

Szakképesítés neve: Szoftverfejlesztő

OKJ száma: 54 213 05

Záródolgozat

Címe: *A Macska Király*

Kajati Ádám

5/13.C

Szabó-Maczkó Henrietta

Témavezető

Sátoraljaújhely, 2020.

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

1. A témaválasztás indoklása

Amióta az eszemet tudom azóta az informatika világa az érdeklődésemet mindig felkeltette. Ezért amikor arra került sor, hogy középiskolát kellett választanom a választás a szülővárosban lévő nagyon jó hírnévben álló „Keri-re” esett és azon belül is informatika ágazatra. Az érettségi sikeres teljesítése után lehetőségem volt az intézmény OKJ-s képzéseire maradni és ezek közül is a szoftverfejlesztés volt számomra a legérdekesebb, de jövő tanévre tervezem, hogy maradok még egy évet az intézményben és elvégzem az OKJ-s rendszergazda képzést, mivel az is érdekelne engem. Középiskolai rendszerben eltöltött négy év alatt rengeteg informatikai tudást „szívtam” magamba és azok közül is a programozás és a webfejlesztés érdekelt a legjobban. Középiskolában a Cisco Packet Tracer nevezetű programban készítettünk virtuálisan hálózatokat és azon kívül még Microsoft Visual Studio nevezetű programban a C# programozási nyelvvel ismerkedtünk meg. A szakdolgozat témaválasztásánál nehéz helyzetben voltam, mert annyi sok téma volt a fejemben, amiről készíthettem volna a szakdolgozatomat, de aztán az egyik nap megjött a tökéletes ötlet: Egy macskákról szóló weboldalt lett a választásom, ahol az egyes macskafajtákat mutatom be képekkel együtt, macskáról szóló érdekességeket gyűjtöm össze és még ezeken kívül a felhasználó a saját kisállatáról (macskájáról) képet tud feltölteni és megtudja azt osztani a weboldalon. Ezt az ötletet meg lehetett volna oldani egy komplett programban is csak akkor annyi lenne a hátránya, hogy akkor a többi felhasználó nem tudná megosztani a saját kedvencéről készített képét és egy weboldalon személy szerint jobban megtudom valósítani az ötletemet. Másrészt azért weboldalra esett a választás, mert a weboldal készítés mindig közelebb állt a szívemhez és középiskolai tanulmányaim során is jobban szerettem weboldalt készíteni, mint programot írni. A macskás témára pedig azért esett a választás, mert imádom a macskákat és úgy gondoltam, hogy arról a témáról lehetne számomra a legjobb szakdolgozatot írni. Nagyon örülök, hogy szakdolgozatot kell készítenem, mert ennek elkészítése alatt tudom fejleszteni a programozási képességeimet és a későbbiekben egy nagyon jól használható referencia munkát tudok készíteni, amit a pár év múlva akár tovább is tudok fejleszteni.

2. Fejlesztői dokumentáció

A szakdolgozatom elkészítéséhez jelentős részben a Microsoft Visual Studio Code-ot használtam mivel ez a szoftver nagyon felhasználóbarát, könnyen kezelhető, nem foglal nagy mennyiségű tárhelyet és ingyenesen használható. A másik program amit használtam még az a Sublime Text 3 de ezt csak azért, hogy a PHP kódokat megtudjam írni mivel ez a szoftver natívan támogatja a PHP nyelvben való programozást. Azért kényszerültem arra, hogy ezt a szoftvert használjam, mert a Visual Studio Code nem támogatja natívan a PHP-t és ezért abban megírni a kódot kissé körülményes lett volna. Ennek a program előnyei szinte megegyeznek a VS Code előnyeivel. A szakdolgozatom adatbázisos részéhez a XAMMP nevezetű programot használtam, amivel egyszerűen mindenki a saját számítógépén létre tud hozni egy localhost-ot (helyi kiszolgáló) melynek segítségével szerver bérlete nélkül tud futtatni egy adott weblapot, de csak és kizárólag helyi hálózaton belül. A PHPMYADMIN segítségével lehet kezelni az adatbázissal való összeköttetéseket, tulajdonságokat, paramétereket. Kettő darab böngésző segítségével teszteltem, hogy a weblap nem csak egy webböngészőből fut rendesen. Ez a kettő pedig nem más volt, mint a: Google Chrome és a Microsoft Edge. A weblapot 1366x768-as felbontás szerint készítettem el mivel az otthoni monitorom csak erre a felbontásra képes, emiatt nagyobb felbontású monitoron előfordulhat az, hogy a tartalom nem megfelelően jelenik meg de csak néhány helyen a szöveg nem pontosan azt a pozíciót foglalja el, mint az én monitoromon. Más problémával nem találkoztam, amivel ezzel kapcsolatos lett volna.

3. Használt programozási nyelvek bemutatása és a szoftver választás indoklása

A szakdolgozatom elkészítéséhez ezeket a programozási nyelveket használtam:

- **HTML (HyperText Markup Language)**

A HTML egy olyan nyelv, melyet weboldalak elkészítésére használhatunk. Minden weboldal mögött HTML található. A kifejezés a Hypertext Markup Language (hiperszoveges jelölőnyelv) rövidítése, és egy utasításkészlet a webböngésző számára. Az utasítások betartásával a böngésző úgy jeleníti meg a weboldalt, ahogy azt mi szeretnénk.

A hiperszoveg elnevezés már több mint 50 éves, és eredetileg arra szolgált, hogy leírjon linkeket, melyek egy dokumentumban szerepeltek, és ezáltal lehetővé tegye, hogy az olvasó a dokumentum egyik pontjára annak másik pontjára, vagy akár egy egészen más dokumentumba kerüljön.

Amit fontos tudni, hogy a HTML nem egy programozási nyelv, hanem egy jelölőnyelv. A HTML-t viszont ilyen feladatokra nem tudod használni, csak arra szolgál, hogy meghatározza egy weboldal megjelenését.

A HTML felépítése a következő: Címke, Attribútum.

- **CSS (Cascading Style Sheets)**

A CSS egy stíluslap, mely leírja egy weboldal megjelenését. A CSS azt mondja meg a böngészőnek, hogy egy HTML dokumentum egyes elemei miként jelenjenek meg a felhasználók számára. Legfontosabb jellemzője, hogy képes különválasztani a weboldal-tartalmat a designelemektől.

- **JavaScript**

A JavaScript programozási nyelv egy objektumorientált, prototípus alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. Eredetileg Brendan Eich, a Netscape Communications mérnöke fejlesztette ki; neve először Mocha, majd LiveScript volt, később „JavaScript” nevet kapott, és szintaxisa közelebb került a Sun Microsystems Java programozási nyelvéhez. A jelenleg is érvényes szabvány az ECMA-262 Edition 3 (1999. december), ami a JavaScript 1.5-nek felel meg. Ez a szabvány egyben ISO szabvány is.

- **PHP**

A PHP elterjedt nyílt forráskódú szerver-oldali programozási nyelv, amely az SSI és a Perl ötvözésére hivatott. Szintaktikája leginkább a C programozási nyelvéhez hasonlít. Megalkotója Rasmus Lerdorf. A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak, mint formázást használják, ugyanis ezen lapok teljes funkcionalitása a PHP-re épül. Amikor egy PHP-ben megírt oldalt akarunk elérni, a kiszolgáló először feldolgozza a PHP utasításokat, és csak a kész (HTML) kimenetet küldi el a böngészőnek. Ehhez egy ún. interpretert (értelmezőt) használ, amely általában egy külső modulja a webszervernek. A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek (vagy ha igen, korlátozottan). Ilyen pl. a bejelentkezés, az adatbázis kezelés, fájlkezelés.

- **SQL**

Az SQL, azaz Structured Query Language relációsadatbázis-kezelők lekérdezési nyelve. Az SQL nyelvi elemeket 4 részre, adatdefiníciós (Data Definition Language, DDL), adatkezelési (Data Manipulation Language, DML), lekérdező (QUERY (Language - QL)) és adatvezérlő (Data Control Language, DCL) részekre lehet bontani. A nyelvben az utasításokat a pontosvessző választja el egymástól.

A szakdolgozatom elkészítéséhez ezeket a programokat használtam:

- **Microsoft Visual Code**

A Visual Studio Code (rövidítve: VS Code) egy ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és OS X operációs rendszerekhez. Támogatja a hibakeresőket, valamint beépített Git támogatással rendelkezik, továbbá képes az intelligens kódkezelésre az IntelliSense segítségével. Ezen felül testre szabható, így a felhasználók megváltoztathatják a kinézetet (témát), megváltoztathatják a szerkesztő gyorsbillentyű-kiosztását, az alapértelmezett beállításokat és még sok egyebet. A Visual Studio Code az Electron nevű keretrendszeren alapszik. A szoftver teljes mértékben ingyenesen használható

A szakdolgozatom 90%-át ezzel a programmal készítettem el. Nagyon felhasználóbarát, könnyen kezelhető és kompatibilis rengeteg programozási nyelvvel. Ezt a programot még a barátaim ajánlották számomra és azóta bármilyen weblapot szeretnék szerkeszteni azóta csak ezt a szoftvert használom.

- **Sublime Text**

A Sublime Text egy Visual Studio Code-hoz hasonló szerkesztő program, ami szintén rengeteg programozási nyelvet használ. A szoftvert ingyenesen is lehet használni próbaidő alatt különben fizetős. Azért használtam ezt a programot mert a Visual Studio Code-nak nincsen beépített natív PHP nyelv támogatása viszont a Sublime Text tartalmaz ilyen támogatást. Az összes PHP-hez kapcsolódó fejlesztést ebben a szoftverben hajtottam végre.

- **XAMPP**

A XAMPP egy szabad és nyílt forrású platformfüggetlen webservert-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webservert, a MariaDB adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői. Hivatalosan a XAMPP tervezői az eszközt egy fejlesztőrendszernek szánták, amellyel a web-tervezők és programozók internetes kapcsolat nélkül fejleszthetik és tesztelhetik alkalmazásaikat. A XAMPP telepítése után a helyi gép (a localhost) hálózati gépként is hozzáférhetővé válik.

3. Weboldal navigációjának ismertetése



A tervezés első pillanatától kezdve egy modern kinézetű weboldalt szerettem volna létrehozni. Az emberek számára sokkal vonzóbb egy olyan oldal, ahol a készítő ügyelt a kinézetre és nem csak olyan színeket használt, amik egyáltalán nem illenek egymáshoz. Háttérképnek egy színátmenetes hátteret választottam, ami szerintem a „modern” dizájnt kielégíti. A betűtípus kiválasztásánál is ügyeltem arra, hogy ne az alap Times New Roman stílus uralkodjon a betűk felett, hanem egy sokkal letisztultabb, ami nem más, mint a Consolas típus. Fontosnak tartottam azt, hogy ha megnyitjuk az oldalamat akkor ne rögtön a tartalommal ellátott oldal jelenjen meg hanem először egy úgymond „bevezető-üdvözlő” rész következzen. Ahogyan látható az oldal tetején a legfontosabb dolog található az oldal és a szakdolgozatom címe utána középen a navigáló gombok és végül, de nem utolsó sorban az oldal alján szerepel, hogy ki készítette a weblapot. Bármelyik menüpontra visszük az egerünket az fekete színről **pirosra** vált áttünéses animáció segítségével.

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="style.css">
  <title>A Macska Király</title>
</head>
```

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit Általános Iskola, Óvoda

Legelőször a DOCTYPE paranccsal megadom a böngésző számára, hogy ez a fájl HTML típusú, utána a lang="hu" beállítással megadtam, hogy az oldal nyelvezete magyar lesz. Link sor az egyik legfontosabb rész egy HTML dokumentumban, mert ezzel a sorral tudjuk megadni azt, hogy a weboldalunk egy külső CSS fájlra hivatkozik, amivel eléggé meg tudjuk könnyíteni a munkánkat. Ha nem használnánk külső CSS fájlt, hanem belső stílust alkalmaznánk az sokkal bonyolultabb és kevésbé átláthatóbb. A <title> közé az oldal címét adjuk meg. A HTML <head> (fejrészebe) kerülnek a legfontosabb parancsok, információk az oldalról.

```
<body>
  <!-- A szakdolgozat címe -->
  <p id="p1"><b><i>A Macska Király</i></b></p>
  <div>
    <!-- Menüsor létrehozása -->
    <ul id="menu">
      <center>
        <li><a href="Index.html">KEZDŐLAP</a></li>
        <li><a href="Fotók.html">FOTÓK</a></li>
        <li><a href="Tudnivalók.html">TUDNIVALÓK</a></li>
        <li><a href="Érdekességek.html">ÉRDEKESSÉGEK</a></li>
        <li><a href="Viselkedés.html">VISELKEDÉSÜK</a></li>
        <li><a href="login.php">BEJELENTKEZÉS</a></li>
      </center>
    </ul>
  </div>
  <p id="p2"><b><i>Készítette: Kajati Ádám 5/13.c-s tanuló</i></b></p>
</body>
</html>
```

A <p> jelzővel bármilyen szöveget le tudunk írni HTML-ben és még mellé egy azonosítót is megadtam (id). Én ezt arra használtam, hogy az oldal címét, ami középen helyezkedik el megadjam. Az azonosító azért észszerű használni, mert könnyedén meg lehet formázni és egy azonosítót bármennyiszer lehet alkalmazni és minden alkalmazott jelzőnél az a specifikus azonosító fog alkalmazódni. A menü létrehozását és HTML jelzőkkel oldottam meg. Ez a két jelző felsorolások listáért felelős és aztán ezeket úgy formáztam meg,

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit Általános Iskola, Óvoda

hogy egy vízszintesen elhelyezkedő menürendszert kapjak. Az `` arra hasznos, hogy hiperhivatkozást hozzunk létre, amit azt jelenti, hogy egy bármilyen szövegre kattintva egy másik oldal egy külső weboldalról vagy egy másik HTML fájl tud megnyílni. Az egész menürendszert egy `<div>`-en belülre írtam meg ami arra szolgál, hogy a weblapot blokkokra tudjunk osztani, amiket kedvünk szerint tudunk formázni azzal szemben, hogyha az egész oldal egy nagy blokk lenne akkor sokkal nehezebb lenne az egész szerkesztése. A `<center>` jelző használatával sikerült a menüpontokat úgy beállítani, hogy azok középen helyezkedjenek el. Az utolsó „li” -nél észrevehető, hogy ott a hiperhivatkozás amire mutat annak a fájlformátuma különbözik a többitől. Ez azért van, mert a bejelentkezést csak és PHP programozási nyelvben lehet megvalósítani és hogy megfelelően működjön a megírt kód ahhoz a fájlkiterjesztést meg kell változtatni .php-re.

```
body{
    margin:0;
    padding:0;
    background-image: url("gradient.jpg");
    background-repeat: no-repeat;
    background-size: cover;
}
#menu{
    margin:0;
    padding:0;
    margin-top:10px;
}

/* Az "li" megfelelővé formázása */
#menu li{
    list-style:none;
    display: inline-block;
}
```

A body-n (ami a HTML úgynevezett „teste”, ide kerül általában az oldal felépítésére vonatkozó kódok) belül kell beállítani a weboldal háttérképét. A background-repeat paranccsal beállítom azt, hogy a háttérkép ne ismétlődjön hanem fix legyen és a background-size pedig arra szolgál, hogy a háttérkép, amit kiválasztottunk az oldalunkra az teljes

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

mértékben lefedje a hátteret és ne csak egy apró szegletét. A #menu osztálykijelölő segítségével megadtam, hogy a menü felső margója 10px legyen, amivel segítetttem azt, hogy menürendszerben lévő gombok összhangban legyenek egymással. A list-style sorral azt állítottam be, hogy a menürendszernek ne legyen felsorolása. Ezt azért kellett megadni hiszen a menüt felsorolás alapúra készítettem el és így kezdetben egy felsorolás pont volt minden menüpont előtt. A menüt lehetett volna más módszerrel is elkészíteni például a <nav> jelzőkkel, amiket pont azért hoztak létre, hogy menüsort lehessen készíteni vele, de én úgy gondolom, hogy a helyes opciót választottam, mert <nav>-val nehezebb egy cseppet készíteni ilyet, mint felsorolással.

Most pedig a menü kinézetével és elrendezésével fogok foglalkozni:

```
/* Az "li"-n belül lévő "a" tag formázása */
#menu li a{
    text-decoration: none;
    position: relative;
    color:black;
    font-size:40px;
    font-weight: 700;
    font-family: monaco,Consolas,Lucida Console,monospace;
    display: block;
    overflow:hidden;
    transition: 0.7s all;
    padding: 14px 20px;
    text-transform:uppercase;
}
/* Animáció végrehajtása előtt */
#menu li a:before{
    content:'';
    width: 60px;
    position: absolute;
    border-bottom: 8px;
    bottom:0;
    right:350px;
    transition: 0.7s all;
}
```

```
/* Animáció végrehajtása után */  
#menu li a:hover:before{  
    right:0;  
}  
/* Amikor az egér kurzorját rávisszük a menüsorra akkor az piros színre vált  
    át feketéről */  
#menu li a:hover{  
    color:red;  
}
```

A „#menu li a” bekezdés a menüpontokra vonatkozó stílus formázásokat tartalmazza. A text-decoration sor az egyik legfontosabb ebben a blokkban, mert ez a parancs tünteti el az „li” által hozott felsoroláspontokat. Ezen kívül még az van beállítva, hogy milyen színű legyen a szöveg, milyen legyen a betűk nagysága, a betűk vastagsága, betűtípus és ezeken kívül még az, hogy minden egyes szó nagybetűs formázásban részesüljön. A „hover effektért” a „transition” felel, ami hét tizedmásodpercre van beállítva, vagyis, hogy az effekt hosszúsága ekkora időtartamú legyen. A következő blokk a #menu li a:before ami arra hivatott, hogy az effekt kinézetét létrehozza. Az animáció jobb oldalról kezdődik abszolút helyzetből. A következő kettő apró blokk arra hivatott, hogy amikor az egér kurzorát ráhelyezzük a menüsorra akkor milyen színre váltson át.

4. Weboldal oldalainak kódjának bemutatása



Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit Általános Iskola, Óvoda

A Fotók nevezetű oldalra macskákról szóló képeket szerettem volna elhelyezni. Az első tervem az volt, hogy táblázatszerűen rendezem el a képeket egymás mellé, de aztán támadt az az ötletem, hogy erre a feladatra tökéletes lenne egy JavaScript képvetítő. Összesen 10 kép van benne és ezeket gombok segítségével lehet előre és hátra navigálni, viszont, ha nem nyúlunk a gombokhoz akkor a program magától vált képet négyezer milliszekundumomként.

A programkód a következő:

```
<div id="slider">
  <div id="box">
    <!-- Melyik képpel induljon a képvetítés -->
    
  </div>
  <!-- Gombokra meghivatkozom a megfelelő function-t -->
  <button class="prev" onclick="prevImage()"></button>
  <button class="next" onclick="nextImage()"></button>
</div>
```

Először is egy <div>-et létrehoztam, amivel a „slider” azonosítóra hivatkoztam. Ez a <div> felelős a képek váltásáért dizájn szinten. A következő <div> a „box” azonosítóra hivatkozik, ami azt jelenti, hogy ez a <div> azért a „doboz” -ért felel, amiben a képek fognak megjelenni. A következő sorban megadtam azt, hogy melyik képpel kezdődjön a képvetítés. Ezek után két gombot hoztam létre, amik értelemszerűen arra hivatottak, hogy a képek között lehessen manuálisan váltani. A gombok a „prev” és „next” osztálykijelölőt kapták meg és kattintásra a „prevImage()” és „nextImage()” function-ök hajtódnak végre, amik hatására tudjuk ténylegesen változtatni a képeket.

```
<!-- Javascript deklaráció -->
<script type="text/javascript">
  var slider_content = document.getElementById('box');
  // Tömb létrehozása, amiben majd a képeket tudjuk tárolni
  var image = ['macska1', 'macska2', 'macska3', 'macska4', 'macska5', 'macska6', 'macska7', 'macska8', 'macska9', 'macska10'];
  var i = image.length;
  function nextImage(){
```

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám


```
if (i<image.length) {  
    i= i+1;  
}else{  
    i = 1;  
}  
  
// Következő képre történő hivatkozás  
slider_content.innerHTML = "";  
}
```

Egy HTML fájl kétféleképpen tud JavaScript fájllal dolgozni. Az egyik lehetőség az, amikor a HTML fájlban belül a <script> tag-eken belülre írjuk a JavaScript kódot. Ezt akkor szokás használni amikor kevés JavaScript kódot szeretnénk írni. A másik megoldás az, hogy egy különálló JavaScript fájlt hozunk létre, aminek a kiterjesztése .js. Ez a variáció akkor a legkifizetődőbb amikor nagy hosszúságú kódot szeretnénk írni ilyenkor érdemes különálló .js fájlt létrehozni és aztán hivatkozni rá a HTML-en belül. Ennek az egyik legnagyobb előnye az, hogy a különálló fájlt bármilyen más HTML fájlhoz tudjuk csatolni egyszerűen és így nem kell még egyszer megírni a kívánt kódot. A „var image” segítségével egy tömböt hoztam létre, amiben a megjeleníteni kívánt képeket tárolom. Ez tetszőleges hosszúságú lehet. A tömb létrehozásánál meg kell adni a képek nevét, amiket én macska1-macska2... módszerrel készítettem el egészen tízig, vagyis tíz képet tartalmaz a képvtő tömb. Az „i” változó értékét a .Length függvénnyel adtam meg. A .Lenght függvény rengeteg programozási nyelvben szerepel és mindig egy adott változó, tömb vagy lista elemeinek a számát számolja meg. A „slider_content.innerHTML” sor segítségével tudjuk megjeleníteni a képeket mivel tényleges ez a sor váltogatja a képeket.

```
function prewImage(){  
    if (i<image.length+1 && i>1) {  
        i= i-1;  
    }else{  
        i = image.length;  
    }  
  
    // Előző képre történő hivatkozás  
    slider_content.innerHTML = "";  
}  
  
// 4000 miliszekundum után magától vált egy képet előre
```

```
setInterval(nextImage , 4000);
```

Ez a visszafele váltó gombnak a function-je. Szinte teljesen megegyezik az az előrefele ugró gombnak a kódjával csak itt annyi változás van, hogy az „i<image. length+1” után került egy logikai operátor a &&. Ez az „és” logikai operátor, ami segítségével egy „if ciklusba” tudunk helyezni egy olyan feltételt, ami vizsgálja azt, hogy „A” feltétel és „B” feltétel teljesül csak akkor fut le a ciklus. A legvégén pedig a „setInterval” nevezetű kód található, ami a „nextImage” function-re hivatkozik, vagyis a következő kép jelenjen meg és a vessző után az időtartamot lehet beállítani, amit én négyezer milliszekundumra állítottam. A többi oldal kódjával nem fogok foglalkozni, mert semmi összetettebb kód nincs benne ezért nem érzem szükségét, hogy foglalkozzak vele ehelyett inkább rátérek a bejelentkező oldal kódjának magyarázásába.



A bejelentkező felület így néz ki. Felhasználónevet és jelszót kell megadni, de viszont abban az esetben, ha a felhasználónak még nincs regisztrált fiókja akkor a „Kattintson ide a regisztráció” szöveget megnyomva a weboldal átirányítja a regisztrációs felületre. Kicsit nehezen észrevehető, de a bal felső sarokban található a visszalépés a főoldalra gomb. Ha a felhasználó érvényes adatokat ad meg akkor a rendszer automatikusan bejelentkezteti és rögtön visszairányítja a Kezdőlapra. Viszont, ha rossz adatokat adott meg akkor a rendszer hibaiüzenetet jelez. Regisztráció során a felhasználónevet, e-mailt és kétszer a kívánt jelszót kell megadni. A rendszerbe nincs olyan funkció beépítve, ami ellenőrzi azt, hogy a felhasználó csak olyan felhasználónevet tudjon megadni, ami még nem foglalt. Ha az e-mail

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

cím érvénytelen akkor a böngésző hibaüzenetet jelez viszont, ha a két jelszó nem egyezik
akkor a weblap.

```
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
  <title>A Macska Király - Bejelentkezés</title>
  <!-- Stíluslapra történő hivatkozás -->
  <link rel="stylesheet" type="text/css" href="regisztráció.css">
</head>
<body>
  <!-- Vissza a főoldalra gomb -->
  <p class= fooldal Tomb><a href="index.html"> Vissza a főoldalra </a></p>
  <div class="header">
    <h2>Bejelentkezés</h2>
  </div>
  <!-- Form létrehozása -->
  <form method="post" action="login.php">

    <?php include('errors.php'); ?>

    <div class="input-group">
      <!-- Felhasználónév mező -->
      <label>Felhasználónév</label>
      <input type="text" name="username" >
    </div>
    <div class="input-group">
      <!-- Jelszó mező -->
      <label>Jelszó</label>
      <input type="password" name="password">
    </div>
    <div class="input-group">
      <!-- Elküldés -->
      <button type="submit" class="btn" name="login_user">Bejelentkezés</button>
    </div>
    <p>
      Még nincs fiókja? <a href="register.php">Kattitson ide a regisztrációhoz</a>
    </p>
  </form>

</body>
</html>
```

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

Egy form-ot hoztam létre, ami a login.php-re hivatkozik. Igazából az egész bejelentkezés egy form amiken input mezőket használtam, hogy a kívánt adatokkal tudjon foglalkozni a program. A form-ok arra jók, hogy szövegbeviteli mezőket hozzunk létre, amik adataival tudunk a későbbiekben dolgozni. A HTML egyik legjobb tulajdonsága, hogy .html kiterjesztésű fájlba tudunk beleírni PHP kódot és így nem kell külön fájlt létrehozni a PHP számára. A regisztrációs űrlap szinte ugyanígy néz ki szóval azzal nem szeretnék foglalkozni inkább rátérek a server.php fájlra, ami a localserver-rel kapcsolatos dolgokat látja el.

```
<?php
    session_start();

    // változó létrehozása
    $username = "";
    $email     = "";
    $errors = array();
    $_SESSION['success'] = "";

    // csatlakozás az adatbázishoz
    $db = mysqli_connect('localhost', 'root', '', 'registration');

    // Felhasználó regisztrálása
    if (isset($_POST['reg_user'])) {
        // Formról beérkező adatok
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $email = mysqli_real_escape_string($db, $_POST['email']);
        $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
        $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

        // Minden bemeneti adat helyes-e
        if (empty($username)) { array_push($errors, "Felhasználónév megadása szükséges!"); }
        if (empty($email)) { array_push($errors, "Email cím megadása szükséges!"); }
        if (empty($password_1)) { array_push($errors, "Jelszó megadása szükséges!"); }

        if ($password_1 != $password_2) {
            array_push($errors, "A jelszavak nem egyeznek!");
        }

        // Ha nincs semmi hiba akkor a felhasználót regisztrálja
        if (count($errors) == 0) {
            $password = md5($password_1); // Jelszó titkosítása
            $query = "INSERT INTO users (username, email, password)";
```

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám


```
VALUES('$username', '$email', '$password');"
mysqli_query($db, $query);

$_SESSION['username'] = $username;
$_SESSION['success'] = "Be lett jelentkeztetve";
header('location: index.php');
}

}
```

Legelőször is a XAMPP által üzemeltett PHPMyAdmin-ban kell létrehozni egy új adatbázist, abban új táblát, amiben majd a PHP kód tudja tárolni az adatokat. A PHP kódban pontosan ugyanazokat a neveket kell megadni, amiket a PHPMyAdmin-ban létrehoztunk mert különben nem fogja tudni eltárolni az adatokat. Legelőször a változókat kell létrehozni azután pedig az adatbázishoz való kapcsolódást kell megadni. Egy „if” ciklusba helyezzük az egész regisztráció folyamatát. Négy sorban a form-ról beérkező adatokat tároljuk el helyi változóba, ami nagyon fontos hiszen e nélkül a kapott adatokkal nem tudnánk dolgozni. A jelszóból azért van „password1” és „password2”, mert jelszó megerősítéses a regisztráció és a „password2” arra szolgál, hogy az elsőnek beírt jelszót megismételjük ezzel kivédekezve azt a hibát, hogy különböző jelszavakat adunk meg. Ezek után a bemeneti adatokat ellenőrizzük. Ha a mezők nincsennek kitöltve akkor a weblap hibaüzenetet dob ki, hogy ellenőrizzük a beírt adatok helyesek-e. Ezek után a két jelszó egyezését vizsgálja, vagyis a „password1” és a „password2” -nek egyeznie kell, ha nem egyezik szintén hibaüzenetet dob ki. Viszont, ha semmilyen hibát nem vétett a felhasználó akkor a rendszer elfogadja a regisztrációt és a jelszót MD5-ös titkosítás segítségével titkosítja és eltárolja ezt a karakterláncot, amit az MD5 hozott létre az adatbázisban. Az MD5 egy 128 bites, egyirányú kódolási algoritmus. Az RFC 1321-es internet szabványban foglaltak szerint használják internetes adatok kódolására, illetve titkosítási kulcsokban. Bármilyen adatból – függetlenül a méretétől vagy a típusától – egy 16 bájt hosszú bináris hash-t eredményez (32 hexadecimális számjeggyel írható le). A kódolás egyirányú, így nem lehet visszafejteni. Éppen emiatt alkalmas eredetiség ellenőrzésre. Manapság már ezt a titkosítást kevés helyen használják mivel elég sebezhető titkosításról van szó és már léteznek az informatika világában sokkal megbízhatóbb titkosítási rendszerek. Én azért használtam az MD5-öt mivel egyszerű beépíteni egy PHP kódba és nem hiszem, hogy még jelenleg a weboldalamon tárolt

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

jelszavaknak erősebb védelemre lenne szükségük. Ezek után a „\$query” változó segítségével az adatbázis tábláját feltöltjük a form-ról kapott adatokkal. Ha mindez megtörtént után a weblap átirányítást végez a Kezdőlapra.

```
// Felhasználó bejelentkeztetése
if (isset($_POST['login_user'])) {
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $password = mysqli_real_escape_string($db, $_POST['password']);

    if (empty($username)) {
        array_push($errors, "Username is required");
    }
    if (empty($password)) {
        array_push($errors, "Password is required");
    }

    if (count($errors) == 0) {
        $password = md5($password);
        $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
        $results = mysqli_query($db, $query);

        if (mysqli_num_rows($results) == 1) {
            $_SESSION['username'] = $username;
            $_SESSION['success'] = "Bejelentkezett";
            header('location: index.php');
        }else {
            array_push($errors, "Rossz jelszó és felhasználónév kombináció!");
        }
    }
}
```

Ha a felhasználónév és a jelszó kitöltetlenül maradt akkor a hibát jelez a weblap. Viszont, ha nem történt semmi hiba akkor a beírt jelszó „hash” karakterláncát összehasonlítja az adatbázisban tárolt jelszó karakterláncával és ha ezek megegyeznek akkor megtörténhet a bejelentkezés. A „\$query” változóval az adatbázisból lekéri az adatokat, hogy azok egyeznek-e, ha egyeznek akkor megtörtént a bejelentkezés. Sikeres bejelentkezés után a weblap a Kezdőlapra irányít.

```
.header {
  margin: 50px auto 0px;
  color: black;
  background: transparent;
  text-align: center;
  border: 1px solid #B0C4DE;
  border-bottom: none;
  border-radius: 10px 10px 0px 0px;
  padding: 20px;
}

.input-group {
  margin: 10px 0px 10px 0px;
}

.input-group input {
  height: 30px;
  width: 93%;
  padding: 5px 10px;
  font-size: 16px;
  border-radius: 5px;
  border: 1px solid gray;
}

.btn {
  padding: 10px;
  font-size: 15px;
  color: white;
  background: #5F9EA0;
  border: none;
  border-radius: 5px;
}

.error {
  width: 92%;
  margin: 0px auto;
  padding: 10px;
  border: 1px solid #a94442;
  color: #a94442;
  background: #f2dede;
  border-radius: 5px;
  text-align: left;
}

.success {
  color: #3c763d;
  background: #dff0d8;
  border: 1px solid #3c763d;
  margin-bottom: 20px;
}
```

Ez a registration.css ami az összes regisztrációs laphoz tartozó .css fájl. A „.header” osztály a weboldal fejrészában található dolgok formázására jó. A háttér áttetsző, szövegek középre vannak igazítva a text-align sor segítségével, betűszínük pedig fekete. A keret (border) 1 pixel vastagságú „solid” stílusú, ami #B0C4DE hexadecimálisan megadott színű. A „border-radius” a keret széleinek lekerekítéséért felel (Az első érték a bal felső sarokra szól, a második a jobb felső sarok, a harmadik a jobb alsó sarok és a negyedik az a bal alsó sarokra szól).

5. Továbbfejlesztési lehetőség

Sok mindenben lehetne még továbbfejleszteni az oldalamat, de szerintem a legfontosabb fejlesztés az oldalam számára, hogy a felhasználók egy fórum segítségével tudnának egymással beszélgetni, tippeket és tanácsokat megosztani egymással. Nem sikerült azt megvalósítanom, hogy ha egy felhasználó bejelentkezik akkor az minden egyes oldalon mutassa (igazából a rendszer ténylegesen belépteti a felhasználót és az adatait eltárolja az adatbázisba csak a PHPMyAdmin-ban tudom látni, hogy új értékek lettek hozzáadva a táblához), ezt a majd idővel megpróbálom beépíteni a weblapomba. Másik funkció az lenne, amit szívem szerint beépítenék, hogy az oldalt látogató emberek képeket tudjanak feltölteni a macskájukról és az a Fotók aloldalon lévő képnézegetőbe automatikusan megjelenjen úgy, hogy kiírja azt, hogy melyik felhasználó töltötte fel ezt a képet az oldalra. Reszponzívvá is lehetne tenni az oldalt, ami nagy segítség lenne azoknak a felhasználók számára, akik mobiltelefonról vagy táblagépről próbálják elérni a weblapomat. Eddig csak a Kezdőlap a rezponzív. A rezponzívitás azt jelenti, hogy egy oldal tud alkalmazkodni egyes készülékekhez és aszerint tudja a tartalmat megjeleníteni anélkül, hogy az weblap tartalma elcsússzon és minden össze-vissza legyen. Egy oldallátogató számlálót is beépítenék, amivel tudom követni azt, hogy egyes időközönként hány látogatója volt az oldalnak. Másik nagyon fontos szempont még az lenne, hogy a weblap egy rendes szerverről fusson ne csak egy lokális kiszolgáló által fusson, mert így csak az tud hozzáférni a weblapomhoz, aki velem egy hálózatban tartózkodik. Persze az ilyen szerverek pénzbe kerülnek és természetesen a szakdolgozatomat nem fogom az interneten mindenki számára elérhetővé tenni, esetleg majd csak akkor, ha a fentebb említett funkciókat beleépítettem a weblapomba

és úgy érzem, hogy ez a weblap készen áll arra, hogy a nagyközönség számára is elérhető legyen.

6. Tesztelési dokumentáció

7. Rendszerkövetelmény

A minimum rendszerigény a következő:

- Processzor: Intel i3-4130 2.90 Ghz
- Memória: 2Gb DDR3
- Videókártya: Bármilyen processzorba integrált videokártya
- Háttértár: 128Gb HDD
- Operációs rendszer: Windows 10 Home 1705-ös verziószám

Az általam használt számítógép alkatrészeit szabtam meg, mint ajánlott rendszerigényt:

- Processzor: Intel i5-3470 3.20 Ghz
- Memória: 12Gb DDR3
- Videókártya: Nvidia GeForce GTX960 2Gb
- Háttértár: 1TB + 320Gb HDD
- Operációs rendszer: Windows 10 Pro 1909-es verziószám

Szoftverkövetelmény:

A lehető legjobb élmény és megjelenítés érdekében érdemes ezeket a szoftvereket használni:

- Androidos okostelefon esetében a minimum Android verziószám a 7.0-ás rendszer, az ajánlott a legújabb 10.0-ás rendszer használata.
- IOS rendszer esetében a 9-es verziószám a minimum, ajánlott pedig az IOS 13.

Számítógép esetén ezekkel a szoftverekkel kell rendelkezni ahhoz, hogy a weblaphoz hozzátudjunk férni:

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

- Microsoft Windows 10
- XAMPP
- Google Chrome, Mozilla Firefox és Microsoft Edge (értelmesebb az Edge-ből a legfrissebb kiadást használni, ami már Chromium alapokra épül és ennek köszönhetően a weblap kinézete egy az egyben megegyezik a Chrome által mutatott képpel)

8. Felhasználói dokumentáció

Kedves felhasználó, beszeretném számodra mutatni a weblapom használatát, hogy effektíven, gyorsan és könnyedén tudd használni az oldalt. Érdemes az oldal használatához Google Chrome böngészőt alkalmazni mivel én a böngésző segítségével teszteltem mindig az oldalt. Az oldalt megpróbáltam a modern weblap dizájnok szerint fejleszteni, ami azt jelenti, hogy nem használok „elavult” egyszínű háttérrel és régimódi betűtípust. Először is, amikor a weblapom betöltődött egy kezdőlap fogad téged, amivel minden egyes aloldalra tudsz navigálni. Érdemes a bejelentkezéssel kezdeni. Kattints a „BEJELENTKEZÉS” fülre és valószínűleg még nem vagy regisztrált tag ezért tedd meg azt, hogy rákattintasz a „Kattintson ide a regisztrációhoz” és ott add meg a felhasználónevedet, e-mail címedet és kétszer meg kell erősítened a kívánni használt jelszavadat. Ha minden megadott adat megfelel akkor a rendszer létrehozza a fiókját és egyben be is jelentkezteti. Ezek után automatikusan pár másodperc után vissza lesz irányítva a Kezdőlapra, ahonnan elkezdheti böngészni az oldalamat. Fotók oldalon egy képnézegető várja a látogatókat, amiben a legcukibb macskás képeket próbáltam összegyűjteni. A „TUDNIVALÓK” oldalon a macskákról általánosan írtam, hogy az ember hogyan házasította meg őket, mióta élnek emberek társaságában. A következő oldal, ami szerintem a legérdekesebb és legizgalmasabb, az „ÉRDEKESSÉGEK”, ahol húsz darab tényt gyűjtöttem össze a macskákról és fontos, hogy az oldal alján található egy gomb amire kattintva átvizsgáljuk a következő 10 tényre. Megpróbáltam olyan tényeket összeszedni, amiket még nem nagyon láthatott semmilyen más oldalon. „VISELKEDÉSÜK” oldalon pedig a macskák általános viselkedéséről gyűjtöttem össze információkat például, hogy mit tegyünk azért, hogy a macskánk jól érezze

magát nálunk. Ennyit funkciót tartalmaz a weblapom, de koránt sem ennyit fog mindig tartalmazni. Vannak tervek a továbbfejlesztéséhez, amiket a jövőben meg fogok valósítani.

9. Összegzés

Még szeptemberben amikor elkezdtem a tanévet és közölték velünk, hogy szakdolgozatot kell írunk (egy játékot vagy weblapot) őszintén meg ilyedtem, hogy én milyen módon leszek képes arra, hogy egy kész weboldalt elkészítsek. De aztán az idő előrehaladtával egyre jobban csökkent benenem ez a félelemérzet. Amikor elkezdtem a szakdolgozatom megírásához nem is tudtam, hogy mit kezdjek magammal annyi ötletem volt, melyiket valósítsam meg először. Azért macska témájú oldalt hoztam létre mivel imádom a macskákat és gondoltam, hogy ha ilyen témájú weblapot kell készítenem akkor az könnyeben fog menni mivel egy olyan témával kell foglalkoznom, ami mindig is nagyon érdekelt. Tisztába vagyok vele, hogy még sok funkció hiányzik belőle, ami sokkal jobba tehetné az oldalamat, de ezeket muszáj voltam kihagyni belőle mivel nem tudtam volna ezeket elkészíteni és ráadásul a szakdolgozatra szánt idő sem volt a legtöbb. A szakdolgozatom elkészítése alatt a legnehezebb akadály, amibe beleütköztem az a PHP adatbázis volt. Több heti munkával és nagyon sok videós, osztálytársak segítségével sikerült leküzdenem ezt az akadályt. A HTML, CSS és JavaScript kódokat meglepően egyszerűen meg tudtam írni ahhoz képest, hogy szeptemberben amikor még kezdtem az évet akkor még nem sok mindent tudtam volna magamtól megírni, de aztán több mint fél évnyi tanulás után sikerült egy kész weblapot készítenem, ami hatalmas nagy öröm és büszkeség számomra, hogy ezt a kihívást tudtam teljesíteni. Minél jobban haladtam előre a szakdolgozatommal annál nagyobb volt az önbizalmam a szakdolgozattal szemben és ez az önbizalom hajtott egyre tovább és tovább, hogy elkészítsem a weblapot olyanra amilyenre én megálmodtam. Hasznos volt ez a projekt számomra mivel rengeteg mindent tanultam meg magamtól és most már sokkal jobban értek egy weblaphoz és egy adatbázishoz mint, egy évvel ezelőtt. Remélem ezt a megszerzett tudást tudom majd máshol hasznosítani például a ránk váró vizsgákon, ami után megszerezzen az OKJ-s szoftverfejlesztői szakmát. Persze azzal teljes mértékben tisztában vagyok, hogy tanulás itt nem állt meg még ennél sokkal, de sokkal több mindent kell tudnom

ahhoz, hogy egy kiváló informatikai szakember legyek. Mivel nekem célom, hogy tanulmányaimat folytathassam felsőoktatásban én úgy gondolom, hogy ez a ráadás egy év OKJ-s képzéssel nem jártam rosszul mivel ez egy biztos alapot adott ahhoz, hogy egyetemen tudjam folytatni a tanulmányaimat.

10. Mellékletek

10.1 Felhasznált képek

macska1.jpg <https://www.pexels.com/photo/light-landscape-sky-sunset-35888/> (2020. március 29. 12:57)

macska2.jpg <https://www.pexels.com/photo/adorable-animal-bed-bedroom-236606/> (2020. március 29. 13:01)

macska3.jpg <https://www.pexels.com/photo/yawning-cat-1576193/> (2020. március 29. 13:03)

macska4.jpg <https://www.pexels.com/photo/white-and-black-cat-156934/> (2020. március 29. 13:05)

macska5.jpg <https://www.pexels.com/photo/russian-blue-kitten-on-brown-woven-basket-127027/> (2020. március 29. 13:09)

macska6.jpg <https://www.pexels.com/photo/close-up-photography-of-white-cat-besides-christmas-lights-735423/> (2020. március 29. 13:12)

macska7.jpg <https://www.pexels.com/photo/adorable-animal-baby-british-shorthair-460797/> (2020. március 29. 13:20)

macska8.jpg <https://www.pexels.com/photo/adorable-animal-baby-cat-290164/> (2020. március 29. 13:22)

macska9.jpg <https://www.pexels.com/photo/adorable-animal-cat-cat-s-eyes-357141/> (2020. március 29. 13:25)

macska10.jpg <https://www.pexels.com/photo/cat-biting-persons-finger-1202481/> (2020. március 29. 13:30)

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

macskak.jpg <https://www.pexels.com/photo/kitten-cat-rush-lucky-cat-45170/> (2020.
március 29. 14:58)

macskafan.jpg <https://www.pexels.com/photo/photo-of-gray-cat-on-tree-branch-1693441/>
(2020. március 29. 15:00)

ulomacska.jpg <https://www.pexels.com/photo/close-up-photo-of-tabby-cat-1457792/>
(2020. március 29. (15:12)

egymacska.jpg <https://www.pexels.com/photo/asian-semi-longhair-cat-on-grass-1296660/>
(2020. március 29. 15:10)

macskafog.jpg <https://www.pexels.com/photo/animal-kitten-cat-funny-56857/> (2020.
március 29. 16:20)

feketemacska.jpg <https://www.pexels.com/photo/black-cat-walking-on-road-1510543/>
(2020. március 29. 16:48)

bajusz.jpg <https://www.pexels.com/photo/depth-of-field-photograph-of-white-and-black-kitten-1120049/> (2020. március 29. 17:09)

gradient.jpg <https://unsplash.com/photos/eICUFSeirc0> (2020. március 3. 14:02)

10.2 Felhasznált források

A macskákról való tudnivalók és érdekességek:

- <https://hu.wikipedia.org/wiki/Macska> (2020. március 15. 17:10)
- <https://www.origo.hu/tafelspicc/20130418-a-macska-helyes-etetese-taplalasa.html>
(2020. március 16. 12:46)
- <https://www.zooplus.hu/magazin/macska-magazin/macskatartas> (2020. március 15.
17:30)
- <https://divany.hu/offline/2019/05/05/boldog-macska-viselkedese/> (2020. március
16. 18:32)
- <https://www.zooplus.hu/magazin/macska-magazin/macskatartas/a-boldog-macska-viselkedese> (2020. március 16. 19:22)

Szakdolgozat címe: A Macska Király
Készítette: Kajati Ádám

V. István Katolikus Szakgimnázium És Gimnázium, Árpád-házi Szent Margit
Általános Iskola, Óvoda

- <https://mivoltma.444.hu/2017/04/21/23-erdekes-teny-a-cicakrol-amiket-eddig-talan-nem-tudtal> (2020. március 14. 15:07)
- <https://nlc.hu/eletmod/20141029/20-erdekes-teny-a-macskakrol/> (2020. március 17. 18:09)
- <https://starity.hu/magazin/49164-79-oruletes-dolog-amit-nem-tudtal-a-macskakrol/> (2020. március 18. 22:06)

Kezdőlap <https://www.youtube.com/watch?v=wVSjKfLKvbK> (2020. március 2. 15:08)

Képvetítő <https://www.youtube.com/watch?v=sQuniQCzNIY> (2020. március 2. 18:44)

Regisztráció és a PHP adatbázis <https://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database> (2020. március 8. 23:05)

Menü <https://www.youtube.com/watch?v=mFgltjfVV6A> (2020. március 2. 16:59)

HTML <https://webshark.hu/hirek/html/#toc0> (2020. január 10. 16:35)

CSS <https://webshark.hu/hirek/html/#toc0> (2020. január 10. 17:01)

JavaScript <https://hu.wikipedia.org/wiki/JavaScript> (2020. január 10. 17:04)

PHP <https://hu.wikibooks.org/wiki/PHP> (2020. január 10. 17:09)

SQL <https://hu.wikipedia.org/wiki/SQL> (2020. január 10. 18:01)

Microsoft Visual Studio Code https://hu.wikipedia.org/wiki/Visual_Studio_Code (2020. január 11. 16:28)

Sublime Text 3 https://en.wikipedia.org/wiki/Sublime_Text (2020. január 11. 17:01)

XAMPP <https://hu.wikipedia.org/wiki/XAMPP> (2020. január 11. 16:35)

MD5 <https://hu.wikipedia.org/wiki/MD5> (2020. március 22. 19:55)

Reszponzív https://www.w3schools.com/html/html_responsive.asp (2020. március 14. 22:55)

JavaScript HTML-ben https://www.w3schools.com/html/html_scripts.asp (2020. március 12. 17:36)

HTML szöveg formázás https://www.w3schools.com/html/html_formatting.asp (2020.
március 5. 18:09)

JavaScript functions https://www.w3schools.com/js/js_functions.asp (2020. március 7.
22:01)

JavaScript tömb https://www.w3schools.com/js/js_arrays.asp (2020. március 8. 17:34)

Tartalom:

<i>1. A témaválasztás indoklása.....</i>	<i>2</i>
<i>2. Fejlesztői dokumentáció.....</i>	<i>3</i>
<i>3. Használt programozási nyelvek bemutatása és a szoftver választás indoklása.....</i>	<i>4</i>
<i>3. Weboldal navigációjának ismertetése.....</i>	<i>7</i>
<i>4. Weboldal oldalainak kódjának bemutatása.....</i>	<i>11</i>
<i>5. Továbbfejlesztési lehetőség.....</i>	<i>20</i>
<i>6. Tesztelési dokumentáció.....</i>	<i>21</i>
<i>7. Rendszerkövetelmény.....</i>	<i>21</i>
<i>8. Felhasználói dokumentáció.....</i>	<i>22</i>
<i>9. Összegzés.....</i>	<i>23</i>
<i>10. Mellékletek.....</i>	<i>24</i>

10.1	<i>Felhasznált képek</i>	24
10.2	<i>Felhasznált források</i>	25