



ZÁRÓDOLGOZAT

Készítették:

Tóth Viktor – Kiss Gergő – Kutasi Erzsébet

Konzulens:

Kerényi Róbert Nándor

Miskolc

2024.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

Szoftverfejlesztő- és tesztelő szak

ZÁRÓDOLGOZAT

Syntax Quest

Játék

Tóth Viktor – Kiss Gergő – Kutasi Erzsébet

2023-2024

Tartalom

Tartalom	3
Témaválasztás indoklása	5
Felhasznált technológiák, programozási nyelvek.....	6
Frontend	6
HTML	6
CSS	6
BOOTSTRAP	6
Axios	6
React.....	7
React Router	7
JWT	7
Backend	8
C#.....	8
Asp.Net Core API	8
Entity Framework Core.....	8
MySQL.Data.....	9
Adatbázis	9
MySQL.....	9
Programozási környezet.....	10
Visual Studio Code.....	10
Visual Studio 2022	10
XAMPP	10
Kommunikációs felületek	11
Discord.....	11
Github.....	11
Trello.....	11
A webes alkalmazás szerkezete.....	12
Adatbázis szerkezete	12
Backend szerkezete	17
Frontend szerkezete	18
Az alkalmazás bemutatása	19
Tesztelés	23
Lighthouse	23
Selenium test.....	24

Unit test.....	25
Mock.....	26
Forrás:.....	27

Témaválasztás indoklása

A záródolgozat témáját a csapatunk közösen választotta, ami egy böngészős rogue-like játék, amelyben a játékosnak különböző pályákon kell túlélnie, miközben folyamatosan fejlődik a karaktere. A játékmenet során a karakter különböző ellenségekkel találkozik, amelyek legyőzésével tapasztalatot, tárgyakat szerez. A karakterek fejlődése a tapasztalatspontok és a gyűjtött tárgyak segítségével történik, amelyek lehetővé teszik új képességek és felszerelések megszerzését. A játékosnak taktikai döntéseket kell hoznia, hogy mikor támadjon vagy meneküljön, és hogy milyen erőforrásokat gyűjtsön össze a túlélés érdekében. A cél az, hogy a játékos a legtöbb pontot összegyűjtse.



1. ábra: Syntax Quest játék

Felhasznált technológiák, programozási nyelvek

Frontend

HTML

A HTML a Hypertext Markup Language rövidítése. Ez egy kódoló nyelv, amelyet weboldalak tartalmának létrehozására és strukturálására használnak. Címkék, attribútumok segítségével határoz meg különböző elemeket, például címsorokat, bekezdéseket, képeket és hivatkozásokat. Lényegében a HTML a weboldal gerince, amely lehetővé teszi az interneten megtekinthető tartalom létrehozását.

CSS

A CSS az angol „cascading style sheets” kifejezés rövidítése, ami magyarul „egymásba ágyazott stíluslapokat” jelent. A hangsúly a „stíluson” van, míg a HTML a weblap szerkezetét határozza meg (főcímek, bekezdések) és lehetővé teszi, hogy különböző elemeket (képek, videók) ágyazzunk webes dokumentumba, addig a CSS a weblap vizuális stílusáért felel, az oldal elrendezéséért, a színekért, a betűkészletekért, azok méretéért.

BOOTSTRAP

A Botosra egy HTML, CSS és JS könyvtár, amely az informatív weboldalak fejlesztésének leegyszerűsítésére összpontosít. A webes projektekhez való hozzáadásának elsődleges célja az, hogy a Bootstrap által választott színt, méretet, betűtípust és elrendezést alkalmazza az adott projektre. A Bootstrap alapvető stílusdefiníciókat biztosít az összes HTML-elemhez. Az eredmény a próza, a táblázatok és az űrlapelemek egységes megjelenése a böngészőkben.

Axios

Az Axios, amely egy népszerű könyvtár, amely a háttérrel való kommunikációra szolgál, főként aszinkron HTTP-kérések REST-végpontokhoz való küldésére szolgál. Ez a könyvtár nagyon hasznos CRUD műveletek végrehajtásához. Az Axios használatával API kéréseket készítünk az alkalmazásunkban. Miután a kérés megtörtént, megkapjuk az adatokat a Return-ban, majd ezeket az adatokat felhasználjuk projektünkben.

React

A React egy deklaratív, effektív, és rugalmas JavaScript könyvtár, felhasználói felületek készítéséhez. Lehetővé teszi komplex felhasználói felületek összeállítását izolált kódrészletekből, amiket “komponenseknek” hívunk. A ReactJS segítségével épített alkalmazások az egyszeri újrafelhasználhatóság elvét alkalmazzák. Ez arra az elvre épül, hogy a weboldalak és alkalmazások komponensekből, egyirányú folyamattal épüljenek fel. A Reactben létezik az állapotok és a megváltoztathatatlanság fogalma. A szülő és a gyermek komponensek tekintetében a komponensek szigorú hierarchiával rendelkeznek. A React esetében egy komponens egy meglévő kódrészlet, amely megírása a tiszta kód (Clean Code) elvén alapult.

React Router

A React Router egy különálló könyvtár/csomag, amit React-hoz készítettek. Komponens alapú navigációt biztosít. Ez annyit jelent, hogy az URL címünk mindig szinkronban van az alkalmazással és a megfelelő komponenst jeleníti meg. Lehet úgy is használni, hogy egy útvonal egyetlen egy komponenst jelenít meg vagy bizonyos részei az oldalunknak (Navbar, Footer stb.) statikusak maradnak az útvonaltól függetlenül, és így mindig csak a tartalmi rész fog render-elődni.

JWT

A hitelesítés után az alkalmazás session-token-t vagy JSON-token-t (JWT) használhat a felhasználó azonosítására, és hozzáférést biztosít a védett végpontokhoz. Ezeket a tokeneket általában minden kéréssel együtt elküldik a kiszolgálónak, és a kiszolgáló ellenőrzi őket, mielőtt engedélyezné a hozzáférést a kért erőforráshoz.

Backend

C#

A C# (más néven "See Sharp") egy modern, objektumorientált és típusbiztos programozási nyelv. A C# lehetővé teszi a fejlesztők számára, hogy sokféle biztonságos és robusztus alkalmazást készítsenek, amelyek a .NET-ben futnak. A C# a C nyelvcsaládban gyökerezik, és azonnal megismerkedik a C, a C++, a Java és a JavaScript programozókkal. A C# a kezdetek óta olyan funkciókkal bővült, amelyek új számítási feladatokat és új szoftvertervezési eljárásokat támogatnak.

Asp.Net Core API

A webes API egy alkalmazásprogramozási felület egy webalkalmazás vagy webszerver számára. HTTP-protokollt használ az ügyfelek és a webhelyek közötti kommunikációhoz az adatokhoz való hozzáférés érdekében. A felhasználó különböző eszközökről szeretné elérni az alkalmazást, például mobilról, böngészőről, Google-eszközökről stb. Ebben az esetben a Web API hasznos lehet. Különböző eszközök kérnek Web API-t, és a Web API JSON formátumban válaszol. A legtöbb eszköz képes megérteni a JSON-kimenetet.

Entity Framework Core

Az Entity Framework Core egy adatkezelő keretrendszer, amely lehetővé teszi az alkalmazások számára az adatok adatbázisból történő lekérdezését és manipulációját. Segítségével könnyedén lekérdezhetjük, beszúrhatjuk, frissíthetjük vagy törölhetjük az adatokat az adatbázisban anélkül, hogy közvetlenül SQL lekérdezéseket kellene írunk. Az Entity Framework Core alapvetően objektum-orientált módon kezeli az adatbázist, így az alkalmazásunkban az adatokat objektumokként kezelhetjük. Támogatja az adatbázis-séma kezdeti létrehozását, migrációit, valamint az adatbázis és az alkalmazás közötti adatok szinkronizálását. Rugalmas és könnyen testre szabható, így alkalmazkodni tud az alkalmazásunk igényeihez és az adatbázis változásaihoz.

MySQL.Data

A MySQL.Data package egy programkönyvtár, amely lehetővé teszi az alkalmazások számára, hogy kommunikáljanak egy MySQL adatbázissal. Segítségével könnyedén hozhatunk létre kapcsolatot egy MySQL adatbázissal, és végrehajthatunk rajta különböző műveleteket, mint például adatok lekérdezése vagy módosítása. A MySQL.Data package segítségével az alkalmazások egyszerűen és hatékonyan kezelhetik a MySQL adatbázisokkal kapcsolatos feladatokat, például adatok beolvasását vagy mentését.

Adatbázis

MySQL

A MySQL az Oracle globális rendszere, amelyet adatbázisok kezelésére használnak. A relációs algebrán alapul, és elsősorban a webszolgáltatások adatainak tárolására használják. A MySQL, ahogy a neve is mutatja, egy SQL-alapú adatbázis-kezelő rendszer. A MySQL képes az adatok táblázatokban történő tárolására, kezelésére és megjelenítésére. Kliens-szerver rendszerként működik.

Programozási környezet

Visual Studio Code

A Visual Studio Code egy könnyű, de hatékony forráskód-szerkesztő, amely az asztalon fut, és elérhető Windows, macOS és Linux rendszereken. Beépített JavaScript-, TypeScript- és Node.js-támogatással érkezik, és gazdag ökoszisztémával rendelkezik más nyelvekhez és futtatókörnyezetekhez (például C++, C#, Java, Python, PHP, Go, .NET). Az VS Code rendelkezik olyan funkciókkal, mint a kódszínezés, automatikus javítás, Git-integráció, kiterjesztések és sok más, amelyek segítenek felgyorsítani a fejlesztési folyamatot. A felhasználók széles választékát szolgálja ki, beleértve a kezdő és haladó fejlesztőket is. Az alkalmazás kialakítása modern, intuitív és könnyen használható. A felhasználóbarát felületen a különböző funkciókhoz könnyen hozzáférhetünk, ami javítja a produktivitást és a hatékonyságot. Az VS Code továbbá kiterjesztések által támogatja a testreszabhatóságot és az új funkciók hozzáadását.

Visual Studio 2022

A Visual Studio a Microsoft integrált fejlesztői fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvel bővült. A Visual Studio 36 különböző programozási nyelvet támogat, és lehetővé teszi a kódszerkesztőnek és a hibakeresőnek (különböző mértékben) szinte bármilyen programozási nyelv támogatását. A beépített nyelvek közé tartozik a C, C++, C++ / CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML és CSS. Más nyelvek, például a Python, a Ruby, Node.js és M támogatása plug-inek segítségével érhető el.

XAMPP

A XAMPP egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webszerver, a MariaDB adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői. Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.

Kommunikációs felületek

Discord

A Discord egy ingyenes VoIP-alkalmazás és digitális terjesztési platform, amelyet legfőképp videójáték-közösségek számára terveztek, de nem zár ki semmilyen más témájú közösséget sem. A Discordot úgy tervezték, hogy nagy rendszerigényű programok (leginkább videójátékok) futtatása mellett is gördülékenyen lehessen használni. A platform rendelkezik szöveges, kép- és videó-, valamint audiokommunikációval is. A Discord Windows, MacOS, Android, iOS, Linux operációs rendszereken és böngészőkön fut. Ez a felhasználók számára hozzáférést biztosít a kommunikációs szolgáltatások köré összpontosító eszközökhöz, például a hang- és videohívásokhoz, az állandó csevegőszobákhoz, valamint a többi játékos-központú szolgáltatással való integrációhoz, valamint a közvetlen üzenetek küldésének és a személyes csoportok létrehozásának általános képességéhez.

Github

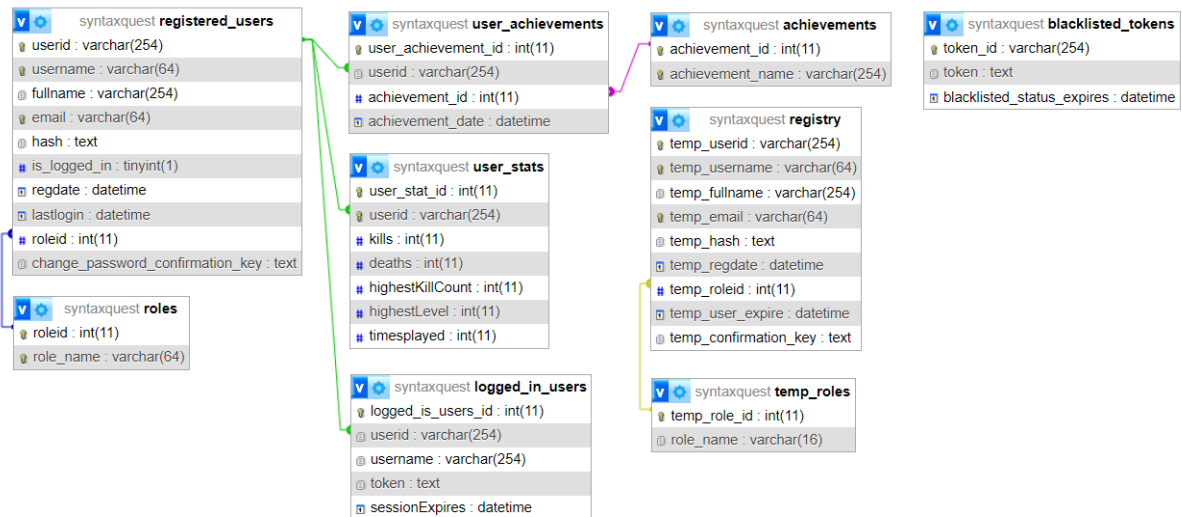
A GitHub, Inc. egy egyesült államokbeli nemzetközi vállalat, amely a Git segítségével szoftverfejlesztési verziókövetés-szolgáltatást nyújt. 2018-ban a Microsoft leányvállalata lett 7,5 milliárd dollárért. Saját funkcióin felül a Git elosztott verziókövetését és forráskódkezelését (SCM) teszi elérhetővé. Hozzáférés-kezelést és számos együttműködési funkciót nyújt, mint például bugkövetés, szolgáltatáslekérés, valamint feladatkezelés.

Trello

A Trello egy webalapú, Kanban stílusú listakészítő alkalmazás, amely az Atlassian leányvállalata. A felhasználók különböző oszlopokkal hozhatják létre a feladattábláikat, és áthelyezhetik a feladatokat közöttük. Az oszlopok általában olyan feladatállapotokat tartalmaznak, mint a Teendő, Folyamatban, Elkészült. Az eszköz személyes és üzleti célokra használható, ideértve az ingatlankezelést, a szoftverprojekt-menedzsmentet, az iskolai hirdetőtáblákat, az óratervezést, a könyvelést, a webtervezést, a játékokat és az ügyvédi irodák esetkezelését.

A webes alkalmazás szerkezete

Adatbázis szerkezete



2. ábra: syntaxquest adatbázis szerkezete

Az adatbázisok létrehozásához phpMyAdmin programot használtunk, az ábrán a tervezői nézetet láthatjuk és a táblák közötti kapcsolatot. A syntaxquest adatbázis alapértelmezett karakterkészlete utf8mb4, az illesztése utf8mb4_general_ci. Az adatbázis 9 táblát tartalmaz: a registered_users, roles, user_achievements, user_stats, logged_in_users, achievements, registry, temp_roles, blacklisted_tokens.

registered_users tábla:

userid	username	fullname	email	hash	is_logged_in	regdate	lastlogin	roleid	change_password_confirmation_key
12de0596-f95c-412b-b9db-076b8c8f5b3b	erzsi	erzsi	kutasierzsikee@gmail.com	\$2a\$11\$kv4n8UxyEE7LBVq2ShMveId3.Rk5JWOvmzOCSXdGY...	1	2024-04-23 08:59:14	2024-04-23 10:59:51	2	NULL
9855c870-1056-4090-be42-37850360319d	vitya0718	vitya	twiktor20000717@gmail.com	\$2a\$11\$R9eJxvRnKioLRB1zoFgesUxP9UzZyYFz5CjKbtg6...	0	2024-03-31 10:22:31	NULL	2	NULL
abded31c-296b-4c1d-bd4c-96a0dade6c9c	vitya0717	Tóth Viktor	lothv@kkszki.hu	\$2a\$11\$dhHcE3PHMHahro54DFsx1eRN1UTOhgV8YQIXp.2nJh...	1	2024-03-30 18:27:11	2024-04-29 13:03:06	1	NULL

3. ábra: registered_users tábla

userid: A játékos/felhasználó azonosítója, VARCHAR típusú, PRIMARY KEY

username: A játékos/felhasználó neve a játékban, VARCHAR típusú, UNIQUE KEY

fullname: A játékos/felhasználó teljes neve, CHAR típusú

email: A játékos/felhasználó e-mail-címe, amit regisztrációkor megad, CHAR típusú, UNIQUE KEY

hash: A játékos/felhasználó egyedi hash kódja, TEXT típusú

is_logged_in: A játékos/felhasználó éppen be van jelentkezve, vagy sem, TINYINT típusú

regdate: A játékos/felhasználó regisztrációjának a dátuma és ideje, DATETIME típusú éééé-hh-nn óó-pp-mm formátumban

lastlogin: A játékos/felhasználó utolsó bejelentkezésének a dátuma és ideje, DATETIME típusú éééé-hh-nn óó-pp-mm formátumban

roleid: A játékos/felhasználó szerepkörének az azonosító száma, INT típusú, FOREIGN KEY

change_password_confirmation_key: A játékos/felhasználó jelszó megváltoztató megerősítő kulcsa, TEXT típusú

roles tábla:

roleid	role_name
1	Admin
2	User
4	Vip

4. ábra: roles tábla

id: A szerepkör azonosító száma, INT típusú, PRIMARY KEY

role_name: A szerepkör megnevezése, VARCHAR típusú, UNIQUE KEY

user_achievements tábla:

user_achievement_id	userid	achievement_id	achievement_date
6	9855c870-1056-4090-be42-37650360319d	4	2024-03-31 13:38:50
7	abded31c-296b-4c1d-bd4c-96a0dade6c9c	4	2024-03-31 13:38:50

5. ábra: user_achievements tábla

user_achievement_id: A játékos/felhasználó által elért teljesítmény azonosító száma, INT típusú, PRIMARY KEY

userid: A játékos/felhasználó azonosító száma, INT típusú, FOREIGN KEY

achievement_id: A teljesítmény azonosító száma, INT típusú, FOREIGN KEY

achievement_date: A teljesítmény feloldásának a dátuma és ideje, DATETIME típusú éééé-hh-nn óó-mm formátumban

user_stats tábla:

user_stat_id	userid	kills	deaths	highestKillCount	highestLevel	timesplayed
5	abded31c-296b-4c1d-bd4c-96a0dade6c9c	591	28	205	19	28
7	9855c870-1056-4090-be42-37650360319d	5000	121	5001	0	30
8	12de0596-f95c-412d-b9db-076f8c8f5b3b	352	13	188	18	13

6. ábra: user_stats tábla

user_stat_id: A játékos/felhasználó statisztikájának az azonosító száma, INT típusú, PRIMARY KEY

userid: A játékos/felhasználó azonosítója, VARCHAR típusú, UNIQUE KEY

kills: A játékos/felhasználó által legyőzött ellenségeknek a száma, INT típusú

deaths: A játékos/felhasználó halálainak száma, INT típusú

highestKillCount: A játékos/felhasználó által elért legtöbb legyőzött ellenségek száma, INT típusú

highestLevel: A játékos/felhasználó által elért legnagyobb szint száma, INT típusú

timesplayed: A játékos/felhasználó hány alkalommal játszott, INT típusú

logged_in_users tábla:

logged_in_users_id	userid	username	token	sessionExpires
16	12de0596-f95c-412d-b9db-076f8c8f5b3b	erzsi	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOi...	2024-04-24 08:59:51
21	abded31c-296b-4c1d-bd4c-96a0dade6c9c	vitya0717	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOi...	2024-04-30 11:03:06

7. ábra: logged_in_users tábla

logged_in_users_id: A bejelentkezett játékos/felhasználó azonosító száma, INT típusú,

PRIMARY KEY

userid: A játékos/felhasználó azonosítója, VARCHAR típusú, FOREIGN KEY

username: A játékos/felhasználó neve a játékban, VARCHAR típusú

token: Egy karakterlánc, például egy kulcsszó vagy azonosító, amelyet a szintaktikai elemzés vagy lexikai elemzés során egy programban használnak, TEXT típusú

sessionExpires: A munkamenet lejáratának a dátuma és ideje, DATETIME típusú éééé-hh-nn óó-pp-mm formátumban

achievements tábla:

achievement_id	achievement_name
4	First time died
1	Firt time playing

8. ábra: achievements tábla

achievement_id: A teljesítmény azonosító száma, INT típusú, PRIMARY KEY

achievement_name: A teljesítmény neve, VARCHAR típusú, UNIQUE KEY

temp_roles tábla:

temp_role_id	role_name
1	Temp

9. ábra: temp_roles tábla

temp_role_id: Ideiglenes szerepkör azonosító száma, INT típusú, PRIMARY KEY

role_name: Szerepkör megnevezése, VARCHAR típusú

blacklisted_tokens tábla:

token_id	token	blacklisted_status_expires
01d301b5-76bb-43e5-8f98-eb135010fce5	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-24 08:24:11
047b931f-1991-45fe-981c-afbe2c9f5f92	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-25 08:18:35
084ec64d-a613-4fc6-b693-4aa09e141c49	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-19 08:21:22
1683b6df-171a-4042-9a87-01bdb74bd036	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-01 12:40:36
19a7cdaf-603e-43ea-b9ac-6a69e643a673	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-24 11:48:26
1ee523c6-a130-482d-a1c8-2107d4df6f23	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-01 13:16:01
28c584fd-c1f8-4c49-b404-6a85bc0c1888	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-01 12:42:41
3c3c71b4-0260-41bf-8769-6c23010ea8fa	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-25 08:35:41
3f8c2a4f-d067-45e8-97cf-8ae9138caf04	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-30 10:52:20
429dd34a-a6ef-412b-b778-dbbc739a931a	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-17 10:49:17
5e536d2c-2b82-40bc-9c61-5a56927ebf17	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiO...	2024-04-24 08:24:21

10. ábra: blacklisted_tokens tábla

token_id: A token azonosítója, VARCHAR típusú, PRIMARY KEY

token: Egy karakterlánc, például egy kulcsszó vagy azonosító, amelyet a szintaktikai elemzés vagy lexikai elemzés során egy programban használnak, TEXT típusú

blacklisted_status_expires: A fekete listára került tokenek lejáratának dátuma és ideje, DATETIME típusú éééé-hh-nn óó-mm formátumban

Backend szerkezete

A backend alapját az Asp.Net Core API keretrendszer adja, törekedtünk a lehető legfrissebb sdk használatára, ezért 8.0 lett használva.

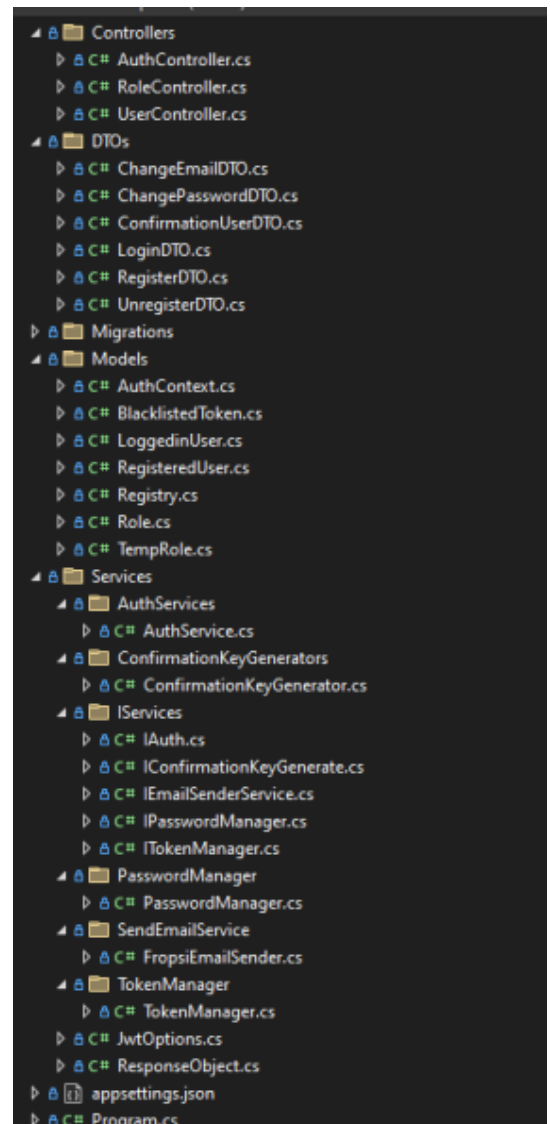
A Controller-ek felelnek a HTTP kérések (GET, POST, PUT, DELETE) fogadásáért és válaszok generálásáért. Ezek az osztályok kezelik a bejövő kéréseket, feldolgozzák azokat, és visszaadják a megfelelő válaszokat.

A DTO-k (Data Transfer Objects) olyan osztályok, amelyek célja az adatok átvitele különböző rétegek között az alkalmazásban.

A models mappában található osztályok általában az alkalmazás által használt adatmodelljeit tartalmazzák. Ezek az osztályok definiálják az adatstruktúrákat és azok tulajdonságait.

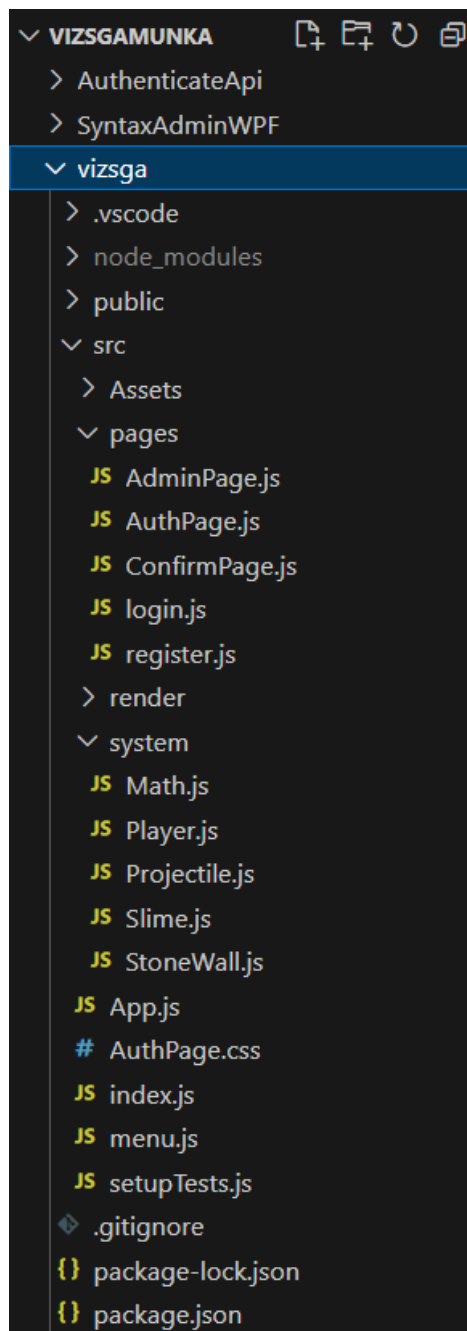
A Services egy fontos része az ASP.NET Core alkalmazásoknak, és általában a szolgáltatásokat tartalmazza. Az IService interfész általános szerződést

határoz meg az alkalmazás számára, melyet az egyéb szolgáltatások megvalósítanak. Az AuthService és a TokenManager gyakran részei a biztonsági rétegnek, ahol az AuthService feladata a felhasználók azonosítása és az engedélyek kezelése, míg a TokenManager felelős az autentikációs tokenek kezeléséért.



11. ábra: Solution explorer

Frontend szerkezete



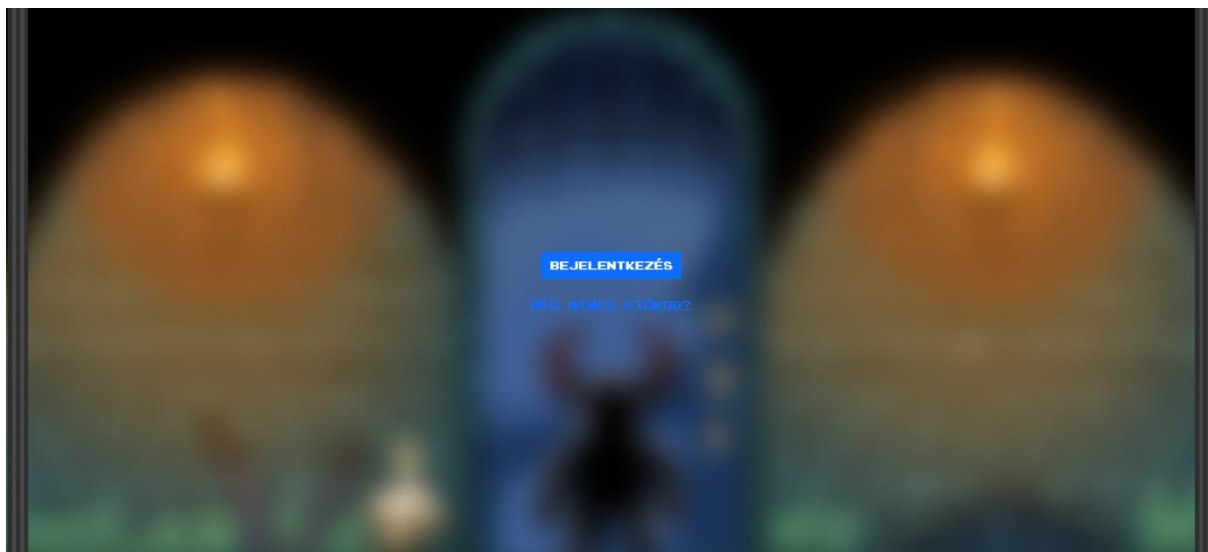
12. ábra: Fájkezelő

A szakdolgozatban bemutatjuk a React-ot, mint frontend keretrendszert, amelyet egy játék fejlesztéséhez alkalmazunk. A fő jellemzője, hogy a legtöbb adatot lokálisan tároljuk, minimalizálva ezzel a hálózati kérések számát. Emellett különböző package-eket is felhasználunk, hogy optimalizáljuk és kiegészítsük a fejlesztés folyamatát. A React egy modern és hatékony eszköz a felhasználói felületek készítéséhez, amely lehetővé teszi a komponens alapú fejlesztést és a könnyű újrahasznosíthatóságot.

A projekt során kihasználtuk a react-router-dom könyvtárat a navigáció kezelésére. Ennek segítségével összekötöttük a bejelentkezési és regisztrációs oldalakat a fő hitelesítési oldallal, hogy a felhasználók könnyen elérhessék ezeket. Az Authpage.js ebben az összefüggésben szolgált központi pontként, amely összehangolta a bejelentkezési és regisztrációs folyamatot. A react-router-dom használatával a felhasználók könnyen léphetnek át az egyes lépések között, és egyetlen fiókkal játszhatnak a játékban. A komponensek közötti átmenetek és az oldalváltások egyértelműek és intuitívak lettek a react-router-dom segítségével. A react-router-dom integrációja

hozzájárult a felhasználói élmény javításához és a regisztrációs/bejelentkezési folyamat zökkenőmentes lebonyolításához a játékhoz.

Az alkalmazás bemutatása



13. ábra: Menü

Regisztráció:

A regisztráció oldalon hozhatunk létre új felhasználót.

A regisztráció során egy egyedi felhasználónevet, egy e-mail-címet, amelyen még nincs regisztrálva felhasználó, egy jelszót,

valamint ennek a jelszónak a megerősítését kell megadnunk.

Erre az e-mail címre küldjük majd az e-mail cím megerősítőt, jelszó emlékeztetőt, figyelmeztetést a jelszó változására.

A jelszónak legalább 8 karakternek kell lennie.

Tartalmaznia kell egy kis-, és nagybetűt, egy számot, valamint egy speciális karaktert.

Amennyiben valamelyik mező hibásan lett kitöltve, a hibának megfelelő hibaüzenet jelenik meg.

Sikeres regisztráció esetén egy megerősítő e-mailt küldünk ki a megadott e-mail címre.

Ebben az üzenetben egy linket kapunk, melyben található a validációs token.

Erre a linkre kattintva a kliens elküldi a hitelesítési kérést a szervernek.

Ez a token 1 hétig használható fel, amennyiben ez az 1 hét alatt a felhasználó nem kerül megerősítésre, újat kell igényelni.

A felhasználó jelszava a megerősítés után titkosítva el lesz tárolva.

Bejelentkezés:

Amennyiben már létezik egy felhasználója a játékosnak, a jelentkezés felületen látható beviteli mezőkben kell megadni a szükséges adatait a jelentkezéshez - ez indítja el a szerver felé a hitelesítést és jelentkezést.

Sikeres hitelesítés esetén a szerver visszaküld egy JSON Web Token-t (JWT) a kliensnek, amit a böngésző helyi tárolójában (Local storage) mentünk el, és a felhasználót átirányítjuk a főmenübe. Viszont, ha a játékos már jelentkezett és szeretne kijelentkezni, azt a Kijelentkezés gomb megnyomásával tudja végrehajtani.

Jelenleg a tervben szerepel, hogy a felhasználó választhat, hogy elmenti az adatait, amellyel automatikusan belépteti a rendszer.

Amennyiben az adatok nem helyesek, vagy a felhasználó nem létezik, akkor annak megfelelő hibaüzenetet jelenítünk meg, illetve, ha az adatok helyesek, de még nincs megerősítve az e-mail, ennek megfelelő hibaüzenetet, és egy új megerősítés újra küldési lehetőséget kapunk.

A jelentkezés gomb alatt található a regisztráció és az elfelejtett jelszó link, melyek átirányítanak a megfelelő oldalra.

A jelentkezés és a főmenü felülete is a pixelgrafika stílusát öltötte, hogy tökéletesen illeszkedjen a játék pixeles világához, amely a retro érzést erősíti. Ez a dizájnválasztás különösen a rogue-like műfaj rajongói számára jelentős, akik értékelik a nosztalgikus hangulatot és a letisztult, klasszikus megjelenést

Regisztráció megerősítése:

Regisztráció esetén szükséges az email cím megerősítése, melyről a rendszer automatikusan küld egy levelet a regisztrált email címre.

Ez a levél tartalmazza a linket egy tokennel, melyre kattintva a felhasználó megerősítheti a regisztrációt.

Amennyiben ez a levél nem érkezett meg, vagy a token lejár, új levelet kell kérni.

Főmenü:

Miután bejelentkeztünk látni fogjuk a felhasználó nevünket, amit regisztrációkor megadtunk és a bejelentkezés során használtunk, továbbá a főmenüt, ahol új játékba kezdhetünk, a ranglétrát, amelyeken más játékosok pontszámait láthatjuk, vagy megnézhetjük saját beállításokat, amit játékos be tud állítani a saját igényei szerint, pl.: hangerő beállítás. Különböző kitüntetések, elért eredményeket is itt láthatjuk ha rámegyünk a Kitüntetések gombra.

További teljesítményeket is meg tudunk tekinteni, amit a játékos a játék során elért.

Pálya:

A játék pályája a klasszikus rogue-like stílusban épül fel, ahol falak és akadályok tarkítják az utat. Ezek a falak lehetőséget adnak a játékosnak a védelemre is, miközben taktikázik és manőverezik a harcokban. A játékosoknak ügyesen kell kihasználniuk ezeket a falakat, hogy elkerüljék a lövedékeket és hatékonyan megvédjék magukat az ellenségek támadásaival szemben.

Felvehető tárgyak:



A harc során az ellenség megölése után néhány ellenség felvehető tárgyakat dobnak el a tapasztalat pontok mellett. Az egyik példa erre az Életerő elixír, ami a játék életerejét tölti fel.

14. ábra: Életerő elixír

Ellenségek:



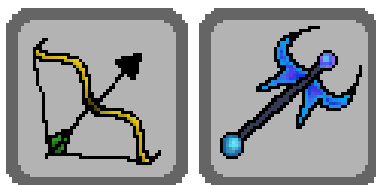
15. ábra: Slime

A játékosnak meg kell küzdenie az ellenséges lényekkel, amik próbálják megállítani a játékos. A megfelelő eszközökkel könnyedén le tudja győzni őket. Szintlépéskor a játékos gyorsabban tud végezni velük, viszont szintenként egyre többen lesznek.

Tapasztalat pontok:

Tapasztalat pontokat úgy tud szerezni a játékos, ha legyőzi az ellenséget. Az ellenség a legyőzésekor dob tapasztalat pontokat és így tud szintet lépni a játékos. Szintlépéskor az életereje is növekszik.

Fegyverek:



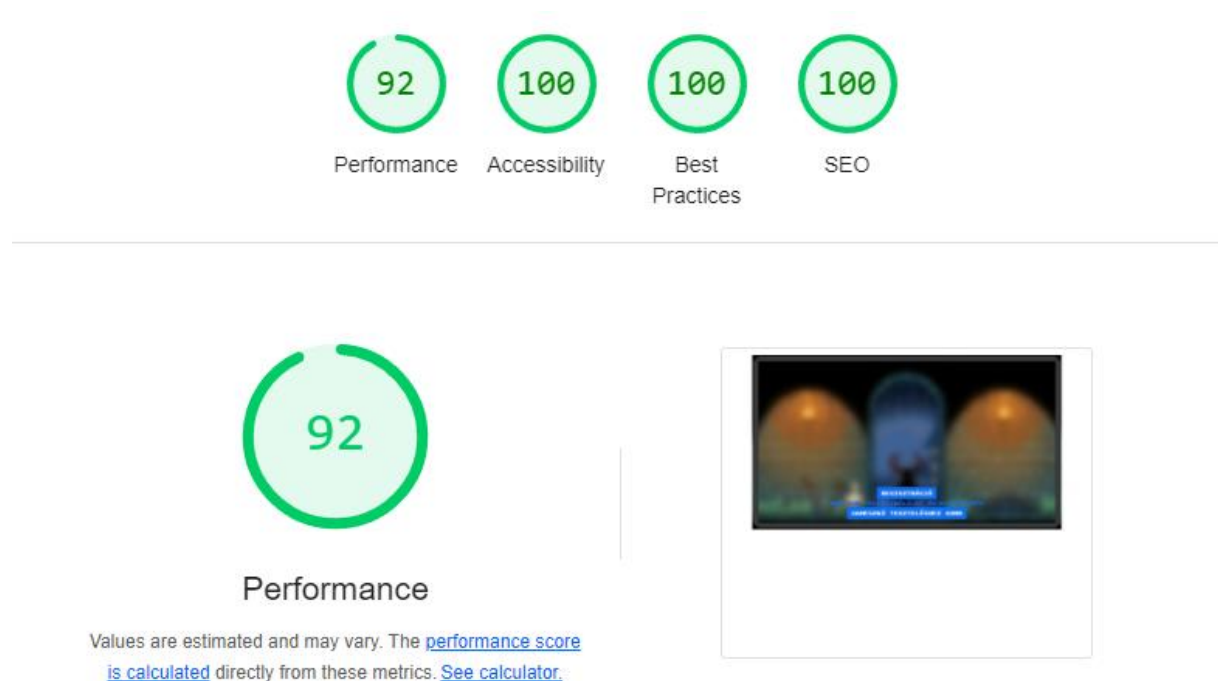
16. ábra: Fegyverek

Szintlépéskor a játékos erősebb fegyvereket kap, a leltárában fogja látni az új fegyverét a játékos. A fegyverek segítségével erősebb sebzést tud ejteni az ellenségen, így könnyebben legyőzi őket.

Tesztelés

Lighthouse

A Lighthouse a Google egyik ingyenes, nyílt forráskódú, és automatizáltan is használható elemző eszköze, amely segítségével a weboldalak fejlesztői, üzemeltetői folyamatosan monitorozhatják az oldalak betöltési sebességét és optimalizálását. Bármilyen típusú weboldallal kompatibilis, legyen az nyilvános vagy hitelesítést igénylő.



17. ábra: Lighthouse teszt

Selenium test

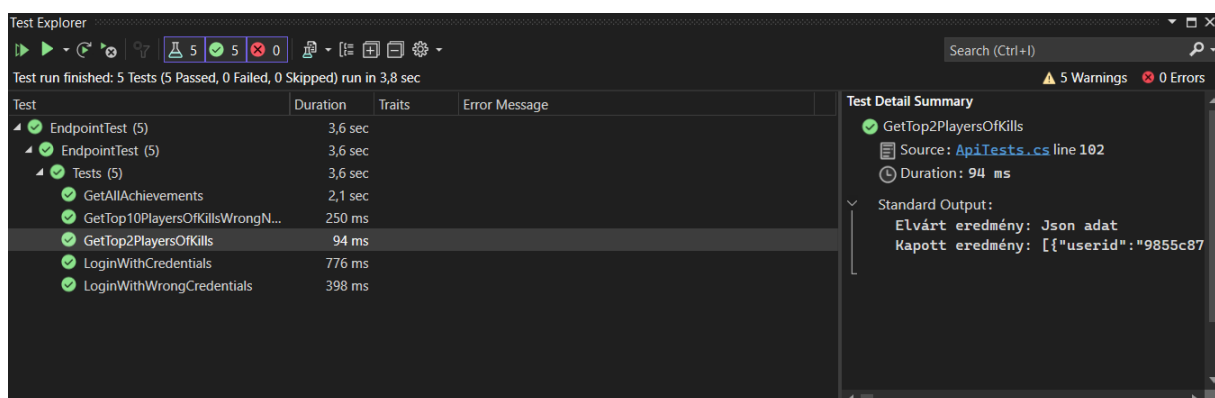
A Selenium egy nyílt forráskódú eszköz, amelyet webes alkalmazások automatizált tesztelésére használnak. Segítségével lehetőség van a webes alkalmazások interakcióinak szimulálására, mint például a kattintások és a űrlapok kitöltése. Ezáltal lehetőség nyílik a tesztelőknek a webes alkalmazások működésének automatizált ellenőrzésére és hibák felderítésére.

```
1  import time
2  from selenium.webdriver.common.by import By
3
4  from selenium import webdriver
5
6
7  driver = webdriver.Chrome()
8
9  driver.get("http://localhost:3000/")
10 driver.maximize_window()
11 time.sleep(3)
12
13 # A főoldal betöltődik bejelentkezés nélkül
14 login_button = driver.find_element(by=By.XPATH, value="/html/body/div/div/div/div/a[1]")
15 login_button.click()
16 time.sleep(3)
17 driver.save_screenshot("login_page.png");
18 #-----
19
20
21 #--- Bejelentkezés tesztelése ---
22 #1. Felhasználónév kitöltés
23 username_field = driver.find_element(by=By.XPATH, value="/html/body/div/div/div/form/div[2]/input")
24 username_field.send_keys("vitya0717")
25 #2. Jelszó kitöltés
26 password_field = driver.find_element(by=By.XPATH, value="/html/body/div/div/div/form/div[3]/input")
27 password_field.send_keys("Alma123@")
28 driver.save_screenshot("filled_fields.png")
29 #3. Bejelentkezés gombra kattintás
30 send_login = driver.find_element(by=By.XPATH, value="/html/body/div/div/div/form/button")
31 send_login.click()
32 time.sleep(3)
33 driver.save_screenshot("successfully_login.png")
34 #-----
35
36 #--- Statok lekérése a gombra kattintva ---
```

18. ábra: Selenium test kódrészlete

Unit test

A unit teszt egy olyan szoftverfejlesztési gyakorlat, amelynek során a kódot apró részekre, egységekre bontjuk, majd ezeket az egységeket külön-külön teszteljük. Célja az, hogy megbizonyosodjunk arról, hogy az egyes egységek megfelelően működnek-e izolált környezetben, és hogy azok a várt eredményeket adják vissza. A unit tesztek segítenek javítani a kód minőségét, megelőzni a hibákat, és könnyebben karbantarthatóvá tenni az alkalmazást.



18. ábra: Unit test

Mock

A unit teszt Mock-olása során ál adatokat és viselkedést állítunk elő a tesztelendő objektumok számára, hogy izolált környezetben tesztelhessük őket. Ez lehetővé teszi, hogy teszteljük az objektumok működését anélkül, hogy függene más komponensektől vagy külső rendszerektől. A Mock-olás segít abban, hogy hatékonyan és megbízhatóan teszteljük a szoftverünket, miközben minimalizáljuk a függőségeket és a külső erőforrásokat a tesztelés során.

```
[Fact]
public async Task ConfirmAccount_ShouldReturnBadRequest()
{
    // Arrange
    var registriesMock = new List<Registry>
    {
        new Registry() { TempConfirmationKey = "Doryal" },
        new Registry() { TempConfirmationKey = "Masku!" }
    }.AsQueryable();

    var authContextMock = new Mock<AuthContext>();

    var registriesObSetMock = new Mock<ObSet<Registry>>();
    registriesObSetMock.AsIQueryable<Registry>().Setup(m => m.Provider).Returns(registriesMock.AsQueryable().Provider);
    registriesObSetMock.AsIQueryable<Registry>().Setup(m => m.Expression).Returns(registriesMock.AsQueryable().Expression);
    registriesObSetMock.AsIQueryable<Registry>().Setup(m => m.ElementType).Returns(registriesMock.AsQueryable().ElementType);
    registriesObSetMock.AsIQueryable<Registry>().Setup(m => m.GetEnumerator()).Returns(registriesMock.GetEnumerator());

    authContextMock.Setup(method => method.Registries).Returns(registriesObSetMock.Object);
    var authService = new AuthService(new Mock<ITokenManager>().Object, new Mock<IPasswordManager>().Object, new Mock<IConfirmationKeyGenerate>().Object, new Mock<IEmailSenderService>().Object, authContextMock.Object);

    // Act
    var result = (ResponseObject)await authService.ConfirmAccount("Bad Confirmation Key!");

    // Assert
    using (new AssertionScope())
    {
        result.Should().NotBeNull();
        result.Should().BeOfType<ResponseObject>();
        result.Status.Should().Be(400);
        result.ResponseMessage.Should().Be("Hibás kulcs, vagy nem létező fiók!");
    }
}
```

19. ábra: Registries mock

```
[Fact]
public async Task ConfirmAccount_ShouldReturnFromCatch()
{
    // Arrange
    var authContextMock = new Mock<AuthContext>();

    // Act
    var authService = new AuthService(new Mock<ITokenManager>().Object, new Mock<IPasswordManager>().Object, new Mock<IConfirmationKeyGenerate>().Object, new Mock<IEmailSenderService>().Object, authContextMock.Object);

    var result = (ResponseObject)await authService.ConfirmAccount("Bad Confirmation Key!");

    // Assert
    using (new AssertionScope())
    {
        result.Should().NotBeNull();
        result.Should().BeOfType<ResponseObject>();
        result.Status.Should().Be(400);
        result.ResponseMessage.Should().Be("Value cannot be null. (Parameter 'source')");
    }
}
```

20. ábra: Auth mock

Forrás:

2024.03.08. - Mi a HTML? - <https://www.websiterating.com/hu/web-hosting/glossary/what-is-html/>

2024.03.08. - CSS - <https://matebalazs.hu/css.html>

2024.03.08. - Bootstrap - [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

2024.03.08. - React - <https://hu.legacy.reactjs.org/tutorial/tutorial.html>

<https://www.prooktatas.hu/hirek/front-end-webfejlesztes-react-tel-miert-jo-valasztas>

2024.03.08. - React Router - <https://attila.gludovatz.hu/posts/reszletesebb-ismerkedes-a-react-tel-routing-2-resz>

2024.03.08. - Axios - <https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/>

2024.03.08. - JWT - <https://blog.openreplay.com/using-jwt-for-authentication-in-react/>

2024.03.09. - C# - <https://learn.microsoft.com/hu-hu/dotnet/csharp/tour-of-csharp/>

2024.03.09. - Asp.Net Core API - <https://www.c-sharpcorner.com/article/asp-net-core-5-0-web-api/>

2024.03.09. - Visual Studio Code - <https://visualstudio.microsoft.com/>

<https://hub.hellowp.io/docs/tudasbazis/hasznos-eszkozok/kodszerkeszto-vscode/>

2024.03.09. - Visual Studio - <https://www.kersoft.hu/szoftverek/microsoft/visual-studio/>

2024.03.09. - XAMPP - <https://hu.wikipedia.org/wiki/XAMPP>