



Vas Megyei Szakképzési Centrum
Nádasy Tamás Technikum és Kollégium

SZAKDOLGOZAT

CyberBuild

Bíró Bálint, Németh Márk

Konzulens:
Balics Gábor

2023

Nyilatkozat

Alulírott, **Bíró Bálint**, **Németh Márk** szakos hallgató kijelentem, hogy a CyberBuild című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Csepreg, 2023.04.23.

hallgató

Kivonat

[CyberBuild]

[1 oldalas, magyar nyelvű tartalmi kivonat]

Abstract

[CyberBuild]

[1 oldalas, angol nyelvű kivonat]

Tartalomjegyzék

1.	BEVEZETŐ	7
1.1.	TÉMAVÁLASZTÁS	7
1.1.1.	<i>Igények.....</i>	<i>9</i>
1.1.2.	<i>Ötletmerítés, könyvtárprogramok</i>	<i>10</i>
2.	PROGRAMOZÁSI NYELV	11
2.1.	C#	11
2.2.	REACT.....	12
2.3.	SQL	12
3.	ADATBÁZIS.....	14
3.1.	LÉTREHOZÁS	14
4.	PROGRAMOK.....	16
4.1.	MICROSOFT VISUAL STUDIO 2022	16
4.2.	MICROSOFT SQL	17
5.	FELHASZNÁLÓI FELÜLET (UI)	18
5.1.	KEZDŐKÉPERNYŐ	18
5.1.1.	<i>Logó</i>	<i>20</i>
5.1.2.	<i>Kereső és fejléc</i>	<i>21</i>
5.2.	FŐMENÜ	22
5.3.	FELHASZNÁLÓ	23
6.	FIZETÉS ÉS SZÁLLÍTÁS	25
7.	TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	28
8.	ÖSSZEGZÉS.....	30
9.	BIBLIOGRÁFIA.....	31
10.	IRODALOMJEGYZÉK.....	ERROR! BOOKMARK NOT DEFINED.
11.	MELLÉKLETEK.....	32
11.1.	[A DOLGOZAT MELLÉKLETEI, HA VANNAK]	32

1. Bevezető

CyberBuild

Köszöntjük a CyberBuild számítástechnikai webáruház dokumentációjában!

Ebben a jegyzőkönyvben bemutatjuk az általunk létrehozott projektfeladat elkészítését és fejlesztését, nulláról a végkifejletig.

A témák nagybetűvel, az alcímek pedig kisbetűvel jelölve szerepelnek!

A dokumentáció segítséget nyújt a CyberBuild webshop könnyebb használatához, ezzel problémák sokaságát kiküszöbölve.

A webáruházunk minden olyan ember számára ajánlott, akik számítógépet szeretnének vásárolni, esetleg a meglévőt fejleszteni, vagy csupán érdekli őket, hol, hova is tart jelenleg ez az iparág és a technológia.

Szeretnénk köszönetet mondani oktatóinknak, akik a konzultációk során biztosított létfontosságú tanácsaikkal hatalmas segítséget nyújtottak nekünk.

Köszönjük mindenkinek!

1.1. Témaválasztás

2022 szeptemberében kezdetét vette az ötödik és egyben utolsó tanévünk az iskolában, ami nem jelentett mást, mint, hogy projektfeladatot kell készítenünk az év során, mely szükséges az év végén esedékes szakmai vizsgához.

Minimum kettő, de maximum három főből álló fejlesztői csapatokat kellett alkotni.

Fontos volt egymás ismerete, maga a háttér és hogy együtt tudjunk működni.

Programozói tudásunk nagyjából egy szinten áll és régóta ápolunk baráti viszonyt, ezért jött létre a csapatunk.

Témaválasztásnál a célkitűzés az volt, hogy egy egyedi szoftveralkalmazást fejlesszünk le, amely egy életszerű, ám ugyanakkor egy valódi problémára nyújthat megoldást.

Nem szeretnénk volna túlvállalni magunkat egy nagyon nehéz és összetett projekttel, ezért egy számítástechnikai webáruház lefejlesztésében gondolkodtunk.

A téma leadási határideje október eleje volt. Pár hétbe ugyan beletelt, mire meg tudtuk beszélni a véleményünket egymással, rengeteg ötlet szóba jött, többek között egy videojáték is felmerült, de itt az elképzeléseink és az elvárásunk túl magas volt, így ez el is lett vetve. Azonban hosszas halasztások után végül letettük a voksunkat a webshop mellett. Ekkor kezdődött a tervezés fázisa.

Rengeteg papírmunka volt vele, készültek sorban a PowerPoint prezentációk és a Word dokumentumok, melyeket oktatóink felé kellett benyújtani hétről-hétre, hogy ők maguk is lássák a fejlesztés folyamatát.

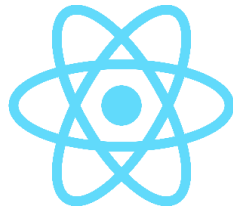
Egyre inkább körvonalazódott, hogy mit is akarunk létrehozni a projektből, felosztottuk egymás között a feladatokat, ki-ki maga tudásához mérten vállalta a feladatrészeket.

Teltek-múltak a hetek, az iskola mellett alakulóban volt a szakdolgozat, mindennap összeültünk, és egy keveset hozzáadtunk a már meglévő anyagokhoz.

A következő probléma, amibe ütköztünk az a technológia kiválasztása volt. Itt is sorban jöttek az ötletek, hol jobbak, hol rosszabbak.

Először a „Node.js” mellett gondolkodtunk, aztán jött a „Blazor”, később a jelenlegi és végleges „C# és React” kombináció.

Úgy találtuk, hogy ehhez a két nyelvhez találhatjuk a legtöbb segítséget, és tanáraink is napi szinten használják őket, ezáltal naprakész információval tudnak minket ellátni arra az esetre, ha bármilyen kérdés adódna a fejlesztés során.



1.1.1. Igények

Az egyik legfontos szempont az volt, hogy egy működő, a mindennapok problémáira megoldást nyújtó szoftveralkalmazást fejlesszünk le, amely mögött serveroldali adatbázis is szerepel.

Amikor kezdetét vette a tervezés fázisa, többféle igény állt előttünk, jobbnál-jobb tervekkel, ötletekkel.

Túlvállalni ugyan nem szerettük volna magunkat, viszont egy egyszerű szoftvert sem szerettünk volna kiadni a kezeink közül.

Elkezdtek felmérni az igényeket, melyeket papírra vetettünk, órák hosszát beszéltek át.

Végül az alábbi igényeket szedtük össze:

- termékek széles választéka → minden vásárló megtalálja azt a terméket, amire vágyik
- kosár, kívánságlista → elengedhetetlen funkciók egy webshopnál, kosárban tudjuk összegyűjteni a kiválasztott termékeket
- keresőmotor → az áruház termékeire kulcsszavakkal rá tudunk keresni
- gépösszerakó funkció → a vásárló kiválasztja az alkatrészeket, majd az oldal kiírja neki, hogy az adott komponensek kompatibilisek-e egymással

1.1.2. Ötletmerítés, könyvtárprogramok

Mikor kiválasztottuk a webáruházat, mint ötletet, felkerestük nevesebb magyar cégek oldalait, amelyek sok segítséget tudtak nyújtani.

Ilyen volt például a „PCX”, amelynél maga a felhasználói felület adta az inspirációt (később ez a rész bővebb kifejtést kap).

Tovább mentünk, és felfedeztük az iPon.hu-t és a Konzolvilágot. Támpontot ugyan tudtak adni (például a termékeket itt néztük), de főképp a PCX-et használtuk segítségnek.

A korábban említett gépösszerakó funkciót már több magyar, illetve külföldi webshop is előszeretettel használja. Nagyon tetszett nekünk ez a tulajdonság.

2. Programozási nyelv

2.1. C#

A C# egy modern, objektumorientált programozási nyelv, amelyet a Microsoft fejlesztett ki. Alkalmazása széleskörű, használják például webfejlesztésre, asztali alkalmazásokhoz és játékfejlesztésre is.

A C#-ban írt programokat a .NET Framework vagy .NET Core futtatja. A nyelv előnyei közé tartozik a szintaktikai egyszerűség, az erős típusos rendszer és a memóriakezelés automatizálása.

A C#-ban írt kódokat egyszerűen integrálhatjuk más Microsoft által fejlesztett alkalmazásokkal, például az SQL szerverrel vagy a SharePointtal.

A C# egy nagyon rugalmas nyelv, amely lehetővé teszi a fejlesztők számára, hogy különböző platformokra fejlesszenek, mint például Windows, Mac vagy Linux.

Az objektumorientált jellegének köszönhetően lehetőséget ad az alkalmazások moduláris kialakítására és a kód újra felhasználására.

A nyelv támogatja a többszálú programozást, így a fejlesztők hatékonyan tudják kihasználni a többmagos processzorok adta előnyöket.

A C# fejlesztéséhez szükséges eszközök, mint például a Visual Studio, elérhetőek az ingyenes Visual Studio Community verzióban is, így bárki elkezdheti tanulni és használni a nyelvet.

A C# nyelvet először 2022 őszén kezdtük el használni és tanulni az iskolában.

Mivel elég könnyen megértettük a felépítését és a logikáját, így döntöttünk amellett, hogy az egész projektfeladatunkat ezen a nyelven írjuk meg.

2.2. React

A React egy nyílt forráskódú, JavaScript alapú front-end keretrendszer, amelyet a Facebook fejlesztett ki. A React lehetővé teszi a fejlesztők számára, hogy hatékonyan hozzanak létre felhasználói felületeket, amelyek interaktívak és gyorsak.

A keretrendszer az ún. "komponens" alapú megközelítésre épül, amely lehetővé teszi a fejlesztők számára, hogy újra felhasználják az előre elkészített részegységeket a felhasználói felületeken, így egyszerűsítve a fejlesztési folyamatot és javítva a kód újra felhasználhatóságát.

A Reactot általában más eszközökkel kombinálják, mint például a Redux vagy a React Router, hogy még hatékonyabb és kiterjedtebb alkalmazásokat lehessen fejleszteni vele.

A React a virtuális DOM (Document Object Model) koncepciójára épül, amely lehetővé teszi a gyorsabb és hatékonyabb frissítéseket a felhasználói felületeken, így javítva a felhasználói élményt.

A React széles körben alkalmazható webalkalmazások, weboldalak, mobilalkalmazások és asztali alkalmazások fejlesztésére. A React-ot használó fejlesztőknek elengedhetetlen a JavaScript ismerete, valamint a modern fejlesztői eszközök használata, mint például a Node.js és a Babel.

A React ökoszisztémája nagyon aktív, számos fejlesztői eszközt, könyvtárat és modult kínál, amelyek lehetővé teszik a fejlesztők számára, hogy még hatékonyabban dolgozzanak vele.

2.3. SQL

Az SQL (Structured Query Language) egy speciális programozási nyelv, amelyet a relációs adatbázis-kezelő rendszerekben használnak. Az SQL lehetővé teszi a felhasználók számára, hogy lekérdezéseket hajtsanak végre, módosítsák és kezeljék az adatbázisban tárolt adatokat.

Az SQL használatával az adatok egyszerűen szervezhetők és könnyen kezelhetők, mivel az SQL lehetővé teszi az adatok egységes kezelését, a redundancia minimalizálását és az

adatintegritás biztosítását.

Az SQL-t az üzleti alkalmazások, a webes alkalmazások és az adatfeldolgozás számos más területén alkalmazzák.

Az SQL-nél a lekérdezéseknek számos formája van, beleértve a SELECT, INSERT, UPDATE és DELETE parancsokat.

Ezek a parancsok lehetővé teszik az adatok lekérdezését, új adatok hozzáadását, meglévő adatok frissítését és adatok törlését az adatbázisból.

Az SQL-t széles körben használják a vállalati környezetben, ahol nagy mennyiségű adatot kell kezelni, valamint az olyan területeken, mint a pénzügy, az egészségügy, az oktatás, a kereskedelem és az ipar.

Az SQL folyamatosan fejlődik és korszerűsödik, így az SQL szakembereknek mindig naprakésznek kell lenniük a legújabb trendekkel és technológiákkal, hogy maximális hatékonysággal tudják használni az adatbázis-kezelő rendszereket.

Az SQL-t először a 2021/2022-es tanévben kezdtük el tanulni, méghozzá MySQL Workbenchben.

Maga a program többek között képes adatmodellt készíteni, adatbázisokat létrehozni és szerkeszteni, valamint adatbázis-kapcsolatokat menedzselni.

A felhasználóbarát felülete lehetővé teszi a könnyű navigációt és az adatok vizualizációját. Kimondottan egyszerű a felépítése, és könnyen lehet benne dolgozni.

Ezen felül, a MySQL Workbench segítségével lehetőség van SQL parancsok végrehajtására is, így könnyen és hatékonyan lehet adatokat lekérdezni, módosítani vagy éppen új adatokat hozzáadni az adatbázisokhoz.

Emellett a program biztonsági mentéseket is készít az adatbázisokról, így lehetőség van az adatok visszaállítására esetleges hibák vagy adatvesztések esetén.

Összességében a MySQL Workbench egy nagyon hasznos eszköz az adatbázisok tervezéséhez, karbantartásához és kezeléséhez.



3. Adatbázis

3.1. Létrehozás

Az első lépés az volt, hogy kiválasszuk azt az adatbázis programot, amellyel könnyen foguk tudni majd dolgozni tovább, egyszerű lesz a kezelése és az átláthatósága a csapat többi tagjának is.

A következő lépés a csapat számára a megfelelő és a webshophoz elengedhetetlen táblák deklarálása volt. Ilyen táblák például a felhasználó, termékek, termék típusok és így tovább. Miután a csapatban közös megegyezés jött létre a szükséges táblákhoz, neki lehetett állnia ezeknek a kifejtésének és megtervezésének.

Mondhatni talán ez volt a leghosszabb része a feladatnak, ugyanis ebben a fázisban kell minden táblának megfogalmazni a tartalmát, hogy mi hogyan fog elhelyezkedni benne és hogyan fog kapcsolódni a többi táblához.

A táblakészítés ezen része azért nagyon fontos, mert egy apró figyelmetlenség, egy kihagyott sor, vagy akár egy rosszul összekapcsolt táblába hatalmas kellemetlenségeket tud szülni.

Ezek leginkább akkor jönnek elő, amikor a backendben meg akarjuk hívni az adatbázist, vagy amikor a feltöltésnél vagyunk.

A hibáknak a felismerése egy nehéz rész, ha az adatbázist koherensen lett megalkotva, akkor a felismerés egyszerűen javítható.

Ugyanakkor még ide tartoznak azok a táblák is, amelyekre későbbi módosítások miatt van szükség, ha esetleg több funkciót akarunk hozzáadni a webshophoz.

Table Name: Scheme:

Character/Collection: Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UN	B	UN	2P	AZ	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
user_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
date	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
total_price	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
shipping_address	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
billing_address	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
payment_method	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
status	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
product	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
quantity	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
shipping_method	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
trading_number	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Column Name: Data Type:

Character/Collection: Collation:

Comments:

Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☒ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Columns Indices Foreign Keys Triggers Partitioning Options

4. Programok

4.1. Microsoft Visual Studio 2022

A Microsoft Visual Studio 2022 egy integrált fejlesztői környezet (IDE), amelyet a Microsoft fejlesztett ki szoftvertervezőknek és programozóknak. Az IDE számos nyelvet támogat, beleértve a C#, a C++, a Visual Basic, az F#, a Python és az TypeScript nyelveket.

A Visual Studio 2022 lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan fejlesszenek különböző alkalmazásokat és programokat, beleértve a webes alkalmazásokat, asztali alkalmazásokat, játékokat és mobilalkalmazásokat is.

Az IDE tartalmaz számos funkciót, amelyek megkönnyítik a fejlesztők munkáját, mint például a kód automatikus kiegészítése, a hibakereső eszközök, a verziókezelés, az adatbázis-kezelő eszközök és az integrált tesztelő eszközök.

A Visual Studio 2022-nek új kiadása van, amely már 64-bites natív alkalmazásként fut, így jobb teljesítményt és gyorsabb betöltési időket biztosít. A felhasználói felület is megújult, többek között az ikonok és a menük is modernizálódtak, valamint a fejlesztői élmény is tovább lett javítva.

Az új kiadás támogatja az új .NET 6-os keretrendszert, valamint több fejlesztői eszközt is tartalmaz, mint például a Live Share, a GitHub integráció és a Blazor WebAssembly. Emellett az új verzióban új és javított funkciók is elérhetőek, például a forráskód szerkesztőben a kódkiegészítési javaslatok és a refaktorálási lehetőségek jobb kezelése.



4.2. Microsoft SQL

Az MSSQL (Microsoft SQL Server) egy relációs adatbázis-kezelő rendszer, amelyet a Microsoft fejlesztett ki. Az MSSQL számos verziója elérhető, beleértve az Express, a Standard, az Enterprise és a Developer kiadásokat is.

Az MSSQL-t általában üzleti alkalmazásokhoz használják, és támogatja az SQL nyelvet és az ACID tranzakciókat. Az MSSQL beépített eszközöket tartalmaz az adatbázis-kezeléshez, az adatok biztonsági mentéséhez, a teljesítményfigyeléshez és a karbantartáshoz.

Az MSSQL-t integrálni lehet más Microsoft technológiákkal, például a .NET keretrendszerrel, az Excellel és az Active Directory-val.

Az MSSQL népszerűsége miatt számos támogatási lehetőség áll rendelkezésre az interneten, például dokumentációk, fórumok és online tanfolyamok.



5. Felhasználói felület (UI)

5.1. Kezdőképernyő

A legkedvesebb rész volt számunkra a kezdőképernyő megtervezése.

Mindannyian kigondoltuk, hogy hogyan is nézhet majd ki az egész webshop arculata. Ennek a megtervezése a Figma alkalmazással készült.

Figma egy felhőalapú kollaboratív tervezői platform, amely lehetővé teszi a felhasználók számára, hogy webes, mobilalkalmazások, vagy bármilyen digitális felület tervezésére használják. Az alkalmazás intuitív, könnyen használható, és számos funkciót kínál a tervezők számára, mint például az elemek közötti áttétel, a prototípus készítése, az együttműködési lehetőségek, valamint az integráció más tervezői eszközökkel és alkalmazásokkal. A Figma alkalmas mind az egyéni tervezőknek, mind a csapatoknak, hogy egyszerűen és hatékonyan dolgozzanak együtt a tervezési projekteken, valamint biztosítja, hogy a designok mindig naprakészek legyenek.

Több verzió is napvilágot látott, nagyjából hetente változtattunk rajta kisebbeket-nagyobbakat.

Ilyen volt például a színek kigondolása, ami az egyik legfontosabb szempont volt, melyet a Coolors weboldalon kerestünk. Itt különböző színekombinációkat fedezhetünk fel és készíthetünk el. Továbbá minden színkódot kimásolhatunk, ezzel könnyen megtaláljuk az arculattervező oldalon.

A színek rengeteg különböző érzelmi és pszichológiai hatással bírnak, és gyakran használják őket a tervezésben, reklámozásban és a marketingben.

A piros például az életerő, szenvedély, szeretet és figyelemfelkeltés szimbóluma lehet. Ezért gyakran használják az éttermekben, hogy serkentsék az étvágyat és növeljék az érzékelhető energiát.

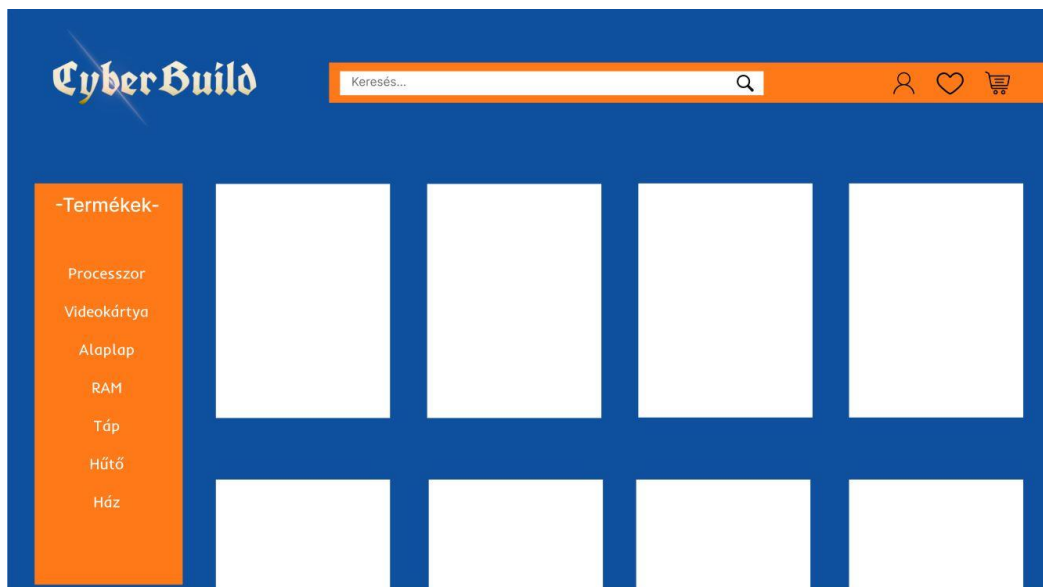
A kék szín a bizalom, stabilitás, nyugalom és megbízhatóság jelképe. Általában az egészségügyi és technológiai ágazatokban használják, hogy bizalmat ébresszenek.

A zöld szín általában a természet, a frissesség és a harmónia jelképe. Ezért gyakran használják az élelmiszeriparban és a környezetbarát cégekben.



Először a fekete és a szürke szín hozott lázba minket, ám ezt hamar dobtuk, mivel túlságosan is sötét volt az egész webáruház témája. Ezt szerettük volna kiküszöbölni. Ezután fedeztük fel a királykék és narancssárga színekombinációt, ami a végleges is lett. Nagyon tetszett mindannyiunknak, sőt még külsősöknek is.

Ennél a verziónál a keresőablak jobb oldalra csúszott, ahol található még a felhasználó fiókja, egy kívánságlista és maga a bevásárlókosár.



Bal oldalt helyet kap egy terméklista, ahol kategóriák szerint böngészhetjük a számítógép alkatrészeket. Minden kategória kap egy rövid szöveget, hogy a vásárlók megtudják, hogy az az adott termék mire képes.

Például: Hűtés → „Akármilyen jól is építette meg számítógépét, nem sokat ér, ha percek alatt túlmelegszik. De ne aggódjon! Nálunk erre is talál megoldást! Legyen szó lég-, vagy akár vízhűtésről, mi mindenre kínálunk megoldást, még hozzá kedvező árakon.”

Tápegység → „Mire jó egy számítógép áram nélkül? Válassza meg bölcsen a tápegységét, remek besorolású tápegységeink garantáltan nem lesznek zárlatosak, és nem fognak csalódást okozni.”

5.1.1. Logó



A webáruház logójának elkészítésekor két szempont volt szem előtt:

- egyrészt ne legyen túl bonyolult az elkészítése,
- másrészt könnyen felismerhető legyen.

A program, amit a megtervezésére használtunk a PicsArt volt.

A PicsArt egy mobilalkalmazás, amely lehetővé teszi a felhasználók számára, hogy kreatív módon szerkeszthessék a képeiket. Az alkalmazás rengeteg szűrőt, efféket, rajzolósi eszközöt és egyéb grafikai elemet kínál, amelyekkel a felhasználók személyre szabhatják és javíthatják a fotóikat. A PicsArt lehetővé teszi a felhasználók számára, hogy közösségi hálózati platformon belül osszák meg és kövessék egymást, továbbá az alkalmazás

magában foglal egy fotószerkesztői eszköztárat, amelyet a felhasználók azonnal használhatnak a képek szerkesztéséhez.

Rendelkeztünk már tapasztalattal, számtalan képet szerkesztettünk már vele, így ezt tűnt a legjobb és legcélszerűbb megoldásnak.

A logón olvasható a CyberBuild felirat (a webáruház neve), amely modernizált, cyberpunk stílusban íródott.

A „b” betűt egy piros lézernyaláb keresztezi, mely a számítógép fényére utal.

Illetve a felirat sötétből világosba, illetve világosból sötétbe megy át, ami szintén a PC-re utal.



5.1.2. Kereső és fejléc

Mint minden webáruházban, elengedhetetlen funkció egy keresőmotor. Ennek használatával kutathatunk a webshop termékei közt, ha tudjuk azoknak pontos neveit.

Amikor rákeresünk a kiszemelt termékre, akkor ezt feldobja az oldal a névvel, árával, képével és leírásával.



Részlet a keresőmotor kódjából:

```
protected void BindData()
{
    string connectionString = "your_connection_string_here";
    string keyword = txtSearch.Text;

    using (SqlConnection connection = new SqlConnection(connectionString))
```

```

    {
        string query = "SELECT * FROM Products WHERE ProductName LIKE '%' +
@keyword + '%";
        SqlCommand command = new SqlCommand(query, connection);
        command.Parameters.AddWithValue("@keyword", keyword);

        SqlDataAdapter adapter = new SqlDataAdapter(command);
        DataSet ds = new DataSet();
        adapter.Fill(ds);

        gvSearchResults.DataSource = ds;
        gvSearchResults.DataBind();
    }
}

```

Amikor a felhasználó megad egy kulcsszót a keresőmezőben, a kód lekérdezi az adatbázisban található "Products" táblából az összes olyan terméket, amelynek neve tartalmazza a kulcsszót.

Az eredményeket egy "GridView" objektumon keresztül jeleníti meg az oldalon.

A kódrészlet az adatbázis-kapcsolatot egy "SqlConnection" objektum segítségével hozza létre, majd az SQL "SELECT" parancsot alkalmazza a kulcsszóval megegyező termékek lekérdezéséhez.

5.2. Főmenü

Ezzel a képpel találkozik először a felhasználó. Fontos, hogy jól átlátható legyen, és lehetőleg térjen el több pontban, mint más hasonló webshopok kinézete.

Az alábbi lehetőségeket építettük be a webáruházunkba:

- a korábban említett kereső és fejléc
- felhasználói fiók

- kosár és kívánságlista
- termékek kategóriákra lebontva
- néhány naprakész hír a főoldalon

5.3. Felhasználó

A felhasználó regisztrációja az egyik legfontosabb funkciója egy webáruháznak.

A regisztráció lehetőséget ad a felhasználóknak, hogy saját fiókot hozzanak létre, és így könnyen hozzáférjenek az oldal különböző tartalmaihoz, mint például az alkatrészek megrendeléséhez.

A regisztráció általában egy egyszerű folyamat, az alap dolgokat kérjük csak ilyenkor, mint a teljes név, email, illetve a jelszó.

The image shows a registration form on a blue background. At the top, the title 'Regisztráció' is displayed in a large, dark font. Below the title, there are four input fields, each preceded by a label in blue text: 'Név' (Name), 'Email', 'Jelszó' (Password), and 'Jelszó ismét' (Repeat Password). The first field contains the text 'Pelda Janos', the second contains 'youremail@gmail.com', the third contains 'Password', and the fourth contains 'Repeat Password'. Below these fields is a button labeled 'Regisztráció' in a black box. At the bottom of the form, there is a link in blue text that reads 'Már van felhasználói Fiókja? Jelentkezzen be itt!'.

A regisztráció során szükség van az adatvédelmi szabályok és a felhasználási feltételek elfogadására is.

Szem előtt tartottuk, hogy a regisztrációs folyamat egyszerű és felhasználóbarát legyen. Az űrlapot egyszerű és érthető módon kellett megterveznünk.

Bejelentkezés

Email

youremail@gmail.com

Jelszó

Bejelentkezés

[Nincs felhasználói Fiókj? Kattintson, és hozzon létre egyet!](#)

Bejelentkezésnél pedig csupán csak az email címünket, illetve a jelszót kell megadnunk.

6. Fizetés és szállítás

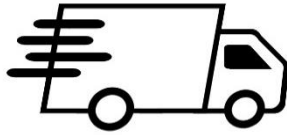
A webshopoknak biztosítaniuk kell azt, hogy a vásárlók könnyen és biztonságosan tudjanak fizetni az adott termékért, termékekért.

Ahhoz, hogy egy webshop rugalmas fizetési lehetőséget tudjon kínálni, többféle módon is lehetséges a sikeres vásárlás:

- utánvét: A termék átvételekor fizeti ki a vásárló a vételárat a futárnak.
- bankkártyás fizetés: A vásárló bankkártyájával fizethet az áruház weboldalán.
- átutalás: A vásárló átutalással fizetheti ki a vételárat a megadott bankszámlaszámra.
- PayPal: A vásárló a saját PayPal számlájával egyenlítheti ki a vételárat.

A szállításnál is nemcsak egyetlen lehetőség áll rendelkezésre, az alábbiakban hármat is felsorolunk:

- házhozszállítás: Az áruház a megrendelt terméket eljuttatja a vásárló lakhelyére.
- A szállítási díj általában függ a termék súlyától és értékétől, valamint a szállítási címtől.
- személyes átvétel: A vásárló általában az áruházban veheti át a megrendelt terméket.
- Ez általában ingyenes, és lehetővé teszi a vásárló számára, hogy ne várjon a szállításra.
- Pick Pack Pont: Az áruházak többsége rendelkezik olyan Pick Pack Pontokkal, amelyek az ország különböző pontjain találhatóak. A vásárló itt veheti át a terméket, ami általában olcsóbb, mint a házhoz szállítás. Ehhez hasonló a Posta Pont is.



A mi webshopunkban a fizetésnél utánvét és bankkártyás fizetés, míg a szállításnál csak házhozszállítás lehetséges.

Ezek a fizetési és szállítási lehetőségek már alapvetően biztosítják a vásárlók számára a kényelmes és biztonságos vásárlási élményt.

Az alábbi kód a fizetési kérés feldolgozására szolgál:

try

```
{
    if (string.IsNullOrEmpty(paymentRequest.CardNumber))
    {
        return BadRequest("Érvénytelen bankkártyaszám");
    }
    if (string.IsNullOrEmpty(paymentRequest.ExpiryMonth) ||
        string.IsNullOrEmpty(paymentRequest.ExpiryYear))
    {
        return BadRequest("Érvénytelen lejárat dátum");
    }
    if (string.IsNullOrEmpty(paymentRequest.Cvv))
    {
        return BadRequest("Érvénytelen CVV kód");
    }
    if (paymentRequest.Amount <= 0)
    {
        return BadRequest("Érvénytelen összeg");
    }
}
```

```

        var paymentProcessor = new PaymentGateway();
        var paymentResult =
paymentProcessor.ProcessPayment(paymentRequest);

        return Ok(paymentResult);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, "Hiba lépett fel fizetés közben!");
        return StatusCode(500, "Hiba lépett fel fizetés közben!");
    }
}
}

```

A kód először ellenőrzi, hogy a megadott bankkártyaszám, lejárat dátum, CVV kód és fizetési összeg érvényes-e. Ha valamelyik érvénytelen, akkor a program BadRequest választ ad vissza a megfelelő hibaüzenettel.

Ha minden adat helyes, akkor egy PaymentGateway objektumot hoz létre a fizetési folyamat végrehajtásához, majd a ProcessPayment metódussal feldolgozza a fizetési kérést és visszaadja az eredményt az Ok válasszal.

Ha bármilyen hiba történik a fizetés közben, a program hibajelzést küld és StatusCode(500) választ ad vissza a "Hiba lépett fel fizetés közben!" üzenettel.

7. Továbbfejlesztési lehetőségek

A webshopok fejlesztése sosem áll meg, mindig van lehetőség az újításra, frissítésekre és az optimalizálásra.

A technológiai változásokkal együtt mindig megjelennek új funkciók és lehetőségek, amelyeket a webshop tulajdonosnak érdemes megfontolnia.

Az egyik legfontosabb tényező az ügyfélközpontság, vagyis a vásárlói élmény javítása. Egyik lehetőség az ajánló rendszer bevezetése, amely a korábbi vásárlások és érdeklődések alapján ajánl nekik termékeket. Ezt akár tovább lehet fejleszteni, hogy az ajánló rendszer ne csak az adott felhasználónak, hanem az általános felhasználói adatok alapján is tudjon ajánlani termékeket.

Az automatikus készletkezelés bevezetése is nagy segítség lehet. Ezzel a webshop tulajdonosának nem kell manuálisan figyelnie a készletet, a rendszer automatikusan értesíti őt, amikor a termék készlete alacsony szintre csökken.

Fontos az is, hogy a webshop biztonságos legyen, ezért a biztonsági intézkedések optimalizálása is fontos feladat. A személyes adatok védelme kulcsfontosságú, ezért az SSL tanúsítványok használata és a kétlépcsős azonosítás bevezetése is javasolt.

Az integráció más alkalmazásokkal is nagy előny lehet, például a logisztikai rendszerrel, amely lehetővé teszi a vásárlók számára, hogy nyomon kövessék a szállítást.

Az integráció az online fizetési rendszerekkel is segíti a vásárlói élményt, így a vásárlók könnyedén és biztonságosan tudják kifizetni a vásárlást.

Az SEO optimalizálás is elengedhetetlen a sikeres webshop üzemeltetéséhez. A megfelelő kulcsszavak kiválasztása és használata, a megfelelő meta leírások és címek használata, valamint az oldal sebességének optimalizálása mind hozzájárulnak a webshop eredményességéhez.

A webshop mobilbaráttá tétele is fontos. Az okostelefonok és táblagépek használata folyamatosan nő, ezért elengedhetetlen, hogy a webshop mobilbarát legyen. A mobilbarát webshop nem csak javítja a vásárlói élményt, hanem az SEO eredményességét is.

A termékismertető fontos szerepet játszanak a vásárlói döntéshozatalban. Ezért érdemes lehet az ilyen leírásokat optimalizálni, és például videókkal vagy képekkel kiegészíteni, hogy a vásárlók könnyebben tudjanak választani a termékek között.

8. Összegzés

Ezen dokumentáció célja a CyberBuild számítástechnikai webshop elkészítésének dokumentálása, az elkészítést övező út bemutatása, nehézségeink ismertetése. A dokumentum bemutatja a felhasznált programokat, az adatbázis létrehozását, a felhasználói felületet, valamint a fizetési és szállítási lehetőségeket.

Jelen dokumentációnk további fejlesztési lehetőségeket is helyez kilátásba a projekthez. Az összegzés alapján a dokumentum alaposan bemutatja a webshop elkészítését, és bőséges információt biztosít arról, hogy milyen lehetőségeket is rejt magában valójában a megfelelő szaktudással párosítva.

Boldogan mondhatjuk azt is, hogy az év során rengeteg tapasztalatot, új ismeretet, és egyéb mást sajátíthattunk el iskolánkban, amelyhez nagyban hozzájárult ez a Szakdolgozat. Új dolgokkal ismerkedhettünk meg, megtanulhattunk csapatban, egységként dolgozni, és megtapasztalhattuk azt is, milyen ha cserben hagyunk minket.

A közös munka rengeteg előnyt jelentett számunkra. Egyesíthettük tudásunkat, kiegészíthettük egymást a bajban, és támogathattuk egymást, ha úgy éreztük már nincs remény.

Emellett említésre méltó, hogy bár a munkába hárman kezdtük bele, az élet más döntést hozott és társunk arra kényszerült, hogy itt hagyjon bennünket. Ez rátette a bélyegét a csapatunk moráljára, de nem adtuk fel, támogattuk egymást a végsőig és megpróbáltuk tudásunkhoz mértén a legtöbbet kihozni a projektünkéből.

Azonban mindennek voltak árnyoldalai is. Természetesen mint sok más az életben, ez sem ment zökkenőmentesen. Az egymástól eltérő tudás, munka tempó és esetenként hozzáállás rengeteg kihívást gördített elénk az utunk során, amiket nagy nehézségek árán, de sikerrel vettünk újra és újra. Nehéz volt összehangolni a munkánkat, néha elég sokat kellett motiválnunk egymást hogy még egy ilyen nehéz helyzetben se adjuk fel, és folytassuk az utunk előre, az év végi vizsga felé.

Összefoglalásképp bátran jelenthetjük ki, hogy mindketten rengeteg pozitív, a jövőre nézve hasznos ismeretet, szaktudást és tapasztalatot szereztünk a szakdolgozaton való közös munka eredményeképp, és a későbbi életben ennek rengeteg aspektusát tudjuk majd hasznosítani.

9. Bibliográfia

<https://www.pcx.hu> (2022.10.24.)

<https://www.konzolvilag.hu/> (2022.10.24.)

<https://ipon.hu/> (2022.10.29.)

https://hu.wikipedia.org/wiki/C_Sharp (2022.11.08.)

<https://nofluffjobs.com/hu/blog/hogyan-kezdjuk-el-react-tanulasat-2021-ben-1-resz/>
(2022.11.20.)

<https://hu.reactjs.org/> (2023.01.08.)

<https://hu.wikipedia.org/wiki/SQL> (2023.01.17.)

https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio (2023.02.11.)

10. Mellékletek

10.1. [A dolgozat mellékletei, ha vannak]