Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 2

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
|  |  |  |
| Link ka Github-u |  | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc154095131)

[2 Implementacija 4](#_Toc154095132)

[2.1 Model 4](#_Toc154095133)

[2.2 Migracije 4](#_Toc154095134)

[2.3 Kontroleri 6](#_Toc154095135)

[2.4 Api rute 7](#_Toc154095136)

[3 REST API 8](#_Toc154095137)

[4 Korisničko uputstvo 17](#_Toc154095138)

# Korisnički zahtev

Aplikacija pruža raznovrsne funkcionalnosti za upravljanje ishranom i planovima obroka, fokusirajući se na pružanje personalizovanih dijetetskih planova i praćenje nutritivnih informacija o hrani. Korisnici imaju mogućnost da se registruju i prijave u sistem, gde mogu da koriste funkcije kao što su kreiranje individualizovanih planova ishrane pomoću OpenAI GPT modela. Ova funkcionalnost je dostupna kroz posebnu API rutu koja omogućava korisnicima da unesu svoje specifične potrebe i preferencije, na osnovu kojih se generiše prilagođeni plan ishrane.

Pored toga, aplikacija sadrži detaljno strukturiranu bazu podataka o hrani i obrocima, omogućavajući korisnicima da pregledaju, dodaju, ažuriraju i brišu informacije o različitim stavkama hrane i obrocima. Postoji i funkcionalnost za detaljnu pretragu hrane, koja korisnicima omogućava da pretražuju bazu podataka na osnovu različitih kriterijuma kao što su sastojci, kalorijske vrednosti i pripadnost određenim obrocima. Kroz ove funkcije, aplikacija pruža korisne alate za osobe koje žele da prate svoju ishranu, bilo za lične ciljeve u vezi sa zdravljem i fitnesom ili za profesionalne potrebe kao što su nutricionisti i dijetetičari.

# Implementacija

## Model

Veze medju modelima su prikazane dijagramom ispod

Slika na kojoj se nalazi dijagram, crtež, skeč

Opis je automatski generisan

## Migracije

1. **2024\_01\_30\_222451\_add\_columns\_to\_users\_table.php**: Ova migracija dodaje nove kolone u tabelu **users**. Specifično, dodaje se kolona **age** kao celobrojna vrednost koja može biti null, kolona **gender** kao enumeracija sa vrednostima 'male', 'female', 'other' koja takođe može biti null, kolone **weight** i **height** kao decimalne vrednosti koje mogu biti null, kolona **activity** kao enumeracija sa vrednostima koje predstavljaju različite nivoe aktivnosti koje mogu biti null, i kolona **dietary\_restrictions** kao tekst koja može biti null. Dodatno, dodaje se i kolona **notes** kao tekst.

Slika na kojoj se nalazi tekst, snimak ekrana

Opis je automatski generisan

1. **2024\_01\_30\_222535\_create\_diet\_plans\_table.php**: Ova migracija je odgovorna za kreiranje nove tabele nazvane **diet\_plans**. Tabela uključuje nekoliko kolona kao što su auto-inkrement ID, **user\_id** kao neoznačeni veliki celi broj, **start\_date** i **end\_date** kao vremenski pečati, kolona **total\_calories** kao celobrojni broj, i druge atribute vezane za plan ishrane.

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Opis je automatski generisan

1. **2024\_01\_30\_222607\_add\_foreign\_key\_to\_diet\_plans.php**: migracija dodaje strani ključ tabeli **diet\_plans**.

Slika na kojoj se nalazi tekst, snimak ekrana, Font, linija

Opis je automatski generisan

1. **2024\_01\_30\_222632\_create\_meals\_table.php**: kreira se tabela **meals**, koja bi sadržala detalje o pojedinačnim obrocima koji mogu biti deo plana ishrane.

Slika na kojoj se nalazi tekst, snimak ekrana, ekran, softver

Opis je automatski generisan

1. **2024\_01\_30\_222730\_create\_food\_items\_table.php**: migracija kreira tabelu **food\_items**, koja bi sadržala detalje o namirnicama koje se mogu uključiti u obroke, kao što su ime, količina, jedinica i kalorije po jedinici.

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Opis je automatski generisan

1. **2024\_01\_30\_222820\_rename\_activity\_column\_in\_users\_table.php**: Ova migracija menja ime kolone u tabeli **users**. Konkretno, preimenuje se kolona **activity** u **activity\_level**.

Slika na kojoj se nalazi tekst, snimak ekrana, Font, linija

Opis je automatski generisan

## Kontroleri

Metoda kreirajPlanGPT u DietPlanController-u koristi se za kreiranje personalizovanog plana ishrane koristeći OpenAI servis. Prvo se izvršava validacija zahteva da bi se osiguralo da su potrebni podaci, poput user\_id, dostavljeni i ispravni. Ako validacija ne prođe, vraća se JSON odgovor sa greškama. Ukoliko su podaci validni, koristi se servis OpenAIService da se generiše plan ishrane na osnovu korisničkog unosa, koji uključuje ID korisnika, period za koji treba kreirati plan, dijetetske preference i kalorijski cilj. Rezultat koji servis vrati se zatim šalje korisniku kao JSON odgovor.

Metoda index je standardna metoda kontrolera koja vraća listu svih planova ishrane iz baze podataka. Ova metoda dohvaća sve instance DietPlan modela, zatim ih transformiše u resurse pomoću DietPlanResource i vraća ih kao kolekciju u JSON formatu. Ovo omogućava korisnicima da pregledaju sve dostupne planove ishrane.

Metoda show se koristi za prikazivanje specifičnog plana ishrane. Ona traži plan ishrane sa određenim ID-om pomoću Eloquent metode findOrFail, koja će baciti ModelNotFoundException ako plan sa tim ID-om ne postoji. Ako plan postoji, on se vraća kao instanca DietPlanResource, koja pruža konzistentan i čist format odgovora.

Metoda store omogućava kreiranje novog plana ishrane i njegovo čuvanje u bazi podataka. Prvo se vrši validacija primljenih podataka. Ako podaci prođu validaciju, kreiraju se i čuvaju u bazi podataka koristeći masovno dodeljivanje (create metoda Eloquent ORM-a). Kreirani plan ishrane se zatim vraća kao DietPlanResource.

Metoda update koristi se za ažuriranje postojećeg plana ishrane. Slično kao kod metode store, prvo se izvršava validacija zahteva. Nakon toga, metoda dohvaća postojeći plan ishrane sa prosleđenim ID-jem i ažurira ga sa novim podacima iz zahteva. Ažurirani plan ishrane se vraća kao DietPlanResource.

Metoda destroy se koristi za brisanje plana ishrane iz baze podataka. Ona pronalazi plan ishrane sa određenim ID-jem i briše ga. Nakon uspešnog brisanja, vraća se JSON odgovor sa porukom da je plan ishrane obrisan.

<?php

namespace App\Http\Controllers;

use App\Models\DietPlan;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

use App\Http\Resources\DietPlanResource;

use App\Services\OpenAIService;

class DietPlanController extends Controller

{

    protected $openAIService;

    public function \_\_construct(OpenAIService $openAIService)

    {

        $this->openAIService = $openAIService;

    }

    public function kreirajPlanGPT(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'user\_id' => 'required|integer',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $userInput = [

            'user\_id' => $request->input('user\_id'),

            'period' => $request->input('period'),

            'preferences' => $request->input('preferences'),

            'calories' => $request->input('total\_calories'),

        ];

        $generatedPlan = $this->openAIService->generateDietPlan($userInput);

        return response()->json($generatedPlan, 200);

    }

    public function index()

    {

        $dietPlans = DietPlan::all();

        return DietPlanResource::collection($dietPlans);

    }

    public function show($id)

    {

        $dietPlan = DietPlan::findOrFail($id);

        return new DietPlanResource($dietPlan);

    }

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'user\_id' => 'required|integer',

            'start\_date' => 'required|date',

            'end\_date' => 'required|date',

            'total\_calories' => 'required|numeric',

            'goal' => 'required|string',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $dietPlan = DietPlan::create($request->all());

        return new DietPlanResource($dietPlan);

    }

    public function update(Request $request, $id)

    {

        $validator = Validator::make($request->all(), [

            'user\_id' => 'required|integer',

            'start\_date' => 'required|date',

            'end\_date' => 'required|date',

            'total\_calories' => 'required|numeric',

            'goal' => 'required|string',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $dietPlan = DietPlan::findOrFail($id);

        $dietPlan->update($request->all());

        return new DietPlanResource($dietPlan);

    }

    public function destroy($id)

    {

        $dietPlan = DietPlan::findOrFail($id);

        $dietPlan->delete();

        return response()->json(['message' => 'Diet plan deleted'], 200);

    }

}

Metode u FoodItemController služe za upravljanje resursom FoodItem :

Metoda index se koristi za dohvatanje i prikaz svih FoodItem instanci iz baze podataka. Ona koristi Eloquent metod all() za dohvatanje svih zapisa iz FoodItem tabele, a zatim ih konvertuje u FoodItemResource kako bi ih vratila u standardizovanom JSON formatu. Ovo omogućava pregled svih dostupnih stavki hrane u sistemu.

Metoda show se koristi za dohvatanje i prikaz jedne specifične instance FoodItem na osnovu njenog ID-a. Ako ne postoji zapis sa datim ID-jem, Laravel baca izuzetak ModelNotFoundException. Ako zapis postoji, on se vraća kao FoodItemResource, omogućavajući detaljan prikaz specifične stavke hrane.

Metoda store se koristi za kreiranje nove instance FoodItem. Prvo se vrši validacija podataka iz zahteva. Ako podaci zadovoljavaju zadate uslove, nova instanca FoodItem se kreira i čuva u bazi podataka koristeći primljene podatke. Kreirani zapis se zatim vraća kao FoodItemResource.

Metoda update ažurira postojeći FoodItem na osnovu njegovog ID-a. Nakon što se validiraju podaci iz zahteva, metoda traži odgovarajuću instancu u bazi podataka. Ako instanca postoji, ona se ažurira sa novim podacima iz zahteva i vraća kao ažurirani FoodItemResource.

Metoda destroy se koristi za brisanje instance FoodItem na osnovu njenog ID-a. Pronalazi se odgovarajuća instanca i briše iz baze podataka. Nakon uspešnog brisanja, vraća se JSON odgovor sa porukom o uspešnom brisanju.

Metoda search omogućava pretragu FoodItem instanci na osnovu različitih kriterijuma kao što su meal\_id, name, quantity, unit, i calories\_per\_unit. Metoda prvo validira primljene podatke. Zatim, na osnovu prisutnosti i vrednosti parametara iz zahteva, formira se upit za pretragu u bazi podataka. Rezultati se vraćaju kao kolekcija FoodItemResource.

<?php

namespace App\Http\Controllers;

use App\Models\FoodItem;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

use App\Http\Resources\FoodItemResource;

class FoodItemController extends Controller

{

    public function index()

    {

        $foodItems = FoodItem::all();

        return FoodItemResource::collection($foodItems);

    }

    public function show($id)

    {

        $foodItem = FoodItem::findOrFail($id);

        return new FoodItemResource($foodItem);

    }

    public function store(Request $request)

    {

        $validator = Validator::make($request->all(), [

            'meal\_id' => 'required|integer',

            'name' => 'required|string',

            'quantity' => 'required|numeric',

            'unit' => 'required|string',

            'calories\_per\_unit' => 'required|numeric',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $foodItem = FoodItem::create($request->all());

        return new FoodItemResource($foodItem);

    }

    public function update(Request $request, $id)

    {

        $validator = Validator::make($request->all(), [

            'meal\_id' => 'required|integer',

            'name' => 'required|string',

            'quantity' => 'required|numeric',

            'unit' => 'required|string',

            'calories\_per\_unit' => 'required|numeric',

        ]);

        if ($validator->fails()) {

            return response()->json(['errors' => $validator->errors()], 400);

        }

        $foodItem = FoodItem::findOrFail($id);

        $foodItem->update($request->all());

        return new FoodItemResource($foodItem);

    }

    public function destroy($id)

    {

        $foodItem = FoodItem::findOrFail($id);

        $foodItem->delete();

        return response()->json(['message' => 'Food item deleted'], 200);

    }

    public function search(Request $request)

{

    $validator = Validator::make($request->all(), [

        'meal\_id' => 'nullable|integer',

        'name' => 'nullable|string',

        'quantity' => 'nullable|numeric',

        'unit' => 'nullable|string',

        'calories\_per\_unit' => 'nullable|numeric',

    ]);

    if ($validator->fails()) {

        return response()->json(['errors' => $validator->errors()], 400);

    }

    $foodItems = FoodItem::query();

    if ($request->has('meal\_id')) {

        $foodItems->where('meal\_id', $request->meal\_id);

    }

    if ($request->has('name')) {

        $foodItems->where('name', 'like', '%' . $request->name . '%');

    }

    if ($request->has('quantity')) {

        $foodItems->where('quantity', $request->quantity);

    }

    if ($request->has('unit')) {

        $foodItems->where('unit', 'like', '%' . $request->unit . '%');

    }

    if ($request->has('calories\_per\_unit')) {

        $foodItems->where('calories\_per\_unit', $request->calories\_per\_unit);

    }

    $foodItems = $foodItems->get();

    return FoodItemResource::collection($foodItems);

}

}

## Api rute

1. **Registracija Korisnika (/register)**:
   * **POST /register**: Ova ruta omogućava novim korisnicima da se registruju u aplikaciji. Kada se pošalje zahtev sa odgovarajućim korisničkim podacima, ruta obrađuje registraciju koristeći **register** metodu iz **AuthController**-a.
2. **Prijava Korisnika (/login)**:
   * **POST /login**: Ova ruta omogućava korisnicima da se prijave u aplikaciju. Korisnici šalju svoje akreditive, a ruta koristi **login** metodu iz **AuthController**-a za autentifikaciju i vraća token za autorizaciju ili grešku ako su podaci neispravni.
3. **Kreiranje Plana Ishrane (/dietPlans/kreirajPlanGPT)**:
   * **POST /dietPlans/kreirajPlanGPT**: Ova ruta omogućava korisnicima da kreiraju plan ishrane pomoću OpenAI GPT modela. Korisnici šalju specifične zahteve, a ruta koristi **kreirajPlanGPT** metodu iz **DietPlanController**-a za generisanje plana.
4. **Pretraga Stavki Hrane (/foodItems/search)**:
   * **GET /foodItems/search**: Ruta omogućava korisnicima da pretražuju stavke hrane na osnovu različitih parametara. Koristi **search** metodu iz **FoodItemController**-a za filtriranje i vraćanje rezultata pretrage.
5. **Resurs Stavki Hrane (foodItems)**:
   * **Route::resource('foodItems', FoodItemController::class)**: Ova ruta registruje više ruta za standardne CRUD operacije (Create, Read, Update, Delete) vezane za stavke hrane. Koristi standardne metode definisane u **FoodItemController**-u kao što su **index**, **show**, **store**, **update**, i **destroy**.
6. **Resurs Planova Ishrane (dietPlans)**:
   * **Route::resource('dietPlans', DietPlanController::class)**: Slično kao i za **foodItems**, ova ruta definiše standardne CRUD operacije za planove ishrane koristeći metode iz **DietPlanController**-a.
7. **Resurs Obroka (meals)**:
   * **Route::resource('meals', MealController::class)->except(['create', 'edit'])**: Ova ruta definiše CRUD operacije za obroke, ali isključuje rute za **create** i **edit** jer te akcije obično ne zahtevaju poseban API endpoint (mogu se obaviti preko **store** i **update** metoda).
8. **Odjava Korisnika (/logout unutar 'auth' middleware grupe)**:
   * **Route::post('/logout', [AuthController::class, 'logout'])**: Ova ruta omogućava autentifikovanim korisnicima da se odjave. Ruta je zaštićena **auth** middleware-om, što znači da je dostupna samo autentifikovanim korisnicima.

# REST API

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje plana ishrane sa GPTijem |
| HTTP metoda | POST |
| URL | /api/dietPlans/kreirajPlanGPT |
| URL parametri | period=7&user\_id=1&preferences=vegetarian&total\_calories=1800 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": "chatcmpl-8n8wh1sGtWhzoZzdbStPjC54CsHh9",      "object": "chat.completion",      "created": 1706723151,      "model": "gpt-3.5-turbo-0613",      "choices": [          {              "index": 0,              "message": {                  "role": "assistant",                  "content": "Day 1:\nBreakfast: \n- Scrambled tofu with vegetables (200 calories)\n- Whole wheat toast with avocado (150 calories)\n\nLunch: \n- Chickpea salad with mixed greens, tomatoes, cucumber, and lemon-tahini dressing (400 calories)\n\nDinner: \n- Vegetable stir-fry with tofu and brown rice (450 calories)\n\nSnacks: \n- Apple slices with almond butter (150 calories)\n- Greek yogurt with berries (250 calories)\n\nTotal calories: 1600\n\nDay 2:\nBreakfast: \n- Overnight oats with almond milk, chia seeds, and mixed berries (300 calories)\n\nLunch: \n- Quinoa salad with roasted vegetables and a lemon-herb dressing (400 calories)\n\nDinner: \n- Lentil curry with brown rice (450 calories)\n\nSnacks: \n- Carrot sticks with hummus (150 calories)\n- Trail mix with nuts and dried fruit (200 calories)\n\nTotal calories: 1500\n\nDay 3:\nBreakfast: \n- Spinach and mushroom omelette with whole wheat toast (300 calories)\n\nLunch: \n- Black bean and corn salad with avocado and lime dressing (400 calories)\n\nDinner: \n- Vegetable lasagna with a side salad (450 calories)\n\nSnacks: \n- Greek yogurt with honey and nuts (200 calories)\n- Roasted chickpeas (150 calories)\n\nTotal calories: 1500\n\nDay 4:\nBreakfast: \n- Vegan protein smoothie with banana, spinach, almond milk, and protein powder (300 calories)\n\nLunch: \n- Quinoa-stuffed bell peppers with a side of mixed greens (400 calories)\n\nDinner: \n- Sweet potato and black bean enchiladas with salsa (450 calories)\n\nSnacks: \n- Celery sticks with peanut butter (150 calories)\n- Air-popped popcorn (200 calories)\n\nTotal calories: 1500\n\nDay 5:\nBreakfast: \n- Veggie breakfast burrito with scrambled tofu, peppers, onions, and salsa (300 calories)\n\nLunch: \n- Greek salad with feta cheese, olives, tomatoes, cucumbers, and a lemon-olive oil dressing (400 calories)\n\nDinner: \n- Chickpea and vegetable curry with basmati rice (450 calories)\n\nSnacks: \n- Banana with almond butter (150 calories)\n- Roasted edamame (200 calories)\n\nTotal calories: 1500\n\nDay 6:\nBreakfast: \n- Avocado toast with cherry tomatoes and feta cheese (300 calories)\n\nLunch: \n- Spinach and feta quiche with a side salad (400 calories)\n\nDinner: \n- Vegetable stir-fry with tofu and quinoa (450 calories)\n\nSnacks: \n- Mixed fruit salad (150 calories)\n- Veggie sticks with tzatziki sauce (200 calories)\n\nTotal calories: 1500\n\nDay 7:\nBreakfast: \n- Berry and spinach smoothie bowl with granola and almond slices (300 calories)\n\nLunch: \n- Caprese salad with fresh mozzarella, tomatoes, basil, and balsamic glaze (400 calories)\n\nDinner: \n- Lentil and vegetable soup with whole grain bread (450 calories)\n\nSnacks: \n- Greek yogurt with granola and honey (200 calories)\n- Roasted pumpkin seeds (150 calories)\n\nTotal calories: 1500\n\nEnd of diet plan."              },              "logprobs": **null**,              "finish\_reason": "stop"          }      ],      "usage": {          "prompt\_tokens": 91,          "completion\_tokens": 702,          "total\_tokens": 793      },      "system\_fingerprint": **null**  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih food items-a |
| HTTP metoda | GET |
| URL | /api/foodItems |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {     {      "data": [          {              "id": 1,              "meal\_id": 1,              "name": "voluptatem",              "quantity": "179.00",              "unit": "komad",              "calories\_per\_unit": "15.00"          },          {              "id": 2,              "meal\_id": 1,              "name": "qui",              "quantity": "238.00",              "unit": "komad",              "calories\_per\_unit": "18.00"          },  .. |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje food items-a |
| HTTP metoda | POST |
| URL | /api/foodItems |
| URL parametri | meal\_id=1&name=jaje&quantity=1&unit=komad&calories\_per\_unit=30 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "data": {          "id": 21,          "meal\_id": "1",          "name": "jaje",          "quantity": "1",          "unit": "komad",          "calories\_per\_unit": "30"      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | IZmena food itemsa |
| HTTP metoda | PUT |
| URL | /api/ foodItems |
| URL parametri | 21?meal\_id=1&name=jaje&quantity=1&unit=komad&calories\_per\_unit=30 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "data": {          "id": 21,          "meal\_id": "1",          "name": "jaje",          "quantity": "1",          "unit": "komad",          "calories\_per\_unit": "30"      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje food items-a |
| HTTP metoda | DELETE |
| URL | /api/ foodItems |
| URL parametri | 21 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Food item deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih obroka |
| HTTP metoda | GET |
| URL | /api/ meals |
| URL parametri | nema |
| HTTP body parametri | nema |
| Format HTTP body parametara | JSON |
| Izlazni parametri | "data": [          {              "id": 1,              "diet\_plan": {                  "id": 6,                  "user": {                      "id": 11,                      "name": "Maria Gusikowski IV",                      "email": "madelynn.bailey@example.org",                      "age": 33,                      "gender": "female",                      "weight": 142,                      "height": 205,                      "activity\_level": "sedentary",                      "dietary\_restrictions": **null**                  },                  "start\_date": "2008-11-04T00:00:00.000000Z",                  "end\_date": "1989-11-26T00:00:00.000000Z",                  "total\_calories": 2998,                  "goal": "dobijanje mišićne mase"              },              "name": "atque",              "description": "Alias quod ducimus consequatur repudiandae consequatur.",              "time\_of\_day": "uzina",              "calories": "573.00",              "proteins": "20.00",              "carbs": "87.00",              "fats": "25.00",              "food\_items": [                  {                      "id": 1,                      "meal\_id": 1,                      "name": "voluptatem",                      "quantity": "179.00",                      "unit": "komad",                      "calories\_per\_unit": "15.00"                  },                  {                      "id": 2,                      "meal\_id": 1,                      "name": "qui",                      "quantity": "238.00",                      "unit": "komad",                      "calories\_per\_unit": "18.00"                  },                  {                      "id": 3,                      "meal\_id": 1,                      "name": "tempora",                      "quantity": "290.00",                      "unit": "g",                      "calories\_per\_unit": "43.00"                  },                  {                      "id": 6,                      "meal\_id": 1,                      "name": "dolorem",                      "quantity": "364.00",                      "unit": "ml",                      "calories\_per\_unit": "85.00"                  },                  {                      "id": 9,                      "meal\_id": 1,                      "name": "sint",                      "quantity": "303.00",                      "unit": "ml",                      "calories\_per\_unit": "26.00"                  },                  {                      "id": 14,                      "meal\_id": 1,                      "name": "minus",                      "quantity": "283.00",                      "unit": "ml",                      "calories\_per\_unit": "48.00"                  },                  {                      "id": 16,                      "meal\_id": 1,                      "name": "incidunt",                      "quantity": "316.00",                      "unit": "komad",                      "calories\_per\_unit": "100.00"                  }, |
| Format izlaznih parametara | Application/json |
|  |  |
| Opis funkcije | Kreiranje obroka |
| HTTP metoda | POST |
| URL | /api/ meals |
| URL parametri | diet\_plan\_id=1&name=obrok&time\_of\_day=dorucak&calories=600&proteins=50&carbs=300&fats=20 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "data": {          "id": 11,          "diet\_plan": {              "id": 1,              "user": {                  "id": 6,                  "name": "Jon Blanda MD",                  "email": "xgleason@example.org",                  "age": 20,                  "gender": "other",                  "weight": 109,                  "height": 155,                  "activity\_level": "sedentary",                  "dietary\_restrictions": "Aut ea et inventore numquam doloribus enim. Ut qui ratione dolorum suscipit inventore voluptas voluptatibus officiis. Atque neque eaque et temporibus qui error ut voluptates."              },              "start\_date": "2017-08-15T00:00:00.000000Z",              "end\_date": "2007-04-06T00:00:00.000000Z",              "total\_calories": 1408,              "goal": "gubitak težine"          },          "name": "obrok",          "description": **null**,          "time\_of\_day": "dorucak",          "calories": "600",          "proteins": "50",          "carbs": "300",          "fats": "20",          "food\_items": []      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Ažuriranje obroka |
| HTTP metoda | PUT |
| URL | /api/ meals |
| URL parametri | 11?diet\_plan\_id=1&name=obrok&time\_of\_day=dorucak&calories=600&proteins=50&carbs=300&fats=20 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "data": {          "id": 11,          "diet\_plan": {              "id": 1,              "user": {                  "id": 6,                  "name": "Jon Blanda MD",                  "email": "xgleason@example.org",                  "age": 20,                  "gender": "other",                  "weight": 109,                  "height": 155,                  "activity\_level": "sedentary",                  "dietary\_restrictions": "Aut ea et inventore numquam doloribus enim. Ut qui ratione dolorum suscipit inventore voluptas voluptatibus officiis. Atque neque eaque et temporibus qui error ut voluptates."              },              "start\_date": "2017-08-15T00:00:00.000000Z",              "end\_date": "2007-04-06T00:00:00.000000Z",              "total\_calories": 1408,              "goal": "gubitak težine"          },          "name": "obrok",          "description": **null**,          "time\_of\_day": "dorucak",          "calories": "600",          "proteins": "50",          "carbs": "300",          "fats": "20",          "food\_items": []      }  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje obroka |
| HTTP metoda | DELETE |
| URL | /api/ meals |
| URL parametri | 11 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Meal successfully deleted"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prijava korisnika |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | email=britney.barrows@example.com&password=password |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "user": {          "id": 13,          "name": "Elody Konopelski PhD",          "email": "britney.barrows@example.com",          "age": 62,          "gender": "other",          "weight": 41,          "height": 193,          "activity\_level": "very\_active",          "dietary\_restrictions": "Repudiandae quisquam explicabo odit est occaecati. Sed delectus similique impedit blanditiis quis. Nisi facilis nemo est voluptatem et."      },      "token": "1|miZycPoIgSZaacyXdwlOXZTEP349MuswLgs9WNybcff3be80"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Odjava korisnika |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | nema |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "You have been successfully logged out"  } |
| Format izlaznih parametara | application/json |

# Korisničko uputstvo

Login

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Kreiranje plana na osnovu OpenAi apija

Slika na kojoj se nalazi tekst, snimak ekrana, broj, Font

Opis je automatski generisan

Prikaz svih food items

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Kreiranje novog food items

Slika na kojoj se nalazi tekst, snimak ekrana, broj, Font

Opis je automatski generisan

Izmena food items

Slika na kojoj se nalazi tekst, snimak ekrana, broj, Font

Opis je automatski generisan

Brisanje food items

Slika na kojoj se nalazi tekst, snimak ekrana, broj, Paralela

Opis je automatski generisan

Prikaz svih obroka

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Kreiranje obroka

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Azuriranje

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Brisanje

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Pretraga food itemsa

Slika na kojoj se nalazi tekst, snimak ekrana, softver, broj

Opis je automatski generisan

Registracija novih korisnika

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Opis je automatski generisan

Registracija novih korisnika

Slika na kojoj se nalazi tekst, snimak ekrana, broj, softver

Opis je automatski generisan

Odjava

Slika na kojoj se nalazi tekst, snimak ekrana, Font

Opis je automatski generisan