A képen clipart, macska, illusztráció, rajzfilm látható

Automatikusan generált leírás

**Széchenyi István Katolikus Technikum és Gimnázium**

**Szoftverfejlesztő és -tesztelő projektfeladat**

**„Airsoft-marketplace”**

Készítették:

  Kovács Martin

Kovács Ákos

**Ózd, 2025**

**Bevezetés**

A projektünk célja egy olyan airsoft piaci webalkalmazás kidolgozása, amely hatékony megoldást nyújt arra, hogy az emberek könnyebben hozzájuthassanak kiegészítőikhez, fegyvereikhez. Fegyverek alatt a pisztolyokat, géppisztolyokat, illetve gépkarabélyokat értjük, kiegészítők alatt pedig például sisakot és mellényt.

A digitális világ fejlődése az online piacterek népszerűségének növekedéséhez vezetett, így a különböző közösségek számára létrehozott platformok egyre nagyobb szerepet játszanak a termékek vásárlásában és eladásában. Az airsoft-közösség is növekszik, a játékosok pedig egyre nagyobb igényt támasztanak egy olyan online piactérre, amely lehetőséget biztosít airsoft eszközök, felszerelések és tartozékok gyors és egyszerű vásárlására és eladására.

Mint minden ember, akinek van egy hobbija vagy versenyszerűen űz egy sportot, mindig szeretne fejlődni benne, ami azzal jár, hogy fejlesztenie kell eszközeit. Ez a webalkalmazás erre nyújt megoldást az airsoft közösségben. Kezdők meg tudják venni a haladók eszközeit, a haladók pedig tudják értékesíteni a használt, de jó állapotban lévő fegyvereiket, kiegészítőiket.

Gtihub link: https://github.com/kovacsakos05/airsoftmarketplace-public.git

**Felhasználói dokumentáció**

**Rendszerkövetelmény**

**Minimális gépigény**

|  |  |
| --- | --- |
| Processzor (CPU) | 2 magos, 2.0 GHz vagy gyorsabb |
| Memória (RAM) | 4 GB |
| Háttértár | 10 GB szabad hely |
| Operációs rendszer | Windows 10 / macOS Mojave / Ubuntu 18.04 vagy újabb |
| Böngésző | Chrome 90+ / Firefox 88+ |
| További követelmények | Node.js 16+, NPM 8+, PHP 8+, Composer, MySQL vagy MariaDB |

**Ajánlott gépigény**

|  |  |
| --- | --- |
| Processzor (CPU) | 4 magos, 3.0 GHz vagy gyorsabb |
| Memória (RAM) | 8 GB vagy több |
| Háttértár | 20 GB szabad hely |
| Operációs rendszer | Windows 11 / macOS Ventura / Ubuntu 22.04 |
| Böngésző | Chrome 110+ / Firefox 105+ |
| További követelmények | Node.js 18+, PHP 8.2+, Composer 2+, MySQL 8.0 vagy MariaDB 10.6 |

**Webalkalmazás indítása**

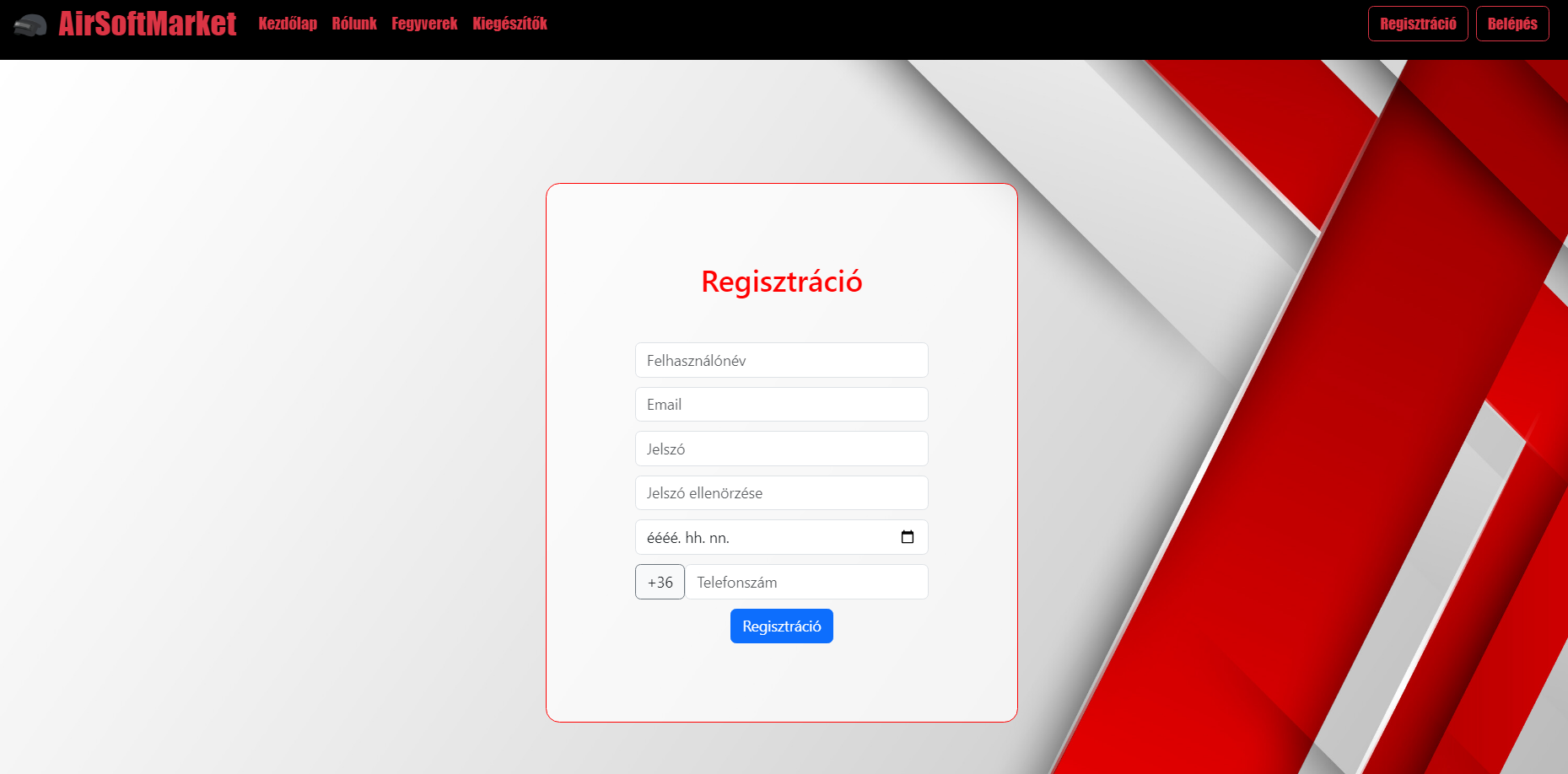
Először az adatbázisba fel kell tölteni a szükséges adatokat. A Visual Studio Code alkalmazáshoz hozzá kell adni az Angulart az **npm install -g @angular/cli** paranccsal, valamint a **NodeJs**-t is le kell töltenünk hozzá. Ha ezzel megvagyunk, a frontend-et az “ng serve” utasítással, a backend-et pedig a “php artisan serve” utasítással hozhatjuk működésbe.

**Webalkalmazás használata**

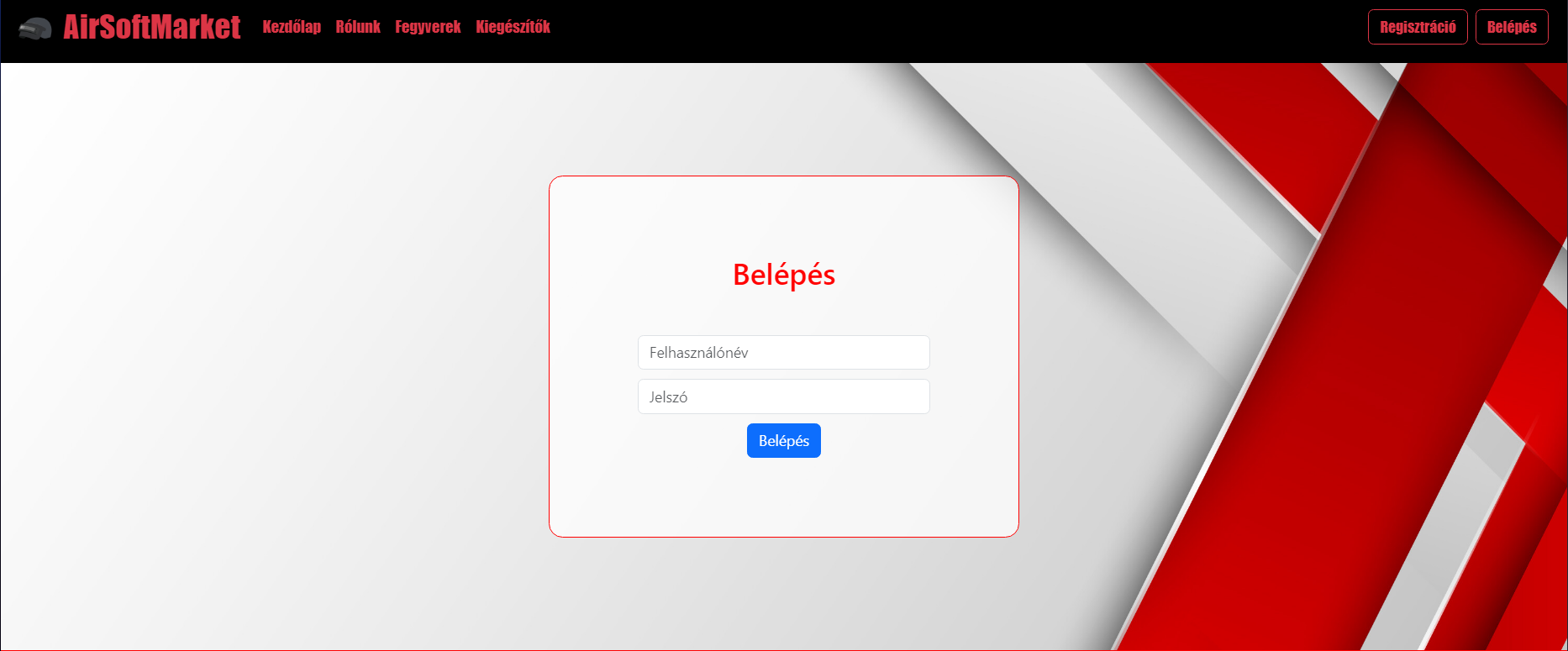
A használat megkezdése után rögtön a kezdőlapon találjuk magunkat, ahol leírjuk, hogy mit is nyújt ez a webalkalmazás. A fejlécben több menüelem is található. Bal oldalon négy, a “Kezdőlap”, a “Rólunk”, a “Fegyverek” és a “Kiegészítők”, jobb oldalon pedig a “Regisztráció”, illetve a “Bejelentkezés”. A regisztrációra kattintva létre hozhatunk egy saját felhasználó fiókot, ha ez sikerült akkor be tudunk jelentkezni és a főoldalra navigál minket az alkalmazás. A “Fegyverek” és a “Kiegészítők” fülre kattintva megnézhetjük a már fent lévő airsoft eszközöket. A saját darabok feltöltése úgy néz ki, hogy a jobb felső sarokban, a felhasználónevünkre kell rákattintani, ott kiválasztani a termékfeltöltést, utána pedig eldönteni, hogy fegyvert vagy kiegészítőt akarunk értékesíteni. Kiegészítő és fegyver esetén is szükséges 3 képet feltölteni az adott eszközről. Amikor minden beviteli mezőt kitöltöttünk, nincs más hátra, mint a feltöltés.



1. kép: Kezdőlap



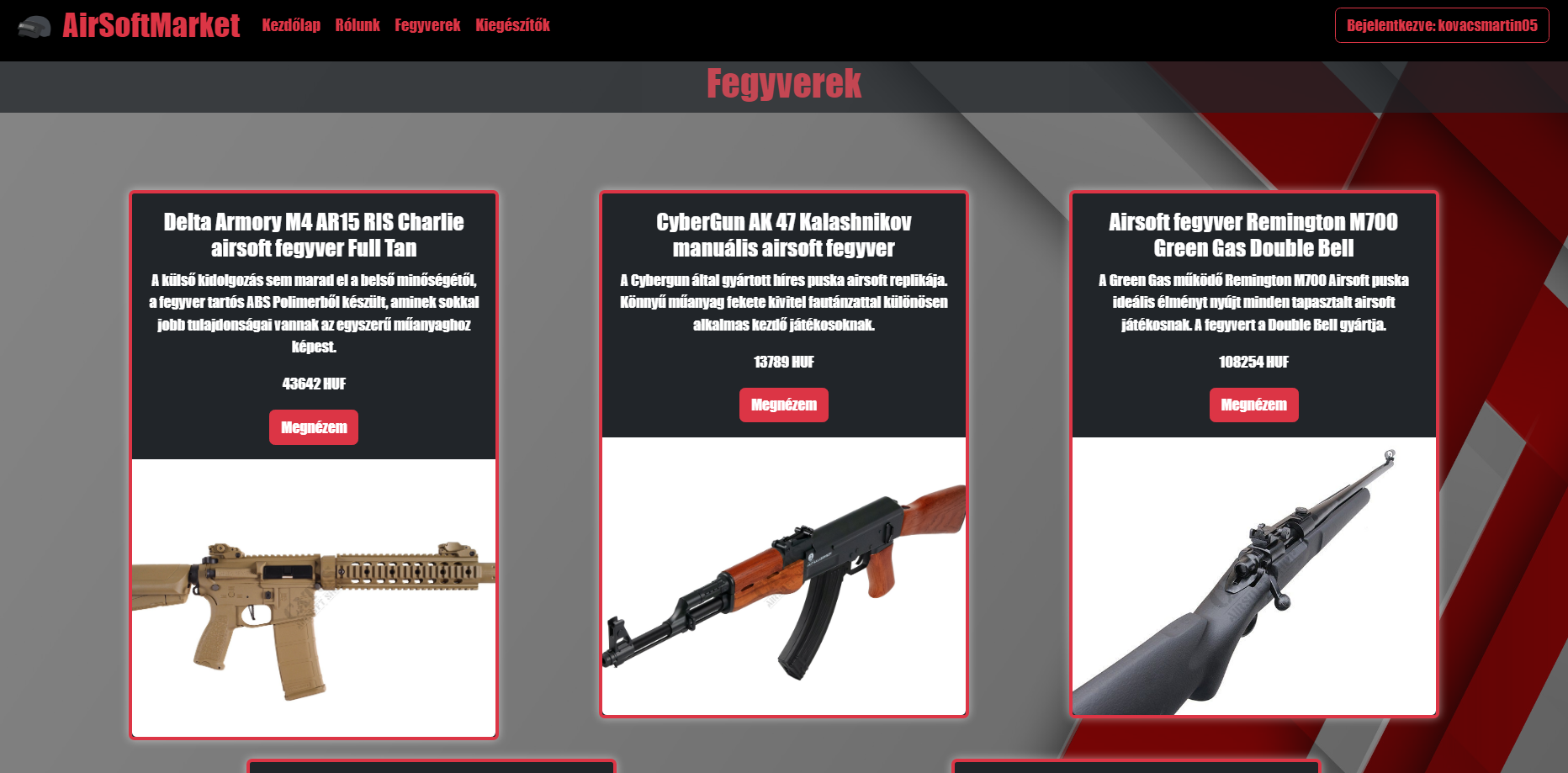
2. kép: Regisztráció



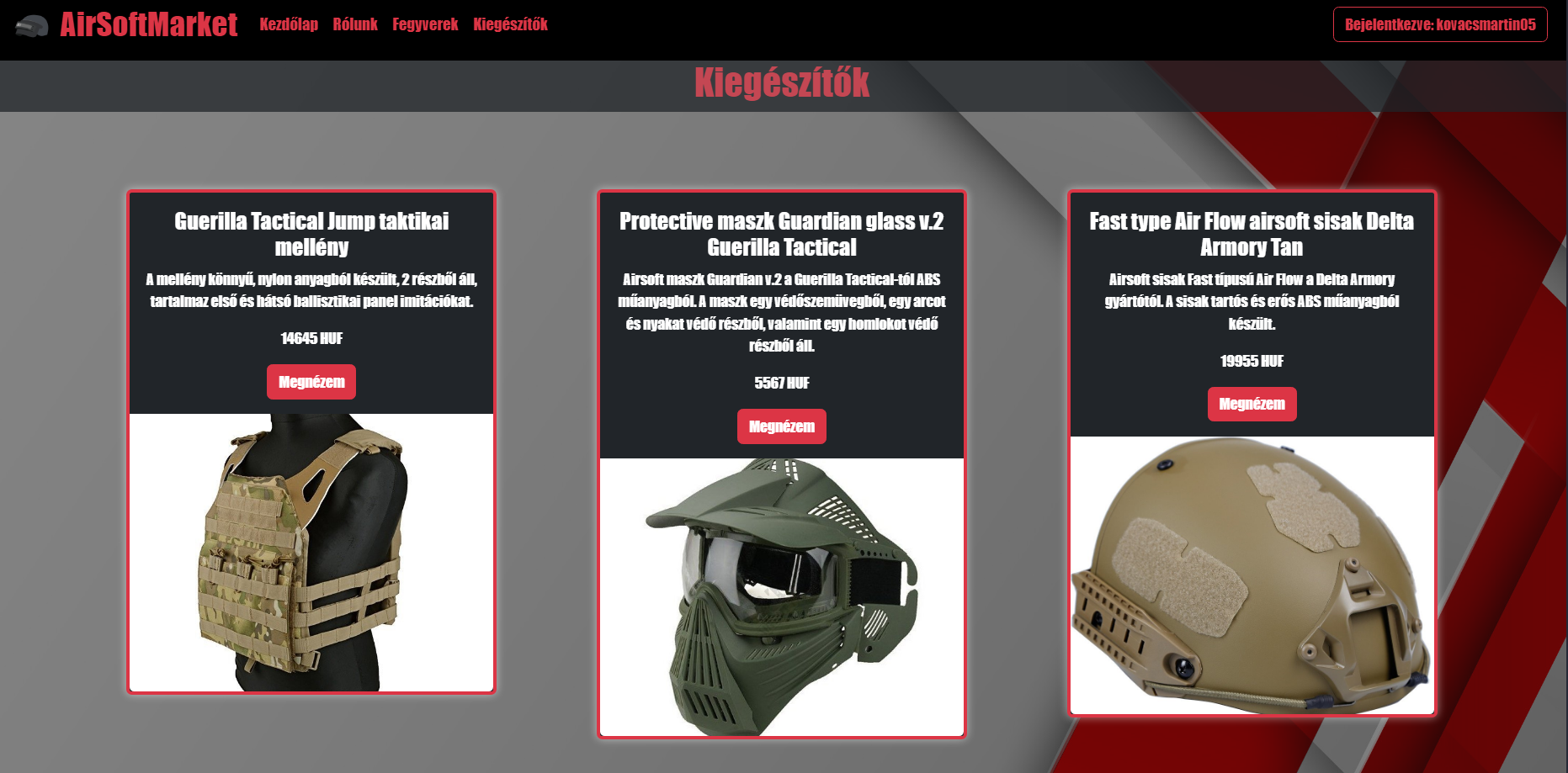
3. kép: Belépés



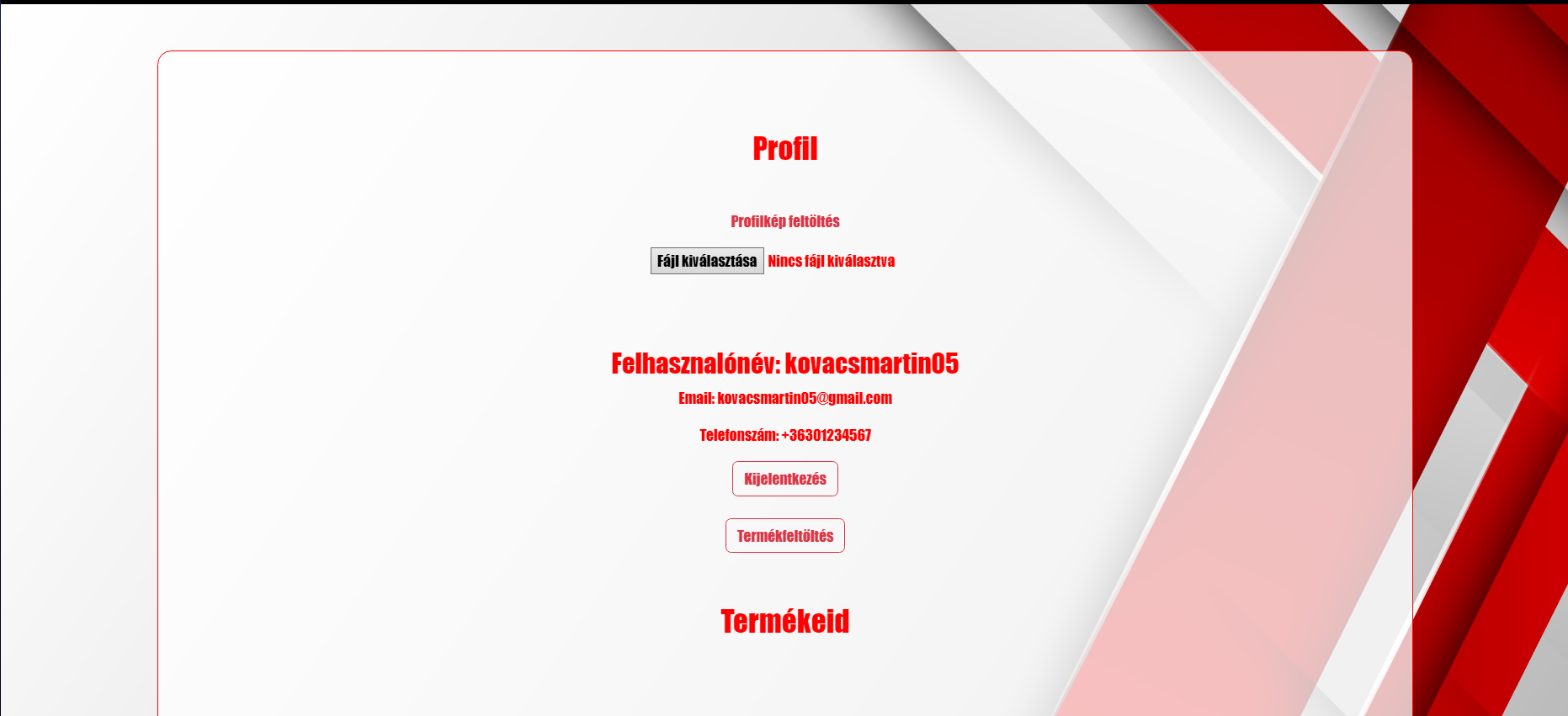
4. kép: Rólunk



5. kép: Fegyverek



6. kép: Kiegészítők



7. kép: Profil



8. kép: Termék feltöltés (kiegészítő)

**Fejlesztői dokumentáció**

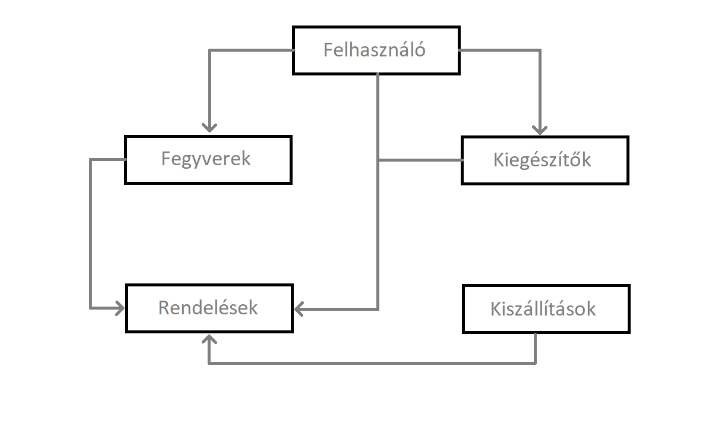
**Alkalmazott fejlesztői és csoportmunka eszközök**: A csoportmunka eszközeit online térben oldottuk meg. A kommunikáció online hívásban, a Discord alkalmazásban, a fájlok átadása és tárolása pedig Google Drive alkalmazásban történt meg. A webalkalmazás stílusát a figma.com weboldalon terveztük meg. Amit a csapatmunkatervező szoftver, amit használtunk, a Trello.

**Adatbázis**: Az adatbázist MYSQL-ben terveztük meg és készítettük el, phpmyadmin-ban.

**Frontend, backend**: A frontend és backend kódját a Visual Studio Code alkalmazásban készítettük el. Az előbbit Angular keretrendszerben Typescript, HTML, CSS és Bootstrap segítségével, az utóbbit pedig Laravel keretrendszerben készítettük el PHP programnyelven.

**Adatbázis**

Az adatbázisban 5 tábla található a felhasználó, a rendelés, a fegyverek, a kiegészítők és a kiszállítás. Három tábla kapcsolódik a rendelés táblához, amely a rendeléshez szükséges adatokat tárolja el. A felhasználó tábla a felhasználó regisztrációja során megadott adatokat tárolja el. A fegyverek és kiegészítők tábla tartalmazza az adott termék adatait és idegenkulcs mezőjében a tulajdonos felhasználójának azonosítóját. A kiszállítások tábla pedig a kiszállítás módját és árát tartalmazza



9. kép: Adatbázis táblái

**Felhasználó adattábla**

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Típus** | **Kötelező mező** | **Adat** |
| id | int | igen | azonosító |
| nev | varchar | igen | felhasználónév |
| email | varchar | igen | email |
| jelszo | varchar | igen | jelszó |
| telefonszam | varchar | igen | telefonszám |
| profilkep | varchar | nem | profilkép elérési útja |
| szuldatum | date | igen | születési dátum |

**Fegyverek adattábla**

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Típus** | **Kötelező mező** | **Adat** |
| id | int | igen | azonosító |
| nev | varchar | igen | fegyver neve |
| tipus | varchar | igen | fegyver típusa |
| ar | int | igen | fegyver ára |
| leiras | varchar | igen | fegyver leírása |
| mechanika | varchar | igen | fegyver mechanikája |
| fps | int | igen | fegyver erőssége |
| fegyverek\_img1 | varchar | igen | fegyver képeinek elérési útja |
| fegyverek\_img2 | varchar | igen | fegyver képeinek elérési útja |
| fegyverek\_img3 | varchar | igen | fegyver képeinek elérési útja |
| suly | int | igen | fegyver súlya |
| hossza | int | igen | fegyver hossza |
| csohossz | int | nem | fegyver csőhossza |
| tuzmod | varchar | igen | fegyver tűzmódja |
| tulajdonos | int(idegen kulcs) | igen | fegyever tulajdonosa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Kiegészítők adattábla**

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Típus** | **Kötelező mező** | **Adat** |
| id | int | igen | azonosító |
| nev | varchar | igen | kiegészítő neve |
| tipus | varchar | igen | kiegészítő típusa |
| ar | int | igen | kiegészítő ára |
| leiras | varchar | igen | kiegészítő leírása |
| kiegeszitok\_img | varchar | igen | kiegészítő képeinek elérési útja |
| kiegeszitok\_img2 | varchar | igen | kiegészítő képeinek elérési útja |
| kiegeszitok\_img3 | varchar | igen | kiegészítő képeinek elérési útja |
| tulajdonos | int(idegen kulcs) | igen | kiegészítő tulajdonosa |

**Kiszállítás adattábla**

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Típus** | **Kötelező mező** | **Adat** |
| id | int | igen | azonosító |
| kiszallitas\_fajtaja | varchar | igen | kiszállítás fajtája |
| kiszallitas\_ara | varchar | igen | kiszállítás költsége |

**Rendelés adattábla**

|  |  |  |  |
| --- | --- | --- | --- |
| **Név** | **Típus** | **Kötelező mező** | **Adat** |
| id | int | igen | azonosító |
| fegyverek\_id | int(idegenkulcs) | nem | fegyver(ek) azonosítója |
| kiegeszitok\_id | int(idegenkulcs) | nem | kiegészítő(k) azonosítója |
| felhasznalok\_id | int(idegenkulcs) | igen | felhasználó azonosítója |
| keresztnev | varchar | igen | megrendelő keresztneve |
| vezeteknev | varchar | igen | megrendelő vezetékneve |
| iranyitoszam | int | igen | megrendelő irányítószáma |
| varos | varchar | igen | megrendelő települése (ahová szeretné rendelni a terméket) |
| utca\_hazszam | varchar | igen | megrendelő otthona (ahová szeretné rendelni a terméket) |
| telefonszam | varchar | igen | megrendelő telefonszáma |
| kiszallitas | int(idegenkulcs) | igen | kiszállítás fajtája |

**Frontend**

Az alkalmazás felhasználói felülete Angular keretrendszerben készült, és modern, komponens-alapú struktúrával dolgozik. Az egyes oldalak funkcionálisan jól elkülönülő Angular-komponensekbe vannak szervezve, amelyek különböző típusú termékek (fegyverek, kiegészítők stb.) kezelésére szolgálnak. A frontend teljes egészében reszponzív, Bootstrap

alapú dizájnelemeket alkalmaz, és az animációk révén vizuálisan is élvezhető felhasználói élményt biztosít.

**A frontend komponensei**

**home komponens – Kezdőoldal**

A home komponens az alkalmazás nyitóoldalaként szolgál, és célja, hogy bemutassa a platform fő szolgáltatásait, valamint ízelítőt adjon az elérhető termékekből. A komponens két fő szakaszra oszlik: egy bevezető blokkra és egy termékajánló részre.

A bevezető szekcióban nagy méretű címsor (AirSoftMarket) és egy rövid leírás jelenik meg, amely elmagyarázza a weboldal célját – egy közösségi alapú adok-veszek piactér az airsoft közösség számára. Az oldal felhívja a figyelmet arra, hogy a felhasználók saját termékeket tölthetnek fel, valamint böngészhetnek mások ajánlatai között is. A leírás esztétikus környezetben, középre igazítva jelenik meg, és egy “Termékek” gomb navigálja a látogatót a következő szakaszra.

A második részben két szekcióban, görgetéshez kötött animációkkal (egyéni animation() metódusok meghívásával) jelennek meg először a fegyverek, majd a kiegészítők. Mindkét kategóriában maximum három termék kerül megjelenítésre (slice:0:3), amely vizuálisan vonzó, kártyaszerű elrendezésben történik. Minden termék képpel, névvel, rövid leírással, valamint egy "Megnézem" gombbal rendelkezik, amely routerLink segítségével navigál a részletes termékoldalra (/ad-guns/:id és /ad-supplementarys/:id).

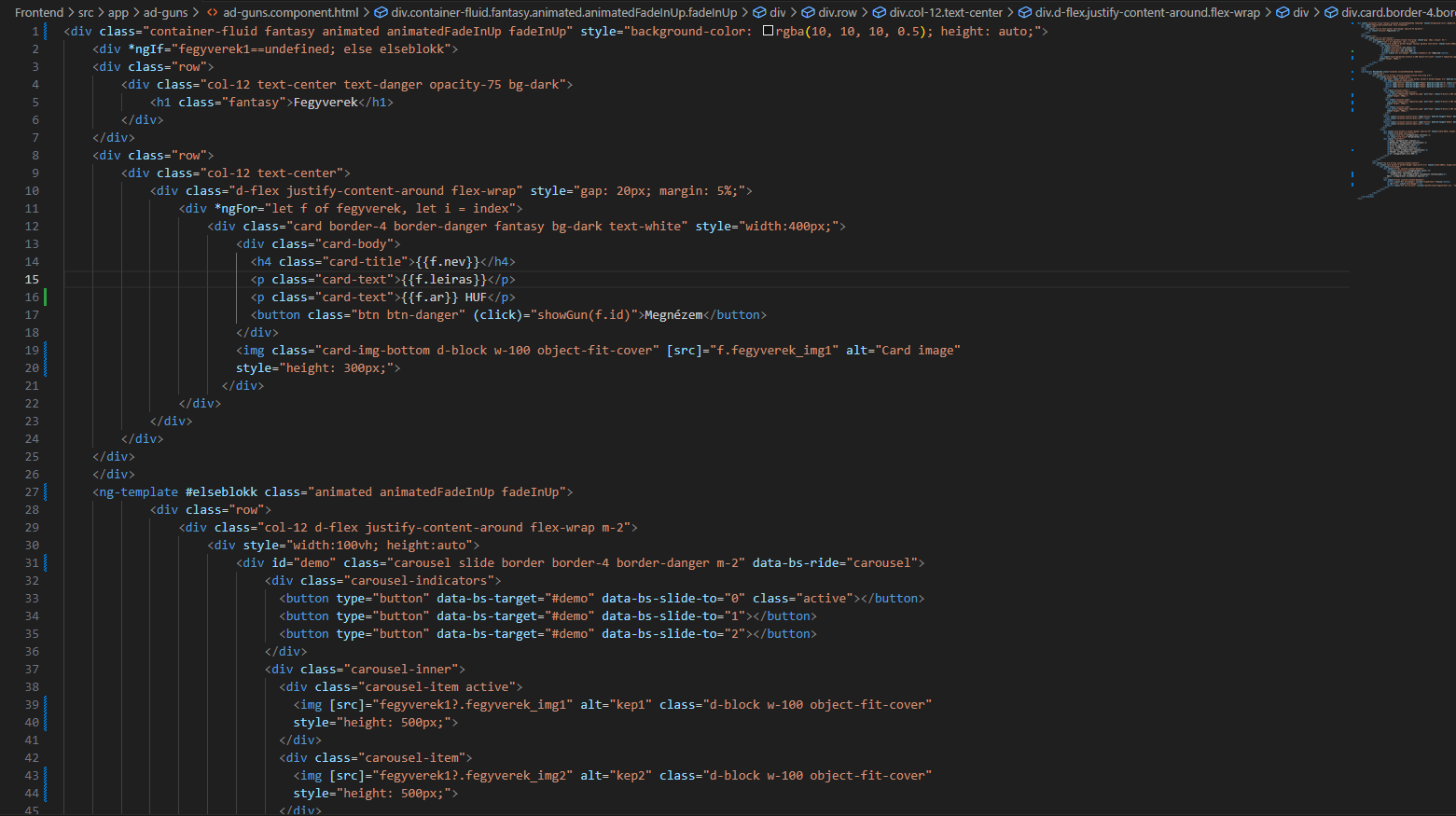
Ez a komponens egyértelmű belépési pontot nyújt az alkalmazásba, informálja a látogatót a lehetőségekről, és vizuálisan is ízelítőt kínál a legfrissebb termékekből. A görgetéshez rendelt animációk fokozzák a dinamizmust és a modern érzetet.

**ad-guns komponens**

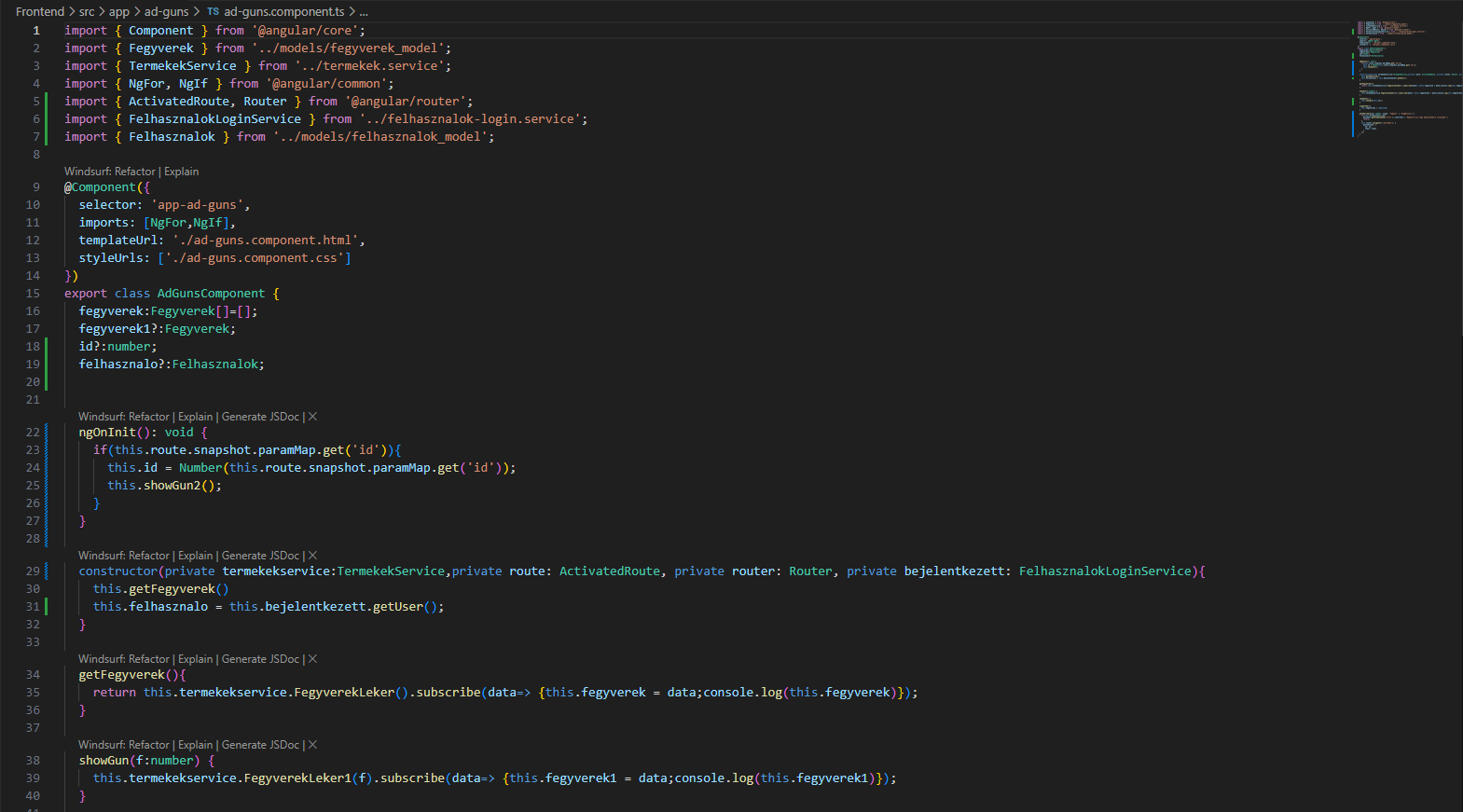
Az ad-guns komponens felelős a fegyverek listázásáért, valamint azok részletes megjelenítéséért. Két fő nézetet kezel: a terméklista megjelenítését, illetve a kiválasztott fegyver részletes adatait tartalmazó nézetet. A komponens betöltése során, amennyiben nincs kiválasztva konkrét termék, megjelenik az összes elérhető fegyver kártyás elrendezésben. A kártyák mindegyike tartalmazza a fegyver nevét, rövid leírását, árát, valamint egy képet, amelyet a backendről betöltött elérési útvonal alapján jelenít meg.

A felhasználó a “Megnézem” gomb megnyomásával válthat át a részletes nézetre. Ilyenkor az adott fegyver három képe egy Bootstrap-carousel komponensbe töltődik be, ahol a felhasználó lapozhat a különböző képek között. A részletes nézetben megjelennek a termék műszaki paraméterei is, például a típus, mechanika, erősség (FPS), súly, hossz, csőhossz és tűzmód. Az oldal alján a tulajdonos elérhetőségei is láthatók, ami közvetlen kommunikációt is lehetővé tehet.

A komponens interaktív elemei között szerepel a „Vissza” gomb, amellyel a felhasználó visszatérhet a terméklistához, illetve a „Megveszem” gomb, amely a goToPurchase() metódus segítségével továbbítja a vásárlási szándékot egy másik nézet vagy folyamat felé.



10. kép: ad-guns.component.html

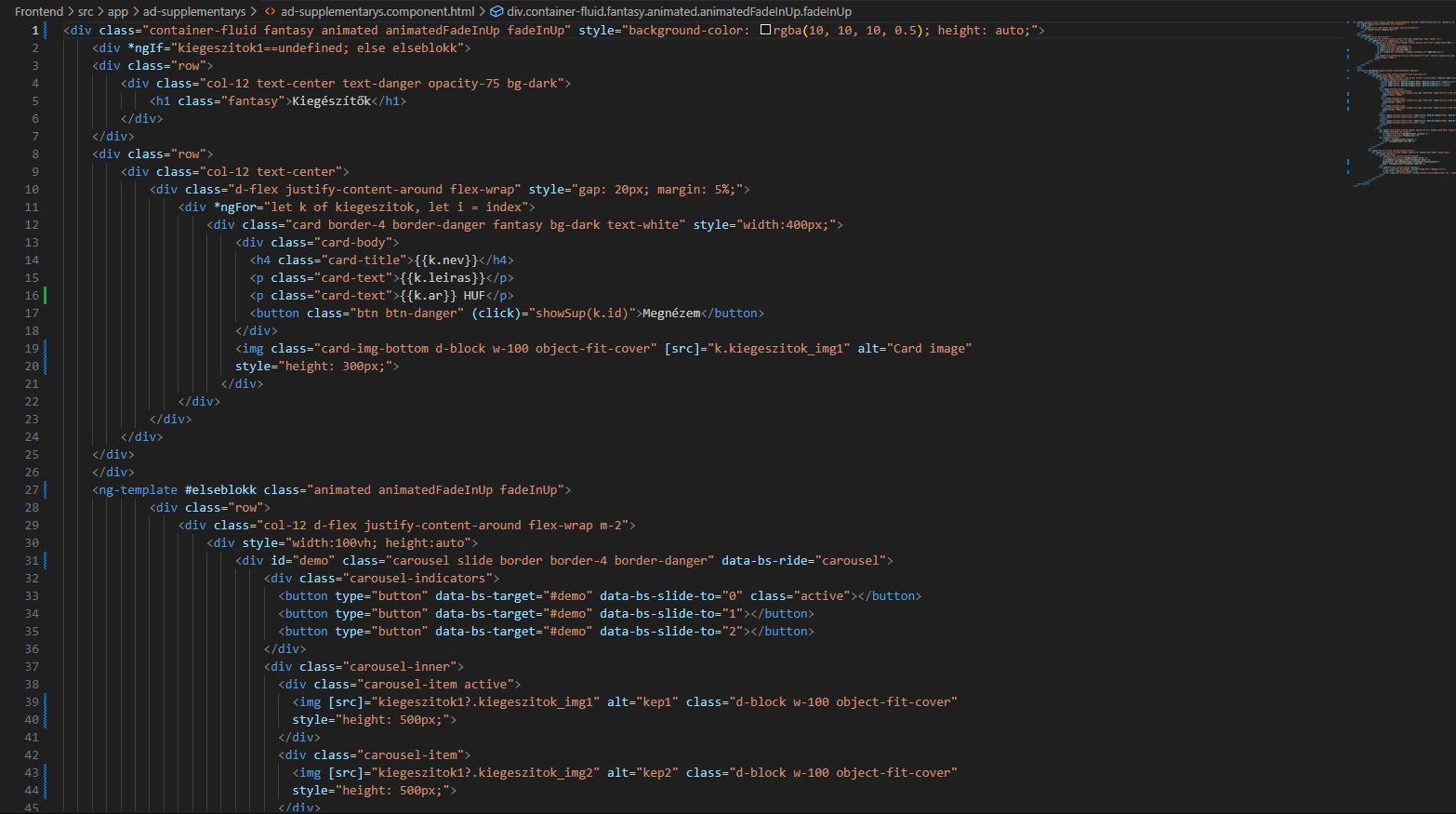


11. kép: ad-guns.component.ts

**ad-supplementarys komponens**

A ad-supplementarys komponens hasonló logikával épül fel, mint a fegyvereket kezelő társa, de ez a kiegészítő termékeket jeleníti meg. Az oldal első betöltésekor szintén egy lista nézet jelenik meg, ahol a rendszer az összes elérhető kiegészítőt sorolja fel kártyák formájában. A kártyák tartalmazzák a termék nevét, leírását, árát, valamint az első képet. A vizuális kialakítás itt is sötét, military-hangulatú stílusú, piros keretekkel és animációs áttűnésekkel.

A felhasználó egy adott termék részletes nézetéhez a „Megnézem” gombra kattintva jut el, amely egy képgalériát és a termék részletes jellemzőit jeleníti meg. Ezek között megtalálható a termék típusa és ára, valamint a képek közötti lapozás lehetősége carousel elemen keresztül. A nézet alsó részén a termék tulajdonosának elérhetőségeit tünteti fel a rendszer, hasonló módon, mint a fegyverek esetében. A „Vissza” gomb visszaviszi a felhasználót a listához, míg a „Megveszem” gomb egy vásárlási folyamatot indít el a megfelelő azonosítóval és típussal.



12. kép: ad-supplementarys.component.html



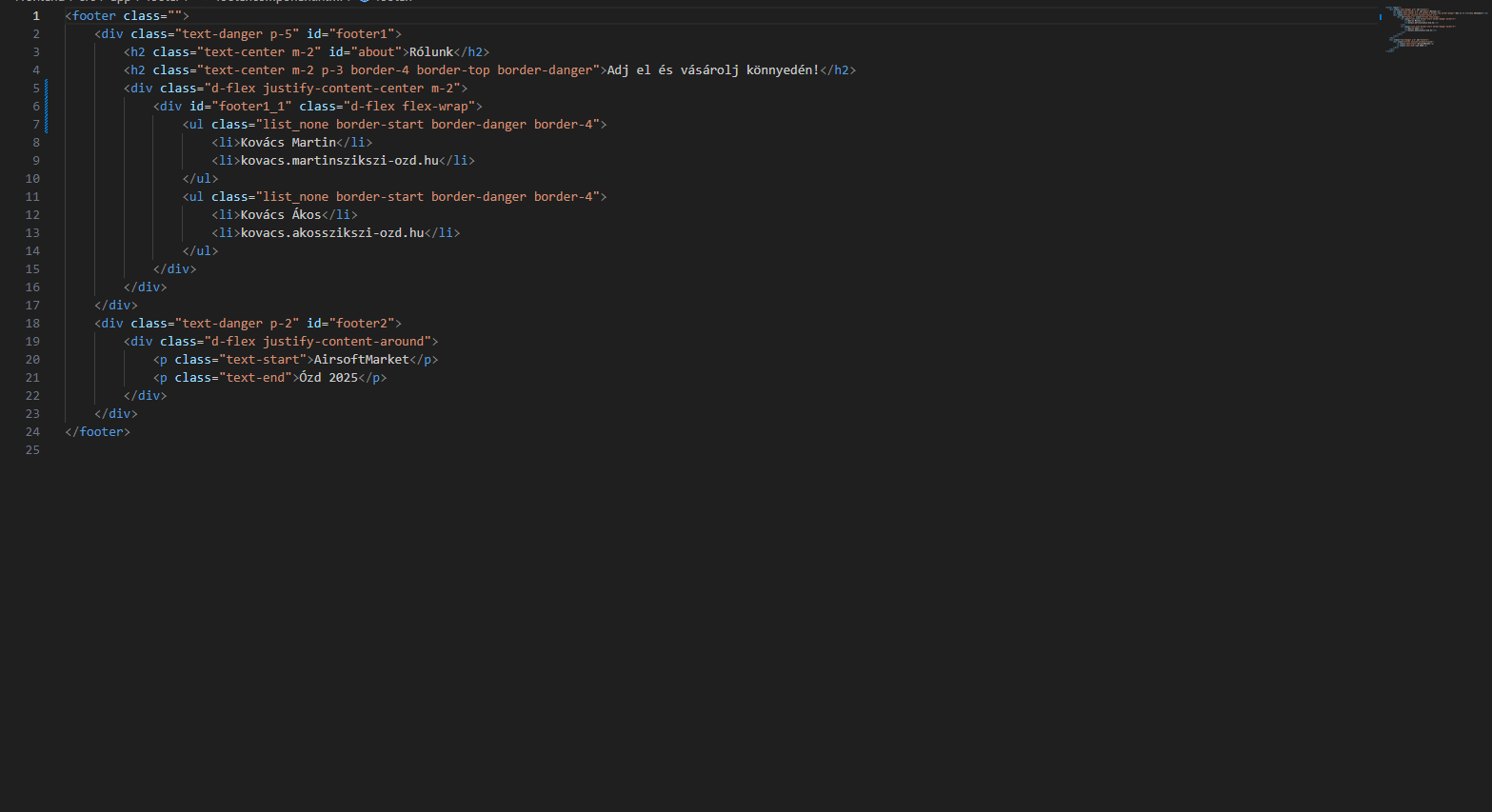
13. kép: ad-supplementarys.ts

**footer komponens – Lábléc**

A footer komponens biztosítja a weboldal alsó információs szekcióját. Két fő részből áll: egy bemutatkozó és egy záró szakaszból.

Az első részben egy "Rólunk" szekció található, amely rövid, motiváló szlogennel („Adj el és vásárolj könnyedén!”) buzdítja a felhasználókat az oldal használatára. Alatta két lista formájában jelennek meg a fejlesztői csapat tagjai: Kovács Martin és Kovács Ákos, valamint e-mail címeik. Az elrendezés rugalmas, középre igazított, és vizuálisan is illeszkedik a főoldal stílusához, piros szegélyekkel és letisztult tipográfiával.

A második rész a lábléc alsó szegmense, amely a weboldal nevét (AirsoftMarket) és az aktuális évet (Ózd 2025) jeleníti meg egymással szemközti elhelyezésben. Ez a szakasz egyszerre nyújt zárást és identitást, egyfajta márkajelzésként funkcionál.



14. kép: footer.component.html

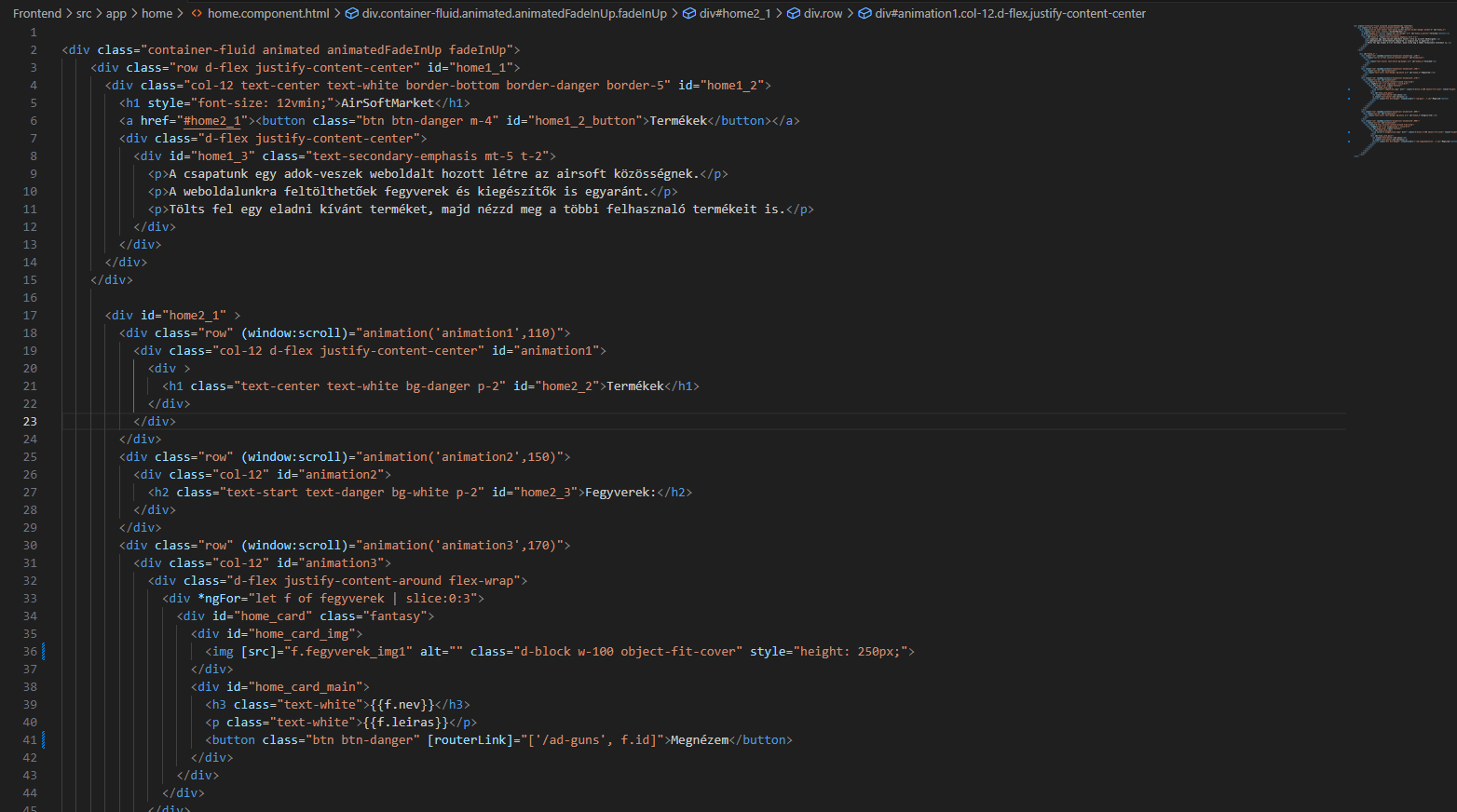
**login komponens – Felhasználói hitelesítés**

A login komponens biztosítja az alkalmazásba történő belépési lehetőséget a felhasználók számára. A komponens egy letisztult, középre pozícionált bejelentkezési panelt jelenít meg, amely két mezőt tartalmaz: felhasználónév és jelszó. A komponens megjelenése illeszkedik az alkalmazás sötét, modern vizuális világához, animált áttűnéssel (animatedFadeInUp) és Bootstrap-alapú stíluselemekkel.

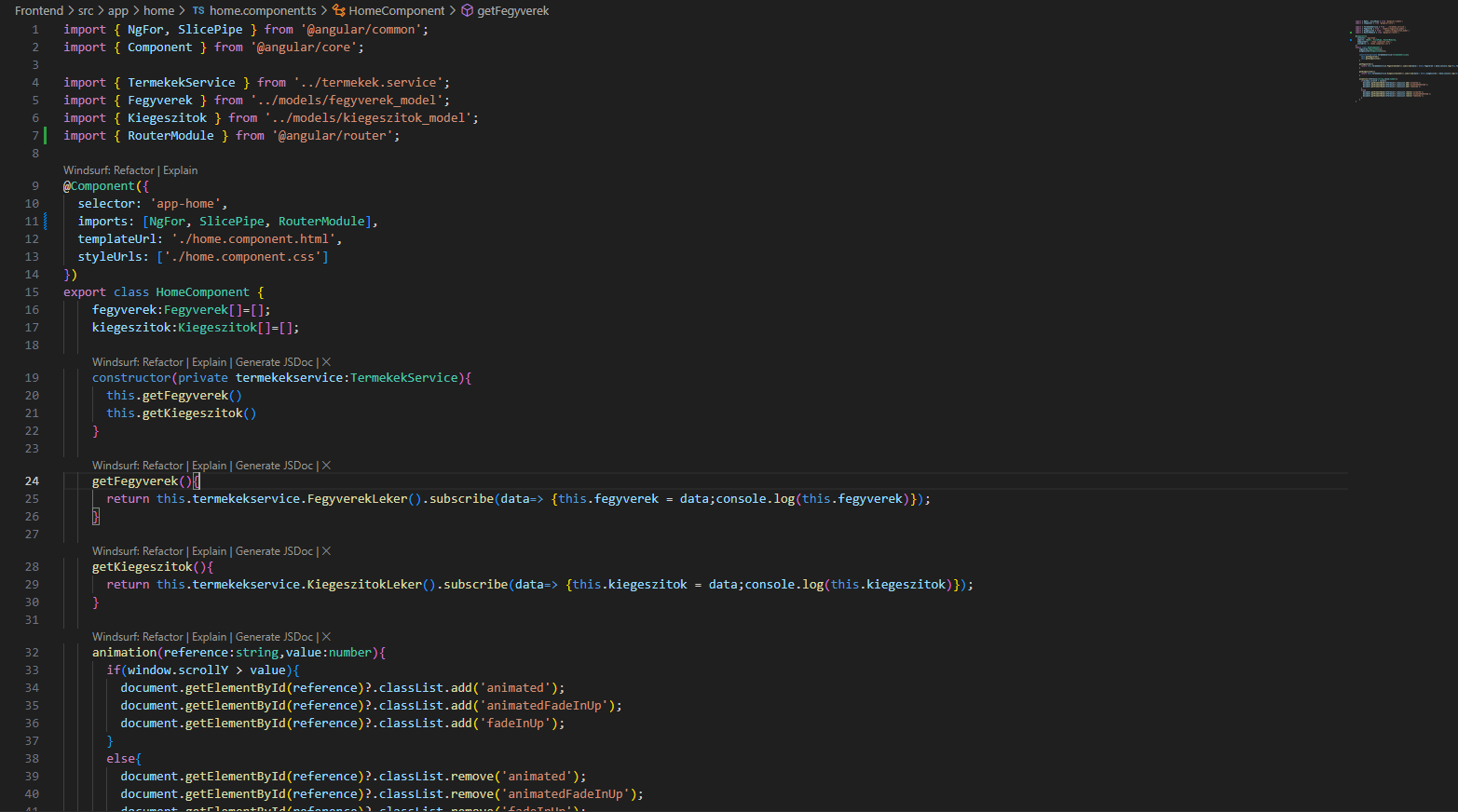
A form működése Angular ngForm direktívával történik, és az adatok kétirányú adatkapcsolattal ([(ngModel)]) kötődnek a komponens belső változóihoz: felhasznalonev és felhasznalojelszo. A form beküldése (ngSubmit) egy login() metódust aktivál, amely feltételezhetően egy szolgáltatáson keresztül (pl. AuthService) kommunikál a backenddel, ahol a hitelesítési logika történik.

Amennyiben a felhasználó érvénytelen adatokat ad meg, vagy a hitelesítés sikertelen, a felület visszajelzést ad a hibáról. A visszajelzés dinamikusan jelenik meg a form alatt, a visszajelzes változó alapján, és a hiba vagy sikeres állapottól függően piros (text-danger) vagy zöld (text-success) színű szövegként formázódik. Sikeres belépés esetén egy plusz üzenet is megjelenik: „Átnavigálás a Főoldalra”, amely egyértelmű jelzést ad a felhasználónak a folyamat végéről.

A komponens vizuálisan is jól strukturált: a formot egy .login nevű CSS osztály keretezi, amely egy minimális szélességet biztosít (min-width: 400px), így garantálva a megfelelő megjelenést kisebb képernyőkön is. A form elemei (form-control, btn btn-primary) a Bootstrap rendszer szabványos komponenseit használják, ami biztosítja a konzisztenst megjelenést az alkalmazás egészében.



15. kép: login.component.html



16. kép: login.component.ts

**profile komponens – Felhasználói fiók kezelése**

A profile komponens a bejelentkezett felhasználó személyes fiókját jeleníti meg és kezeli. Ez az oldal többféle funkciót is ellát: a profiladatok megjelenítését, profilkép feltöltést, saját termékek listázását és kezelését, valamint megrendelések áttekintését és végső soron a fiók törlését is lehetővé teszi.

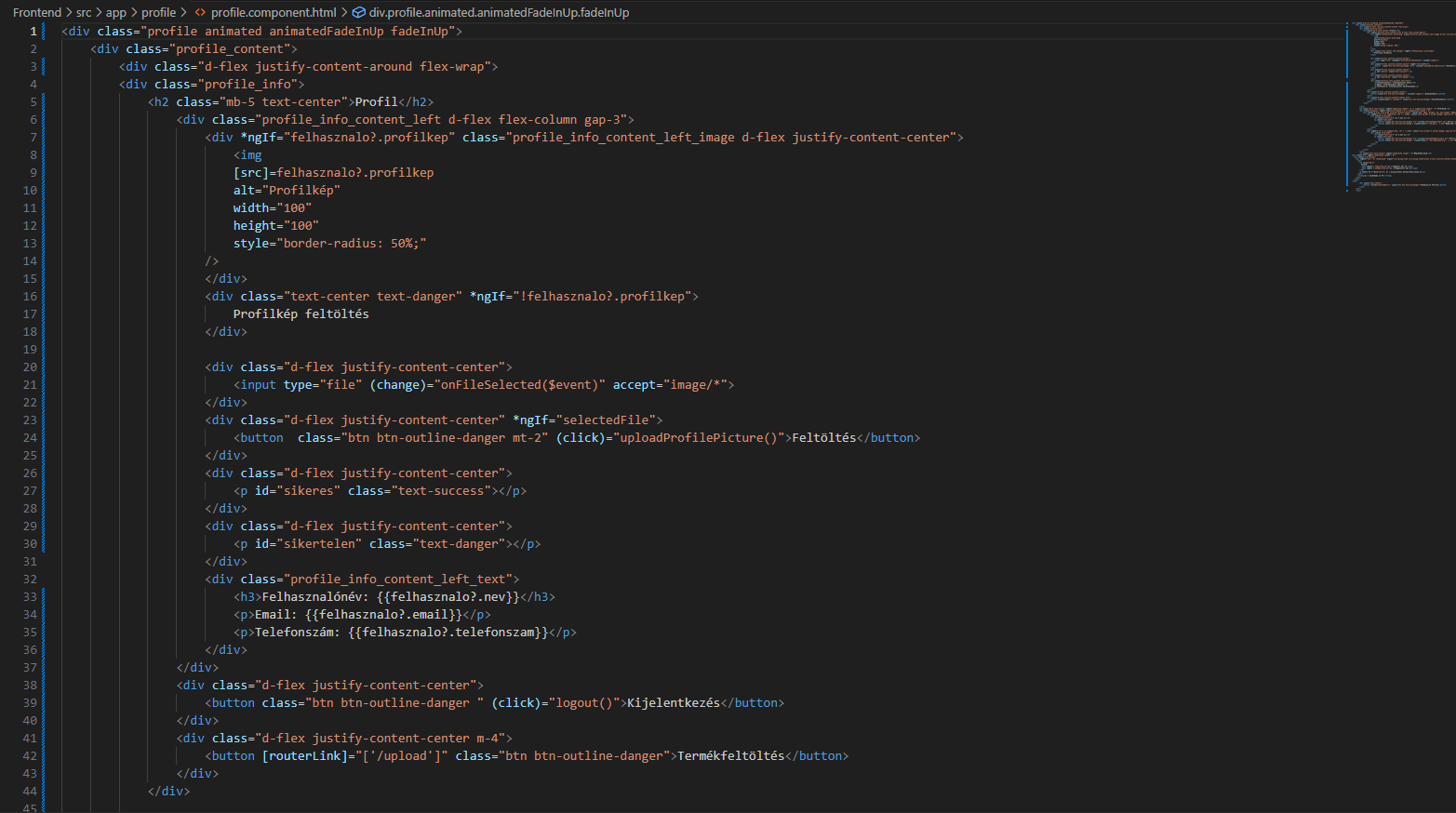
A komponens felső részén a felhasználó neve, e-mail címe és telefonszáma jelenik meg. Amennyiben a felhasználónak van profilképe, az egy kör alakú kép formájában kerül megjelenítésre. Ha nincs, a rendszer egy szöveges figyelmeztetést jelenít meg, és lehetőséget kínál kép feltöltésére. A kép kiválasztása után egy „Feltöltés” gomb válik elérhetővé, amely a uploadProfilePicture() metódust hívja meg. Sikeres vagy sikertelen feltöltés esetén a felhasználó visszajelzést kap a megfelelő színkóddal (zöld vagy piros szöveg).

A komponens tartalmaz egy „Kijelentkezés” gombot, amely a logout() metódushoz van kötve, valamint egy „Termékfeltöltés” gombot, amely a termékfeltöltési oldalra navigál (/upload).

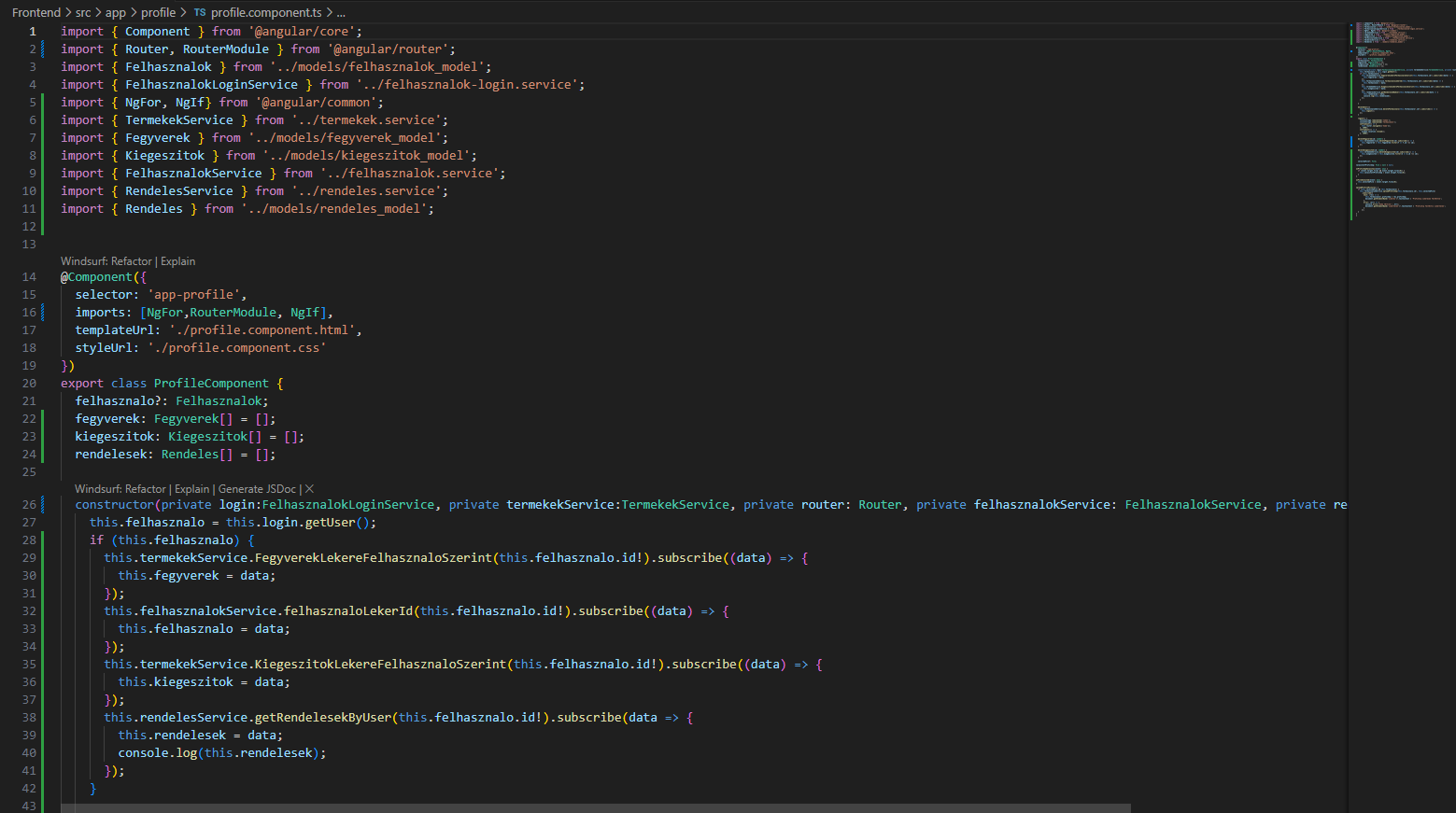
Lejjebb a komponens a felhasználó által korábban feltöltött termékeket jeleníti meg két külön kategóriában: fegyverek és kiegészítők. Ezek listázása kártyaszerű formában történik, ahol minden kártyán szerepel a termék neve, és két gomb: „Megnézem” a részletes megtekintéshez, és „Törlés” a termék eltávolításához. A törlés a megfelelő metódus (deleteFegyverek() vagy deleteKiegeszitok()) meghívásával történik.

Végül, ha a felhasználónak volt megrendelése, azok is megjelennek egy jól strukturált, listacsoportosított nézetben. Itt látható a termék neve, kiszállítás típusa, valamint az ár. Az oldal végén egy „Felhasználó törlése” gomb található, amellyel a fiók véglegesen eltávolítható.

Ez a komponens egyértelműen a felhasználói élmény egyik központi eleme: kényelmesen hozzáférhetők belőle a legfontosabb fiókműveletek, miközben a felület letisztult, vizuálisan egységes és gördülékenyen működik.



17. kép: profile.component.html



18. kép: profile.component.ts

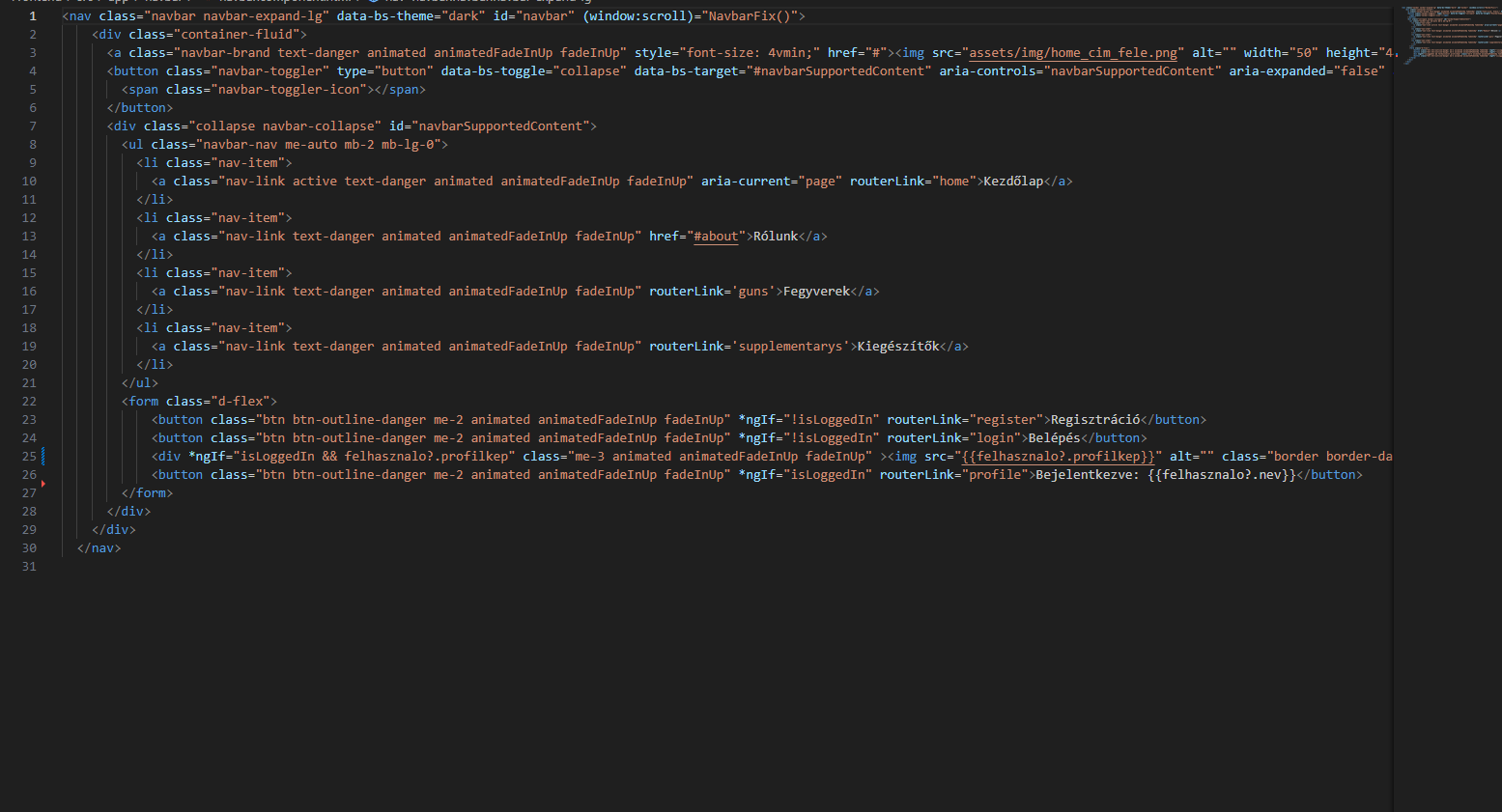
**navbar komponens – Navigációs sáv**

A navbar komponens a webalkalmazás felső navigációs sávját tartalmazza, amely minden oldalon elérhető. A dizájn Bootstrap-alapú, sötét témában megjelenő elemekkel, és animált áttűnésekkel (animatedFadeInUp) lett kialakítva. A navigáció a márka logójával kezdődik, amely egy képet is tartalmaz, és a „AirSoftMarket” névvel hivatkozik a kezdőlapra.

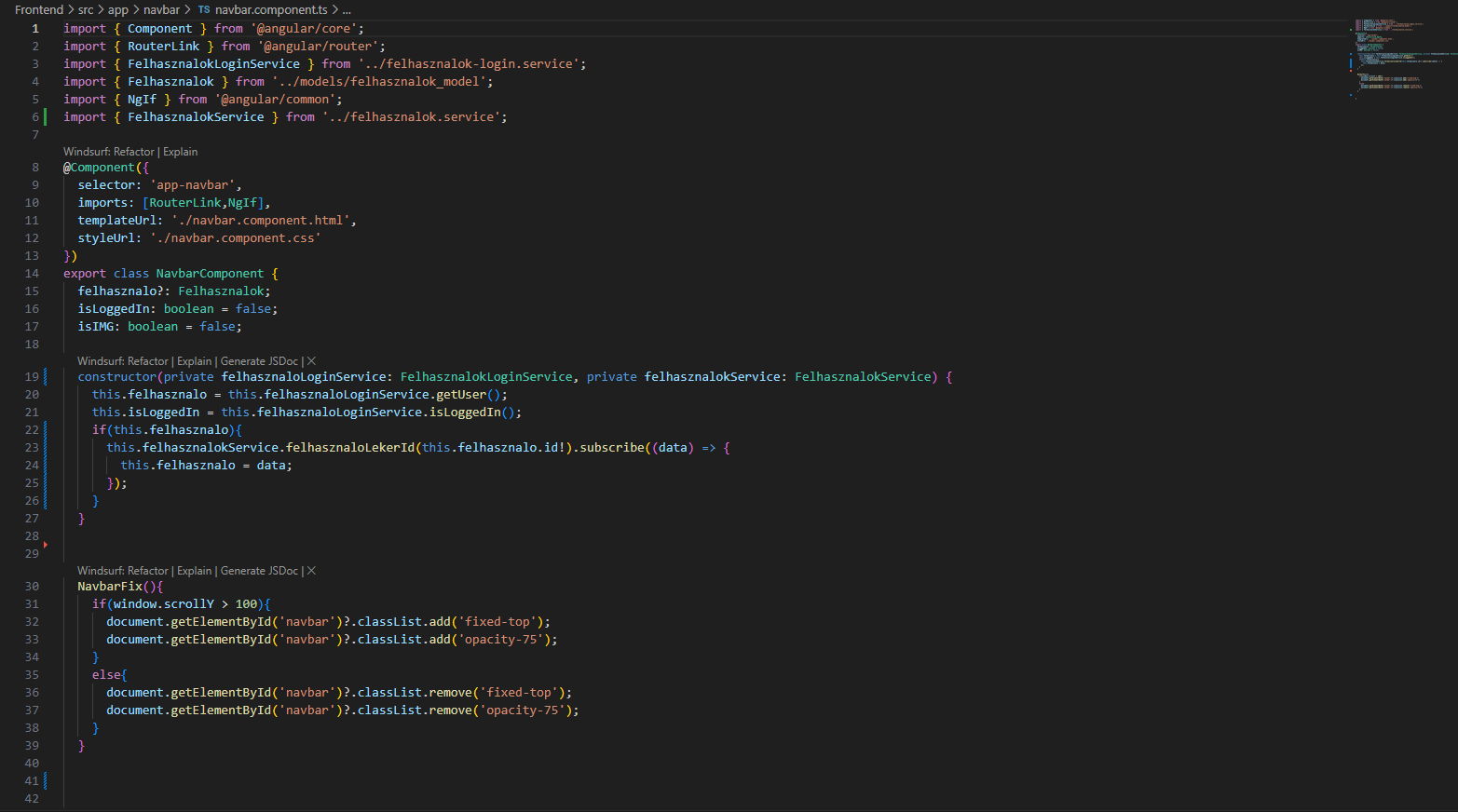
A menü mobilbarát: kisebb kijelzőkön automatikusan összecsukódik, és egy „hamburger” ikonon keresztül válik elérhetővé. Az alapértelmezett navigációs elemek között szerepel a „Kezdőlap”, a „Rólunk” szekcióra mutató hivatkozás (egy oldalon belüli link formájában), valamint a „Fegyverek” és „Kiegészítők” oldalak.

A jobb oldalon a rendszer dinamikusan jeleníti meg a bejelentkezési lehetőségeket. Amennyiben a felhasználó nincs belépve, „Regisztráció” és „Belépés” gombok jelennek meg. Ha viszont a felhasználó be van jelentkezve, megjelenik a profilképe egy kör alakú keretben, valamint egy gomb, amely a „Bejelentkezve: [név]” szöveget jeleníti meg és a profiloldalra navigál.

A navigációs sáv nem csak esztétikai, hanem funkcionalitás szempontjából is kulcsfontosságú része az alkalmazásnak, hiszen általa érhető el az oldal összes fő tartalma. Az alkalmazott NavbarFix() metódus valószínűleg dinamikus stílusváltást végez görgetés alapján, tovább erősítve a reszponzív élményt.



19. kép: navbar.component.html



20. kép: navbar.component.ts

**purchase komponens – Rendelési folyamat**

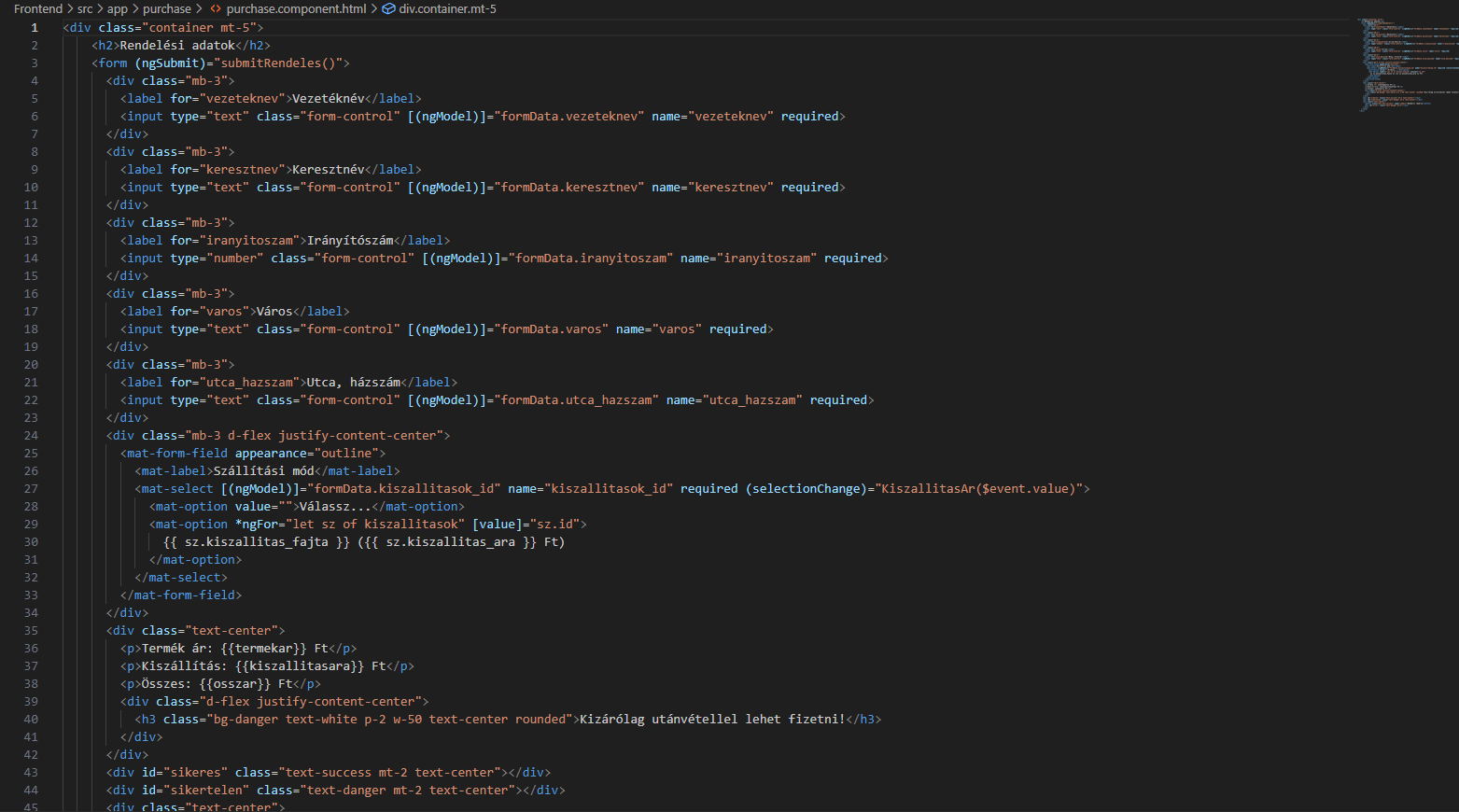
A purchase komponens biztosítja a felhasználó számára a termék(ek) megrendeléséhez szükséges űrlapot és folyamatot. A vizuális felület egy jól strukturált, színes dobozban jelenik meg, amely a piros árnyalataival és árnyékolással hangsúlyozza a rendelés fontosságát, miközben fenntartja az oldal egységes, „fantasy” hangulatú dizájnját.

A komponens egy Angular-űrlap segítségével gyűjti be a szállításhoz szükséges adatokat: vezetéknév, keresztnév, irányítószám, város, utca és házszám. Ezen mezők mind kötelezőek, és a form integrálva van az Angular ngModel struktúrájába, így a beviteli értékek valós időben frissülnek a formData objektumban.

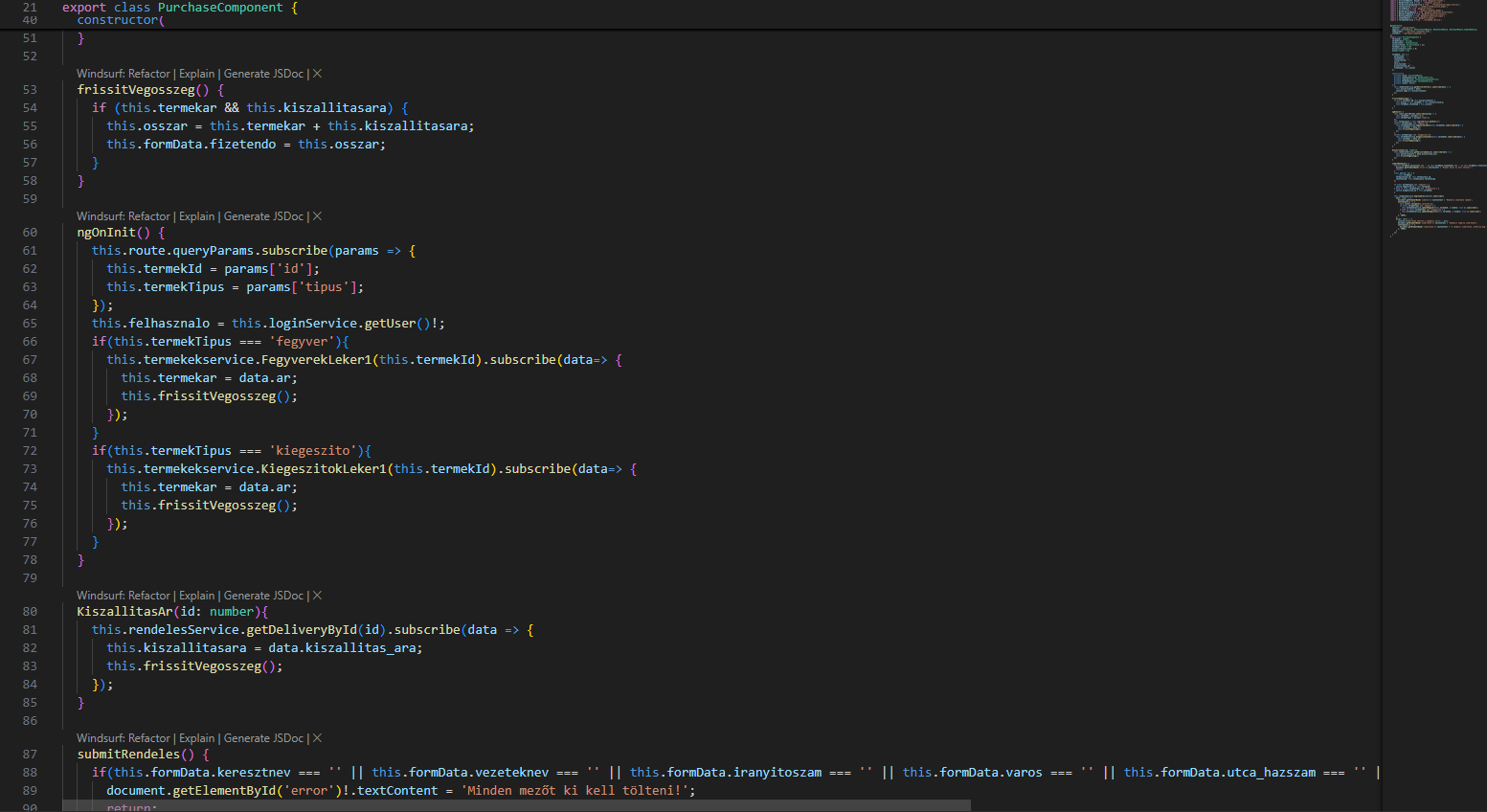
A kiszállítási mód kiválasztása egy Angular Material mat-select segítségével történik, amely lehetővé teszi, hogy a felhasználó válasszon a backend által szolgáltatott opciók közül. A kiválasztott szállítási mód ára automatikusan frissül, és az összesített végösszeg is kiszámításra kerül a termék és a szállítás árának összeadásával.

Külön kiemelésre kerül, hogy a fizetés kizárólag utánvéttel lehetséges, amit a komponens egy jól látható figyelmeztető dobozban jelez. Az oldal tartalmaz sikeres és sikertelen visszajelzések megjelenítésére szolgáló mezőket is, valamint egy „Rendelés leadása” gombot, amely aktiválja a submitRendeles() metódust.

Ez a komponens egyszerű, de fontos része az alkalmazásnak, hiszen a vásárlói élmény egyik kulcspontját – a vásárlás végső lépését – testesíti meg. Letisztult felépítése révén könnyen kezelhető és egyértelmű a látogatók számára.



21. kép: purchase.component.html



22. kép: purchase.component.ts

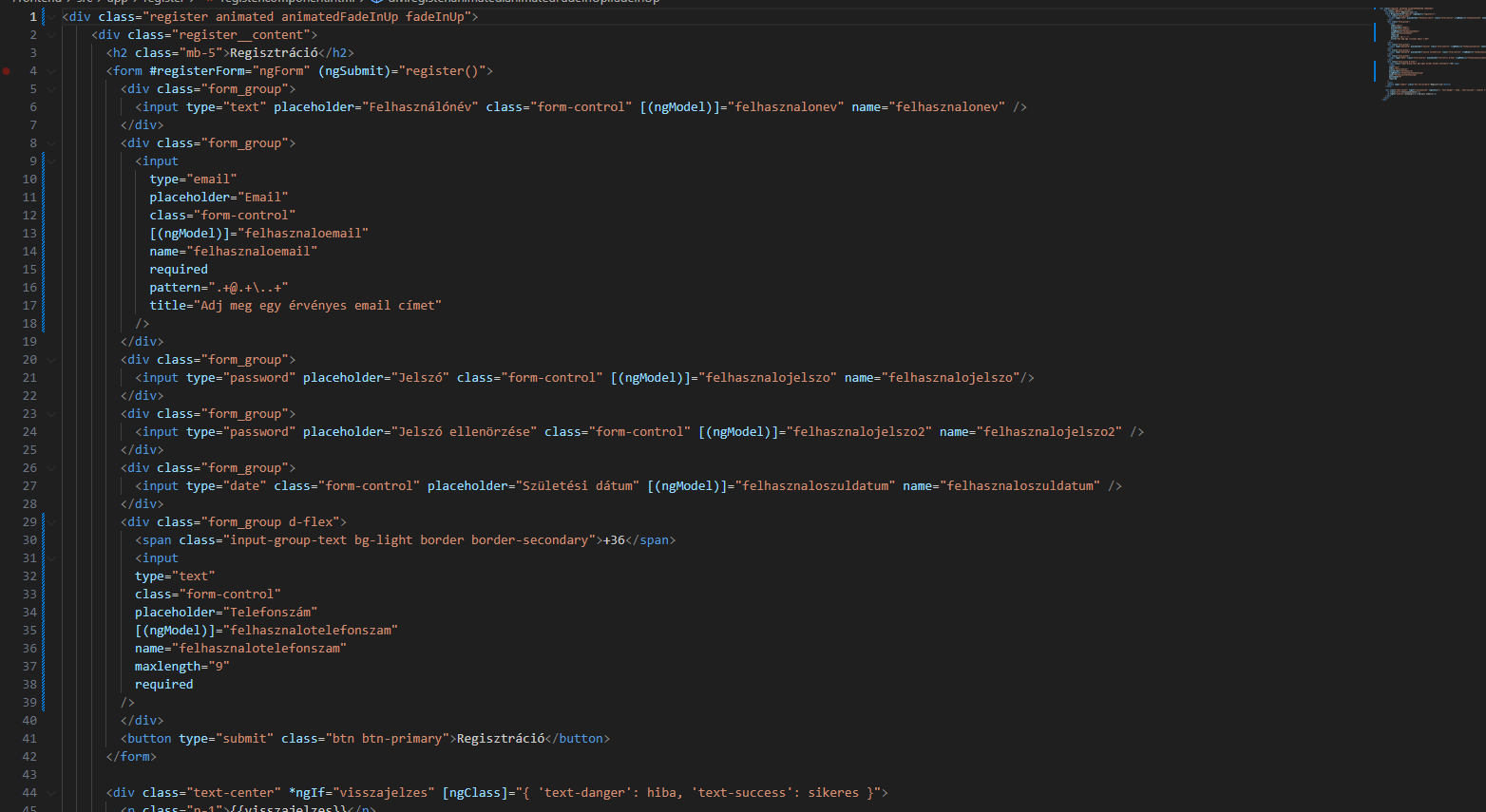
**register komponens – Felhasználói regisztráció**

A register komponens lehetővé teszi új felhasználók számára a fiók létrehozását. A komponens kinézete és elrendezése hasonlít a bejelentkezési oldalhoz: középen elhelyezkedő, animált megjelenésű regisztrációs panel, amely minden szükséges mezőt tartalmaz a regisztrációhoz.

A form mezői a következők: felhasználónév, e-mail cím (valós e-mail validációval), jelszó és annak megerősítése, születési dátum, valamint telefonszám, amely a magyar szabványnak megfelelő +36 előhívóval jelenik meg. A mezők [(ngModel)] kötéssel kapcsolódnak az Angular komponens megfelelő változóihoz, biztosítva a gördülékeny adatkezelést.

A form kitöltését követően a felhasználó a „Regisztráció” gombra kattintva kezdeményezheti a fiók létrehozását. A regisztráció eredményéről dinamikus visszajelzés történik: siker esetén zöld, hiba esetén piros szövegezéssel. Sikeres regisztráció után egy további sor tájékoztatja a felhasználót arról, hogy a rendszer át fogja navigálni őt a belépési oldalra.

Ez a komponens logikailag jól strukturált és intuitív – kezdő felhasználók számára is egyszerűen használható. A felhasználói élményt fokozza az egyértelmű visszajelzésrendszer, a beépített validáció, valamint a reszponzív és animált megjelenés.



23. kép: register.component.html



24. kép: register.component.ts

**upload komponens – Termékfeltöltés**

Az upload komponens biztosítja a felhasználók számára, hogy új termékeket adhassanak hozzá a piactérhez, legyen szó akár fegyverről, akár kiegészítőről. A felület logikusan felépített és vizuálisan is egyértelműen elkülöníti a különböző terméktípusokat. Az űrlap megjelenése egy világos, keretezett dobozban történik, amelyet vörös szegély és „fantasy” betűtípus tesz különlegessé – ez a vizuális stílus jól illeszkedik a weboldal általános hangulatához.

A folyamat első lépése a terméktípus kiválasztása. Az Angular Material mat-select komponens segítségével a felhasználó eldöntheti, hogy kiegészítőt vagy fegyvert kíván feltölteni. A kiválasztott érték alapján dinamikusan változik a form tartalma, \*ngIf direktívával szabályozva.

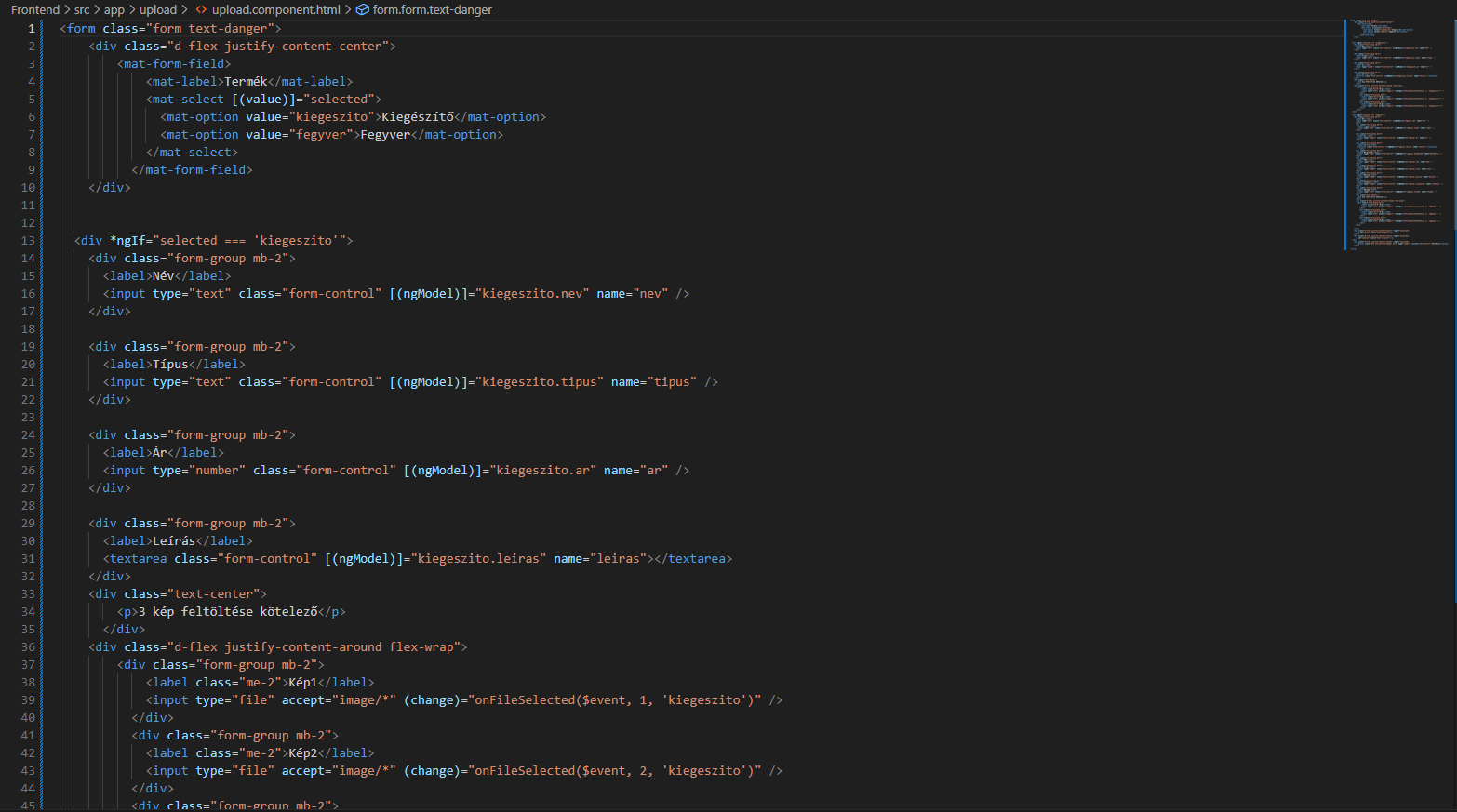
Amennyiben a felhasználó kiegészítőt választ, egy egyszerű űrlap jelenik meg, amely a következő mezőket tartalmazza: név, típus, ár és leírás. Ezután három kép feltöltésére is lehetőség van, amelyek a kiegészítőhöz kapcsolódnak.

Ha a fegyver opciót választja, az űrlap kibővül specifikus műszaki paraméterekkel is: mechanika, FPS (lövedék sebesség), súly, hossz, csőhossz, tűzmód. Természetesen itt is kötelező három kép feltöltése.

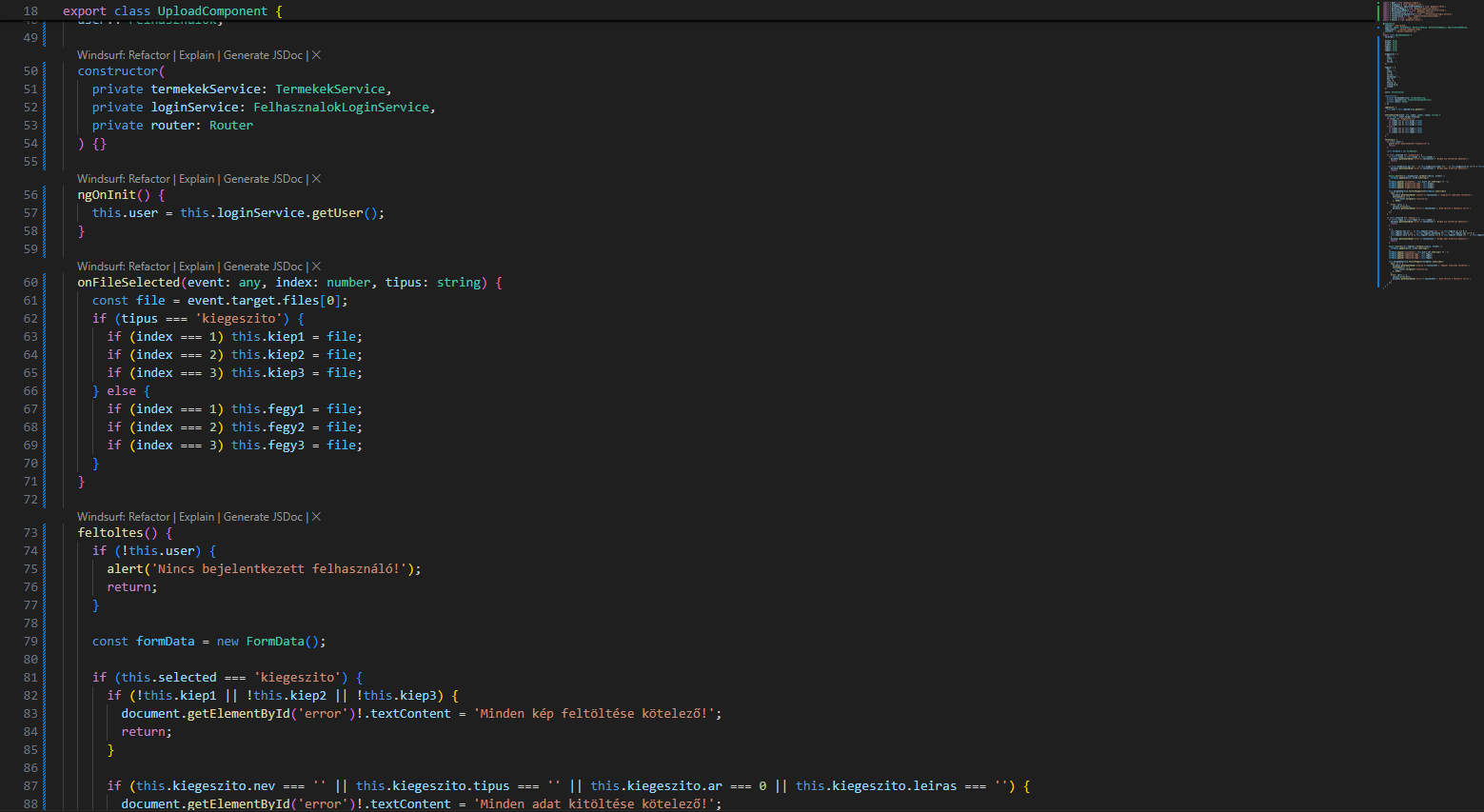
A képválasztás eseményvezérelten történik, ahol minden kép onFileSelected() metóduson keresztül kerül feldolgozásra, és eltárolásra a komponensben. A feltöltés során külön visszajelző mezők (#error, #sikeres) jelenítik meg a folyamat eredményét, legyen szó sikeres feltöltésről vagy hibáról.

A form egyetlen gombbal zárul, amely aktiválja a feltoltes() metódust. Ez végzi el az adatok tényleges továbbítását a backend felé, ahol azok tárolásra kerülnek.

Ez a komponens fontos része az alkalmazás funkcióinak, hiszen lehetővé teszi a közösség számára, hogy aktívan részt vegyenek a piactér tartalmának bővítésében. Egyszerűsége és jól strukturált felépítése révén a kezdő felhasználók számára is könnyen használható.



25. kép: upload.component.html



26. kép: upload.component.ts

**Útvonalkezelés**

Az alkalmazás útvonalait az app.routes.ts fájl tartalmazza, amely az Angular Routes tömbjén keresztül definiálja, hogy mely URL-útvonal melyik komponenshez vezet. Az útvonalak révén biztosított a navigációs logika, amely lehetővé teszi, hogy a felhasználók különböző oldalakat érjenek el a megfelelő komponenseken keresztül.

Az útvonalak szerkezete jól átgondolt és egyértelmű:

* a kezdőoldal a HomeComponent-hez navigál, biztosítva, hogy az alapértelmezett gyökérútvonal is a nyitóoldalra mutasson
* a guns és supplementarys útvonalak a terméklisták oldalaira visznek, külön komponensekkel a fegyverek (AdGunsComponent) és a kiegészítők (AdSupplementarysComponent) számára
* a register és login útvonalak a regisztrációs, illetve bejelentkezési oldalakat jelenítik meg
* a profile útvonal a bejelentkezett felhasználó saját adatlapjához és termékeihez navigál, míg a upload útvonal lehetőséget biztosít új termék feltöltésére
* a purchase útvonal egy különálló megrendelési űrlapot jelenít meg (PurchaseComponent), amely a kiválasztott termék megvásárlásához kapcsolódik
* a ad-guns/:id és ad-supplementarys/:id dinamikus útvonalak paramétert (termék azonosítót) fogadnak, és az adott fegyver vagy kiegészítő részletes adatlapját jelenítik meg

A route definíciók tehát nemcsak funkcionálisan osztják fel az alkalmazást, hanem tiszta URL-struktúrát is biztosítanak, amely keresőbarát és felhasználóbarát navigációt eredményez. A :id paraméterek használata lehetővé teszi, hogy minden egyes termék saját URL-elérési úton keresztül is hivatkozható legyen.

**Szolgáltatások**

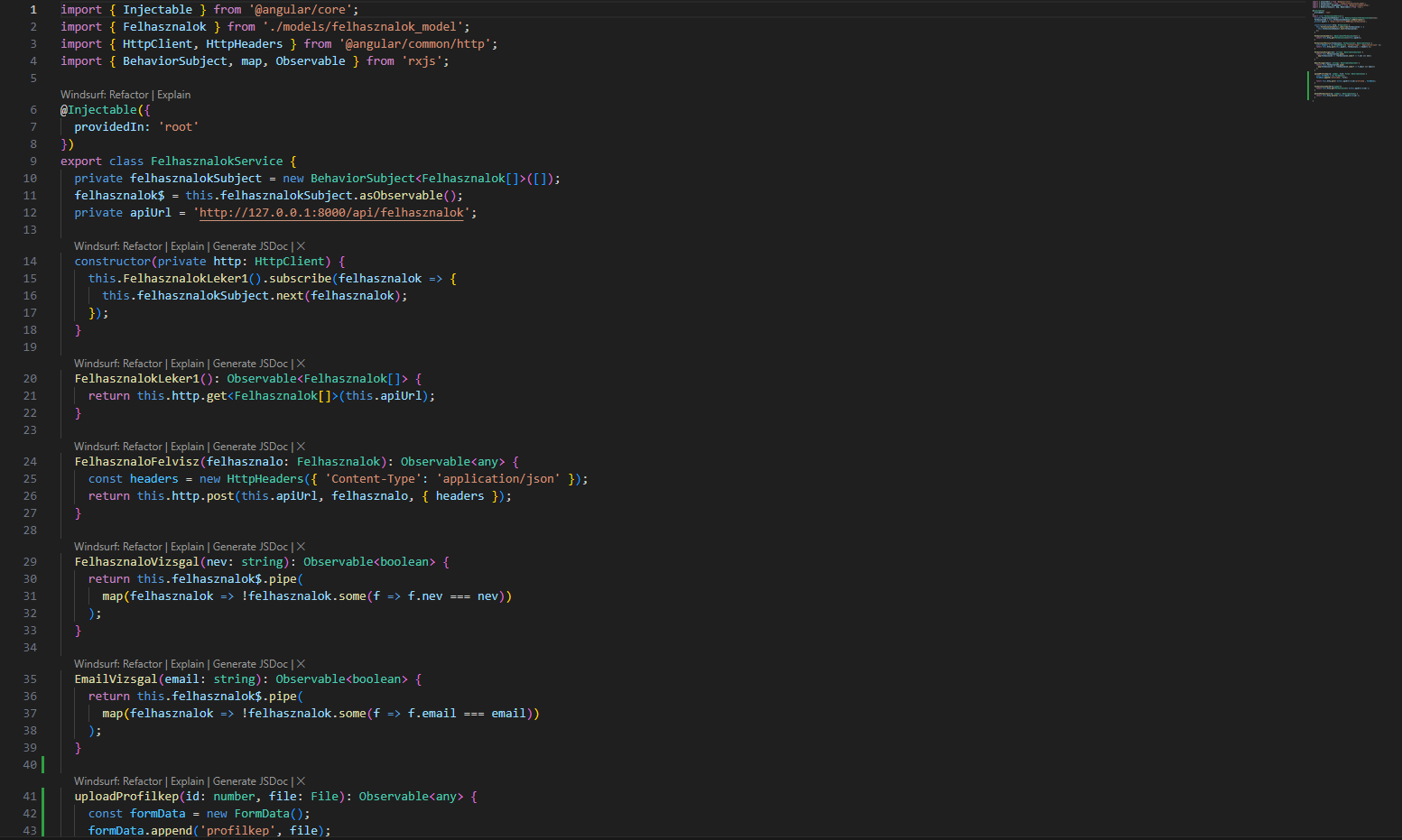
**FelhasznalokService – Felhasználói adatok kezelése**

A FelhasznalokService a rendszer központi szolgáltatása a felhasználói entitások kezelésére. Ez a service felelős az adatok backend API-val történő kommunikációjáért, valamint azok állapotának helyi kezeléséért is egy BehaviorSubject segítségével. Ennek köszönhetően az alkalmazás különböző komponensei real-time hozzáférhetnek a felhasználók aktuális listájához (felhasznalok$).

Fő funkciók:

* FelhasznalokLeker1(): HTTP GET kérés az összes felhasználó lekérdezésére
* FelhasznaloFelvisz(): HTTP POST kérés új felhasználó rögzítéséhez
* FelhasznaloVizsgal() / EmailVizsgal(): Lokális szűrés BehaviorSubject alapon, hogy az adott név vagy e-mail egyedi-e
* uploadProfilkep(): Profilkép feltöltése a backend felé, FormData használatával
* felhasznaloLekerId(): Egy adott felhasználó adatainak lekérése azonosító alapján
* deleteFelhasznalo(): Egy felhasználói fiók végleges törlése

Ez a szolgáltatás erősen támaszkodik az Angular HttpClient moduljára, és REST API-alapú kommunikációt használ a backend Laravel-réteg felé.



27. kép: felhasznalok.service.ts

**FelhasznalokLoginService – Bejelentkezési állapot és hitelesítés**

Ez a lightweight szolgáltatás a frontend oldali hitelesítési állapot kezeléséért felelős. Nem végez HTTP-hívásokat, kizárólag a localStorage használatával kezeli a bejelentkezési állapotot és a felhasználó adatait.

Fő funkciók:

* isLoggedIn(): Igaz/hamis értékkel tér vissza attól függően, hogy van-e token tárolva a böngészőben
* getUser(): A localStorage-ban eltárolt felhasználó adatait JSON-ként visszaadja
* logout(): A felhasználói adatok és token eltávolítása, majd az oldal újratöltése

Ez a szolgáltatás fontos közvetítő szerepet tölt be a navigációs sávban (navbar), a profilkezelésben (profile komponens), valamint az auth-guard logikákban is használható lenne.

****

28. kép: felhasznalok-login.service.ts

**TermekekService – Fegyverek és kiegészítők kezelése**

A TermekekService az alkalmazás fő szolgáltatása, amely a különböző termékek (fegyverek és kiegészítők) adatainak kezelésére szolgál. A szolgáltatás REST-alapú HTTP-kérésekkel kommunikál a Laravel backend API-val, és külön metódusokat biztosít minden fő műveletre.

Főbb funkciók:

* Lekérdezések:
  + FegyverekLeker() és KiegeszitokLeker() – minden fegyver és kiegészítő lekérése.
  + FegyverekLeker1(id) és KiegeszitokLeker1(id) – adott ID alapján lekér egy konkrét terméket.
  + FegyverekLekereFelhasznaloSzerint(id) és KiegeszitokLekereFelhasznaloSzerint(id) – csak az adott felhasználóhoz tartozó termékeket szűri le lokálisan.
* Törlés:
  + deleteFegyverek(id) és deleteKiegeszitok(id) – a backend megfelelő API végpontján keresztül törli az adott terméket.
* Feltöltés:
  + feltoltKiegeszito() és feltoltFegyver() – új termék beküldése FormData objektumon keresztül (fájlokat is támogat).
* Frissítés:
  + updateFegyver(id, data) és updateKiegeszito(id, data) – meglévő termék adatainak módosítása PUT kéréssel.

Ez a szolgáltatás kulcsszerepet játszik mind a terméklistázásban, mind a profiloldalon történő saját termékek kezelésében.



29. kép: termekek.service.ts

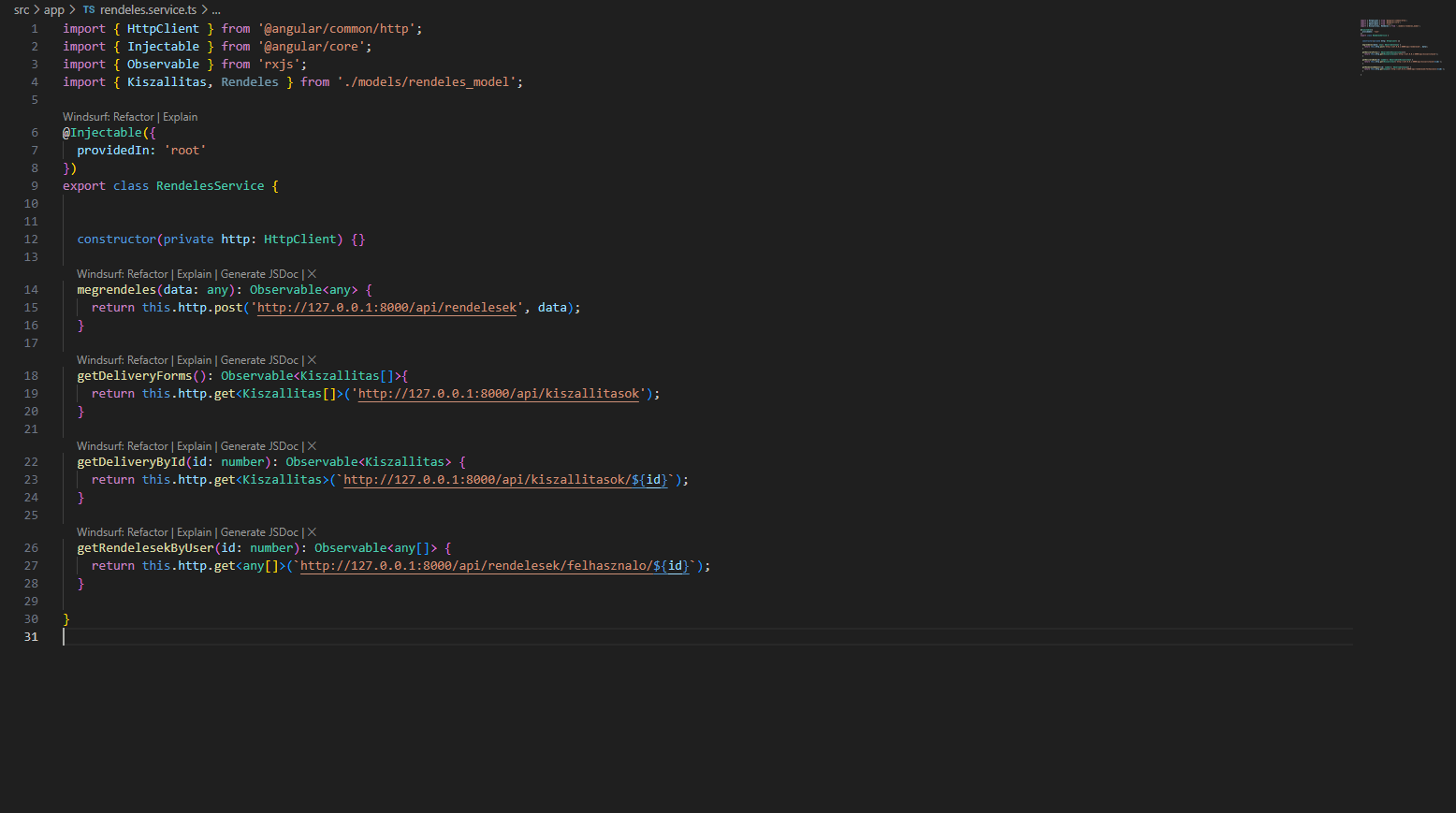
**RendelesService – Megrendelések és kiszállítások**

A RendelesService biztosítja a termékek megrendeléséhez és a szállítási módok kezeléséhez szükséges backend-kommunikációt. Ez a szolgáltatás kapcsolja össze a felhasználót a vásárlási folyamat végső lépésével.

Főbb funkciók:

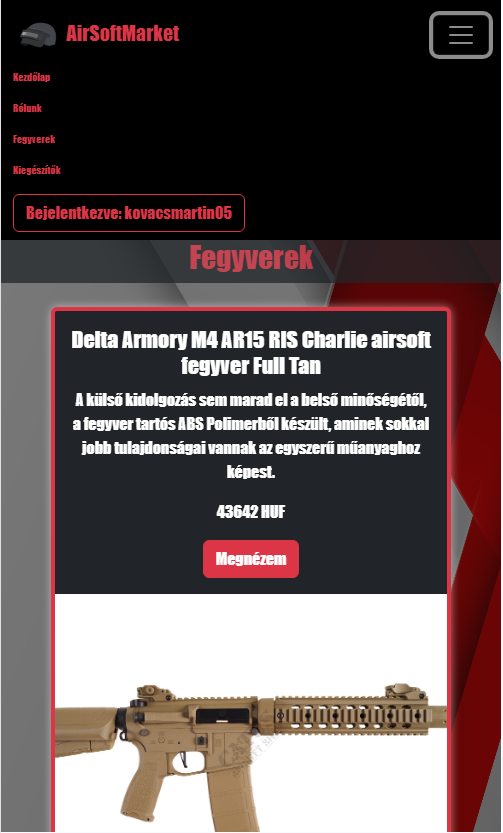
* Rendelés leadása:
  + megrendeles(data) – HTTP POST kérés az új rendelés rögzítésére. A beküldött adat a purchase komponensből származik, amely tartalmazza a felhasználó nevét, címét, a kiválasztott kiszállítási módot és a rendelt terméket.
* Szállítási opciók:
  + getDeliveryForms() – az összes elérhető kiszállítási mód lekérése.
  + getDeliveryById(id) – egy adott szállítási típus lekérdezése azonosító alapján.
* Felhasználói rendelések:
  + getRendelesekByUser(id) – egy adott felhasználóhoz tartozó rendelések listázása, amely megjelenik a profile komponens „Megrendeléseid” szekciójában.

A szolgáltatás logikája egyszerű, mégis robusztus, és hatékonyan kapcsolódik az adatbázis-réteghez. Az Angular HttpClient segítségével könnyedén skálázható, valamint bővíthető például rendelés státuszának kezelésével vagy értesítési rendszerrel is.



30. kép: rendeles.service.ts

Az AirSoftMarket webalkalmazás frontendjének kialakításakor kiemelt figyelmet kapott a reszponzív megjelenítés, hogy az oldal bármilyen képernyőméreten – asztali számítógépen, laptopon, tableten vagy okostelefonon – megfelelően használható és élvezhető legyen.



**Backend**

**Backend kontrollerek dokumentáció**

Az alábbi dokumentáció bemutatja a backend rendszer főbb kontrollereit, azok funkcióit, valamint működésük logikáját. A kontrollerek felelősek az adatok fogadásáért, validálásáért, és továbbításáért a megfelelő modellek és adatbázis-műveletek felé.

1. **A FelhasznaloController**

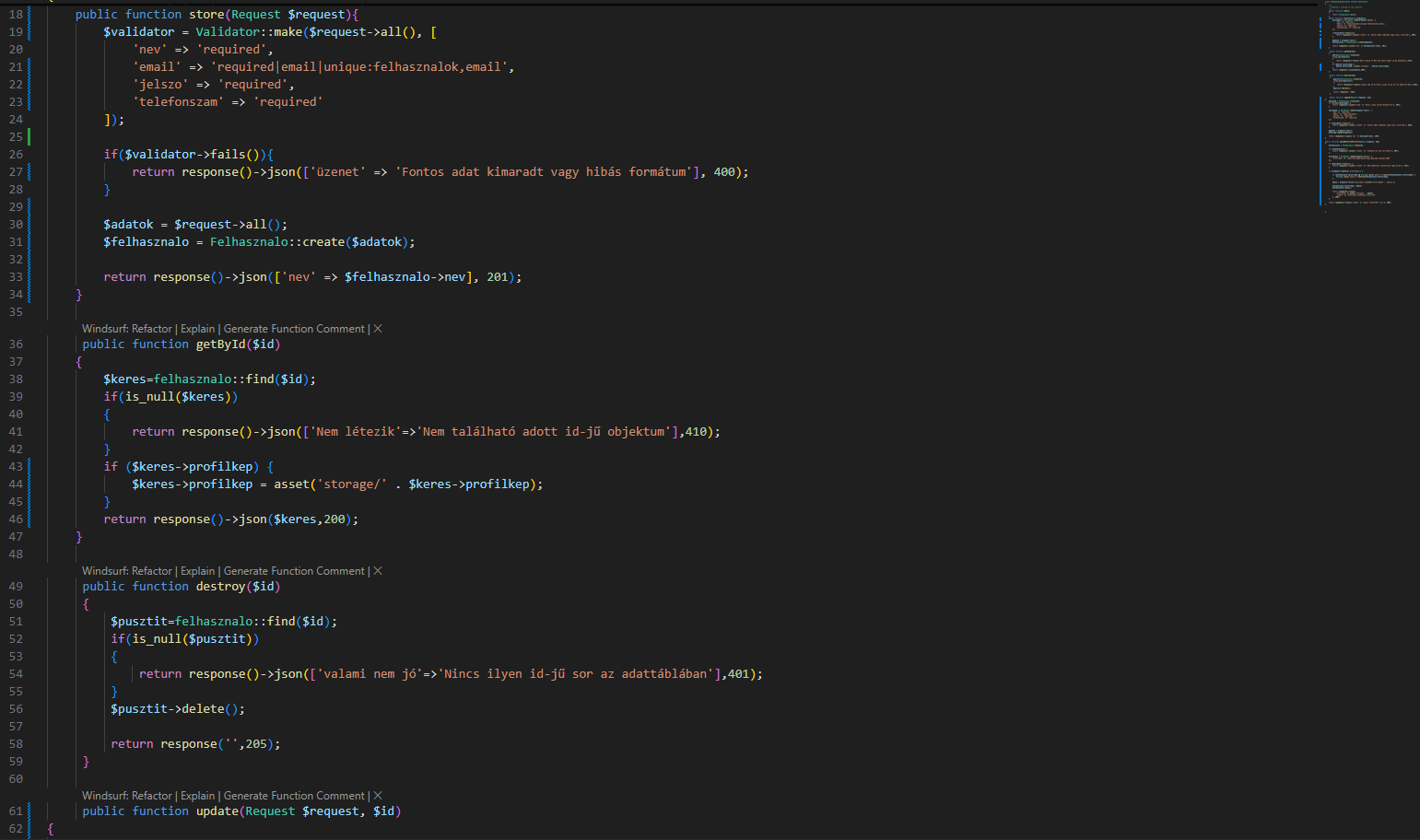
Ez a kontroller kezeli a felhasználók adatainak létrehozását, lekérését és egyéni azonosító alapján történő lekérdezését.

index() – Lekéri az összes felhasználót az adatbázisból.

store(Request $request) – Új felhasználó létrehozása. Validálja a következő mezőket: név, email, jelszó, telefonszám. Az email mező egyediségét is ellenőrzi.

getById($id) – Egy konkrét felhasználó adatainak lekérdezése azonosító alapján.

Validáció hiba esetén JSON válaszban visszaadja a hibák listáját. Sikeres mentés esetén visszaküldi a létrehozott felhasználó nevét.



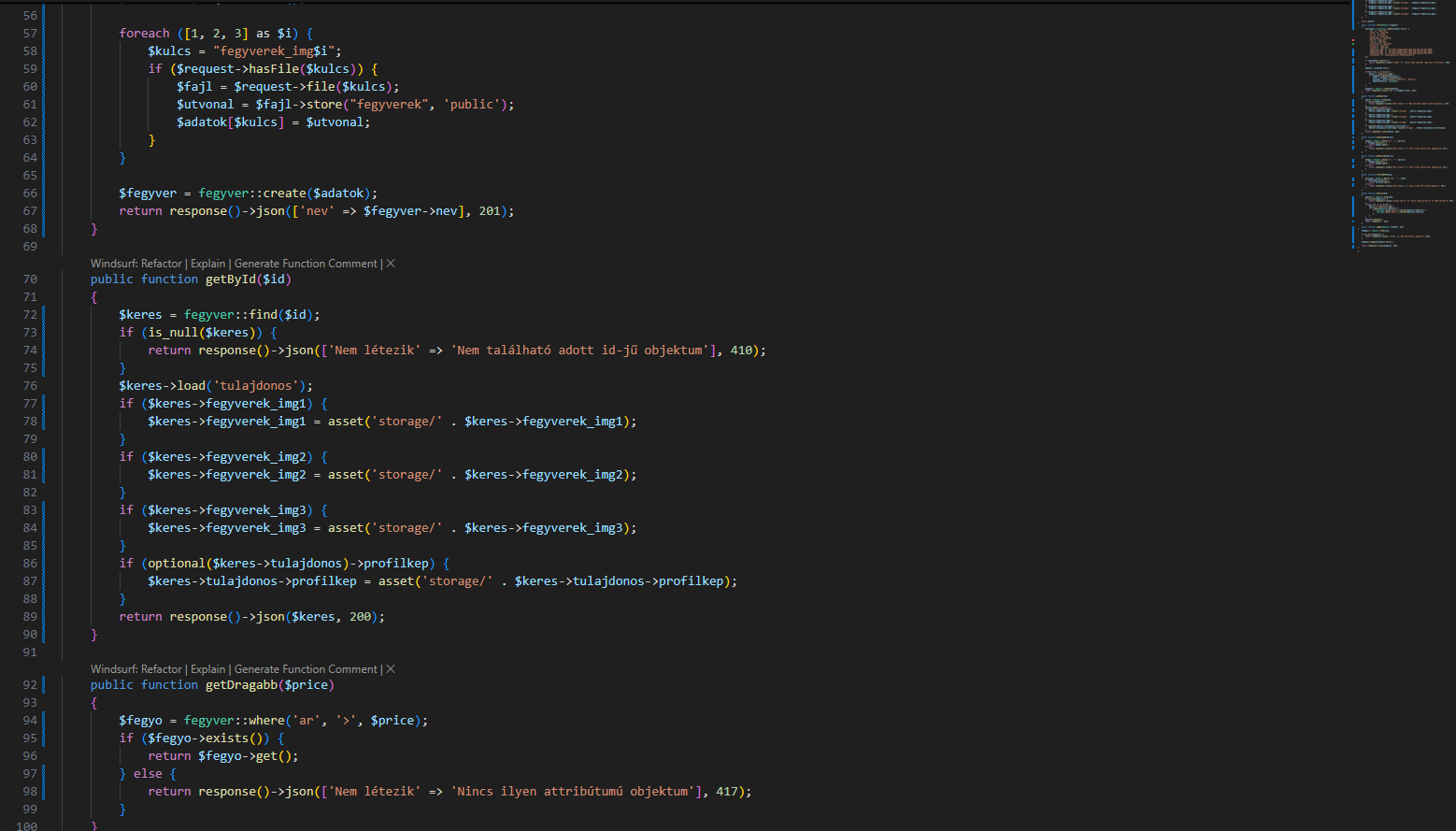
31. kép: FelhasznaloController.php.

1. **FegyverController**

A FegyverController felelős a fegyverek adatainak kezeléséért, különös figyelmet fordítva a képek elérési útvonalainak beállítására.

index() – Lekéri az összes elérhető (el nem adott) fegyvert, a hozzájuk tartozó tulajdonossal együtt. Beállítja a képek (img1, img2, img3) URL-jét a storage/ mappából.

store(Request $request) – Új fegyver létrehozása validált bemeneti adatok alapján.



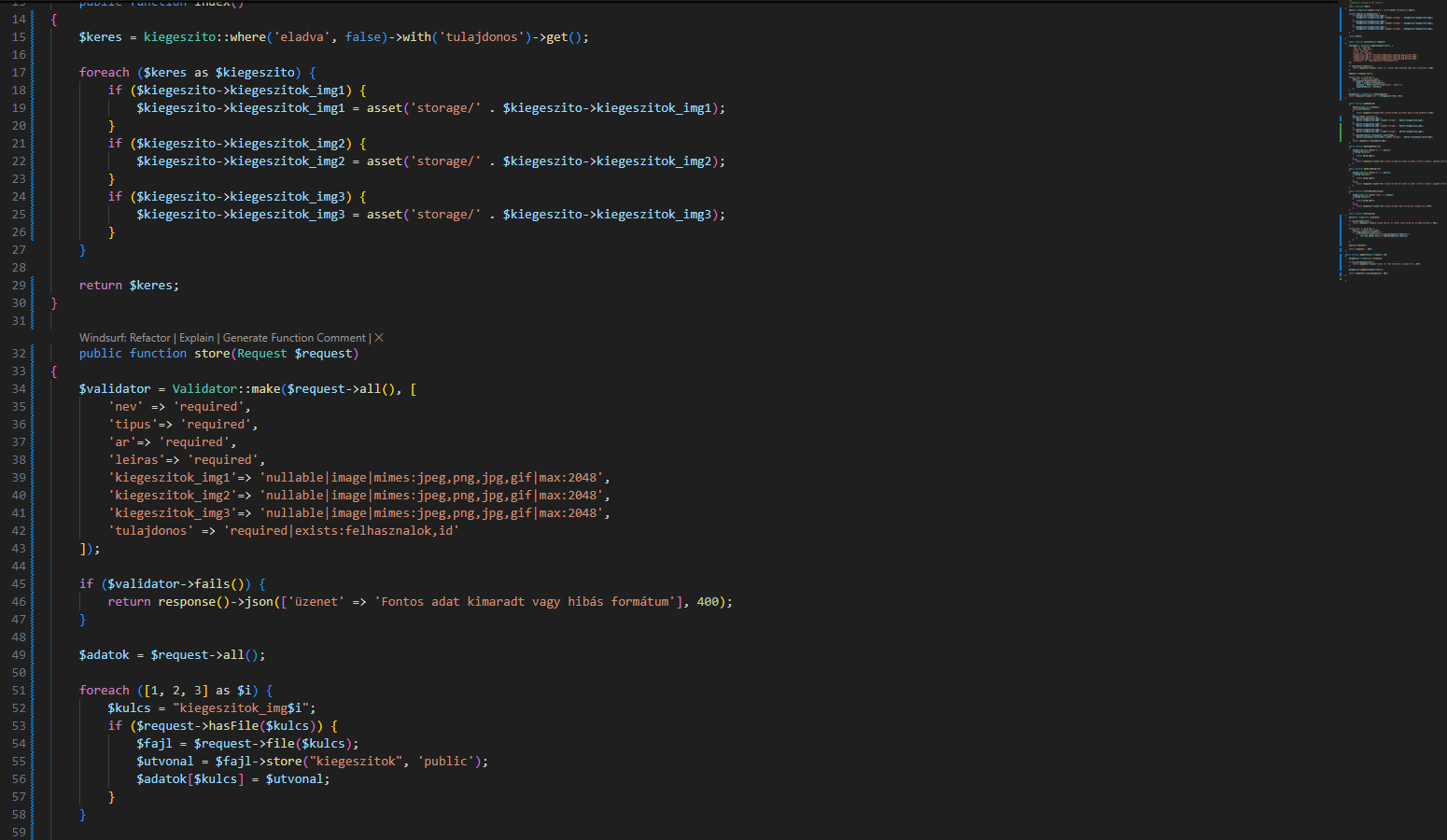
32. kép: FegyverController.php

1. **KiegeszitoController**

A KiegeszitoController működése szinte megegyezik a FegyverController-rel, azonban kiegészítő termékekre vonatkozik.

index() – Lekéri az összes elérhető kiegészítőt, a tulajdonossal együtt. Kezeli a képek URL-jeit.

store(Request $request) – Új kiegészítő hozzáadása. Validálja az adatokat, és JSON válaszként visszaadja az eredményt.



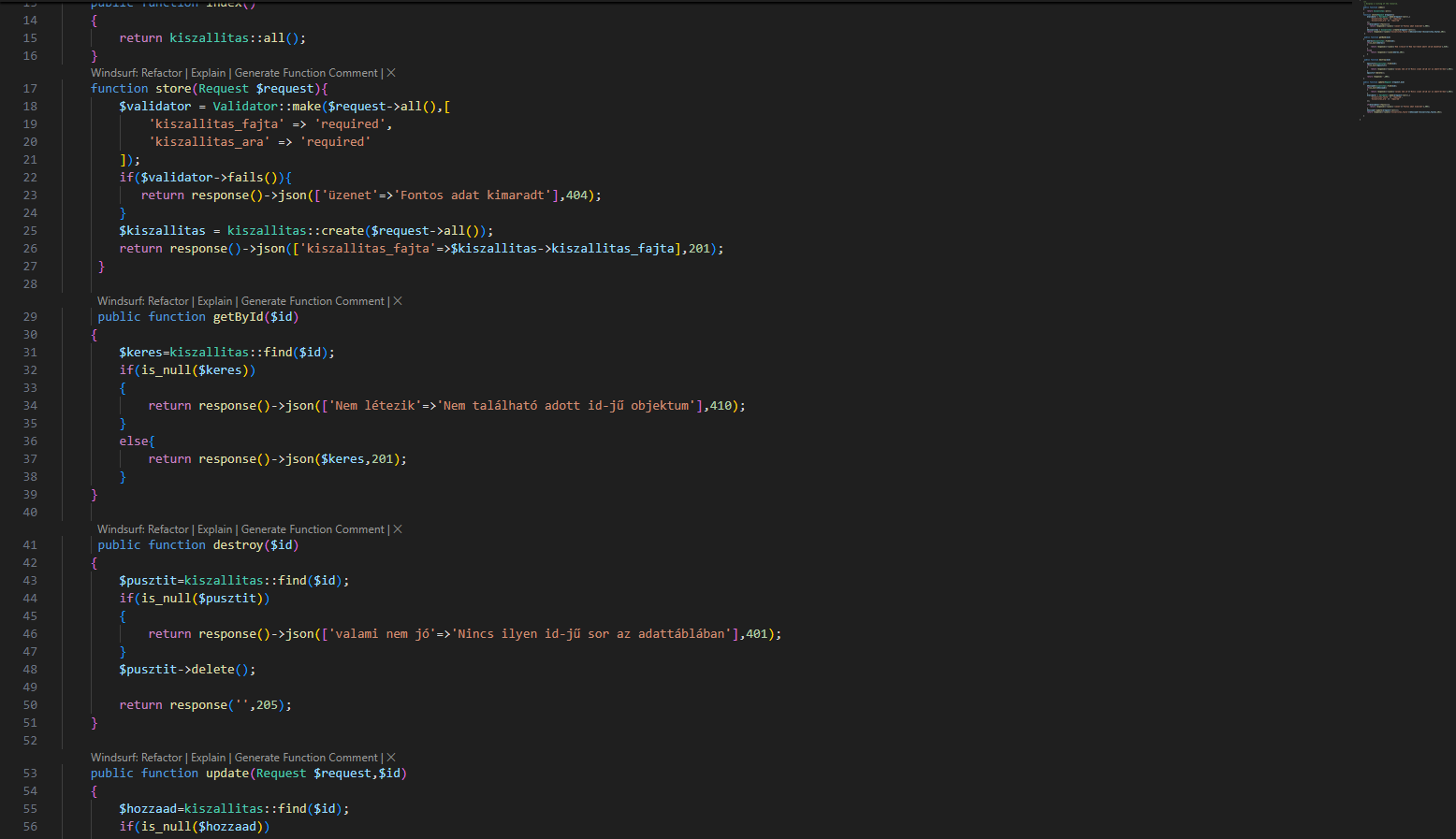
33. kép:KiegeszitoController.php

1. **KiszallitasController**

Ez a kontroller a kiszállítási opciók kezeléséért felel.

index() – Az összes kiszállítási lehetőség listázása.

store(Request $request) – Új kiszállítási típus létrehozása. Kötelező mezők: kiszallitas\_fajta, kiszallitas\_ara.

getById($id) – Egy adott kiszállítási opció lekérdezése ID alapján.

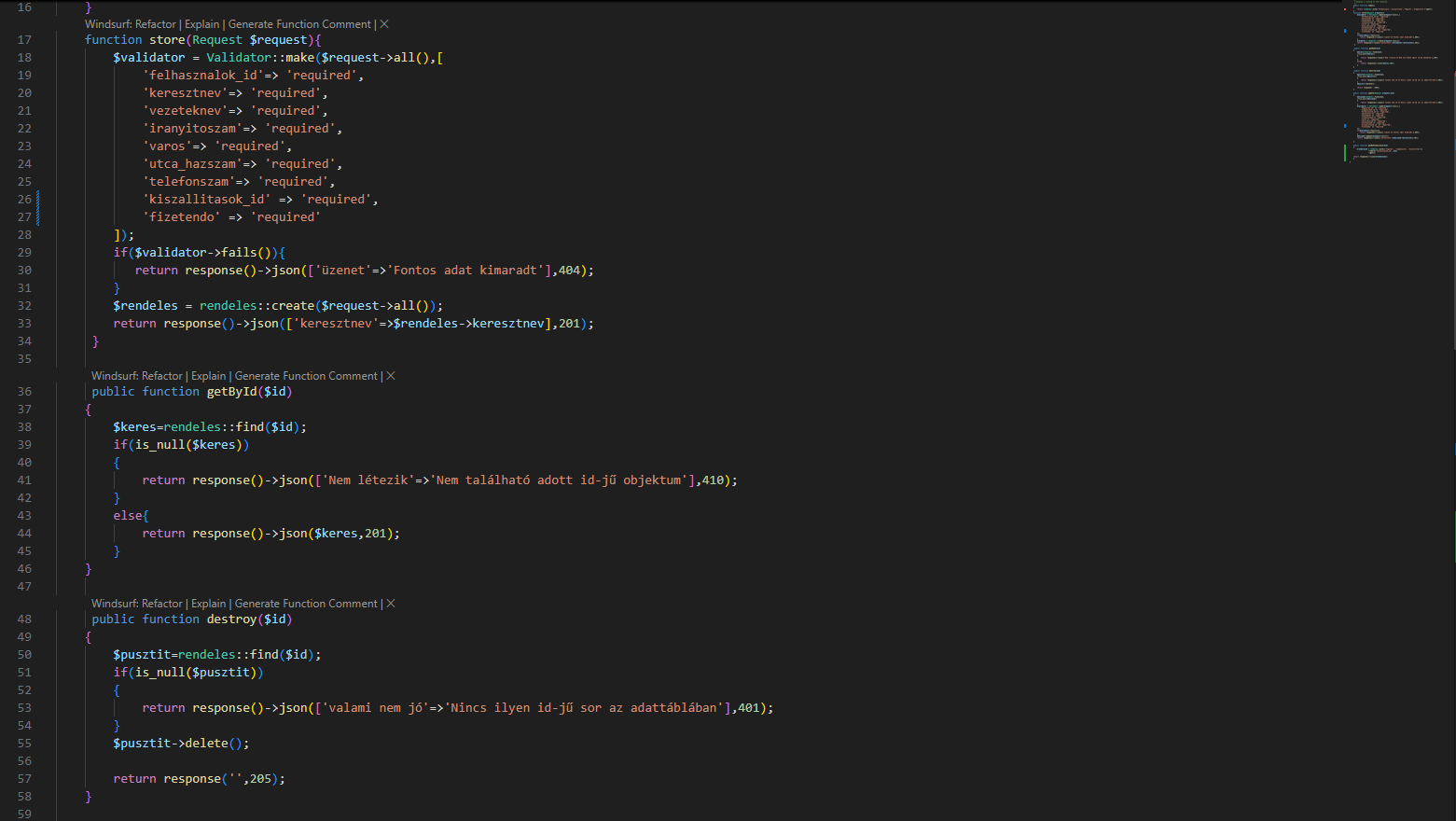
34. kép: KiszallitasController.php

1. **RendelesController**

A RendelesController koordinálja a rendelések kezelését. Egy rendeléshez több modell is kapcsolódik: felhasznalo, kiszallitas, fegyver, kiegeszito.

index() – Minden rendelés listázása, kapcsolt entitásokkal együtt.

store(Request $request) – Új rendelés létrehozása részletes validáció mellett. A kötelező mezők: felhasználói adatok, szállítási adatok, valamint a fizetendo érték.



35. kép: RendelesController.php

**Backend modellek dokumentáció**

1. **felhasznalo modell**

A felhasznalo modell a rendszer regisztrált felhasználóit reprezentálja.

Adatbázistábla: felhasznalok

Mezők:

* nev – Felhasználó neve
* email – Egyedi e-mail cím
* jelszo – Jelszó
* telefonszam – Telefonszám
* profilkep – Profilkép URL
* szulDatum – Születési dátum
* regisztracio\_datuma – Regisztráció időpontja

Kapcsolat:

Kapcsolatban áll fegyver, kiegeszito, rendeles modellekkel

1. **fegyver modell**

A fegyver modell különféle eladásra kínált fegyvereket reprezentál. Adatbázistábla: fegyverek

Mezők:

* nev, tipus, ar, leiras, mechanika, fps, suly, hossza, csohossza, tuzmod
* fegyverek\_img1, fegyverek\_img2, fegyverek\_img3 – Képek tárolási útvonala
* tulajdonos – idegen kulcs a felhasznalo tábla felé
* eladva – boolean típus, jelzi, hogy el van-e adva

Kapcsolat:

belongsTo(felhasznalo) – A metódus mutatja, hogy a fegyvernek van egy tulajdonosa

1. **kiegeszito modell**

Kiegészítők adatait reprezentáló modell logikája hasonló a fegyver modellhez. Adatbázistábla: kiegeszitok

Mezők:

* nev, tipus, ar, leiras
* kiegeszitok\_img1, kiegeszitok\_img2, kiegeszitok\_img3
* tulajdonos, eladva

Kapcsolat:

belongsTo(felhasznalo) – A metódus mutatja, hogy a kiegészítőnek van egy tulajdonosa

1. **kiszallitas modell**

Kiszállítási módokat és árakat tároló modell. Adatbázistábla: kiszallitasok

Mezők:

* kiszallitas\_fajta, kiszallitas\_ara

Kapcsolat:

Része a rendeles modellnek belongsTo kapcsolaton keresztül

1. **rendeles modell**

A rendszer rendeléseit reprezentáló összetett modell, amely több entitást kapcsol össze. Adatbázistábla: rendelesek

Kapcsolatok:

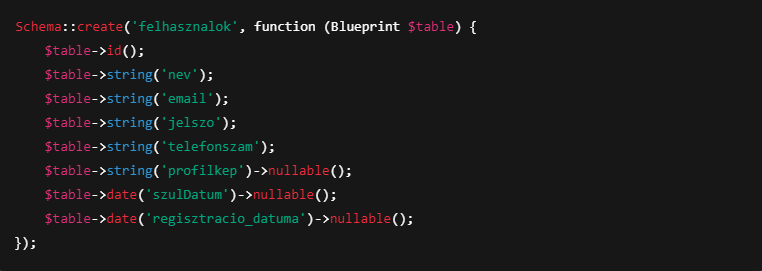
* belongsTo(felhasznalo, 'felhasznalok\_id')
* belongsTo(kiszallitas, 'kiszallitasok\_id')
* belongsTo(fegyver, 'fegyverek\_id')
* belongsTo(kiegeszito, 'kiegeszitok\_id')

Minden kapcsolat withDefault()-tal van ellátva, így üres érték esetén is működő objektumot ad vissza.

**Adatbázis struktúra és migrációk**

Az alábbi szekció bemutatja az alkalmazás adatbázis-struktúráját és a hozzá tartozó Laravel migrációs fájlok logikáját. A migrációk felelősek az adatbázistáblák létrehozásáért, a mezők definiálásáért, valamint az idegen kulcsos kapcsolatok kialakításáért.

1. **felhasznalok tábla**



36. Ez a tábla tartalmazza a felhasználók alapadatait. Opcionálisan megadható profilkép, születési dátum és regisztráció ideje.

1. **fegyverek tábla**

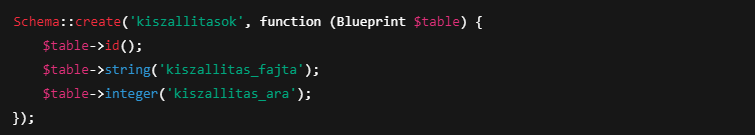
37. kép: A táblában rögzítésre kerül minden releváns fegyverparaméter, valamint a tulajdonos idegen kulccsal

1. **kiegeszitok tábla**



38. kép: A kiegeszitok tábla működése hasonló a fegyverek táblához, de külön kezeli a kiegészítőket. Itt is szerepelnek képek, leírás és tulajdonos.

1. **kiszallitasok tábla**



39. kép: Ez a tábla tartalmazza a szállítási típusokat és azok árait.

1. **rendelesek tábla**



40. kép: A rendelesek tábla köti össze a különféle entitásokat: felhasználó, termékek (fegyver/kiegészítő), szállítás. Emellett tartalmaz minden szükséges szállítási adatot.

**Összegzés**

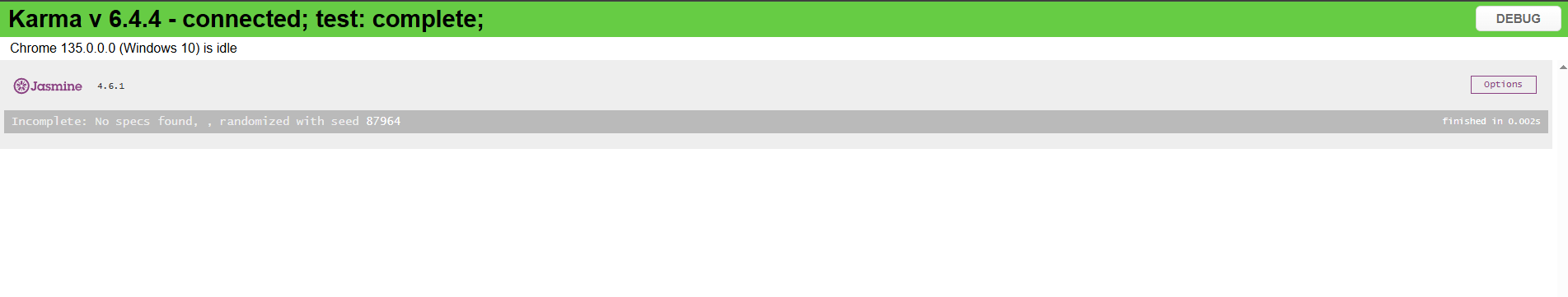
A migrációs struktúra jól követi a Laravel konvencióit:

* minden entitás külön táblában
* idegen kulcsos kapcsolatok explicit megadva
* opcionális mezők nullable-ként definiálva
* dátumok típusosan kezelve

**Tesztelés**

**Backend tesztelés**

**Frontend tesztelés**



**Továbbfejlesztési lehetőségek**

Bár az AirSoftMarket webalkalmazás jelenlegi formájában is funkcionálisan teljes, számos olyan bővítési és fejlesztési lehetőség kínálkozik, amelyek tovább növelhetik a rendszer használhatóságát, biztonságát és felhasználói élményét.

**1. Felhasználói jogosultságok kezelése**

Jelenleg minden regisztrált felhasználó azonos jogkörrel rendelkezik. Egy későbbi fejlesztés során érdemes lehet adminisztrátori szerepkör bevezetése, amely lehetővé tenné termékek, rendelések, vagy felhasználók moderálását.

* Admin-panel kialakítása (külön route és komponens)
* Jogosultság-ellenőrzés middleware szinten (Laravel-ben)

**2. Termékkereső és szűrők**

A termékkínálat növekedésével hasznos lenne egy részletes kereső- és szűrőrendszer:

* Keresés név, ár, típus szerint
* Ár szerinti szűrés (pl. csúszkával)
* Csak elérhető (nem eladott) termékek szűrése

**3. Értesítések és visszajelzések**

A felhasználói élmény javítása érdekében implementálható lenne egy értesítési rendszer:

* Toast-üzenetek sikeres műveletek után (pl. vásárlás sikeres)
* E-mail értesítések új rendelések vagy fiók aktivitás esetén
* Profilkép feltöltéskor azonnali visszajelzés

**4. Fizetési integráció**

Jelenleg kizárólag utánvétes fizetés támogatott. A későbbi bővítés során érdemes lehet online fizetési megoldásokat is integrálni:

* PayPal, Stripe vagy SimplePay integráció
* Fizetési státuszok kezelése adatbázisban

**5. Reszponzivitás és PWA támogatás**

Bár az alkalmazás jelenleg reszponzív, a teljes Progressive Web App (PWA) támogatás további előnyöket biztosítana:

* Offline működés alapvető funkciókkal
* „Add to Home Screen” funkció mobileszközökön
* Gyorsabb betöltés és asset-cache kezelés

**6. Képfeltöltés optimalizálása**

A jelenlegi képfeltöltés funkció működik, de további fejlesztések javíthatják a teljesítményt és a felhasználói élményt:

* Képek tömörítése feltöltés előtt (frontend oldalon)
* Előnézeti kép megjelenítése feltöltés előtt
* Maximális fájlméret és formátumellenőrzés

**7. SEO és Metaadatok javítása**

* A webalkalmazás Google-barátabbá tétele érdekében érdemes lenne:
* Oldalonként dinamikus meta title és description beállítása
* Canonical linkek és SEO-barát URL-ek alkalmazása

**8. Biztonság és adatvédelem**

A biztonsági szint növelése elengedhetetlen lenne egy éles környezethez:

* JWT token-alapú hitelesítés fejlesztés
* CSRF védelem backend oldalon (Laravel alapértelmezett)
* HTTPS kötelezővé tétele éles szerveren
* Rate limiting a kritikus API végpontokon

**Irodalomjegyzék**

* <https://v17.angular.io/docs>
* <https://laravel.com/docs/12.x>
* <https://nodejs.org/docs/latest/api/>
* <https://www.php.net/docs.php>
* <https://dev.mysql.com/doc/>
* <https://getbootstrap.com/docs/5.3/getting-started/introduction/>