

JEGYZŐKÖNYV

Számítógép Architektúrák

Féléves feladat

1.Feladat

Készítette: **Ferencsik Róbert**

Neptunkód: **BQLOTW**

Dátum: 2023.11.30

Tartalomjegyzék 1.Feladat

1. Bevezetés.....	3
1.1. Dokumentum célja.....	3
1.2. Saját gondolatok.....	3
2. Követelmény elemzés.....	3
2.1. Szerkezet és kötelező elemek.....	3
2.2. Döntés a tartalomról.....	3
3. Dizájn.....	4
3.1. Elemek elhelyezkedése.....	4
3.2. Színek és grafikai elemek.....	4
4. Fejlesztés és tesztelés.....	5
4.1. Index.....	5
4.2. Többi HTML és CSS.....	5
4.3. Javascript.....	5

Tartalomjegyzék 2.Feladat

1. Bevezetés.....	6
1.1. Dokumentum célja.....	6
1.2. Saját gondolatok.....	6
2. Követelmény elemzés.....	6
2.1. A feladat kiírás és értelmezése.....	6
2.2. Döntés a tartalomról.....	6
3. A szkript elemei és feladatuk.....	7
3.1. Mappa ellenőrzése elkészítése.....	7
3.2. Zip törlése.....	7
3.3. Fájlok törlése.....	7
3.4. Letöltés, kibontás, és ezek sikeressége... ..	8
3.5. Szűrés.....	8
4. Fejlesztés és tesztelés.....	8

Bevezetés

A számítógép Architektúrák első beadandó feladatának dokumentációját tartja kezében. A feladat maga egy statikus weboldal elkészítése volt. Ebben alkalmazni kellett HTML4/4, CSS illetve JavaScript elemeket. Ennek a feladatnak a megoldásáról olvashat az alábbiakban.

Izgalmas és kihíváskkal teli feladat volt, mivel nem csak a kódolási tudás szükséges, hanem a tervezés és a kreativitás is nagy szerepet élvez egy weboldal elkészítésének folyamata során. Maga a feladat kiírás nem volt túl bonyolult, és személy szerint nem estem túlzásokba a funkcionalitás terén, a leírás elemzése után mégis úgy gondolom minden elem megtalálható amelynek meg kell jelennie az oldalamon. Igyekeztem inkább a színeket és a megjelenést össze harmonizálni.

Követelmény elemzés

A feladatban szükséges volt HTML CSS és JavaScript kódokat alkalmazni. Első lépésként a feladat kiírást elemeztem, és az alapján gondoltam végig, hogy mégis milyen elemekből, mennyi darab fájlból (és milyenekből) fog állni az elkészült oldal. Az első .html a főoldalé, mely három kurzus menüpontját tartalmazza. Ebből adódóan a következő három az aloldalaké, és még a főoldalt ketté vettem egy index fájl és egy főoldal részre. Az index fájl tartalmazza az oldalak felépítését (header, menü, lábléc). Egy CSS fájlt is alkalmaztam, igaz magában a HTML kódba ágyaztam is egykét formázást, és a JavaScriptjeimet is külön fájlba vettem.

Egy mappát létrehoztam a média fájloknak, és utána jött az első döntés. Ki kellett választani a három kurzust, mert azok alapján neveztem volna el a html fájlokat. Gyors döntés volt, olyan

tárgy amit tanultam az idei éven és utána csak választottam hármat: Diszkrét Matematika, Számítógép Architektúra és Programozás Alapjai.



A kurzusokhoz a tematikát a ME GEIK oldaláról letölthető illetve megtekinthető dokumentumokból néztem. Illetve a főoldalon is elhelyeztem erről ezt a linket ([Gépészmérnöki és Informatikai Kar \(ME-GEIK\) - Oktatott tárgyak \(uni-miskolc.hu\)](http://uni-miskolc.hu))

Dizájn

Csak egy egyszerű vázlatot készítettem az elemek elhelyezkedéséről mielőtt nekiláttam volna a kódolásnak. A header témáját megválasztottam, a menüpont elhelyezkedését, körülbelüli méreteit az oldalhoz képest, és a bodyban a szöveg és média helyét. Természetesen kódolás közben megoldást kellett találnom sok témára amivel nem számoltam a dróthálót és webdesignot egybefogó vázlatom készítése közben.



Vázlat a Dizájnról

Mégis a feladat megoldásomban nagy szerepet kapott később a megjelenés. A kék szín mellett már a vázlat készítésekor döntöttem, utána jött a neheze. Össze gyűjtöttem grafikai elemeket az elképzelt színvilág mellett, és utána olvastam a kontrasztoknak. Későn realizáltam, hogy a gépemen nincs képszerkesztő program, így online szerkesztőt használtam. A képek szerkesztéséhez a OIE-Online Image Editor-t, favicon készítéséhez pedig a favicon.io oldalt.

Favicon.io: [The best Favicon Generator \(completely free\) - favicon.io](https://favicon.io)

OIE: [Free Online Image Editor \(online-image-editor.com\)](https://online-image-editor.com)

Fejlesztés és tesztelés

Az index.html kódját írtam meg először. Ebben helyezkedik el a weboldalam fej (a menüvel) és láblece. Egy iframe használatával jelenítem meg benne a többi oldal tartalmát ami igazából egy oldalon belül segít megjeleníteni egy másik oldalt. Ennek az alapértelmezett forrása a fooldal.html.

```

13     <tr style="...">
14         <td></td>
15         <td class="menu-container" onmouseover="showMenu()" onmouseout="hideMenu()" colspan="3" style="...">
16             <a href="fooldal.html" target="keret"></a>
17         </td>
18     </tr>
19
20
21     <tr class="menu-items" id="menuItems" onmouseover="showMenu()" onmouseout="hideMenu()" >
22         <td class="padding"></td>
23         <td class="menu-cell"><a href="diszkret.html" target="keret">Diszkrét matematika</a></td>
24         <td class="menu-cell"><a href="architektura.html" target="keret">Számítógép architektúrák</a></td>
25         <td class="menu-cell"><a href="progalapjai.html" target="keret">Programozás alapjai</a></td>
26         <td class="padding"></td>
27     </tr>
28

```

A menü kódja az index fájlban

Elkészítettem a főoldalt és egy kurzus kódját is. Először tartalom nélkül és formázás nélkül, készítettem bele egy javascriptet ami a menüt változtatta meg lenyílóvá. Utána kezdtem bele a CSS megírásába. Itt össze fonódott a tartalom felvitele és a kódírás, mivel látni szerettem volna a változtatásaim. Itt jött elő, hogy a háttér nagy kontrasztja miatt változtatni kell a szöveg színén és a mögötte lévő háttér áttetszőségén.

Egy videót is helyeztem el az egyik aloldalon, ez szinte az utolsó lépés volt és így bővítettem a javascriptjeimet is, mivel letisztultabbnak találtam, ha saját gombokat hozok létre hozzá a keretén kívül. A tesztelést nem vettem külön, igazából mikor elkészítettem egy kód részletet megnéztem az eredményt, illetve kipróbáltam, ha valamilyen funkcionalitással rendelkezett.

Összességében azt tudom mondani, hogy a feladat megoldása közben tanultam. Sokszor előfordult, hogy nem tudtam elsőre elkészíteni az ötleteim, azonban a w3schools és pár fórum böngészése megoldást nyújtott.

W3schools: [W3Schools Online Web Tutorials](https://www.w3schools.com/)

Bevezetés

A számítógép Architektúrák második beadandó feladatának dokumentációját tartja kezében. A feladat bash script elkészítése volt. A program működéséről írtam részletesebben, és az elkészítés folyamatáról is szó lesz alább.

Mivel betegség miatt egyedül készültem fel így kihívásnak bizonyult a feladat, azonban türelemmel és próbálkozások sorával elkészítettem az első saját szkriptemet. Nagyon izgalmas és motiváló gondolat, hogy a saját eszközömmön az unalmas egyszerű, és sokszor elvégzendő feladatokat automatizálhatom, ezzel kiváltva őket, egy egyszeri összetettebb programozási feladatra, majd annak az elkészítése után egy egyszerű futtatásra.

Követelmény elemzés

A feladat kiírása alapján az MVK Zrt GTFS adatbázisát kellett a szkriptnek letöltenie, amely egy zip fájl. Szóval kitömörítés után rá kellett szűrnie a letöltött adatbázisok egyikében azokra az útvonalakra mely a saját lakóhelyhez legközelebbi megállóból indul, vagy odaérkezik.

Mind ezek mellett ellenőrzéseket is kellett végeznie, még hozzá sorban a következőket. Ha a letöltendő fájl már egyszer le volt töltve, akkor az újbóli letöltés előtt törölje az előző verziót! Ha egy mappába már korábban ki lett tömörítve a letöltött állomány, akkor az újbóli kitömörítés törölje a mappa tartalmát!

2. feladat – Mentés: *lakohely_gtfs.sh*

Az MVK Zrt. elérhetővé tesz egy szabványos GTFS adatbázist a fejlesztők számára, hogy a *menetrendi adatokat* a saját applikációba tudják integrálni.

Írjon egy shell script fájlt, amely *letölti ezt az adatbázist, és kilistázza belőle azokat az utakat, amelyek a Lakóhelyből* indulnak, ill. a *Lakóhelybe* érkeznek!

Parancsok: *wget, unzip (kitömörítésre), cat, grep*

Szükséges ellenőrzések!

Ha a letöltendő fájl már egyszer le volt töltve, akkor az újbóli letöltés előtt törölje az előző verziót!

Ha egy mappába már korábban ki lett tömörítve a letöltött állomány, akkor az újbóli kitömörítés törölje a mappa tartalmát!

Ha nincs ilyen menetrendi adatok, akkor Miskolc várost használják a feladat elkészítéséhez!

Az eredeti feladat kiírás szövege

A szkriptet lakohely_gtfs.sh néven kell menteni, és beadáskor látszódnia kell a lefutott program eredményének, amit én úgy oldottam meg, hogy .txt fájlba írja ki a szűrés eredményét a

programom. Így látszódik, hogy sikeresen megtörtént a szűrés, de igazából az elérési útvonal változóban van tárolva, úgyhogy annak a módosításával lehet használni másik eszközön is.

A szkript elemei és feladatuk

Ahogy már írtam a jegyzék útvonala változóban van tárolva *output_folder* néven, így könnyen kezelhető a programon belül és a letöltendő fájl url címe is változóban van tárolva, így letisztultabb és könnyen változtatható a program.

Az első lépésben ellenőrzésre kerül egy if feltételében, hogy a jegyzék amiben szeretnénk “dolgozni” létezik-e. Maga a feltétel pontosan így szerepel a kódban: `[! -d "$output_folder"]` ahol a felkiáltójel negálja az értelmét annak, hogy a dollár jel után lévő változóban tárolt nevű fájl létezik, és jegyzék amit a *-d* vel adunk meg a programnak. Természetesen ha az előbbi feltétel igazat ad vissza, az azt jelenti, hogy nincs ilyen nevű jegyzék, és így belépve a feltételbe egy *mkdir* paranccsal elkészít azt.

Majd a *cd* megváltoztatja az aktuális munka könyvtárat a GTFS nevű vagy létező vagy létrehozott jegyzékre. A `||` vagyfeltétel után lévő *exit* pedig kilép a végrehajtásból, ha nem sikerült a könyvtár váltás.

A következő két feltételes elágazás sorban a következőkért felelős: Vizsgálat, hogy a letölteni kívánt tömörített fájl létezik-e (*-e* avagy *exists* szűréssel), amennyiben létezik, kiírja a konzolra, hogy mit csinál épp, és törli a fájlt magát. Majd ez megtörténik a mappa tartalmára is, azt nézi a program feltétele, hogy létezik-e elem a munkakönyvtárban (`["$(ls -A .)"]` nem veszi figyelembe a speciális *..* fájlokat). Ha a kiértékelés igaz, akkor *rm -r ** al rekurzívan törli a munkakönyvtárban lévő összes fájlt és alkönyvtárat.

```
if [ -e "downloaded_file.zip" ]; then
    echo "Deleting previous version of downloaded ZIP file."
    rm downloaded_file.zip
fi

if [ "$(ls -A .)" ]; then
    echo "Deleting previous content of the extraction folder."
    rm -r *
fi
```

A fent említett két feltételes elágazás

A letöltés után egymásba ágyazva két feltétel következik melyek csak akkor futnak le ha az előttük lévő utasítás vissza térési értéke 0 melyet a program kódban így tudunk megadni feltételként `[$? -eq 0]`. Szóval ha a letöltés sikeres volt akkor a *unzip* utasítással kicsomagoljuk

a letöltött fájlt, és ha az is sikeres akkor a sikeresség tényét kiíratjuk a képernyőre ellenkező esetekben *exit 1* et adunk vissza. Itt az egyes jelzi, hogy hiba történt futás közben.

Majd létrehozuk a *lakohelyutvonal.txt* fájlt amit eltárolunk egy változóba rögtön utána, és rászűrünk a megállóral melyre szeretnénk és adjuk is tovább a fájlnkba a szűrés eredményét egy *>>* csővezeték operátorral. Több fájlt is tartalmaz kicsomagolás után a mappa, de a *routes.txt* re volt szükségem, a végén egy egyszerű ellenőrzés, ugyan úgy az előző lépés sikerességét nézve kiírja, hogy sikeres volt a szűrés, vagy sem.

Fejlesztés és tesztelés

Koránt sem tekintemvéglegesnek a programom, azonban a sok teendőm mellett próbáltam annyira egyszerűen megoldani a feladatot amennyire lehet. Ez magában a kódban nyilvánul meg, hiszen egy egyszerű “spagetti” kódot készítettem. Szeretném majd a jövőben átdolgozni, szépen kiszedni függvényekbe a funkciókat, és másfajta megoldásokat is esetleg tanulmányozni, készíteni. A feladat elkészítéséhez találtam egy hasznos weboldalt, amin Alexandra Dobosné munkálya tekinthető meg egy prezentáció formájában. Az alapján szépen sorban oldottam meg a kis részekre felosztott feladatot.

[Számítógépes alapismeretek Shell script programozás - ppt letölteni \(slideplayer.hu\)](https://slideplayer.hu)

A részek amiken egymás után külön dolgoztam megegyeznek az előző részben ahogyan tagoltam a program működését. Nem találtam annyira nagy feladatnak, hogy előre készítsek róla egy tervet, habár érdemes lett volna, hogy azt is gyakoroljam (megint csak az idő gazdálkodásom tudom felhozni mentséggént).

[Maga a tesztelés is elég egyenesen ment. Megírtam egy részt, lefuttattam többször, úgy hogy a feltételek mindkét ága előjőjön és ha nem az történt amit szerettem volna, akkor olvasás elírás keresése míg meg nem oldottam.](#)