** **

**ZÁRÓDOLGOZAT**

Készítették:

Tóth Viktor – Kiss Gergő – Kutasi Erzsébet

Konzulens:

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

[Témaválasztás indoklása 3](#_Toc161128501)

[Felhasznált technológiák, programozási nyelvek 5](#_Toc161128502)

[Frontend 5](#_Toc161128503)

[HTML 5](#_Toc161128504)

[CSS 5](#_Toc161128505)

[BOOTSTRAP 5](#_Toc161128506)

[Axios 5](#_Toc161128507)

[React 6](#_Toc161128508)

[React Router 6](#_Toc161128509)

[JWT 6](#_Toc161128510)

[Backend 7](#_Toc161128511)

[C# 7](#_Toc161128512)

[Asp.Net Core API 7](#_Toc161128513)

[Entity Framework Core 7](#_Toc161128514)

[MySQL.Data 8](#_Toc161128515)

[Adatbázis 8](#_Toc161128516)

[MySQL 8](#_Toc161128517)

[Programozási környezet 8](#_Toc161128518)

[Visual Studio Code 8](#_Toc161128519)

[Visual Studio 2022 9](#_Toc161128520)

[XAMPP 9](#_Toc161128521)

[Kommunikációs felületek 9](#_Toc161128522)

[Discord 9](#_Toc161128523)

[Github 10](#_Toc161128524)

[Trello 10](#_Toc161128525)

[A webes alkalmazás szerkezete 11](#_Toc161128526)

[Adatbázis szerkezete 11](#_Toc161128527)

[game adatbázis 11](#_Toc161128528)

[auth adatbázis 14](#_Toc161128529)

[Backend szerkezete 17](#_Toc161128530)

[Frontend szerkezete 18](#_Toc161128531)

[Az alkalmazás bemutatása 19](#_Toc161128532)

[Tesztelés 20](#_Toc161128533)

[Lighthouse 20](#_Toc161128534)

[Forrás: 21](#_Toc161128535)

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 tapasztalatpontok 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.

# 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](https://en.wikipedia.org/wiki/HTML_element) . Az eredmény a próza, a táblázatok és az űrlapelemek egységes megjelenése a [böngészőkben](https://en.wikipedia.org/wiki/Web_browser).

### 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 írnunk.

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

### game adatbázis

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

Az adatbázisok létrehozásához phpMyAdmin programot használtunk, az ábrán a tervezői nézetet láthatjuk és a táblák közötti kapcsolatot. A game adatbázis alapértelmezett karakterkészlete utf8mb4, az illesztése utf8mb4\_general\_ci. A game adatbázis 6 táblát tartalmaz: a user, a roles, a userstats, a user\_achievements, a user\_achievement\_details és az achievements.

user tábla



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

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

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

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, FOREING KEY

user\_stats\_id: A játékos/felhasználó statisztikájának az azonosító száma, INT típusú, FOREIGN KEY

roles tábla

A képen szöveg, Betűtípus, sor, képernyőkép látható

Automatikusan generált leírás

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

role\_name: A szerepkör megnevezése, CHAR típusú

userstats tábla

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

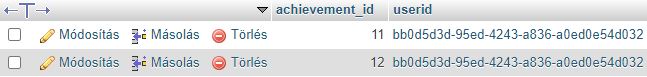
user\_stat\_id: A játékos/felhasználó statisztikájának az azonosító száma, INT típusú, PRIMARY KEY

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

deaths: A játékos/felhasználó halálainak száma, INT típusú

timesplayed: A játékos/felhasználó hányszor játszott a játékkal, INT típusú

user\_achievements tábla



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

achievement\_id: A teljesítmény azonosító száma, INT típusú, PRIMARY KEY

user\_achievement\_details tábla



achievement\_detail\_id: A teljesítmény részletének az azonosító száma, INT típusú, PRIMARY KEY

achievement\_id: A teljesítmény azonosító száma, INT típusú, FOREIGN KEY

user\_achievement\_id: A játékos/felhasználó teljesítményének az azonosító száma, INT típusú, FOREIGN KEY

achievement\_date: A játékos/felhasználó által elért teljesítményének a dátuma és ideje, DATETIME típusú éééé-hh-nn óó-pp-mm formátumban

achievements tábla

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

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

achievement\_name: A teljesítmény neve, CHAR típusú

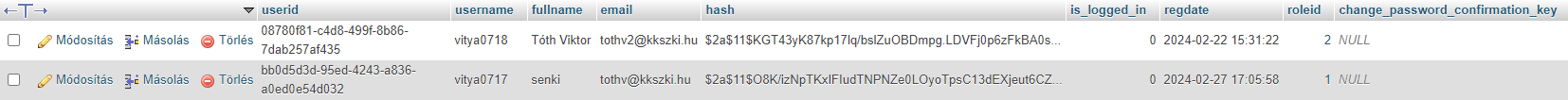
### auth adatbázis

A képen szöveg, szoftver, Betűtípus, szám látható

Automatikusan generált leírás

Az auth adatbázis alapértelmezett karakterkészlete utf8mb4, az illesztése utf8mb4\_general\_ci. Az auth adatbázisban 6 tábla található: registered\_users, roles, registry, temp\_roles, blacklisted\_tokens, logged\_in\_users.

registered\_users tábla



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

username: A játékos/felhasználó felhasználóneve, VACHAR típusú

fullname: A játékos/felhasználó teljes neve, VARCHAR típusú

email: A játékos/felhasználó e-mail-címe, VARCHAR típusú

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

is\_logged\_in: A játékos/felhasználó jelenleg be van-e jelentkezve, TINYINT típusú

regdate: A játékos/felhasználó regisztrációjának 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 amikor jelszót szeretne változtatni, akkor itt lesz tárolva ez a kulcs, aminek a segítségével tud jelszavat megváltoztatni, TEXT típusú

roles tábla

A képen szöveg, Betűtípus, sor, képernyőkép látható

Automatikusan generált leírás

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

role\_name: A szerepkör megnevezése, CHAR típusú

registry tábla

temp\_roles tábla



temp\_role\_id: A játékos/felhasználó ideiglenes szerepkörének az azonosító száma, INT típusú, PRIMARY KEY

role\_name: A játékos/felhasználó ideiglenes szerepkörének a neve, VARCHAR típusú

blacklisted\_tokens tábla:

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

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-pp-mm formátumban

## Backend szerkezete

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírásA 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 authentikációs tokenek kezeléséért.

## Frontend szerkezete

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 képen szöveg, képernyőkép, szoftver, Betűtípus látható

Automatikusan generált leírás

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

# 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.

# Forrás:

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

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

Bootstrap - <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>

React - <https://hu.legacy.reactjs.org/tutorial/tutorial.html> <https://www.prooktatas.hu/hirek/front-end-webfejlesztes-react-tel-miert-jo-valasztas>

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

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

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

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

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

Visual Studio Code - <https://visualstudio.microsoft.com/> <https://hub.hellowp.io/docs/tudasbazis/hasznos-eszkozok/kodszerkeszto-vscode/>

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

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