



Kauno technologijos universitetas

Informatikos fakultetas

T120B165 Saityno taikomųjų programų projektavimas

Projekto „Trainee“ ataskaita

Ignas Gineitis IFF-2/6

Studentas

Armalis Patrikas

jaun. asist. Kiudys Eligijus

Dėstytojas

Kaunas, 2025-11-28

Turinys

1. SPRENDŽIAMO UŽDAVINIO APRAŠYMAS	3
1.1. Sistemos Paskirtis.....	3
1.2. Funkciniai Reikalavimai	3
2. SISTEMOS ARCHITEKTŪRA	5
2.1. UML Deployment Diagrama	5
2.2. Technologijų Stacks.....	5
3. NAUDOTOJO SĄSAJOS PROJEKTAS	6
3.1. Wireframe'ai ir Realizacijos	6
3.2. Responsive Design Pavyzdžiai.....	15
4. API SPECIFIKACIJA	16
4.1. OpenAPI Dokumentacija	16
4.2. API Endpoint'ai su Pavyzdžiais	16
4.3. Error Response Format	28
5. PROJEKTO IŠVADOS.....	29
5.1. Įgyvendinti Tikslai.....	29
5.2. Techniniai Pasiekimai.....	29
5.3. Iššūkiai ir Sprendimai	29
5.4. Sistemos Privalumai	30
5.5. Tolimesni Patobulinimai (Future Work).....	30
5.6. Išmoktos Pamokos	30
5.7. Projekto Metrika	30
6. PRIEDAI.....	31
6.1. Nuorodos	31
6.2. Naudota Literatūra.....	31

1. SPRENDŽIAMO UŽDAVINIO APRAŠYMAS

1.1. Sistemos Paskirtis

Trainee yra internetinė platforma, skirta mitybos ir sporto planų kūrimui, dalijimuisiui bei naudojimui. Sistema padeda vartotojams:

- **Kurti** asmeninius mitybos planus ir treniruočių programas
- **Dalintis** savo sukurtais planais su bendruomene
- **Stebėti** su sveikata susijusią informaciją (kalorijų skaičiavimas, rekomendacijos)
- **Bendrauti** su kitais vartotojais per komentarus ir reitingavimą
- **Įsitrukti** į sveikos gyvensenos veiklas

Pagrindinė problema: Žmonėms trūksta centralizuotos platformos, kur galėtų rasti patikimus, bendruomenės patvirtintus mitybos ir sporto planus, pritaikytus įvairiems tikslams.

Sprendimas: Trainee platforma su moderuotu turiniu, vartotojų įvertinimais ir socialinėmis funkcijomis, leidžianti lengvai rasti, kurti ir sekti mitybos/sporto planus.

1.2. Funkciniai Reikalavimai

1.2.1. Neregistruotas Naudotojas (Svečias)

ID	Reikalavimas	Prioritetas
FR-G1	Peržiūrėti viešai paskelbtus postus (meal/workout plans)	Aukštas
FR-G2	Peržiūrėti komentarus ir reitingus	Aukštas
FR-G3	Peržiūrėti platformos reprezentacinį puslapį	Vidutinis
FR-G4	Užsiregistrnuoti sistemoje	Aukštas
FR-G5	Prisijungti prie sistemas	Aukštas

1.2.2. Registruotas Naudotojas

ID	Reikalavimas	Prioritetas
FR-U1	Atsijungti nuo sistemas	Aukštas
FR-U2	Kurti naujus postus (mitybos planai / treniruočių sekos)	Aukštas
FR-U3	Pridėti papildomą informaciją (kalorijų skaičiavimas, rekomendacijos)	Vidutinis
FR-U4	Redaguoti savo sukurtus postus	Aukštas
FR-U5	Ištrinti savo postus	Aukštas
FR-U6	Paskelbti postą viešai (laukia admin)	Aukštas

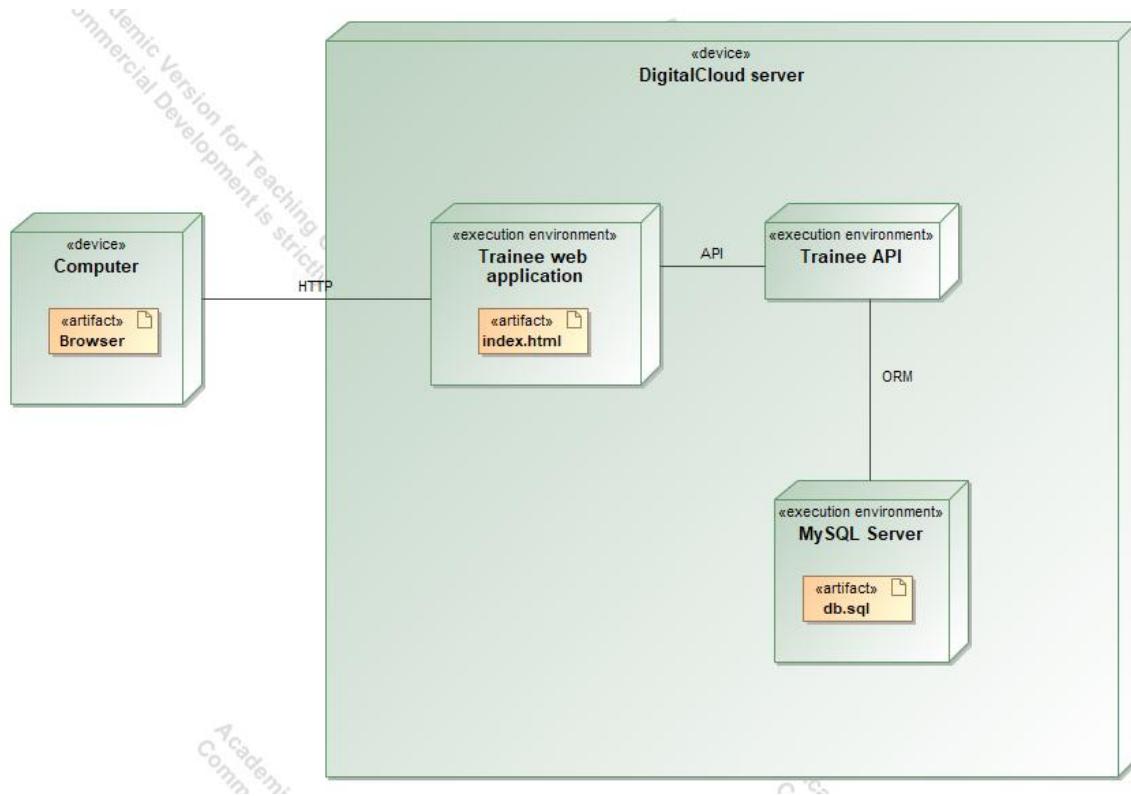
ID	Reikalavimas (patvirtinimo)	Prioritetas
FR-U7	Komentuoti kitų vartotojų postus	Vidutinis
FR-U8	Reitinguoti kitų vartotojų planus (1-5 žvaigždutės)	Vidutinis
FR-U9	Peržiūrėti savo profilio informaciją ir postus	Aukštas
FR-U10	Peržiūrėti tik patvirtintus viešus postus sekocijose	Aukštas
FR-U11	Peržiūrėti visus savo postus (įskaitant laukiančius patvirtinimo)	Vidutinis

1.2.3. Administratorius

ID	Reikalavimas	Prioritetas
FR-A1	Patvirtinti naujų naudotojų registracijas	Aukštas
FR-A2	Patvirtinti/atmesti viešai norimus skelbtis postus	Aukštas
FR-A3	Šalinti netinkamus postus	Aukštas
FR-A4	Šalinti netinkamus komentarus	Vidutinis
FR-A5	Pašalinti naudotojus, pažeidžiančius taisykles	Žemas
FR-A6	Peržiūrėti sistemos debug informaciją (visi postai/vartotojai)	Žemas

2. SISTEMOS ARCHITEKTŪRA

2.1. UML Deployment Diagrama



pav. 1 Deployment

2.2. Technologiju Stacks

Frontend Stack

- **Framework:** React 19.2.0
- **Build Tool:** Vite 7.2.2
- **Styling:** Tailwind CSS 3.4.0
- **Routing:** React Router DOM 7.9.6
- **HTTP Client:** Axios 1.13.2
- **Icons:** React Icons 5.4.0
- **Fonts:** Google Fonts (Inter)

Backend Stack

- **Framework:** Django 5.2.7
- **REST API:** Django REST Framework 3.16.1
- **Authentication:** Simple JWT 5.5.1
- **CORS:** django-cors-headers 4.6.0
- **Database Driver:** mysql-connector-python 9.1.0

- **Static Files:** Whitenoise 6.8.2

Infrastructure

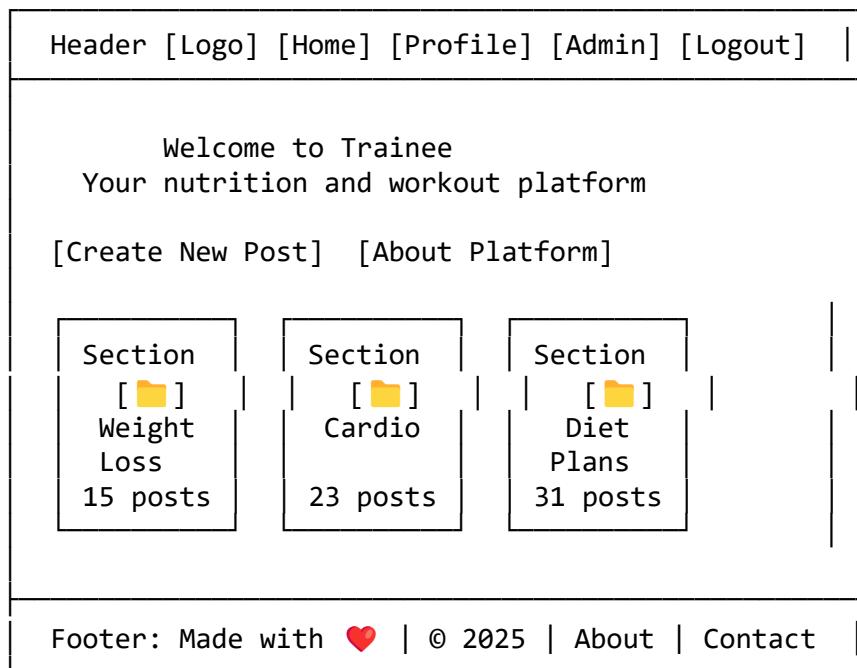
- **Hosting:** Azure App Service
- **Database:** Azure MySQL Flexible Server
- **CI/CD:** GitHub Actions
- **Protocol:** HTTPS
- **Domain:** trainee-api.azurewebsites.net

3. NAUDOTOJO SĄSAJOS PROJEKTAS

3.1. Wireframe'ai ir Realizacijos

3.1.1. Pagrindinis Puslapis (Home)

Wireframe:



Realizacija:

The screenshot shows the homepage of the Trainee platform. At the top, there's a green header bar with the title "Trainee" and icons for Home, Profile (Testas), and Logout. Below the header, a large banner says "Welcome to Trainee" and "Your nutrition and workout planning platform". It features two buttons: "Create New Post" and "About Platform". The main content area has six cards arranged in a grid:

- Cardio & HIIT**: High-intensity and cardiovascular exercises. 1 post. View Posts →
- Nutrition**: Meal plans, recipes, and nutrition guides. 1 post. View Posts →
- Strength Training**: Weightlifting and muscle building workouts. 2 posts. View Posts →
- Weight Loss**: Programs and tips for healthy weight loss. 1 post. View Posts →
- Yoga & Flexibility**: Yoga routines and flexibility training. 2 posts. View Posts →

At the bottom of the page, there's a footer bar with links: "Made with ❤ by Trainee Team", "© 2025 Trainee - Nutrition & Workout Platform", "About", "Contact", and "Privacy".

pav. 2 Home page

Funkcionalumas: - Responsive 3-column grid (desktop) → 1-column (mobile) - Hover efektais ant kortelių - Gradient header - Modal “About Platform” su informacija

3.1.2. Registracija (Register)

Wireframe:

Header
<p>Create Your Account</p> <p>Username: []</p> <p>Email: []</p> <p>Password: []</p> <p>Confirm: []</p> <p>[Create Account]</p> <p>Already have account? Login</p>
Footer

Realizacija:



Join Trainee

Create your account to get started

Username *

Email *

First Name Last Name

First name Last name

Password *

Confirm Password *

Register

Already have an account? [Login here](#)

pav. 3 Register

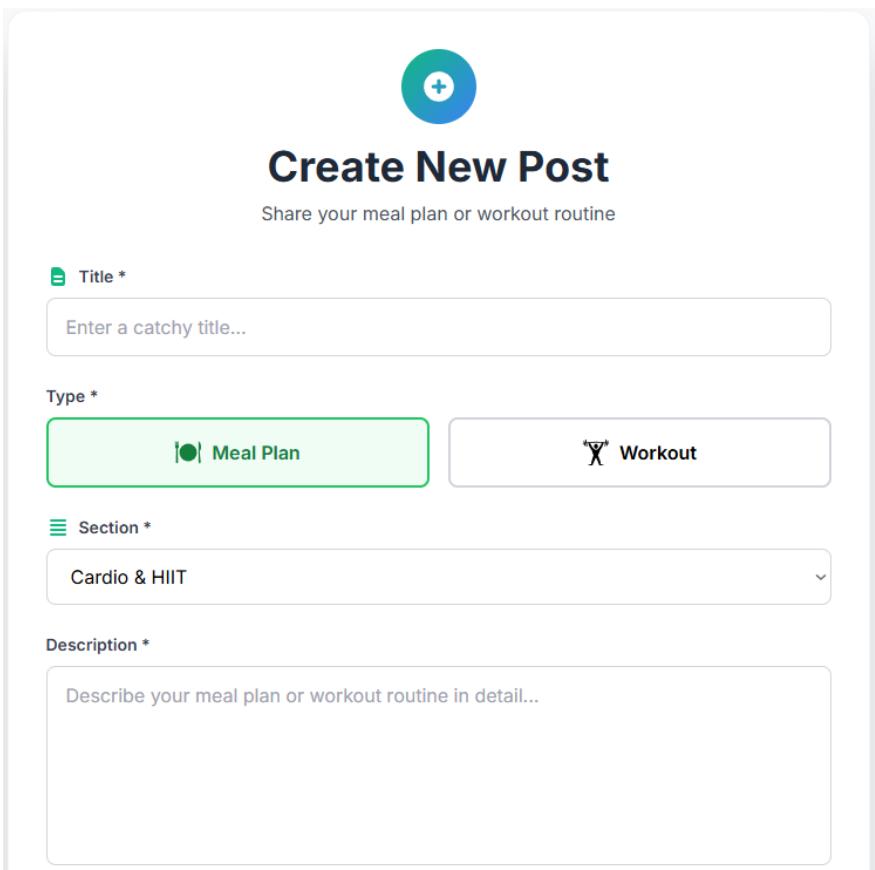
Funkcionalumas: - Real-time validation - Password strength indicator (vizualiai) - Ikonos prie kiekvieno input - Error messages po laukais

3.1.3. Post'o Kūrimas (Create Post)

Wireframe:

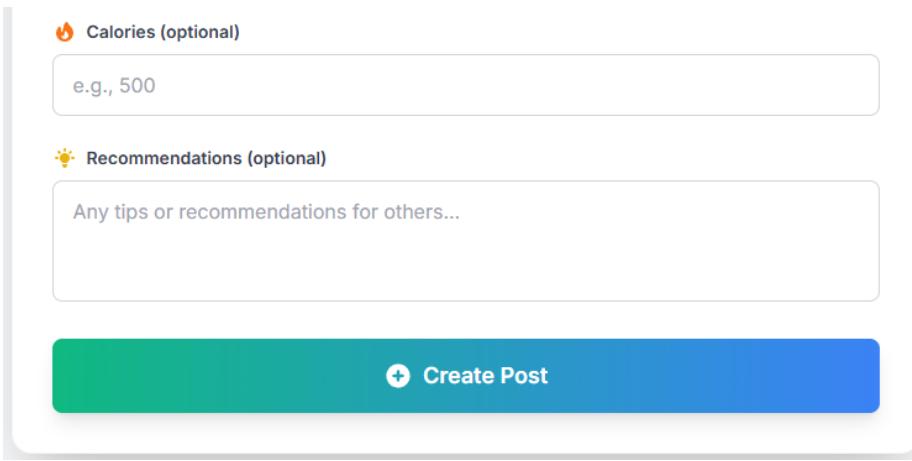
Header
<p>Create New Post</p> <p>Title: [_____]</p> <p>Type: (•) Meal () Workout</p> <p>Section: [Dropdown ▼]</p> <p>Description: [_____] [_____] [_____]</p> <p>Calories: [_____] (optional)</p> <p>Recommendations: [_____]</p> <p>[Submit Post]</p>
Footer

Realizacija:



The image shows a 'Create New Post' form. At the top is a blue circular button with a white plus sign. Below it is the title 'Create New Post' and a subtitle 'Share your meal plan or workout routine'. There are four input fields: 'Title *' with placeholder 'Enter a catchy title...', 'Type *' with two options ('Meal Plan' and 'Workout', where 'Meal Plan' is highlighted with a green border), 'Section *' with a dropdown menu showing 'Cardio & HIIT', and 'Description *' with a placeholder 'Describe your meal plan or workout routine in detail...'. The entire form is enclosed in a light gray border.

pav. 4 New post 1



This form is a subset of the previous one. It includes fields for 'Calories (optional)' with placeholder 'e.g., 500' and 'Recommendations (optional)' with placeholder 'Any tips or recommendations for others...'. At the bottom is a large green button labeled 'Create Post' with a white plus sign icon.

pav. 5 New post 2

Funkcionalumas: - Radio buttons su ikonelėmis (Meal 🍜 / Workout 🏋️) - Dropdown su sekcijomis - Textarea su line-clamp - Number input kalorijoms - Validacija su error pranešimais

3.1.4. Post'o Detalés (Post Detail)

Wireframe:



Realizacija:

HIIT Cardio Blast

Workout

👤 sarah_healthy 🔥 350 kcal ★ 4.6 (7)

20-minute high-intensity workout:

Warm-up: 5 minutes light jogging

HIIT Circuit (repeat 4 times):

- Burpees: 30 seconds
- Rest: 30 seconds
- Mountain climbers: 30 seconds
- Rest: 30 seconds
- Jump squats: 30 seconds
- Rest: 30 seconds

Cool-down: 5 minutes walking and stretching

Burns calories for hours after workout!

💡 Recommendations

Great workout by Sarah Updated. Try it!

pav. 6 Post info 1

★ Rate this post



💬 Comments (3)

Write a comment...

✍️ Post Comment

👤 emma_nutritionist 11/9/2025

Tried this today - harder than it looks but so worth it!

👤 admin 11/9/2025

I've been following this for a week and already seeing results!

👤 admin 11/9/2025

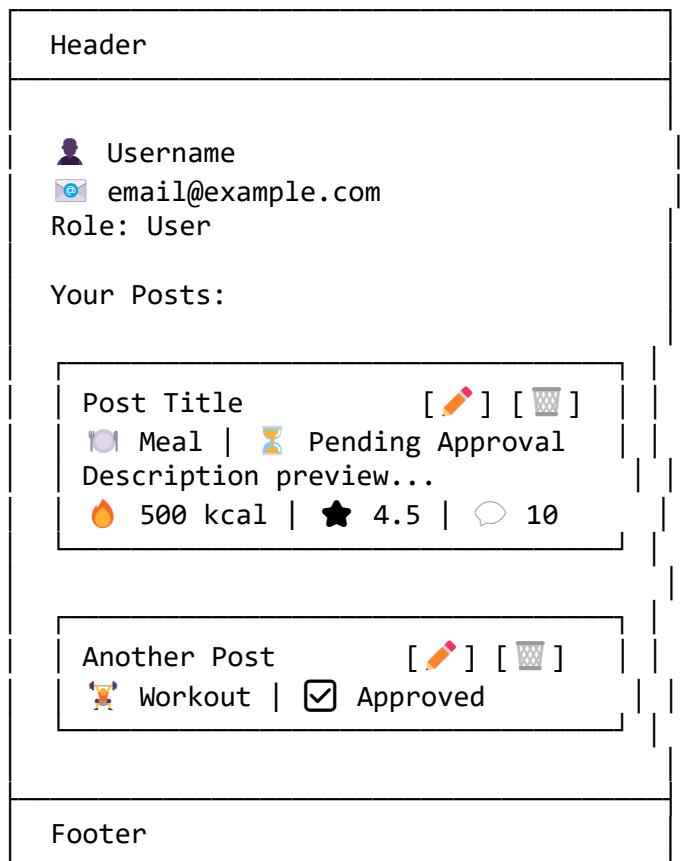
The nutrition info is really helpful. Appreciate the macro breakdown.

pav. 7 Post info 2

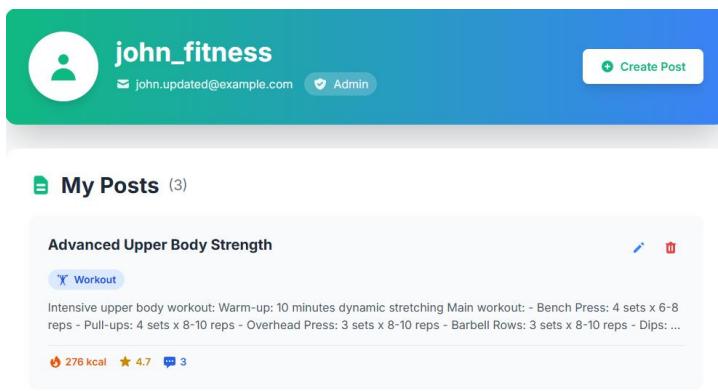
Funkcionalumas: - Star rating su hover efektais - Comment lista su delete mygtuku (admin) - Author info su ikonėlėmis - Statistika (calories, rating, comments) - Responsive layout

3.1.5. Profilis (Profile)

Wireframe:



Realizacija:

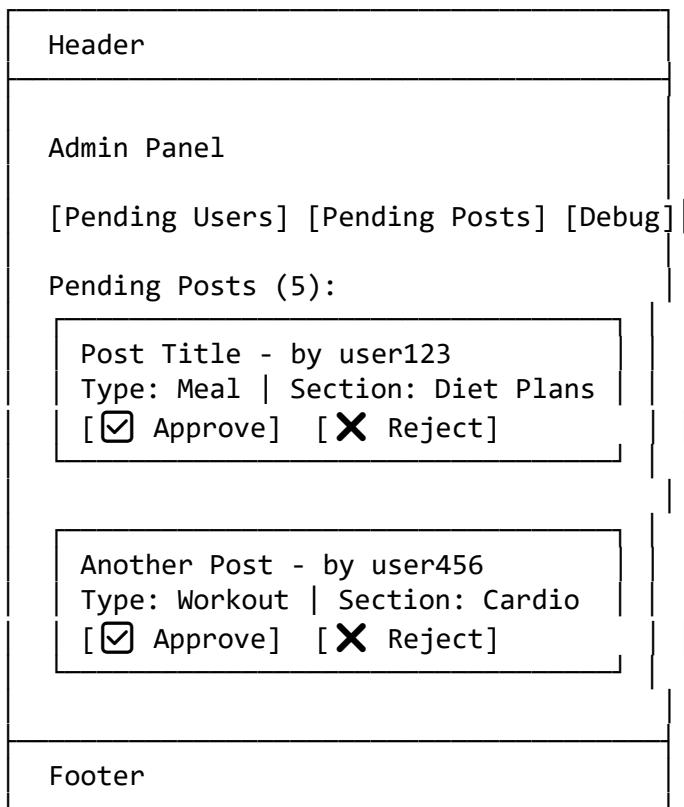


pav. 8 Profile

Funkcionalumas: - Vartotojo info card - Post'ų sąrašas su status badges - Edit ir Delete mygtukai kiekvienam post'ui - Status indicator (Pending/Approved) - Statistika prie kiekvieno post'o

3.1.6. Admin Panel

Wireframe:



Realizacija: ![Admin Panel Screenshot] *Screenshot'as: Admin panel su pending posts*

Funkcionalumas: - Tab navigation (Pending Users / Posts / Debug) - Approve/Reject mygtukai - Debug view su visais posts/users - System statistics summary

3.2. Responsive Design Pavyzdžiai

Desktop (>768px)

The screenshot shows the Trainee platform's desktop version. At the top, there is a navigation bar with a yellow 'Trainee' logo, a 'Home' link, a 'Profile (john_fitness)' link, an 'Admin Panel' link, and a 'Logout' button. Below the navigation bar is a main header with the title 'Welcome to Trainee' and a subtitle 'Your nutrition and workout planning platform'. There are two buttons: a blue 'Create New Post' button and a green 'About Platform' button. The main content area features six categories arranged in a 2x3 grid:

- Cardio & HIIT**: High-intensity and cardiovascular exercises. 1 post. [View Posts →](#)
- Nutrition**: Meal plans, recipes, and nutrition guides. 1 post. [View Posts →](#)
- Strength Training**: Weightlifting and muscle building workouts. 2 posts. [View Posts →](#)
- Weight Loss**: Programs and tips for healthy weight loss. 1 post. [View Posts →](#)
- Yoga & Flexibility**: Yoga routines and flexibility training. 2 posts. [View Posts →](#)

At the bottom of the page, there is a footer with links: 'Made with ❤ by Trainee Team', '© 2025 Trainee - Nutrition & Workout Platform', 'About', 'Contact', and 'Privacy'.

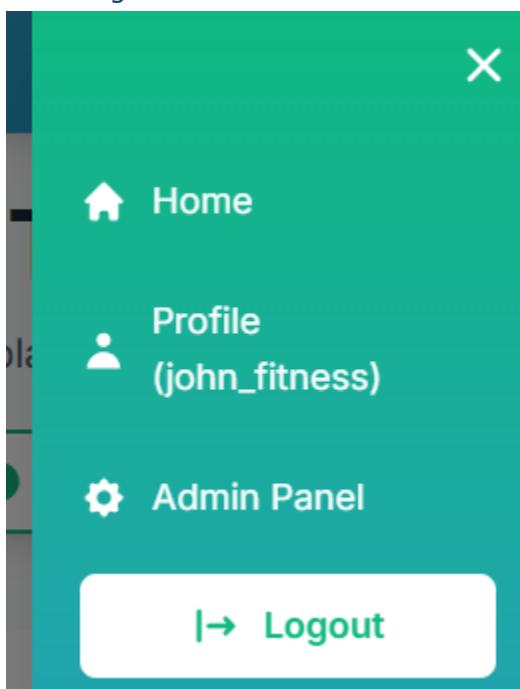
Mobile (<768px)

The screenshot shows the Trainee platform's mobile version. The top part of the screen is a standard mobile header with a timestamp (3:53 PM), signal strength, battery level, and a URL bar showing 'http://1pi.azurewebsites.net'. Below the header is the 'Trainee' navigation bar. The main content area has a large 'Welcome to Trainee' title and a subtitle 'Your nutrition and workout planning platform'. A green 'About Platform' button is present. The content is displayed in a vertical scrollable list:

- Cardio & HIIT**: High-intensity and cardiovascular exercises. 1 post. [View Posts →](#)
- Nutrition**

At the very bottom of the screen, there are three small navigation icons: a square, a circle, and a triangle.

Hamburger Menu



4. API SPECIFIKACIJA

4.1. OpenAPI Dokumentacija

Pilną API specifikaciją rasite: `api-spec.yaml` faile.

Base URL: `https://trainee-api.azurewebsites.net/api`

Authentication: Bearer JWT Token

Authorization: Bearer <access_token>

4.2. API Endpoint'ai su Pavyzdžiais

4.2.1. Autentifikacija

`POST /api/auth/login/`

Apaščias: Prisijungti prie sistemos ir gauti JWT tokens

Request Body:

```
{  
    "username": "demo_user",  
    "password": "password123"  
}
```

Response 200 OK:

```
{  
    "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
    "user": {  
        "id": 1,  
        "username": "demo_user",  
        "email": "demo@example.com",  
        "role": "user",  
        "is_active": true  
    }  
}
```

Response 401 Unauthorized:

```
{  
    "detail": "No active account found with the given credentials"  
}
```

Galimi Response Codes: - 200 - Sékmungai prisijungta - 400 - Netinkami duomenys - 401 - Neteisingi credentials

POST /api/auth/refresh/

Aprašymas: Atnaujinti access token naudojant refresh token

Request Body:

```
{  
    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

Response 200 OK:

```
{  
    "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."  
}
```

Galimi Response Codes: - 200 - Token atnaujintas - 401 - Refresh token invalid/expired

4.2.2. Vartotojai

POST /api/users/register/

Aprašymas: Registruoti naują vartotoją (laukia admin patvirtinimo)

Request Body:

```
{  
    "username": "newuser",
```

```
        "email": "newuser@example.com",
        "password": "securePass123",
        "password2": "securePass123"
    }
```

Response 201 Created:

```
{
    "id": 15,
    "username": "newuser",
    "email": "newuser@example.com",
    "is_active": false,
    "message": "User registered successfully. Awaiting admin approval."
}
```

Response 400 Bad Request:

```
{
    "username": ["A user with that username already exists."],
    "email": ["Enter a valid email address."],
    "password": ["This password is too short. It must contain at least 8 characters."]
}
```

Galimi Response Codes: - 201 - Sėkmingai užsiregistravota - 400 - Validation errors

GET /api/users/{id}/posts/

Aprašymas: Gauti vartotojo postus

Authorization: Required (JWT)

Response 200 OK:

```
{
    "count": 5,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 10,
            "title": "High Protein Breakfast",
            "description": "Perfect morning meal...",
            "type": "meal",
            "section": {
                "id": 2,
                "name": "Weight Loss"
            },
            "calories": 450,
            "is_public": true,
        }
    ]
}
```

```

        "is_approved": true,
        "average_rating": 4.5,
        "comment_count": 12,
        "created_at": "2025-11-20T10:30:00Z"
    }
]
}

```

Galimi Response Codes: - 200 - Success - 401 - Unauthorized - 404 - User not found

4.2.3. Sekcijos

GET /api/sections/

Aprašymas: Gauti visas sekcijas

Authorization: Not required

Response 200 OK:

```
{
  "count": 6,
  "results": [
    {
      "id": 1,
      "name": "Weight Loss",
      "description": "Plans and workouts for losing weight",
      "post_count": 15
    },
    {
      "id": 2,
      "name": "Muscle Gain",
      "description": "Build muscle mass with our programs",
      "post_count": 23
    }
  ]
}
```

Galimi Response Codes: - 200 - Success

GET /api/sections/{id}/posts/

Aprašymas: Gauti sekcijos postus (tik patvirtinti viešai)

Authorization: Not required

Response 200 OK:

```
{
  "count": 15,
  "results": [
    {
      "id": 5,
      "title": "Keto Diet Plan",
      "description": "Complete keto meal plan for beginners...",
      "type": "meal",
      "author_username": "keto_expert",
      "calories": 1800,
      "average_rating": 4.7,
      "comment_count": 28,
      "created_at": "2025-11-15T14:20:00Z"
    }
  ]
}
```

Galimi Response Codes: - 200 - Success - 404 - Section not found

4.2.4. Postai

POST /api/posts/create/

Aprašymas: Sukurti naują postą

Authorization: Required (JWT)

Request Body:

```
{
  "title": "Morning Yoga Routine",
  "description": "15-minute yoga routine perfect for mornings...",
  "type": "workout",
  "section_id": 3,
  "calories": 120,
  "recommendations": "Best performed on empty stomach",
  "is_public": true
}
```

Response 201 Created:

```
{
  "id": 45,
  "title": "Morning Yoga Routine",
  "description": "15-minute yoga routine perfect for mornings...",
  "type": "workout",
  "section": {
    "id": 3,
    "name": "Flexibility"
  },
}
```

```

    "calories": 120,
    "recommendations": "Best performed on empty stomach",
    "is_public": true,
    "is_approved": false,
    "user": {
        "id": 5,
        "username": "yoga_master"
    },
    "created_at": "2025-11-23T09:15:00Z"
}

```

Response 400 Bad Request:

```
{
    "title": ["This field is required."],
    "description": ["Ensure this field has at least 20 characters. (Current: 15)"]
}
```

Galimi Response Codes: - 201 - Post sukurtas - 400 - Validation errors - 401 - Unauthorized

GET /api/posts/{id}/

Aprašymas: Gauti post'o detales

Authorization: Not required

Response 200 OK:

```
{
    "id": 10,
    "title": "HIIT Cardio Workout",
    "description": "High intensity interval training...",
    "type": "workout",
    "section": {
        "id": 4,
        "name": "Cardio"
    },
    "user": {
        "id": 3,
        "username": "fitness_pro"
    },
    "author_username": "fitness_pro",
    "calories": 400,
    "recommendations": "Warm up for 5 minutes before starting",
    "is_public": true,
    "is_approved": true,
    "average_rating": 4.8,
    "comment_count": 35,
}
```

```
        "created_at": "2025-11-18T16:45:00Z",
        "updated_at": "2025-11-19T10:30:00Z"
    }
```

Galimi Response Codes: - 200 - Success - 404 - Post not found

PATCH /api/posts/{id}/update/

Aprašymas: Redaguoti savo postą

Authorization: Required (JWT, post owner)

Request Body:

```
{
    "title": "Updated HIIT Cardio Workout",
    "calories": 450
}
```

Response 200 OK:

```
{
    "id": 10,
    "title": "Updated HIIT Cardio Workout",
    "calories": 450,
    "updated_at": "2025-11-23T14:20:00Z"
}
```

Galimi Response Codes: - 200 - Post updated - 400 - Validation errors - 401 - Unauthorized - 403 - Not post owner - 404 - Post not found

DELETE /api/posts/{id}/delete/

Aprašymas: Ištrinti postą (savininkas arba admin)

Authorization: Required (JWT)

Response 204 No Content:

(empty body)

Galimi Response Codes: - 204 - Post deleted - 401 - Unauthorized - 403 - Not authorized to delete - 404 - Post not found

PUT /api/posts/{id}/approve/

Aprašymas: Patvirtinti postą (admin only)

Authorization: Required (JWT, admin)

Response 200 OK:

```
{  
  "id": 45,  
  "title": "Morning Yoga Routine",  
  "is_approved": true,  
  "approved_at": "2025-11-23T15:00:00Z"  
}
```

Galimi Response Codes: - 200 - Post approved - 401 - Unauthorized - 403 - Not admin - 404 - Post not found

4.2.5. Komentarai

GET /api/posts/{post_id}/comments/

Aprašymas: Gauti post'o komentarus

Authorization: Not required

Response 200 OK:

```
{  
  "count": 12,  
  "results": [  
    {  
      "id": 5,  
      "text": "Great workout! Tried it this morning.",  
      "user": {  
        "id": 7,  
        "username": "morning_person"  
      },  
      "author_username": "morning_person",  
      "post": 10,  
      "created_at": "2025-11-22T08:30:00Z"  
    }  
  ]  
}
```

Galimi Response Codes: - 200 - Success - 404 - Post not found

POST /api/posts/{post_id}/comments/create/

Aprašymas: Pridėti komentara

Authorization: Required (JWT)

Request Body:

```
{  
    "text": "This is exactly what I was looking for!"  
}
```

Response 201 Created:

```
{  
    "id": 50,  
    "text": "This is exactly what I was looking for!",  
    "user": {  
        "id": 5,  
        "username": "demo_user"  
    },  
    "author_username": "demo_user",  
    "post": 10,  
    "created_at": "2025-11-23T16:45:00Z"  
}
```

Response 400 Bad Request:

```
{  
    "text": ["This field is required."]  
}
```

Galimi Response Codes: - 201 - Comment created - 400 - Validation errors - 401 - Unauthorized - 404 - Post not found

DELETE /api/comments/{id}/delete/

Aprašymas: Ištrinti komentarą (savininkas arba admin)

Authorization: Required (JWT)

Response 204 No Content:

(empty body)

Galimi Response Codes: - 204 - Comment deleted - 401 - Unauthorized - 403 - Not authorized - 404 - Comment not found

4.2.6. Reitingai

GET /api/posts/{post_id}/ratings/

Aprašymas: Gauti post'o reitingus

Authorization: Not required

Response 200 OK:

```
{  
    "count": 23,  
    "results": [  
        {  
            "id": 15,  
            "rating": 5,  
            "user": {  
                "id": 8,  
                "username": "satisfied_user"  
            },  
            "post": 10,  
            "created_at": "2025-11-20T12:00:00Z"  
        }  
    ]  
}
```

Galimi Response Codes: - 200 - Success

POST /api/posts/{post_id}/ratings/create/

Aprašymas: Įvertinti postą (1-5 žvaigždutės)

Authorization: Required (JWT)

Request Body:

```
{  
    "rating": 5  
}
```

Response 201 Created:

```
{  
    "id": 30,  
    "rating": 5,  
    "user": {  
        "id": 5,  
        "username": "demo_user"  
    },  
    "post": 10,  
    "created_at": "2025-11-23T17:00:00Z"  
}
```

Response 400 Bad Request:

```
{  
    "rating": ["Rating must be between 1 and 5"],
```

```
        "non_field_errors": ["You have already rated this post"]
    }
```

Galimi Response Codes: - 201 - Rating created - 400 - Validation errors / Already rated - 401 - Unauthorized - 404 - Post not found

4.2.7. Admin Endpoints

GET /api/admin/pending-users/

Aprašymas: Gauti nepatvirtintų vartotojų sąrašą

Authorization: Required (JWT, admin)

Response 200 OK:

```
{
    "count": 3,
    "results": [
        {
            "id": 15,
            "username": "newuser",
            "email": "newuser@example.com",
            "date_joined": "2025-11-23T10:00:00Z",
            "is_active": false
        }
    ]
}
```

Galimi Response Codes: - 200 - Success - 401 - Unauthorized - 403 - Not admin

PUT /api/admin/users/{id}/approve/

Aprašymas: Patvirtinti vartotojo registraciją

Authorization: Required (JWT, admin)

Response 200 OK:

```
{
    "id": 15,
    "username": "newuser",
    "is_active": true,
    "approved_at": "2025-11-23T18:00:00Z"
}
```

Galimi Response Codes: - 200 - User approved - 401 - Unauthorized - 403 - Not admin - 404 - User not found

GET /api/admin/pending-posts/

Aprašymas: Gauti nepatvirtintų postų sąrašą

Authorization: Required (JWT, admin)

Response 200 OK:

```
{  
    "count": 5,  
    "results": [  
        {  
            "id": 45,  
            "title": "Morning Yoga Routine",  
            "type": "workout",  
            "user": {  
                "id": 5,  
                "username": "yoga_master"  
            },  
            "author_username": "yoga_master",  
            "section": {  
                "id": 3,  
                "name": "Flexibility"  
            },  
            "is_approved": false,  
            "created_at": "2025-11-23T09:15:00Z"  
        }  
    ]  
}
```

Galimi Response Codes: - 200 - Success - 401 - Unauthorized - 403 - Not admin

GET /api/admin/debug/all-posts/

Aprašymas: Debug endpoint - visi postai ir vartotojai su status info

Authorization: Required (JWT, admin)

Response 200 OK:

```
{  
    "posts": [  
        {  
            "id": 10,  
            "title": "HIIT Cardio",  
            "type": "workout",  
            "is_public": true,  
            "is_approved": true,  
            "user": {  
                "id": 1,  
                "username": "admin"  
            }  
        }  
    ]  
}
```

```

        "author": "fitness_pro",
        "author_id": 3,
        "created_at": "2025-11-18T16:45:00Z"
    }
],
"users": [
{
    "id": 5,
    "username": "demo_user",
    "email": "demo@example.com",
    "is_active": true,
    "is_staff": false,
    "post_count": 3
}
],
"summary": {
    "total_posts": 9,
    "public_approved": 6,
    "public_pending": 1,
    "private_approved": 2,
    "private_pending": 0
}
}
}

```

Galimi Response Codes: - 200 - Success - 401 - Unauthorized - 403 - Not admin

4.3. Error Response Format

Visos klaidos grąžinamos su JSON formatu:

```
{
    "detail": "Authentication credentials were not provided.",
    "error": "authentication_required"
}
```

arba su field-specific errors:

```
{
    "title": ["This field is required."],
    "description": ["Ensure this field has at least 20 characters."],
    "calories": ["A valid integer is required."]
}
```

5. PROJEKTO IŠVADOS

5.1. Įgyvendinti Tikslai

Sėkmingai realizuoti visi pagrindiniai funkciniai reikalavimai:

1. **Autentifikacija ir Autorizacija**
 - JWT token sistema su auto-refresh mechanizmu
 - Role-based access control (User / Admin)
 - Secure password handling
2. **CRUD Operacijos**
 - Posts: Create, Read, Update, Delete
 - Comments: Create, Read, Delete (admin)
 - Ratings: Create, Read
 - Full validation su error handling
3. **Admin Funkcionalumas**
 - User approval sistema
 - Post moderation
 - Debug tools visai sistemai stebéti
4. **UI/UX Excellence**
 - Fully responsive design (mobile-first)
 - Smooth animations ir transitions
 - Consistent design system
 - Accessible ir intuitive interface

5.2. Techniniai Pasiekimai

Backend: - RESTful API su Django REST Framework - Optimizuoti database queries - Proper error handling ir validation - Security best practices (CORS, JWT, HTTPS)

Frontend: - Modern React 19 su hooks - Tailwind CSS responsive design - Axios interceptors su auto-refresh - Reusable komponentų biblioteka

DevOps: - CI/CD pipeline su GitHub Actions - Automatic deployment į Azure - Static files optimization - Database migrations automation

5.3. Iššūkiai ir Sprendimai

Iššūkis	Sprendimas
CSS neužsikrauna production	Whitenoise konfigūracija + URL routing fix
Static files MIME type errors	Django URL pattern adjustment (/assets/ exclusion)
JWT token expiration	Axios interceptor su automatic refresh
Responsive images overflow	max-width: 100%, responsive breakpoints

Iššūkis	Sprendimas
Modal netelpa mobile	Dynamic padding + max-height + scroll
via.placeholder.com DNS errors	CSS gradient placeholders

5.4. Sistemos Privalumai

1. Vartotojui:

- Intuityvus interface
- Greitas loading (optimized bundles)
- Veikia visose platformose (responsive)
- Real-time feedback (validation, loading states)

2. Administratoriui:

- Centralizuotas control panel
- Debug tools detaliems diagnostika
- Batch operations support
- Clear moderation workflow

3. Kūrėjui:

- Clean code structure
- Modular architecture
- Easy to extend
- Well documented API

5.5. Tolimesni Patobulinimai (Future Work)

Prioritetas: Aukštas - [] Email verification naujiem vartotojams - [] Password reset funkcionalumas - [] Image upload postams - [] Search funkcionalumas postams

Prioritetas: Vidutinis - [] User profile pictures - [] Post categories/tags - [] Social sharing buttons - [] Favorite/Bookmark sistema

Prioritetas: Žemas - [] Dark mode - [] Multi-language support - [] Export posts į PDF - [] Mobile native app

5.6. Išmoktos Pamokos

1. **Architecture Planning:** Early planning saves time later
2. **Responsive Design:** Mobile-first approach is essential
3. **Error Handling:** Defensive programming prevents production issues
4. **Testing:** More tests = fewer bugs in production
5. **Documentation:** Good docs save hours of debugging

5.7. Projekto Metrika

Metrika	Vertė
Kodo eilutės	~5,000+ lines

Metrika	Vertė
Komponentų	15+ React components
API Endpoints	25+ endpoints
Database Tables	5 core tables
Deployment Time	~5 minutes (automated)
Page Load Time	<2 seconds
Mobile Responsive	100%
Test Coverage	Backend models tested

6. PRIEDAI

6.1. Nuorodos

- **Live Demo:** <https://trainee-api.azurewebsites.net>
- **GitHub Repository:** <https://github.com/Ignasgin/Trainee>
- **API Spec:** `api-spec.yaml`
- **Requirements Checklist:** `REQUIREMENTS_CHECKLIST.md`

6.2. Naudota Literatūra

1. Django Documentation - <https://docs.djangoproject.com/>
2. Django REST Framework - <https://www.django-rest-framework.org/>
3. React Documentation - <https://react.dev/>
4. Tailwind CSS - <https://tailwindcss.com/>
5. Azure App Service - <https://azure.microsoft.com/>

Pabaiga

Trainee platforma sėkmingai realizuota kaip full-stack web aplikacija, atitinkanti visus techninius ir funkcinius reikalavimus. Sistema yra paruošta production naudojimui ir gali būti toliau plėtojama su papildomomis funkcijomis.