

Győri SZC Jedlik Ányos Gépipari Informatikai  
Technikum és Kollégium

# GameStrix webáruház

Gamer webshop



GAMESTRIX

Boglári Bence Krisztián

Horváth Máté

Vida Csaba Zsombor

Győr, 2024

## Záródolgozat feladatkiírás

Tanuló(k) neve<sup>1</sup>: **Boglári Bence Krisztián, Horváth Máté, Vida Csaba Zsombor**

Képzés: nappali

5 0613 12 03 Szoftverfejlesztő és tesztelő technikus

A záródolgozat címe:

**Gamestrix webáruház**

Konzulens: Bolya Gábor

Beadási határidő: 2024. 04. 22.

Győr, 2024. 04. 22

---

**Módos Gábor**  
igazgató

---

<sup>1</sup> Szakmajegyzékes záródolgozat esetében több szerzője is lehet a dokumentumnak, OKJ-s záródolgozatnál egyetlen személy ad le záródolgozatot.

## Konzultációs lap<sup>2</sup>

	A konzultáció		Konzulens aláírása
	ideje	témája	
1.	2023.11.11.	Témaválasztás és specifikáció	
2.	2024.04.10.	Záródolgozat készültségi fokának értékelése	
3.	2024.04.21.	Dokumentáció véglegesítése	

## Tulajdonosi nyilatkozat

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

Győr, 2024. április 28.

---

Boglári Bence Krisztián

---

Horváth Máté

---

Vida Csaba Zsombor

---

<sup>2</sup> Szakmajegyzékes, csoportos konzultációs lap

## Tartalom

Weboldal neve .....	6
Rövid leírás .....	6
Cél .....	6
Terjedelmesebb leírás: .....	6
Publikáció .....	8
Munkafolyamatok kiosztása .....	9
Szerepek.....	10
Horváth Máté .....	10
Vida Csaba Zsombor .....	10
Boglári Bence Krisztián.....	10
Csapatmunka .....	10
Programok.....	11
Visual Studio Code.....	11
Git, Github.....	12
Vue.....	12
Laravel.....	12
Database: MySQL.....	12
Mérföldkövek.....	13
Módszertani terv .....	13
Fejlesztés megkezdése előtt .....	13
Fejlesztés megkezdése után .....	14
Megvalósítás .....	14
A fejlesztés során használt projektszervezési eszközök .....	15
1.Meeting.....	15
2. Discord .....	15
3. Github .....	15
Adatbázis.....	16
Adatbázis ábra .....	16
Adatbázis leírása .....	17
A Webszerver elérése: .....	18
A backend fejlesztése.....	19

Laravel bemutatása .....	19
Alapvető Laravel Setup .....	20
Backend Routeok.....	21
Kérések Részletezése .....	22
Backend tesztje.....	24
Category Test .....	24
Login Test .....	25
Product Test .....	27
Register Test .....	29
Frontend fejlesztése.....	31
Fejléc: .....	32
Navbár teszt:.....	32
Nyitó oldal:.....	33
.....	34
Termékek Oldal: .....	35
Adatok lekérése: .....	36
Kosárba helyezés: .....	37
Szűrés: .....	37
ProductDetails oldal:.....	38
Rólunk oldal: .....	39
Elérhetőségek oldal: .....	40
.....	40
Kosár: .....	41
.....	41
Bejelentkezés/Regisztráció: .....	42
Belépés folyamata:.....	43
Regisztráció folyamata: .....	43
Regisztráció és bejelentkezés tesztje: .....	44

# Tervek (Alapító okirat alapján)

## Weboldal neve

Gamestrix Webáruház

## Rövid leírás

Számítógépes hardverekkel kereskedő webáruház amely lehetővé teszi a legmodernebb hardverek megvásárlását az interneten keresztül.

## Cél

A csapattagok számára új technológiák megismerése, tapasztalatszerzés. Szerettük volna megismerni a legújabb technológiákat a webfejlesztésben. Célunk volt hogy egy olyan oldalt készítsünk amely akár hosszútávon megállja a helyét és megfelelően hiba nélkül funkcióit ellátja.

## Terjedelmesebb leírás:

A webshopok fontossága ma már szinte vitathatatlan. Nem csupán a

kényelem és az egyszerűség miatt, hanem azért is, mert olyan lehetőségeket kínálnak, amelyek révén a gamerek könnyedén hozzájuthatnak a legfrissebb és legjobb hardverekhez és kiegészítőkhöz, és ezzel javíthatják játékélményüket és versenyképességüket a virtuális világban. A Gamestrix célja, hogy ebben a folyamatban segítséget és inspirációt nyújtson minden egyes látogatójának, hogy álmai játékélményét valóra válthassa.

A Gamestrix névre hallgató webshop létrehozásának háttérében egy egyszerű, ámde szenvedélyes misszió áll: az, hogy az online térben egy olyan platformot teremtsünk, ahol minden gamer megtalálhatja az álmai hardvereit, kiegészítőit és minden egyébét, amire szüksége lehet az élmény teljes kibontakozásához.

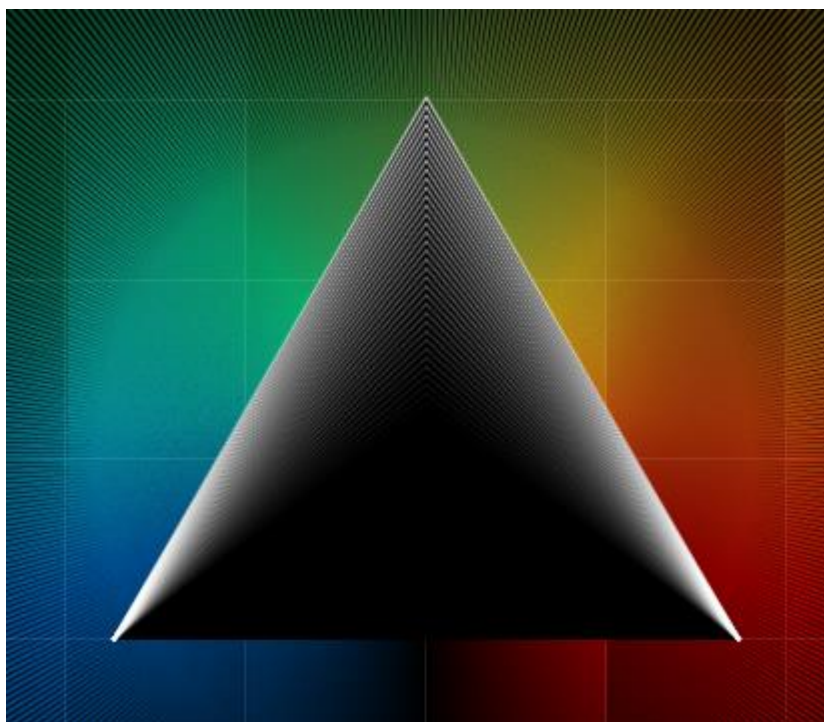
A csapatban magunk is részei vagyunk ennek a közösségnek, így pontosan érezzük és értjük azokat az elvárásokat és kihívásokat, amelyekkel a gamerek szembesülnek, amikor új hardvereket szeretnének vásárolni.

A Gamestrix nem csupán egy egyszerű bolt, hanem egy olyan virtuális központ, ahol a gamerek nem csak termékeket találnak, hanem információkat, útmutatást és inspirációt is. A széles termékkínálat és a részletes termékleírások segítenek abban, hogy a vásárlók pontosan megtalálják azt, amire szükségük van, miközben a blogbejegyzések és videós tartalmak segítenek nekik abban, hogy mélyebben megismerjék a legújabb trendeket, technológiákat és játékokat.

## Publikáció

A Gamestrix Webáruház fejlesztése során eredetileg a Vercel.app nevű platformot használtuk a webalkalmazás hosztolására és tesztelésére. A Vercel könnyen kezelhető felhőalapú platformot kínál, amely lehetővé teszi a fejlesztők számára, hogy gyorsan és hatékonyan hozzáférjenek a projektjükhöz, és könnyen teszteljék és publikálják azt.

Azonban a fejlesztés előrehaladtával és a projekt komplexitásának növekedésével úgy döntöttünk, hogy áthelyezzük a webalkalmazást egy helyi felhőszerverre. Ennek oka több tényezőn alapult. Először is, a helyi szerver lehetőséget biztosított számunkra arra, hogy teljes mértékben testre szabjuk a környezetet és a konfigurációt a saját igényeink szerint. Másodsorban, a helyi szerveren való hosztolás fokozottabb ellenőrzést és biztonságot nyújtott a projekt felett.





## Munkafolyamatok kiosztása

A munkafolyamatok hatékony kiosztása és irányítása az egyik sarokköve a sikeres csapatmunkának. A Gamestrix Webáruház fejlesztése során kiemelt figyelmet fordítottunk arra, hogy minden csapattag számára megfelelő feladatokat osszunk ki, amelyek összhangban vannak a képességeikkel, tapasztalataikkal és érdeklődési körükkel.

A hatékony kommunikáció kulcsszerepet játszik ebben a folyamatban. A csapatnak nyitottnak és befogadónak kell lennie a kommunikációra, hogy mindenki szabadon kifejezhesse az igényeit, érdeklődési körét és véleményét a feladatokkal kapcsolatban. Ez segíti a csapatot abban, hogy jobban megértsék egymás szükségleteit és elvárásait, valamint könnyebben kezeljék az esetleges konfliktusokat vagy problémákat.

A rugalmasság és az alkalmazkodóképesség elengedhetetlenek egy dinamikusan változó környezetben. A technológiai fejlődés és a piaci igények gyors változása új kihívásokat és helyzeteket teremt, amelyekre a csapatnak gyorsan és hatékonyan kell reagálnia. Ezért fontos, hogy a csapat rugalmas legyen és képes legyen alkalmazkodni az új körülményekhez és követelményekhez.

A feladatkörök megfelelő elosztása a hatékony munka érdekében szintén alapvető fontosságú. Minden csapattagnak olyan feladatokat kell kapnia, amelyek megfelelnek a képességeinek és szaktudásának, és amelyek hozzájárulnak a projekt sikeréhez. Ennek eredményeként mindenki hatékonyan tudja végezni a feladatait, és a csapat egésze összehangoltan tud dolgozni a közös cél elérése érdekében.

Ezek az alapelvek és gyakorlatok segítettek abban, hogy a Gamestrix Webáruház fejlesztése során hatékony és produktív csapatmunka valósuljon meg, amelynek eredményeként egy stabil, innovatív és felhasználóbarát webáruház jött létre.

## Szerepek

### Horváth Máté

Horváth Máté szerepe a csapatban a fő Backend fejlesztő. Ennek megfelelően ő felelős az adatbázis megtervezéséért, megvalósításáért és kezeléséért. Az ő feladatai közé tartozik az adatstruktúrák kidolgozása, az adatbázis optimalizálása és a háttérrendszer megvalósítása. Máté kiváló szakértő a szerveroldali technológiák terén, és tapasztalatai segítenek abban, hogy a projekt hatékonyan működjön.

### Vida Csaba Zsombor

Vida Csaba Zsombor fő feladata a Frontend fejlesztés, valamint a felhasználói felület megtervezése és megvalósítása. Ő felelős azért, hogy a webáruház felhasználóbarát és vonzó legyen a látogatók számára. Emellett Csaba Zsombor a projekt dokumentációjának elkészítését is ellátja, amely segít a csapatnak és a későbbi fejlesztőknek megérteni és karbantarthatóvá tenni a kódbázist.

### Boglári Bence Krisztián

Boglári Bence Krisztián a csapat Fő Frontend fejlesztője, akinek fő feladata a komplikáltabb funkciók elkészítése és implementálása a webáruházban. Ő felelős azért, hogy az előre megtervezett felhasználói élményt valóra váltsa és biztosítsa a weboldal megfelelő működését minden platformon és eszközön. Krisztián kiválóan ért a modern webes technológiákhoz, és kreatív megoldásokkal áll elő a felmerülő kihívásokra.

## Csapatmunka

A csapat kiválóan működik együtt, és dinamikusan képes alkalmazkodni az előforduló helyzetekhez és kihívásokhoz. A közös munka során gyakran tanulnak egymástól, és segítik, elemzik egymás munkáját. Az együttműködés megkezdésekor mindenki világosan meghatározta, hogy milyen feladatokat vállal és milyen területen érzi magát otthonosan. Ennek köszönhetően minden csapattag hatékonyan tudja végezni a rábízott feladatokat, és a projekt zökkenőmentesen halad előre a kitűzött célok felé.

## Eszközök

Környezet	Programok
Code Editor	<a href="#">Visual Studio Code</a>
Version Control System	<a href="#">Git</a> , <a href="#">Github</a>
Frontend	<a href="#">Vue</a>
Backend	<a href="#">Laravel</a>
Database	<a href="#">Mysql</a>

## Programok

A csapat által használt környezet és programok széles skálája elengedhetetlen a hatékony és zökkenőmentes fejlesztéshez. Az alábbiakban részletesen bemutatom ezeket a programokat és eszközöket:

### Visual Studio Code

A Visual Studio Code egy erőteljes és könnyen testreszabható fejlesztői környezet, amelyet a Microsoft fejlesztett ki. Számos előnye közé tartozik a gyors teljesítmény, a kiterjeszthetőség és a széles körű támogatottság számos programozási nyelvhez és kerethez. A VS Code kiválóan alkalmas webfejlesztéshez, kódszerkesztéshez és hibakereséshez, valamint integrált támogatást nyújt a Git verziókezelő rendszerhez.

## Git, Github

A Git egy elosztott verziókezelő rendszer, amely lehetővé teszi a fejlesztők számára, hogy nyomon kövessék a projekt változásait, és hatékonyan együttműködjenek más fejlesztőkkel. A Git segítségével könnyen kezelhetők a változtatások, lehetőség van külön ágak létrehozására és összefésülésére, valamint visszaállításra korábbi verziókhoz. A GitHub egy távoli Git tárhelyszolgáltató, amely lehetővé teszi a projekt megosztását, nyilvános vagy privát repository-k létrehozását, valamint a csapatmunka könnyű koordinálását.

## Vue

A Vue.js egy progresszív JavaScript keretrendszer, amelyet különösen a felhasználói felület fejlesztésére terveztek. A Vue rendkívül könnyű tanulni és használni, és kiválóan alkalmas a dinamikus és interaktív webalkalmazások fejlesztéséhez. A Vue komponensalapú megközelítése lehetővé teszi a kód újrafelhasználását és a komplex felhasználói felületek egyszerűbb kezelését.

## Laravel

A Laravel egy PHP alapú, nyílt forráskódú webes keretrendszer, amelyet a gyors és hatékony webalkalmazások fejlesztésére terveztek. A Laravel számos előnyt kínál, beleértve a beépített hitelesítési, adatbázis-kezelési és útvonalkezelési funkciókat, valamint a könnyű kódolhatóságot és karbantarthatóságot. A Laravel erőteljes eszközökkel rendelkezik a fejlett webalkalmazások fejlesztéséhez, miközben egyszerű és intuitív szintaxist kínál.

## Database: MySQL

A MySQL nyílt forráskódú relációs adatbázis-kezelő rendszer, amelyet a legtöbb webalkalmazás háttértárolójaként használnak. A MySQL gyors és megbízható adatbázis-kezelési megoldást nyújt, támogatja a tranzakciókat, és könnyen integrálható a legtöbb webes alkalmazásba.

## Mérföldkövek

Látványterv	2023.12.15.
Működési felépítési, tervezet	2023.12.19.
Adatbázis felépítése	2024.01.11.
Backend	2024.04.03.
Frontend kapcsolat felépítése, felhasználói tevékenységek	2024.04.15.
Prototípus	2024.04.15.
Határidő	2022.04.15.

## Módszertani terv

A módszertani terv alapvetően meghatározza, hogy milyen elvek és módszerek mentén fogunk haladni a fejlesztés során. Ez biztosítja, hogy a csapat egységesen és hatékonyan tudjon dolgozni, és könnyen kezelni tudja az előforduló kihívásokat.

## Fejlesztés megkezdése előtt

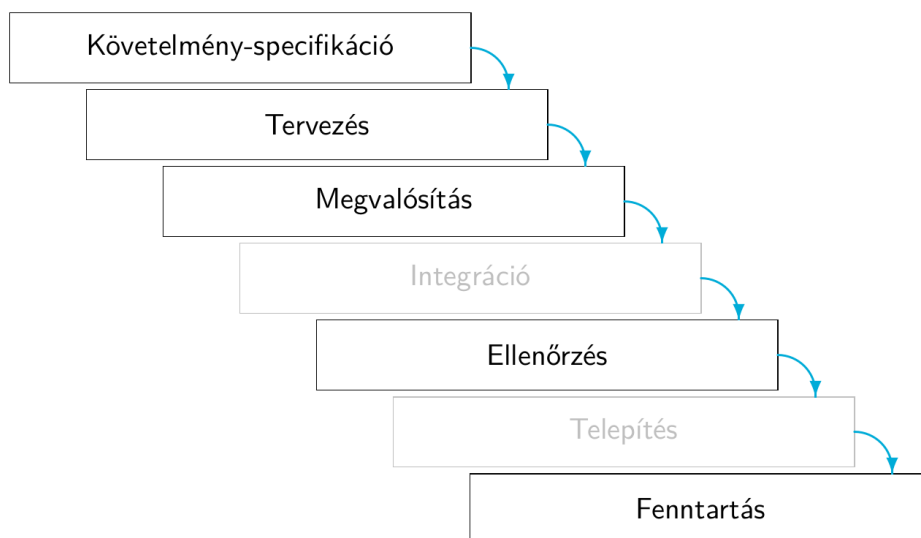
Miután megismertük a rendelkezésre álló lehetőségeket a szoftverfejlesztési módszertanok terén, alaposan mérlegeltük azokat. Végül egyeztettünk arról, hogy a vízesés modellt választjuk a fejlesztés alapjául. Ennek oka az volt, hogy ez a módszertan jól alkalmazkodik az előre meghatározott lépésekhez, és lehetőséget ad a részletes tervezésre és dokumentálásra, ami kritikus fontosságú a projekt sikeréhez.

## Fejlesztés megkezdése után

Miután elindult a fejlesztés, a csapat megfelelően tanult és alkalmazkodott a változó környezethez és kihívásokhoz. Ennek eredményeként a fejlesztés módszertanának alkalmazása és követése természetesen beállt, és a csapat már nem igényli annak folyamatos felügyeletét vagy ellenőrzését.

## Megvalósítás

A fejlesztés során két fontos leadási mérföldkövet határoztunk meg. Az első a beta verzió kiadásának dátuma volt, melyet 2024. március 17-ére terveztünk. Ez az időpont lehetővé tette számunkra, hogy egy korai verziót megosszunk a csapattagokkal és érdekeltekkel, hogy visszajelzéseik alapján tovább finomíthassuk és javíthassuk a webáruházat. Azután, hogy a beta verzió kikerült, folytattuk a további funkciók integrálását és a végső simítások elvégzését, hogy a leadási határidőig egy stabil és kiforrott terméket kínálhassunk a felhasználóknak. Ez a módszertan segített a projekt strukturált és hatékony előrehaladásában, miközben lehetővé tette a csapatnak, hogy időben reagáljon az előforduló kihívásokra és változásokra.



# A fejlesztés során használt projektszervezési eszközök

A fejlesztés során számos projektmenedzsment eszközt használtunk, hogy hatékonyan tudjunk együtt dolgozni és a projektet strukturált módon előrehaladását biztosítsuk.

## 1.Meeting

Szinte napi rendszerességgű személyes gyűlések: Az egyik legfontosabb eszközünk a személyes találkozók, melyek gyakorisága szinte napi szinten volt. Mivel ugyanazon nappali tagozatos képzésben veszünk részt, minden munkanapon találkoztunk, ami lehetővé tette számunkra, hogy azonnal megbeszéljük a projekt állását, felmerülő kérdéseket, és tapasztalatokat osszunk meg egymással. Ezek a találkozók nemcsak a munkafolyamat hatékony irányítását szolgálták, hanem a csapat összetartását is erősítették.

## 2. Discord

A Discord egy másodlagos, de rendkívül hasznos kommunikációs forma volt számunkra. Gyors üzenetváltásokra és azonnali problémamegoldásra használtuk. Emellett lehetőséget adott az oktatási anyagok megosztására és az információk gyors terjesztésére a csapat tagjai között.

## 3. Github

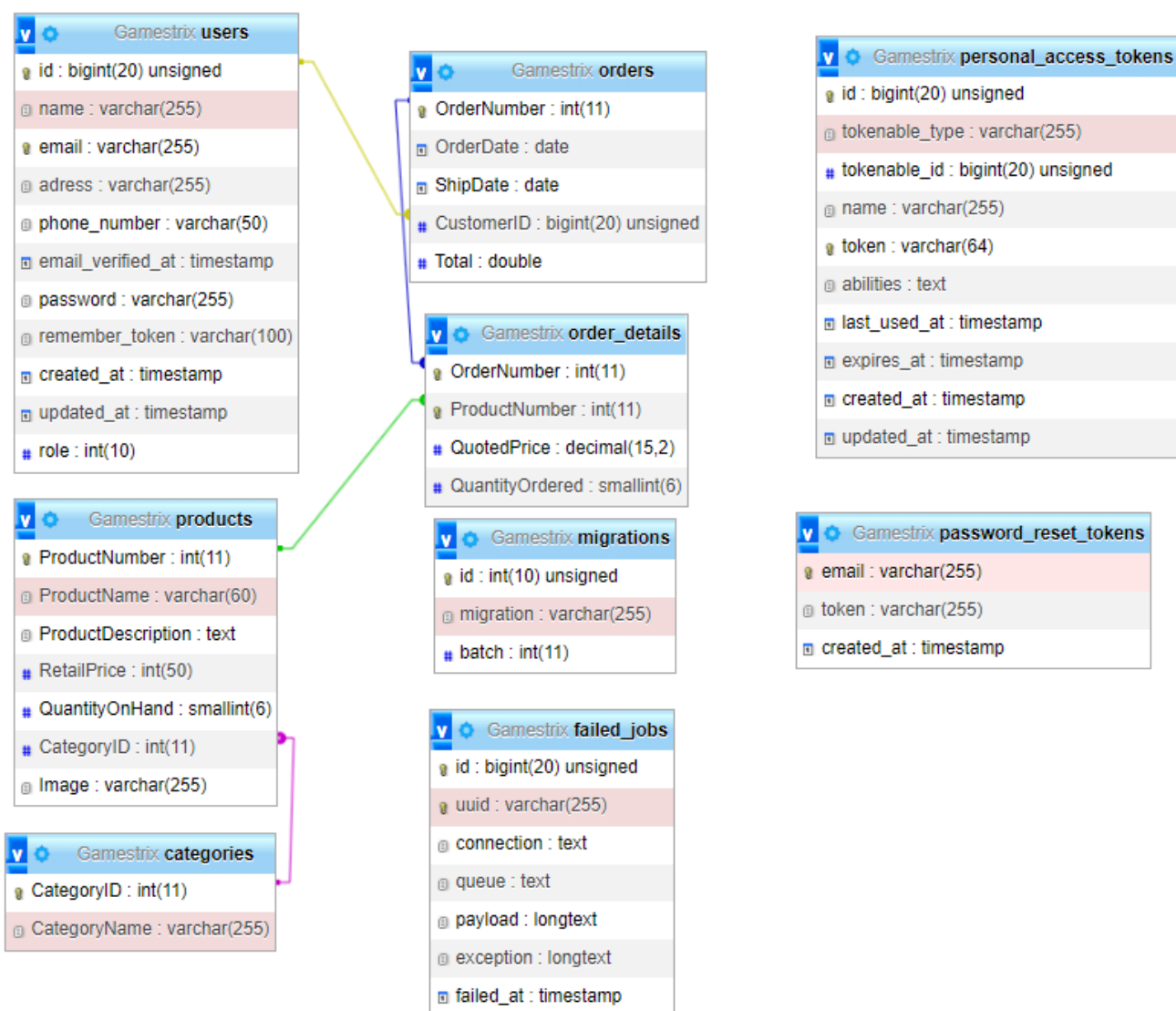
A Githubot használtuk a projektek megosztására és változtatásainak nyomon követésére. Ez a platform lehetővé tette számunkra, hogy közösen dolgozzunk azonos kódbázison, könnyen kezeljük a verziókat és összehangoljuk a fejlesztés folyamatát. A Github segítségével könnyen nyomon követhettük az egyes fejlesztési feladatok állapotát, valamint könnyen koordinálhattuk a csapatmunkát a különböző funkciók és modulok fejlesztése során.

Ezen eszközök kombinációja biztosította, hogy a csapat hatékonyan tudjon együtt dolgozni, gyorsan reagálhasson az előforduló kihívásokra és változásokra, valamint hatékonyan tudja irányítani és előrehaladását a projektnek.

# Szakdolgozat részeinek bemutatása

## Adatbázis

### Adatbázis ábra





## Adatbázis leírása

Az adatbázis-struktúra mögötti logika az, hogy hatékonyan szervezze és kezelje az adatokat a webáruház felhasználói, termékei és rendelései kapcsán:

**Felhasználók tábla:** A regisztrált felhasználók adatait tárolja, mint például az ID, név, e-mail cím, jelszó stb. Ez lehetővé teszi a felhasználók kezelését és az hitelesítési folyamatokat.

**Termékek tábla:** A webáruházban elérhető termékek adatait tartalmazza, mint például a termékszám, név, leírás stb. Ez lehetővé teszi a termékek listázását és kezelését.

**Kategóriák tábla:** Kategorizálja a termékeket, hogy javítsa a felhasználói élményt, lehetővé téve a felhasználók számára, hogy specifikus kategóriák alapján böngésszenek az elemek között.

**Rendelések tábla:** Rögzíti a rendelésspecifikus információkat, mint például a rendelésszám és a vásárló ID, biztosítva, hogy minden tranzakció rögzítésre kerüljön a feldolgozáshoz és nyomon követéshez.

**Rendelési részletek tábla:** Részletes információkat tartalmaz az egyes rendelésekben található tételekről, ami lehetővé teszi a készletkezelést és az értékesítés nyomon követését.

**Sikertelen munkák tábla:** Rögzíti a rendszerben sikertelenül végrehajtott munkákat vagy feladatokat, biztosítva, hogy a problémákat azonosítani és gyorsan megoldani lehessen a rendszer integritásának fenntartása érdekében.

Ez a struktúra biztosítja az adatintegritást, támogatja a jelentéskészítéshez szükséges összetett lekérdezéseket, miközben skálázható a felhasználók vagy a terméklisták növekedésének kezelésére. Ez a struktúra lehetővé teszi a webáruház hatékony működését és a felhasználói élmény javítását.



GAMESTRIX

Készítette: Boglári Bence Krisztián, Horváth Máté, Vida Csaba Zsombor

Az adatbázist phpmyadmin alkalmazásban hoztuk létre. Bólya Gábor tanár úr segítségével fel tudtuk tölteni az adatainkat egy webszerverre, így azt bárhol el tudtuk érni. Ez jelentősen megkönnyítette a munkafolyamatot.

## A Webszerver elérése:

[GameStrix webáruház \(jedlik.eu\)](http://jedlik.eu)

<https://bgs.jedlik.eu/gamestrix/>

## A backend fejlesztése

### Laravel bemutatása

A döntésünk a Laravel keretrendszer és a PHP nyelv mellett nem csupán a technológiai előnyökön alapult, hanem azoknak a szélesebb körű előnyeiben és az általuk kínált lehetőségekben is gyökerezett, melyek ideálisak egy gamer webáruház backendjéhez.

A Laravel egy olyan keretrendszer, amely tökéletes egyensúlyt teremt a fejlett funkcionalitás és az egyszerűség között. A kiváló dokumentáció és az aktív fejlesztői közösség biztosítja, hogy könnyen és hatékonyan haladhassunk előre a fejlesztésben. Ez lehetővé teszi számunkra, hogy a tervezéstől a kódírásig és a tesztelésig minden lépést gyorsan és hatékonyan hajtsunk végre.

A Laravel széleskörű funkciókat és szolgáltatásokat kínál, amelyek megkönnyítik a fejlesztői munkát. Az előre elkészített sablonok és az autentikációs rendszer lehetővé teszi, hogy gyorsan és hatékonyan kezeljük a felhasználói fiókokat és az azokhoz kapcsolódó funkciókat. Emellett az adatbázis-kezelési lehetőségek segítségével könnyedén kezelhetjük és manipulálhatjuk az adatokat, ami kritikus fontosságú egy webáruházban.

A PHP nyelv és a Laravel keretrendszer skálázhatóak és teljesítménycentrikusak, ami különösen fontos egy gamer webáruház esetében. A rendszernek képesnek kell lennie kezelni a nagy terhelést és a nagy forgalmat, amely egy online platformhoz társul. Azáltal, hogy a Laravelra és a PHP nyelvre építünk, biztosíthatjuk a rendszer megbízhatóságát és stabilitását még a legnagyobb terhelés mellett is.

A Laravel keretrendszer beépített biztonsági funkciói és a PHP nyelv általánosan biztonságos jellege garantálja, hogy a webáruházunk védett legyen a különféle online támadásokkal szemben. Ez az átfogó biztonsági megközelítés lehetővé teszi, hogy ügyfeleink biztonságban érezzék magukat és bízzanak abban, hogy adataik és tranzakcióik védettek.

Végül, a Laravel és a PHP választása nemcsak a jelenlegi igényeket

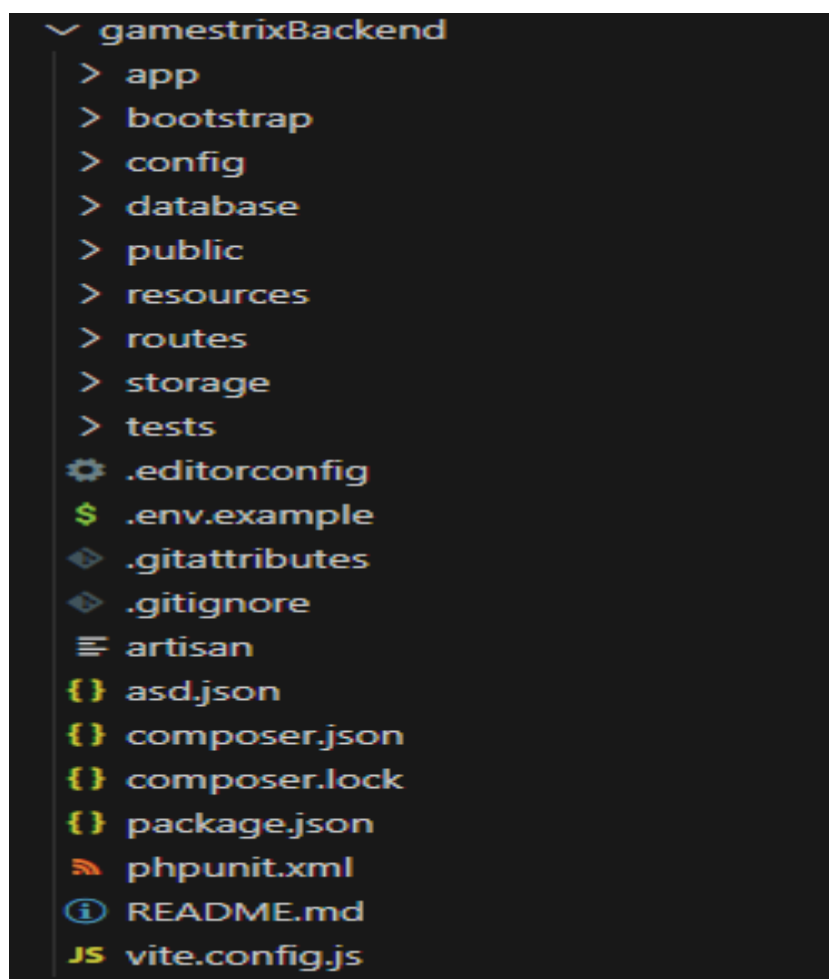


GAMESTRIX

Készítette: Boglári Bence Krisztián, Horváth Máté, Vida Csaba Zsombor

szolgálja ki, hanem a jövőbeni bővítési lehetőségeket is biztosítja. Rugalmas és könnyen karbantartható kódot eredményez, ami lehetővé teszi, hogy gyorsan reagáljunk az ügyfelek igényeire és a piaci változásokra, és továbbfejlesszük a webáruházunkat a jövőben.

## Alapvető Laravel Setup



## Backend Routeok

```
Route::get('/categories', [CategoriesController::class, 'index']);
Route::get('/categories/{id}', [CategoriesController::class, 'show']);

Route::get('/products', [ProductController::class, 'index']);
Route::get('/productsPaged', [ProductController::class, 'getProductsWithPage']);
Route::get('/products/{id}', [ProductController::class, 'productsById']);
Route::get('/product/{id}', [ProductController::class, 'productsSingleOne']);

Route::post('/register', [AuthController::class, 'register']);
Route::post('/login', [AuthController::class, 'login']);
Route::post('/forgotPassword', [AuthController::class, 'ResetPasswordToken']);
Route::post('/resetPassword', [AuthController::class, 'ResetPassword']);
```

```
Route::group(['middleware' => ['auth:sanctum']], function () {
    Route::get('/getAllUser', [UserController::class, 'getAllUser']);
    Route::get('/usersstuff', [UserController::class, 'index']);
    Route::post('/updateProfile', [UserController::class, 'updateProfile']);
    Route::get('/OrdersByUser', [UserController::class, 'getOrdersByUser']);

    Route::post('/logout', [AuthController::class, 'logout']);

    Route::post('/orders', [OrderController::class, 'makeOrder']);

    Route::get('/orderedDetails', [OrderDetailsController::class, 'index']);

    Route::post('/makeProducts', [ProductController::class, 'store']);
    Route::post('/updateProducts/{id}', [ProductController::class, 'updateProduct']);
    Route::delete('/deleteProcts/{id}', [ProductController::class, 'destroy']);
});
```

## Kérések Részletezése

1. **Route::post('/register', [AuthController::class, 'register']);** Ez a kérés lehetővé teszi a felhasználók számára, hogy új fiókot hozzanak létre az alkalmazásban. A post metódus azt jelenti, hogy az alkalmazás várja a felhasználói adatokat (például a felhasználónév és jelszó), amelyeket aztán felhasznál a regisztrációhoz.
2. **Route::post('/login', [AuthController::class, 'login']);** Ez a kérés lehetővé teszi a felhasználók számára, hogy bejelentkezzenek a már létrehozott fiókjukba. Itt is a post metódust használjuk, mert a felhasználónak meg kell adnia a bejelentkezési adatait.
3. **Route::post('/forgotPassword', [AuthController::class, 'ResetPasswordToken']);** Ha a felhasználó elfelejtette a jelszavát, ezzel a kéréssel kérhet egy új jelszót. A kérés során a rendszer elküld egy e-mailt a felhasználónak egy linkkel, amellyel új jelszót állíthat be.
4. **Route::post('/resetPassword', [AuthController::class, 'ResetPassword']);** Ez a kérés lehetővé teszi a felhasználó számára, hogy beállítson egy új jelszót, miután megkapta a jelszó-visszaállítási linket. A post metódus itt is használatos, mert a felhasználónak meg kell adnia az új jelszót.
5. **Route::get('/getAllUser', [UserController::class, 'getAllUser']);** Ez a kérés visszaadja az összes regisztrált felhasználót. A get metódus azt jelenti, hogy az alkalmazás adatokat küld vissza a felhasználónak.
6. **Route::get('userstuff', [UserController::class, 'index']);** Ez a kérés visszaadja a bejelentkezett felhasználó adatait.
7. **Route::post('updateProfile', [UserController::class, 'updateProfile']);** Ez a kérés lehetővé teszi a felhasználó számára, hogy frissítse a profilját. A post metódus itt is használatos, mert a felhasználónak meg kell adnia az új profiladatokat.

8. **Route:get**('OrdersbyUser',[UserController:class,getOrdersByUser]); Ez a kérés visszaadja a bejelentkezett felhasználó összes rendelését.
9. **Route:post**('logout',[AuthController:class,'logout']); Ez a kérés lehetővé teszi a felhasználó számára, hogy kijelentkezzen az alkalmazásból.
10. **Route:post**('orders',[OrderController:class,'makeOrder']); Ez a kérés lehetővé teszi a felhasználó számára, hogy új rendelést adjon le. A post metódus itt is használatos, mert a felhasználónak meg kell adnia a rendelés adatait.
11. **Route:get**('orderDetails',[OrderDetailsController:class,index]); Ez a kérés visszaadja a bejelentkezett felhasználó összes rendelésének részleteit.
12. **Route:post**('makeProducts',[ProductController:class,'store']); Ez a kérés lehetővé teszi az adminisztrátor számára, hogy új terméket adjon hozzá az alkalmazáshoz. A post metódus itt is használatos, mert az adminisztrátornak meg kell adnia a termék adatait.
13. **Route:post**('updateProducts/{id}',[ProductController:class,'updateProduct']); Ez a kérés lehetővé teszi az adminisztrátor számára, hogy frissítse a termék adatait. A post metódus itt is használatos, mert az adminisztrátornak meg kell adnia az új termékadatokat.
14. **Route:delete**(deleteProcts/{id},[ProductContoller,class],'destroy'); Ez a kérés lehetővé teszi az adminisztrátor számára, hogy törölje a terméket az alkalmazásból. A delete metódus azt jelenti, hogy az alkalmazás törli a megadott azonosítójú terméket.



Ezek a az útvonalak szükségesek, hogy a Frontend a megfelelő adatokkal tudjon dolgozni a backend információival.

## Backend tesztje

### Category Test

```
class CategoryTest extends TestCase
{
    use RefreshDatabase;
    public function test_get_valid_categories(): void
    {
        Category::factory()->create([
            'CategoryName' => 'testcategory',
            'CategoryId' => 1
        ]);
        $response = $this->getJson('/api/categories');
        $response->assertStatus(200);
        // dd($response[0]);
        $response->assertJson([
            [
                'CategoryName' => 'testcategory',
                'CategoryId' => 1
            ]
        ]);
    }

    public function test_try_get_categories_with_invalid_root(): void{
        $response = $this->getJson('/api/categoriess', []);
        $response->assertStatus(404);
    }
}
```

1. **test\_get\_valid\_categories()**: Ebben a függvényben létrehozunk egy kategóriát egy factory segítségével, majd lekérdezzük az API végponton (/api/categories) keresztül a kategóriákat. A válasz státuszát ellenőrzi (200-asnak kell lennie), és azt is, hogy a válasz tartalmazza-e a várt JSON adatokat.
2. **test\_try\_get\_categories\_with\_invalid\_root()**: Ez a függvény próbálkozik kategóriák lekérdezésével egy érvénytelen API végponton (/api/categoriess), és ellenőrzi, hogy a válasz státusza 404 (Not Found).



## Login Test

```
class LoginTest extends TestCase
{
    use RefreshDatabase;
    public function test_user_login_with_valid_data(): void
    {
        $email = 'test@test.com';
        $password = '12345678';

        $user = \App\Models\User::factory()->create([
            'email' => $email,
            'password' => Hash::make($password)
        ]);

        // dd($user)
        $response = $this->postJson('/api/login', [
            'email' => $user->email,
            'password' => $password
        ]);

        // dd($response);

        $response->assertStatus(200);
        $response->assertJson(["user" => [
            'email' => $email,
        ]]);

        $response->assertJsonStructure(['token']);
    }
}
```

```
public function test_invalid_user_cannot_login(): void
{
    $email = 'non_existing_user@test.com';
    $password = '12345678';

    $response = $this->postJson('/api/login', [
        'email' => $email,
        'password' => $password
    ]);

    $response->assertStatus(401);
}
```

1. **test\_user\_login\_with\_valid\_data():** Ez a függvény teszteli, hogy a felhasználó be tud-e jelentkezni érvényes adatokkal.
  - Létrehoz egy felhasználót 'test@test.com' e-mail címmel és '12345678' jelszóval.
  - POST kérést küld az '/api/login' végpontra a felhasználó e-mail címével és jelszavával.
  - A teszt ellenőrzi, hogy a válasz státusza 200 (OK) kell legyen, és ellenőrzi, hogy a visszaküldött adatok tartalmazzák-e a helyes e-mail címet és egy tokenet.
  
2. **test\_invalid\_user\_cannot\_login():** Ez a függvény teszteli, hogy egy érvénytelen felhasználó nem tud bejelentkezni.
  - Megpróbál bejelentkezni egy nem létező e-mail címmel ('non\_existing\_user@test.com') és '12345678' jelszóval.
  - A teszt ellenőrzi, hogy a válasz státusza 401 (Unauthorized) kell legyen, ami a sikertelen bejelentkezést jelzi.

## Product Test

```
class ProductTest extends TestCase
{
    use RefreshDatabase;
    public function test_get_valid_products(): void
    {
        // Category::factory()->create([
        //     'CategoryName' => 'testcategory',
        //     'CategoryId' => 1
        // ]);
        $categories = \App\Models\Category::factory()->create([
            'CategoryName' => 'testcategory',
            'CategoryId' => 1
        ]);
        Product::factory()->create([
            'ProductNumber' => 1,
            'ProductName' => 'testproduct',
            'ProductDescription' => 'testproductdescription',
            'RetailPrice' => 1000,
            'QuantityOnHand' => 10,
            'CategoryID' => $categories->CategoryId,
            'Image' => 'testimage.jpg',
        ]);
        $response = $this->getJson('/api/products');
        $response->assertStatus(200);
        $response->assertJson([
            [
                'ProductNumber' => 1,
                'ProductName' => 'testproduct',
                'ProductDescription' => 'testproductdescription',
                'RetailPrice' => 1000,
                'QuantityOnHand' => 10,
                'CategoryId' => 1,
                'image' => 'testimage.jpg',
            ]
        ]);
    }
    public function test_try_get_products_with_invalid_root(): void{
        $response = $this->getJson('/api/productss', []);
        $response->assertStatus(404);
    }
}
```

1. **test\_get\_valid\_products():** Ez a függvény teszteli, hogy érvényes termékek lekérdezhetőek-e egy API végponton.
  - Létrehoz egy kategóriát és egy terméket egy factory segítségével.
  - GET kérést küld az '/api/products' végpontra.
  - A teszt ellenőrzi, hogy a válasz státusza 200 (OK) kell legyen, és ellenőrzi, hogy a visszaküldött adatok tartalmazzák-e a várt JSON adatokat.
  
2. **test\_try\_get\_products\_with\_invalid\_root():** Ez a függvény teszteli, hogy mi történik, ha érvénytelen gyökérrel próbálunk termékeket lekérdezni.
  - GET kérést küld az '/api/productss' végpontra.
  - A teszt ellenőrzi, hogy a válasz státusza 404 (Not Found) kell legyen, ami azt jelzi, hogy a kérés sikertelen volt.

## Register Test

```
class RegisterTest extends TestCase
{
    use RefreshDatabase;

    /**
     * A basic feature test example.
     */
    public function test_user_registration_success_with_valid_data(): void
    {
        $name = "Jóska";
        $email = 'aaaa4444aaas@gmail.com';
        $adress = '2676 Cserhátsturány Telep utca 4';
        $phone_number = '+5544445555';
        $password = '12345678';

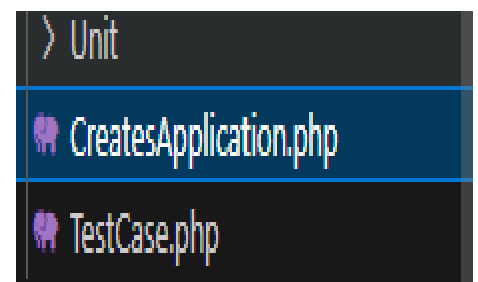
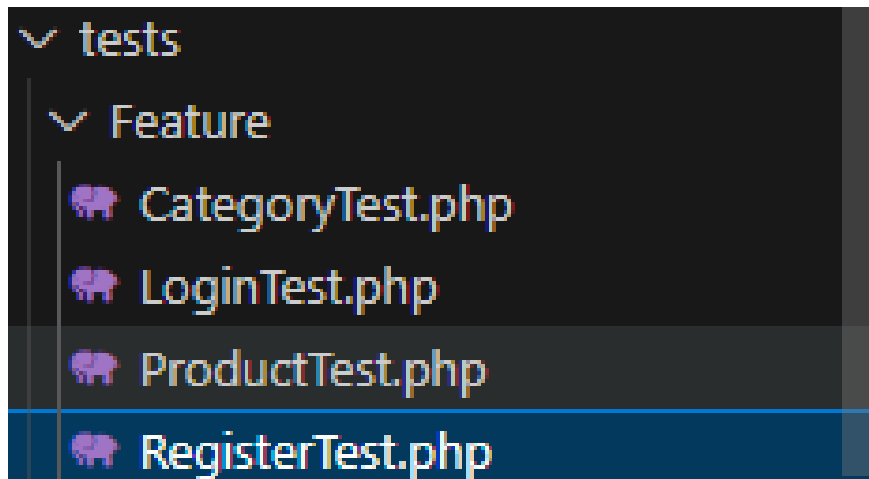
        $response = $this->postJson('/api/register', [
            'name' => $name,
            'email' => $email,
            'adress' => $adress,
            'phone_number' => $phone_number,
            'password' => Hash::make($password)
        ]);

        $response->assertStatus(201);
        $response->assertJson([ "user" => [
            'name' => $name,
            'email' => $email,
            'adress' => $adress,
            'phone_number' => $phone_number,
            // 'password' => Hash::make($password)
        ]
        ]);
        $response->assertJsonStructure(['token']);
    }

    public function test_user_registration_fails_without_valid_data(): void
    {
        $response = $this->postJson('/api/register', []);

        $response->assertStatus(422);
    }
}
```

1. **test\_user\_registration\_success\_with\_valid\_data():** Ez a függvény teszteli, hogy a felhasználó regisztrációja sikeres-e érvényes adatokkal.
  - Definiál néhány változót mintaértékekkel, mint például a név, e-mail cím, cím, telefonszám és jelszó.
  - POST kérést küld az '/api/register' végpontra a felhasználó adataival.
  - A teszt ellenőrzi, hogy a válasz státusza 201 (Created) kell legyen, ami azt jelzi, hogy a regisztráció sikeres volt.
2. **test\_user\_registration\_fails\_without\_valid\_data():** Ez a függvény teszteli, hogy a felhasználó regisztrációja sikertelen-e érvénytelen adatokkal. Sajnos a teljes függvény nem látható a képen.



## Frontend fejlesztése

A frontend fejlesztés során a Vue.js keretrendszert választottuk, és ezt a döntést számos előny és szempont indokolta, különösen egy gamer webáruház szemszögéből.

A Vue.js egy könnyen tanulható és használható keretrendszer, amely lehetővé teszi a gyors és hatékony frontend fejlesztést. A rendkívül intuitív és moduláris szerkezet lehetővé teszi számunkra, hogy gyorsan létrehozzunk összetett felhasználói felületeket és dinamikus webalkalmazásokat, ami kritikus fontosságú egy webáruház esetében.

A Vue.js kiváló teljesítményt nyújt, ami különösen fontos egy olyan platformon, amely magas interaktivitást és gyors válaszidőket igényel, mint egy gamer webáruház. A könnyű súlyú és hatékony virtuális DOM alapú megközelítés lehetővé teszi a gyors oldalbetöltést és az élő, folyamatos élményt a felhasználók számára.

Az előre elkészített komponensek és moduláris felépítés lehetővé teszi, hogy újrafelhasználjuk és újrahasznosítsuk a kódot, ami gyorsabb és hatékonyabb fejlesztést eredményez. Emellett a Vue.js széles körű közössége és az aktív fejlesztői közösség biztosítja, hogy mindig hozzáférhetünk a legfrissebb eszközökhöz, megoldásokhoz és támogatáshoz.

A Vue.js segítségével könnyedén kezelhetjük az adatokat és az állapotokat az alkalmazásban, ami lehetővé teszi számunkra, hogy dinamikus és interaktív felhasználói felületeket hozzunk létre. Ez lehetővé teszi, hogy az ügyfelek könnyen böngésszenek és vásároljanak a webáruházunkban, miközben élvezik a sima és gyors felhasználói élményt.


Végül, a Vue.js könnyen integrálható más technológiákkal és szolgáltatásokkal, így lehetővé teszi számunkra, hogy rugalmasan bővítsük és fejlesszük a webáruházunkat a jövőbeni igények szerint. Ez biztosítja, hogy a webáruházunk mindig naprakész és versenyképes maradjon a gyorsan változó piaci környezetben.

## Fejléc:


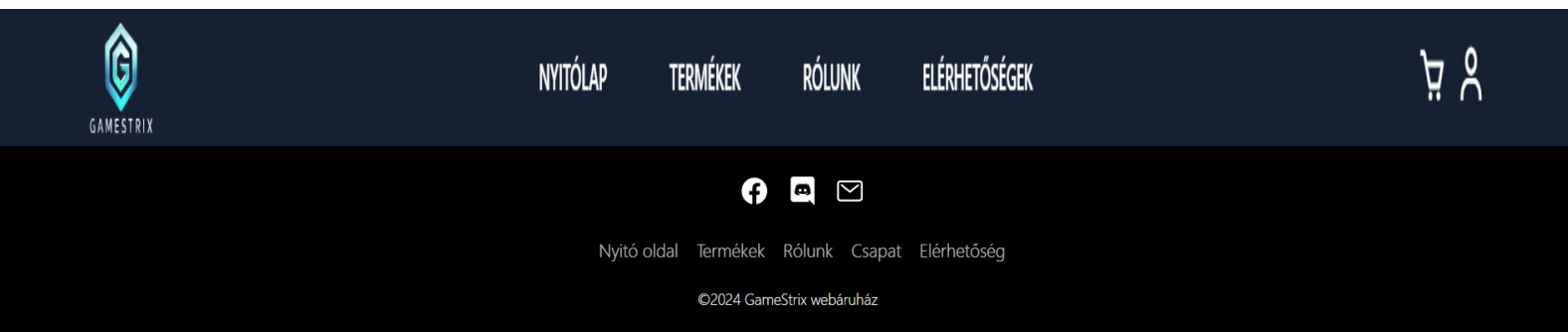
A weboldal fejlesztését egy egyszerű, ám hatékony szerkezettel indítottuk, amely tartalmazott egy fejléct (navbar) és egy lábléct (footer). Ezek a részek szolgáltak az alapjául annak, hogy megbeszéljük és eldöntsük az összes további oldal elrendezését és tartalmát a weboldalon.

Az egyszerű szerkezet mellett döntöttünk azért, hogy könnyű és átlátható legyen a navigáció a felhasználók számára. Az egyértelmű és könnyen követhető elrendezés segít abban, hogy a látogatók gyorsan megtalálják az általuk keresett információkat anélkül, hogy elvesznének az oldalon.

A Header elemei az AppNav.vue állományban találhatóak:

 AppNav.vue

A Footer elemei az AppFooter.vue állományban találhatóak:

 AppFooter.vue

## Navbár teszt:

```
describe('NavbárTesztje', () => {
  it('LéteziktermékekFül', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Termékek').click()
    cy.url().should('include', '/termek')
  })
  it('LétezikBejelentkezésFül', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Bejelentkezés').click()
    cy.url().should('include', '/bejelentkezes')
  })
  it('LétezikRegisztrációFül', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Regisztráció').click()
    cy.url().should('include', '/regisztracio')
  })
  it('LétezikNyitóOldalFül', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Nyitó oldal').click()
    cy.url().should('include', '/')
  })
})
```

### NavbárTesztje

- ✓ LéteziktermékekFül
- ✓ LétezikBejelentkezésFül
- ✓ LétezikRegisztrációFül
- ✓ LétezikNyitóOldalFül

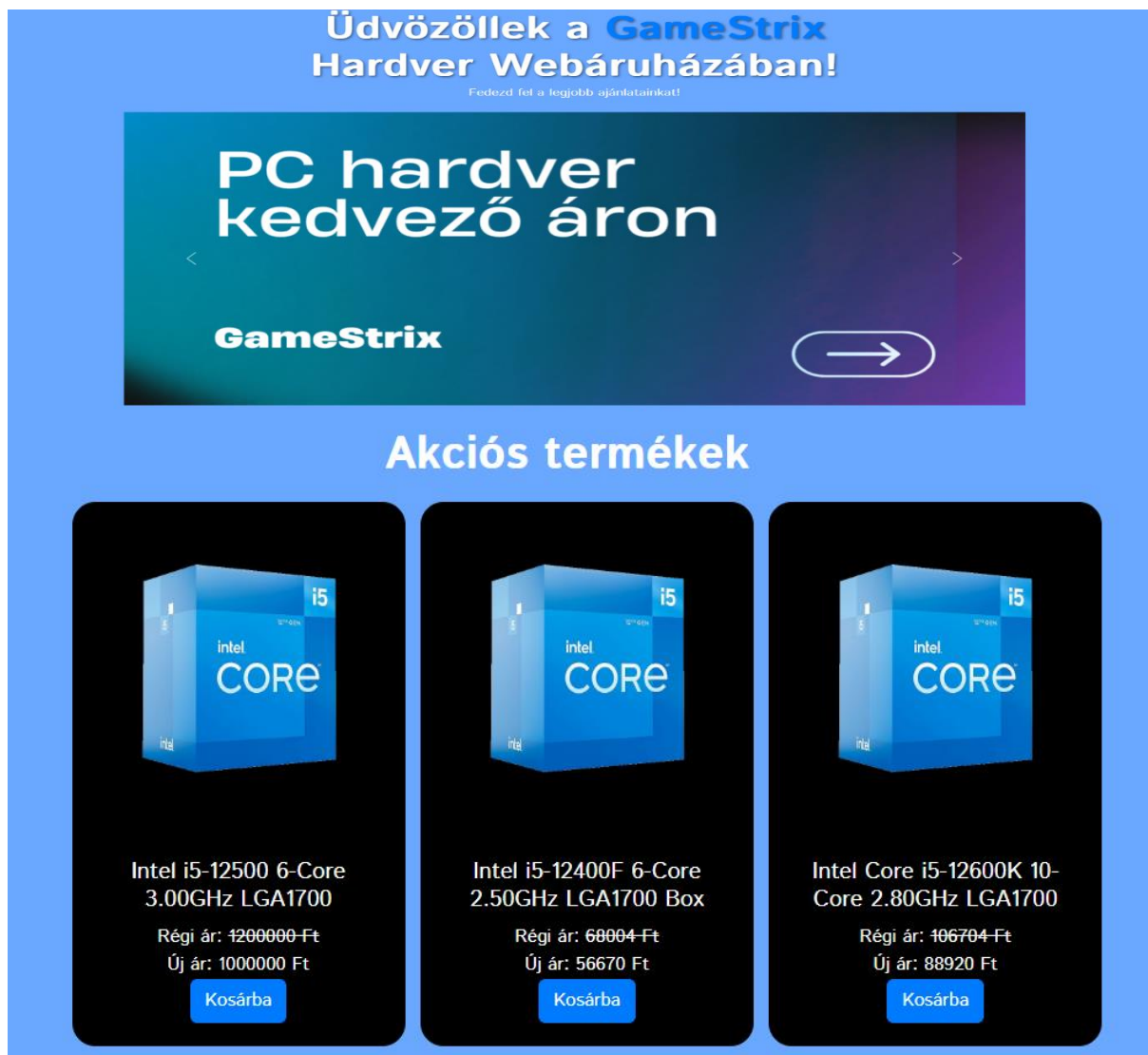


## Nyitó oldal:

A nyitóoldal a weboldalunk egyik látványos és fontos része. Ez az oldal szolgál bevezetőként, ahol a látogatók elsőként találkozhatnak a kínálatunkkal.

A nyitóoldalon kilenc, kedvezményes árú PC alkatrész található, amelyek között a látogatók válogathatnak. Minden termékhez tartozik egy rövid leírás és egy "Megnézem" gomb, amely további információkhoz vezet.

Ez a bevezető oldal szolgál előzetes betekintést nyújt a termékeinkbe, és felkelti a látogatók érdeklődését, hogy tovább böngésszenek a teljes termékoldalon. A nyitóoldal kialakítása során arra törekedtünk, hogy egy vizuálisan vonzó és informatív felületet hozzunk létre, amely megfelel a felhasználók igényeinek és lehetővé teszi számukra, hogy könnyedén és kényelmesen válogassanak a kedvezményes alkatrészek között.



Üdvözöllek a **GameStrix** Hardver Webáruházában!

Fedezd fel a legjobb ajánlatainkat!

# PC hardver kedvező áron

**GameStrix**

## Akciós termékek

Intel i5-12500 6-Core 3.00GHz LGA1700	Intel i5-12400F 6-Core 2.50GHz LGA1700 Box	Intel Core i5-12600K 10-Core 2.80GHz LGA1700
Régi ár: 1200000 Ft Új ár: 1000000 Ft	Régi ár: 68004 Ft Új ár: 56670 Ft	Régi ár: 106704 Ft Új ár: 88920 Ft
<a href="#">Kosárba</a>	<a href="#">Kosárba</a>	<a href="#">Kosárba</a>

```
onMounted(() => {  
  shoppingStore.fetchProducts();  
  useProductData();  
});
```

```
function getPorducts(page) {  
  isPaged.value = true  
  console.log(actualPage.value);  
  productservices.getProductsPaged(actualPage.value).then((resp) => {  
    products.value = resp.data;  
    products.value.pop()  
    lastPage.value = Math.ceil(resp.total / 10);  
    window.scrollTo({  
      top: 0,  
      behavior: "smooth"  
    })  
  });  
}
```


```
import {  
  ref,  
  onMounted,  
  computed  
} from "vue";  
import {  
  useShoppingStore  
} from "@stores/shoppingStore";  
import productservices from "../services/productservices";  
import {  
  useRouter  
} from "vue-router";
```

Az akciós termékek cardokba kérését úgy oldottuk meg, hogy a pagináláshoz használt oldalanként 10-es termék lekérést használjuk akciós termékeknek és egy elemet pop függvény segítségével eltávolítottuk az esztétika miatt.


## Termékek Oldal:

A Termékek oldal egy áttekinthető és funkcionális felületet biztosít a látogatóknak, ahol könnyedén és kényelmesen válogathatnak a különböző PC alkatrészek között. A termékek részletes bemutatása és az átlátható kategóriák segítenek a felhasználóknak a keresésben és a döntésben. Ez az oldal hozzájárul a felhasználói élmény javításához és a weboldalunk sikeréhez.


Összes termék
Processzor
Videokártya
Alaplap
Ram
Tápegység
Háttértár
Gépház




INTEL I5-12500 6-CORE  
3.00GHZ LGA1700  
**1000000 Ft**




INTEL I5-12400F 6-CORE  
2.50GHZ LGA1700 BOX  
**56670 Ft**




INTEL CORE I5-12600K 10-CORE  
2.80GHZ LGA1700  
**88920 Ft**




ASUS GEFORCE RTX 4070  
DUAL 12GB GDDR6X OC  
**261710 Ft**




GIGABYTE GEFORCE RTX 4060  
8GB GDDR6  
**137550 Ft**




ASUS GEFORCE RTX 4090  
OC 24GB GDDR6X  
**943690 Ft**




ASUS TUF GAMING B760M-PLUS  
ALAPLAP  
**70539 Ft**



ASUS ROG STRIX B760-G  
GAMING WIFI D4 ALAPLAP  
**78340 Ft**



ASUS TUF GAMING Z690-PLUS  
ALAPLAP  
**79990 Ft**



G.SKILL TRIDENT Z5 RGB  
32GB (2X16GB) DDR5  
7200MHZ  
**70940 Ft**

1
Next

Amennyiben a kurzort a kártyákra húzzuk megjelenik kosárba helyezés gomb ha rákattintunk akkor a termék azonnal a kosárba kerül, azonban aha a kártyára kattintunk az adott termék információit láthatjuk.

## Adatok lekérése:

```
onMounted(() => {  
  shoppingStore.fetchProducts();  
  useProductData();  
});
```

```
function getPorducts(page) {  
  isPaged.value = true  
  console.log(actualPage.value);  
  productservices.getProductsPaged(actualPage.value).then((resp) => {  
    products.value = resp.data;  
    products.value.pop()  
    lastPage.value = Math.ceil(resp.total / 10);  
    window.scrollTo({  
      top: 0,  
      behavior: "smooth"  
    })  
  });  
}
```

```
import {  
  ref,  
  onMounted,  
  computed  
} from "vue";  
import {  
  useShoppingStore  
} from "@stores/shoppingStore";  
import productservices from "../services/productservices";  
import {  
  useRouter  
} from "vue-router";
```

```
const goToProductPage = (ProductName) => {  
  router.push({  
    name: "Termek",  
    params: {  
      ProductName  
    }  
  });  
};
```

A termékeket oldalanként 10 kártyában jelenítjük meg.  
Gomb segítségével tudunk léptetni az oldalak között Paginálás segítségével.

## Kosárba helyezés:

```
const selectedCategory = ref(null);

const store = useShoppingStore();
const rout = useRoute();

const SelectedProduct = computed(() => {
  return store.products.find((product) => product.ProductName === rout.params.ProductName);
});

const addToCart = (product) => {
  store.addToCart(product);
};
```

## Szűrés:

Összes termék   Processzor   Videokartya   Alaplap   Ram   Tápegység   Háttértár   Gépház

```
function selectCategory(id) {
  productservices.getProductByCategoryId(id).then((resp) => {
    isPaged.value = false
    actualPage.value = 1;
    products.value = resp;
  })
}
```

## ProductDetails oldal:



### Intel I5-12400F 6-Core 2.50GHz LGA1700 Box

Ár: 56670 Ft

#### Termék Leírás:

A 12. generációs Intel® Core™ asztali processzor újradefiniálja az x86 architektúra teljesítményét. Bemutatjuk új hibrid teljesítményű architektúránkat, amely a játék, a termelékenység és a kreativitás fokozása érdekében egyesíti a teljesítménymagokat a hatékony magokkal. Ezek a forradalmi processzorok intelligensen optimalizálják a munkaterhelést, és megnyitják az utat a processzorvezetés jövőbeli ugrásai előtt. Élvezze a legújabb platform-innovációk teljes skáláját, például az iparágban elsőként elérhető PCIe 5.0 és a DDR5 memória. Az Intel® UHD grafikus processzorral merüljön el a lenyűgöző vizuális élményben az akár 8K HDR-támogatással és a 4 K kijelző egyidejű megtekintésének lehetőségével. A 12. generációs Intel® Core™ asztali processzorok minden olyan funkciót biztosítanak, amelyre a játékhoz, munkához és alkotáshoz szüksége van, mint még soha.

Raktáron: 100 db

Kategória:

Szállítás: Belföldön bárhova.

Szállítási költség: 1.500 Ft (100.000 Ft felett ingyenes)

Kosárba 

```

<div class="product-detail">
  <h2>Termék leírás: </h2>
  <p>{{ SelectedProduct.ProductDescription }}</p>
  <ul>
    <li>Raktáron: <span>{{ SelectedProduct.QuantityOnHand }} db</span></li>
    <li>Kategória: <span>{{ selectedCategory.Category }}</span></li>
    <li>Szállítás: <span>Belföldön bárhova.</span></li>
    <li>Szállítási költség: <span>1.500 Ft (100.000 Ft felett ingyenes)</span></li>
  </ul>
</div>


```

Egy esztétikus kis oldal ahol szépen lekéri az adott termék adatait.

Lehetőségünk van itt is kosárba helyezni a terméket.

## Rólunk oldal:

A Rólunk oldal olyan rész, amely bemutatja az üzletet, szolgáltatást vagy vállalkozást, és részletesen ismerteti annak céljait, küldetését, értékeit és a mögötte álló csapatot. Általában információkat nyújt az üzlet tevékenységi köréről, annak történetéről, filozófiájáról és az ügyfelek számára nyújtott előnyökről. A "Rólunk" oldal fontos lehet az üzlet hitelességének és megbízhatóságának megteremtésében, valamint az ügyfelek bizalmának növelésében. Általában az oldalon szöveges tartalmak, képek és esetlegesen videók is helyet kapnak, hogy teljes képet adjanak az üzletről vagy vállalkozásról.



GAMESTRIX

## Rólunk

Üdvözöllek a Gamestrix webshop oldalán! Mi, a Gamestrix csapata, három szenvedélyes gamer vagyunk, akik azért alapították ezt a webshopot, hogy az emberek számára könnyen hozzáférhetővé tegyünk a legjobb gamer hardvereket a legjobb áron.

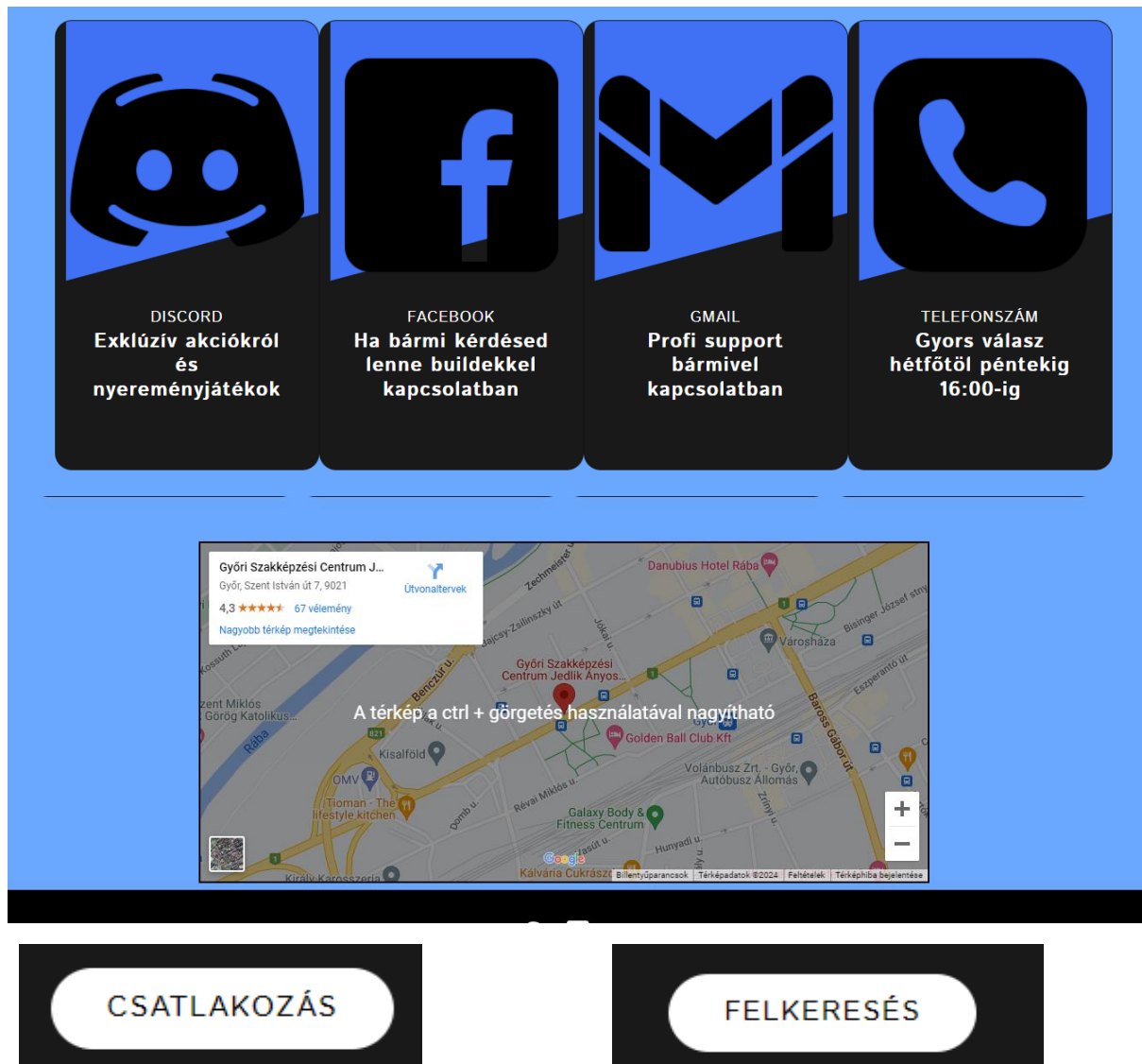
A Gamestrix webshopot azon küldetés vezérli, hogy lehetővé tegye a gamerek számára, hogy kiváló minőségű, megbízható és versenyképes árú hardverekhez jussanak. Termékeink között megtalálhatóak a legújabb videokártyák, processzorok, alaplapok, memóriamodulok, SSD-k, házak, tápegységek és egyéb kiegészítők a legnépszerűbb márkáktól.

Nemcsak a termékekről van szó - a Gamestrix egy olyan közösség is, amely segít a gamereknek a legjobb felszereléssel és információkkal rendelkezni. Szakértő csapatunk készséggel áll rendelkezésedre, hogy tanácsot adjon és segítsen kiválasztani a legmegfelelőbb hardvereket az igényeidnek megfelelően.

Legyen szó akár egy új PC építéséről, akár csak egy egyszerű frissítésről, a Gamestrix mindent megtesz azért, hogy biztosítsa, hogy minden ügyfelünk elégedett legyen a vásárlás után. Csatlakozz hozzánk, és tapasztald meg a legjobb gamer hardvereket és szolgáltatásokat a Gamestrix-től!



## Elérhetőségek oldal:



Egyszerű ,azonban stílusos kártyás rendezés .

Amennyiben a kurzort az adott kártyára irányítjuk megjelenik a kapcsolatfelvételi gomb az adott metódusnak megfelelően.

A kártyák alatt láthatjuk a az áruház pontos címét a google mapson megjelölve.



## Kosár:

A kódok a webáruház kosár funkcióját kezelik, számított tulajdonságokat biztosítva a kosárban lévő tételek összértékéhez és a szállítási költséghez, valamint lehetőséget biztosít a rendelés leadására.

```
import { computed } from "vue";
import { storeToRefs } from "pinia";
import { useUserStore } from "../stores/userstore";
import { useRouter } from "vue-router";
const { status } = storeToRefs(useUserStore());

// get store
const data = useShoppingStore();
const alapSzallitas = 1500;
const ingyenesSzallitasHatar = 100000;
const router = useRouter();

const teljesAr = computed(() => {
  return data.cartItems.reduce(
    (osszeg, t) => osszeg + t.RetailPrice * t.quantity,
    0
  );
});

const szallitas = computed(() => {
  return teljesAr.value >= ingyenesSzallitasHatar ? 0 : alapSzallitas;
});

const teljesArSzallitassal = computed(() => {
  return teljesAr.value + szallitas.value;
});
```

```
const placeOrder = () => {

  const orderData = [];
  data.cartItems.forEach((product) => {
    orderData.push({
      ProductNumber: product.ProductNumber,
      QuantityOrdered: product.quantity,
      QuotedPrice: product.RetailPrice,
    });
  });
  data.placeOrder(orderData);
};
```

## KOSÁR TARTALMA



**Intel Core i5-12600K 10-Core  
2.80GHz LGA1700**

88920 Ft

⊕ 1 ⊖

Törlés

Termék(ek) ára: 88920 Ft

Szállítás költsége: 1500 Ft

Összesen 90420 Ft

 **Megrendelés**

## Bejelentkezés/Regisztráció:

### Bejelentkezés

Email cím

Jelszó

LOGIN

Elfelejtett jelszó? [Visszaállítás](#)

Nincs még fiókod? [Hozz létre egyet!](#)

### Regisztráció

Név

E-mail

Jelszó

Lakcím

Telefonszám

Regisztráció

Van már fiókod? [Jelentkezz be!](#)

## Belépés folyamata:

```
import { ref } from "vue";
import { storeToRefs } from "pinia";
import { useUserStore } from "../../stores/userstore";
import { useRouter } from "vue-router";
import "primeicons/primeicons.css";
const { login } = useUserStore();
const { status } = storeToRefs(useUserStore());

const router = useRouter();

const loginForm = ref({});
const logSuccess = ref(false);
const loginError = ref(false);

function onLogin() {
  loginError.value = false; // Alaphelyzetbe állítás minden bejelentkezési kísérlet előtt
  login(loginForm.value).then(() => {
    logSuccess.value = true;
    setTimeout(() => {
      router.push("/");
    }, 2000);
  }).catch(() => {
    loginError.value = true; // Hiba esetén a loginError értékét true-ra állítjuk
  });
}
```

## Regisztráció folyamata:

```
const router = useRouter();
const regForm = ref({
  name: "",
  email: "",
  password: "",
  confirm_password: "",
  adress: "",
  phone_number: "",
});

const regSuccess = ref(false);
const errorMessages = ref({});
```

```
function onRegister() {
  // console.log(regForm.value);
  userservice
    .registerUser(regForm.value)
    .then(() => {
      regSuccess.value = true;
      setTimeout(() => {
        router.push("/bejelentkezés");
      }, 2000);
    })
    .catch((err) => {
      // console.log(err.data.data);
      errorMessages.value = err.data.data;
    });
}
```

## Regisztráció és bejelentkezés tesztje:

```
describe('RegisztrációsTeszt', () => {
  it('RegisztrációTeszt', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Regisztráció').click()
    cy.get('.nev').type('Proba')
    cy.get('.email').click().type('ErwI@L')
    cy.get('.jelszo').click().type('321')
    cy.get('.telefonszam').click().type('0631234567')
    cy.get('.lakcim').click().type('Tes213fass')
    cy.get('.submit').click()
  })
})

describe('BejelentkezésTeszt', () => {
  it('BejelentkezésTeszt', () => {
    cy.visit('http://localhost:5173')
    cy.contains('nav a', 'Bejelentkezés').click()
    cy.get('#email').type('admin@admin.com')
    cy.get('#password').click().type('admin123')
    cy.get('#submitlog').click()
    cy.url().should('include', '/')
  })
})
```

- ▼ RegisztrációsTeszt
  - ✓ RegisztrációTeszt
- ▼ BejelentkezésTeszt
  - ✓ BejelentkezésTeszt

## Összegzés

Az egész folyamatot visszatekintve, elmondhatom, hogy izgalmas és tanulságokban gazdag volt. A feladat nagyon érdekes volt, és nagyon örültünk, hogy egy ilyen izgalmas projektbe foghattunk. Természetesen voltak nehézségek és kihívások, de ezek mind részei voltak az építő folyamatnak.

A csapatunk fantasztikus volt. Mindannyian lelkesen és elkötelezetten dolgoztunk azon, hogy létrehozzunk valami kivételeset. A kommunikációnk hatékony volt, és mindig nyitottak voltunk egymás ötleteire és visszajelzéseire. Bár mindenki más volt a szakterületén, nagyon jól működöttünk együtt, és támogattuk egymást a feladatok végrehajtásában.

A tudásunkat tekintve mindannyian sokat tanultunk a projekt során. Mindannyian kihoztuk magunkból a legtöbbet, és megmutattuk, hogy mire vagyunk képesek. Persze, voltak olyan területek, ahol kevésbé voltunk jártasak, vagy ahol nehezebben boldogultunk, de ezek mind hozzájárultak a fejlődésünkhöz.

Sikerült létrehoznunk egy nagyszerű webshopot, amelyre büszkék lehetünk. Természetesen voltak olyan dolgok, amiket jobban csinálhattunk volna, vagy amiket későbbre kellett halasztanunk, de ezek mind részei voltak a tanulási folyamatnak. Mindent egybevetve, ez egy fantasztikus tapasztalat volt, amit soha nem felejtünk el.

# Források

A frontend: [Sajtoschips/Vizsgaremek\\_2 \(github.com\)](https://github.com/Sajtoschips/Vizsgaremek_2)

Backend: [Horvathmate7514/gamestrixBackend \(github.com\)](https://github.com/Horvathmate7514/gamestrixBackend)

Mobil app: [Horvathmate7514/GameStrixMobil \(github.com\)](https://github.com/Horvathmate7514/GameStrixMobil)