**Web programozás 1**

**Beadandó feladat fejlesztői dokumentációja**

**Neumann János Egyetem 2019**

**GAMF Műszaki és Informatikai Kar, Mérnökinformatikus szak**

**Készítője:** Nagy Noel

**Neptun kód:** HEJMA7

**Eredeti weboldal elérése:** *http://mssz.hu*

**Fejlesztett weboldal elérése:** *http://www.nemhivatalosmssz.nhely.hu/*

**GitHub projekt URL címe:** *https://github.com/NagyNoel/Web-programoz-s-beadand-2019*

**Gyakorlatvezető neve:** Medgyes Krisztián *(Csütörtök 1000-1130)*

**Beadás dátuma:** 2019. április 25.

**Fejlesztői dokumentáció**

**Feladat ismertetése**

Keressen egy tetszőleges nonprofit szervezetet (alapítvány, egyesület, stb.), amelynek weboldalt fog készíteni. (Mintha a jelenlegi helyett egy új verziót kellene fejlesztenie). Olyan honlapot keressen, ahol van elegendő szöveges és képi információ a feladat elkészítéséhez.

**Feladat végrehajtása**

Az általam választott weboldal a Magyar Súlyemelő Szövetség hivatalos honlapjára esett, mivel jómagam is versenyszerűen emelek, így az oldal felépítése számomra jól ismert továbbá, sokszor gondolkoztam az oldal fejlesztésén, mivel erre a szövetség nem költ, így az oldal már nem naprakész.

Mikor nekikezdtem a weboldalnak, ami GitHub-on is WebProgBeadando\_NagyNoel\_V1.zip tömörített mappában is látható, elsőként html dokumentumokat hoztam létre, ennek oka, hogy akkor nem jártunk még ott az előadásban, viszont el szerettem volna már kezdeni a feladatot, és látni az oldal felépítését, design-át.

Ekkor 5 darab html file létezett, amik nagy segítségeket tudtak nyújtani a további munkában a weboldal megvalósításában.

Következő feltöltött verzióban WebProgBeadando\_NagyNoel\_V2.zip mappában már mapparendszert is létrehoztam, illetve itt már php fájlokat hoztam létre a html helyett és további 1 css fájlt a stílus elkülönítésére. Viszont az egyedi stíluselemeket a php fájlokba is megjelennek.

A mapparendszer elhelyeztem a „/xampp/htdocs/beadando/” mappába és létrehoztam szintén ide a fő „index.php”-t. Ez a php fájl fogja irányítani az oldalak betöltését a kofig mappában elhelyezett konfigurációs php-kel együtt.

Ezután az előadás diákból, illetve a példa weboldalak segítségével megalkottam az említett „index.php”-t és a konfig mappában a „konfig.php” fájlt. Az „index.php” fájl ekkor már kész volt, azonban a „konfig.php”-t mindig az újonnan létrehozott oldalakkal folyamatosan bővítenem kellett a *„OldalNév' => array('fajl' => 'OldalNév', 'cim' => 'FejlécCím')”* sorral.

Következett az oldal css fájl, a lépcsőzetes stíluslap-dokumentum, amibe a menüben lévő linkek és maga egy táblázat középre igazító stílust adtam meg. Ezt a fájlt a /beadando/stilus/ mappába mentettem és

A létrehozott fájlrendszer tartalmaz ebben az állapotban: „template” mappát a különböző oldalaknak, itt helyezkedik el a második „index.php” amiben a főmenü sávját, a borítóképet és minden nem változó elemet behelyeztem, többi php fájl ebben a mappában az előbb említett „index.php” alatt fog megjelenni.

Létrehoztam a 404-es hibakód számára az oldalt, ami akkor kerül betöltésre, ha az oldalt nem találja a fő „index.php” az adott oldalt. Ez az oldal egy táblázatot tartalmaz, amiben egy rontott súlyemelő gyakorlat látható, a mellette lévő oszlopban pedig kiírja, hogy „Hiba! Az oldal nem található!” Ennek az oldalnak sok hasznát tudtam venni a fejlesztésben.

Ezek után létrehoztam a „versenyek.php” és a „statisztika.php” oldalt. A versenyek oldalt feltöltöttem adatokkal, amit az mssz.hu/versenyek oldalról másoltam át és rendeztem egy táblázatba. A statisztika oldal adatait, viszont még nem töltöttem fel, ezt a bonyolultabb feladatok utánra időzítettem.

Mivel a feladat nem kérte az egész oldal teljes megvalósítását ezért, a további menüpontokat átirányítottam az mssz.hu adott oldalára, ezek a menüpontok a Héraklész, Utánpótlás és a Masters fül.

Ekkor a főoldal még fehér volt, de mivel az első verzióban megterveztem a <form> belépés és regisztráció részét, így azt választottam következő feladatnak. Ezért a beadando/template mappába elkészítettem a „belep.php”, „belepes.php”, „kilep.php” és a „regisztral.php”-t. Továbbá a beadando/logical mappába is „belep.php”, „kilep.php” és a „regisztral.php”-t.

Ezekre azért volt szükség, mivel a template mappában lévő php-k a „sikeres” vagy „nem sikeres” üzenet megjelenítésére szolgál, viszont a tényleges belépés, kilépés és regisztrációt, ami az adatbázisból olvas vagy abba ír, azt a logical mappában lévő php-k fogják végrehajtani.

A „belepes.php”-ba behelyeztem a <form>-ot ezután, teszteltem, hogy hibás adattal is átenged-e illetve, hogy minden adat megadása esetén valósuljon csak az akció.

Ezután következett az adatbázis megtervezése phpmyadmin-ba. Ehhez első sorban a konfig mappában kellett egy adatb.php fájlt létrehozni, ahol definiáltam a phpmyadmin belépési útját a felhasználó nevét és jelszavát.

A böngésző címsorából elérve a localhost/phpmyadmin címen beléptem és létrehoztam a felhasznalok táblát 4 oszloppal „id username password nev”. A regisztrációnál a családnév és utónév külön van megadva, azonban ezt az adatbázisba való beszúrásnál egy stringgé írom össze „nev” oszlopba.

A táblát a grafikus felületen hoztam létre, azonban külön leírtam egy txt fájlba, hogy bármikor egyszerűen létre lehessen hozni, bárhol. Ezen túl, mivel tudtam kell egy üzenetek tábla is a következő feladatokban, így létrehoztam egy mappát is, PhpMyAdmin\_Tables néven, a txt fájlnak, ahova belemásoltam, illetve ahová az „uzenetek” tábla elkészítését is belemásolom.

Mivel tudni kell, hogy be van e lépve a felhasználó vagy sem, mivel a „Belépés” linket át kell írni kilépés linkre, így létrehoztam egy *$\_SESSION['login']* –t, amivel tudom figyelni, hogy a felhasználó be van-e lépve vagy sem.

Belépés esetén ebbe a változóba beleteszem a felhasználónevét, a belépett felhasználónak, kilépés esetén unset-et teszek erre a SESSION-re. A „kilep.php” ezért egyszerűen megvalósítható lett, azonban a belépés sokkal nehezebben kivitelezhető, hiszen ott elösször vizsgálni kell, hogy létezik e felhasználó és jó e jelszava és utána be is kell léptetni.

Erre első sorban, mint ahogy a „regisztral.php” be kell írni az *„include('./konfig/adatb.php');”* sort, majd ezután, megvizsgálni a post metódussal érkezett adatokat, hogy léteznek-e, erre az isset() függvényt használva.

Továbbá létrehozni egy változót az adatbázisnak, ami segítségével, lekérdezéseket tudunk rá alkalmazni. Illetve még egy változót amibe beleteszem az adatbázisra írt lekérdezés és ezt execute() paranccsal végrehajtom.

Ha sikeres a lekérdezés, tehát talált ilyen felhasználójú és jelszójú sort az adatbázisban akkor, beállítja a SESSION[’login’]-t a felhasználó nevére, minden más esetben és hiba esetén is, hibát tesz az $uzenet változóba.

Regisztráció esetén, a lépések nem változnak érdemlegesen. Több adatot kell vizsgálni, hogy létezik-e, illetve a jelszavaknak meg kell egyeznie. Ezt követően szintén lekérdezést hozunk létre, ahol megnézzük létezik-e ilyen nevű felhasználó, ha igen akkor hibaüzenetet írunk a felhasználónak, ha nem beszúrjuk az adatbázisba ezzel pedig létrehoztuk a felhasználót.

Azonban ezek végrehajtásával még a Belépés link nem változik meg. Ekkor kell belenyúlni a „/template/index.php”-ba. Itt megkerestem a Belépés link sorát és beletettem egy *<?php?>* elágazásba, ahol a $\_SESSION[’login’] létezését vizsgálom, hiszen ha létezik be van lépve valaki, ha nem akkor nincs.

Következett a fooldal.php megtervezése. Ide szerettem volna beszúrni a két videót. Az oldal elejére a youtube.com-ról, a videó mellett 3 kép fog megjelenni, ami az oldalon belül egy-egy cikkre fog mutatni és az egyik cikkhez pedig a beépített videót.

Itt jegyezném meg, hogy a videók és cikkek teljesen sajátkészítésűek, ezáltal nem plágium egyik sem, forrásmegjelölést nem igényelnek! Azért döntöttem így, mivel ezek a cikkek számomra készen álltak és könnyebbség, nem leszedni az mssz.hu-ról a cikkeket és hivatkozni rájuk.

Készítettem egy táblázatott, majd annak a második oszlopába ismét létrehoztam egy táblázatott 3 sorral, ahova a képek kerültek. A videót <iframe> blokkal hoztam létre, a képeket <img> -el továbbá ezeket link <a> blokkba is beletettem és oldalon belüli hivatkozássá tettem őket.

Az oldal alján a táblázaton kívül egy <div>-be raktam bele a különböző cikkeket <article> blokkba. Ide illesztettem be a cikkeket, mindegyiket különbözőbe és „id”-vel amire hivatkoztam. Ezáltal a képre rákattintva leugrik az adott cikkhez.

Következhet a galéria. Mindenekelőtt létrehoztam egy $galeria változót a konfig.php fájlban, amibe beletette a galéria elérési útját. Scandir-el végigolvasom a galéria mappát és belerakom egy $files változóba, ezt követően foreach-el végigjárom őket és egyesével megjelenítem őket egy táblázatban, ami 4 oszlopot tartalmaz. Ha egy oszlop megtelik egy sorral lentebb kezdi a következő képet.

Ehhez egy $Oszlop változót hoztam létre, ami minden képre nő eggyel, és ha eléri 4-et, akkor újra nulla lesz, illetve kiír egy sorugrást is a táblázatba.

A kép feltöltéséhez, létrehoztam egy <form>-ot ami ugyanezt a php-t fogja meghívni. Az első egy „file” típusú input elem a másik pedig egy submit gomb.

Mikor rákattintunk a gombra és ismét betölti a lapot, megvizsgálja, létezik-e fájl a feltöltéses input helyen, ha igen megpróbálja feltölteni a képet. Ezt egy elágazásba raktam, így ha sikertelen a feltöltés, hibaüzenetet ír ki a következő sorba.

A képek amiket feltöltöttem, a 2019-es MEFOB képei, azonban nagyon nagy méretük miatt, elsőre nem volt képes felengedni, ezért a xampp -> Apache -> Config -> PHP (php.ini), fájlt megnyitottam kikerestem „upload\_max\_filesize=2M” sort és a kettest tízesre cseréltem: upload\_max\_filesize=10M. Ezután egyből felengedte a nagyméretű képet.

A méretek miatt, az egész Zip nehezen mozgatható volt, így kitöröltem a képeket és csak egyet hagytam benne, ezáltal fel tudtam tölteni GitHub-ra is. Amennyiben találok az oldalra megfelelő méretű és témához illő saját képeket, ezekkel bővítem majd a galériát.

Következett az üzenetes oldal. Elsőként létre kellett hozni az adatbázist, ezt ugyanúgy csináltam, mint a belépésnél, tehát ezt nem részletezném. A tábla legyártásának a parancsa a már említett mappában található.

Az input elemeken keresztül megadjuk a nevünket és email címünket illetve ezek alatt az üzenetet. Végül küldés gombbal bekerül az adatbázisba, ez a regisztrációval azonos formát követ, csak az adatok táblák mezőnevek másak.

Az üzenet elküldése az adatbázisnak a kuldes.php végzi, ugyanez a php fogja az elküldött üzenetet megjeleníteni, ahogy a feladat kérte új oldalon.

A kapcsolatokon belül találunk még egy linket, ami az üzenetek.php ra vezet minket, ahol az eddigi összes üzenetet meg tudjuk tekinteni, névvel, email címmel és üzenettel együtt. Ehhez nem kell belépve lennünk, mivel a feladat erre nem tért ki, azonban ezt a $\_SESSION[’login’]-nal ahogy a belépés kilépés gombnál úgy itt is letilthatjuk, hogy belépés nélkül ne lehessen az üzeneteket megtekinteni.

Az üzenetek megjelenítésére szolgáló uzenetek.php tartalma a belépéshez hasonló módon íródott, azzal a nagy változással, hogy itt a galériához hasonló táblázatos formában jeleníti meg az üzeneteket, 3 oszlopban.

Mivel valószínűleg lesznek hosszabb üzenetek, ezért a táblázat soraiban lévő elemeket felfelé igazítottam, ezáltal olvashatóbbá és egységesebbé vált az üzenetek lap.

Ekkor visszatértem a főoldalra, és a borítóképet linkké tettem, mivel a feladatleírásban volt egy pont, ami kérte, hogy legyen egy link az oldalon, ami az eredeti weboldalra mutat.

Ezt követően létrehoztam egy videok mappát és beletettem egy mp4 fájlt, amit a Felnőtt OB-s cikk aljára beszúrtam, hogy ne csak yoputubról, hanem mappákból előhívott videó is legyen. Beállítottam hozzá pár argumentumot, így folyamatosan ismétli a videót, amit automatikusan indít és kezdeti állapotban némára teszi, hogy mikor odakerül a felhasználó, mert épp olvassa a cikket, ne vágjon bele a gondolataiba a hanggal.

Végül a Google keresést kellett megvalósítani, amihez fel kell tölteni a projektet az internetre. Erre a célra a NetHely oldalát választottam, ahol beregisztráltam, így kaptam 255MB ingyenes tárhelyet.

Ezt követően intéző segítségével FTP protokollon keresztül egyszerűen átmásoltam a fájlokat, miután aktiváltam a nethely tárhelyemet. Innentől elérhetővé vált mindenki számára az oldal, viszont az adatbázisok nem lettek elérhetők, mivel azok a localhoston voltak.

Ezek miatt az adatbázis nevének és információit tartalmazó php-t módosítani kellett és a net tárhelyen létre kellett hozni az adatbázisokat. Ezek után online rendszeren is működött az adatbázis.

**Fejlesztési lehetőségek**

Minden weboldalnak számos fejlesztési lehetősége létezik, folyamatosan fejleszteni bővíteni kell, hogy mindig naprakész adatokkal stílussal várja a felhasználókat.

Az eredeti mssz weboldal egyik fő hibája, ami megnehezíti a felhasználóinak az egyszerűséget az, hogy keresni nehéz az oldalon. Arra gondolok, mikor egy adott emelő adatait, versenyeredményeit szeretnénk megtekinteni. Ezt jelenleg csak úgy tudjuk megtenni, ha keresünk olyan versenyt, amin részt vett az illető és onnan tudunk tovább navigálni a saját oldalára.

Ezen felül lehetséges lenne egy súlyemelő közösséget építeni a weboldal segítségével, ha lenne egy megosztó felület, ahol a versenyzők edzők egymással meg tudják osztani az aktuális történéseket, akár edzéstervet.

Aktualizálni is lehetne a weboldalt, akár aktív youtube beszámolókkal, közleményekkel, amivel segíthetnék az emelőket felkészülni a versenyekre, hogy lássanak új edzésmódszert, új gyakorlatot, amivel fellendíthetnénk a Magyar Súlyemelés világát!