

***ZÁRÓDOLGOZAT***

**Készítették:**

**Turnyánszki Ákos - Bencsik Bence**

**Konzulens:**

**Horvát István**

**Miskolc**

**2025.**

**Miskolci SZC Kandó Kálmán Informatikai Technikum**

**Miskolci Szakképzési Centrum**

**SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK**

**Dokumentáció**

**tudnivalók a záródolgozathoz**

Turnyánszki Ákos - Bencsik Bence

**2024-202**

[Téma és célkitűzés: 2](#_Toc1579272956)

[1. Fejlesztőkörnyezet ismertetése 3](#_Toc558944237)

[HTML (Hypertext Markup Language) 7](#_Toc1866749496)

[A HTML célja és funkciói 7](#_Toc145419632)

[HTML és más technológiák együttműködése 7](#_Toc1874154878)

[HTML struktúra 8](#_Toc472230894)

[CSS (Cascading Style Sheets) 9](#_Toc1560561892)

[A CSS célja és funkciói 9](#_Toc1359019361)

[CSS működési elve 9](#_Toc51030413)

[CSS és HTML közötti kapcsolat 10](#_Toc117158755)

[CSS előnyei a webfejlesztésben 10](#_Toc637700506)

[Bootstrap 11](#_Toc1402803315)

[Bootstrap főbb jellemzői 11](#_Toc1914807109)

[Hol használható a Bootstrap? 12](#_Toc1683076766)

[Miért érdemes használni a Bootstrapet? 12](#_Toc184121877)

[Lehetőségek és kiegészítések a Bootstrapben 12](#_Toc604516724)

[Hátrányok vagy kihívások 13](#_Toc6552277)

[JavaScript (JS) 13](#_Toc1317678912)

[Fő jellemzői és felhasználási területei 13](#_Toc1931313852)

[Kapcsolat más technológiákkal 14](#_Toc1462808154)

[Kulcsfontosságú különbségek a Java és a JavaScript között 14](#_Toc328631012)

[XAMPP 14](#_Toc1893901895)

[Főbb jellemzők 15](#_Toc1516087377)

[Használat előnyei 15](#_Toc2084774507)

[Történeti háttér 15](#_Toc38675843)

[Telepítési követelmények 15](#_Toc1057052058)

[Console App 16](#_Toc111179522)

[Fő jellemzők 16](#_Toc1584194780)

[Hol hasznosak a konzolalkalmazások? 16](#_Toc646396612)

[Előnyei a konzolalkalmazásoknak 17](#_Toc1812005472)

[Hátrányok és korlátok 17](#_Toc1630078092)

[Fejlesztési lehetőségek és környezetek 17](#_Toc1085115888)

[A konzolalkalmazások helye a modern világban 17](#_Toc1563785866)

[WPF (Windows Presentation Foundation) – Áttekintés 18](#_Toc180108657)

[Mire jó a WPF? 18](#_Toc792194184)

[Fő előnyei 18](#_Toc1428953397)

[Hátrányai és korlátai 19](#_Toc286732668)

[Fejlesztési környezet és lehetőségek 19](#_Toc517656387)

[A WPF szerepe a modern fejlesztésben 19](#_Toc985225286)

[React 19](#_Toc214473575)

[Mire jó a React? 19](#_Toc1811985992)

[Fő előnyei 20](#_Toc1250755825)

[Hátrányai és korlátai 20](#_Toc2051023838)

[Fejlesztési környezet és lehetőségek 20](#_Toc1799364008)

[A React szerepe a modern fejlesztésben 20](#_Toc1077523546)

[CastleGame – Összefoglaló 21](#_Toc2086852667)

[Forrásaink: 22](#_Toc1850138138)

# Téma és célkitűzés:

Az elkészült programunk egy platformer típusú webes játék, . A célunk az volt, hogy egy olyan projektet hozzunk létre, amely nemcsak szórakoztató a játékosok számára, hanem lehetőséget ad a programozási készségeink fejlesztésére és bemutatására is.

Téma választásának indokai:

A játékfejlesztés régi álmunk volt, hiszen a játékok mindig is különleges helyt foglaltak el a digitális világban, és programozóként izgalmas kihívást jelentettek. A Mario által inspirált platformer játékok egyszerűsége és univerzális vonzereje ideális alapnak tűnt egy tanulási célú projekt számára.

Szakmai hasznosulás:

Ez a projekt lehetőséget adott arra, hogy integráltan alkalmazzuk mindazt, amit eddig tanultunk a programozás, a grafikai tervezés és a játéklogika terén. Emellett betekintést nyertünk a játékfejlesztési folyamatba, ami hosszú távon hasznos lehet, akár más projektekben vagy karrierépítés során.

Kihívások és tanulságok:A projekt során számos kihívással találkoztunk, beleértve a játékmechanikák kialakítását, a grafikai elemek megtervezését, valamint a megfelelő platformfüggetlen működés biztosítását. Ezek megoldása segített abban, hogy fejlesszük a problémamegoldó készségeinket, és jobban megértsük, milyen lépések szükségesek egy komplett program elkészítéséhez.

Középkori tematika:A középkori lovag karakter kiválasztása nemcsak a játék vizuális megjelenését teszi különlegessé, hanem egy történeti és fantáziaelemet is ad a játéknak, ami még izgalmasabbá és hangulatosabbá teszi a játékélményt.

Jövőbeli lehetőségek:Bár a projekt célja elsősorban a vizsgakövetelmények teljesítése volt, a játék továbbfejlesztése is a terveink között szerepel. Olyan funkciók hozzáadását tervezzük, mint például több szint, fejlettebb ellenfelek és interaktív elemek, amelyek még változatosabbá tehetik a játékmenetet.

# 1. Fejlesztőkörnyezet ismertetése

*A hardverek és szoftverek áttekintése, választásuk indoklása*

Hardverek:Fejlesztéshez használt eszköz: Dell Inspiron laptop, Intel i5 processzor, 8 GB RAM, 256 GB SSD.Tesztelés: különféle eszközök, mint például asztali PC, Android okostelefon, iPad.Indoklás: A választott hardverek biztosítják a zökkenőmentes fejlesztést és tesztelést különböző platformokon.

Szoftverek:Visual Studio Code: A könnyen bővíthető pluginrendszer és a széles körű nyelvi támogatás miatt választottuk.MySQL: Megbízható adatbázis-kezelő, amely skálázható és jól integrálható.Bootstrap: Egyszerűsíti a reszponzív dizájn kialakítását.GoogleChrome DevTools: Tesztelésre és hibakeresésre.GitHub: Verziókezeléshez és csapatmunka támogatásához.

2. Kialakított adatszerkezet bemutatása

*A program belső adatkezelésének bemutatása, adatbázisszerkezet részletezése*Adatbázisséma:Táblák:users: a játékosok adatai (ID, felhasználónév, jelszó hash, regisztráció dátuma).scores: Kapcsolatok: users.ID és scores.player\_ID idegen kulcs kapcsolatban áll.Adatfájlok:Beépített játékszintek adatai JSON formátumban tárolva.Statisztikák exportálása CSV formátumban a könnyebb elemzés érdekében

3. Tipikus algoritmusok bemutatása

Játékos mozgásának kezelése:

Fizikai alapú mozgásimplementáció (gravitáció, ugrás, sebesség).

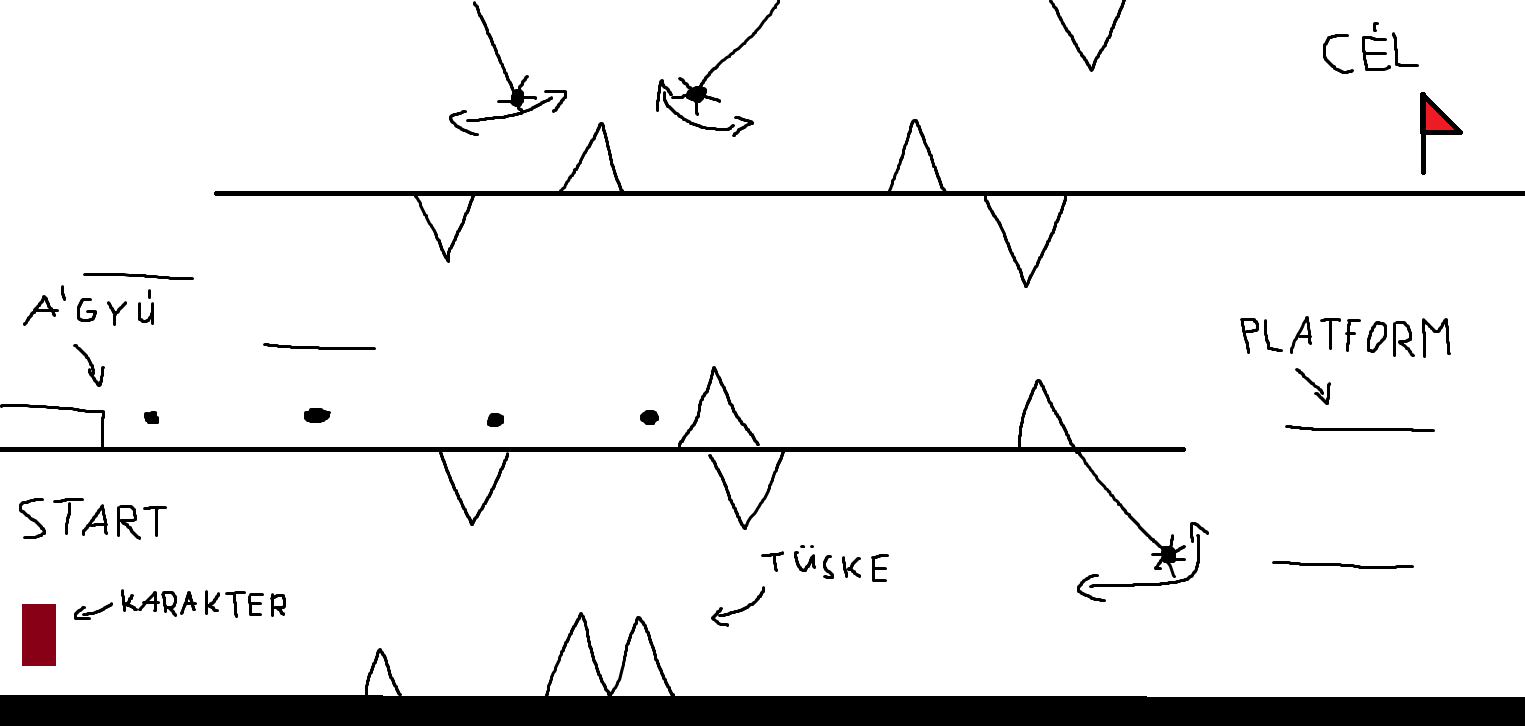
Kulcsfontosságú logika: egyszerű ütközésdetektálás a játékbeli akadályokkal.

Pontszámok mentése:

REST API hívások használata az eredmények adatbázisba mentésére.

Példa SQL lekérdezés:

INSERT INTO scores (player\_ID, level\_ID, score, date)  
VALUES (?, ?, ?, NOW());



4. Tesztelés leírása és tesztdokumentáció

Hardveres és szoftveres tesztelés:

Hardverek:

Asztali PC: Windows 10, 1920x1080 felbontás.

Mobil: Android 13, 1080x2400 képernyőfelbontás.

Tablet: iOS 16, 1536x2048 felbontás.

Tesztelés eredménye:

Optimális környezet: Minden funkció megfelelően működött.

Nem optimális használat:

Hiányzó adat esetén figyelmeztető üzenet jelenik meg.

Nagy mennyiségű adat esetén a program teljesítménye stabil maradt.

Helyes működés igazolása:

Teszt felhasználónév és jelszó:

Felhasználónév: test\_user

Jelszó: Test1234

Hibás adatbevitel kezelése: az alkalmazás megfelelő hibaüzeneteket ad.

5. Fejlesztési lehetőségek ismertetése

További funkciók:

Többszemélyes játék online mód.

Több szint hozzáadása változatosabb akadályokkal és ellenfelekkel.

Haladó statisztikai elemzések a játékos teljesítményének nyomon követéséhez.

Felhasználói élmény javítása:

Hanghatások és zenék hozzáadása.

Karakterek testreszabása.

Technológiai frissítések:

WebGL integráció a látványosabb grafikai megjelenítéshez.

AI-alapú ellenfelek a dinamikusabb játékmenet érdekében.

Program célja és lényegesebb funkciói

A játékunk egy HTML-alapú platformer, amely egy középkori lovag kalandjaira épül. A játék célja, hogy a játékos különböző akadályokat és ellenfeleket leküzdve elérje a pályák végét, miközben pontokat gyűjt és különféle interakciókat hajt végre a játéktér elemeivel. A játék fejlesztéséhez Bootstrap-et is alkalmaztunk, hogy modern, reszponzív felületet biztosítsunk. Az adatbázis integráció pedig lehetővé teszi a pontszámok nyilvántartását és a játékosok eredményeinek mentését.

A program legfontosabb funkciói:

Platformer játékmechanika: ugrás, futás, akadályok leküzdése.

Pontgyűjtő rendszer: a játékos pontokat kap az akadályok és ellenfelek legyőzéséért.

Adatbázis kapcsolat: a játékos pontszámainak mentése és megjelenítése.

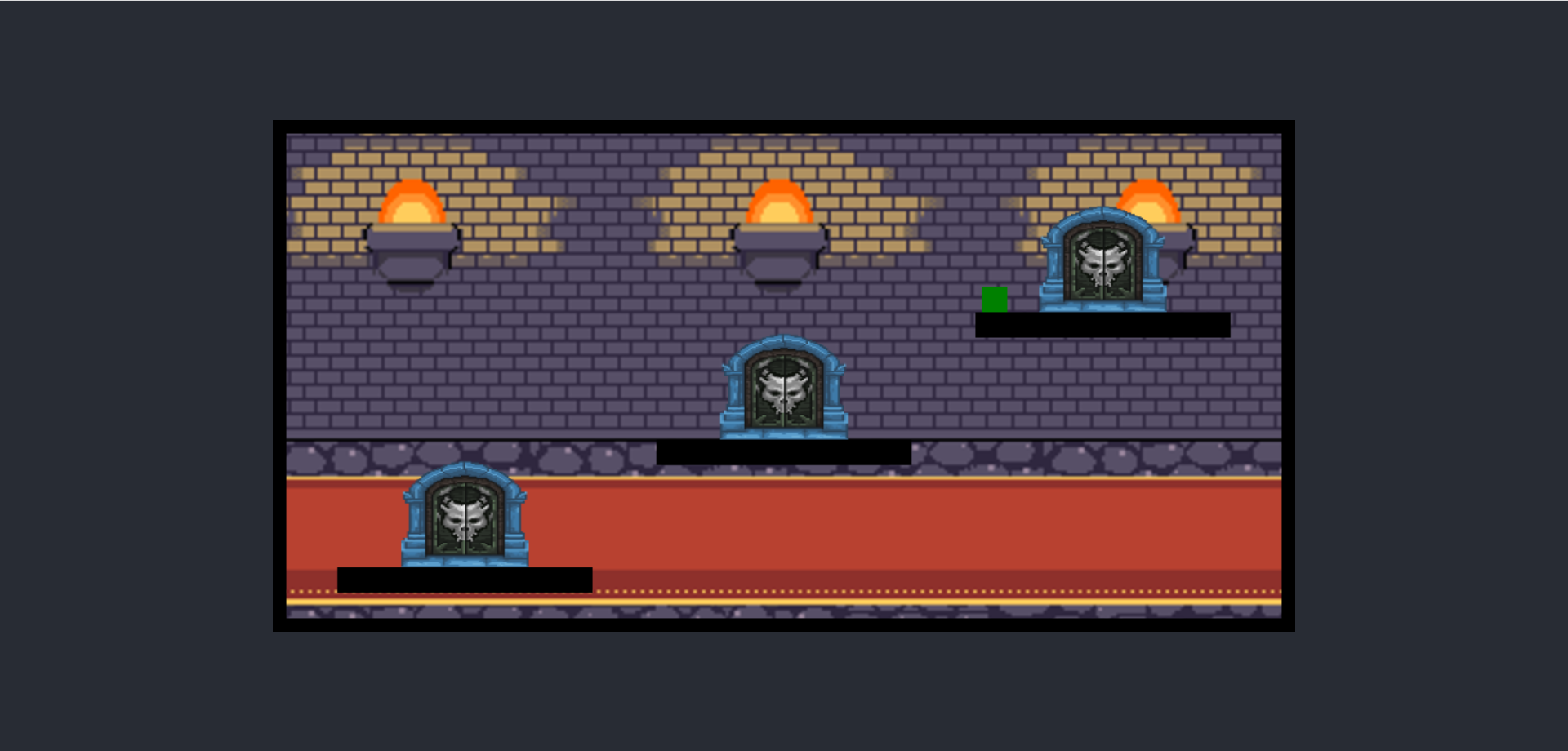
Reszponzív dizájn: a játék különböző eszközökön történő zökkenőmentes futtatása.

Egyszerű, intuitív kezelőfelület: a könnyen elsajátítható irányítás érdekében.

Szükséges hardvereszközök és szoftverek

*Hardvereszközök:*

Asztali vagy hordozható számítógép: modern processzorral és legalább 4 GB RAM-mal.

Internetkapcsolat: a játék online elérése és az adatbázis funkciók használata érdekében.

### HTML (Hypertext Markup Language)

A **HTML** (Hypertext Markup Language) a weboldalak alapvető építőköve, amely a webtartalom struktúráját és tartalmát határozza meg. A HTML egy jelölőnyelv, amelyet a webböngészők használnak annak érdekében, hogy a webszerverről érkező dokumentumokat multimédiás weboldalakká alakítsák.

#### **A HTML célja és funkciói**

**Tartalom és struktúra meghatározása**: A HTML a weboldalak alapvető struktúráját adja meg, például a címsorokat, bekezdéseket, listákat, hivatkozásokat, és más elemeket, amelyek segítik a dokumentum szervezését és navigálhatóságát.

**Interaktív elemek**: Az HTML segítségével interaktív elemek, például űrlapok, képek, videók és hivatkozások ágyazhatók be egy weboldalba. Ezen elemekkel a felhasználó kapcsolatba léphet az oldallal.

**Szemantikai struktúra**: A HTML segít a weboldal tartalmának szemantikai elrendezésében, például a <header>, <footer>, <nav>, <article> és <section> elemek használatával, amelyek logikusan felépítik a dokumentumot.

**Linkek és navigáció**: A HTML egyik alapvető funkciója, hogy lehetőséget biztosít más oldalakra mutató hivatkozások (linkek) létrehozására a <a> elem segítségével, ami a felhasználót más oldalra vagy az oldalon belső szakaszokra irányíthatja.

#### **HTML és más technológiák együttműködése**

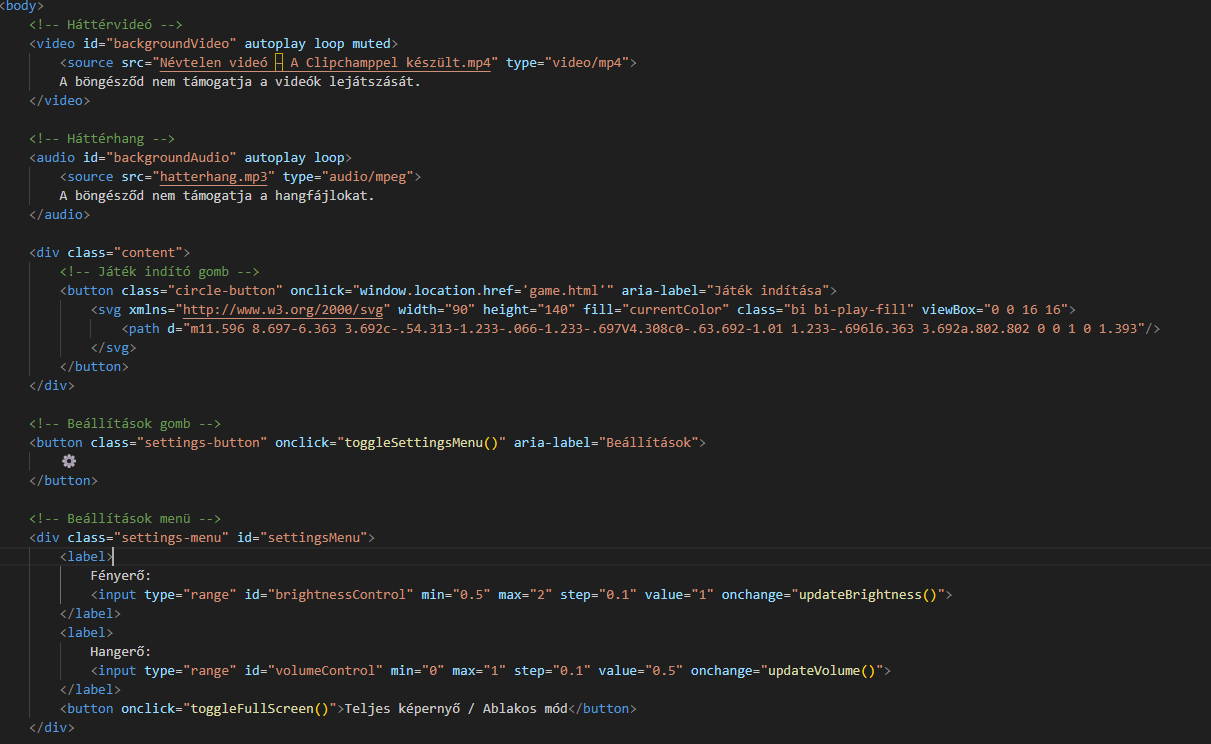
**CSS**: A HTML önállóan csak a struktúrát adja, a megjelenést a **CSS** (Cascading Style Sheets) végzi, amely a weboldal stílusát szabályozza, mint például a színek, betűtípusok, elrendezések és margók.

**JavaScript**: A HTML interaktív elemeket tartalmazhat, például gombokat és űrlapokat, amelyek viselkedését a **JavaScript** irányítja. A JavaScript lehetőséget ad a dinamikus tartalom frissítésére anélkül, hogy újratöltenénk az oldalt.

**HTML5**: A HTML5 új lehetőségeket biztosít a multimédiás tartalom (pl. videók, hangok) egyszerű kezelésére az oldalon belül, az olyan elemekkel, mint a <video> és a <audio>. Ezen kívül a HTML5 tartalmaz új formázási és interaktivitási lehetőségeket, mint a <canvas> grafikus megjelenítéshez.

#### **HTML struktúra**

A HTML kód **címkékkel** van körülvéve, amelyeket a webböngésző értelmez, hogy megjelenítse a weboldal tartalmát. A címkék például a <html>, <head>, <body>, <p>, <h1> stb. A címkéknek lehetnek **attribútumaik** is, amelyek további információkat adnak meg, például a <img> elem esetében az **src** attribútum, amely meghatározza a kéElérhetőségét.



# CSS (Cascading Style Sheets)

A **CSS** (Cascading Style Sheets) egy olyan stíluslapnyelv, amelyet a HTML dokumentumok megjelenésének és formázásának meghatározására használnak. A CSS lehetővé teszi a tartalom vizuális megjelenítésének szabályozását, beleértve a színeket, betűtípusokat, elrendezéseket és egyéb vizuális elemeket.

#### **A CSS célja és funkciói**

**Megjelenés testreszabása**: A CSS segítségével meghatározhatjuk az oldal vizuális megjelenését, beleértve a színek, háttérképek, szövegek színét és betűtípusait. Ezen kívül meghatározhatjuk az oldal elrendezését, például oszlopokba rendezhetjük a tartalmat, vagy definiálhatjuk a margókat, távolságokat.

**A tartalom és a stílus szétválasztása**: A CSS lehetővé teszi a weboldal tartalmának és stílusának szétválasztását, így a tartalom (HTML) módosítása nem befolyásolja közvetlenül a megjelenést, és fordítva. Ezáltal a weboldal karbantartása és frissítése könnyebbé válik.

**Responsív dizájn**: A CSS lehetőséget ad reszponzív dizájn kialakítására, amely biztosítja, hogy az oldal minden eszközön (asztali számítógép, mobiltelefon, tablet) jól nézzen ki. Ezt a **media queries** (mediakérdések) segítségével érhetjük el, amelyek lehetővé teszik a stíluslapok dinamikus alkalmazását a képernyőméretek és eszközök függvényében.

**Szétválasztás és újrahasználhatóság**: A CSS fájlokat külső stíluslapként tárolhatjuk, így több HTML oldal is ugyanazt a stílust használhatja, ami csökkenti a redundanciát és javítja az oldal betöltési sebességét.

#### **CSS működési elve**

**Szabályok és deklarációk**: A CSS egy sor szabályból áll, amelyek meghatározzák az elemek stílusát. A szabályok a következő formában vannak: <selector> { <property>: <value>; }. Például egy p { color: red; } szabály azt jelenti, hogy a <p> (bekezdés) elemek szövege piros színű lesz.

**Prioritás**: A CSS szabályok prioritása meghatározza, hogy ha egy elemhez több szabály is kapcsolódik, melyik alkalmazódik. A CSS rendelkezik egy **kascádoló szabállyal**, amely a specifikusságot és a szintaxist figyelembe véve határozza meg a legnagyobb priorítású szabályt.

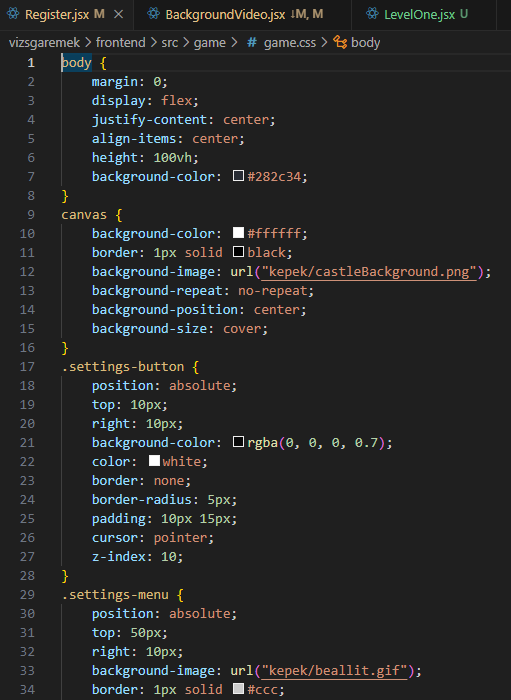
#### **CSS és HTML közötti kapcsolat**

**HTML** adja a tartalmat, és meghatározza az oldal struktúráját, míg a **CSS** felelős annak megjelenítéséért. A CSS segít abban, hogy a HTML dokumentumok esztétikailag vonzóak legyenek, ugyanakkor biztosítja a reszponzív dizájnt is, amely lehetővé teszi a weboldalak zökkenőmentes megjelenítését különböző eszközökön.

### **CSS előnyei a webfejlesztésben**

**Karbantartás és rugalmasság**: A CSS lehetővé teszi, hogy az oldal megjelenését külön fájlban tároljuk, így a tartalom frissítése nem befolyásolja a stílust. Ez egyszerűsíti a karbantartást, mivel egyetlen stíluslap módosításával az összes oldalon változást érhetünk el.

**Weboldalak optimalizálása**: A CSS fájlok különböző böngészők között is gyorsan megoszthatók, és a böngészők gyorsítótárazhatják a stílusokat, így a betöltési idő csökkenthető.



### Bootstrap

A Bootstrap egy nyílt forráskódú frontend keretrendszer, amelyet a Twitter fejlesztett ki, és mára az egyik legnépszerűbb eszköz a modern weboldalak és webes alkalmazások fejlesztéséhez. A célja, hogy felgyorsítsa a fejlesztési folyamatot előre elkészített komponensekkel, sablonokkal és stíluslapokkal.

### Bootstrap főbb jellemzői

Reszponzív tervezés:A Bootstrap automatikusan gondoskodik arról, hogy a weboldalak különböző eszközökön (mobiltelefon, tablet, asztali gép) megfelelően jelenjenek meg.

Ez a *grid rendszerén* (rácsszerkezetén) alapul, amely lehetővé teszi a tartalom könnyű elrendezését akár különböző képernyőméretekre is.Komponensek:Tartalmaz előre elkészített elemeket, például gombokat, navigációs sávokat, kártyákat, űrlapokat és modális ablakokat. Ezek testre szabhatók, hogy illeszkedjenek a weboldal dizájnjához.Előre definiált stílusok:Egyszerű alapstílusokat biztosít a HTML elemekhez, mint például címsorok, táblázatok és listák, így az alapvető megjelenés azonnal professzionálisnak tűnik.Bővíthetőség:Könnyedén kiegészíthető egyedi CSS vagy JavaScript kódokkal, hogy a weboldal teljes mértékben az adott projekt igényeire szabható legyen.Ikonok és egyéb vizuális elemek:Integrált ikoncsomagot kínál, például a Bootstrap Icons-t, amely számtalan felhasználási lehetőséget biztosít vizuális megjelenítésekhez.

### Hol használható a Bootstrap?

Egyéni weboldalak és blogok: Könnyű és gyors megoldás kezdőknek és profiknak egyaránt.

Üzleti weboldalak: Reszponzív elemekkel gyorsan készíthető modern megjelenésű weboldal.

Webes alkalmazások: Adminisztrációs felületek, felhasználói irányítópultok létrehozása egyszerűen megoldható a Bootstrap használatával.

Prototípusok készítése: Ideális gyors prototípus-fejlesztéshez, mivel a rendelkezésre álló eszközökkel percek alatt összeállítható egy működőképes alap.

### Miért érdemes használni a Bootstrapet?

Időmegtakarítás: Az előre elkészített stílusok és sablonok révén jelentősen csökkenti a fejlesztés idejét.

Könnyen tanulható: Egyszerű dokumentációja miatt gyorsan elsajátítható, még kezdők számára is.

Kompatibilitás: Az elemek már alapból működnek a modern böngészőkben, ami minimalizálja a kompatibilitási problémákat.Közösségi támogatás: A Bootstrap köré egy nagy és aktív fejlesztői közösség épült, így rengeteg példa, sablon és bővítmény érhető el online.

### Lehetőségek és kiegészítések a Bootstrapben

Animációk és interakciók: Kombinálható JavaScript kódokkal vagy jQuery-vel, hogy dinamikus és interaktív elemeket hozzunk létre.

Testreszabás: A Bootstrap saját build készítőjével személyre szabhatók a stílusok és komponensek, így csak azokat az elemeket használjuk, amelyek ténylegesen szükségesek.

Integráció más technológiákkal: Tökéletesen együttműködik más frontend keretrendszerekkel, például a React vagy Angular rendszerekkel, így nagyobb projekteknél is kiváló választás.

Támogatás SASS alapokon: Lehetővé teszi az egyedi stílusok fejlettebb módosítását, ha SASS-t használunk a Bootstraphez.

### Hátrányok vagy kihívások

Egységes megjelenés: Az alapértelmezett stílusok miatt sok Bootstrap alapú weboldal hasonló kinézetű lehet, ha nem alkalmaznak testreszabást.

Teljesítmény: Ha az összes funkcióját egyszerre töltjük be, a Bootstrap növelheti a weboldal betöltési idejét, ezért érdemes a szükséges elemekre szűkíteni a használatát.

### JavaScript (JS)

A **JavaScript** egy dinamikus, magas szintű programozási nyelv, amely a weboldalak viselkedésének meghatározására szolgál. A HTML és a CSS mellett a webfejlesztés három alapvető technológiájának egyike, amelyet a weboldalak 99%-án használnak kliensoldali szkriptek futtatására.

#### **Fő jellemzői és felhasználási területei**

**Kliensoldali működés**: A JavaScript kódot a böngészők **JavaScript motorjai** hajtják végre, például a Google Chrome V8 motorja. Ezek a motorok biztosítják a weboldalak interaktív működését, például űrlapok ellenőrzését, animációk megjelenítését vagy adatok dinamikus frissítését.

**Futásidejű környezet**: A böngészőkön kívül a JavaScript használható szerveroldalon is, például a **Node.js** platformmal, amely lehetővé teszi webalkalmazások vagy API-k létrehozását.**Sokoldalúság**: A JavaScript támogatja az eseményvezérelt, objektumorientált és funkcionális programozási paradigmákat. API-kon keresztül dolgozik a dokumentumobjektum-modellel (DOM), és lehetővé teszi interaktív felhasználói élmény létrehozását.

**Szemantika**: A JavaScript megfelel az **ECMAScript** szabványnak, amely biztosítja a nyelv hordozhatóságát különböző környezetek között.

#### **Kapcsolat más technológiákkal**

**HTML és CSS**: A HTML biztosítja a tartalom struktúráját, a CSS a megjelenést, míg a JavaScript gondoskodik az interakciókról és a dinamikus funkciókról.

**API-k**: A JavaScript segítségével hozzáférhetünk webböngészők által biztosított API-khoz, például hálózati kérések, multimédiás kezelések és grafikus megjelenítés terén.

#### **Kulcsfontosságú különbségek a Java és a JavaScript között**

A név hasonlósága ellenére a **Java** egy különálló programozási nyelv, míg a **JavaScript** elsősorban a böngészőben működő szkriptnyelv.



### **XAMPP**

Az **XAMPP** egy ingyenes és nyílt forráskódú, többplatformos webszervercsomag, amely lehetővé teszi fejlesztők számára, hogy helyi környezetben teszteljék webalkalmazásaikat. Az XAMPP az **Apache**, a **MariaDB**, a **PHP**, és a **Perl** alapvető összetevőiből áll.

#### **Főbb jellemzők**

**Platformfüggetlenség**: Az XAMPP használható Windows, macOS és Linux rendszereken.

**Komponensek integrációja**:

**Apache**: Webszerver, amely kezeli a HTTP-kéréseket.

**MariaDB**: Adatbázis-kezelő rendszer.

**PHP**: Dinamikus weboldalak létrehozására szolgáló nyelv.

**Perl**: További szkriptek futtatására.

**Egyszerű telepítés**: Az XAMPP könnyen telepíthető és minimális konfigurációt igényel, így gyorsan elérhetővé teszi a fejlesztői környezetet.**Tesztelés**: A helyi környezetben futtatott webalkalmazások lehetővé teszik a hibák gyors azonosítását és javítását.

**Kompatibilitás**: Az XAMPP ugyanazokat az összetevőket tartalmazza, mint a legtöbb élő webszerver, így az alkalmazások átvitele egyszerű.

**Bővíthetőség**: A Bitnami integrációval további csomagok, például WordPress vagy Joomla! könnyedén telepíthetők.Az XAMPP eredeti neve a **Cross-Platform + Apache + MySQL + PHP + Perl** rövidítése volt, azonban 2015-től a **MariaDB** váltotta fel a MySQL-t. A név azóta az **XAMPP Apache + MariaDB + PHP + Perl** jelentést hordozza.

#### **Telepítési követelmények**

Windows esetén az XAMPP futtatásához a **Microsoft Visual C++ Redistributable** szükséges. A csomag **.zip**, **.tar**, vagy **.exe** formátumban letölthető, és azonnal futtatható.

# Console App

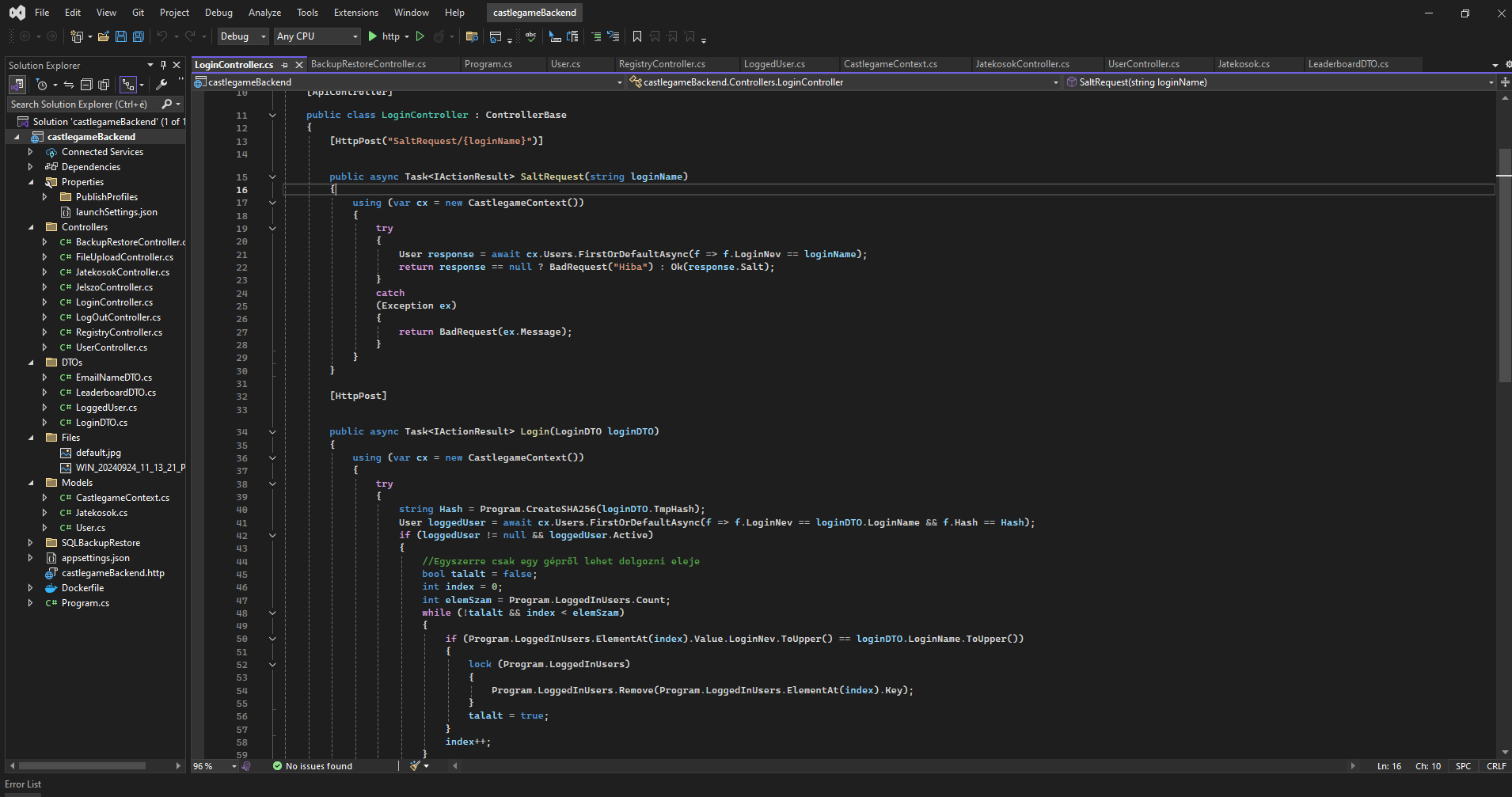
A konzolalkalmazások olyan programok, amelyek kizárólag szöveges alapon kommunikálnak a felhasználóval, jellemzően a billentyűzet és a kijelző használatával. Ezek szemben állnak a grafikus felhasználói felülettel rendelkező alkalmazásokkal (GUI), amelyek interaktívabb, vizuálisan fejlettebb eszközöket, például egér vagy érintőképernyő használatát is megkövetelhetik.

### Fő jellemzők

Egyszerűség:A konzolalkalmazások egyszerűbbek, mivel nem tartalmaznak vizuális elemeket, például gombokat vagy ablakokat.A programozók számára gyakran egyértelműbbé teszik a kód logikájának és működésének megértését.Interakció:A felhasználó inputot adhat a billentyűzet segítségével (például parancsokat vagy adatokat), míg a kimenet általában szöveg formájában jelenik meg a konzolablakban.Gyorsaság és hatékonyság:A szöveges alapú interfész általában kevesebb erőforrást igényel, így a konzolalkalmazások gyorsabban futnak, és hatékonyabban használják a rendszer erőforrásait.

### Hol hasznosak a konzolalkalmazások?

Rendszergazdai eszközök:Számos rendszergazdai és automatizációs feladatot konzolalkalmazásokkal oldanak meg, például a parancssori eszközök segítségével (PowerShell, Bash, CMD).Ezek a feladatok lehetnek fájlkezelés, adatbázis-kezelés, vagy hálózati konfiguráció.

Oktatás és programozás tanulása:Kezdő programozók számára ideális, mivel a GUI komplexitásának kiküszöbölésével lehetőséget ad a programozási alapelvek elsajátításáraEgyszerű, jól követhető példákat kínál új programozási nyelvek tanulásakor.Speciális ipari alkalmazások:Egyes szervezetek továbbra is használják konzolalkalmazásaikat, mivel ezek megbízhatóak, karbantartásuk egyszerű, és több évtizedes rendszereken is működnek.Ilyenek például banki rendszerek, ipari vezérlők vagy speciális adatfeldolgozó rendszerek.Automatizált feldolgozás:Gyakran használnak konzolalkalmazásokat adatfeldolgozási és adattranszformációs feladatokhoz, mivel ezek gyorsan és hatékonyan dolgoznak nagy mennyiségű adattal.

### Előnyei a konzolalkalmazásoknak

Kis méret és erőforrásigény: Nincs szükség bonyolult grafikus komponensekre, így a futtatásuk kevesebb memóriát és CPU-t igényel.

Gyors fejlesztés: Az egyszerű interfész miatt gyorsabban fejleszthetők és tesztelhetők.

Rugalmasság: Különösen jól alkalmazhatók bonyolult szkriptek, automatizációs feladatok és speciális algoritmusok esetén.

Hordozhatóság: Általában könnyen átültethetők más operációs rendszerekre, mivel kevés függőséggel rendelkeznek.

### Hátrányok és korlátok

Felhasználói élmény: Az egyszerű szöveges interfész nem nyújt olyan felhasználói élményt, mint a grafikus alkalmazások.

Tanulási görbe: A kezdő felhasználók számára kevésbé intuitív, különösen, ha bonyolult parancsokat kell megadniuk.

### Fejlesztési lehetőségek és környezetek

Modern programozási környezetek támogatása:Olyan eszközök, mint a Visual Studio vagy az IntelliJ IDEA, megkönnyítik a konzolalkalmazások fejlesztését, akár .NET, Java vagy Python nyelveken.

Integráció más rendszerekkel:Konzolalkalmazások könnyen integrálhatók automatizált szkriptekkel vagy más programokkal, amelyekkel a háttérben dolgoznak együtt.

### A konzolalkalmazások helye a modern világban

Bár a grafikus felületek egyre fejlettebbé váltak, a konzolalkalmazások továbbra is relevánsak, különösen ott, ahol gyorsaság, hatékonyság és egyszerűség szükséges. Az adminisztrációs feladatok, adatfeldolgozás és az oktatás területein még mindig elengedhetetlen eszközök, amelyek egyszerűségük miatt időtálló megoldásként szolgálnak.

# WPF (Windows Presentation Foundation) – Áttekintés

A **WPF (Windows Presentation Foundation)** a Microsoft által fejlesztett grafikus keretrendszer, amelyet modern és interaktív asztali alkalmazások fejlesztésére terveztek a .NET platformon. Lehetővé teszi a fejlett UI-komponensek használatát, valamint támogatja a multimédiát, adatvizualizációt és animációkat.

### **Mire jó a WPF?**

**Üzleti alkalmazások**: Komplex adatkezelési és vizuális megoldások, például táblázatok, jelentések, dashboardok.**Multimédiás és interaktív szoftverek**: 3D grafikák, videók és animációk könnyed kezelése.**GIS és térinformatikai rendszerek**: Nagy teljesítményű adatmegjelenítés, térképes vizualizáció.**Tudományos és mérnöki alkalmazások**: Interaktív adatmodellezés és valós idejű elemzés.**Korszerű UI fejlesztés**: Reszponzív, testreszabható és skálázható felhasználói felületek kialakítása.

### **Fő előnyei**

**XAML-alapú fejlesztés**: A megjelenítés és a logika elválasztható, így a UI könnyebben kezelhető és módosítható.**Hardveres gyorsítás**: A WPF kihasználja a GPU-t, így gyors és hatékony grafikai megjelenítést biztosít.**Erőteljes adatkezelés**: Az MVVM (Model-View-ViewModel) architektúra támogatásával a kód tiszta és jól szervezett marad.**Testreszabható megjelenés**: Stílusok, sablonok és témák segítségével egyedi és modern UI hozható létre.**Skálázhatóság és DPI-függetlenség**: A vektorgrafikus megjelenítés révén az alkalmazások különböző felbontásokon is jól mutatnak.

### **Hátrányai és korlátai**

**Meredek tanulási görbe**: A kezdők számára az XAML, az MVVM és az eseményvezérelt modell elsajátítása időigényes lehet.**Nagyobb erőforrásigény**: A WPF alkalmazások több memóriát és CPU-t igényelhetnek, különösen bonyolult UI esetén.**Windows-specifikus technológia**: Nem fut natívan más operációs rendszereken, bár az újabb .NET Core verziók némi kompatibilitást biztosítanak.

### **Fejlesztési környezet és lehetőségek**

**Fejlesztői eszközök**: A Visual Studio a legnépszerűbb fejlesztői környezet WPF alkalmazásokhoz.**Programozási nyelvek**: A WPF főként C# és VB.NET nyelveken használható.**MVVM minta**: A legjobb gyakorlat szerint az MVVM architektúra alkalmazása biztosítja a kód tisztaságát és a könnyebb karbantarthatóságot.

### **A WPF szerepe a modern fejlesztésben**

Bár a webes és mobil technológiák egyre népszerűbbek, a WPF továbbra is jelentős szerepet játszik az üzleti, ipari és tudományos szoftverek fejlesztésében. Azok számára, akik teljes értékű asztali alkalmazást szeretnének fejleszteni Windows platformra, a WPF egy erős, rugalmas és hosszú távon is fenntartható megoldást kínál.

# React

A **React** egy nyílt forráskódú JavaScript-könyvtár, amelyet a Facebook fejlesztett ki interaktív felhasználói felületek létrehozására. A komponensalapú architektúra és a deklaratív programozási modell segítségével hatékonyan kezelhető a felhasználói felületek állapota és dinamikus frissítése.

### **Mire jó a React?**

**Webes alkalmazások**: Dinamikus, reszponzív UI-k fejlesztésére alkalmas.**Mobilfejlesztés**: A React Native segítségével natív mobilalkalmazások készíthetők.**Single Page Applications (SPA)**: Hatékonyan kezeli az oldal frissítés nélküli tartalomváltásokat.**Komplex UI-k kezelése**: Moduláris komponensrendszer segíti a nagyobb projektek átlátható fejlesztését.

### **Fő előnyei**

**Komponensalapú struktúra**: Az UI kisebb, újrahasznosítható komponensekre bontható.**Virtuális DOM**: Hatékony frissítési mechanizmus gyors változáskezeléshez.**Deklaratív szintaxis**: Az UI állapota egyszerűbben kezelhető.**Erőteljes ökoszisztéma**: Nagy közösség, bővítő modulok, és integrációs lehetőségek.**SEO-barát**: Server-Side Rendering (SSR) támogatás a Next.js által.

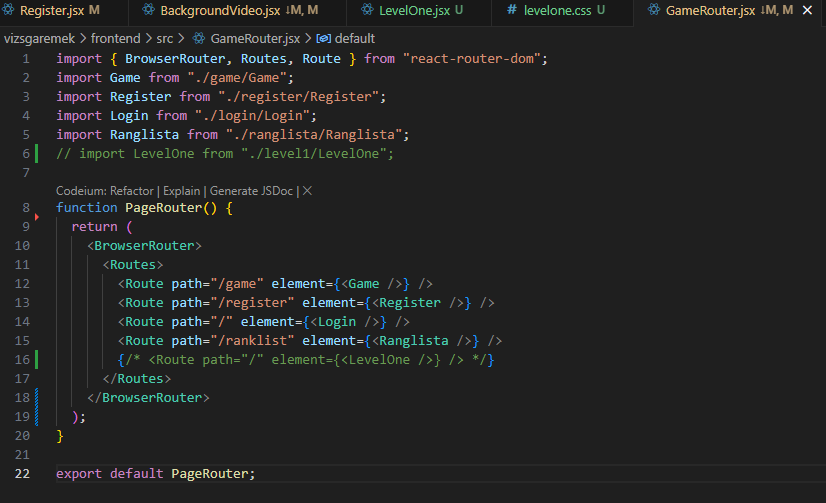
### **Hátrányai és korlátai**

**Tanulási görbe**: Kezdőknek komplex lehet a JSX, state management és lifecycle hook-ok kezelése.**Gyorsan fejlődő ökoszisztéma**: A folyamatosan változó eszközkészlet miatt a fejlesztőknek naprakésznek kell lenniük.**Nem teljes keretrendszer**: Kiegészítő eszközökre, pl. Reduxra vagy React Routerre lehet szükség komplex projektekben.

### **Fejlesztési környezet és lehetőségek**

**Fejlesztői eszközök**: Visual Studio Code, React DevTools.**State management**: Context API, Redux, Recoil vagy Zustand.**Tesztek és build rendszerek**: Jest, React Testing Library, Webpack, Vite.**Backend integráció**: REST API vagy GraphQL segítségével könnyen kapcsolható szerveroldali szolgáltatásokhoz.

### **A React szerepe a modern fejlesztésben**

A React napjaink egyik legnépszerűbb frontend technológiája, amely alkalmas kis és nagy webes alkalmazások fejlesztésére egyaránt. A folyamatos fejlődés, a közösségi támogatás és a bővítő eszközök biztosítják, hogy hosszútávon is releváns maradjon a modern webfejlesztésben.

# CastleGame – Összefoglaló

**Projekt célja:**  
 A CastleGame egy középkori tematikájú platformer webes játék, amelyet a programozási készségek fejlesztése és bemutatása céljából hoztunk létre. A játék Mario-stílusú mechanikákat követ, ahol a játékos akadályokat és ellenfeleket leküzdve haladhat előre a pályákon.

**Fejlesztőkörnyezet:**  
 A fejlesztés Dell Inspiron laptopon zajlott, Visual Studio Code szerkesztővel és MySQL adatbázissal. A játék reszponzivitását Bootstrap biztosítja, a verziókezelést pedig GitHub segíti. Tesztelés több platformon történt, beleértve PC-t, Android és iOS eszközöket.

**Adatszerkezet:**  
 A játék adatbázisa a játékosokat és pontszámokat tárolja MySQL-ben, JSON formátumot használ a pályaadatok tárolására, és CSV exportálási lehetőséget kínál.

**Fontosabb algoritmusok:**Fizikai alapú mozgás (gravitáció, ugrás, ütközésdetektálás)Pontszámok mentése és lekérdezése REST API-n keresztül

**Tesztelés és eredmények:**  
 A játék tesztelése különböző eszközökön történt. Az alkalmazás stabilan futott, a hibakezelés megfelelően működött, és a teljesítmény nagy adatforgalom mellett is optimális maradt.

**Jövőbeli fejlesztések:**Többszemélyes online mód.Több pálya és összetettebb ellenfelek.AI-alapú ellenfelek és WebGL-alapú grafikai fejlesztések.Hanghatások és testreszabható karakterek

**Főbb funkciók:**Platformer mechanikák (ugrás, futás, akadályok).Pontgyűjtő rendszer és eredménymentés adatbázisban.Reszponzív kialakítás, több eszközön futtatható játék

A CastleGame nemcsak egy szórakoztató játék, hanem egy komplex fejlesztési projekt, amely számos technológiai és programozási kihívást ötvöz.

# Forrásaink:

<https://chatgpt.com/>

<https://en.wikipedia.org/wiki/HTML> 2024 dec 3.

<https://en.wikipedia.org/wiki/CSS> 2024 dec 4.

<https://getbootstrap.com/> 2024 dec 8.

<https://hu.wikipedia.org/wiki/JavaScript> 2024 dec 10.

<https://en.wikipedia.org/wiki/Console_application> 2024 dec 12.

<https://en.wikipedia.org/wiki/XAMPP> 2024 dec 14.

<https://en.wikipedia.org/wiki/PHP> 2024 dec 19.

<https://hu.pinterest.com/> 2025 marc 18.

<https://watabou.github.io/> 2025 marc 18.

# Trello

<https://trello.com/b/2L97k7uV/kingdom>

# Github

<https://github.com/turnyanszkia/vizsgaremek>

<https://github.com/turnyanszkia/castlegameBackend>

https://github.com/turnyanszkia/CastleGameAdmin