

Gyógynövénykereső weboldal és webáruház

Fejlesztői dokumentáció

Felhasznált technológiák

Backend

- **Laravel API** (PHP keretrendszer) ^11.31
- **Laravel-sanctum** (Autentikációs kiegészítő csomag) ^4.0
- **MariaDB server** (Adatbázis kiszolgáló) v8.2.12

Készítéshez használt programok

- Visual Studio Code v1.98.2
- Insomnia v11.0.0
- XAMPP Control Panel v3.3.0
- phpMyAdmin v5.2.1

Frontend

- Angular keretrendszer ^19.1.0
- Bootstrap ^5.3.3
- Chart.js ^4.4.8
- @popperjs/core ^2.11.8

Telepítés

- **Frontend:**
 - `npm install`
 - `ng serve` (fejlesztői szerver indítása, alapértelmezett URL: `http://localhost:4200`)
- **Backend:**
 - `composer install`
 - `.env` fájl létrehozása és adatbázis adatok beállítása
 - Adatbázist kiszolgáló szerver elindítása (pl. fejlesztői környezetben XAMPP)
 - `php artisan app:setup`

Miután a telepítés közben létrehoztuk a Szuper Admin-t, a következő linken léphetünk be:

`{Frontend URL}/admin/login` (pl.: `http://localhost:4200/admin/login`)

- Amennyiben a Szuper Admin létrehozása nem futott volna le, vagy újabbat szeretnénk létrehozni (a Szuper Admin tud Admin jogot adni regisztrált felhasználóknak az Admin Panel felületén), akkor a következő paranccsal indíthatunk egy újabb létrehozást:

Tartalom:

1. Bevezető
2. Backend
 - Backend projekt struktúra
 - Adatbázis szerkezete
 - Modellek és kapcsolataik
 - Vezérlők (Controllers)
 - Middleware komponensek
 - Console Commandok
 - Útvonalak (Routes)
 - Email küldési rendszer
 - Jelszó-visszaállítási rendszer
 - Végpontok megnevezése és leírása
3. Frontend
 - Az alkalmazás struktúrája és alapszerkezete
 - A fejléc és a keresési funkciók
 - Kezdőlap és termékmegtekintés
 - Kosárkezelés
 - Rendelések kezelése
 - Webshop főoldal és szűrési funkciók
 - Hitelesítési és adminisztrációs funkciók
 - Jelszó-visszaállítási komponensek
 - Kapcsolódó tartalmak kezelése
 - Komponensek és osztályváltozók listája
 - Frontend projekt struktúra
 - Fontos frontend fájlok
 - Adminisztrációs felület komponensei
 - Webshop komponensei

1. Bevezető

A Gyógynövénykereső egy innovatív online platform, amely a természetes alapú egészségmegőrzést és életmódot támogatja egy különleges keresőfunkcióval. Az oldal segítségével a felhasználók betegségek vagy gyógynövények alapján kereshetnek információkat, és részletes leírásokat találhatnak a különböző gyógynövények tulajdonságairól, hatásairól és felhasználási módjairól.

Az oldal különlegessége, hogy a felhasználók aktívan hozzájárulhatnak az információk bővítéséhez, saját bejegyzések, tapasztalatok vagy észrevételek megosztásával. Ez a közösségi jelleg segíti az adatbázis folyamatos fejlődését és a tudásbázis gyarapodását.

A keresőhöz szorosan kapcsolódik egy webáruház, amely a találatokhoz igazított, releváns étrend-kiegészítőket, gyógyhatású termékeket és egyéb természetes megoldásokat kínál. Ez a kombinált rendszer

megkönnyíti a felhasználók számára a számukra megfelelő termékek azonosítását és beszerzését.

Az oldalon a vásárlási folyamat átlátható és felhasználóbarát. A termékek böngészésekor a vásárlók megtekinthetik az árakat, mennyiségeket, készletinformációkat és kedvezményeket. A kosárba helyezett termékek után azonnal látható a részösszeg, a szállítási költség és a végösszeg. Az oldal emellett tájékoztat arról, mennyit kell még vásárolni az ingyenes szállítás eléréséhez.

A rendelés során a felhasználók megadhatják számlázási és szállítási adataikat, valamint kiválaszthatják a szállítási módot (például postai kézbesítést vagy expressz szállítást) és a fizetési lehetőséget (kártyás fizetés, utánvét stb.).

A Gyógynövénykereső célja, hogy egy átfogó, intuitív és hatékony platformot biztosítson, amely egyszerre segíti az információkeresést, a közösségi együttműködést és a termékbeszerzést. Az oldal ösztönzi a természetes alapú megoldások felfedezését, és olyan vásárlási élményt kínál, amely egyesíti a tudás, a közösségi támogatás és a praktikum előnyeit.

Az oldal készítői:

- **Backend:** Stark Ernő
- **Frontend:** Nádassy Ilona és Markó Dániel
- **Teszttek:** Markó Dániel

2. Backend

Az oldal adatbázisszerkezete és végpontjai gondosan lettek kialakítva, hogy egyszerűvé és hatékonyá tegyék a felhasználói interakciókat, valamint az adminisztrációs folyamatokat. Az adatbázis jól strukturált táblákból áll, amelyek egymással összekapcsolódnak, így biztosítva az összes szükséges adat tárolását és feldolgozását.

Backend projekt struktúra

A Laravel projekt standard MVC (Model-View-Controller) architektúrát követ, az alábbi főbb könyvtárszerkezettel:

```
backend/  
├── app/                                # Az alkalmazás fő könyvtára  
│   ├── Console/                       # Konzol parancsok  
│   │   └── Commands/                 # Egyedi parancsok definíciói  
│   ├── Exceptions/                   # Kivételkezelés  
│   ├── Http/                         # HTTP kapcsolódó kódok  
│   │   ├── Controllers/              # Vezérlők - üzleti logika  
│   │   │   ├── Admin/               # Admin vezérlők  
│   │   │   ├── User/                # Felhasználói vezérlők  
│   │   │   └── PasswordResetController.php # Jelszó-visszaállítás kezelő  
│   │   ├── Middleware/               # Kérések köztes feldolgozása  
│   │   └── Requests/                 # Űrlap kérések validációja  
│   ├── Mail/                         # Email sablonok és logika  
│   └── Models/                       # Adatbázis modellek
```

	Providers/	# Szolgáltatás-szolgáltatók
	bootstrap/	# Alkalmazás indítási szkriptek
	config/	# Konfigurációs fájlok
	database/	# Adatbázis-kapcsolódó fájlok
	factories/	# Modellek gyártó osztályai
	migrations/	# Adatbázis migrációs fájlok
	seeders/	# Adatbázis kezdeti feltöltés
	sql/	# SQL dump fájlok
	public/	# Nyilvánosan elérhető fájlok
	storage/	# Feltöltött fájlok tárolása
	resources/	# Nyers erőforrások
	css/	# CSS fájlok
	js/	# JavaScript fájlok
	views/	# Blade nézetsablonok
	routes/	# Útvonal definíciók
	api.php	# API útvonalak
	web.php	# Web útvonalak
	storage/	# Alkalmazás által generált fájlok
	tests/	# Tesztfájlok

Console Commands

A projekt két fontos egyedi konzolos parancsot tartalmaz:

1. **CreateAdmin** ([app/Console/Commands/CreateAdmin.php](#)): Szuper Admin felhasználó létrehozására szolgál, opcionálisan létrehozhat felhasználót is admin joggal.
2. **SetupCommand** ([app/Console/Commands/SetupCommand.php](#)): Az alkalmazás kezdeti beállítását végzi, migrációkat futtat, betölti az SQL dump-ot és létrehoz egy admin felhasználót.

Controllers

A vezérlők a következő fő csoportokra oszthatók:

1. **Admin vezérlők:** Admin felülethez kapcsolódó műveletek

- AdminOrderController
- PaymentMethodController
- ShippingMethodController
- StatisticsController

2. **Autentikációs vezérlők:**

- AdminAuthController
- UserController (regisztráció, bejelentkezés, profil)
- PasswordResetController (jelszó-visszaállítás)

3. **Közös vezérlők:**

- CartController (kosárkezelés)
- CategoryController (kategóriák)

- CheckoutController (fizetés)
- ContactController (kapcsolatfelvétel)
- PostController (blogbejegyzések)
- ProductController (termékek)
- ProductCategoryController (termékkategóriák)
- NewsletterController (hírlevélkezelés)

Adatbázis szerkezete, táblák neve és mezőnevek

1. **users** - Felhasználói adatok tárolása

- id: Azonosító
- name: Név
- email: Email cím
- email_verified_at: Email megerősítés ideje
- password: Jelszó
- country, postal_code, city, street, address_line_2: Felhasználó címe
- is_admin: Admin jog státusz
- remember_token: Emlékezési token
- created_at, updated_at: Létrehozási és módosítási idő

2. **password_reset_tokens** - Jelszó visszaállító tokenek tárolása

- email: Azonosító
- token: Visszaállító token
- created_at: Token létrehozási idő

3. **sessions** - Felhasználói munkamenet adatok tárolása

- id: Munkamenet azonosító
- user_id: Kapcsolat a felhasználóhoz
- ip_address: IP cím
- user_agent: Felhasználói ügynök információ
- payload: Munkamenet adatok
- last_activity: Utolsó aktivitás

4. **cache** - Gyorsítótár adatok tárolása

- key: Kulcs
- value: Érték
- expiration: Lejáratási idő

5. **personal_access_tokens** - Hozzáférési tokenek tárolása

- id: Azonosító
- tokenable: Kapcsolat a token tulajdonoshoz
- name: Token neve
- token: Token értéke
- abilities: Engedélyek

- last_used_at: Utolsó használat
- expires_at: Lejáratási idő
- created_at, updated_at: Létrehozási és módosítási idő

6. **product_categories** - Termékkategóriák tárolása

- id: Azonosító
- name: Kategória neve
- slug: Egyedi URL azonosító
- description: Kategória leírása
- created_at, updated_at: Létrehozási és módosítási idő

7. **products** - Termékek részletei

- id: Azonosító
- category_id: Kapcsolat kategóriához
- name: Termék neve
- slug: Egyedi URL azonosító
- latin_name: Latin név
- description: Leírás
- usage: Használat módja
- price: Ár
- discount_price: Kedvezményes ár
- stock_quantity: Készlet mennyisége
- unit: Egység
- is_available: Elérhetőség
- is_featured: Kiemeltség
- view_count: Megtekintésszám
- sale_count: Eladásszám
- created_at, updated_at: Létrehozási és módosítási idő

8. **product_images** - Termékképek tárolása

- id: Azonosító
- product_id: Kapcsolat a termékhez
- image_url: Kép útvonala
- is_primary: Elsődleges kép jelzés
- display_order: Megjelenítési sorrend
- created_at, updated_at: Létrehozási és módosítási idő

9. **admins** - Adminok adatai

- id: Azonosító
- name: Név
- email: Email cím
- password: Jelszó
- created_at, updated_at: Létrehozási és módosítási idő

10. **categories** - Blogbejegyzések kategóriáinak tárolása

- id: Azonosító
- name: Kategória neve
- slug: Egyedi URL azonosító
- description: Leírás
- parent_id: Szülő kategória azonosító
- created_at, updated_at: Létrehozási és módosítási idő

11. **posts** - Blogbejegyzések tárolása

- id: Azonosító
- title: Bejegyzés címe
- slug: Egyedi URL azonosító
- content: Tartalom
- image_path: Kép útvonala
- excerpt: Kivonat
- diseases: Kapcsolódó betegségek
- featured: Kiemeltség
- author_id: Kapcsolat szerzőhöz
- author_type: Szerző típusa (user/admin)
- category_id: Kapcsolat kategóriához
- published_at: Publikálás ideje
- status: Státusz (draft/published/archived)
- created_at, updated_at: Létrehozási és módosítási idő

12. **comments** - Hozzászólások tárolása

- id: Azonosító
- post_id: Kapcsolat bejegyzéshez
- user_id: Kapcsolat felhasználóhoz
- content: Tartalom
- status: Státusz (approved/pending/spam)
- created_at, updated_at: Létrehozási és módosítási idő

13. **tags és post_tag** - Címkék és blog-címke kapcsolat táblák

- A tags tábla: id, name, slug
- A post_tag tábla: id, post_id, tag_id

14. **contacts** - Kapcsolatfelvételi üzenetek

- id: Azonosító
- name: Név
- email: Email cím
- subject: Tárgy
- message: Üzenet
- is_read: Olvasottság jelzése
- created_at, updated_at: Létrehozási és módosítási idő

15. **order_statuses** - Rendelési státuszok

- id: Azonosító
- name: Státusz neve
- color: Színkód
- description: Leírás
- sort_order: Megjelenítési sorrend
- created_at, updated_at: Létrehozási és módosítási idő

16. **shipping_methods** - Szállítási módok

- id: Azonosító
- name: Szállítási mód neve
- description: Leírás
- cost: Szállítási költség
- estimated_delivery_days: Becsült szállítási idő napokban
- is_active: Aktív státusz
- created_at, updated_at: Létrehozási és módosítási idő

17. **payment_methods** - Fizetési módok

- id: Azonosító
- name: Fizetési mód neve
- description: Leírás
- is_active: Aktív státusz
- sort_order: Megjelenítési sorrend
- created_at, updated_at: Létrehozási és módosítási idő

18. **orders** - Rendelések

- id: Azonosító
- order_number: Rendelésszám
- user_id: Kapcsolat a felhasználóhoz
- status_id: Kapcsolat a státuszhoz
- shipping_method_id: Kapcsolat a szállítási módhoz
- payment_method_id: Kapcsolat a fizetési módhoz
- billing_* és shipping_*: Számlázási és szállítási adatok
- subtotal, shipping_cost, tax_amount, discount_amount, total: Összegek
- notes: Megjegyzés
- paid_at, shipped_at, completed_at: Fontos dátumok
- created_at, updated_at: Létrehozási és módosítási idő

19. **order_items** - Rendelési tételek

- id: Azonosító
- order_id: Kapcsolat a rendeléshez
- product_id: Kapcsolat a termékhez
- product_name, product_latin_name: Termék neve és latin neve
- quantity: Mennyiség
- unit: Egység

- unit_price, discount_price: Árak
- subtotal: Részösszeg
- created_at, updated_at: Létrehozási és módosítási idő

20. **order_status_histories** - Rendelési státuszváltozások

- id: Azonosító
- order_id: Kapcsolat a rendeléshez
- status_id: Kapcsolat a státuszhoz
- user_id: A státuszt módosító felhasználó
- comment: Megjegyzés
- created_at, updated_at: Létrehozási és módosítási idő

21. **newsletter_subscribers** - Hírlevél-feliratkozók

- id: Azonosító
- email: Email cím
- user_id: Kapcsolat a felhasználóhoz (ha regisztrált)
- is_active: Aktív státusz
- token: Egyedi token leiratkozáshoz
- subscribed_at: Feliratkozás ideje
- unsubscribed_at: Leiratkozás ideje
- created_at, updated_at: Létrehozási és módosítási idő

22. **carts** - Kosarak tárolása

- id: Azonosító
- user_id: Kapcsolat a felhasználóhoz
- session_id: Munkamenet azonosító
- status: Státusz (active/converted)
- expires_at: Lejárat ideje
- created_at, updated_at: Létrehozási és módosítási idő

23. **cart_items** - Kosártételek tárolása

- id: Azonosító
- cart_id: Kapcsolat a kosárhoz
- product_id: Kapcsolat a termékhez
- quantity: Mennyiség
- unit_price: Egységár
- discount_price: Kedvezményes ár
- created_at, updated_at: Létrehozási és módosítási idő

Modellek és kapcsolataik

Az alkalmazás számos modellt használ az adatbázis tábláinak kezeléséhez. A legfontosabb modellek és kapcsolataik:

1. **User**

- Egy felhasználónak lehet több rendelése (one-to-many)
- Lehet admin joga (is_admin attribútum)
- Lehetnek közvetlen blog bejegyzései (polymorphic relationship)

2. Admin

- Különálló model az adminisztrátorok kezelésére
- Lehetnek blog bejegyzései (polymorphic relationship)

3. Product

- Tartozik egy termékkategóriához (belongs-to)
- Lehetnek képei (has-many)
- Lehet a kosárban és rendelési tételekben
- Virtual attributeok: primary_image, current_price, is_on_sale, discount_percentage

4. ProductCategory

- Lehetnek termékei (has-many)
- Lehet szülő-gyermek kapcsolata (self-referencing)

5. Post

- Tartozik egy kategóriához (belongs-to)
- Tartozik egy szerzőhöz (polymorphic relationship)
- Lehetnek címkéi (many-to-many)
- Lehetnek hozzászólásai (has-many)

6. Category

- Lehetnek blogbejegyzései (has-many)
- Lehet szülő-gyermek kapcsolata (self-referencing)

7. Cart és CartItem

- Egy kosárnak lehetnek tételei (has-many)
- A CartItem tartozik egy kosárhoz és egy termékhez (belongs-to)

8. Order és OrderItem

- Egy rendelésnek lehetnek tételei (has-many)
- Az OrderItem tartozik egy rendeléshez és egy termékhez (belongs-to)
- Egy rendelésnek lehet státusza és státusztörténete

Vezérlők (Controllers)

A backend vezérlők kezelik a bejövő kéréseket, végrehajtják az üzleti logikát és visszaadják a megfelelő válaszokat. A főbb vezérlők:

1. AdminAuthController: Admin bejelentkezés és kijelentkezés kezelése

- `login()`: Admin bejelentkeztetése token létrehozással

- `logout()`: Admin kijelentkeztetése token törlésével

2. **CartController**: Kosárkezelés

- `index()`: Kosár tartalmának lekérése
- `addItem()`: Termék kosárba helyezése
- `updateItem()`: Kosárban lévő termék mennyiségének módosítása
- `removeItem()`: Termék eltávolítása a kosárból
- `clear()`: Kosár teljes kiürítése

3. **CheckoutController**: Rendelési folyamat

- `checkout()`: Rendelés feldolgozása, mentése, kosár konvertálása

4. **PostController**: Blogbejegyzések kezelése

- `index()`, `show()`: Bejegyzések listázása és részletes megtekintése
- `store()`, `update()`, `destroy()`: Bejegyzések kezelése (CRUD)
- `getFeaturedPosts()`, `getLatestPosts()`: Kiemelt és legújabb bejegyzések lekérése
- `searchByDiseases()`, `searchInContent()`: Keresési funkciók

5. **ProductController**: Termékek kezelése

- CRUD műveletek termékekhez
- Képek kezelése
- Megtekintések számolása

6. **Admin/AdminOrderController**: Admin rendeléskezelés

- `index()`, `show()`: Rendelések listázása és részletes megtekintése
- `updateStatus()`: Rendelési státusz módosítása
- `getStatuses()`: Státuszok lekérése

7. **User/OrderController**: Felhasználói rendelések

- `index()`, `show()`: Bejelentkezett felhasználó rendeléseinek listázása és részletes megtekintése

8. **NewsletterController**: Hírlevél kezelése

- `subscribe()`: Feliratkozás kezelése
- `unsubscribeWithToken()`: Leiratkozás egyedi token alapján
- `toggleStatus()`: Feliratkozás státuszának módosítása

9. **PasswordResetController**: Jelszó-visszaállítás kezelése

- `forgotPassword()`: Jelszó-visszaállítási link küldése
- `resetPassword()`: Jelszó visszaállítása a kapott tokennel

Middleware komponensek

A middleware komponensek a kérések feldolgozását segítik a végpontok elérése előtt vagy után. A fontosabb middleware-ek:

1. **AdminMiddleware:** Ellenőrzi, hogy a kérés admin felhasználótól származik-e
 - Teszteli a kérést a `admin` guardból vagy az `is_admin` attribútumot a `sanctum` token alapján
 - Egyes útvonalakhoz Szuper Admin jogosultságot követel meg
2. **Authenticate:** Hitelesítést végez az aktuális guard alapján
 - Ha a kérés nem hitelesített, a bejelentkezési oldalra irányít
3. **EncryptCookies:** A süti titkosítását végzi
4. **TrustProxies:** Proxy szerverektől érkező kérések kezelését segíti
5. **ValidateSignature:** Aláírt URL-ek ellenőrzésére szolgál

Console Commandok

A Laravel artisan konzolos parancsai fontos szerepet játszanak a fejlesztésben és telepítésben:

1. **admin:create:** Szuper Admin létrehozása
 - Paraméterek: `--name`, `--email`
 - Létrehozza a szükséges admin felhasználót és opcionálisan egy regular user-t is admin joggal
2. **app:setup:** Alkalmazás telepítését segítő parancs
 - Ellenőrzi az `.env` fájl létezését
 - Létrehozza az alkalmazás kulcsot
 - Futtatja a migrációkat
 - Betölti a seeder adatokat
 - Opcionálisan létrehoz egy Szuper Admin

Jelszó-visszaállítási rendszer

A backend oldalon a jelszó-visszaállítás a `PasswordResetController.php` vezérlőn keresztül valósul meg, amely két fő végpontot biztosít:

1. **/forgot-password** (POST):
 - E-mail cím elfogadása és validálása
 - Rate limiting alkalmazása (maximum 3 kérés/óra ugyanazon e-mail és IP kombinációra)
 - Jelszó-visszaállító token generálása és e-mail küldése
 - A token a `password_reset_tokens` táblában tárolódik
2. **/reset-password** (POST):
 - Token, e-mail és új jelszó elfogadása
 - Jelszó validálása (min. 8 karakter, nagy- és kisbetű, szám)
 - Jelszó frissítése és a token törlése

A kontroller a Laravel beépített jelszó-visszaállítási rendszerét használja (**Password** facade), amely automatikusan kezeli a tokenek generálását, e-mail küldést és adatbázis műveleteket.

Biztonsági funkcióként a kontroller rate limitet alkalmaz a **RateLimiter** facade segítségével, megakadályozva a túl sok kérést azonos e-mail címről vagy IP-címről. Ez véd a brute force támadások és szolgáltatás-megtagadási támadások ellen.

Email küldési rendszer

A jelszó-visszaállítási rendszer az alapértelmezett Laravel e-mail sablon rendszert használja. A sablon testreszabása a `/resources/views/vendor/notifications/email.blade.php` fájlban keresztül történik, amely a következő jellemzőkkel rendelkezik:

- Professzionális és felhasználóbarát üzenet stílus
- Kiemelt gomb a jelszó-visszaállítási linkhez
- Explicit magyarázat a link érvényességi idejéről (60 perc)
- Alternatív hozzáférési lehetőség szöveges URL formájában
- Explicit tájékoztatás, hogy mi a teendő, ha nem a felhasználó kérte a jelszó-visszaállítást

A jelszó-visszaállító e-mail tartalma az **AuthServiceProvider.php** fájlban is testreszabható a **ResetPassword::toMailUsing()** metódus segítségével, amely lehetővé teszi a témák, üzenetek és további paraméterek módosítását.

Végpontok megnevezése és leírása

Végpont	Metódus	Kapcsolat típusa	Leírás
/register	POST	Nyilvános	Felhasználó regisztrálása
/login	POST	Nyilvános	Felhasználó bejelentkezése
/logout	POST	Autentikált (auth:sanctum)	Felhasználó kijelentkezése
/profile	PUT	Autentikált (auth:sanctum)	Felhasználói profil frissítése
/getprofile	GET	Autentikált (auth:sanctum)	Felhasználói profil lekérdezése
/orders	GET	Autentikált (auth:sanctum)	Felhasználói rendelések listázása
/orders/{id}	GET	Autentikált (auth:sanctum)	Egy adott rendelés részleteinek lekérése
/admin/login	POST	Nyilvános	Admin bejelentkezés
/admin/logout	POST	Autentikált (auth:sanctum)	Admin kijelentkezés

Végpont	Metódus	Kapcsolat típusa	Leírás
/admin/posts	GET	Adminisztrátor (auth:sanctum)	Blogbejegyzések listázása
/admin/posts/{id}	GET	Adminisztrátor (auth:sanctum)	Egy adott blogbejegyzés részleteinek lekérése
/admin/posts	POST	Adminisztrátor (auth:sanctum)	Új blogbejegyzés létrehozása
/admin/posts/{id}	POST	Adminisztrátor (auth:sanctum)	Blogbejegyzés frissítése
/admin/posts/{id}	DELETE	Adminisztrátor (auth:sanctum)	Blogbejegyzés törlése
/admin/categories	GET	Adminisztrátor (auth:sanctum)	Blog kategóriák listázása
/admin/categories	POST	Adminisztrátor (auth:sanctum)	Új blog kategória létrehozása
/admin/categories/{id}	DELETE	Adminisztrátor (auth:sanctum)	Blog kategória törlése
/admin/product-categories	POST	Adminisztrátor (auth:sanctum)	Új termékkategória létrehozása
/admin/product-categories/{id}	PUT	Adminisztrátor (auth:sanctum)	Termékkategória frissítése
/admin/product-categories/{id}	DELETE	Adminisztrátor (auth:sanctum)	Termékkategória törlése
/admin/products	POST	Adminisztrátor (auth:sanctum)	Új termék létrehozása
/admin/products/{id}	POST	Adminisztrátor (auth:sanctum)	Termék frissítése
/admin/products/{id}	DELETE	Adminisztrátor (auth:sanctum)	Termék törlése
/admin/messages	GET	Adminisztrátor (auth:sanctum)	Üzenetek listázása
/admin/messages/{id}/read	PUT	Adminisztrátor (auth:sanctum)	Üzenet olvasottként jelölése
/admin/messages/{id}	DELETE	Adminisztrátor (auth:sanctum)	Üzenet törlése

Végpont	Metódus	Kapcsolat típusa	Leírás
/admin/shipping-methods	GET	Adminisztrátor (auth:sanctum)	Szállítási módok listázása
/admin/shipping-methods	POST	Adminisztrátor (auth:sanctum)	Új szállítási mód hozzáadása
/admin/shipping-methods/{id}	PUT	Adminisztrátor (auth:sanctum)	Szállítási mód frissítése
/admin/shipping-methods/{id}	DELETE	Adminisztrátor (auth:sanctum)	Szállítási mód törlése
/admin/shipping-methods/{id}/toggle-active	PUT	Adminisztrátor (auth:sanctum)	Szállítási mód aktív státuszának váltása
/admin/payment-methods	GET	Adminisztrátor (auth:sanctum)	Fizetési módok listázása
/admin/payment-methods	POST	Adminisztrátor (auth:sanctum)	Új fizetési mód hozzáadása
/admin/payment-methods/{id}	PUT	Adminisztrátor (auth:sanctum)	Fizetési mód frissítése
/admin/payment-methods/{id}	DELETE	Adminisztrátor (auth:sanctum)	Fizetési mód törlése
/admin/payment-methods/{id}/toggle-active	PUT	Adminisztrátor (auth:sanctum)	Fizetési mód aktív státuszának váltása
/admin/orders	GET	Adminisztrátor (auth:sanctum)	Admin rendeléslista megtekintése
/admin/orders/{id}	GET	Adminisztrátor (auth:sanctum)	Egy adott rendelés megtekintése adminisztrátor által
/admin/orders/{id}/status	POST	Adminisztrátor (auth:sanctum)	Rendelési státusz frissítése
/admin/order-statuses	GET	Adminisztrátor (auth:sanctum)	Rendelési státuszok listázása
/posts/featured	GET	Nyilvános	Kiemelt blogbejegyzések lekérése
/posts/latest	GET	Nyilvános	Legújabb blogbejegyzések lekérése
/products	GET	Nyilvános	Termékek listázása
/products/{id}	GET	Nyilvános	Egy termék részleteinek lekérése

Végpont	Metódus	Kapcsolat típusa	Leírás
/cart	GET	Nyilvános	Kosár tartalmának lekérése
/cart/items	POST	Nyilvános	Termék hozzáadása a kosárhoz
/cart/items/{id}	PUT	Nyilvános	Kosárban lévő tétel frissítése
/cart/items/{id}	DELETE	Nyilvános	Tétel eltávolítása a kosárból
/cart	DELETE	Nyilvános	Kosár tartalmának törlése
/checkout	POST	Nyilvános	Rendelés leadása
/shipping-methods	GET	Nyilvános	Elérhető szállítási módok lekérése
/payment-methods	GET	Nyilvános	Elérhető fizetési módok lekérése
/newsletter/subscribe	POST	Nyilvános	Feliratkozás a hírlevélre
/newsletter/unsubscribe/{token}	GET	Nyilvános	Leiratkozás a hírlevélről token alapján
/contact	POST	Nyilvános	Kapcsolatfelvételi üzenet küldése
/forgot-password	POST	Nyilvános	Jelszó-visszaállítási link küldése
/reset-password	POST	Nyilvános	Új jelszó beállítása a kapott token alapján

3. Frontend

A frontend az Angular keretrendszerre épül, amely egy modern, komponensalapú architektúrát biztosít a Gyógynövénykereső weboldal és webáruház számára. A felület reszponzív, Bootstrap alapú dizájnnal készült, és támogatja mind a publikus, mind az adminisztrációs funkciókat. Az alkalmazás biztosítja a keresési funkciókat, a kosárkezelést, a rendelések kezelését és a tartalomkezelést, miközben intuitív felhasználói élményt nyújt.

Az alkalmazás struktúrája és alapszerkezete

Az Angular alkalmazás modularizált felépítésű, külön modulokkal az adminisztrációs és publikus felületek számára. A fő komponensek a **src/app** mappában találhatók, és a következő főbb részekre oszlanak:

- **Admin:** Adminisztrációs felület komponensei és szolgáltatásai
- **Webshop:** Webáruházhoz kapcsolódó komponensek (pl. terméklista, kosár, rendelések)
- **Layout:** Általános elrendezési komponensek (fejléc, lábléc, kezdőlap)
- **Shared:** Közös szolgáltatások és segédosztályok
- **Related:** Kapcsolódó tartalmak kezelése (pl. kapcsolódó bejegyzések, termékek)

A fejléc és a keresési funkciók

A fejléc (`header.component`) tartalmazza a navigációs menüt, a keresősávot (`search.component`) és a felhasználói autentikációval kapcsolatos elemeket (`auth.component`). A keresési funkció lehetővé teszi a termékek és blogbejegyzések szűrését név, leírás vagy betegségek alapján. A keresősáv valós idejű szűrést biztosít, és a találatok a `webshop-homepage.component` vagy a `post-single-view.component` komponensekben jelennek meg.

Kezdőlap és termékmegtekintés

A kezdőlap (`home.component`) kiemelt termékeket és blogbejegyzéseket jelenít meg, dinamikusan betöltve az adatokat a backend API-n keresztül. A termékmegtekintés a `product-detail.component` komponensben történik, amely részletes információkat (ár, készlet, képek, leírás) mutat, és lehetőséget biztosít a kosárba helyezésre.

Kosárkezelés

A kosárkezelés a `cart-page.component`, `cart-dropdown.component`, `cart-icon.component` és `cart-item.component` komponensek együttműködésével valósul meg. A `CartService` kezeli a kosár tartalmának szinkronizálását a backenddel. A felhasználók hozzáadhatnak, módosíthatnak vagy törölhetnek termékeket, és valós idejű visszajelzést kapnak a készletinformációkról és az árakról.

Rendelések kezelése

A rendelések kezelését a `orders.component` és `order-details.component` komponensek biztosítják. Az `orders.component` listázza a felhasználó korábbi rendeléseit, míg az `order-details.component` részletes információkat mutat a rendelés státuszáról, termékeiről, szállítási és számlázási adatokról, valamint a rendelési előzményekről.

Webshop főoldal és szűrési funkciók

A webshop főoldala (`webshop-homepage.component`) egy dinamikus terméklistát kínál, amely támogatja a kategóriák szerinti szűrést, keresést és rendezést (pl. ár, név, újdonság szerint). A komponens kiemelt és akciós termékeket is megjelenít egy karuszenen keresztül, és reszponzív dizájnnal alkalmazkodik különböző képernyőméretekhez.

Hitelesítési és adminisztrációs funkciók

A hitelesítést a `auth.component` (felhasználói) és `admin-auth.component` (admin) komponensek kezelik. Az adminisztrációs felületet az `AdminGuard` védi, amely biztosítja, hogy csak jogosult felhasználók férjenek hozzá. Az adminisztrációs funkciók a következőket tartalmazzák:

- Blogbejegyzések és kategóriák kezelése (`posts.component`, `create-post.component`, `categories.component`)
- Termékek és termékkategóriák kezelése (`products.component`, `create-product.component`, `edit-product.component`, `product-categories.component`)
- Rendelések kezelése (`orders.component`, `order-detail.component`)
- Szállítási és fizetési módok kezelése (`shipping-methods.component`, `payment-methods.component`)

- Üzenetek és hírlevél-feliratkozók kezelése (`admin-messages.component`, `subscribers.component`)
- Statisztikák megtekintése (`product-statistics.component`)

Jelszó-visszaállítási komponensek

A felhasználói jelszó-visszaállítás kezelésére két dedikált komponens szolgál:

1. **ForgotPasswordComponent** (`layout/header/auth/forgot-password/forgot-password/`):

- E-mail cím bekérése a jelszó-visszaállításhoz
- Validációs ellenőrzések (kötelező mező, érvényes e-mail formátum)
- Kérés küldése a backend API-nak
- Sikeres kérés esetén megerősítő képernyő megjelenítése
- Hibaüzenetek kezelése (pl. nem létező e-mail, túl sok kérés)
- Bejelentkezéshez visszatérés lehetősége

2. **ResetPasswordComponent** (`layout/header/auth/forgot-password/reset-password/`):

- URL paraméterek feldolgozása (token és e-mail cím)
- Új jelszó és megerősítés bekérése
- Validációs ellenőrzések (min. 8 karakter, nagybetű, kisbetű, szám)
- Token és új jelszó elküldése a backendnek
- Sikeres jelszóváltoztatás esetén megerősítés és bejelentkezési lehetőség
- Hibaüzenetek kezelése (érvénytelen token, gyenge jelszó)

Mindkét komponens összhangban van az alkalmazás általános dizájnjával, és Bootstrap-alapú stílusokat használ a felhasználói élmény javítására, beleértve a kártyaelemeket, animációkat és reszponzív megjelenést.

A jelszó-visszaállítási folyamat további funkcióit a `ModalService` teszi lehetővé, amely a bejelentkező modalt nyitja meg, amikor a felhasználó vissza szeretne térni a bejelentkezéshez. Az `AuthService` szolgáltatás biztosítja a backend API-val való kommunikációt, beleértve a jelszó-visszaállítási kéréseket és a jelszófrissítést.

Kapcsolódó tartalmak kezelése

A kapcsolódó tartalmakat a `related-posts.component` és `related-products.component` komponensek kezelik. Ezek a komponensek a `RelatedContentService` segítségével dinamikusan töltik be a releváns blogbejegyzéseket vagy termékeket a felhasználó aktuális megtekintése alapján.

Komponensek és osztályváltozók listája

- **AdminAuthComponent**: Admin bejelentkezés kezelése (`loginForm`, `submitted`)
- **CategoriesComponent**: Blogkategóriák kezelése (`categories`, `categoryForm`, `successMessage`, `errorMessage`)
- **DashboardComponent**: Adminisztrációs műszerfal
- **SubscribersComponent**: Hírlevél-feliratkozók kezelése (`subscribers`, `loading`, `error`, `successMessage`, `showDeleteConfirmation`, `subscriberToDelete`)

- **OrdersComponent:** Felhasználói rendelések listázása (`orders`, `isLoading`, `error`)
- **OrderDetailComponent:** Rendelési részletek (`order`, `isLoading`, `error`, `orderStatuses`, `standardStatuses`, `storageUrl`)
- **WebshopHomepageComponent:** Webshop főoldal (`products`, `filteredProducts`, `categories`, `featuredProducts`, `discountedProducts`, `loading`, `storageUrl`, `selectedCategory`, `searchTerm`, `sortOption`, `toastMessage`)
- **ForgotPasswordComponent:** Jelszó-visszaállítás kezdeményezése (`forgotPasswordForm`, `loading`, `submitted`, `emailSent`, `toastMessage`, `error`)
- **ResetPasswordComponent:** Új jelszó beállítása (`resetForm`, `loading`, `submitted`, `resetComplete`, `token`, `email`, `toastMessage`, `error`)

Frontend projekt struktúra

A frontend Angular alapú, a következő főbb könyvtárszerkezettel:

```

frontend/
├── public/                                # Nyilvános statikus fájlok
├── src/
│   ├── app/                              # Az alkalmazás fő könyvtára
│   │   ├── admin/                        # Adminisztrációs komponensek
│   │   │   ├── admin-auth/              # Admin bejelentkezés
│   │   │   ├── categories/              # Blogkategóriák kezelése
│   │   │   ├── dashboard/              # Admin műszerfal
│   │   │   │   ├── subscribers/         # Hírlevél-feliratkozók
│   │   │   ├── messages/              # Üzenetek kezelése
│   │   │   ├── orders/                 # Rendelések kezelése
│   │   │   │   ├── order-detail/        # Rendelési részletek
│   │   │   ├── payment-methods/        # Fizetési módok
│   │   │   ├── posts/                  # Blogbejegyzések
│   │   │   │   ├── create-post/         # Új bejegyzés létrehozása
│   │   │   ├── product-categories/     # Termékkategóriák
│   │   │   ├── product-statistics/     # Termékstatisztikák
│   │   │   ├── products/              # Termékek kezelése
│   │   │   │   ├── create-product/     # Új termék
│   │   │   │   ├── edit-product/       # Termék szerkesztése
│   │   │   ├── services/              # Adminisztrációs szolgáltatások
│   │   │   ├── shipping-methods/       # Szállítási módok
│   │   │   └── users/                  # Felhasználók kezelése
│   │   ├── guards/                     # Hozzáférési védelmek
│   │   ├── layout/                     # Elrendezési komponensek
│   │   │   ├── content/                # Tartalmi komponensek
│   │   │   │   ├── detail/             # Részletes nézetek
│   │   │   ├── footer/                 # Lábléc
│   │   │   │   ├── pages/              # Statikus oldalak
│   │   │   ├── header/                 # Fejléc
│   │   │   │   ├── auth/               # Autentikáció
│   │   │   │   │   ├── forgot-password/ # Jelszó-visszaállító komponensek
│   │   │   │   │   │   ├── forgot-password/ # E-mail bekérő komponens
│   │   │   │   │   └── reset-password/ # Új jelszó megadása komponens
│   │   └── header/                     # Navigáció

```

└─ search/	# Keresés
└─ home/	# Kezdőlap
└─ related/	# Kapcsolódó tartalmak
└─ related-posts/	# Kapcsolódó bejegyzések
└─ related-products/	# Kapcsolódó termékek
└─ shared/	# Közös erőforrások
└─ services/	# Szolgáltatások
└─ auth.service.ts	# Felhasználói autentikáció
szolgáltatás	
└─ modal.service.ts	# Modal kezelő szolgáltatás
└─ toast.service.ts	# Értesítések szolgáltatás
└─ webshop/	# Webáruház komponensei
└─ cart/	# Kosárkezelés
└─ cart-dropdown/	# Kosár legördülő
└─ cart-icon/	# Kosár ikon
└─ cart-item/	# Kosártétel
└─ cart-page/	# Kosár oldal
└─ checkout/	# Pénztár
└─ discounted-products/	# Akciós termékek
└─ featured-products/	# Kiemelt termékek
└─ order-details/	# Rendelési részletek
└─ orders/	# Rendelések
└─ webshop-homepage/	# Webshop főoldal
└─ app.component.*	# Alkalmazás gyökérkomponens
└─ app.config.ts	# Alkalmazás konfiguráció
└─ app.routes.ts	# Útvonalak definíciója
└─ assets/	# Statikus erőforrások
└─ img/	# Képek
└─ environments/	# Környezeti változók
└─ environment.development.ts	# Fejlesztői környezet
└─ environment.ts	# Alapértelmezett környezet
└─ index.html	# Fő HTML fájl
└─ main.ts	# Alkalmazás belépési pont
└─ styles.css	# Globális stílusok
└─ storage/	# Tárolt fájlok (pl. képek)
└─ .editorconfig	# Szerkesztő konfiguráció
└─ .gitignore	# Git figyelmen kívül hagyás
└─ angular.json	# Angular CLI konfiguráció
└─ package.json	# Függőségek és szkriptek
└─ README.md	# Projekt leírás
└─ tsconfig.app.json	# TypeScript konfiguráció (app)
└─ tsconfig.json	# TypeScript konfiguráció (általános)
└─ tsconfig.spec.json	# TypeScript konfiguráció (tesztek)

Fontos frontend fájlok

1. .editorconfig

- Egységesíti a kódformázást (pl. indentálás: 2 szóköz, UTF-8 kódolás).

2. .gitignore

- Kihagyja a generált fájlokat (pl. `node_modules`, `dist`) a verziókövetésből.

3. `angular.json`

- Az Angular CLI konfigurációs fájlja, meghatározza a build, szerver és tesztelési beállításokat.
- Tartalmazza a statikus erőforrások (pl. `favicon`, `assets`) és stílusok (Bootstrap, globális CSS) kezelését.

4. `package.json`

- Felsorolja a függőségeket (pl. Angular, Bootstrap, Chart.js) és fejlesztési szkripteket (pl. `ng serve`, `ng build`).

5. `README.md`

- Leírja a projekt telepítését, fejlesztői szerver indítását, buildelését és tesztelését.

6. `environments/environment.development.ts`

- Fejlesztői környezet beállításai (pl. API URL: `http://localhost:8000/api`).

7. `index.html`

- Az alkalmazás belépési pontja, betölti a Bootstrap Icons-t és az Angular gyökérkomponenst.

8. `main.ts`

- Az Angular alkalmazás bootstrapelése.

9. `styles.css`

- Globális stílusok, importálja a Bootstrapet és meghatározza az alapvető elrendezést (pl. `padding`, `min-height`).

10. `app.routes.ts`

- Az útvonalak definíciója, beleértve a jelszó-visszaállítási útvonalakat is:

```
{ path: 'forgot-password', component: ForgotPasswordComponent },  
{ path: 'reset-password', component: ResetPasswordComponent }
```

Adminisztrációs felület komponensei

Az adminisztrációs felület a `/admin` útvonalon érhető el, és az `AdminModule` modulba szervezett komponensekből áll. A főbb komponensek:

- **AdminAuthComponent** (`admin-auth/`): Kezeli a Szuper Admin bejelentkezést, validálja az emailt és jelszót, és token alapú autentikációt használ.
- **DashboardComponent** (`dashboard/`): Az adminisztrációs műszerfal, amely navigációs menüt biztosít a különböző admin funkciókhoz.

- **CategoriesComponent** ([categories/](#)): Blogkategóriák létrehozása, listázása és törlése. Toast üzenetekkel jelzi a sikeres műveleteket.
- **SubscribersComponent** ([dashboard/subscribers/](#)): Hírlevél-feliratkozók kezelése, aktiválás/deaktiválás és törlés, responzív táblázattal és kártyanézetrel.
- **OrdersComponent** ([orders/](#)): Adminisztrátori rendelések listázása és kezelése.
- **OrderDetailComponent** ([orders/order-detail/](#)): Egy rendelés részletes nézete, státuszfrissítési lehetőséggel.
- **ProductsComponent** ([products/](#)): Termékek listázása, létrehozása és szerkesztése.
- **ProductCategoriesComponent** ([product-categories/](#)): Termékkategóriák kezelése.
- **ShippingMethodsComponent** ([shipping-methods/](#)): Szállítási módok kezelése.
- **PaymentMethodsComponent** ([payment-methods/](#)): Fizetési módok kezelése.
- **ProductStatisticsComponent** ([product-statistics/](#)): Termékekkel kapcsolatos statisztikák (pl. eladások, megtekintések).

Webshop komponensei

A webáruház funkcionalitását a [webshop/](#) mappában található komponensek biztosítják:

- **WebshopHomepageComponent** ([webshop-homepage/](#)): A webshop főoldala, amely akciós és kiemelt termékeket mutat karuszenen, valamint szűrhető terméklistát kínál. Támogatja a kategóriaválasztást, keresést és rendezést.
 - **OrdersComponent** ([orders/](#)): Felhasználói rendelések listázása táblázatos formában, részletek linkkel.
 - **OrderDetailsComponent** ([order-details/](#)): Egy rendelés részletes nézete, amely mutatja a státuszt, szállítási/fizetési adatokat, termékeket és rendelési előzményeket. Progress bar jelzi a rendelés előrehaladását.
 - **CartPageComponent** ([cart/cart-page/](#)): A kosár tartalmának részletes nézete, módosítási és törlési lehetőségekkel.
 - **CheckoutComponent** ([checkout/](#)): A rendelés leadásának felülete, számlázási és szállítási adatok megadásával.
 - **DiscountedProductsComponent** ([discounted-products/](#)): Akciós termékek kiemelése.
 - **FeaturedProductsComponent** ([featured-products/](#)): Kiemelt termékek megjelenítése.
-