



Vas Megyei Szakképzési Centrum  
Nádasy Tamás Technikum és Kollégium

# **PROJEKTFELADAT**

## **HS Market**

**Horváth Bálint**

**Szenti Ádám Patrik**

**Konzulens:  
Balics Gábor**

**2025**

# Nyilatkozat

Alulírott, Horváth Bálint és Szeni Ádám Patrik, kijelentjük, hogy a HS Market című projektfeladat kidolgozása a saját munkánk, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtuk fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeink saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudomásunk szerint hitelesek.

Csepreg, 2025. 04. 01.

---

Hallgató

---

Hallgató

# Kivonat

## HS Market

Szakdolgozatunk egy olyan webes alapú szoftvert mutat be, mely kifejezetten a számítógépes alkatrészek értékesítésére készült. A HS Market egyszerű és gyors vásárlási lehetőséget biztosít a felhasználók számára, akik könnyedén megtalálhatják a számukra legalkalmasabb terméket.

A szoftverünk egy modern megjelenést nyújtó online vásárlási platform, mellyel a vásárlók nem csak időt takaríthatnak meg a megfelelő termék kiválasztásában, hanem információval is bővíthetnek az adott termék kapcsán a részletes termékleírásnak köszönhetően.

A felhasználóbarát és interaktív felhasználói felületünk biztosítja a látogatóink számára, hogy a keresendő terméket a lehető leghamarabb megtalálják. Az oldalunk különböző szűrési lehetőségeket kínál, amely nem csak a termékkategóriánkénti keresést teszi lehetővé, hanem ártartomány, gyártó illetve különböző szempontok, például ár és betűrend szerinti rendezés, segíti a vásárlót a számára legalkalmasabb termék kiválasztásában.

# **Abstract**

## HS Market

Our thesis presents a web-based software designed specifically for the sale of computer hardware. HS Market provides users with a simple and quick shopping experience, allowing them to easily find the most suitable products for their needs.

The software offers a modern-looking online shopping platform, where customers can not only save time in choosing the right product but also gain valuable information about the product through detailed descriptions.

Our user-friendly and interactive interface ensures that visitors can quickly find the product they are searching for. The site offers various filtering options, enabling searches by product category, price range, manufacturer, and other criteria such as price and alphabetical order, helping customers select the most suitable product for their needs.

# Tartalomjegyzék

1.	Bevezetés.....	8
2.	Hasonló Weblapok.....	10
3.	Felhasználói dokumentáció .....	11
3.1.	Hardware és Software igény.....	12
3.2.	Főoldal és tartalma .....	12
3.3.	Funkciók ismertetése .....	14
3.3.1.	Regisztráció/Bejelentkezés.....	14
3.3.2.	Termékek oldal .....	14
3.3.3.	Részletes termékoldal.....	15
3.3.4.	Kosár.....	16
3.3.5.	Stripe fizetés .....	17
3.3.6.	Vezérlőpult <i>guest</i> szerepkörrel.....	18
3.3.7.	Vezérlőpult admin szerepkörrel .....	18
3.3.8.	Értékelések.....	20
3.3.9.	Egyéb funkciók.....	21
4.	Fejlesztői dokumentáció .....	23
4.1.	Prisma ORM (Object Relational Mapping).....	23
4.2.	Adatbázis táblái és kapcsolatai .....	23
4.2.1.	product tábla .....	24
4.2.2.	imageurl tábla .....	24
4.2.3.	user tábla.....	25
4.2.4.	cart tábla .....	25
4.2.5.	cartitem tábla .....	25
4.2.6.	coupon tábla.....	26

4.2.7.	review tábla.....	26
4.3.	Szerver-Kliens oldali kapcsolatok (API).....	27
4.3.1.	API működése .....	27
4.3.2.	Postman .....	28
4.4.	Autentikáció és middleware .....	28
4.4.1.	JSON Web Token (JWT) .....	28
4.5.	Stripe integráció.....	29
4.6.	Adatbiztonság .....	29
4.6.1.	Szemponatok .....	30
5.	UX (User Experience) és UI (User Interface) .....	31
5.1.	Tervezés .....	31
5.2.	Komponensek .....	31
5.2.1.	Orange-/BlueButton .....	31
5.2.2.	Cart .....	32
5.2.3.	LoggedValidation és LoggedInButtons .....	32
5.2.4.	Payment .....	32
5.2.5.	ProductCard .....	32
5.2.6.	ProductSwiper .....	33
5.2.7.	Recommendation .....	33
5.2.8.	RecommendationCard .....	33
5.2.9.	Reviews, ReviewCard .....	33
5.2.10.	SearchButton .....	34
6.	Fejlesztési lehetőségek .....	35
6.1.	Egyedi fizetési rendszer elkészítése .....	35
6.2.	Számítógépösszerakó.....	35
6.3.	Hűségprogram .....	35

6.4.	Termékösszehasonlítás .....	35
6.5.	Termékenkénti vélemény .....	36
6.6.	Ügyfélszolgálat rendszer kialakítása .....	36
7.	Összegzés .....	37
8.	Következtetés.....	39
9.	Irodalomjegyzék .....	40

# 1. Bevezetés

A 2025-ös záróvizsgánk beadandó témájának egy olyan webes alkalmazás kidolgozását választottunk, amely kifejezetten a számítógép-alkatrészek eladását helyezi középpontba. Célunk, hogy a vásárlók szabadon böngézhessenek a könnyen kezelhető felhasználói felületünkön és egy gördülékeny vásárlási procedúrát vigyenek véghez, bármilyen fennakadás nélkül.

Választásunk azért esett egy ilyen projektre, mivel Magyarországon kevés hasonló weboldallal lehet találkozni. Kettő kiemelkedő magyar oldal van, ami hasonló témakörben működik, viszont a felhasználói felületük és felépítésük a funkcionalitást helyezi előtérbe, nem pedig a felhasználói élményt.

Ha a felhasználó úgy dönt, hogy nem szeretne az oldalra beregisztrálni az oldalon továbbra is tud határtalanul keresgélni, viszont a vásárláshoz a regisztráció nélkülözhetetlen.

Amennyiben a felhasználó úgy dönt, hogy beregisztrál, a kosár funkciója és a vásárlás lehetősége megnyílik számára. A felhasználónak megadjuk a lehetőség a felhasználónév és az email cím módosítása mellett, a korábbi rendelések nyomon követésére is.

Megfelelő jogosultsággal könnyedén kilistázhatjuk a korábbi megrendeléseket, az oldalon megtalálható termékeket, beregisztrált felhasználókat és a létrehozott kuponokat.

Közvetlenül a felületen új termék létrehozására és a módosításra is van lehetőség, továbbá lehetőség van a felhasználónak a szerepkörét állítani, vagy esetleg egyedi kupont létrehozni, mellyel a vásárlók kedvezményesen vásárolhatnak.



A szoftver elkészítéséhez Next.js keretrendszer programozási ismereteinket és MySQL adatbázis-kezelő rendszer készségeinket használtuk fel.

- Az alkalmazáshoz Visual Studio Code 1.95 később 1.97-es verziójának programozási környezetet használtunk és Next.js 15.0.2-es keretrendszert hozzá.
- Laragon a lokális környezet kialakításához.
- Az adatbázis tervezéshez és annak elkészítésében a HeidiSQL segített.
- Az adatok tárolása pedig egy MySQL szerverén keresztül történt.
- API hívások teszteléséhez Postman nyújtott segítséget.

## 2. Hasonló Weblapok

Számos hasonló weblap található meg az interneten hasonló témában, viszont nagyrészüket saját tapasztalatból is kiindulva, gyakran bonyolult és átláthatatlan rendszerekkel rendelkeznek. Ezért is hoztunk létre egy felhasználóközpontú áruházat, ami kielégíti a felhasználók igényeit.

Magyar szinten kettő weblapot emelnék ki, ami hasonló célt szolgál:

pcx.hu és ipon.hu: Mindkettő egy népszerű elektronikai webáruház, ahol számítógépes alkatrészekről kezdve háztartási termékek is megtalálhatóak, viszont a felhasználói felületük inkább a funkcionalitást részesíti előnyben, mint pedig a megjelenést.

Világi távlatban gondolkodva a legkiemelkedőbb:

newegg.com: Főként Amerikában van jelentős népszerűsége, viszont Európában nem érhető el és magyar nyelv sem támogatott az oldalon. Letisztultabb külsővel rendelkezik, mint a pcx vagy az ipon, viszont első látásra bonyolult lehet.

### 3. Felhasználói dokumentáció

Az oldal fő célja, hogy a lehető legjobb felhasználói élmény biztosítása mellett könnyedén tudjanak a felhasználók vásárlást végrehajtani.

Lehetőségeink nem regisztrált felhasználóként:

- termékek közötti böngészés

Lehetőségeink regisztrált felhasználóként:

- minden nem regisztrált felhasználói funkció
- kosár használata
- vásárlások lebonyolítása
- saját profil szerkesztése
- saját rendelések megtekintése

Lehetőségeink admin szerepkörrel rendelkező felhasználóként:

- minden regisztrált felhasználói funkció
- összes termék listázása, létrehozása, módosítása
- felhasználói szerepkörök módosítása
- összes rendelés listázása
- kupon létrehozása

### 3.1. Hardware és Software igény

A weboldal futtatásához hálózati kapcsolat szükséges.

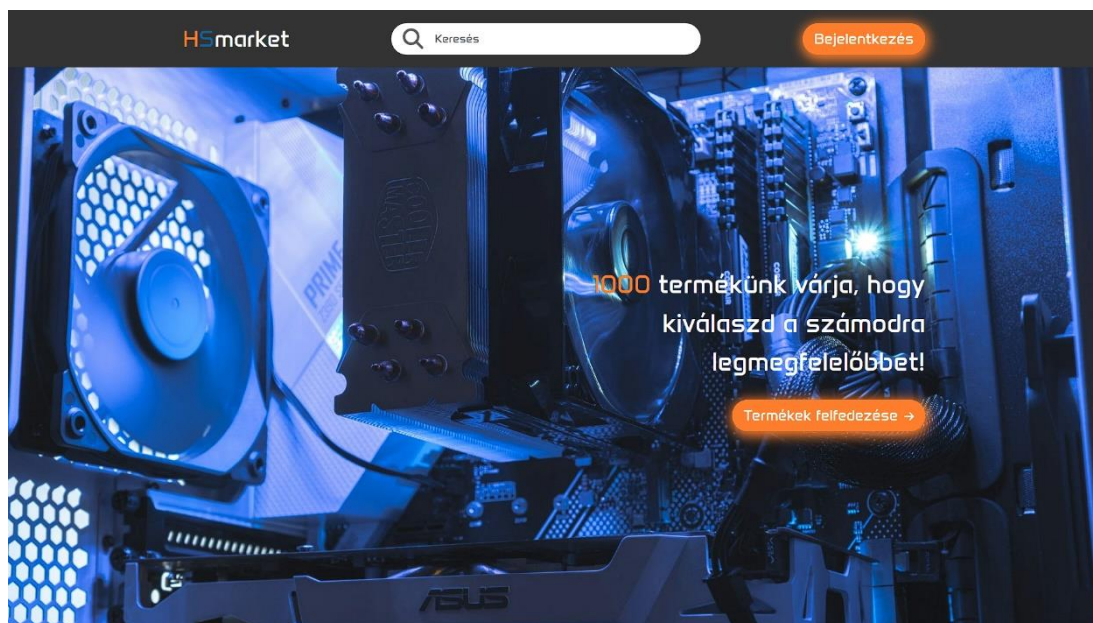
Tovább a következő specifikáció ajánlott:

- Node verzió: minimum 20.17.0-ás
- MySQL verzió: 8.0.30
- Böngészőként az alábbi verziók ajánlottak:
  - Chrome 60+
  - Firefox 60+
  - Safari 12+
  - Edge 79+
  - Internet Explorer nem támogatott

### 3.2. Főoldal és tartalma

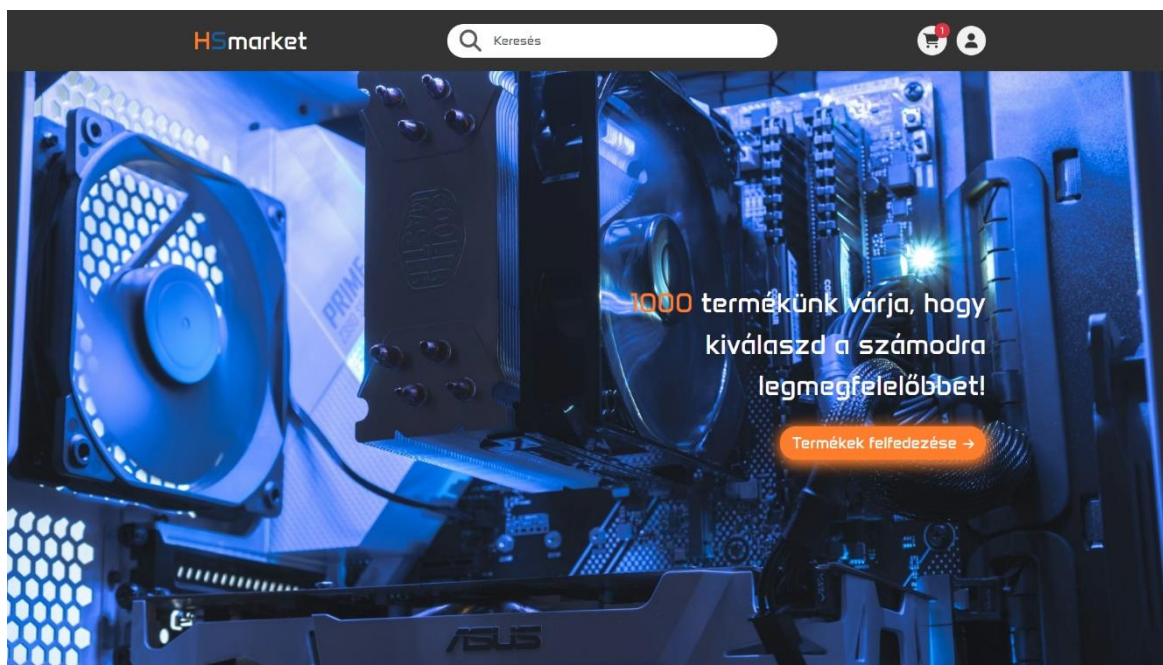
A főoldal minden felhasználó számára megegyezik, kivéve a navigációs sáv, amit bejelentkezett és nem bejelentkezett felhasználók között különböztetünk meg.

Mivel regisztráció nélkül a felhasználóinak nincsen saját profilja és kosara, így ennek a helyére egy bejelentkezés gomb került.



*1.ábra a nem bejelentkezett felhasználói felület*

Amennyiben a felhasználó belépett a fiókjába a bejelentkezés gomb helyére egy profil és egy kosár ikon jelenik meg.



*2.ábra a bejelentkezett felhasználói felület*

### 3.3. Funkciók ismertetése

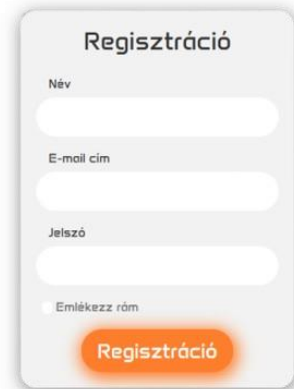
#### 3.3.1. Regisztráció/Bejelentkezés

A teljes felhasználói élmény biztosítása érdekében javasoljuk az oldalra történő regisztrálást.

Regisztráláskor egy felhasználónevet, email címet és egy jelszót kell megadnia a felhasználónak.

Bejelentkezéskor pedig elég az email cím és a jelszó.

Amennyiben úgy dönt a felhasználó, hogy szeretne bejelentkezve maradni, lehetősége van rá, hogy 2 hétig megmaradjanak a bejelentkezési adatai, ezt az *Emlékezz rám* gomb megnyomásával tudják megtenni viszont, ha úgy dönt, hogy nem akkor 24 óra után automatikusan újra be kell jelentkeznie.

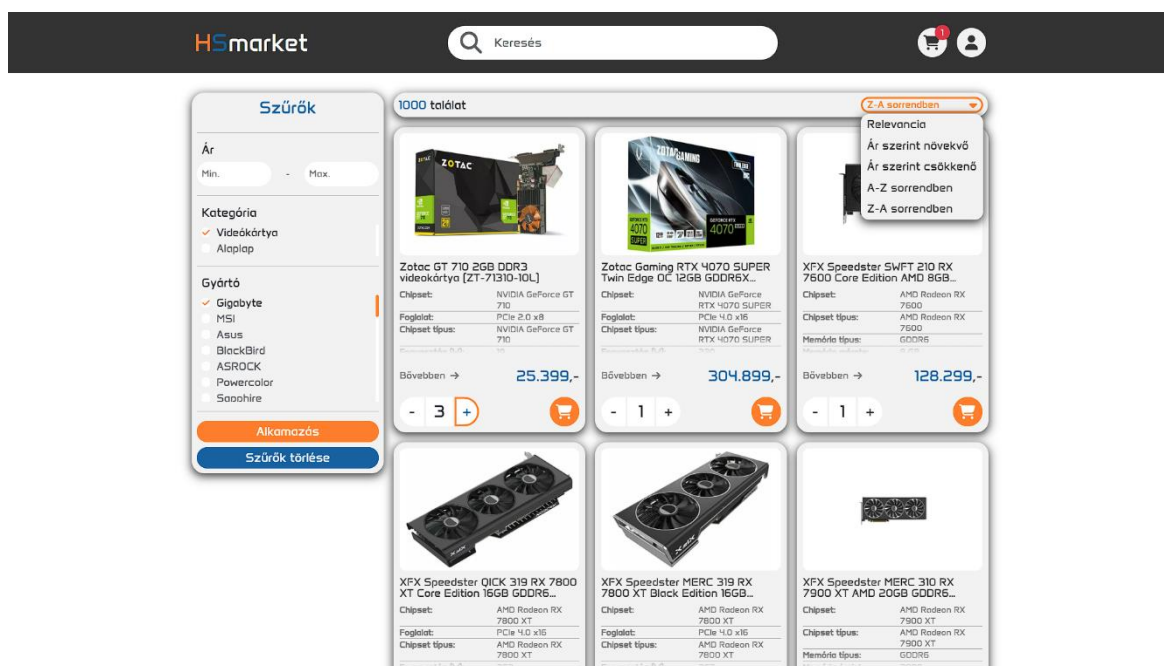


3.ábra a regisztrációs panel

#### 3.3.2. Termékek oldal

Ezen az oldalon az összes termék (vagy ha keresési paramétert adott meg, akkor a keresett kulcsszóra talált termékek) listázása látható. 3 féle szűrési lehetőséget kínál az oldalunk, ár tartomány alapú, kategória, illetve gyártó szerinti keresés. Ezen felül relevancia, ár és betűrend szerinti rendezést is biztosítunk a felhasználóink számára.

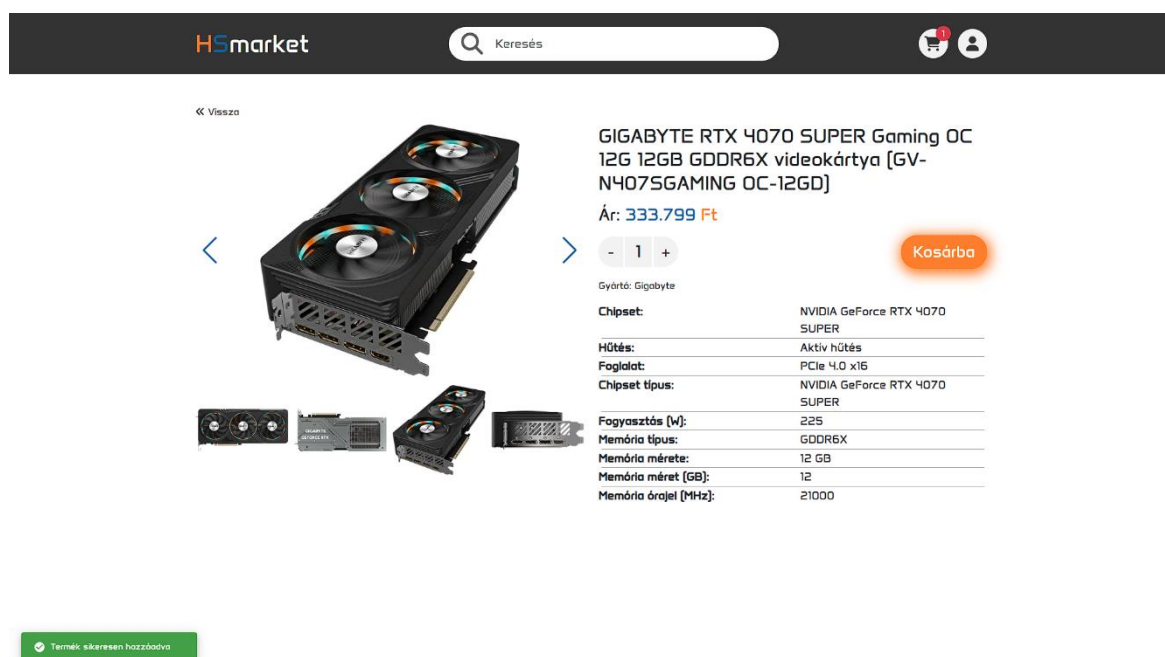
Külön kiemelném, hogy amennyiben kiválasztásra kerül egy termékkategória, dinamikus frissül a gyártói szűrőnk is figyelembe véve a termékkategóriát (ez fordítva is igaz), ezáltal megakadályozva, hogy esetlegesen olyan termékkategória és gyártó legyen kiválasztva amire nincsen keresési találat. Itt lehetőséget kínálunk, amennyibe a felhasználó bejelentkezett, hogy kosárba tegye a számára megtetszett terméket és állítsa a kívánt mennyiséget. Viszont, ha nem győződött meg a termékről, akkor a részletes oldalán, amit a *Bővebben* gomb megnyomása után elér, további betekintést nyerhet.



4.ábra a termékek oldalunk néhány szűrővel

### 3.3.3. Részletes termékoldal

Amennyiben a felhasználó úgy dönt, hogy szeretne többet megtudni a termékről, a kép vagy termék neve vagy bővebben gomb megnyomása után átirányítjuk a termék aloldalára, ahol a termék specifikus tulajdonságai jelennek meg. Ezen az oldalon megtalálható a termék összes adata, képe és ára. Itt is bele tudja tenni a kosárba és állítani tudja a mennyiséget, ha esetleg többet szeretne vásárolni. Ha úgy dönt, hogy meg szeretné vásárolni a terméket akkor a *Kosárba* gomb lenyomása után a bal alsó sarokban egy visszajelzést kap a folyamatról, hogy sikerült vagy esetleg hibába ütközött a kosárba helyezés során. Továbbra is a folyamat felhasználói fiókhoz kötött!

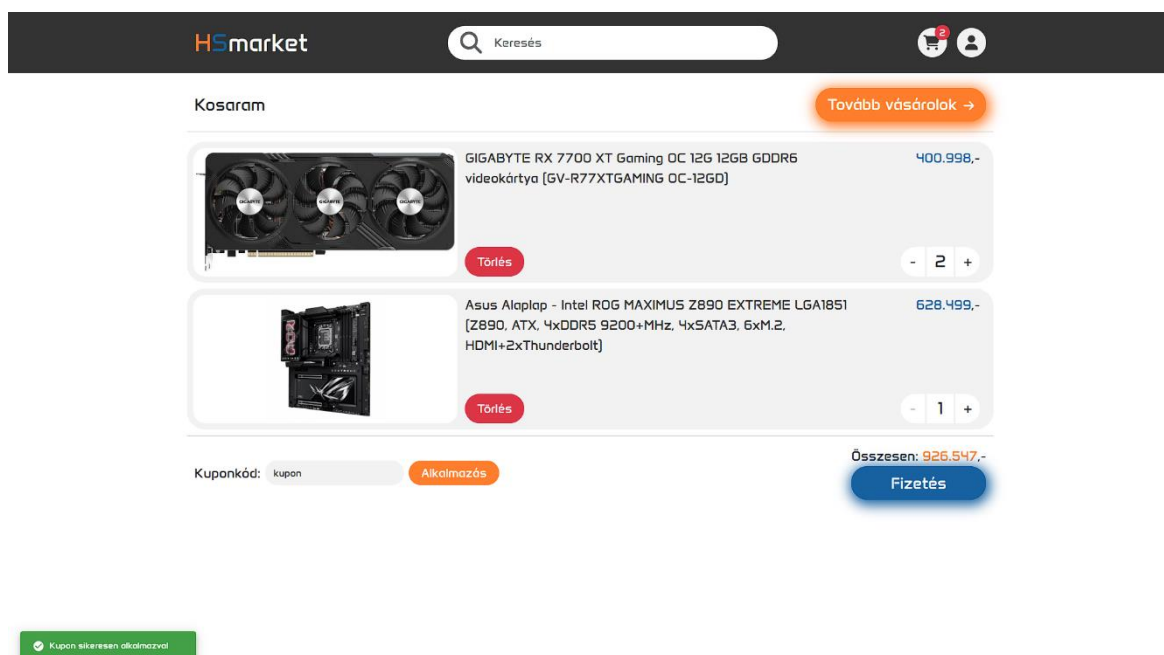


5.ábra termék részletes oldala, szimulálva egy kosárba helyezést

### 3.3.4. Kosár

Ezen az oldalon az adott felhasználó kosara látható, ahol szükség esetén tudja a termék mennyiségét módosítani vagy esetleg eltávolítani a kosárból a termékeket. Kuponkód megadására is van lehetőség a kedvezményes vásárlás lebonyolításához. Fizetés gomb megnyomása után a stripe segítségével történik a vásárlás.

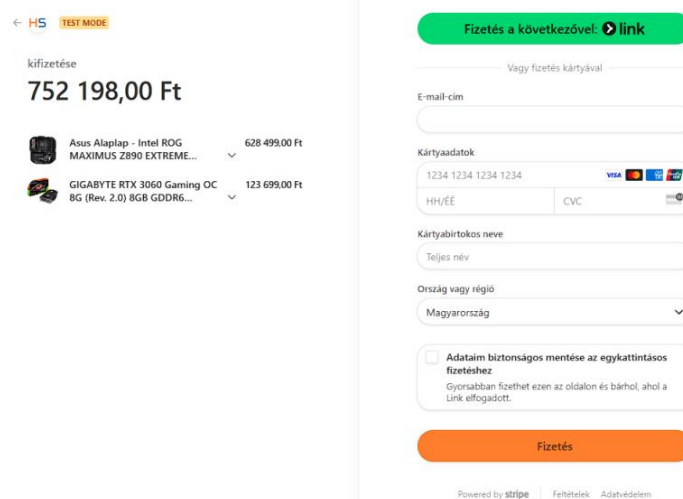




6.ábra kosár oldal kuponkód alkalmazásával

### 3.3.5. Stripe fizetés

Az oldalunkon a stripe segít a fizetés lebonyolításában. *Fizetés* gomb megnyomása után, egy integrációs felület fogad minket, ahol email cím megadása után bankkártya adatok kell megadni. Fizetés szimulációhoz a stripe biztosít bankkártya adatokat, ezért fejlesztői környezetbe is könnyedén le lehet tesztelni. Sikeres fizetés lebonyolítása után a *sikeres* oldalra kap átirányítást.



### 3.3.6. Vezérlőpult *guest* szerepkörrel

A *vezérlőpult/profil* oldalon a saját profilját tekintheti meg a felhasználó. Lehetősége van módosítani a profilképét, a felhasználónevét és az email címét. Jobb oldalon pedig a korábbi rendelései láthatóak az adott felhasználónak, ezáltal könnyen meg tudja nézni a korábbi rendeléseit egy részletes rendelésinformációnak köszönhetően. Vezérlőpult oldalon bal oldalt navigációs sáv található. Ez a rész eltér egy sima regisztrált felhasználó és az admin jogosultsággal rendelkező felhasználók között. Következő szekcióban kitérek hogyan néz ki és milyen funkciókkal rendelkezik egy admin.

The screenshot shows the HSmarket user interface. On the left is a dark sidebar with navigation links: 'Profil', 'Értékelés', and 'Kijelentkezés (-)'. The main content area has a header with the 'HSmarket' logo, a search bar, and user icons. Below the header, the user's profile is shown with a circular profile picture of a dog, the name 'guest', and the email 'guest@guest.guest'. To the right, under the heading 'Korábbi rendelések', there is a list of five orders. Each order card displays an ID, a status (either 'paid' in green or 'unpaid' in orange), a date (all are '2025. 03. 07.'), and a list of products. The products listed are 'GIGABYTE RTX 3060 Gaming OC 8G [Rev. 2.0] 8GB GDDR6 videokártya [GV-...]'. The total amount for the first three orders is 'Összeg: 123 699.-', and for the last two is 'Összeg: 618 495.-'.

7.ábra saját profil sima felhasználóval

### 3.3.7. Vezérlőpult admin szerepkörrel

Hasonlóan, mint egy átlagos felhasználó, tudja módosítani a profilképét, a felhasználónevét és az email címét, valamint láthatja a saját rendeléseit, amennyiben van neki. A navigációs sáv 3 új opcióval bővül. Termékek, felhasználók és kuponok. A termékek oldalon listázásra kerül az oldalon megtalálható összes termék és tulajdonságaik, melyeket szabadon módosíthatóak, törölhetőek és hozzáadhatóak. Ugyan ez igaz a felhasználói oldalra is, viszont a biztonság érdekében csak törlés és felhasználói szerepkör módosításra van lehetőség. Kuponok oldalon lehet létrehozni és törölni a kuponokat. A következő mezőket kell megadni egy sikeres kupon létrehozásához: kupon neve, kedvezmény (%-ban), minimum összeg, lejárat dátum (opcionális) és felhasználási limit.

				<input type="text" value="Keresés"/>			
Profil	ID	Név	Ár	Kategória			
Értékelés	1	GIGABYTE RTX 3060 Gaming OC 8G [Rev. 2.0] 8GB GDDR6 videokártya [GV-N3060GAMING OC-8GD 2.0]	123.699.-	Videokártya			
Termékek	2	MSI RTX 3060 VENTUS 2X 12G OC videokártya [V397-022R]	133.400.-	Videokártya			
Felhasználók	3	GIGABYTE RX 7700 XT Gaming OC 12G 12GB GDDR6 videokártya [GV-R77XTGAMING OC-12GD]	200.499.-	Videokártya			
Megrendelések	4	Gigabyte GeForce RTX 4070 Ti SUPER GAMING OC nVidia 16GB DDR6X 256bit PCIe videokártya GV-N407T...	433.399.-	Videokártya			
Kuponok	5	GIGABYTE RTX 4060 Ti Eagle OC 8G 8GB GDDR6 videokártya [GV-N406TEAGLE OC-8GD]	198.899.-	Videokártya			
Kijelentkezés (+)	6	Gigabyte RTX 3050 WINDFORCE OC V2 8G GV-N3050WF2OCV2-8GD	88.400.-	Videokártya			
	7	ASUS Videokártya PCI-Ex16x nVIDIA RTX 4060 8GB DDR6 OC 90Y0JC7-MONAOO	143.999.-	Videokártya			
	8	MSI RTX 4060 Ti Ventus 2X Black 8G OC 8GB GDDR6 videokártya [V51S-017R]	188.299.-	Videokártya			
	9	GIGABYTE RTX 4060 Windforce OC 8G 8GB GDDR6 videokártya [GV-N4060WF2OC-8GD]	141.199.-	Videokártya			
	10	BLACKBIRD Videokártya PCI-Ex16x nVIDIA GT 1050 4GB DDR5 BH1568	46.098.-	Videokártya			
	11	ASRock Radeon RX 6600 Challenger D 8GB videokártya [RX6600 CLD 8G]	100.700.-	Videokártya			
	12	VGA PowerColor AMD RX 7900 XTX Hellhound 24GB GDDR6 - RX7900XTX 24G-L/OC	410.500.-	Videokártya			
	13	GIGABYTE RTX 4070 Ti SUPER Eagle OC 16G 16GB GDDR6X videokártya [GV-N407TSEAGLE OC-16GD]	450.499.-	Videokártya			
	14	MSI RTX 4060 Ventus 2X Black 8G OC 8GB GDDR6 videokártya [V516-004R]	145.499.-	Videokártya			
	15	Sapphire RX 7800 XT Pulse Gaming 16GB GDDR6 videokártya [11310-02-20G]	230.798.-	Videokártya			
	16	GIGABYTE RTX 3060 Gaming OC 12G 2.0 12GB GDDR6 videokártya [GV-N3060GAMING OC-12GD 2.0]	138.299.-	Videokártya			
	17	SUPERMICRO kábel GPU Power 8-pin to 8-Pin 20cm CBL-PWEX-IT48-20	6.999.-	Videokártya			
	18	GPU I fűvezeték és Thermal Grease hővezető GPU 1x 12V4PWR normál TG-kábel-HIN	22.998.-	Videokártya			

8.ábra admin termékek oldal

Profil

Értékeles

Termékek

Felhasználók

Megrendelések

Kuponok

Kijelentkezés (+)

HSmarket

Keresés

Kuponkód

Kedvezmény [%]

Minimum összeg

Lejárat dátum

Felhasználási limit

1

0

ééé. hh. nn.

1

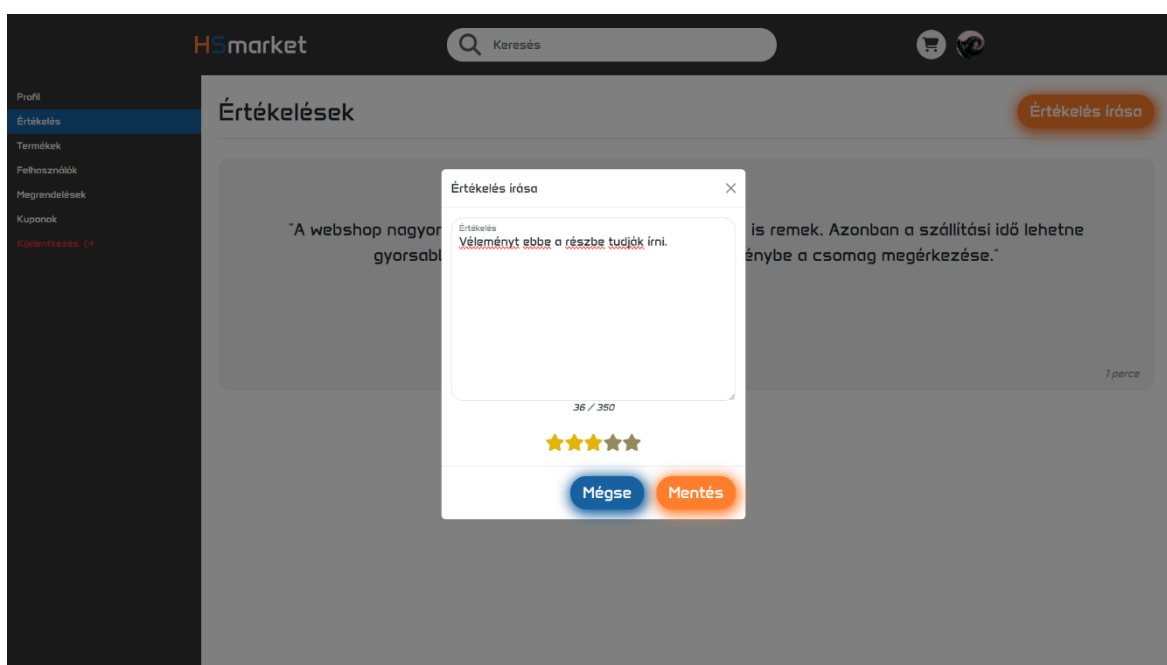
Létrehozás

Kód	Százalék	Minimum ár	Lejárat dátum	Eddig felhasználva
test	10%	0 Ft	Nincs	0/10
kupon	5%	20000 Ft	2025. 05. 01.	0/100

9.ábra kuponok oldal

### 3.3.8. Értékelések

A vezérlőpult oldalon a navigációs sávban elérhető az értékelések menüpont. A felhasználók értékelést tudnak írni az oldal megelégedettségével kapcsolatban. Továbbfejlesztési lehetőségként ezt le lehetne bontani termékenként is. Maximum 350 karakterből álló véleményt tudnak írni, melyről folyamatos információt kapnak hány karakternél tartanak. Ezután már csak csillagot kell megadniuk, hogy összességében milyenre értékelik az oldalt. Bármennyi véleményt létre lehet hozni és törölni is. Amennyiben létrehozták a véleményt sikeresen a főoldalon található vélemény szekcióban esély van rá, hogy megtalálhatja a saját véleményét. Az összes vélemény közül véletlenszerűen kiválaszt 6 darabot és ezek a vélemények fognak megjelenni. Minden oldal újratöltésnél újra választja ki a 6 véleményt.



9.ábra értékelés létrehozása

Mások mondták rólunk...

"Nagyon elégedett vagyok a vásárlással. Minden tökéletesen működik, és az árak is nagyon versenyképesek. Csak javasolnám, hogy a termékleírásokat egy kicsit részletesebben írják meg, de ezen kívül teljesen hibátlan."

★★★★★



- Varga Zoltán

28/08/2024




*10.ábra főoldalon található vélemények szekció*

### 3.3.9. Egyéb funkciók

Továbbiakban megtalálható egy termék ajánlás rész is, ahol véletlenszerűen jelenik meg 10 termék, mindegyik kategóriából 2-2 darab. Keresni is lehet termékekre név alapján. A sok hasznos funkciók között megtalálható az *ugrás az oldal tetejére* gomb, ami felvisz az oldal tetejére, ezáltal is törekedve a felhasználói élmény fokozására. Ha meg vannak elégedve az oldallal a felhasználók, véleményt is tudnak írni, ami a főoldalon jelenik meg egy külön szekcióba.

Termékek, amik tetszhetnek...



Intel Core i3-10100 processzor  
[BX8070110100]

Cache:	6 MB
Foglalat:	LGA1200
Alap órajel:	3.6 GHz

Bővebben →


-

1

+

Kosárba

47.699.-



XFX Speedster QICK 319 RX 7800 XT  
Core Edition 16GB GDDR6 videokártya...

Chipset:	AMD Radeon RX 7800 XT
Foglalat:	PCIe 4.0 x16
Chipset típus:	AMD Radeon RX 7800 XT

Bővebben →


-

1

+

Kosárba

223.699.-



Gigabyte Z790 S DDR4

WiFi:	Nem
Chipset:	Intel Z790
Foglalat:	LGA1700

Bővebben →

-

1

+

Kosárba

74.499.-

<

>

↑

10.ábra termékajánlások szekció

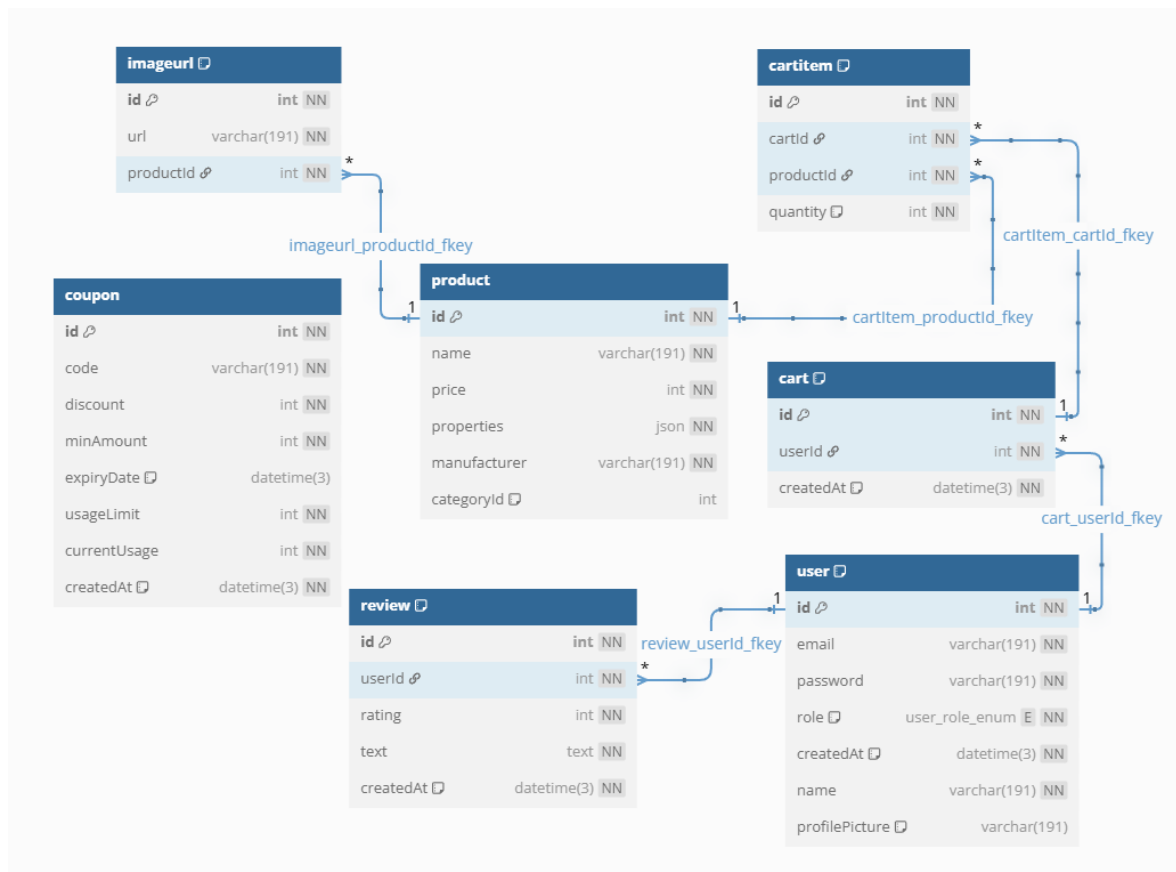
## 4. Fejlesztői dokumentáció

Adatbázisunkhoz MySQL-t használtunk és HeidiSQL adatbázis kezelőt, ami a Laragon-nak az előretelepített adatbázis kezelő rendszere.

### 4.1. Prisma ORM (Object Relational Mapping)

A projekt során a Prisma ORM (Object-Relational Mapping) eszközt használtuk az adatbázisok kezelésére. Lehetővé teszi az egyszerű és típusbiztos kapcsolatot az oldal és az adatbázis között és könnyebbé teszi az adatbázis műveletek kezelését. Típusbiztonságnak köszönhetően az adatbázishoz kapcsolódó összes művelet ellenőrizhető típus alapján és ez miatt az emberi hiba is könnyen észrevehető. Migrációk egyszerűen nyomon követhetőek, ahol láthatjuk az adatbázis struktúra változásait.

### 4.2. Adatbázis táblái és kapcsolatai



11.ábra az adatbázis táblák és a kapcsolatok

A 11.ábrán látható az adatbázisunk felépítése. 7 darab táblából és 5 kapcsolatból áll. Az adatbázisnak futnia kell lokális környezetben, `npm run db:setup` parancs megcsinál

mindent, ami az adatbázishoz köthető, ezáltal a teszteléshez semmi mást nem kell tenni. Néhány tábla fel van töltve adatokkal, hogy teszteléskor minden funkciót teljeskörűen ki lehessen próbálni, de természetesen manuálisan is létrehozható minden az oldalon, ezáltal nem kell az adatbázisban bajlódni.

#### 4.2.1. product tábla

→ Termékek tárolására

Mezőnév	Típus	Megjegyzés
<b>id</b>	INT, PK, NN, AI	Egyedi azonosító
<b>name</b>	VARCHAR, NN	Termék neve
<b>price</b>	INT, NN	Termék ára
<b>properties</b>	JSON, NN	Termék tulajdonságai
<b>manufacturer</b>	VARCHAR, NN	Termék gyártója
<b>categoryId</b>	INT, NULLABLE	Termék kategóriája

#### 4.2.2. imageurl tábla

→ Termékek képeiért felel

Mezőnév	Típus	Megjegyzés
<b>id</b>	INT, PK, NN, AI	Egyedi azonosító
<b>url</b>	VARCHAR, NN	Termékhez tartozó kép
<b>productId</b>	INT, FK, NN	Termék id-ja



### 4.2.3. user tábla

→ Itt tároljuk el a felhasználóknak az információt

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
email	VARCHAR, NN, UK	Felhasználó email címe
password	VARCHAR, FK, NN	Felhasználó jelszava
role	ENUM, NN, def(guest)	Felhasználó szerepköre (alapértelmezz a guest)
createdAt	DATETIME(NOW), NN	Profil létrehozási dátuma
name	VARCHAR, NN	Felhasználónév
profilePicture	VARCHAR, NULLABLE	profilkép (url)

### 4.2.4. cart tábla

→ Ebben a táblában történik a kosár tartalmának eltárolása

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
userId	INT, NN, FK	Felhasználó azonosítója
createdAt	DATETIME(NOW), NN	Létrehozási idő

### 4.2.5. cartitem tábla

→ Kosarat tartalmazó termékek eltárolására szolgál

Mezőnév	Típus	Megjegyzés
<b>id</b>	INT, PK, NN, AI	Egyedi azonosító
<b>cartId</b>	INT, NN, FK	Kosár azonosítója
<b>productId</b>	INT, NN, FK	Termék azonosítója
<b>quantity</b>	INT, def(1)	Termék mennyisége

#### 4.2.6. coupon tábla

→ Kuponok tárolása

Mezőnév	Típus	Megjegyzés
<b>id</b>	INT, PK, NN, AI	Egyedi azonosító
<b>code</b>	VARCHAR, NN	Kuponkód neve
<b>discount</b>	INT, NN	Kupon kedvezménye
<b>minAmount</b>	INT, NN	Termék mennyisége
<b>expiryDate</b>	DATETIME, NULLABLE	Lejárat dátuma
<b>usageLimit</b>	INT, NN	Felhasználási limitje
<b>currentUsage</b>	INT, NN	Jelenleg felhasznált
<b>createdAt</b>	DATETIME(NOW)	Létrehozási dátuma

#### 4.2.7. review tábla

→ Vélemény létrehozása

Mezőnév	Típus	Megjegyzés
<b>id</b>	INT, PK, NN, AI	Egyedi azonosító
<b>userId</b>	INT, NN, FK	Felhasználó azonosítója
<b>rating</b>	INT, NN	Értékelés (csillag)
<b>text</b>	TEXT, NN	Vélemény leírása
<b>createdAt</b>	DATETIME, NULLABLE	Létrehozási dátuma

## 4.3. Szerver-Kliens oldali kapcsolatok (API)

### 4.3.1. API működése

Az API (Application Programming Interface) egy olyan felület, amely lehetővé teszi, hogy külön alkalmazások is képesek legyenek egymással kommunikálni, adatot átadni.

Hogyan működik az API?

A kliens küld egy kérést az API-nak, hogy hozzáférjen egy adott információhoz, majd ezzel az információval válaszol. Ez a kérés általában egy adott URL-en keresztül történik, ami HTTP vagy HTTPS protokollt használ.

Az API a feldolgozott adatokat általában JSON vagy XML formátumban visszaküldi a kliensnek.

HTTP metódusokat használ:

- GET: információ, adat lekérése
- POST: új adatok létrehozása itt történik
- PUT: adatok módosításáért ez a metódus felel
- DELETE: Törli a kért információt, adatot

Az API-k HTTP státuszkódokon küldik a visszajelzést, amiket öt csoportra bontunk:

- Információs válaszok (100 – 199)
- Sikeres válaszok (200 – 299)
- Átírányítási üzenetek (300 – 399)
- Kliens oldali hibák (400 – 499)
- Szerver oldali hibák (500 – 599)

Az oldaunk is API-kon keresztül kommunikál, amelyek segítségével a szükséges adatokat a háttérben található helyi adatbázisból nyeri ki.

### 4.3.2. Postman

A projekt során ezt az alkalmazást használtuk az API-k működésének tesztelésére. Lehetővé tette számunkra, hogy CRUD-os műveletek (Create, Read, Update, Delete) kéréseit küldjük a végpontokra, GET, POST, PUT és DELETE formában. Az alkalmazás segítségével gyorsan és egyszerűen adhattuk meg a szükséges paramétereket, majd kapott válaszokat értelmezhetjük, hogy biztosítsuk a rendszer helyes működését. Különösen hasznos volt a hibakeresés során, mivel azonnal láthattuk, ha esetleg hiba volt a kérésben.

## 4.4. Autentikáció és middleware

Az oldal teljes funkcionalitásának biztosítása miatt, elengedhetetlen volt az autentikáció és a middlewarek használata. Sikeres bejelentkezés és regisztrációkor egy biztonsági *auth\_token* generálódik, amit *cookiban* tárolunk el. Oldalunkon *Emlékezz rám* funkció is van, amit, ha bepipálunk, akkor 14 napig megjegyzi a bejelentkezési adatokat, viszont utána mindenképp újra be kell jelentkezni. Ha nem élünk ezzel a lehetőséggel, akkor 1 nap után kéri újra a bejelentkezési adatokat. Middleware-ünk felel ezeknek a tokeneknek az ellenőrzésére és biztosítja, hogy illetéktelen felhasználók ne férjenek hozzá különböző aloldalakhoz.

### 4.4.1. JSON Web Token (JWT)

Mi is az a JWT?

Egy szabványosított adatstruktúra, melyet főként hitelesítésre használnak webes felületen. A token tartalmazza a felhasználó azonosítóját és egyéb szükséges információkat, amelyeket digitálisan aláírunk, így garantálva a token érvényességét és manipuláció elleni védelmét.

Mi is JWT-t használtunk az API végpontokhoz való hozzáférés kezelésére és a felhasználók azonosítására jogosultságaik vizsgálására.

Sikeres bejelentkezés vagy regisztráció után a *jsonwebtoken* könyvtár generál egy JWT-t és ezt fogja továbbítani API hívások során az *Authorization* fejlécben.

## 4.5. Stripe integráció

Mi az a Stripe?

A Stripe egy online fizetési megoldás, amely lehetővé teszi a webalkalmazások számára, hogy gyorsan és biztonságosan fogadjanak fizetéseket Megfelel a legszigorúbb biztonsági előírásoknak. A Stripe API-ját gyakran használják webshopok, mivel gyorsan, biztonságosan és megbízhatóan működik.

### **Frontend:**

A frontend oldalon a Stripe biztosít előre definiált UI komponenseket, amelyek segítségével könnyen integrálhatóak a fizetések. A stripe validálja a beérkező adatokat például bankkártyaszám, ezért hamis információk kezelését is megoldja nekünk. Színek és logo megadásán kívül a személyre szabhatóság korlátozott. A komponensek nem teszik lehetővé az egyedi dizájn megvalósítását.

### **Backend:**

Az API-nak köszönhetően könnyen kezelhetjük ezeket a tranzakciókat. A Stripe API dokumentációja átlátható, így könnyen implementálható, és sokoldalú megoldást biztosít a backend oldali logika felépítésére. Mivel az összes érzékeny adat, például a bankkártya adatok nem kerülnek tárolásra a saját szerveren, a Stripe biztosítja a PCI-DSS szabványoknak való megfelelést, garantálva a biztonságos adatkezelést, ezért ezzel sem kell külön foglalkozunk, mivel megoldja nekünk. Az API kulcsok kezelésére API tokeneket használtunk.

## 4.6. Adatbiztonság

A felhasználói és egyéb érzékeny adatok tárolására is figyeltünk, amelyek biztosítják a felhasználói adatok védelmét, az adatbázis integritását és néhány támadás elleni védelmet.

Prisma ORM-nek köszönhetően az SQL injection elleni védekezés alapértelmezetten be van építve, így az adatbázis lekérdezések biztonságosan kezelhetők.

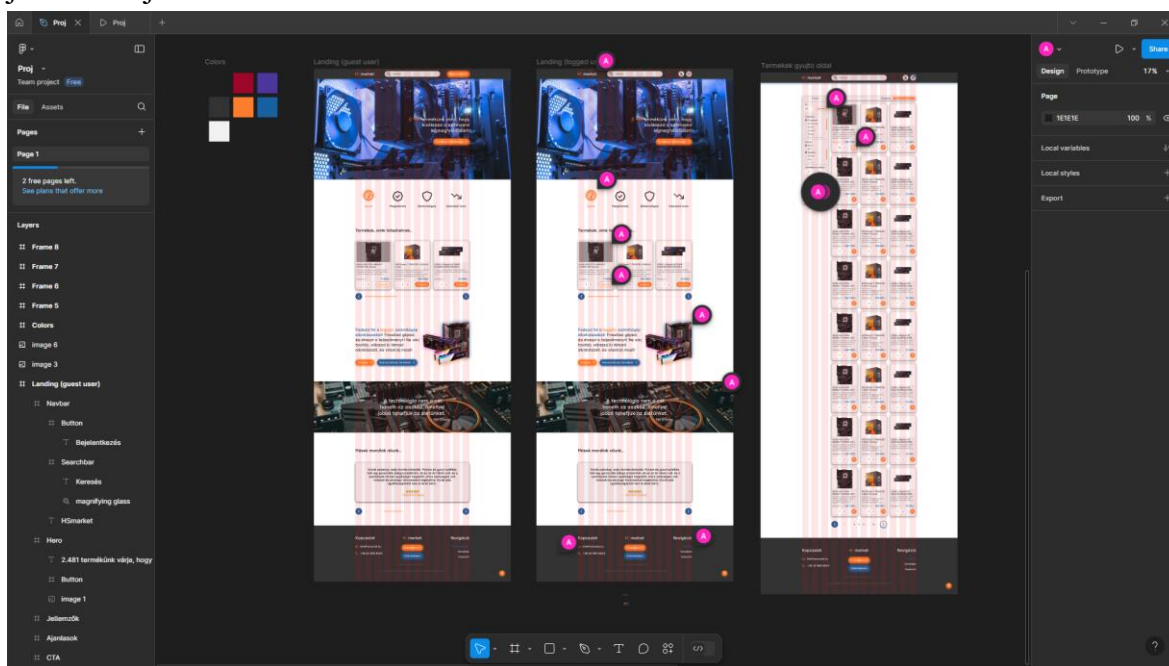
#### 4.6.1. Szempontok

Az oldalunk nagy mértékben megfelel a GDPR előírásainak. A felhasználói adatok tárolásánál a jelszó, *bcrypt* használatára kerül hashelésre, ezáltal nagy mértékben biztosítva van, hogy illetéktelenek férjenek hozzá a jelszóhoz. Adatláthatóság szempontjából az *admin* felületen kizárólag a felhasználónév, e-mail cím, profilkép (amennyiben van) és szerepkör jelenik meg, ezáltal is biztosítva, hogy érzékeny adat ne jelenjen meg. A szerepkör alapú megkülönböztetés is segít, hogy egy átlagos felhasználó ne férjen hozzá mások adataihoz.

## 5. UX (User Experience) és UI (User Interface)

### 5.1. Tervezés

A weboldal dizájnját illetően a tervezésünket egy látványterv létrehozásával kezdtük, aminek köszönhetően kaptunk egy egészen pontos képet milyen arculatot szeretnénk adni neki. Ennek megvalósításához a Figma nyújtott segítséget, amellyel egy grafikai projektben egyszerre többen is tudtunk dolgozni, valamint commenteket hagyni bizonyos részekhez, hogy milyen ötletek vannak az adott rész implementálásához vagy esetleges jövőbeli fejlesztéséhez.



### 5.2. Komponensek

A weboldal több komponensből tevődik össze azok újrahasználatossága, valamint optimalizációja miatt. Ezek egy része úgy nevezett „kliens komponens”, azok interaktív működése miatt, a többi „szerver komponens”, amiket a böngésző statikus kódrészletként kap meg. A főoldalon lévő szekciók egészei egy-egy ilyen újra használható komponensben van megvalósítva későbbi szerkeszthetőség megkönnyebbítéséért.

#### 5.2.1. Orange-/BlueButton

A weboldal majdnem egészen szereplő narancssárga és kék lekerekített gombok ebből a komponensből jönnek. A komponensnek át kell adni, mi szerepeljen feliratként, illetve milyen linkre mutasson. A navigáció React Link element-et használ, ami a HTML 'a'

element-tel ellentétben prefetching-gel egészíti ki a navigációt, ezzel is gyorsítva a betöltési időket, valamint javítva a felhasználói élményt.

### 5.2.2. Cart

Ez a komponens felel a kosárban tárolt termékek megjelenítésére, egy `CartProp` típusú objektumot kell átadni neki, a komponensen belül van lekezelve, hogy amennyiben nincs képe a megadott terméknek, vagy az a megadott kép hibás, azaz nem tölt be egy úgynevezett fallback képet jelenít meg az oldal összképét nem rontva, a felhasználónak egyértelműen jelezve, hogy az adott terméknek nincs képe.

### 5.2.3. LoggedValidation és LoggedInButtons

Ez a kettő komponens nagyban függ egymástól. Az előbbi kettő JSX element-et vár meghíváskor és a komponensen belül vizsgál rá, hogy az adott látogató be van-e jelentkezve vagy sem, és attól függően jeleníti meg a megadott element-ek közül a megfelelőt. Az navigációs sávban is ezzel van megoldva, hogy ha bejelentkezik egy felhasználó, akkor más gombok látszódnak neki, mint egy vendég felhasználónak. Amennyiben a bejelentkezett felhasználónak van beállítva profilképe az az alap felhasználó ikont lecseréli.

### 5.2.4. Payment

A felhasználó korábbi vásárlásainak kilistázásában segít ez a komponens. A felhasználó a profiljára kattintva látja a személyes adatok szerkesztése mellett a korábbi rendeléseit, és annak részleteit például annak azonosítóját, összegét, státuszát, dátumát, illetve a megrendelt termékek listáját.

### 5.2.5. ProductCard

A termékek aloldalon van használva ez a komponens, és minden egyes termék megjelenítésekor ez van meghívva, egy `Product` típusú objektumot vár, a képeket itt is a komponensen kezeli le, amennyiben nincs vagy hibás. A gyűjtőoldalon minden kártyának van egy fallback element-je, ami ameddig a böngésző nem renderelte be összes megjelenített terméket, az összes helyett egy skeletont jelenít meg. Ennek a lényege, hogy a felhasználói élményt javítsa, minél előbbi felület megjelenítésével, valamint a betöltés során próbálja a nem tervezett tartalom elmozdulásokat csökkenteni.



### 5.2.6. ProductSwiper

Minden egyes termék aloldalán jeleníti meg az adott termékhez tartozó képet. Egy kép url-ekből álló tömböt vár, majd megjeleníti azokat, egy nagyobb swiper-t, ahol a felhasználó meg tudja nézni nagyban, hogy néz ki az adott termék, valamint egy másik swiper-ből áll, ami egy kisebb képeket tartalmaz, ez úgynevezett thumb-ként szerepel. Kattinthatóak, és a nagyobbban az adott kép fog megjelenni. A komponensen belül vannak lekezelve esetleges hibák, amikor a képek betöltenek, amennyiben valamilyen hiba van bármelyik képpel, azt nem fogja lerenderelni, nem fog megjelenni.

### 5.2.7. Recommendation

Egy főoldalon megtalálható szekció, külön komponensben kiszedve esetleges jövőbeli újra felhasználhatóság miatt (pl. termékek aloldalai). A komponensen belül vannak fetch-elve a megjelenített termékek, ezek mögött a logika jelenleg az, hogy minden egyes termékkategóriából vesz ki kettő véletlenszerű terméket, és ezeket összekeverve jeleníti meg a felhasználónak, ezzel biztosítva a változatosságot. A megvalósítás biztosítja a jövőbeli fejlesztési lehetőségeket, mint például a megtekintett termékek alapján megjelenő ajánlás.

### 5.2.8. RecommendationCard

Sok dologban hasonlít a ProductCard komponensre, viszont apró eltérések vannak, például a kosárba gomb ennél a kártyánál nem egy icon, hanem szöveg. A Recommendation szekció ezt a komponenset hívja meg és ennek a segítségével jeleníti meg a termékeket külön-külön.

### 5.2.9. Reviews, ReviewCard

Ez is egy a főoldalon megjelenő szekció, amely az oldalra írt értékeléseket jeleníti meg. Minden felhasználó tud írni értékelést, amennyiben regisztrált vagy bejelentkezett. Ezek az értékelések véletlenszerűen jelennek meg a főoldal alján, játszik melyik felhasználó írta és mikor, a 24 órán belül írt értékeléseknél az idő nem dátumot jelenít meg, hanem az értékelés írásától számított időt (pl. 1 órája, 25 perce, épp most). Ez a szekció a ReviewCard komponenset használja a megjelenítésre, amelynek csak az adott értékelést tartalmazó objektumot kell átadni.

### 5.2.10. SearchButton

Ez a komponens található meg a navigációs sávban, a felhasználó itt tud keresni az összes termék között, legalább 3 karakter beírás után ad vissza találatot, amely egy legördülő menüben látszódik, itt legfeljebb 4 találatot jelenít meg, viszont amennyiben ennél több találat van, a további találatok gombra kattintva a felhasználó láthatja az összes eredményt a keresett kulcsszóra vagy karakterekre. A legördülő menüben a talált termékeknek megjelenik egy képe, illetve a neve, erre kattintva pedig átdob az adott termék aloldalára.

## 6. Fejlesztési lehetőségek

### 6.1. Egyedi fizetési rendszer elkészítése

Egy egyedi rendszer kialakítása nagyon sok előnnyel járhatna. Például jelenleg a stripe szervereitől függ a fizetés, így, ha esetleg valamilyen hiba van az ő oldalukon akkor nálunk rendelést sem tudnak leadni. Különböző tranzakciós díjakat is meg tudnánk spórolni így. Teljesen egyedi felhasználói felületet tudnánk létrehozni ezáltal jobban passzolna az oldal dizájnjához. Akár AI alapú fizetésellenőrzés integrációja is lehetséges lenne.

### 6.2. Számítógépösszerakó

Egy ilyen funkció nagyban növelné a felhasználói élményt, hiszen egy oldalon tudnák összerakni a kívánt számítógépüket és megrendelni azt. Ez a rendszer lehetőséget adna arra, hogy a felhasználók kiválaszthassák a számukra legmegfelelőbb alkatrészeket. Kompatibilitás-ellenőrzőt is lehetne bele készíteni, hogy megbizonyosodjanak róla, hogy biztosan kompatibilisek egymással a kiválasztott termékek. Akár ezeket a konfigurációkat el is menthetnék és megoszthatnák az ismerőseikkel.

### 6.3. Hűségprogram

Egy pontgyűjtő rendszer, amely jutalmazná a visszatérő vásárlókat. A felhasználók minden vásárlás után pontokat kapnának, amelyeket később kedvezményekre válthatnának be. Ezzel a módszerrel visszatérő vásárlókat tudnánk ösztönözni.

### 6.4. Termékösszehasonlítás

A vásárlók számára lehetőséget biztosíthatnánk arra, hogy több alkatrészt egymás mellett összehasonlítsanak, így könnyebben eldönthetnék, hogy melyik a legmegfelelőbb választás számukra. A rendszer kilistázná a specifikációkat így könnyen döntésre jutnának a vásárlók.

## 6.5. Termékenkénti vélemény

Oldalunk jelenleg is van lehetőség az oldalról véleményt írni, viszont ennek a funkciónak a megvalósításával a látogatók személyes tapasztalatokat is olvashatnának a megtetszett termékkel kapcsolatban, ezáltal egy tisztább véleményt kapva.

## 6.6. Ügyfélszolgálat rendszer kialakítása

Egy ügyfélszolgálati rendszer megvalósítása hasznos lehetne a webshopunk számára, hiszen biztosítja vásárlóink gyors és hatékony kéréseit. Egy jegykezelő rendszer bevezetése is hasznos lehet, amely segít megszerezni és nyomon követni a vásárlói megkereséseket, biztosítva, hogy minden ügy gyorsan és hatékonyan legyen kezelve és ezt akár a rendelésekbe is lehetne implementálni. Érdeemes lehet beépítenünk egy GYIK (Gyakran Ismételt Kérdések) szekciót, amely gyors válaszokat adhat a leggyakoribb problémákra, így csökkentve az ügyfélszolgálatra nehezedő terhet. Akár AI-alapú chatbotokat, amelyek képesek a vásárlói kérdések megválaszolására, útmutatást adhatnak a vásárlási folyamat során.

## 7. Összegzés

A szakdolgozat középpontjában a HS Market nevű, számítógép-alkatrészek értékesítésére specializált webes alkalmazás áll. A projekt célja egy letisztult, felhasználóbarát és technikailag korszerű platform létrehozása volt, amely túlmutat a hagyományos hazai webáruházakon, kiemelten kezelve a vásárlói élményt és az interaktivitást.

A fejlesztés során a Next.js 15.0.2 keretrendszert, MySQL adatbázist, valamint Prisma ORM-et alkalmaztunk, biztosítva az adatkezelés típusbiztosságát és biztonságát. A rendszer front-end és back-end része szorosan együttműködik REST API-kon keresztül, melyek működését a Postman alkalmazás segítségével teszteltük.

A HS Market regisztráció nélküli és regisztrált felhasználók számára is kínál funkciókat. Előbbiek böngészhetnek a termékek között, utóbbiak számára elérhető a kosár, vásárlás, saját profil szerkesztése és korábbi rendelések megtekintése. Az adminisztrátori jogosultság további funkciókat kínál: termékek, felhasználók és kuponok kezelése.

A platform egyik kulcseleme a széleskörű szűrési és rendezési lehetőség, amely termékkategória, ár, gyártó és betűrend szerint segíti a navigációt. A részletes termékoldalak minden szükséges információt tartalmaznak, és közvetlen hozzáférést biztosítanak a kosárhoz. A vásárlás lebonyolítását a Stripe integráció biztosítja, amely garantálja a gyors és biztonságos fizetést. A jelszavak bcrypt-tel hash-eltek, a bejelentkezéshez pedig JWT alapú autentikációt alkalmaztunk, így a felhasználói adatok védelme garantált.

Az oldal több újrahasznosítható komponensből épül fel, melyek például a termékkártyák, véleménymegjelenítők, keresés, kosárkezelés és fizetés folyamatát is kiszolgálják. A weboldal tervezése Figma segítségével történt, lehetővé téve a kooperatív dizájn kialakítást és kommentálást.

A rendszer rendelkezik véleményírási funkcióval, melyet később termékekre szabott

visszajelzési lehetőséggel lehetne bővíteni. A főoldalon ezekből hat vélemény jelenik meg véletlenszerűen. További extra funkciók közé tartozik a termékajánlás szekció, az oldal tetejére ugró gomb, valamint a részletes kereső és profilkonckciók.

A dokumentáció végén részletes adatbázis-leírás szerepel, mely 7 táblát tartalmaz, világosan strukturálva a rendszer felépítését. A Prisma ORM megoldás révén az SQL injection elleni védelem beépített, a migrációk pedig nyomon követhetők. A JSON Web Token használata lehetővé teszi a jogosultsági szintek egyszerű és biztonságos kezelését.

A jövőbeli fejlesztési lehetőségek között szerepel: saját fizetési rendszer kialakítása, számítógép-összerakó funkció, hűségprogram, termék-összehasonlítás, termékenkénti véleményezés és AI-alapú ügyfélszolgálat.

A projekt nem csupán technikailag igényes, hanem erős felhasználói orientáltságról is tanúskodik. A HS Market példája jól mutatja, hogy a digitális kereskedelemben a funkciógazdagság mellett a jól megtervezett felhasználói élmény is kiemelkedő jelentőséggel bír.

## 8. Következtetés

A projekt során nemcsak programozói tudásunkat fejlesztettük, hanem gyakorlati tapasztalatot is szereztünk a projekt létrehozása közben. Magabiztosan állíthatjuk, hogy az itt megszerzett ismeretek és tapasztalatok szilárd alapot nyújtanak a jövőbeni terveinkhez.

Emellett a közös munka során lehetőségünk nyílt arra is, hogy megtapasztaljuk a csapatmunka valódi erejét. Megtapasztalhattuk a hatékony kommunikációt, megosztottuk egymással ötleteinket, és folyamatosan támogattuk egymást a projekt minden szakaszában. Mindenki hozzátette a maga tudását és tapasztalatát, ami nagyban hozzájárult a projekt elkészítéséhez.

Büszkéek vagyunk arra, amit együtt létrehoztunk, és örülünk, hogy sikerült elérnünk a kitűzött céljainkat. Ez a projekt nemcsak technikai tudásunkat fejlesztette, hanem segített abban is, hogy jobb csapatjátékosokká váljunk, hatékonyabban oldjuk meg a problémákat közösen, és reálisan kezeljük a fejlesztési feladatokat.

## 9. Irodalomjegyzék

Dokumentációk, amiket felhasználtunk a projekt során:

- Next.JS dokumentáció: <https://nextjs.org/docs>
- React dokumentáció: <https://react.dev/reference/>
- Prisma ORM dokumentáció: <https://www.prisma.io/docs>
- Stripe API dokumentáció: <https://docs.stripe.com/api>

Adatbázishoz:

- Diagram: <https://dbdiagram.io>
- Adatbázis lekérdezésekbe segítséget nyújtott: <https://dev.mysql.com/doc/>

További programozási problémák megoldásához:

- Stackoverflow: <https://stackoverflow.com/>
- Github fórum: <https://github.com/issues/>
- Medium: <https://medium.com/>
- Dev.to: <https://dev.to/>



