

**Készítette:** Albert Benedek

Bányai Krisztián

**Tantárgy:** Szoftverrendszerek tervezése

**Tanévszerkezet:** 2023-2024

**Szak :** Automatika és Alkalmazott Informatika

# Tartalom

[Tartalom 2](#_Toc155490915)

[1.Bevezető 2](#_Toc155490916)

[2.Projekt célja 2](#_Toc155490917)

[3.Architektúra diagram 3](#_Toc155490918)

[4.Követelmények 4](#_Toc155490919)

[5.Az adatbázis 8](#_Toc155490920)

[6.Wireframe 9](#_Toc155490921)

[7.Feladat kezelés. Kanban tábla. 10](#_Toc155490922)

[8.Útmutató az alkalmazáshoz 11](#_Toc155490923)

[9.Felhasznált felületek/programozási nyelvek 11](#_Toc155490924)

[10.Következtetések: 12](#_Toc155490925)

[11.Tovább fejlesztési lehetőségek: 12](#_Toc155490926)

[12. Bibliográfia 13](#_Toc155490927)

# 1.Bevezető

Egy weboldal elérése szinte bárhol lehetséges a világon, ami lehetőséget teremt arra, hogy bárki, bármikor hozzáférhessen tartalmakhoz vagy szolgáltatásokhoz.

A weboldal a jelenlegi formájára való fejlődés során sok időt töltöttünk az utánaolvasással és új készségek elsajátításával. A jövőben még tovább fejleszthető lesz új funkciókkal, de mindeközben rendkívül tanulságos volt felfedezni és kipróbálni, hogy egy weboldal programszerkezete hogyan működik.

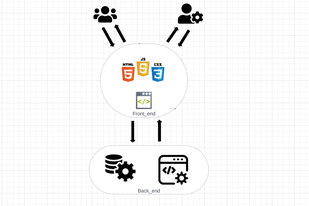
A projekt kezdetén egyikünk sem rendelkezett ismeretekkel Javascript programozásban.  
És még kevésbé tudtuk, hogyan kell létrehozni és összekapcsolni egy adatbázist egy Javascript projektel. Végül azonban sok mindent elsajátítottunk ezekből a területekről.

# 2.Projekt célja

Egy weboldal , amely segítségével az emberek követni tudják a csapat legfrissebb híreit illetve a játékosokkal kapcsolatos újdonságokat. A platform lehetővé tesz egy online áruházat , amely segítségével a szurkolók hozzáférhetnek a csapat által kínált különböző termékekhez.

Vendégként mindenki számára nyitva áll a weboldal, ahol könnyen böngészhetnek és tájékozódhatnak a csapat legfrissebb híreiről . A regisztráció illetve a bejelentkezés csak az online áruház használatához szükséges . Ez azt jelenti, hogy a felhasználók szabadon böngészhetnek a weboldalon, megtekinthetik a csapat legfrissebb tartalmait, de amikor vásárolni szeretnének, regisztrálniuk kell.

# 3.Architektúra diagram



*1. ábra – Architektúra diagram*

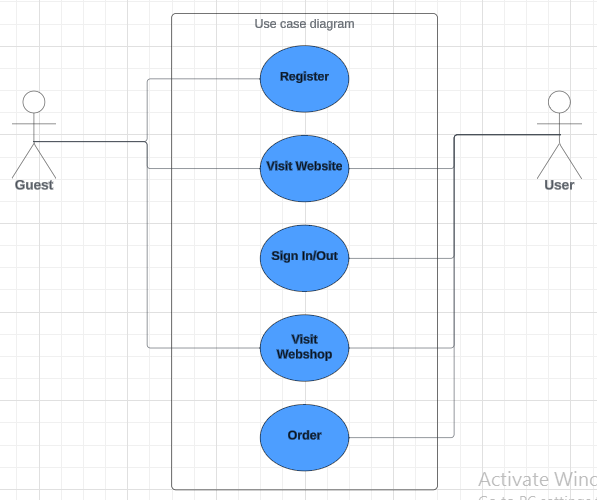
Az architektúra diagram bemutatja a felhasznált komponenseket.   
Nagyon fontos, hogy alaposan kidolgozzuk a szoftverünk architektúráját annak érdekében, hogy a tervezett funkcionalitást minél hatékonyabban megvalósíthassuk.

A webalkalmazást 3 fő alkotóelemre lehet osztani:  
1. Felhasználó  
2. Böngésző  
3.Adatbázis

# 4.Követelmények

1. 1.Felhasználói követelmények . Use Case Diagram.

Célja leírni a modellezendő rendszer és a környezete kapcsolatát.  
A rendszer olyan funkciója vagy funkció halmaza, amely a felhasználó (aktor) céljának elvégzéséhez járul hozzá.

 *2.ábra-Use case diagram*

Amint látjuk két használati esetet tudunk megkülönböztetni , éspedig vendég felhasználót illetve regisztrált felhasználót.  
A vendég felhasználó látogathatja a weboldalt bejelentkezés nélkül is , viszont a regisztrált felhasználó a weboldal keretén belül elérheti az online áruházat is .

1. 2.Rendszer követelmények

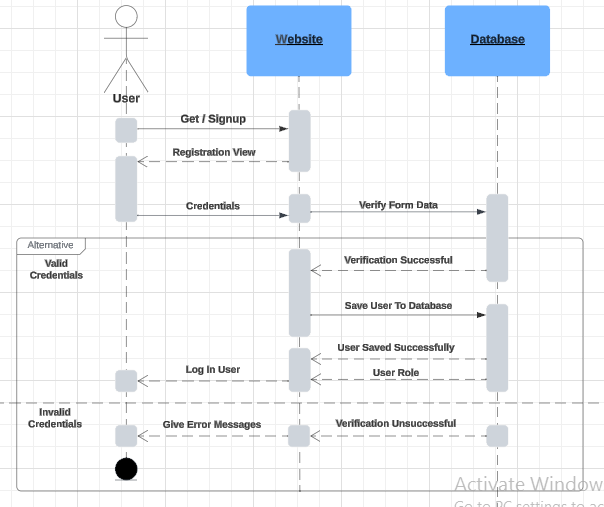
4.2.1.Funkcionális követelmények:

Az weboldal indításakor a kezdő oldal jelenik meg , ahova automatikusan vendégként lépünk be.   
Amint az online aruházat szeretnénk elérni szükségünk lesz regisztrációra illetve bejelentkezésre.

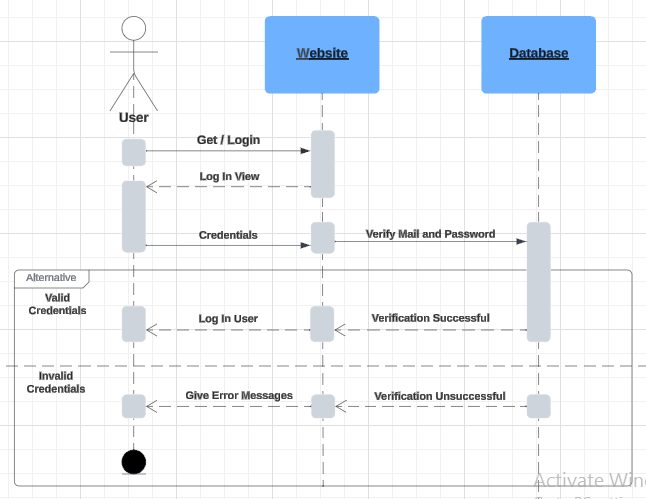
4.2.2.Nem funkcionális követelmények:

* + Számítógép
  + Okostelefon
  + Internetkapcsolat

1. 3.Fontosabb műveletek magyarázata:

Felhasználó regisztrációja  
  
 *3.ábra – Regisztráció(Szekvencia diagram)*

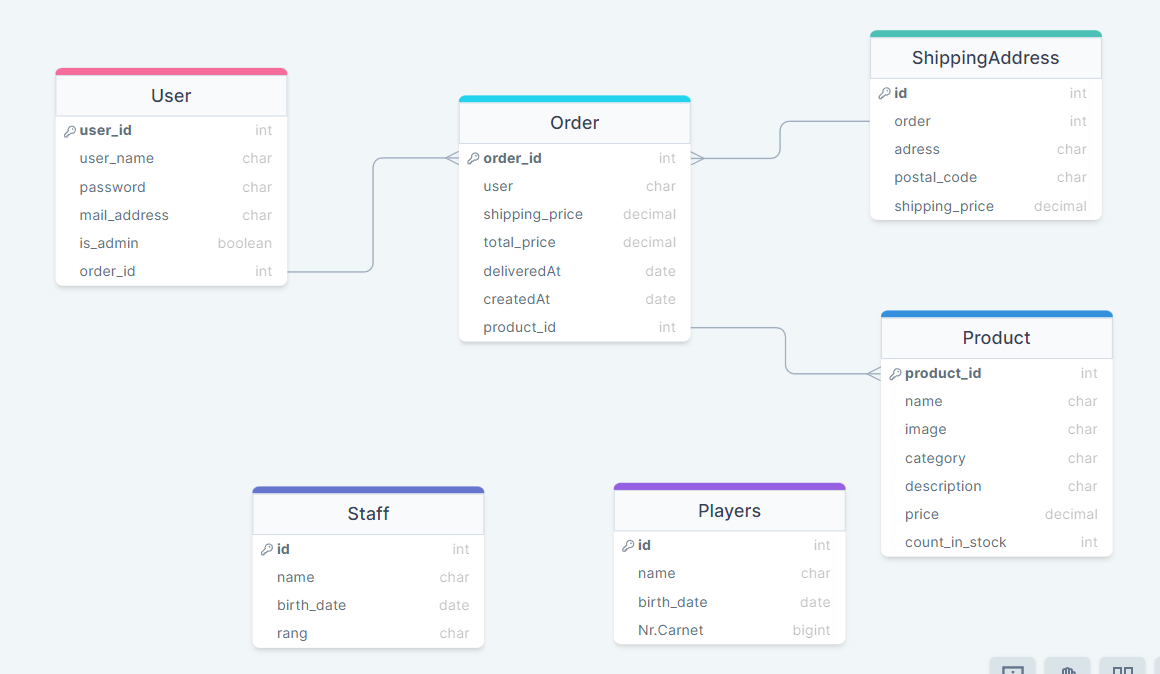
Az előbb említett ábrán látható a felhasználó regisztrációjának folyamata.  
A szükséges adatok megadása után ellenőrzésre kerül a mail cím helyessége , amennyiben ez helytelen , a felhasználó adatai nem kerülnek eltárolásra , és hibaüzenetek formájában tájékoztatjuk a felhasználót.

Felhasználó bejelentkezése  
  
  
 *4.ábra - Bejelentkezés(Szekvencia diagram)*

Az előbb említett ábrán látható a felhasználó regisztrációjának folyamata.  
Hasonlóan történik, mint a felhasználó regisztrációja. Ha helyesek a megadott mail cím és a jelszó , akkor lekérdezzük az adatbázisból a már létező felhasználót.

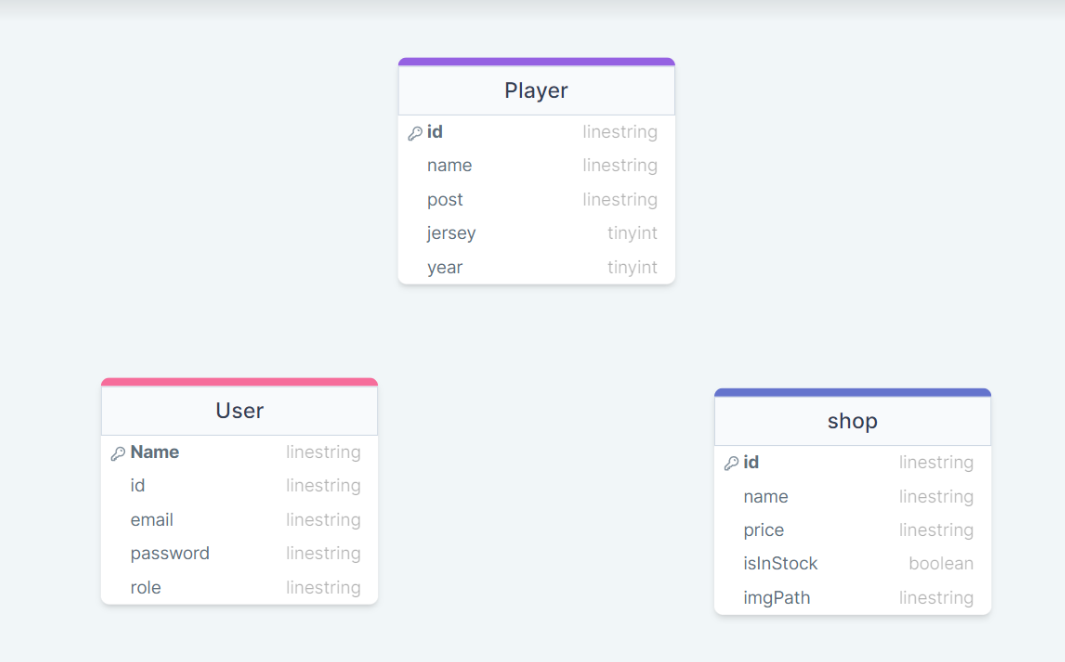
# 5.Az adatbázis

5.1.A rendszer kezdetleges adatbázisa



*5. ábra – Kezdetleges adatbázis terv*

5.2.A rendszer Végleges adatbázisa

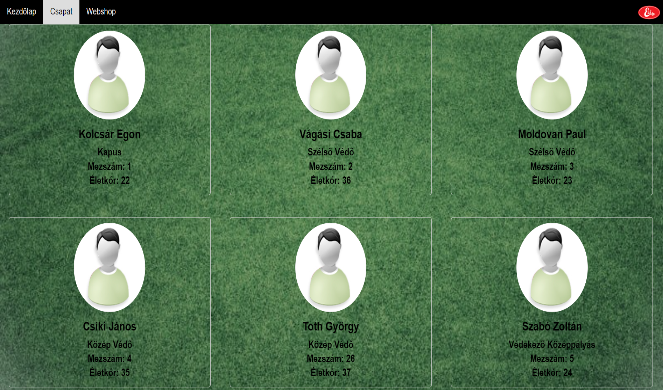
****

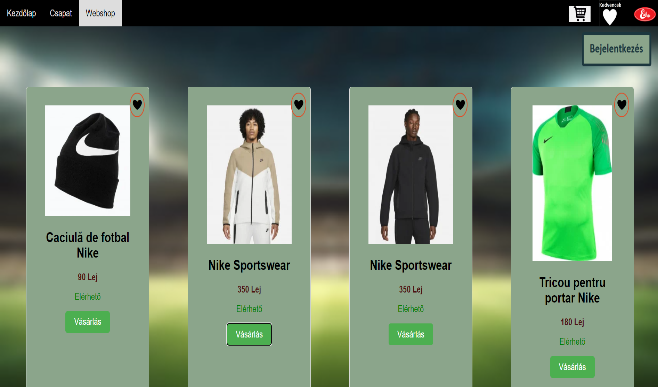
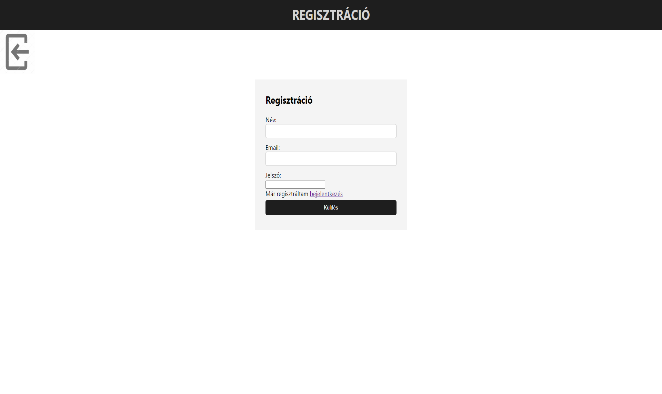
*6.ábra – Végső Adatbázis terv*

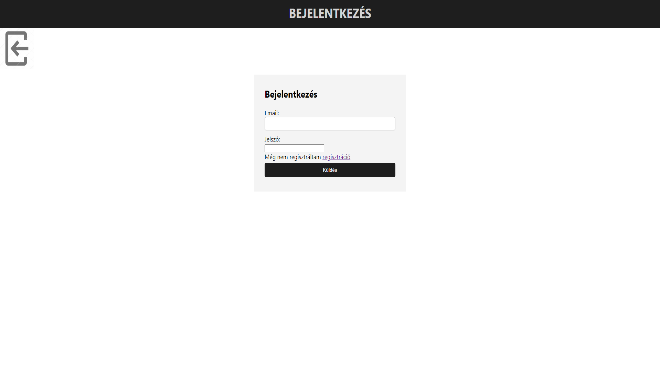
Kezdetben Firebase típusú adatbázisban kezdtünk , de végül találtunk egy könnyebben kezelhetőt , így aztán a MongoDB adatbázisra esett a választásunk.  
Végül az adatbázis terv is egyszerűsödött , így jött létre a 6.ábrán terv.

Az adatbázis 3 darab táblából áll:  
 - Player : A csapat játékosai  
 -User : A beregisztrált felhasználók  
 -Shop : Az online árúház termékei

# 6.Wireframe







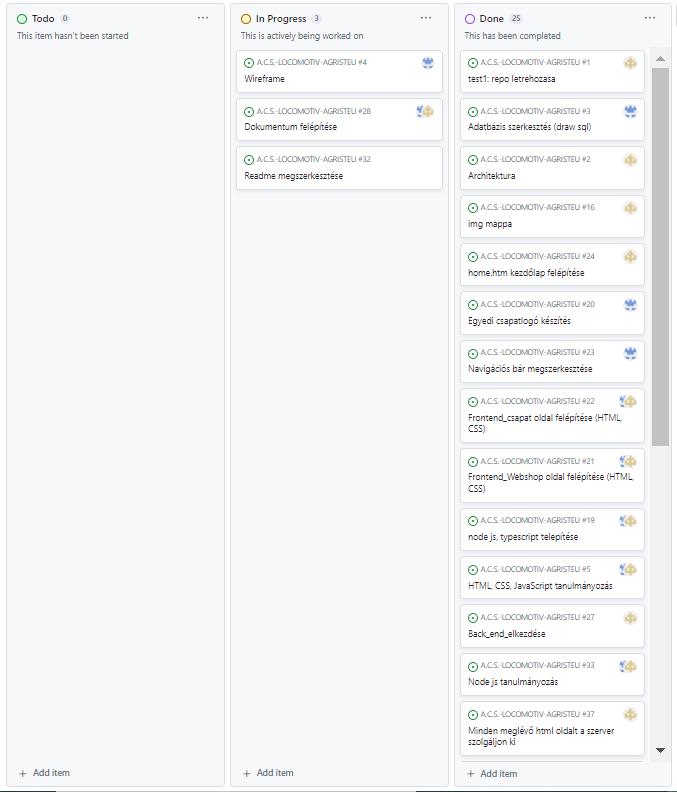
*7.ábra - Wireframe*

# 7.Feladat kezelés. Kanban tábla.

Egy ekkora projektben elengedhetetlen a feladatok pontos elosztása.  
Egy eszköz , amely segíthet ennek a megvalósításában az a Kanban tábla.

A Kanban tábla nem más mint egy munkafolyamat-vizualizációs eszköz, amelyet arra terveztek, hogy segísen egyértelművé tenni a munkafolyamatot , és növelni a hatékonyságot a folyamatban lévő munka korlátozásával.

Ebben a projektben a Kanban tábla GitHub segítségével van megvalósítva.

**

*8.ábra – Kanban tábla*

A fenti ábrán láthatjuk a feladatok menedzselését.  
Láthatjuk azon feladatokat amelyek már ellettek végezve , amelyek folyamatban vannak és azokat , amelyek még nincsenek elkezdve.

# 8.Útmutató az alkalmazáshoz

A weboldalra látogató a kezdőoldalra érkezik , onnan a navigációs bárról tud eljutni a csapat és a webshop oldalakra.

A webshop oldalon bejelentkezetlen felhasználó , a kedvenc listához adás, illetve a vásárlás gombra kattintva kap egy felhívást a bejelentkezésre , amit bejelentkezés gombra kattintva megtehet. A bejelentkezés oldalon tud átlépni a regisztrációs oldalra , ha még nincs fiókja. A formok vizsgállják, hogy helyes formátumú email cím legyen illetve, hogy legyenek kitöltve a mezők.  
 Ha bejelentkezett visszadobja a webshopra. Itt már egy tokennel van megkülönböztetve, ezért ha a kedvenc listára kattint akkor kap egy üzenetet, hogy hozzáadva, ami még nincs megvalósítva. Vásárlás gombra kattintva egy új oldalra tölt be ahól egy formot kitöltve elküldheti a kiszállításhoz szükséges adatait.

Lehetőség van admin módba is bejelentkezni , az adminhoz szükséges bejelentkezési adatokkal. Az admin oldal kirendereli az adatbázisban beregisztrált felhasználókat és ezeket tudja szerkeszteni, törölni, vagy új felhasználót beregisztrálni egy form kitöltésével. Az adatbázisban viszont tényleges változás nem történik, ez még tovább kell fejleszteni.

# 9.Felhasznált felületek/programozási nyelvek

* html
* css
* JavaScript
* node.js (express package, ejs package, mongodb package, jsonwebtoken package, body-parser package, cookie-parser)
* MongoDB Atlas adatbázis (online)

Html, css, javaScript

Alap webfejlesztési nyelvek. A html leírja a weboldal struktúráját és tartalmát, tageket használ. A css felelős a weboldal megjelenéséért, stílusáért. A segítségével formázható a html. A html és css nem olyan módon működnek, mint a hagyományos programozási nyelvek, nem terjed ki az algoritmusok leírására. A JavaScript már programozási nyelvnek számít, növeli a weboldalak interaktivitását, dinamikussá teszi azt, kliensoldali szkriptnyelv, de használlják back-end fejlesztésre is.

Node.js egy szerveroldali JavaScript futtatási környezet, amely lehetővé teszi a JavaScript használatát a szerveroldali fejlesztés során. Node.js egy eseményvezérelt architektúrát használ, és képes kezelni nagy mennyiségű konkurens kapcsolatot anélkül, hogy blokkolná a kódot. Az aszinkron I/O (Input/Output) lehetővé teszi, hogy a kód hatékonyan kezeljen több feladatot egyszerre.   
 A MongoDB egy NoSQL adatbázis. Jellemzője a rugalmasság, nincs szigorú sémadefiníció, így a dokumentumok ugyanabba a gyűjteménybe különböző mezőkkel rendelkezhetnek. Ez lehetővé teszi az alkalmazások fejlesztői számára, hogy dinamikusan változtassák az adatstruktúrát.   
Támogatja az indexeket, amelyek jelentősen javíthatják a lekérdezések teljesítményét.   
Az indexek lehetővé teszik a gyors és hatékony adatkeresést.

# 10.Következtetések:

A projekt megvalósítása során nagyon sok kihívással kellett szembenéznünk, de a célkitűzéseket többnyire sikerült teljesítenünk.   
Az elsődleges szándékunk egy SQL típusú adatbázis használata volt, mivel ezen típussal ismerkedtünk meg az adatbázisok tantárgy keretében ebben a félévben. Azonban ezt nem találtuk annyira optimálisnak, így döntöttünk úgy, hogy áttérünk egy NoSQL típusú adatbázisra, konkrétan a MongoDB-re.

# 11.Tovább fejlesztési lehetőségek:

A weboldalunkhoz szükséges lenne néhány fejlesztés, bővítés. Nem sikerült kitelepitsük , valós http domaint létrehozni.   
Kliens módban a sikeresen bejelentkezett felhasználónak kedvenc listához adásánál valóban meg kellene jelenjenek a termékek.  
 Vásárlásnál először jelenitse meg az annak készített oldalon a terméket, illetve lehessen több darabot is hozzáadni. Lehessen jelszót cserélni, az admin által megszerkesztett felhasználók valóban legyenek módosítva adatbázisban is. Admin módban lehessen termékeket, játékosokat is modosítani. Biztonsági feltételek pl: A tokent a localStorage-ban tároljuk ami nem túl biztonságos.

<% if (token) { %>

localStorage.setItem('token', '<%= token %>');

<% } %>

(webshop.ejs 34. sor)

# 12. Bibliográfia [1] „Lucidchart,” https://lucid.app. [2] https://kodbazis.hu/ [3] <https://drive.google.com/drive/folders/1jKxD_Pv9ElCwxyhkszH0ugz03HJjWO-k> [4] https://drawsql.app/ [5] <https://www.softwaretestinghelp.com/github-desktop-tutorial/>  [6] „Github,” https://github.com/.