** **

**ZÁRÓDOLGOZAT**

Készítették:

Kovács Dániel

Braczkó Tamás

Novák Dominik Viktor

Konzulens:

Farkas Zoltán

Miskolc

2023.

**Miskolci SZC Kandó Kálmán Informatikai Technikum**

**Miskolci Szakképzési Centrum**

Az 5-0613-12-03 számú Szoftverfejlesztő- és tesztelő szak

**ZÁRÓDOLGOZAT**

**Project Galaxy**

Kovács Dániel – Braczkó Tamás – Novák Domink Viktor

2023 – 2024

**TARTALOM**

[**A PROJEKTRŐL 5**](#_heading=h.gjdgxs)

[**FRONTEND 6**](#_heading=h.17dp8vu)

[HTML: 6](#_heading=h.3rdcrjn)

[CSS: 6](#_heading=h.26in1rg)

[React: 7](#_heading=h.lnxbz9)

[**BACKEND 9**](#_heading=h.1ci93xb)

[Asp.NET Core 6.0 Web API: 9](#_heading=h.3whwml4)

[C#: 9](#_heading=h.jwv5uz9w7rvj)

[Entity Framework Core: 10](#_heading=h.2bn6wsx)

[MailKit: 10](#_heading=h.qsh70q)

[XAMPP: 10](#_heading=h.1ip5iviooiuw)

[MYSQL: 11](#_heading=h.v61gkjsojquz)

[Lighthouse: 11](#_heading=h.21p3wq6k6obz)

[JWT TOKEN: 12](#_heading=h.49x2ik5)

[**ADATBÁZIS SZERKEZET 1**](#_heading=h.3tbugp1)**3**

[**FELHASZNÁLT PROGRAMOK 1**](#_heading=h.23ckvvd)**5**

[Visual Studio 2022: 1](#_heading=h.ihv636)5

[Visual Studio Code: 1](#_heading=h.32hioqz)5

[**CSAPATMUNKA 1**](#_heading=h.vx1227)**6**

[Github: 1](#_heading=h.1v1yuxt)6

[Messenger: 1](#_heading=h.4f1mdlm)6

[Discord: 1](#_heading=h.eb1qf22hs6sq)7

[**TESZTELÉS 1**](#_heading=h.eozwttjzh0ll)**8**

[Frontend tesztelés Lighthouse segítségével: 1](#_heading=h.6h2ushsta2jm)8

[Backend tesztelés Unit tesztekkel: 1](#_heading=h.wphg6cut0l0d)8

[**AZ ALKALMAZÁS 1**](#_heading=h.dt2vr6uwc91v)**9**

[Bejelentkezés: 1](#_heading=h.9woxwwvza3ss)9

[Regisztráció:](#_heading=h.28h4qwu) 20

[Bolygó választás:](#_heading=h.71c3uw7qcoi5) 20

[UI:](#_heading=h.az1acmh6mcc) 21

[Fleet show: 18](#_heading=h.cugbaqm5lspf)

[Ship creation: 18](#_heading=h.bhxq05jr5a76)

[Planet info:](#_heading=h.ambk67gqlgmc) 21

[Battle:](#_heading=h.ejc057dchzfm) 22

[Exploration:](#_heading=h.c69bnvs6x31t) 23

[Messages:](#_heading=h.630cqy16psaz) 24

[**FORRÁS 2**](#_heading=h.nmf14n)**5**

# A PROJEKTRŐL

A csoportunk egy patikát választott projektünk számára melyben egészségmegörző és gyógyszertári termékeket kinálunk. Patikánkban megtalálhatóak például gyógyszerek, vitaminok, táplálékkiiegészítők valamint egészségmegörző termékek.

Patikánkban lehetőség van online rendeléseket leadni, és gyors kiszállítással várjuk a megrendeléseket. Ügyfélszolgálatunk készséggel áll rendelkezésre minden kérdés és probléma esetén, hogy biztosítsuk vásárlóink elégedéttségét és bizalmát.

**FELHASZNÁLT TECHNOLÓGIÁK ÉS PROGRAMOK**

**FRONTEND**

**HTML:**

A HTMLaz angolHyperText Markup Languagerövidítése, ami magyarul hiperszöveges jelölőnyelvet jelent. Weboldalak készítéséhez kifejlesztett nyelv, ami az oldal szövege mellett különböző leíró elemeket tartalmaz, amik blokkokba rendezik a tartalmat, illetve olyan plusz elemekkel egészítik ki, mint például képek, videók vagy űrlapok. Mindent tag-ek közé kell írni pl.: <h1> ami a nyitó tag az egyes szintű fejléc szövegért, majd </h1> záró tag-gel lezárjuk.  
 Amikor megnyitsz egy weboldalt, a böngésződ a webszerverről letölt egy HTML formátumban megírt szöveges fájlt (ez a weboldal tartalma), illetve általában egy CSS fájlt is, ami azt határozza meg, hogy ez a tartalom pontosan hogyan nézzen ki. A böngésző futtatja ezt az állományt, majd kirakja a képernyőre.

**CSS:**

Az angol Cascading Style Sheets rövidítése, melynek szószerinti fordítása: “Egymásba ágyazott stíluslapok”. A CSS egy olyan számítástechnikai nyelv, amivel a weboldalak kinézetét lehet megadni, például hogy milyen színű legyen egy honlap háttere és mekkora betűk jelenjenek meg rajta.  
 A CSS első verziója 1996-ban jelent meg. Megalkotásának elsődleges célja az volt, hogy a weboldalak tartalmát(azaz azt, hogy mi jelenjen meg) **elkülönítsék a külalakjától** (azaz attól, hogy hogyan jelenjen meg). Az olvasott szövegek két helyről előhívva jelenik meg: a HTML tartalmazza magukat a mondatokat, és a CSS adja meg, hogy ezek a mondatok pl: fehér alapon fekete színű betűkkel jelenjenek meg, sőt azt is, hogy ez aláhúzott, *ez pedig dőlt* legyen. Ennek köszönhetően egyrészt sokkal kisebb a weboldalak mérete, így gyorsabban betöltődnek, másrészt lényegesen egyszerűbb megváltoztatni a weboldalak arculatát. Ha például kék színben szeretnénk megjeleníteni ezt a szöveget, csupán csak egy helyen, egyetlen sort kellene átírni a kódban ahelyett, hogy mindenhol kiadnánk a parancsot, hogy a betűk kékek legyenek.

**React:**

React egy nyílt forráskódú JavaScript könyvtár, amelyet felhasználói felületek (UI) készítésére használnak. A Facebook fejlesztette ki, és ma már egy nagy közösség által karbantartott technológia.

React segítségével újra felhasználható komponenseket lehet létrehozni, amelyek önállóan működnek, és összeállíthatók, hogy összetett felhasználói felületeket hozzanak létre. React deklaratív megközelítést alkalmaz az UI készítése során, ami azt jelenti, hogy a fejlesztők leírják, hogy az UI hogyan kell kinéznie és hogyan kell viselkednie, ahelyett, hogy manuálisan manipulálnák a DOM-ot.

React egyik kulcsfontosságú jellemzője a virtuális DOM használata. A virtuális DOM egy könnyűsúlyú reprezentációja a tényleges DOM-nak, amely lehetővé teszi, hogy React hatékonyabban frissítse az UI-t. Amikor egy komponens állapota megváltozik, a React frissíti a virtuális DOM-ot, majd összehasonlítja a korábbi virtuális DOM-mal. Ezután alkalmazza a szükséges változásokat a tényleges DOM-ra, ami eredményez egy hatékonyabb és reaktívabb felhasználói felületet.

React-t széles körben használják a webfejlesztésben, és ez egy népszerű választás az összetett, dinamikus felhasználói felületek készítéséhez. Gyakran használják más könyvtárakkal és keretrendszerekkel, mint például a Redux az alkalmazás állapotának kezeléséhez vagy a Next.js a szerveroldali rendereléshez.

**Mui:**

A Material-UI egy olyan népszerű React keretrendszer, amely a Google Material Design alapján készült. A Material Design esztétikus és intuitív felhasználói felületet kínál, amely könnyen használható és modern megjelenést kölcsönöz a webalkalmazásoknak és weboldalaknak. A Material-UI előre elkészített React komponenseket tartalmaz, amelyek segítségével a fejlesztők könnyen és gyorsan építhetnek felhasználói felületeket.

Ezek a komponensek tartalmazzák az alapvető elemeket, mint például gombok, űrlapok, modális párbeszédpanelek, navigációs elemek és még sok más. A Material-UI lehetővé teszi a fejlesztők számára, hogy ezeket a komponenseket testre szabják és integrálják a saját projektjükbe, így könnyen alkalmazkodnak az egyedi tervezési igényekhez.

Az egyik fő előnye a Material-UI-nek az, hogy reszponzív designra van optimalizálva, ami azt jelenti, hogy a webalkalmazások és weboldalak automatikusan alkalmazkodnak a különböző kijelzőméretekhez és készülékekhez.

A Material-UI kompatibilis a legújabb React verziókkal és könnyen integrálható a React projektekbe. Emellett számos kiegészítő és bővítmény is elérhető hozzá, amelyek további funkcionalitást és stílust kínálnak a fejlesztők számára. A Material-UI-t széles körben használják a webfejlesztésben, és népszerű választás a modern, elegáns felhasználói felületek létrehozásához.

**Axios:**

Az Axios egy nyílt forráskódú JavaScript könyvtár, amelyet a böngésző- és szerveroldali HTTP kérések kezelésére használnak. Az Axios használható a modern böngészőkben és a Node.js szerveroldali környezetben is.

Az Axios lehetővé teszi a fejlesztők számára, hogy egyszerű és hatékony módon kommunikáljanak az API-val. A könyvtár nagyon intuitív interfészét kínál, amely lehetővé teszi a HTTP kérések egyszerű és gyors összeállítását, például GET, POST, PUT és DELETE kérések küldése. Az Axios támogatja a Promise API-t, így könnyen kezelhetők az aszinkron kérések és válaszok.

Az Axios egyéb funkciói közé tartozik a fejlesztőbarát hibakezelés, a kérések széles körű konfigurálhatósága, például a timeout kezelése, és a kérésekhez kapcsolódó adatok küldése, például az átadott adatok formátuma vagy a kérés header-jei.

Az Axios népszerű választás a modern JavaScript alkalmazások fejlesztése során, mivel egy könnyűsúlyú és megbízható megoldást kínál a HTTP kérések kezelésére. Az Axios kompatibilis számos modern JavaScript keretrendszerrel és könyvtárral, például a React, a Vue.js és az Angular.

**BACKEND**

**Asp.NET Core 8.0 Web API:**

Az ASP.NET Core 8.0 Web API egy keretrendszer a .NET fejlesztők számára, amely lehetővé teszi a hatékony és korszerű webes alkalmazások fejlesztését. A Web API-k olyan webalkalmazások, amelyek kommunikálnak más rendszerekkel, például mobilalkalmazásokkal vagy webalkalmazásokkal, adatok lekérdezésére, frissítésére vagy adatok előállítására.

Az ASP.NET Core 8.0 Web API lehetővé teszi a fejlesztők számára, hogy egyszerűen és hatékonyan fejlesszenek HTTP API-kat. A keretrendszer tartalmazza azokat az alapvető komponenseket, amelyek szükségesek a Web API-k fejlesztéséhez, például az HTTP kezelést, a route-olást és a modell validációt.

Az ASP.NET Core 8.0 Web API további előnyei közé tartozik a magas teljesítmény, a skálázhatóság és a biztonság. A keretrendszer lehetővé teszi a fejlesztők számára, hogy különféle hitelesítési és azonosítási módszereket alkalmazzanak, például a JWT-t.

Az ASP.NET Core 8.0 Web API támogatja az OpenAPI specifikációt, amely lehetővé teszi a dokumentáció automatikus generálását és a kliensek számára könnyű API használatát. A keretrendszer széles körben használható a modern webalkalmazások fejlesztése során, és lehetővé teszi a fejlesztők számára, hogy gyorsan és hatékonyan készítsenek skálázható és biztonságos API-kat.

**C#:**

A C# a .NET keretrendszer egyik leghasználtabb objektumorientált programozási nyelve. Korábban a .NET alkalmazások csak Windows alatt voltak elérhetőek, azonban 2016-ban kiadásra került a .NET Core.

A .NET Core megjelenésével a Microsoft kiterjesztette a .NET alkalmazhatóságát MacOS és Linux rendszerekre is.

**Entity Framework Core:**

Az Entity Framework Core egy nyílt forráskódú, cross-platform ORM (Object-Relational Mapping) keretrendszer a .NET fejlesztők számára. Az Entity Framework Core lehetővé teszi a fejlesztők számára, hogy könnyen kezeljék az adatokat és az adatbázisokat a .NET alkalmazásaikban.

Az ORM egy olyan technológia, amely lehetővé teszi a fejlesztők számára, hogy a relációs adatbázisokat objektumorientált kódolással kezeljék. Az Entity Framework Core automatikusan generálja az adatbázis-sémát az alkalmazásban használt osztályok alapján, és lehetővé teszi az adatok lekérdezését, frissítését és törlését az adatbázisból.

Az Entity Framework Core lehetővé teszi a fejlesztők számára, hogy a .NET Core-ra, valamint az Entity Framework 6-ra épülő alkalmazásokat is készítsenek. Az Entity Framework Core támogatja a különféle adatbázis-motorokat, például az SQL Server, az SQLite, a PostgreSQL és az MySQL.

Az Entity Framework Core további előnyei közé tartozik a könnyű telepítés és használat, az egységes adatelérési interfész, a LINQ támogatás, a hatékony adatmódosítások kezelése, a szűk keresési kifejezések lehetősége, az általános hibakezelés és a tesztelhetőség. Az Entity Framework Core egy népszerű választás a .NET fejlesztők között, akik gyorsan és hatékonyan szeretnék kezelni az adatokat az alkalmazásaikban.

**MailKit:**

Levelezést segítő keretrendszer, amelyben megadhatunk egy SMTP szervert, amit használva a felhasználóknak küld levelet.

**XAMPP:**

Mysql adatbázis és az ahhoz tartozó apache szerver elindításában nyújt segítséget. Egyszerű használni és telepíteni. Localhost-on indítja el a szervert alapbeállításon port nélkül. Megkönnyíti a programozáshoz szükséges adatcserét.

**MYSQL:**

MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer (RDBMS), amely lehetővé teszi az adatok hatékony kezelését és tárolását. Az adatok táblákban vannak tárolva, és az adatbázis-kezelő rendszer segítségével lehet azokat lekérdezni, frissíteni, törlési vagy beszúrni.

A MySQL az egyik legnépszerűbb relációs adatbázis-kezelő rendszer a világon, és széles körben használják a különféle alkalmazásokban, például webalkalmazásokban, adatbázisokban, szoftverekben és még sok másban.

A MySQL nagyon skálázható és megbízható adatbázis-kezelő rendszer. A rendszer támogatja a transzakciókat, a referenciális integritást és az ACID (atómikus, konzisztens, izolált, tartós) tulajdonságokat, amelyek biztosítják az adatok integritását és konzisztenciáját.

**JWT TOKEN:**

A JWT (JSON Web Token) egy nyílt szabvány, amely egy kódolt JSON objektumot használ a digitális azonosítás és hitelesítés céljára. A JWT tokeneket gyakran használják azonosításra és hitelesítésre a webes alkalmazásokban és API-kban, különösen az állapotmentes környezetben.

A JWT tokenek három részből állnak: az első rész a fejléc, a második a payload (tartalom), és a harmadik pedig a digitális aláírás. A fejléc a JWT típusát és az alkalmazott titkosítási algoritmust tartalmazza, a payload tartalmazza az adatokat, amelyeket a token hordoz, például a felhasználóazonosítót vagy a szerepkört, és a digitális aláírás pedig az adatok biztonságát és épségét garantálja.

A JWT tokeneknek számos előnye van a hagyományos autentikációs módszerekkel szemben. Az egyik legfontosabb előnyük, hogy a tokenek állapotmentesek, vagyis nincs szükség állapottárolásra a szerver oldalon, és így nagyban csökkenthető a szerveroldali terhelés. Emellett a tokenek nagyfokú biztonságot nyújtanak, mivel a digitális aláírásukkal ellenőrizhető a tartalmuk épsége és eredetisége.

A JWT tokenek széles körben használatosak a modern webes alkalmazásokban és API-kban, és számos programozási nyelvben elérhetőek, például a JavaScript-ben, a Python-ban vagy a Java-ban.

A diagram of a structure

Description automatically generated

*2.ábra - JWT token működése*

**ADATBÁZIS SZERKEZET**

A screenshot of a computer

Description automatically generated

*3.ábra - Az adatbázis szerkezete*

Az **‘aspnetroles’** tábla az ASP.NET Core Identity szerepeket tárolja. Tartalmazza azonosítójukat, nevüket, normalizált nevüket és konkurenciájukat, amelyeke segítenek az azonosításukban és kezelésükben.

Az **‘aspnetusers’** tábla az ASP.NET Core Identity felhasználókat tárolja. Tartalmazza azonosítójukat, felhasználónevüket, email címűket, jeészavuk hash-elt verzióját, biztonsági bélyegüket és egyéb adatokat, amelyek az azonosításhoz és hitelesítéshez szükségesek.

Az **‘aspnetuserroles’** tábla a felasználókhoz rendelt szerepeket tárolja. Kapcsolatot teremt a felhasználók és szerepek között, így lehetővé teszi a felhasználók szerepeinek kezelését és az azokhoz való hozzáférést.

A **‘categories’** tábla a termékekhez kapcsolódó kommenteket tárolja, beleértve a felhasználó azonosítóját és nevét, ami segít a termékek csoportosításában és kategorizálásában.

A **‘comments’** tábla a termékekhez kapcsolódo kommeneket tárolja, beleértve a felhaasználó azonosítóját, a termék azonosítóját, értékelésüket, megjegyzéseiket és a felülvizsgálat dátumát. Lehetővé teszu a felhasználók véleményének és értékelésének nyomon követését

Az ‘**orders’** tábla a rendeléseket tárolja, beleértve a felhasználó azonosítóját, a termék azonosítóját, a rendelt mennyiséget, a teljes árat, az állapotát, a rendelés dátumást és szállítási címet. Segítségével nyomon követhetőek és kezelhetőek a felhasználók által leadott rendelések.

A **‘products’** tábla a termékeket tárolja, beleértve a nevüket, leírásukat, árukat, elérhetőségüket, kategória azonosítójukat, Kép RUL-jüket és raktáron lévő számukat. Segítségével rögzíthetőek és kezelhetőek a rendelkezésre álló termékek adatai, ami fontos információ a vásárlók és az értékesítés számára

**FELHASZNÁLT PROGRAMOK**

**Visual Studio 2022:**

A Visual Studio 2022 egy integrált fejlesztői környezet (IDE), amelyet a Microsoft fejlesztett ki. Az IDE a szoftvertervezési, programozási, tesztelési és hibakeresési folyamatok támogatására szolgál. A Visual Studio 2022 számos programozási nyelvet támogat, beleértve a C#, C++, F#, Python, TypeScript és másokat.

Az új Visual Studio 2022 több új és fejlesztett funkcióval rendelkezik, amelyek célja, hogy segítsék a fejlesztőket hatékonyabban dolgozni, és elősegítsék a fejlesztési folyamat felgyorsítását. A Visual Studio 2022 új tervezőfelületet, frissített debuggert, fejlett kódkezelést, optimalizált kódolási élményt, frissített verziókezelést és még sok más funkciót tartalmaz.

A Visual Studio 2022 több platformon fut, beleértve a Windows, a macOS és az Ubuntu rendszereket is, így lehetővé teszi a fejlesztők számára, hogy bármilyen környezetben dolgozzanak. A Visual Studio 2022-t különféle fejlesztők és fejlesztőcsoportok használják világszerte, például az üzleti alkalmazások, a játékok, a mobilalkalmazások és a webes alkalmazások fejlesztéséhez.

**Visual Studio Code:**

A Visual Studio Code egy ingyenes,nyílt forráskódú kódszerkesztő, melyet aMicrosoft fejleszt [Windows](https://hu.wikipedia.org/wiki/Windows),Linux ésmacOS operációs rendszerekhez. Támogatja a hibakeresőket, valamint beépítettGit támogatással rendelkezik, továbbá képes az intelligens kódkiegészítésre azIntelliSence segítségével. A VSCode-ban a felhasználók megváltoztathatják a kinézetet (témát), megváltoztathatják a szerkesztő gyorsbillentyű-kiosztását, az alapértelmezett beállításokat és még sok egyebet. Támogatja a kiegészítőket, melyek segítségével további funkciók, testreszabási lehetőségek érhetőek el.

**CSAPATMUNKA**

**Github:**

A GitHub egy olyan weboldal és felhőalapú szolgáltatás, amely segít a fejlesztőknek kódjuk tárolásában és kezelésében, valamint a kódjukban bekövetkezett változások nyomon követésében és ellenőrzésében. Egy profitorientált vállalat, amely felhőalapú Git-tárhely tárhely szolgáltatást kínál. Lényegében megkönnyíti az egyének és a csapatok számára a Git használatát verziókezelési és együttműködési célokra.A GitHub felülete elég felhasználóbarát ahhoz, hogy még a kezdő programozók is kihasználhassák a Git előnyeit. A GitHub nélkül a Git használata általában egy kicsit több technikai tudást és a parancssor használatát igényli.  
A GitHub azonban annyira felhasználóbarát, hogy egyesek még más típusú projektek például könyvek írása kezelésére is a GitHubot használják.Ráadásul bárki ingyenesen regisztrálhat és üzemeltethet nyilvános kódtárat, ami a GitHubot különösen népszerűvé teszi a nyílt forráskódú projektek körében.

**Messenger:**

A Messenger egy ingyenes üzenetküldési alkalmazás, melyet az amerikai Facebook fejlesztett ki. Tökéletes üzenetküldésre és fogadásra, ami nagyban megkönnyíti a kapcsolattartást kettő vagy akár több személy között is egyszerre. Lehetőségünk van hangüzenet küldésére, fogadására, valamint hívás, videóhívásban való kommunikációra is. Küldhetünk képeket, videókat, szöveges dokumentumokat, de akár még különböző linkeket is megoszthatunk egymással. Különböző hangulatjelekkel is kifejezhetjük véleményünket, érzéseinket egymás felé. Megalkotója Mark Zuckerberg, aki a Facebook-ot 2004. Februárjában indította útjára, míg a Messenger legelőször 2008-ban kezdte meg működését. Azóta az alkalmazás nem csak Android és Ios eszközökre érhető el, hanem számítógépekre is letölthető, használható.

**Discord:**

A Discord egy első körben ingyenes internetes alkalmazás és digitális terjesztési platform, amit leginkább videójátékot játszó közösségek számára fejlesztettek ki, de bárki más is használhatja. Úgy tervezték, hogy nagy rendszerigényű programok, első körben játékok futtatásánál is problémák nélkül működjön. A platform rendelkezik szöveges, kép és videó, valamint audio kommunikációval is. W[indows](https://hu.wikipedia.org/wiki/Windows), [MacOS](https://hu.wikipedia.org/wiki/MacOS), [Android](https://hu.wikipedia.org/wiki/Android_(oper%C3%A1ci%C3%B3s_rendszer)),[iOS](https://hu.wikipedia.org/wiki/IOS), Linux operációs rendszereken és böngészőkön fut. Alapötlete Jason Citrontól származik. Magán és az állami közösségek létrehozására és kezelésére készült. Ez a felhasználók számára hozzáférést biztosít a kommunikációs szolgáltatások köré összpontosító eszközökhöz, például a hang és videohívásokhoz, az állandó csevegőszobákhoz, a többi játékos központú szolgáltatással való integrációhoz, a közvetlen üzenetek küldésének és a személyes csoportok létrehozásának általános képességéhez. Lehetőségünk van Nitro vásárlására is, amelyek segítenek áthidalni például az egyszerre való üzenetküldési limiteket, valamint segít nagyobb felbontású képek megosztásában is.

**AZ ALKALMAZÁS**

**Bejelentkezés:**

A SignIn komponens egy olyan felületet biztosít, ahol a felhasználók bejelentkezhetnek a rendszerbe. A felület tartalmazza az e-mail és jelszó mezőket, amelyeket a felhasználónak kitöltve lehet belépni a rendszerbe.A bejelentkezési adatok helyes megadása esetén a felhasználó belép az alkalmazásba, és tovább navigál a fő oldalra. Ha a bejelentkezési adatok hibásak, akkor a felhasználó egy hibaüzenetet kap a képernyőn.A screenshot of a login screen

Description automatically generated

**Regisztráció:**

A Regisztrációs oldal segítségével a felhasználók új fiókot hozhatnak létre az alkalmazásban. Az oldalon található űrlap lehtővé teszi a felhasználó számára, hogy megadjanak néhány személyes adatot, mint példáúl vezeték- és keresztnevet, e-mail címet és jelszót. Az összes mező kitöltése kötelező, és csak akkor lehet regisztrálni ha a mezők megfelelően vannak kitőltve. Az e-mail címet validálni fogjuk, így csak érvényes formátum esetén fogadka el az űrlap. Ha a megadott jelszavak nem egyeznek meg, a felhasznát figyelmeztető üzenet fogja tákékoztatni a problémáról.

Az ürlap elküldésekor az adatokat továbbítjuk a szervernek, ahol a regisztráció folyamata lebonyolításra kerül, Ha a regisztráció sikeres volt, a felhasználót átirányítjuk a főoldalra. Ha bármilyen hivaba történik a regisztráció során, a feéhasználó az ürlap tetején értesítést kap a hibáról, és llehetősége van újra elküldeni az űrlapot a megfelelő adatokkal.

**Termék:**

A főoldalon található termékek egy csak azt az adott termékre muta oldalra visznek. A termék egyedi oldalát jeleníti meg, amely tartalmazza a termék részleteit. A felhasználók számára lehetőség van a termék mennyiségének kiválasztására és a kosárba helyezésébe. Az oldal alján megjelenik egy csillagokkal jelölt rendszer, amely a termék átlagos értékelését jeleníti meg. A felhasználók által megadott értékelések alapján az átlag és teljes értékelés száma automatikus frissul, Allata a termékekhez kapcsolódó kommenteket Listázza ki a CommentList komponens. Amiben a kommentek modosítása és törlése és megtehető, illetve egy külön komponens ami a CommenListben van meghíva lehetőséget biztosít a komment megírássára és a termék értékeléséhez.

**UI:**

Az alkalmazás egy egyszerű és letisztul felhasználói velülettel rendelkezik, amelly reszponzív böngészőben. A dizájn célja a könnyű kezlhetőség és az olvashatódág biztosítása, valamint a barátságos környezet nyújtása a felhasználók számára. A kategóriák, egyhelyre van öszpontosítva, hogy egyszerűen lehessen szűrni a kivánt termékek típusára, a kereső azonnal működik, valós időben, nem szükséges a pontos nevét tudni a terméknek. Felül van az a felhasználó fiókjának a sáva, ami ha a felhasználó belépet akkor kijelentkezni vagy a profilját tudja megtekinteni, a kosár ikon a kosárra visz amig a kérdőjel a szuppothoz.

**Kosár:**

Ez az alkalmazás azon része ami a felhasználó által kiválasztott termékek megjelenítését és kezelését szolgálja. Az oldal betöltésekor lekérdezi a backend szerverről a kosár tartalmát, majd a felhasználó által eltávolított termékeket is kezeli.

A felhasználó a kosárban található termékeket egyszerűen eltávolíthatja a „Remove” gomb segítségével, ami az adott terméket kiveszi a kosárból, és frissiti az összesített árat. A fizetés gombra kattintva a felhasználó át lesz irányítva a fizetési folyamathoz, a hol véglegesíti a vásárlást.

Összeségében ez a komponens egy felhasználóbarát és könnyen kezelhető felületet biztosít a vásárlóknak, ahol egyszerűen kezelhetik a kosarukat és lebonyolíthatják a fizetést.

## . **Payment:**

Ez a komponens felelős a fizetési folyamat lebonyolításáért a vásárlók számára. A felhasználóknak lehetőségük van megadni a szállítási címet, várost, irányítószámot és telefonszámot, majd a „Pay Now” gombra kattintva megkezdődik a fizetési folyamat. A komponens először is ellenőrzi, hogy a felhasználó be van-e jelentkezve. Ha nincs, akkor megjelenik egy modal ablak, amely felhívja a felhasználó figyelmét arra, hogy be kell jelentkeznie a fizetési folytatáshoz.

Amennyiben a felhasználó be van jelenzkezve, a komponens elkészíti az e-mail tartalmát, amely a fizetés visszaigazolását tartalmazza. Ezt követően a komponens elküldi az e-mail a felhasználó által megadott e-mail címre.

A fizetési folyamat befejezése után, ha minden sikeresen lezajlott. Akkor az email megjelenik a felhasználó email címén.

# FORRÁS

1. Téma: HTML, webcím:<https://hwellkft.hu/marketing-szotar/html>
2. Téma: JWT token, webcím: <https://supertokens.com/blog/what-is-jwt>
3. “2. ábra”: JWT token, webcím: <https://supertokens.com/blog/what-is-jwt> letöltés dátuma: 2023.05.04
4. Téma: Entity Framework, webcím: <https://learn.microsoft.com/en-us/ef/>
5. Téma: ReactJS, webcím: <https://blog.hubspot.com/website/react-js>
6. Téma: MySQL, webcím: <https://www.oracle.com/mysql/what-is-mysql/>