

BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

Vizsgaremek dokumentáció

Kvíz játék – Tentacles Teams

A csoport tagjai:

Csongrádi Róbert

Szuper András

Készítette:

Csongrádi Róbert

Tartalom

1	Bevezetés	3
1.1	Feladat leírás	3
1.2	A felhasznált ismeretek	3
1.3	A felhasznált szoftverek	4
2	Felhasználói dokumentáció	5
2.1	A program általános specifikációja	5
2.2	Rendszerkövetelmények	5
2.2.1	Hardver követelmények	5
2.2.2	Szoftver követelmények	6
2.3	A program telepítése	6
2.4	A program használatának a részletes leírása	7
3	Fejlesztői dokumentáció	11
3.1	Az alkalmazott fejlesztői eszközök	11
3.2	Adatmodell leírása	11
3.3	Részletes feladatspecifikáció, algoritmusok	14
3.4	Tesztelési dokumentáció	18
4	Összefoglalás	22
4.1	Önértékelés	22
4.2	Továbbfejlesztési lehetőségek	23
5	Felhasznált irodalom	24
6	Ábrajegyzék	25

1 Bevezetés

1.1 Feladat leírás

Témaválasztásnál az újszerűség, használhatóság és a kódolási-technikai játékosság bemutatása is vezetett minket. Igyekeztünk több szempontból képet adni a jártasságunkról egyes nyelvek-rendszerek területén, stílusban is változatosan. Az alapokat szeretnénk volna felhasználni. Kerültük a félkész megoldások használatát, ezért igyekeztünk az erősen fejlesztőbarát keretrendszereket és tartalomkezelőket mellőzni. Ezen irányelvek mellett maradtunk egy egyszerű játék bemutatásánál. Valójában nem is az alkalmazásra (játékra) akartuk helyezni a fő hangsúlyt, hanem a kezelésének a körülményeire inkább. Értem ezalatt: egy bevezető és kedvcsináló lap; egy megjelenésében újszerű, az UX/UI irányelveket szem előtt tartó tartalmi főoldal; és az egészet szerves egységként egybefoglaló jogosultság-kezelő és felhasználó nyilvántartó-beléptető rendszer bemutatása voltak a nem titkolt rész-célok részünkről. Andrással a fenti irányelvek mellett vágtunk bele a Kvíz játék elkészítésébe.

1.2 A felhasznált ismeretek

Frontend részen a HTML 5 és CSS 3 alapokkal dolgoztunk. Emellé a Bootstrap 5 keretrendszert és az AOS plugint használtuk főként. Egyedi megoldásokhoz (Intro oldal, maga a játék oldal) a JavaScript, JQuery volt segítségünkre.

Backend főként PHP és picikét Ajax alkalmazásával valósult meg.

Adatbázisunk MySQL alapokon nyugszik. Bérelt domain-en, saját tárterületen fut. Hostunk Apache szerveren, webes kliensként a phpMyAdmin (v. 4.9.7.) használatával, MariaDB-n fut.

Teljes reszponzivitás mellett a 'Mobil First' alapelvek figyelembevételével készítettük el a webalkalmazásunkat. iOS és Android rendszereken, különböző hardver-konfigurációkon teszteltük a működését. Jellemzően az Apple termékek stabilabb futtatást, de némi típushibát produkáltak; míg az Android operációs rendszert használó mobil gépek, közepes teljesítményű hardver szinttől meggyőző működési teljesítményt, hibamentes futást produkáltak. A belépő szintű androidos eszközök esetén több esetben 'laggos' megjelenés és működési anomáliák történtek.

1.3 A felhasznált szoftverek

Kódkészítés, tesztelés, dokumentálás folyamán a következő szoftvereket használtuk fel:

Alapvetően **Windows** operációs rendszert használtunk a fejlesztői pc-ken. (**Win 10**).

Visual Studio Code szerkesztővel és **XAMPP** webszerver-szoftvercsomaggal dolgoztunk (az Apache webszerver, a MariaDB adatbázis-kezelő helyi gépen való szimulálásához). **Git** verziókezelő szoftverrel oldottuk meg a párhuzamos fejlesztést és a publikálást.

2 Felhasználói dokumentáció

2.1 A program általános specifikációja

Klasszikus kvízjáték. Regisztráció, majd Belépés után a játékos 1 percnyi játékidőt kap. Ezalatt igyekezzen minél több kérdésre helyes választ adni. Egy kérdéshez 4 válasz opció tartozik. 1 helyes, 3 helytelen válasszal. Nincs többes-válasz lehetőség.

Az autorizációs rész lehetőséget kínál az elfelejtett jelszó pótlására. A regisztrációkor megadott e-mail címre juttat el az alkalmazás egy random, ideiglenes jelszót. Belépés után bármikor lehetőség van jelszó módosítására is.

Az alkalmazás 3 alapvető részből áll, megjelenését tekintve:

- Kedvcsináló reklámodal (intro oldal), ahol egy alaphangulatot kívántunk létrehozni.
- Főoldal, ahol információkat adunk a játékkal, a készítőkkal, a képzési hellyel kapcsolatban. Kapcsolatfelvételi lehetőségekkel minden szereplő tekintetében.
- Játéoldal, ahol maga a kvíz játék fut.

2.2 Rendszerkövetelmények

Az alábbiakban általános irányelveket adunk meg. Részletesebben a „Tesztelési dokumentáció” fejezetben térnénk ki a személyes tapasztalatainkra.

2.2.1 Hardver követelmények

A minimális rendszerkövetelmények tekintetében az irányadó a JavaScript fejlesztői és működési alapkövetelmény rendszere, amik a következők:

(<https://help.syncfusion.com/js/installation-and-upgrade/system-requirements>)

- Processor: x86 vagy x64
- RAM : 512 MB (minimum), 1 GB (ajánlott)
- HDD: az általános webes böngészéshez szükséges minimum.

2.2.2 Szoftver követelmények

Webböngészők és mobileszközös operációs rendszerek tekintetében az alábbiak javasoltak:

Chrome	Firefox	Opera	Edge	IE	Safari	IOS	Android	Windows Mobile
63 +	58 +	50 +	13 +	11 +	9 +	9 +	4.4 +	IE 11 +

Valamint szükséges az egyes böngészőkben a JavaScript használatának engedélyezése. Chrome böngészőben például itt találjuk ennek a lehetőségét:



1. sz. ábra – JavaScript aktiválása Chrome böngészőben.

A működéshez szükséges, adattáblákat tartalmazó adatbázist 'sql'-exportként mellékeljük. Természetesen a tartalmazott adatok nélkül hiányos a működése is.

2.3 A program telepítése

Nincs szükség telepítésre. Az alkalmazás webes felületen elérhető. Asztali alkalmazás futtatására sem admin, sem felhasználói szempontból nincs szükség.

2.4 A program használatának a részletes leírása

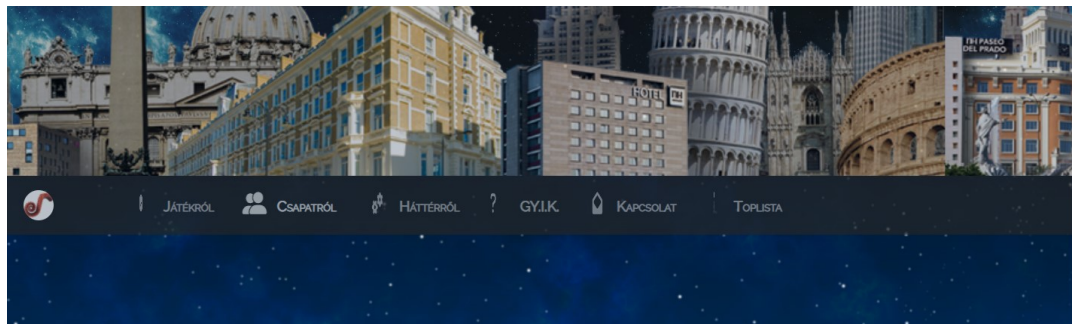
Webes fejlesztésünk a kviz.csongradi.eu URL-en érhető el. Bérelt tárterület, saját domain.

Elsőként egy hangulatkeltő, 'intro' oldal fogadja a látogatót. Itt a jobb alsó sarokban animált 'TOVÁBB' felíratra vagy a felette lévő szintén animált jobbra-nyílra kattintva továbbléphetünk a főoldalra.



2. sz. ábra – Webfejlesztés - Intro oldal képernyőfotója. – jobb alsó sarokban: Tovább

A főoldalon regisztráció és bejelentkezés nélkül általános információk érhetők el. Vagy lefelé gördítéssel, vagy pedig a felső navigációs sorban található ikonokra klikkelve juthatunk el az oldalon belüli tagolt tartalomhoz.



3. sz. ábra – Webfejlesztés – Főoldal – menüsor, ami a tartalmi elemekhez juttat

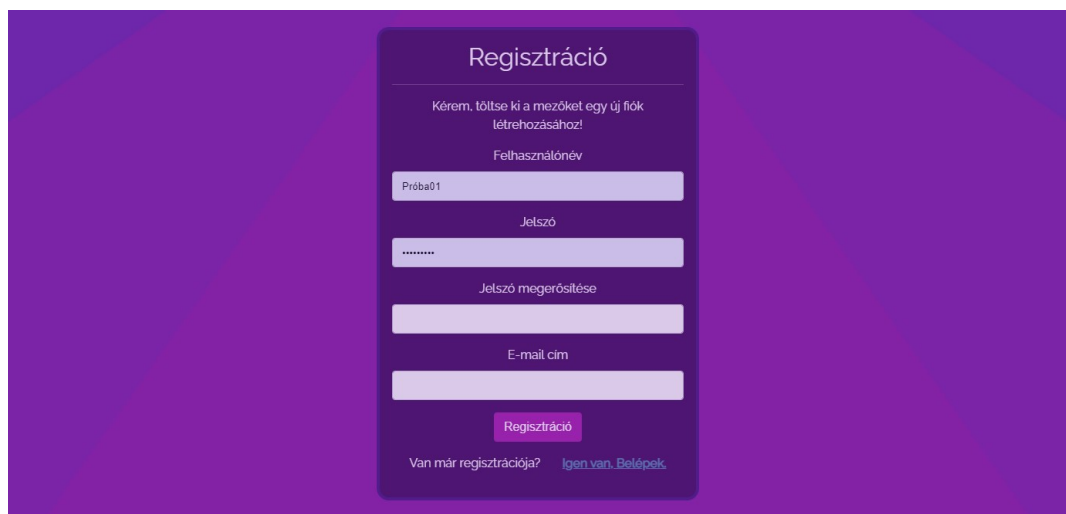
Néhány gyakori kérdésre összefoglaló GY.I.K segíti a látogatókat. A weboldal és az alkalmazás készítőiről flip-flop kártya és onnan direkt link (ami az adott készítő személyes weboldalára irányít át) segítségével nyerhetünk több információt.

Igyekeztünk méltó helyet biztosítani oktatóinknak/tanárainknak. Kifejezve ezzel is elismerésünket feléjük. Kis kártyákon összefoglaljuk iskolán belüli pozíciójukat és oktattott skilljeiket. Animált levélke ikonra klikkelve direkt üzenetet tud a látogató küldeni számukra. Itt található egy direkt link is, ami a szakképző iskola honlapjára mutat.

Regisztrációs folyamatról

Ahhoz, hogy a játékot használni lehessen és a toplistához hozzá lehessen férni, regisztrálni kell. Majd a sikeres regisztráció után az ott megadott adatokkal be kell lépni. Regisztráció során egy tetszőleges felhasználónevet lehet választani, ami minimum 3, maximum 16 karakter hosszú lehet. Tartalmazhat magyar kis- és nagybetűket és számokat is. A jelszó 3-16 karakter hosszú, angol kis- és nagybetűket, valamint számokat is tartalmazhat. E-mail címként érdemes létező és elérhető címet megadni, mert elfelejtett jelszó esetén a program ide küld automatikus üzenetben ideiglenes jelszót.

Jelszavunkat – belépés után – bármikor megváltoztathatjuk. Ha elfelejtettük a jelszavunkat, akkor lehetőségünk van új, ideiglenes jelszó kérésére, amit azonnal elküld a program a megadott e-mail címre egy automatikus üzenetben.



Regisztráció

Kérem, töltsd ki a mezőket egy új fiók létrehozásához!

Felhasználónév

Próba01

Jelszó

Jelszó megerősítése

E-mail cím

Regisztráció

Van már regisztrációja? [Igen van. Belépek.](#)

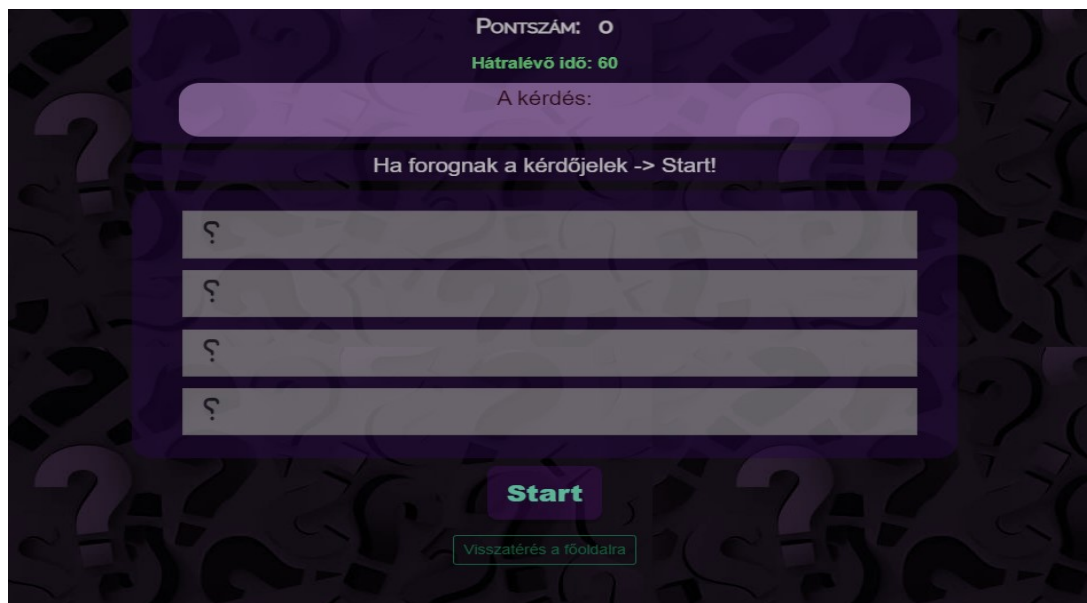
4. sz. ábra – Webfejlesztés – Regisztrációs oldal.

Regisztráció után belépve, lehetőség van az aktuális **Toplista** megtekintésére és megjelenik a beléptetési doboz alatt egy **START** gomb is. Erre kattintva jutunk el a játék oldalára.



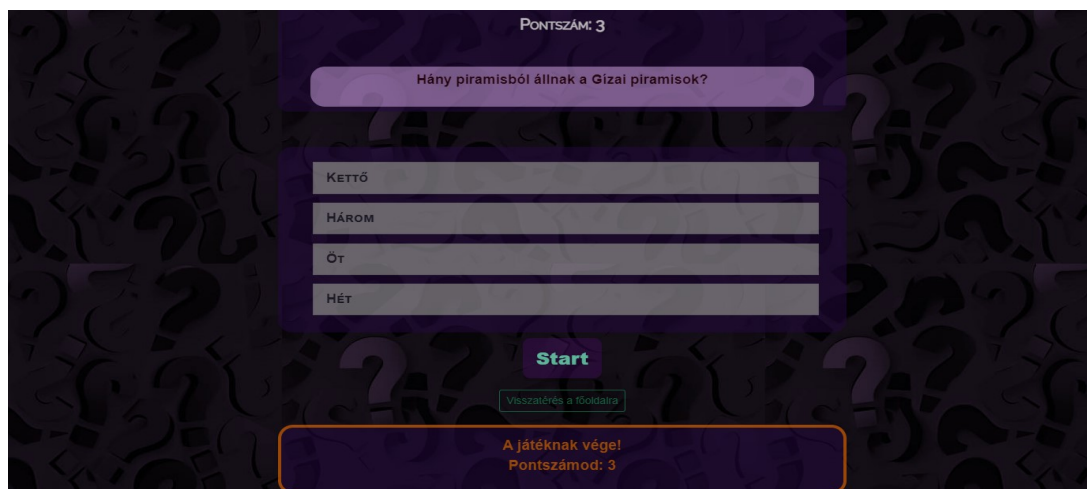
5. sz. ábra – Webfejlesztés – Főoldal – belépés utáni képernyő – Start gomb a játékindításhoz.

Maga a játék klasszikus kvíz játék. A képernyő 3 nagy részre oszlik: információs rész, játékmező és funkcionális lábléc. Az információs részben a Pontszám és a Hátralévő idő látszik. A játékmezőn a kérdés és a 4 lehetséges válasz található játék közben. A funkcionális részen a **Start/Újra** gomb és a Főoldalra való visszatérést adatvesztés nélkül lehetővé tévő gomb található. Itt jelenik meg a játékidő végén az elért eredményről a tájékoztatás is.



6. sz. ábra – Webfejlesztés – Játékoldal – kezdés képernyő – Start gombra klikkelve indul a játék

Játékot indítani a **Start** gombra klikkelve lehet. 60 másodpernyi (1perc) idő van arra, hogy minél több helyes választ adjunk a véletlen sorrendben megjelenő kérdésekre. 1 helyes válasz van minden esetben, tehát a megjelenő 4 válaszlehetőségéből 3 helytelen lesz. Minden helyes válasz megadása 1 pontot jelent. Helytelen válasz megadása esetén nincs pontlevonás. A helyes válaszokat a felső panel bal részén megjelenő **Helyes válasz** ikon, helytelen esetén jobb oldali részen megjelenő **Rossz válasz** ikon felvillanása erősíti meg.



7. sz. ábra – Webfejlesztés – Játékoldal – Játék vége képernyő – elért eredmény kijelzése.

Ha az idő lejár, a játék véget ér. Összegzi az elért eredményt a program és automatikusan feljegyzi egy **Toplistába**. A 10 legjobb eredményt elérő játékosát a Főoldalon a **Toplista** menüpontban meg lehet tekinteni – belépett felhasználóknak.

Üdvözlés **Próba01**

A Kvíz játék aktuális top 10-es listája

Helyezés	Játékosnév	Pontszám
1	Próba01	11
2	Próba01	8
3	Próba01	8
4	Próba01	7
5	Szuper Andras	7
6	Próba01	7
7	Próba01	6
8	nasziteszt	6
9	MrDanielHarka	6
10	MrDanielHarka	5

Visszatérés a főoldalra

8. sz. ábra – Webfejlesztés – Toplista.

3 Fejlesztői dokumentáció

3.1 Az alkalmazott fejlesztői eszközök

Az alkalmazás készítéséhez használt eszközök:

- Visual Studio Code: HTML 5, CSS 3, PHP, MySQL, JS, JQ
- XAMPP: PhpMyAdmin (Apache és MariaDB lokális szimuláció)
- Git, GitHub, GitDesktop
- Favicon - <https://favicon.io/>
- JQuery AJAX - <https://api.jquery.com/jquery.ajax/>
- Font Aqsome 6.1.1. - <https://fontawesome.com/>
- Bootstrap 5 - <https://getbootstrap.com/>
- AOS plugin (Animate On Scroll library) - <https://michalsnik.github.io/aos/>
- Fisher-Yates Shuffle algoritmus -
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle és:
<https://medium.com/javascript-by-doing/learn-fisher-yates-shuffle-in-javascript-275881d36650>

Dokumentáció készítéséhez:

- Win 10 operációs rendszer
- MS Office csomag (Word, Excel, PowerPoint)
- Balsamiq Wireframes
- CorelDraw 12

3.2 Adatmodell leírása

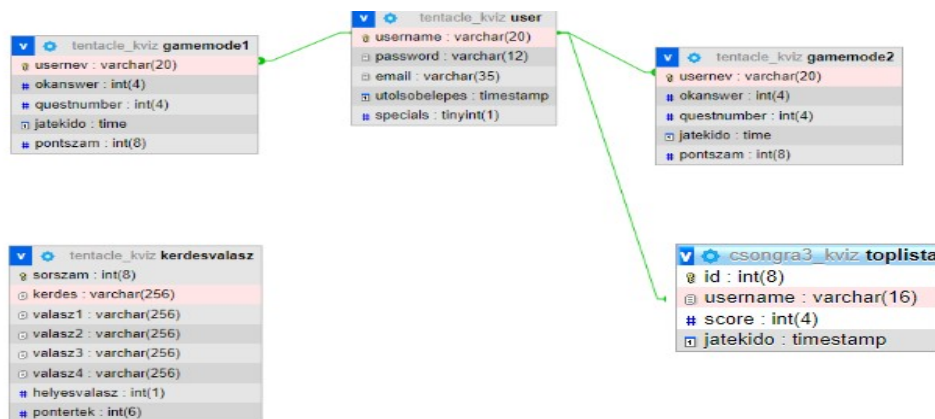
Az alkalmazás fejlesztése során a reprezentálás mellett a teljes értékű működőképesség vezetett minket. Főként az autorizáció (regisztráció, belépés, jelszókezelés, kilépés) folyamatánál és természetesen a kérdés-válaszok tekintetében támaszkodunk adatbázis

műveletekre. Egyértelműen az elért eredményeket (toplista) is táblában tároljuk és hívjuk meg.

Tartalmazza ezt a szót:

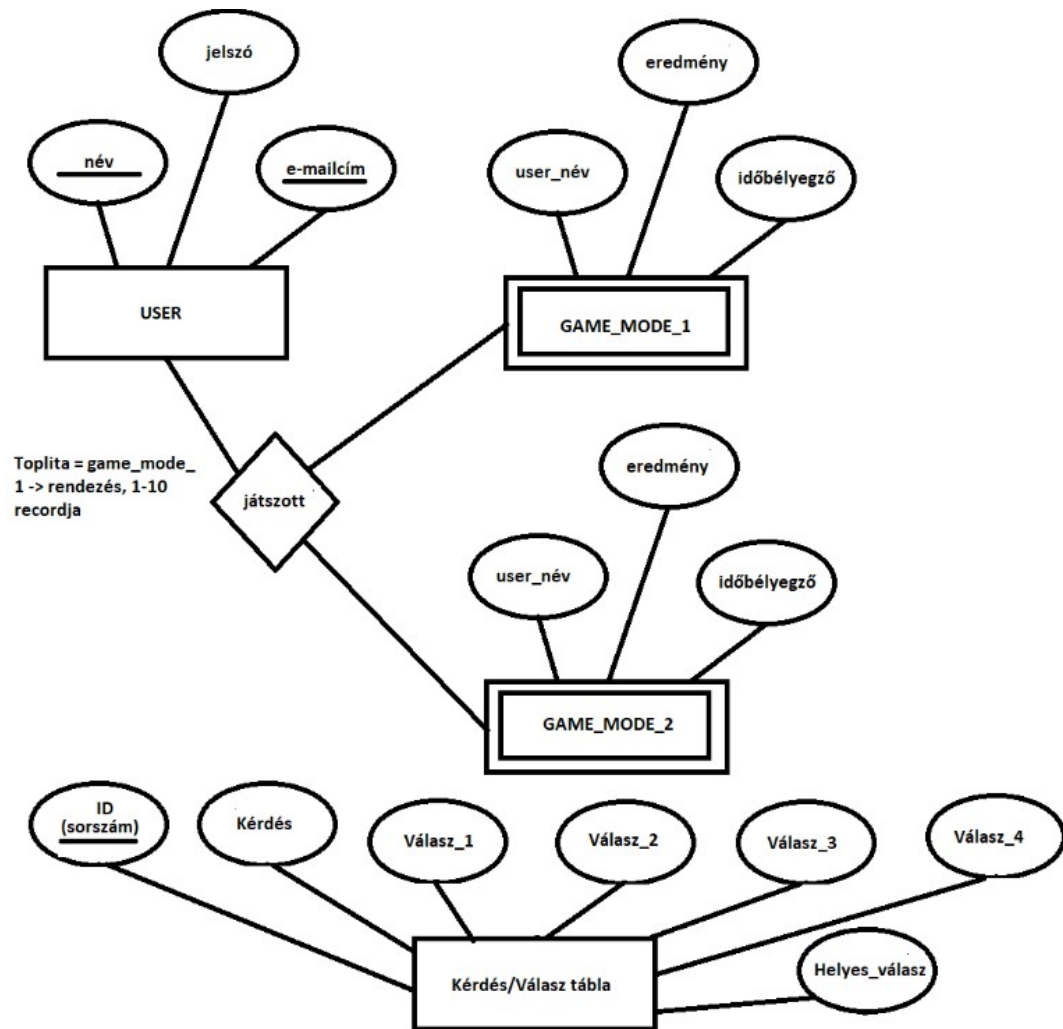
Tábla	Művelet	Sorok	Típus	Illesztés	Méret	Felülírás
<input type="checkbox"/> comment		0	InnoDB	utf8_hungarian_ci	16.0 KB	-
<input type="checkbox"/> gamemode1		0	InnoDB	utf8_hungarian_ci	16.0 KB	-
<input type="checkbox"/> gamemode2		0	InnoDB	utf8_hungarian_ci	16.0 KB	-
<input type="checkbox"/> kerdesvalasz		0	InnoDB	utf8_hungarian_ci	16.0 KB	-
<input type="checkbox"/> toplista		29	InnoDB	latin1_swedish_ci	16.0 KB	-
<input type="checkbox"/> users		11	InnoDB	latin1_swedish_ci	32.0 KB	-
6 tábla	Összesen	40	InnoDB	latin1_swedish_ci	112.0 KB	0 B

9. sz. ábra – Webfejlesztés – Adatbázis szerkezet.



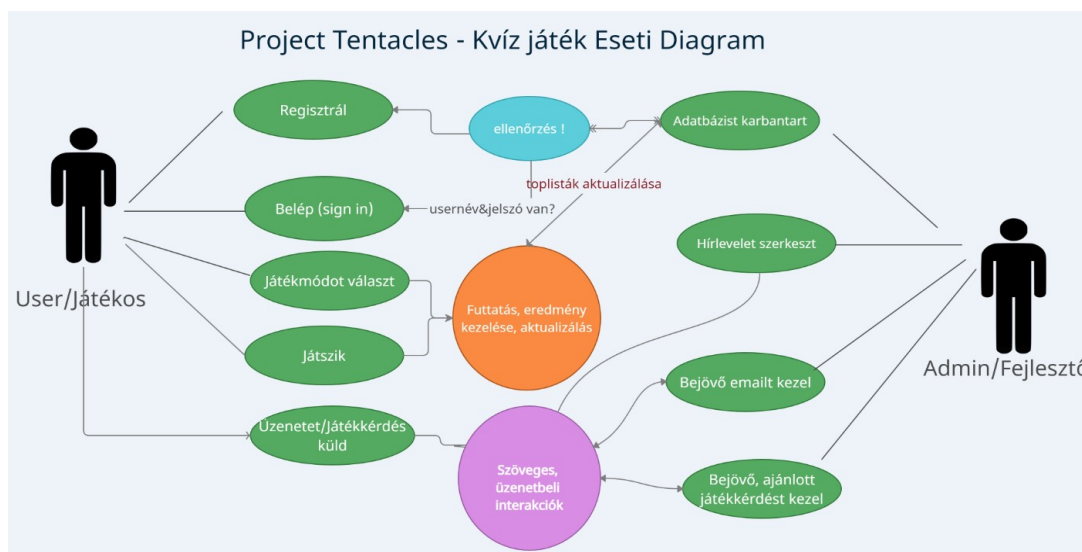
10. sz. ábra – Webfejlesztés – Táblakapcsolat.

A regisztráció folyamán elsődlegesen a usernév a meghatározó. Ez alapján azonosítjuk és kezeljük a felhasználókat. Ennek és a hozzá tartozó titkosított jelszó párosnak a megléte adja a jogosultságot a belépéshez, amit külön Session változóban kezelünk és ez a jogosultságkezelés és a redirectálás alapja. Az alkalmazás és a játék belső dokumentációval rendelkezik. Kódon belüli kommentekkel magyarázzuk mind a működési elvet, vagy az egyes unitok (logikai/működési egységek) funkcióját. Mivel itt egy játékról van szó, ezért fontos szempont volt kódolási szempontból 'elegáns' (minél hatékonyabb és rövidebb) kód megírása is. Az általános működést a következőkben határoztuk meg a fejlesztés kezdetén:



11. sz. ábra – Webfejlesztés – ERD (Entity Relationship Diagram)

3.3 Részletes feladatspecifikáció, algoritmusok



12. sz. ábra – Webfejlesztés – Fejlesztés kezdetén meghatározott UML diagram.

Ahogy jelen dokumentációban már taglaltuk, alapvetően **3 szakaszra** bontottuk a Vizsgamunka webfejlesztésünket.

Első részben egy marketing-jellegű, ráhangoló, kedvcsináló Intro oldalt valósítottunk meg. Itt JavaScript ([intro.js](#) fájlban található a kód) használatával injektálunk be `<div>`-eket, aminek a háttérét egy előre definiált tömbből származtatott véletlen-értéknek megfelelően színezzük be, majd szintén véletlen-érték hívásával pozicionálunk, a `<div>`-ben lévő kérdőjelet szintén fenti elvek alapján színezzük és az egész `<div>`-et egyedi, véletlenszám-alap határértékek szerint pályára állítunk és `<animation>` segítségével mozgatunk illetve elforgatunk. Az előre definiált labda-mennyiségnek (változó értéke) megfelelően egy for-ciklussal eljárással totalizáljuk az inzertálást (balls tömb létrehozásával) és a teljes labdamennyiséget tartalmazó tömböt (balls tömb) egy `ForEach` használatával járjuk be teljesen és ezzel határozzuk meg az egyedi animációt is. Az áttűnéseket a Z-index és az egyedi `{opacity}`-érték alapján jelenítjük meg. A 'KVÍZ' feliratot kis késleltetéssel 'világítjuk' ki, majd egy aszimmetrikus sor (látens ciklus) oda-vissza bejárásával szimulálunk egyfajta nem-szabályos sorrendet, ami véletlen-villódzást okoz. A TOVÁBB gomb egy sorrendjében megjelenített és 'nth-child'-pszeudóval irányított ciklikus színfutást kapott ([intro.css](#)), ami megjelenik a pozicionált, hagyományos text-alapú feliraton is.

Ez a rész teljesen dokumentálva van kódon belül. Gyakorlatilag szinte soronkénti belső kommentekkel láttuk el.

Második fő része a Vizsgamunka webfejlesztésünknek a **Főoldal**. Itt és más, funkcionális aloldalakon is a <body> manipulált animációjával illetve a <div>-ek előre szerkesztett, plugin-nal animált megjelenítésével értünk el dinamikus megjelenést. A mozgások saját animációk és keyframe-ek meghívásával működnek. A <div>-eket az AOS plugin felhasználásával animáljuk. Ezekről is részletes belső dokumentációt adtunk meg mind az adott oldal html részében, mind a kapcsolódó stílusleíró fájlokban (.css).

Fontos formázási elemnek tartjuk a görgetés lágyítását (`scroll-behavior: smooth;`) és a vertikális scroll tiltását is (`overflow-x: hidden;`), ami egyfajta megjelenítési kötöttséget ad számunka, ami az alapvető megjelenés (Mobile First) tesztelése folyamán elengedhetetlen a teljes reszponzivitás biztosítására.

A főoldalon PHP script nyelvet használunk, form-okkal oldjuk meg a kliens-szerver kommunikációt. Az adatok érzékeny minősége miatt a POST metódust használjuk; illetve a bejelentkezési eljárás során a jelszót titkosított formában (egyirányú hash) küldjük el és tároljuk az adatbázis tábla rekordjában. A bejelentkezés során a validációs értékek Igaz értékelése esetén (usernév és hash jelszó egyezősége az adatbázisban tárolt értékpárral) a kiértékelő eljárás Igaz ágában létrehozunk szuperglobális változókat a későbbi eljárások alapértékeként és a visszairányítási eljárások alapértékekén. Egyik ilyen Session-változó a 'loggedin'. Ennek 'true' értéke jelzi, hogy megtörtént a user bejelentkezése és így jogosultsági szint jelzésére alkalmas változó a továbbiakban is.

```
// A találati adatok letárolása a session-változóba, belépési igazolás (loggedin = true)
$_SESSION["loggedin"] = true;
$_SESSION["id"] = $id;
$_SESSION["username"] = $username;
$GLOBALS['username'] = $username;
```

13. sz. ábra – Webfejlesztés – Főoldal – belépési siker esetén létrehozott szuperváltozók dedikálása.

Minden olyan oldalon, aminek tartalmát csak bizonyos jogosultság alapján érheti el felhasználó (pl.: bejelentkezett felhasználó érheti csak el), kötelező vizsgálni a kód első sorában a jogosultsági szint meglétéről.

Ez a mi esetünkben a fenti változó 'igaz' értékének vizsgálatával történik. Amennyiben ez 'hamis', átirányítjuk a felhasználót a főoldalra. Ezzel el tudjuk kerülni mind az url copy-paste (címsorba), mind a kényszerített könyvjelzőhasználat alkalmazását.

```
<?php
// ha nincs belépve, redirection -> főoldalra
session_start();
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"]){}
else{ header('location: fo3.php'); }

?>
```

14. sz. ábra – Webfejlesztés – Jogosultság ellenőrzése, hiányában: átirányítás.

Nagymértékben támaszkodunk a Bootstrap 5 keretrendszer kínálta formázási és működési stílusok használatára. Felhasználjuk a kész elemeket, így például a navigációs sávot és annak gyűjtött, tömörített változatát is (collapszált navigációs csík és lista). Regisztráltunk a FontAwesome oldalon, amiért is látványos és animált ikonokat tudunk felhasználni weboldalunkon. Ezeket ötvöztük a fejléc, navigációs részében:

```
<div class="container-fluid" id="fejléc" style="background-color: rgb(0, 11, 61); opacity: 1;" >
</div>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark sticky-top">

  <div class="container-fluid align-middle">
    <a class="navbar-brand" href="#">
      
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="#D1"><i class="fa-solid fas fa-gamepad fa-flip" style="--fa-animation-duration: 3.5s; font-size: 1.5em;"> </i>Játékról</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

15. sz. ábra – Webfejlesztés – Főoldal – Fejléc – Bootstrap és FontAwesome alkalmazása, kódrészlet.

Jogosultságtól függően változó oldaltartalom és megjelenés érhető el. Jelenleg az oldal újrarahívásával érjük ezt el, PHP adatforgalmazás keretében. Látványosabb számunkra most ez a megoldást, mert az animációk teljes mértékig tettenérhetők ebben a formában; míg AJAX alkalmazásával külön meg kellene hívnunk az oldalt ennek elérésére (pl.

```
window.location.reload(true);
```

Belépés hiányában a Főoldal tartalmi része érhető el. Információ a játékról, céljáról. Információ a készítőkről, oktatóikról, a képző intézményről (iskoláról) és egy GY.I.K. rész, amit Bootstrap accordion használatával és annak egyedivé alakításával értünk el:


```

<script>
var acc = document.getElementsByClassName("accordion");
var i;

for (i = 0; i < acc.length; i++) {
  acc[i].addEventListener("click", function() {
    this.classList.toggle("jelolve");
    var valaszpanel = this.nextElementSibling;
    if (valaszpanel.style.maxHeight) {
      valaszpanel.style.maxHeight = null;
    } else {
      valaszpanel.style.maxHeight = valaszpanel.scrollHeight + "px";
    }
  });
}
</script>

```

16. sz. ábra – Webfejlesztés – Főoldal – GY.I.K. – accordion JS kód.

Az oldalon alkalmazunk még 2D és 3D elforgatásokat a készítő (tanulók) kártyáinak mozgatásánál. Használunk vertikális parallaxot a képző intézmény (iskola) bemutatásánál.

Külön funkciót kapott az 'Elfelejtett jelszó' link. Erre klikkelve új oldalon lehetőség van e-mailben egy új, véletlenszerű, ideiglenes jelszót igényelni és kapni. Ezt a PHP 'mail' beépített függvényének HTML formátumú alkalmazásával értük el.

A felhasználó regisztrál vagy belép. Belépés után elérhető számára a Toplista tartalma és játékot indíthat: megjelenik a Start gomb.

Erre klikkelve indul a **Harmadik rész**, a **Játék** szakasz. Ennek az oldalnak a felépítéséről fentebb már írtam. Jellemzően JS használatával készült. A működés logikai leírása részletes kommentként megtalálható a forráskódban. Itt is igyekeztünk részletesen, fejlesztői eljárásként dokumentálni a kódot. Lett volna lehetőségünk teljesen PHP alapon megoldani a működést, de véleményünk szerint elvesztette volna a most meglévő elegáns megjelenését és rugalmas, könnyed működését felhasználói élmény szempontjából a játék. PHP alapokon (JS mellőzésével) kérdésenként kellett volna egy 'Tovább vagy Mehet' gombot alkalmaznunk, a kérdésenkénti POST/GET küldés érvényesítésére, ami feldarabolta volna a játékmenetet. Így JS és AJAX párosítással oldottuk meg ezt a részt.

Néhány OOP 'eljárás/függvény' alkalmazásával képesek voltunk követni a logikai működés elvárásait és lépéseit. Külön figyelmet fordítottunk itt is a különböző méretű kijelzőkön való megjelenés minőségére (full responsiv) és az olvashatóságra is. Egy ismert eljárást használtunk fel a JS tömb véletlenszerűségnek biztosítására. Ezt kombináltuk azzal, hogy az így létrehozott véletlen-sorrendű tömb utolsó elemét használjuk aktuális kérdés/válaszok azonosítóként, majd ezt a használat után töröljük a

tömbből, így a ciklus (ami a tömb hosszának megfelelő hosszúig fut) következő fordulójában az eggyel rövidebb tömb megint utolsó elemét fogja elővenni. Így kerültük el az ismétlődés lehetőségét a játék folyamán a kérdés/válaszok tekintetében.

```
// A "Fisher-Yates Shuffle" eljárás alkalmazása 'véletlen' sorrendű tömb létrehozásához
function shuffle(array) {
    let counter = array.length; // 'counter' változó = tömb hossza

    while (counter > 0) {
        // Véletlen alapú index létrehozása a tömb hosszát alapul véve
        let index = Math.floor(Math.random() * counter);

        // 'counter' változó értékének csökkentése 1-gyel (max = max -1)
        counter--;

        // ...és itt cserélgetjük össze a tömb elemeit -> ideiglenes = utolsó (max), utolsó = véletlen, véletlen = ideiglenes - ciklusban végig -> első
        // tömbelemig visszafelé
        let temp = array[counter];
        array[counter] = array[index];
        array[index] = temp;
    }

    return array;
}
```

17. sz. ábra – Webfejlesztés – Game/Játék oldal - „Fisher-Yates Shuffle” eljárás.

A játék ebben a formában a beállított 60 másodperces idő leteltéig fut. Ekkor a játékvége (GameOver) eljárást hívjuk meg. Kijelezzük az elért eredményt és AJAX használatával inzeráljuk is a Toplista táblába automatikusan.

A játék folyamán bármikor megszakíthatjuk és újratekzhetjük a játékot. Bármikor visszatérhetünk a Főoldalra. Ezekben az esetekben nem hívjuk meg a Játékvége eljárást és így nem történik rekord bejegyzése a Toplista táblába.

Összegzésként: jelentős figyelmet fordítottunk a belső dokumentálásra. Áttekinthetővé kívántuk tenni a kódot. A működést és belső logikát leíró kommentekkel segítettük az értelmezést és nyomonkövetést. Ezt nemcsak harmadik fél, hanem saját magunk számára is fontosnak tartjuk.

3.4 Tesztelési dokumentáció

A tesztelési eljárások és folyamatok során a meglévő és fellelhető hibákat akartuk megtalálni és javítani. Még élesítés (üzembe helyezés) előtt akartunk arról meggyőzősodni, hogy az alkalmazás az elvárt módon és megfelelően működik-e; valamint az egyes funkciók jól működnek és a kitűzött specifikációkat teljesítik-e?

Folyamatosan teszteltünk. Mind megjelenés, mind működés és visszacsatolt érték/eredmény szempontjából. Egyes kisebb részeknél többször alkalmaztuk egy-egy részfeladat vagy működés értékének konzolra (console.log) való kiíratását is. Nagy segítséget jelentett számunkra a bérelt tárterületen élesített MySQL automatikus 'error_log' naplóbejegyzései a PHP unit teszthei és funkcionális teszthei során. Fő

irányelvként a használat során az elvárt eredmény elérése és emellett más eredmény el nemérése (hiba) volt a fő cél.

```
01-Apr-2022 14:14:52 Europe/Budapest] PHP Warning: mysqli::__construct(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: NO) in /home/csongra3/kviz.csongradi.eu/config.php on line 88
01-Apr-2022 14:15:47 Europe/Budapest] PHP Warning: mysqli::close(): Couldn't fetch mysqli in /home/csongra3/kviz.csongradi.eu/fo3.php on line 88
01-Apr-2022 14:33:28 Europe/Budapest] PHP Warning: mysqli::__construct(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: NO) in /home/csongra3/kviz.csongradi.eu/config.php on line 88
01-Apr-2022 14:33:44 Europe/Budapest] PHP Warning: mysqli::__construct(): (HY000/1045): Access denied for user 'root'@'localhost' (using password: NO) in /home/csongra3/kviz.csongradi.eu/config.php on line 88
01-Apr-2022 14:33:44 Europe/Budapest] PHP Warning: mysqli::close(): Couldn't fetch mysqli in /home/csongra3/kviz.csongradi.eu/fo3.php on line 88
01-Apr-2022 15:14:00 Europe/Budapest] PHP Warning: mysqli_stmt::bind_result(): Number of bind variables doesn't match number of fields in prepared statement in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:17:49 Europe/Budapest] PHP Warning: mysqli_stmt::bind_result(): Number of bind variables doesn't match number of fields in prepared statement in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:18:03 Europe/Budapest] PHP Warning: mysqli_stmt::bind_result(): Number of bind variables doesn't match number of fields in prepared statement in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:19:06 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:20:47 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:21:08 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:23:00 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:36:05 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:37:18 Europe/Budapest] PHP Fatal error: Uncaught Error: Call to undefined function bevitel_szures() in /home/csongra3/kviz.csongradi.eu/reset-password.php:24
Stack trace:
#0 {main}
  thrown in /home/csongra3/kviz.csongradi.eu/reset-password.php on line 24
01-Apr-2022 15:38:01 Europe/Budapest] PHP Notice: session_start(): A session had already been started - ignoring in /home/csongra3/kviz.csongradi.eu/fo3.php on line 59
01-Apr-2022 15:38:11 Europe/Budapest] PHP Fatal error: Uncaught Error: Call to undefined function bevitel_szures() in /home/csongra3/kviz.csongradi.eu/reset-password.php:24
Stack trace:
#0 {main}
```

18. sz. ábra – Webfejlesztés – error_log – részlet.

Mi magunk White-Box teszteket végeztünk, a forráskód és általában az eljárás ismert működése alapján, elvárt eredményre koncentrálnak. Ez mind érték, mind működési eredmény is lehetett.

Volt lehetőségünk Black-Box tesztelésre is, ahol megmondtuk, hogy mit várnak el tőlük input részen és milyen outputot kellene tapasztalniuk majd: családtagok és ismerősök között van aki végzett fejlesztő vagy éppen tanulmányokat folytat ilyen irányban. Velük végeztettünk el jónéhány próbát, melyekről képernyőfotókkal is beszámoltak.

Tesztelési szintek közül mi a **fejlesztői teszteket**, white-box környezetben végeztük el több esetben. Ezek közül:

- komponens teszt (főként unit-teszt és minimális modulteszt: mennyire gyors?)
- integrációs teszt (több komponens együttműködésének vizsgálata – főként Andrásvaló együttműködés alapján került ki innen hibaforrás – változó-kezelés különböző értelmezéséből)
- rendszerteszt (az egész rendszer, minden komponens együttes működésének vizsgálata)

végeztük el, több esetben is.

Átvételi tesztet pedig a bérelt területre feltöltött, szakaszonként aktualizált verziókkal végeztettünk el. Megkértük ismerőseinket és csoporttársainkat is, hogy keressék fel a weboldalt és teszteljék, próbálják meg szokott és nem-szokott használati módokkal is (fekete-dobozos eljárás alapján). Gyakorlati megvalósulásában ezeket alfa- és béta-teszteknek is tekinthetjük. Itt az 'alfa-teszt' folyamán arra kértük őket, hogy próbáljanak teljesen logikátlan viselkedést produkálni (pl.: extrém gyors és céltalan egérklikkelés, el

nem várt adatok megadása a Regisztrációkor, url-copy-k használata a címsorban, SQL-hack és JS-injection tesztelés...stb.). 'Béta-teszt' esetén pedig kértük, hogy lehetőleg minden funkciót próbáljanak ki és erről referáljanak.

A unit-tesztek folyamán sok-sok rész működését vizsgáltuk. Az eredmények alapján finomítottunk a kódon, néha újraírtuk. Például: egész egyszerű feladatnak tűnt az Intro oldalon a 'labdák' véletlenszerű megjelenésének és mozgásának létrehozása. Egy darabot hamar elő tudtunk állítani, de amikor több (és definiáltan több) darabot akartunk injektálni és főként azokat mozgatni animációval, akkor annak kivitelezése sok időt vett igénybe. Elemi áttörést az hozott, amikor szembesültünk azzal, hogy magát az animációt is ciklusba kell foglalnunk és nem célravezető az :nth-child() pszeudó alkalmazása ezek során. (Ami eljárás viszont hibátlanul üzemelt a jobb alsó sarokban lerakott pontok-pöttyök színfutásának 'animálásakor'). Ugyanígy sok fejtörést okozott egy saját fejlesztésű szorzótábla alkalmazás (ezt használtuk fel motorként) átalakítása kvíz-játék alapnak. Nagy segítség volt, hogy rendelkezésünkre állt már a megjelenítésért felelős és magát a játékmenetet meghatározó komponensek; de éppen az eredményvizsgálat logikai eljárása okozott ekkor zavart. Hiszen itt nem számokkal (konkrét változó numerikus értékével), hanem <div>-ek teljes egyezőségével kellett megoldanunk a kiértékelést. Próbáltunk egy másik vonalat is alkalmazni, de a totális egyezőség könnyedebb megoldásnak mutatkozott (másik út: <div>-enként data- prefix alkalmazása és ennek vizsgálata – egyezőség). Sokszor éltünk azzal a módszerrel az **integrációs tesz**t folyamán is, hogy futás közbeni értékeket a Konzolra írtunk ki és ezek vizsgálata derített fényt (vagy okozott még nagyobb kérdést) egyes hibákra.

Sok tesztet generáltunk az **integrációs teszt és a rendszerteszt** folyamán is..

Létrehoztunk próba felhasználókat, regisztráltuk, elfelejtettük a jelszavát, emailben kapott ideiglenes jelszóval bejelentkeztünk, új jelszót állítottunk be, játékot indítottunk, végigjátszottuk, megszakítottuk, újrakezdtük...sok-sok alkalommal. Figyeltük a működési eredményeket és folyamatosan elemeztük a kapott adatokat és ellenőriztük mind a konzol kiírásokat, mind a bérelt területen lévő adatbázis tartalmát is.

Ezek alapján módosítottuk például, hogy a felhasználónév tartalmazhasson magyar kis- és nagybetűket is. Ne lehessen 3 karakternél rövidebb és 16 karakternél hosszabb. Ugyanígy 3-16 karakter hosszúra és angol karakterek használatára állítottuk be a jelszó lehetőségét. Vizsgáltuk innentől, hogy használatban van-e már az adott usernév? Ha regisztrálták már, akkor hibajelzést adunk. Bővítettük a hibajelzési kitételezést, viszont beléptetésnél

butítottuk: ugyanazt az univerzális hibát adjuk akkor is, ha a felhasználónév vagy a jelszó, vagy akár mindkettő téves. Így nem adunk visszaélési információt arra nézve, hogy melyiket rontotta el a felhasználó. Ugyanígy állítottunk a Start gomb megjelenésén, csak bejelentkezés után válik láthatóvá. Egyfajta nyomásgyakorlás után oda jutottunk, hogy a Toplista menü elérhetőségét is csak belépett usereknek tesszük használhatóvá.

Az 'alfa-' és 'béta-teszteknek' köszönhetően kiderült számunkra, hogy egyes mobil eszközön megjelenési anomáliák lépnek fel. A kis teljesítményű, nagyrészt belépési szintű mobil eszköz nem képes megfelelő minőségben megjeleníteni az Intro oldal animációját. Töredezett (laggos) a megjelenítés és a letiltott vertikális és horizontális scrollozást felülírja a készülék és/vagy operációs rendszere. iOS alatt, még régi készülékeken is megfelelően fut a webalkalmazás, de a Főoldal fejlesztői kártyáinak forgatása folyamán a hátoldali háttér helyett az előoldali háttér jeleníti meg x-irányú, tengelyes tükrözéssel (nem hajtja végre 2D forgatás utáni toggle.hidden eljárást).

A teljes tesztelési folyamat végén még többször fésültük át a kódot, szemantikai vagy logikai hibák után kutatva. Teljes biztonsággal nem állíthatjuk, hogy hibamentes a létrehozott alkalmazás, de bízunk a visszajelzések számosságában.

4 Összefoglalás

4.1 Önértékelés

Alapvető cél volt, hogy a Front-end részen végbemenő szemléletváltást bemutassuk a vizsgamunkánk során. Értem ezalatt, hogy a weboldal és webes alkalmazás elsősorban az információt egyértelműen és teljes módon közölje; emellett a megjelenése is legyen dinamikus és formailag is tükrözzön tervezői (design) egységet. Nem szerettünk volna egy sokszor implementált alkalmazást előhozni vagy annak valami kényszerített változatát elkészíteni. Így tehát nem akartunk webáruházat, társkeresőt, online boltot vagy valami reform életmód magazint vizsgamunkaként. Kislányomnak írtam év közben egy szorzótábla alkalmazást és ez jó alapnak tűnt egy kvízzjáték létrehozására. Tanulságos volt, hogy mennyit tud segíteni és mennyire korlátok közé tud szorítani egy meglévő alap vagy alkalmazás átfejlesztése.

Szerettünk volna megjelenési és működési minőségben olyat készíteni, ami méltó arra a kimagaslóan sok figyelemre és beletett energiára, amit a tanárainktól-oktatóinktól kaptunk. Nagyon bízik benne, hogy Andrással létrehozott vizsgamunkánk tükrözi ezt és azt a vágyunkat, hogy a felhasználóink maradandó élményt kapjanak használat során. ...és ez legyen örömteli élmény számukra.

Több opciót nem valósítottunk meg. Például kihagytuk a második játékmód megvalósítását. Nem láttuk volna a gyakorlati hasznát. Nem találtunk indokot arra, hogy mivel mutatna meg többet a képességeinkből, ha egy másik oldalon szinte ismételnénk a meglévő játék kódját. Kiegészítenénk egy (!) új változóval, ami az életek száma és az eredményvizsgálatnál beraknánk, hogy rossz válasz esetén ezt csökkentse eggyel.

Nemcsak a hátralévő időt, hanem az életek számát is vizsgálnánk és egy 'vagy – OR' kapcsolattal meghívnánk a JátékVége eljárást (objektumot). Igazából csak némi kódolási pluszt adnánk hozzá, de alkalmazott képesség tekintetében nem lenne előrevívő.

Ugyanígy, sematikus módon egy másik Toplistát is létrehoznánk és ennek az Insert-jét is valójában másolnánk a meglévő kód alapján.

Fentiek alapján azt gondolom, abban a minőségben készült el a vizsgamunka, ahogy elterveztük. Kimondottan erőteljes megoldásokat hoztunk létre a felhasználók jogosultságának kezelésére, ami – véleményem szerint – bármely weboldalon versenyképes megoldás lehetne.

Számomra a Bootstrap túllépett a 'csak egy formázási keretrendszer' kategórián. Meglepődtem, hogy mennyi kész elemmel bír, amelyeket normalizált .css file alkalmazása nélkül is fel lehet használni a RAID alapelveknek megfelelően. Sokat fejlődtem JS és PHP szkript nyelvekben is. Itt is azt tapasztaltam, hogy minél többet tanulok, annál inkább ítélem meg az elért szintemet egyre alacsonyabbnak. Minden bizonnyal ez egy kód-írás skill paradoxon, de feltétlenül szakmai alázatra nevel, egyre nagyobb tudás-éhség mellett.

4.2 Továbbfejlesztési lehetőségek

Ahogy a GY.I.K. részen is megírtuk, sok fejlesztési terület maradt még. A mi részünkről terv volt egy olyan felület elkészítése, ahol a belépett felhasználók be tudtak volna küldeni egy kérdést és hozzá 4 választ, a jó válasz megjelölésével. Ez tényleg időszúke miatt nem került megvalósításra.

A csapatjátékot szintén érdemes lenne még megvalósítani. Ugyanígy a témakörökre bontott játék is hordoz magában fantáziát.

Bízunk abban, hogy akad majd a jövőben inspirációnk arra, hogy ezek közül valamennyit vagy akár mindet megvalósítsuk.

5 Felhasznált irodalom

- SzoftverTesztelés - <http://aries.ektf.hu/~gkusper/SzoftverTeszteles.pdf> (2022.04.21)
- AOS plugin (Animate On Scroll library) - <https://michalsnik.github.io/aos/> (2022. 04.22.)
- Tutorial.hu: <http://www.tutorial.hu/php-tombok/> (2022.04.13.)
- PHP dokumentáció: <http://php.net/docs.php> (2022.04.02.)
- Aktuális böngésző szabvány: <https://html.spec.whatwg.org/multipage/> (2022.04.23.)
- SZTE TTIK Szoftverfejlesztés Tanszék - Oktatási célú szerver: <https://okt.sed.hu/> (2022.04.23.)
- Animation CSS Generator: <https://webcode.tools/generators/css/keyframe-animation> (2022. 04.11.)
- W3Schools és aloldalai: <https://www.w3schools.com/> (2022.04.23.)
- Webadatbázis programozás: <http://ade.web.elte.hu/wabp/index.html> (2022.04.20.)
- Javascript.info: <https://javascript.info/js-animation> (2022. 04. 08.)
- Datatables.net: <https://datatables.net/> (2022. 04.03.)
- Masonry Layout: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Masonry_Layout (2022: 04. 03.)

6 Ábrajegyzék

1. sz. ábra – JavaScript aktiválása Chrome böngészőben.
2. sz. ábra – Webfejlesztés - Intro oldal képernyőfotója. – jobb alsó sarokban: Tovább
3. sz. ábra – Webfejlesztés – Főoldal – menüsor, ami a tartalmi elemekhez juttat
4. sz. ábra – Webfejlesztés – Regisztrációs oldal.
5. sz. ábra – Webfejlesztés – Főoldal – belépés utáni képernyő –Start gomb a játékindításhoz
6. sz. ábra – Webfejlesztés – Játékoldal – kezdés képernyő – Start gombra klikkelve indul a játék
7. sz. ábra – Webfejlesztés – Játékoldal – Játék vége képernyő – elért eredmény kijelzése.
8. sz. ábra – Webfejlesztés – Toplista.
9. sz. ábra – Webfejlesztés – Adatbázis szerkezet.
10. sz. ábra – Webfejlesztés – Táblakapcsolat.
11. sz. ábra – Webfejlesztés – ERD (Entity Relationship Diagram)
12. sz. ábra – Webfejlesztés – Fejlesztés kezdetén meghatározott UML diagram.
13. sz. ábra – Webfejlesztés – Főoldal – belépési siker esetén létrehozott szuperváltozók dedikálása.
14. sz. ábra – Webfejlesztés – Jogosultság ellenőrzése, hiányában: átirányítás.
15. sz. ábra – Webfejlesztés – Főoldal – Fejléc – Bootstrap és FontAwsome alkalmazása, kódrészlet.
16. sz. ábra – Webfejlesztés – Főoldal – GY.I.K. – accordion JS kód.
17. sz. ábra – Webfejlesztés – Game/Játék oldal - „Fisher-Yates Shuffle” eljárás.
18. sz. ábra – Webfejlesztés – error_log – részlet.