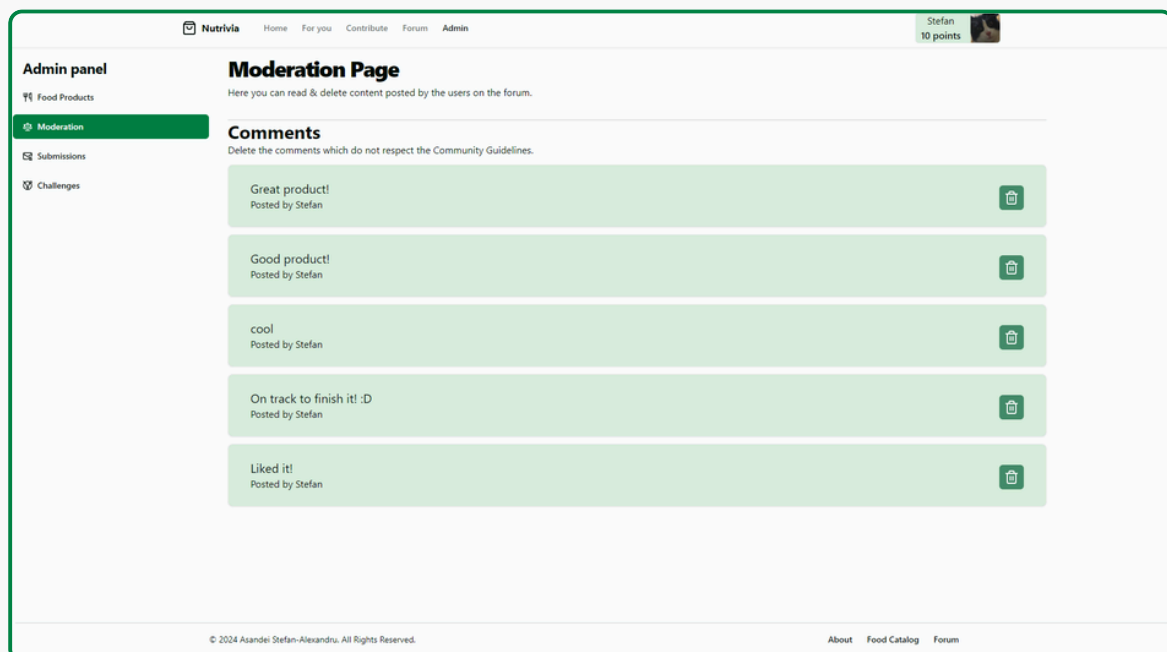
    // add it to the completed challenges "queue"
    return await ctx.db.completedChallenge.create({
      data: {
        challengesId: input.challengeId,
        userId: userId!,
      },
    });
  }),
```



## 3.1 Securitate

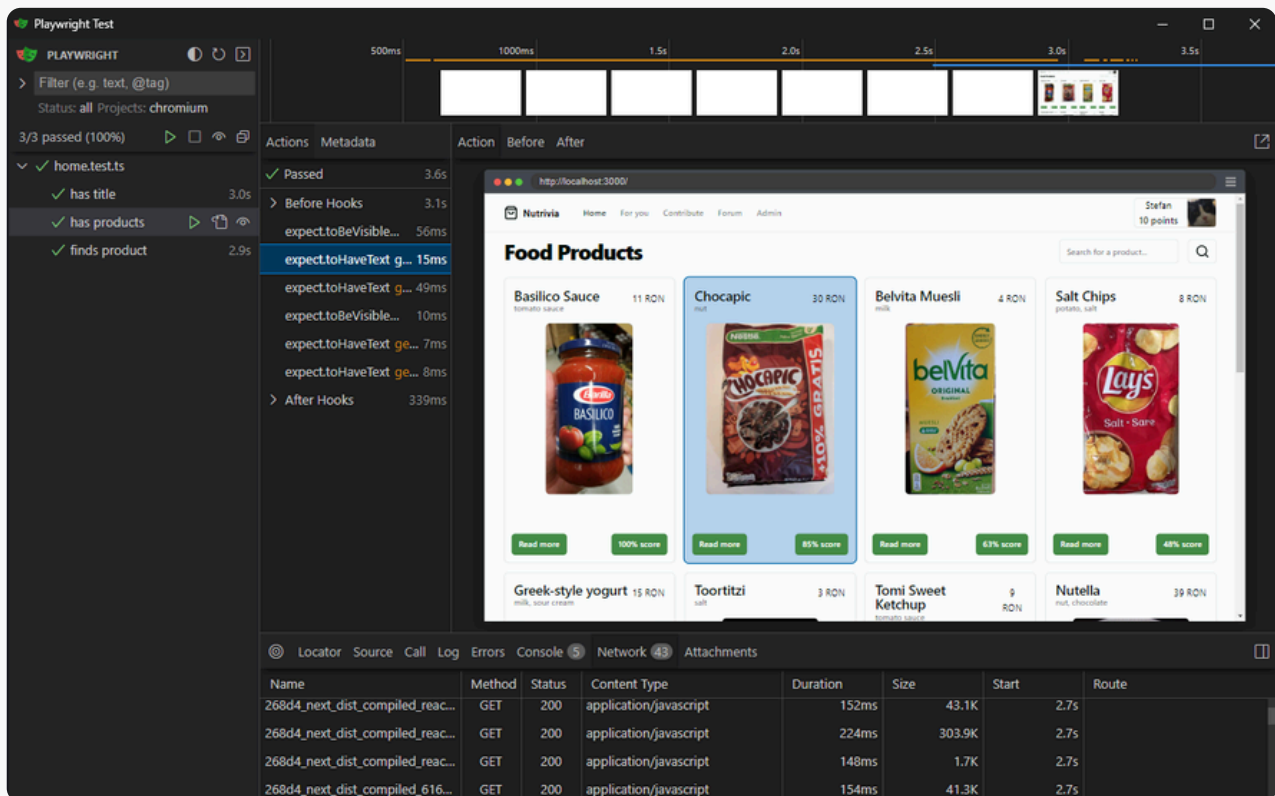
Au fost luate în considerare următoarele aspecte:

- folosirea protocolului HTTPS
- validarea datelor introduse de utilizator
- protecție XSS: utilizatorii pot scrie cod Markdown urmând să fie convertit în HTML (pe forum sau la rețete); înainte de randare, codul va fi sinitizat cu biblioteca DomPurify
- OAuth: autentificarea pe platformă se face exclusiv prin intermediul OAuth 2.0
- autorizație: resursele și paginile pot fi accesate doar de utilizatorii cu permisiunile necesare
- protecție CSRF



## 3.2 Testare

Este folosită biblioteca Playwright pentru testarea de tip end-to-end. Pentru fiecare pagină sunt scrise o serie de teste, astfel asigurând funcționalitatea website-ului. De asemenea, o acțiune GitHub va rula testele în urma unui nou commit.



## 3.3 AI Generativ

Pagina de rețete oferă funcționalitatea de a genera o rețetă, conform preferințelor utilizatorului și alergiilor sale. Pentru acest lucru este folosit un LLM, Mistral 7B. Acesta este operat de un server Ollama, care oferă și o interfață compatibilă cu API-ul OpenAI. Token-urile sunt trimise către client printr-un răspuns de tip streaming, în format JSON, pentru a asigura structura corectă a datelor.





## 3.4 Sistemul de provocări

Provocările au scopul de a motiva utilizatorii să aibă o alimentație sănătoasă. În urma completării, aceștia vor primi puncte, folosite ulterior pentru reduceri. Provocările terminate de un utilizator sunt publice în profilul său, ceea ce poate determina încă o sursă de motivare pentru ceilalți. O provocare poate consta în alegerea unui tip de alimente pentru o masă, prepararea unei rețete, sau alte activități similare. Pentru a realiza acest lucru, am conceput următorul sistem.

Fiecare acțiune de pe website, făcută de utilizator, în urma terminării cu succes a acesteia, poate trimite un eveniment de completare a unei provocări. Odată ce este primit evenimentul, se verifică dacă utilizatorul respectiv are activată în progres provocarea necesară și dacă nu a terminat-o în trecut. Eventual, provocarea va fi marcată ca terminată și ID-ul ei împreună cu ID-ul utilizatorilor vor fi adăugate într-o "coadă". La încărcarea fiecărei pagini, se va verifica dacă există un element în vârful cozii, iar în caz afirmativ va fi afișat un mesaj pop-up pentru a transmite terminarea provocării și acordarea de puncte. După aceea, provocarea va fi scoasă din coadă.

```
import { api } from "@trpc/server";
import RewardPopup from "../reward-popup";

// mounted at the top of the app
export default async function RewardProvider() {
  const challenge = await api.challenge.getQueued.query();

  if (challenge == null) return <></>;

  return <RewardPopup challenge={challenge.challenge} />;
}
```





## 3.5 Serializare

Salvarea și persistarea datelor este gândită pentru două situații:

- accesul imediat (navigare de pe o pagină pe alta, revenirea pe website la scurt timp pe același device)
- accesul ulterior (accesarea website-ului oricând de pe orice device)

Pentru accesul direct este folosit local storage (format JSON) și cookie-urile (pentru sesiunea curentă, utilizatorul autentificat). Datele din sesiune sunt criptate cu un secret unic.

The screenshot shows the Chrome DevTools Storage Inspector. The left sidebar lists the application's storage areas: Manifest, Service workers, Storage, Local storage, Session storage, IndexedDB, Cookies, and Cache storage. The main pane displays the data for the origin `http://localhost:3000`. It shows a table of storage items with keys and values. The `recipe-cache` item is expanded, revealing a JSON object containing recipe data.

Key	Value
theme	light
nextauth.message	{ "event": "session", "data": { "trigger": "signout", "timestamp": 1715591350 } }
recipe-cache	{ "recipes": [ { "prompt": "Tasty and with plenty carbohydrates", "content": { "title": "Two-Person Spaghetti Aglio e Olio with Garlic Bread", "ingredients": [ "2 servings of spaghetti pasta (16 oz)", "1/3 cup extra virgin olive oil", "6 garlic cloves, thinly sliced", "1 red pepper flake (optional)", "Salt to taste", "Freshly ground black pepper to taste", "2 tbsp chopped fresh parsley or basil", "2 tbsp grated Parmesan cheese", "2 loaves of Italian bread", "4 tbsp butter", "1 clove garlic, minced", "1/4 cup grated Parmesan cheese (for garlic bread)" ], "instructions": [ "**Spaghetti Aglio e Olio:**", "1. Cook pasta according to package directions in s" ], "prompt": "Tasty and with plenty carbohydrates" } } ] } }



## 4. Bibliografie

### 1. Documentațiile pentru tehnologiile folosite:

- <https://www.prisma.io/docs>
- <https://tailwindcss.com/docs>
- <https://ui.shadcn.com/>
- <https://playwright.dev/docs/intro>

### 2. API-uri utilizate:

- <https://world.openfoodfacts.org/>
- <https://www.usda.gov/>
- <https://docs.uploadthing.com/>
- <https://ollama.com/>
- <https://mistral.ai/>

Codul sursă este open-source și folosește licența GNU General Public License v3. Acesta este disponibil pe GitHub:

- <https://github.com/stefanasandei/nutrivia>





