**Ez nem része a dolgozatnak, csak jelmagyarázat!!!!!!!**

**Kérjük ezt az oldalt törölni a dolgozatból!!!!**

[A szögletes zárójelben szereplők csak magyarázatként szerepelnek a sablonban. Azokat mindenképpen törölni kell!!!]

<A relációs jelek között szereplő szöveg helyére mindenkinek a saját magára vonatkozó adatokat kell beírni.>

{A kapcsos zárójelek közötti szöveg szükség esetén használandó.}

A sablonban található vörössel szedett szövegrészek színe csak a könnyebb felismerhetőség miatt színezettek. Kérjük azokat feketére állítani szíveskedjenek!

|  |
| --- |
| [Dolgozat gerincén, a fekete kötésen:] <Dolgozat száma> |

**SZAKDOLGOZAT**

**Tökös Ötös**

**DUNAÚJVÁROSI EGYETEM**



**MÉRNÖK INFORMATIKUS BSC**

# SZAKDOLGOZAT

Legyen ön is milliomos

**Tökös Ötös**

mérnök informatikus jelölt

**02 (Tökös ötös)**

**Legyen Ön is milliomos!**

A projekt célja egy Legyen Ön is milliomos

játék publikálása 3 platformon.

Regisztráció kezelése adatok tárolása

adatbázisban, sensitive adatok titkosítva.

Cordova keretrendszerrel UI megvalósítása.

Backend megvalósítása php serverrel.

**Bajnok Tamás Scrum Master**

[bajnoktomi96@gmail.com](mailto:bajnoktomi96@gmail.com)

-Backend fejlesztés(PHP)

-Az alkalmazáshoz szükséges infrastruktúra kiépítése (adatbázis- és webszerver), konfigurálása

-Backend és frontend összekapcsolása (Ajax)

-Esetlegesen felmerülő problémák esetén a csapat többi tagjának támogatása

**Tóth Tamás   Backend Developer**

[ttamas0713@gmail.com](mailto:ttamas0713@gmail.com)

-Backend fejlesztés

-Adatbázis tervezés, fejlesztés

**Szabó Attila Frontend Developer**

[maxxywarrior@gmail.com](mailto:maxxywarrior@gmail.com)

Front-end Grafika (pl.: gombok, megjelenés),

Összejövetelek/Chat dokumentálása

Front-end dokumentálás

Back-end feedback (adatbázis, algoritmusok)

**Varsa László Frontend Designer**

vlz94@citromail.hu

Front-end grafikai tervek (összetevők elhelyezkedése)

Front-end programozása (js, css)

Front-end dokumentáció

**Nagy Ádám Product Owner**

[nagy.adam0901@gmail.com](mailto:nagy.adam0901@gmail.com)

Vállalt feladatok:

-A csapat összefogása

-Megbeszélések ütemezése

-Backend és Frontend összekötésének menedzselése

-Backend fejlesztés

**Ütemezés:**

Minden hét keddi napján Dunaújvárosban személyes megbeszélés, szombati napra törzs idő kijelölése, ezen idő alatt mindenki biztosít elérhetőséget a csapat irányába. (Az időpont dinamikusan változik a személyes megbeszélésen megbeszéltek szerint.)

**[Konzultációs igazolás a Szakdolgozat programból]**

**[Értékelő lap a Szakdolgozat programból]**

Kivonat

[Magyar nyelvű (max. 1 oldalas) összefoglaló]

Abstract

[Angol (kivételesen német) nyelvű, max. 1 oldalas összefoglaló]

[Nyilatkozat a Szakdolgozat programból]

Tartalomjegyzék

[Címsorokból generált tartalomjegyzék!]

## Bevezetés

A projekt célja egy Legyen ön is milliomos játék publikálása 2 platformon (Android és web), illetve demózás IOS platformon. Regisztráció kezelése, adatok tárolása adatbázisban, érzékeny adatok titkosítva. Cordova keretrendszerrel UI megvalósítása. Backend megvalósítása PHP szerverrel.

A projekt sikere nagyban múlik az alap ötlet megvalósíthatóságán, ugyanakkor nagyban függ a projektbe bekapcsolódó emberek mennyire tudnak egy csapatként küzdeni a cél megvalósításának érdekében.

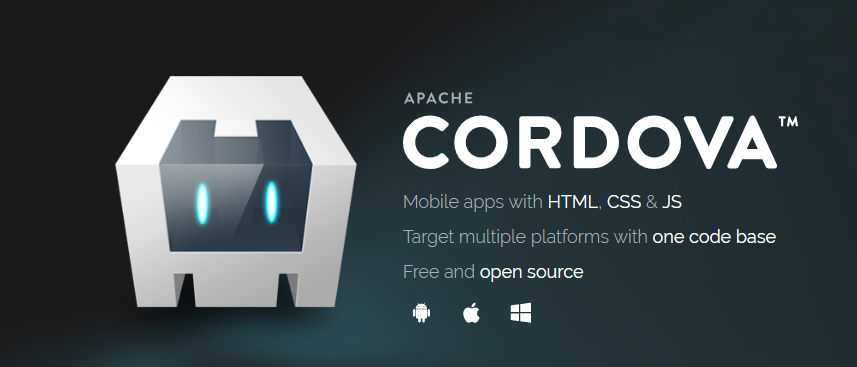
Első lépés ötlet keresés, mely során mérlegelni kell a felmerülő ötlet kivitelezhetőségét, egy eleve rosszul előadott ötletre nehezebb csapatot építeni. A Legyen ön is milliomos önmagában nem egyedi ötlet van már a piacon számtalan verziója sok csapat megvalósította már. Ami ezen tények ellenére is sikeressé tudták tenni az ötletet az a megvalósítási technológia, mely forradalmi és jelen pillanatban kevéssé kiaknázott a fejlesztők körében. Sok fejlesztő környezet és eszköz létezik, de kevés van melynek használatával felhasználhatóvá lefordíthatóvá válik a kód a két legjobban elterjedt mobil operációs rendszerre, ugyanakkor a számítógépet előnyben részesítőket is a kód gyökeres átgondolása nélkül megszólíthat a fejlesztő csapat. Ezen nagy lefedettség tudatában alakult ki a projekt ötlete véglegesen.

Második lépés a csapat kialakítása. Létfontosságú a jó csapat összetétele, a projekt sikerességéhez szükséges a stabil háttér rendszer mivel az applikáció szívét adja, ezen rendszer a kritikus pontja a projekt sikerességének, így elengedhetetlen egy a hálózatokat jól ismerő csapattag. Ugyanakkor egy embernek ezen feladat az adattárólástól a szerver oldali programozáson át nagy feladat így a projekten két ember is megkapta ezen feladatkört. Szükséges a játék dinamikáját és a háttér rendszer közötti részt is jól megvalósítani ezen hiányosságok szintén a teljes projekt vesztét okoznák, így a JS feladatokat szintén két ember kapta meg faladatául. Már csak egy rész nem került említésre ugyanakkor ez is ugyanannyira fontos mint az előző két rész az applikáció megjelenése mely CSS segítségével lett kialakítva az alap HTML kódból, erre a feladatra egy ember lett megbízva a csapatból. A csapattagok csatlakozása után kezdetét vehette a megvalósítás megtervezése.

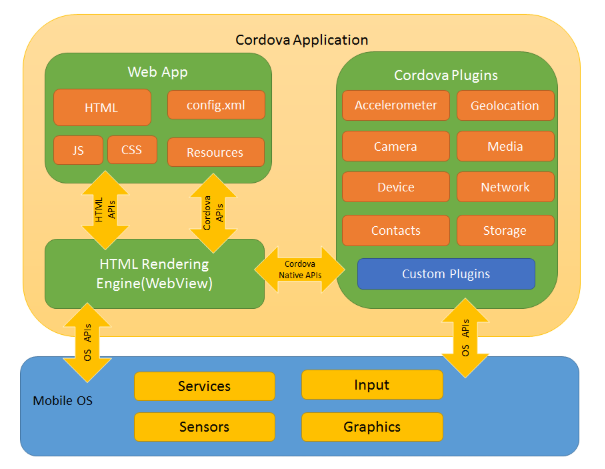
# A tervezés

Egy applikáció fejlesztése nehéz feladat, mivel nehéz döntéseket kell meghozni még a program megírása előtt. Elsősorban fel kellett mérni a csoport preferenciáit. Ez alatt programozási nyelvismeret, integrált fejlesztői környezet (IDE) [[1]](#footnote-1) kiválasztásán volt a legnagyobb hangsúly. Emellett el kellett dönteni, hogy milyen erőforrásra volt szükség. Egy webes / Android applikáció legfőbbként 2 részre osztható: Backend, Frontend. Ezeket a részeket több alrészre is kellett osztani, annak érdekében, hogy megosztott legyen a csapat erőforrása. Személyes megbeszélés során a csapat egyet értett, hogy Cordova keretrendszer lenne a legmegfelelőbb az applikáció fejlesztésére. Emellett az IDE amit mindenki ismert az a Visual Studio. Egyéb lehetőségek között lehetett volna a Netbeans is. A fejlesztéshez szükséges többi erőforrás: Scrum Master, Product Owner, Frontend Developer, Backend Developer, Frontend Designer.

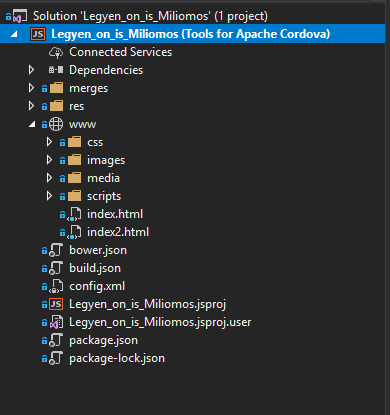
## Cordova



,,Az Apache Cordova egy nyílt forráskódú mobilfejlesztési keret. Ez lehetővé teszi a szabványos webes technológiák - HTML5, CSS3 és JavaScript - használatát a platformok közötti fejlesztéshez. Az alkalmazások az egyes platformokra célzott csomagolásokon belül futnak, és a szabványoknak megfelelő API-kötésekre támaszkodnak az egyes eszközök képességeihez, például érzékelőkhez, adatokhoz, hálózati állapothoz stb.”[[2]](#footnote-2)



1. ábra Cordova felépítése

A Cordova projectet létrehozva az alábbi mappaszerkezet alakul ki

2. ábra Cordova mappa struktúra

A WWW mappa alá kerül be az eszközökre befordítandó mappastruktúra.

A css mappa tartalmazza a kinézetért felelős css fájlokat.

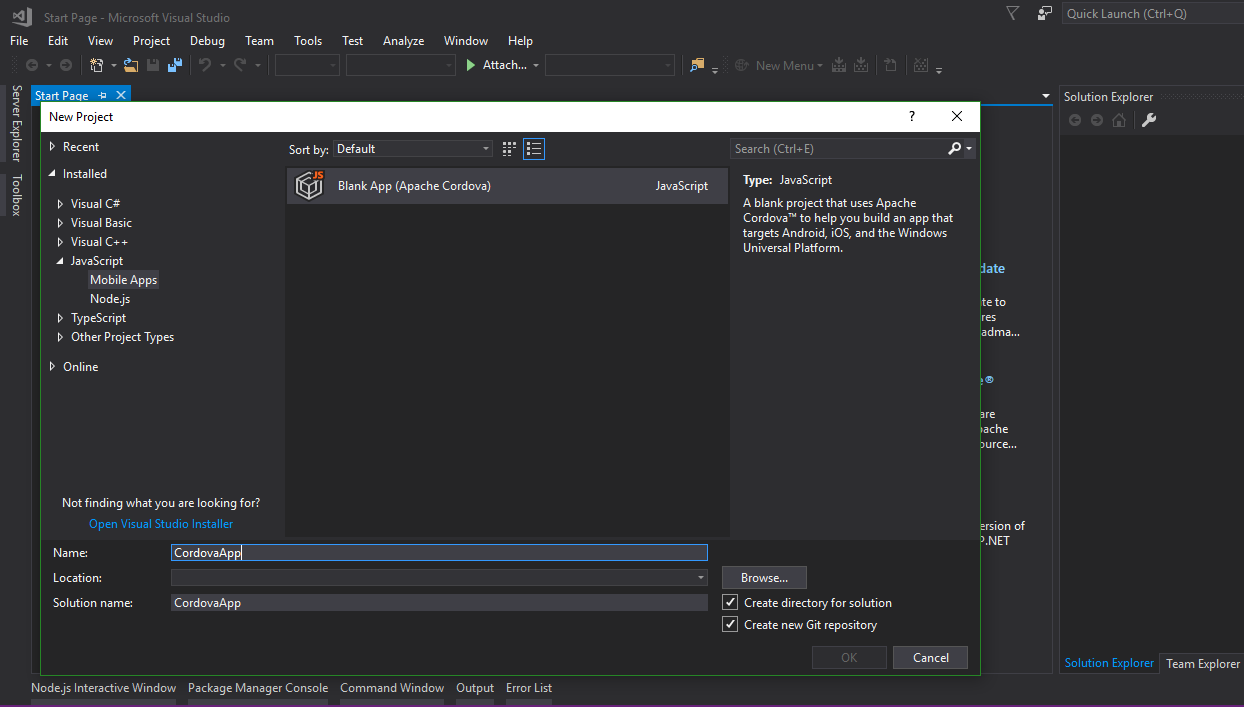
A media mappa az alkalmazásban előforduló képeknek ad helyet.

A script mappa a JS-t tartalmazza mely vezéreli a az egyes interakciók esetén végrehajtandó feladatokat.

## Visual Stúdió

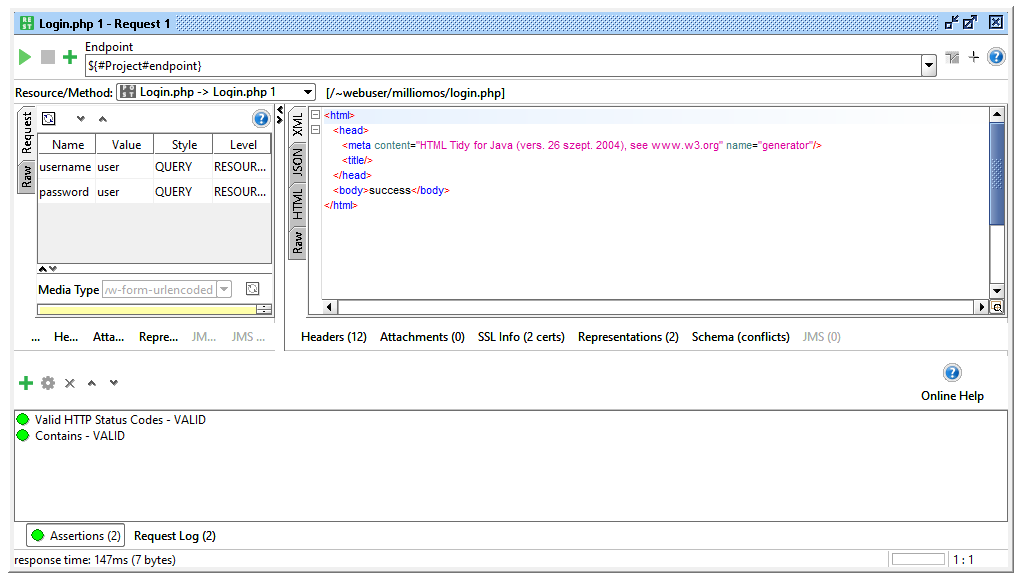
Fejlesztői eszköznek a Visual Stúdió-ra esett a választás, mely támogatja a Cordova projekt létrehozását, elkészít egy alap projektet mely készen is áll a fordításra mindhárom platformra. Ugyanakkor ez egy statikusan megjelenő oldal de a számunkra fontos már említett mappaszerkezetet legenerálja.

A Visual stúdió segítségével lehet fejleszteni a JS,CSS és HTML típusú fájlokat. Ennek segítségével a fordításra kerülő rész teljes egészében elkészíthető.



. ábra A létrehozás folyamata

## SoapUI

Ezen program a háttérrendszer tesztelési folyamatát könnyítette meg a csapat számára. Beküldhető vele a szerverre egy kérés melyre a szerver válaszol tudjuk vizsgálni, hogy a kapott válasz megfele-e a várt formátumnak, tartalmilag annyi és olyan információt kapunk-e vissza amit vártunk. 

4. ábra Sikeres belépés

## A játék, és szabályai

A Legyen ön is milliomos egy televíziós kvízjáték [[3]](#footnote-3). A játék célja, hogy minél több kérdésre válaszoljon a játékos helyesen. A játékos nyereménye annál nagyobb, minél több kérdésre válaszol helyesen jól. A kérdésekre 4 válasz lehetőséget kap a játékos, ezek közül csak egy helyes válasz van. A játékos rendelkezik segítséggel is, amiket csak egyszer használhat egy játék során. A segítség viszont nem garantáltan, de nagy valószínűséggel adnak helyes válasz, ilyenre példa a közönség szavazata.

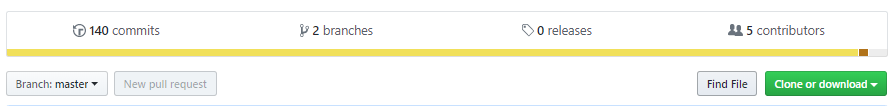
## A project

Az applikáció egyéniségét a téma adja, mivel az egyetemmel kapcsolatos kérdéseket ad a játékosnak. Például Számítógépek és Távközlési hálózatok kérdésekre kell válaszolnia a játékosnak, a következő kérdésben pedig már más kurzusról kap kérdést. Az applikáció elsősorban telefonkészüléken fut majd, de internetes böngészőn keresztül is lehet vele játszani. A játékos ugyanazt az élményt kapja, minden platformon.

## Git

A Git egy változást irányító rendszer, az applikáció fejlesztésre ajánlott, mivel nem csak egy ember dolgozik az applikáción. A Git segítségével egy internetes tárolón van a project, erről a tárolóról le lehet menteni, és fel lehet tölteni a változtatásainkat, amit program kódján vagy különböző részein hajtottunk végre.

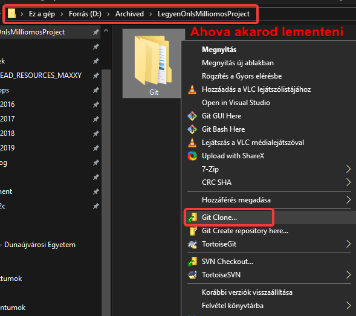
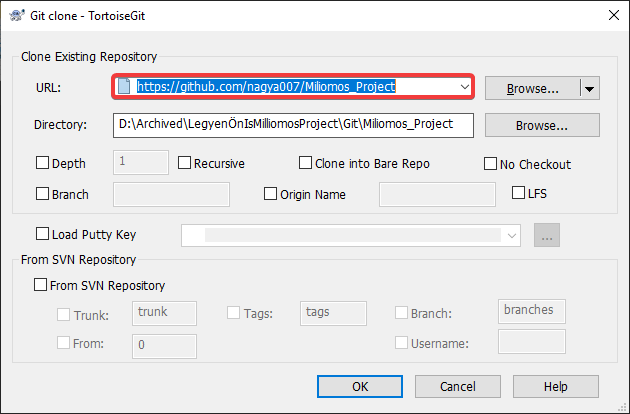
Az irányítás alatt pedig arról beszélünk, hogy minden változást, amit a fejlesztők végre hajtanak, azt menti az internetes tároló. Így lehetőség van nyomon követni változásokat, esetleg visszavonni változásokat.



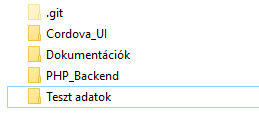
5. ábra A Git követi a válltozásokat

* 1. Használata a Legyen ön is Milliomos projectben

A Git-et lehet használni a hozzávaló terminállal, vagy *TortoiseGit* [[4]](#footnote-4)használatával is. TortoiseGit-el Windows alapú GUI-t kapunk a Git használatához. Emellett rendelkezni kell egy Git felhasználóval is, aminek van engedélye a project letöltéséhez. A project tagok az e-mailjük segítségével regisztráltak a Git honlapjára.[[5]](#footnote-5)



A távoli tárolóról letöltött projectet ezután lehet már használni is, ha pedig szeretnénk feltölteni módosítást a tárolóra, akkor a Git push funkcióját kell használni. Ha pedig mások módosításait akarjunk letölteni, akkor a Git pull funkcióját kell használni. Ajánlott letölteni minden módosítást, még mielőtt mi akarunk módosítani rajta. A TortoiseGit lehetővé teszi ennek a folyamatnak a felgyorsítását, mivel egy GUI[[6]](#footnote-6)-t ad a Git szoftvernek. Így elég gomb kattintással fel és letölteni a project filekat.

A jól szeparáltság érdekében, és a közös munka megkönnyítése érdekében külön mappaszerkezet került kialakításra a backend és a frontend tárolására, ezek mellet továbbá dokumentációknak szintén egy külön mappa lett kialakítva a verziókezelő rendszerben a teszt szintén egy saját mappát kapott a projekt előre haladásával. 

6. ábra Projekt mappaszerkezet

## Dokumentálások

Project folyamata során bővített dokumentumok. A beszélgetések, összejöveteleket egy eseménynaplóba rögzítve lettek, Excel formátumba. A beszélgetések rögzítésével újabb teljesítendő célokat lehetett megszabni, ennek segítségével mindig van egy elérhető dokumentum, ami tartalmazza a teendőket, és korábbi felmerült kérdéseket. A személyes összejövetelek és az online beszélgetés rendkívül segítette az applikáció gyors fejlődését, és a dokumentumok is ezt tükrözik. A dokumentálás része volt a fontosabb hivatkozások mentése, például képek vagy linkek. Az így mentett adatok nem vesznek el soha, mivel a Git verzió kezelés is elősegítette a dokumentumok bővítését.

# Frontend

1. A frontend felépítése
   1. Mi az a frontend?

A frontend alatt a program megjelenítéséről beszélünk, ami a szerverről lekérdezett (backend) adatokat jeleníti meg a felhasználó számára. Megjelenés mellett különböző műveleteket és számolást is végre kell hajtani a frontend felületnek, annak érdekébe, hogy a felhasználó feltudja dolgozni a kapott adatokat. Ez alatt lehet érteni számok összeadását, kivonását, és ezeknek a számoknak az ábrázolását a megfelelő helyen a képernyőn.

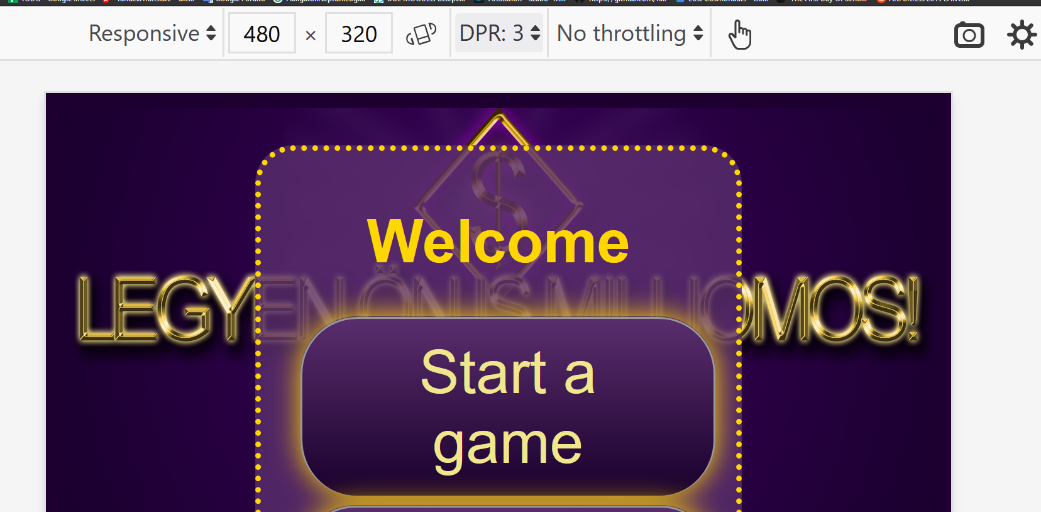
A HTML lehetővé teszi az adatok csoportosítását, megjelenítését megfelelő helyen, és egyéb formai dolgok, például színezés és méretezés. Emellett beépített funkciók és eszközöket is lehet használni vele, például gombok vagy szövegbevitelre mezőt, ahova képes a felhasználó írni. Ezeket az úgynevezett „objektumokat” személyre lehet szabni, és rengeteg tulajdonságot hozzájuk tudunk rendelni, például a felhasználó jelszavát a program csak csillagokkal jeleníti meg, vagy csak bizonyos hosszúságú jelszót írhatunk be. Ilyenkor fontos figyelembe venni, hogy a frontend képes megakadályozni érvénytelen adatoknak a bevitelét,

* 1. Frontend a Legyen ön is Milliomos applikációban



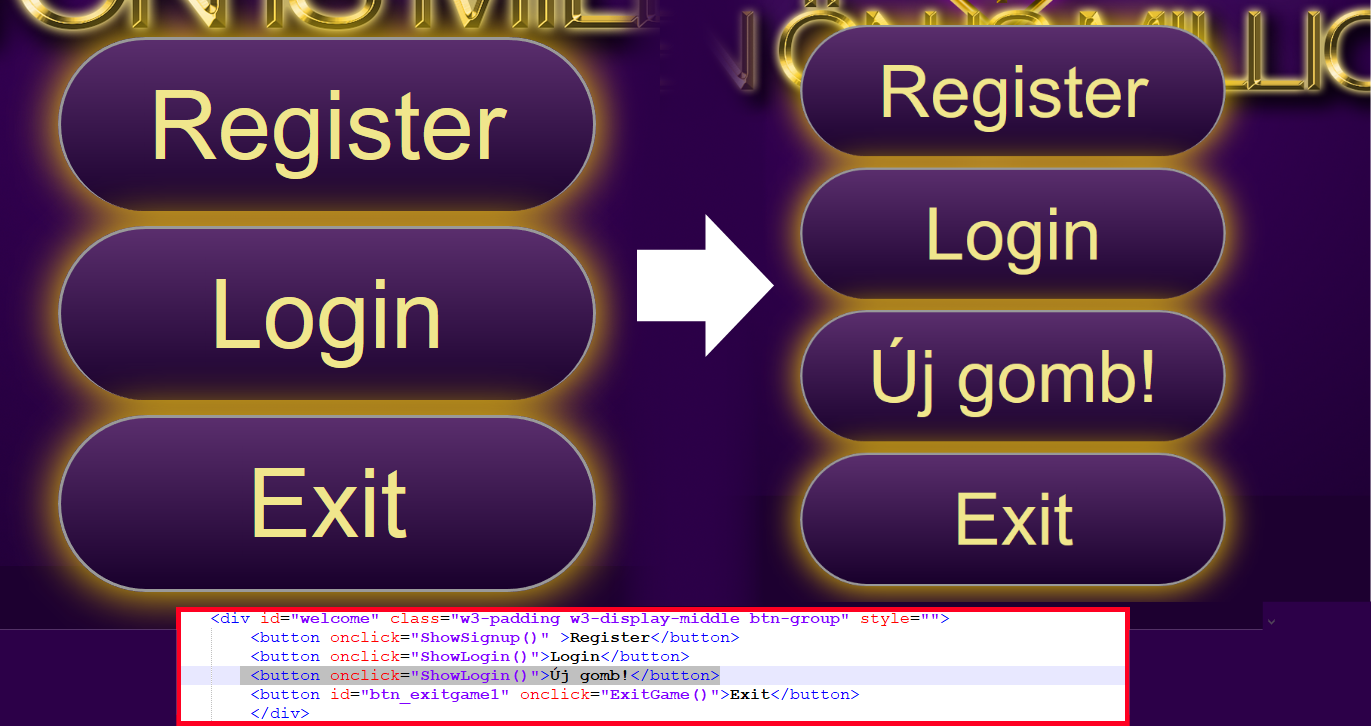
A frontend fejlesztését lehetővé teszi a Cordova, mivel a HTML web programozási nyelvet használja. A Cordova egy futó web applikáció, ami képes a telefonos verzió kezelésére. A Cordova képes mindenféle funkciót kihasználni a mobilokból, ezalatt a billentyűzetet, kamerát, GPS és forgatást képes kezelni. A Legyen ön is Milliomos applikáció esetében a billentyűzet és megjelenésen van a hangsúly.

A fejlesztéshez szükség van egy IDE [[7]](#footnote-7)szoftver letöltéséhez. A legyen ön is milliomos project esetében a Visual Studio 2015 Community Edition-t használta a csoport, és ingyenes is. A Visual Studio emellett nagyon hasznos lesz a „valós idejű” tesztelésre is. A HTML kódot viszont lehet bármilyen más szövegszerkesztővel is használni, például Notepad++, ami egy ingyenes és programozók számára ajánlott szöveg szerkesztő.[[8]](#footnote-8)



A legnagyobb probléma, ami felmerült a fejlesztés során az a megfelelő arányok eltalálása, mivel az applikáció nem csak számítógépen, de telefonon is használható. A képernyő arányához kell viszonyulnia a képernyőnek, és a rajta lévő objektumoknak, például a gombok és szöveg dobozok. Az arányok mellett a HTML korlátozásai is problémát jelenthet, mivel nem minden web böngésző támogatja az összes funkciót. [[9]](#footnote-9)

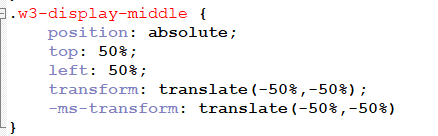
A megfelelő arányokat a HTML képes betartani, ha az objektumokat a képernyőhöz viszonyítva számoljuk. Ezt úgy lehet elérni, hogy folyamatosan rendezzük a csoportokat, például a képernyő közepére, szélére, legaljára rendezzük az objektumokat. Innentől fogva az applikáció minden platformon megfelelő méretezéssel jelenik meg. Ez viszont rengeteg tesztelést igényel, mivel az applikációt folyamatosan bővíteni szeretnénk. Szerencsére a rendezés elősegíti a bővíthetőséget, mivel képesek vagyunk rendezni az elemeket tetszés szerint, a meglévő elemek befolyásolása nélkül. Például, ha új gombot szeretnénk berakni a főmenübe, akkor elég egy új gomb objektumot felvenni a HTML file-ba, és utána az megfelelően rendeződik.



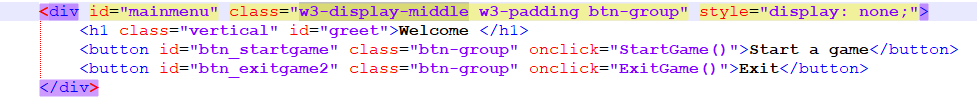
*Új gomb a HTML kódba, megfelelő rendezéssel, csak egy darab új sor kellett*

* 1. Formázás

A HTML és Cordova használata emiatt előnyös, mivel megfelelő felépítés lehetővé teszi az applikáció gyors fejlesztését és tesztelését, egyszerre teszi lehetővé a webes verzió és a telefonos verzió fejlesztését. A megfelelő formázást a HTML által használt .CSS file tartalmazza, itt található különböző rendezés típusok, amikre hivatkozhatunk a HTML-en belül. Ezt a .CSS file-t egy másik mappába lett téve jobb rendezés érdekében. Kifejezetten hasznos a képernyő alapú rendezés, ami %-osan rendezi a tartalmát a HTML elemeknek. Ezek fel vannak osztva több típusú rendezésre, például felfelé rendezés, lefelé, jobbra és balra.[[10]](#footnote-10)



*A középre rendezés tulajdonságai*



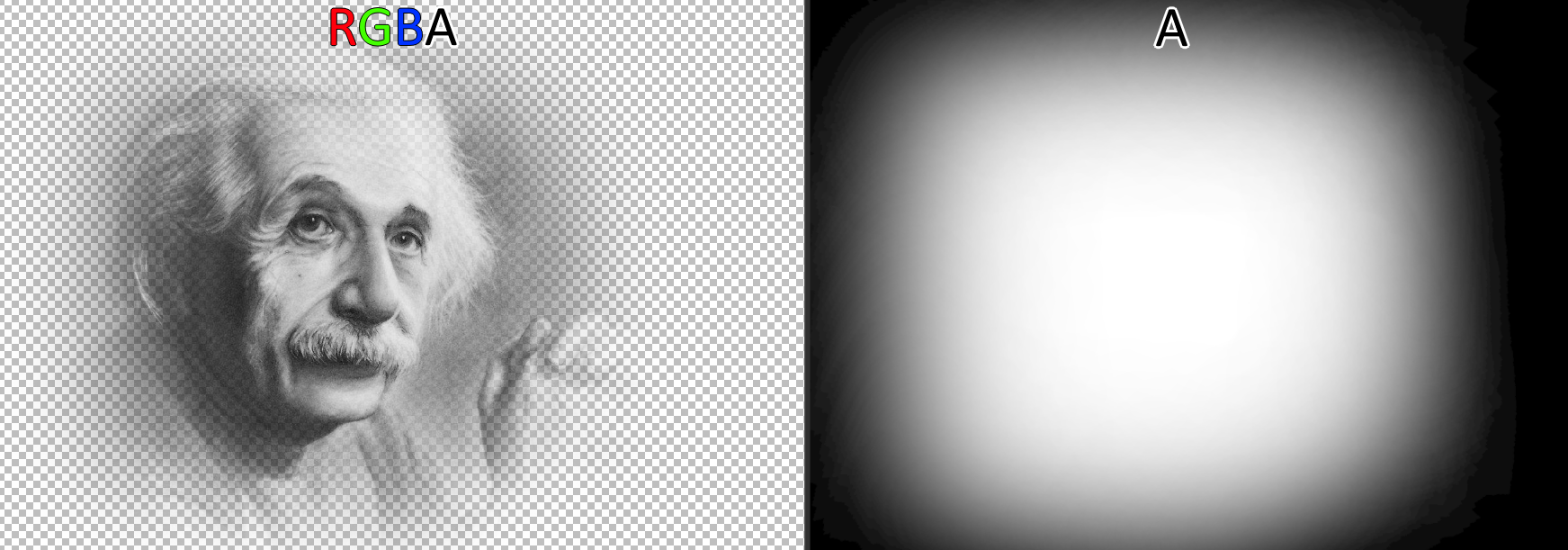
*HTML-en belül, csoport (div) középre van rendezve*

A CSS emellett tartalmazza a többi formai tulajdonságot, például a betűtípus és méretezést, színeket. A CSS lehet akár egy file az interneten is, de ajánlott saját példányt létrehozni, mivel nem biztos, hogy mindig van internet szolgáltatása az applikációnak.

* 1. Formázás, Média

A HTML használatával lehetséges képeket és videókat is importálni, még akár zenét is. Az applikációban használt média elemeket mappába rendezve vannak tárolva, így könnyebb megtalálni a megfelelő file típust. Támogatott file típusok között van: MP4, PNG, WAV. Fontos tudni, hogy ezek a média elemek nem minden böngészőben működnek ugyan úgy, ezért a HTML kódban meg kell szabni, hogy böngészőtől függően hogyan működjön.

A PNG file típus kifejezetten fontos, mivel lehetővé teszi a PNG negyedik szín csatornájának a használatát: Alpha. Az alpha alatt az áttetszőségről beszélünk, a HTML az áttetszőséget a háttérrel való egybeolvasztást teszi lehetővé. Például szeretnénk, hogy a kép a pereme körül átmetszőbb legyen.

 [[11]](#footnote-11)

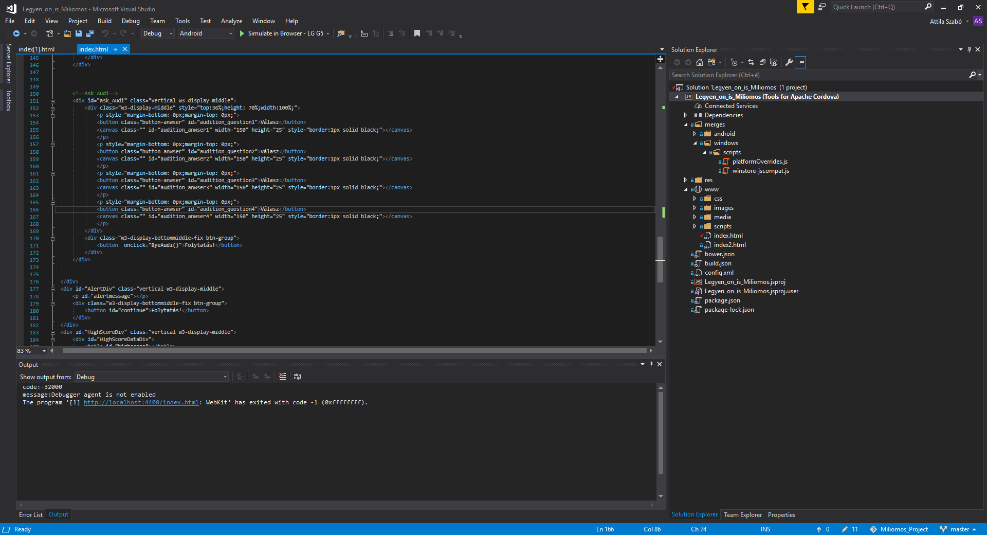
*Alpha Fekete rész = Áttetsző, összeolvad a weboldal színével*

Ez az effektus csak a PNG, vagy TGA file típussal lehetséges, mivel ezek a típusok használnak csak Alpha színe. A hátránya az, hogy több helyet foglal el. Előnye viszont, hogy bármikor, ha változtatjuk a weboldal témáját, az ilyen elemek folyamatosan jól fognak illeszkedni a weboldalra, mivel egyéni áttetszésük van. Ha viszont olyan képformátumra van szükségünk, ami viszonylag jó kép minőséget ad, és file méretet, akkor ajánlott a JPG.

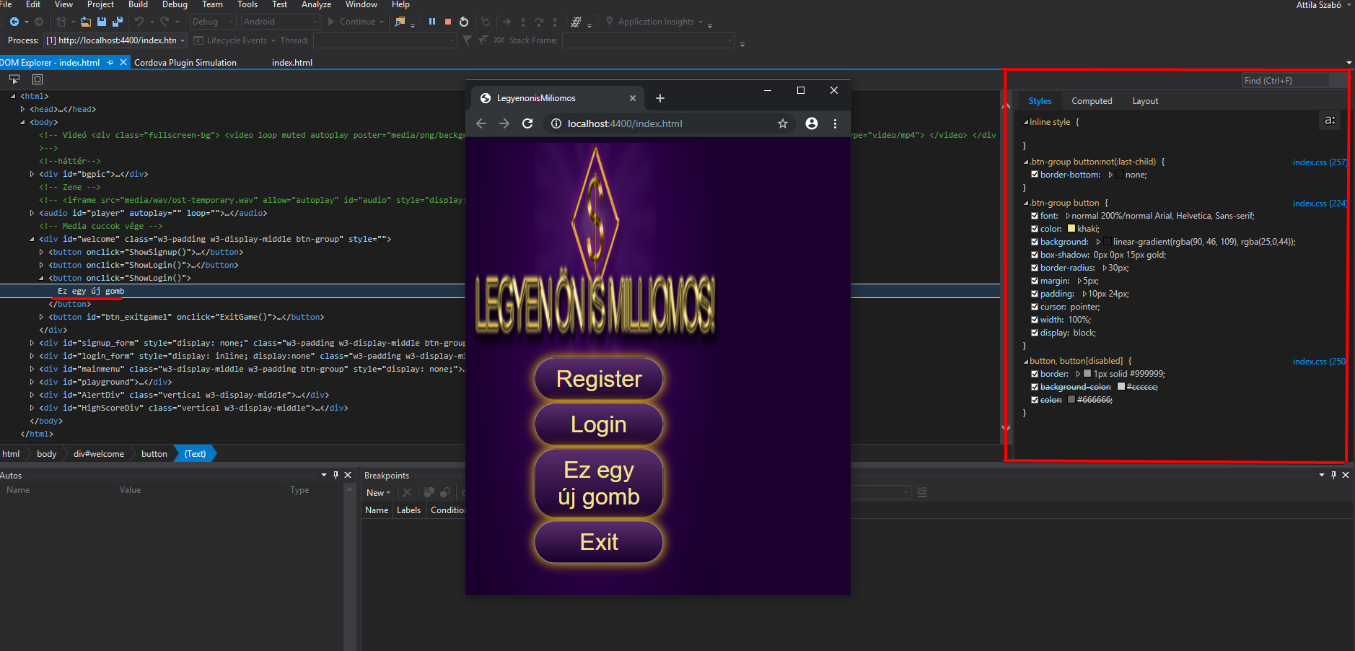
1. Frontend tesztelés

Mivel az applikáció folyamatosan bővül, ezért állandóan tesztelni kell. A tesztelés során a program használata során előforduló hibákat kell megkeresni, általában új funkciók esetében korábban működő részek is elromolhatnak. A tesztelés emellett jó a még félig kész funkciók tesztelésére is gyakorlatban. Mivel webfejlesztésről van szó, ezért gyorsabb a White Box Testing metódus, ami röviden azt jelenti, hogy mi közvetlenül a weboldal fejlesztés során, a metódusok meghívásával kapunk visszajelzést a program működéséről. Így gyorsan tesztelhető az alkalmazás működése, mivel egyből kapunk visszajelzést a működésről[[12]](#footnote-12). A tesztelésre több módszer is lehetséges:

* 1. Visual Studio

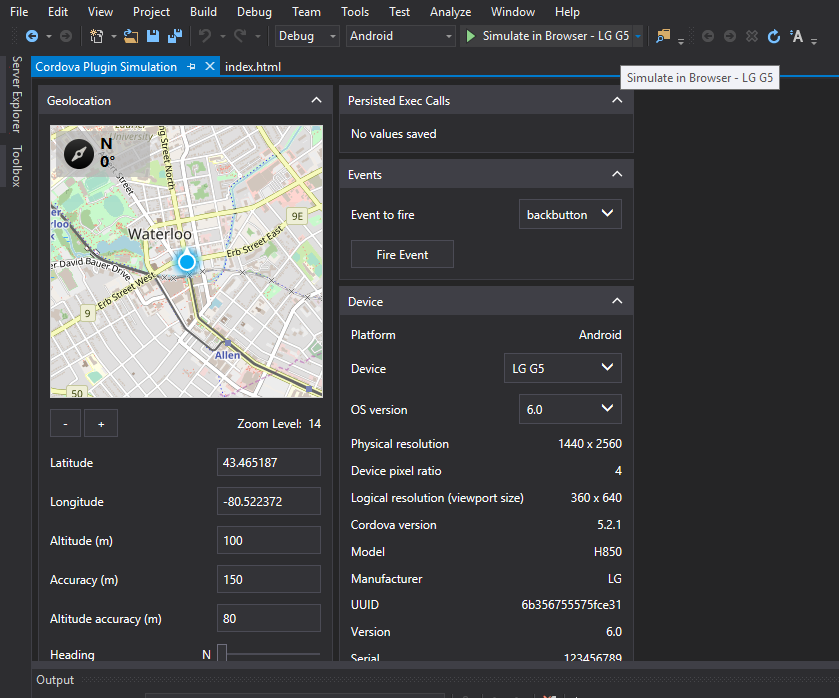


*A Visual Studio file, miután letöltöttük Gitről*



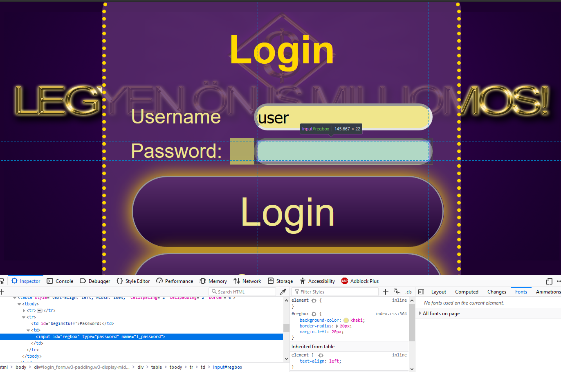
*Valós idejű tesztelésre példa, új sorok egyből megjelennek az applikációban, Visual Studio sajna nem engedi a virtuális képernyő forgatását*

Az applikációt a Visual Studio használatával lehet akár egy telefonos készülékre telepíteni is. A telefonon előbb aktiválni kell a fejlesztői opciókat, így adunk engedélyt a számítógépnek, hogy a telefonra közvetlenül telepítsen alkalmazásokat. A telepítés mellett több lehetőségünk is van, például a debugolás. A debug alatt azt értjük, hogy számítógépünk és a telefonunk össze van kötve, és a kódot látjuk hogyan hívódig meg, még akár változtathatunk is a kódon valós időben. Ez rendkívül hasznos tud lenni, mivel nem kell újra megnyitni az alkalmazást minden alkalommal, amikor változik a kód. Az emuláció [[13]](#footnote-13)emellett rengeteg lehetőségünk is van, akár Android, vagy IOS rendszeren is tudunk emulálni. De ha szimulálni akarunk egy mobil készüléket, akkor annak a beállításainál is lehetőség van az operációs rendszer, és egyéb mobilos paraméterek beállítására, például, hogy mekkora a képarány. Így könnyű tesztelni az alkalmazást, anélkül, hogy vásárolnánk tesztelésre készüléket.



*Android Operációs rendszer beállítása, vagy képernyő arány*

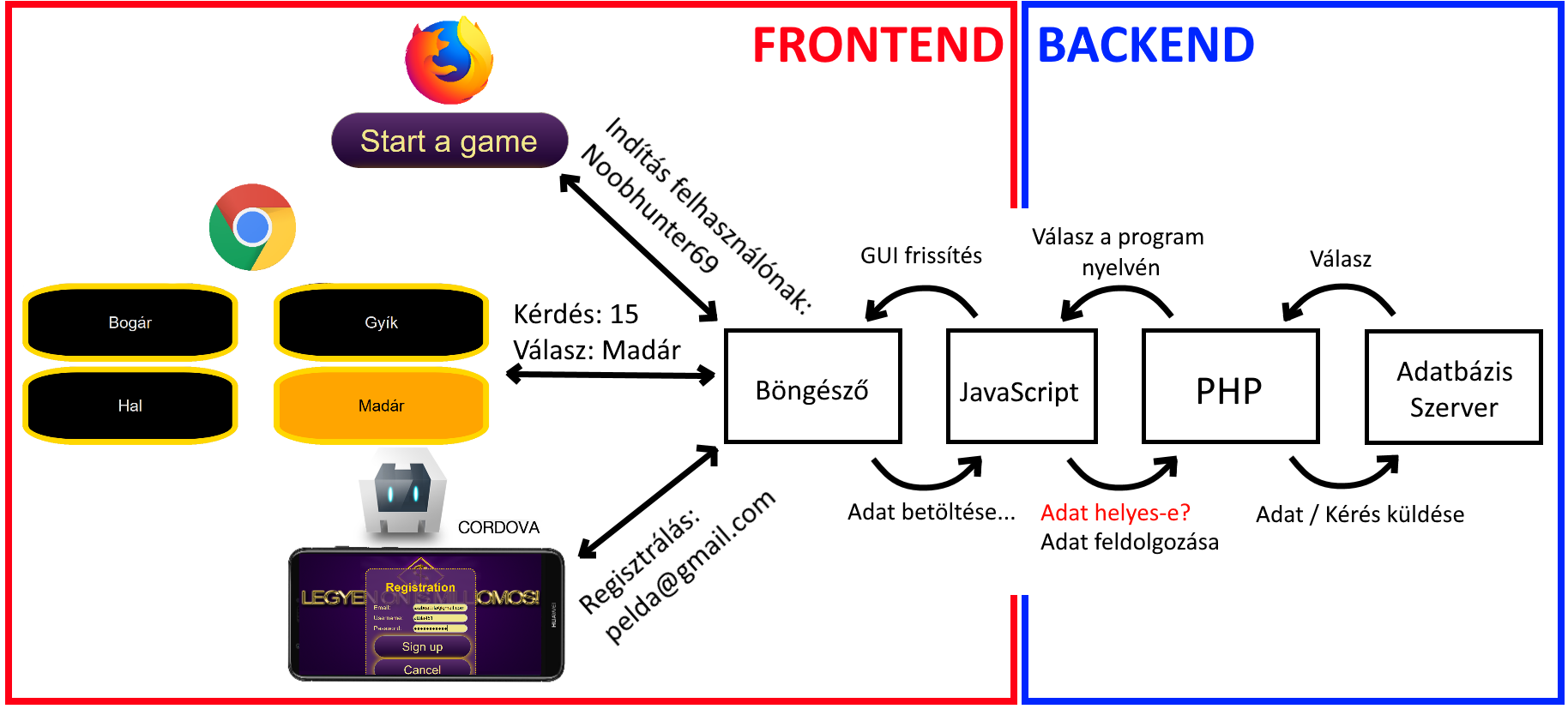
* 1. Böngésző fejlesztői opció



*Firefox böngésző használatával valós idejű szerkesztés*

Lehetséges az applikáció tesztelése Visual Studio nélkül is. A böngészők általában rendelkeznek fejlesztői eszközökkel, ha mi megnyitjuk a HTML file-t, amin dolgoztunk, akkor lehetőségünk van előre megtekinteni a weblap működését. A HTML file megnyitása után a weblap bezárásáig lehetőségünk van bármilyen módosításra. Fontos tudni viszont, hogy ezek a módosítások csak a böngésző példányán jelennek meg, ezeket el kell menteni a HTML file-ba is. A fejlesztői opciók is rendelkezhetnek Androidos, vagy IOS „emulátorral”, ami viszont csak azt csinálja, hogy a weblap telefonos verzióját tölti be, míg a képernyő arányt nem változtatja.

# Fejlesztés



1. Fejlesztés menete

Az alkalmazás bővítésének legfontosabb része a számolások helyes elvégzése, és kiírása. Ennek a fejlesztési rétegnek jól kell kommunikálnia az alkalmazás backend részével. Egy adatbázisba csak helyes értékeket szabad felvinni, különben hibába ütközünk. Ha pedig a felhasználó nem megfelelő értékeket visz fel, akkor azt ki kell jelezni is. Ez nem különbözteti meg azt, hogy a backend ettől függetlenül tartalmazhat hibakezelést, de ezt természetesen tudatni kell a felhasználóval. A fejlesztésnek folyamatosan figyelembe kell vennie a platform függőséget is, például Firefox, Chrome, és telefonra a Cordova.

A programra használt navigálás és műveletek inicializálását a HTML és a JavaScript végzi el. A HTML-ben szereplő objektumok, például gombok által kerülnek meghívásra különböző műveletek.

Ezek között a műveletek között van az adatbázisból való lekérdezés is. Ezekkel az adatokkal pedig úgy kell bánni, hogy azok felhasználhatóak legyenek a megjelenítésre. A feldolgozott adatokat pedig a backend részére, a szerver adatbázisára küldjük további feldolgozásra. Az innen érkező válasz alapján pedig folytatódik a program élet ciklusa. Tehát fontos feltételezni, hogy ezek között a pontok között bármikor lehetnek hibák, és ezeket kezelni kell. Ezek közé a hibajelzések közé sorolható a szövegdobozos hibaüzenetek, amik visszajelzést adnak arról, hogy nem megfelelő adatot vitt fel a felhasználó, vagy ha esetleg éppen nem elérhető a szerver.

1. Backend fejlesztés

Elsősorban az applikáció „agya” kell, ahol tárolni lehet a játékos teljesítményét, és bővíteni lehet a játék során előforduló kérdéseket. Ezt megelőzi az adatbázis tervezése. Fontos tudni, hogy milyen adatokkal akarunk dolgozni, később ezeket kiegészíteni már nehezebb. A szerveroldali kommunikációnak emellett kezelnie kell a helytelen adatokat, máskülönben a felhasználók visszaélhetnek a program hibájával.

1. Frontend fejlesztés

Egy működő alappal már lehet fejleszteni a frontendet. Itt jön figyelembe a tesztelés hangsúlya, mivel van lehetőség arra, hogy a kód futtatása közben fejlesszük a felhasználói felületet. Az adatbázist ilyenkor nem kell már szerkeszteni, később esetleg lehet bővíteni, de több lehetőségünk van a frontenden hasznosítani ezeket. Ugyan azok az adatok több helyen is előfordulnak, és többször lehet újra hasznosítani őket. Ilyenre példa a felhasználó nevének kiírása, vagy a „kérdések” lekérdezése.

Irodalomjegyzék

[A felhasznált szakirodalmat []-ben sorszámozottan – a szerzők szerint ABC sorrendben, egyértelműen azonosíthatóan (szerző, cím, kiadó, kiadás helye, kiadás éve) – kell feltüntetni.]

{Ábrajegyzék/Táblázatok jegyzéke/Mellékletek jegyzéke/stb.}

[A dolgozat ábrái/táblázatai/mellékletei/stb. felsorolva típusonként, ha vannak. Törekedni kell a szövegszerkesztő program által automatikusan generált jegyzék használatára! Elvárt, hogy minden kép/ábra/táblázat/egyenlet/stb. kapjon sorszámot (típusonként külön számozva), amelyre a dolgozat szövegében kereszthivatkozással történjen a hivatkozás!]

Mellékletek

[A dolgozat mellékletei sorszámozottan, ha vannak. A Mellékletek jegyzéke hivatkozzon az adott mellékletre!]

1. https://tudasbazis.sulinet.hu/hu/szakkepzes/informatika/a-program-gyakorlati-megvalositasa/szamitogepes-programozas-a-gyakorlatban/a-vizualis-fejlesztoeszkozok-megismerese-hasznalata [↑](#footnote-ref-1)
2. https://cordova.apache.org/ [↑](#footnote-ref-2)
3. https://hu.wikipedia.org/wiki/Legyen\_%C3%96n\_is\_milliomos! [↑](#footnote-ref-3)
4. https://tortoisegit.org/ [↑](#footnote-ref-4)
5. https://github.com/ [↑](#footnote-ref-5)
6. Graphical user interface [↑](#footnote-ref-6)
7. Integrated Development Environment [↑](#footnote-ref-7)
8. https://notepad-plus-plus.org/ [↑](#footnote-ref-8)
9. https://www.w3schools.com/cssref/css3\_browsersupport.asp [↑](#footnote-ref-9)
10. https://www.w3schools.com/w3css/w3css\_display.asp [↑](#footnote-ref-10)
11. https://opimedia.azureedge.net/-/media/images/utr/editorial/articles/online-articles/2014/04-01/albert-einstein-the-humanitarian/albert-einstein-jpg.jpg [↑](#footnote-ref-11)
12. http://softwaretestingfundamentals.com/white-box-testing/ [↑](#footnote-ref-12)
13. Egy másik készülék használatát utánozza [↑](#footnote-ref-13)