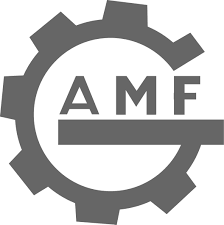
****

**Web-programozás I. beadandó**

**2022.április**

**Géczi Bálint – Neptun azonosító: GDHQXR   
Rab Péter – Neptun azonosító:   
mérnökinformatikus alapszak, levelező tagozat**

**E-mail cím:** [**gbal6565@gmail.com**](mailto:gbal6565@gmail.com)**;**

Tartalomjegyzék

[1 Bevezetés 2](#_Toc101808800)

[2 Fejlesztő dokumentáció 3](#_Toc101808801)

[2.1 A webhely célja 3](#_Toc101808802)

[2.2 Webes felület megvalósításának kezdeti lépései 3](#_Toc101808803)

[2.3 A webes felület megvalósítása Front Controller minta segítségével 3](#_Toc101808804)

[2.4 A felület rövid bemutatása 4](#_Toc101808805)

[2.5 AZ Index.PHP bemutatása és funkciói 5](#_Toc101808806)

[2.6 A képfeltöltő funkció ismertetése 7](#_Toc101808807)

[2.7 Az űrlap bemutatása 8](#_Toc101808808)

[2.8 A felhasználók regisztrációja és az üzenetküldés lehetősége 8](#_Toc101808809)

[3 Felhasználói dokumentáció 8](#_Toc101808810)

[3.1 A futtatáshoz szükséges gépigény 8](#_Toc101808811)

[3.2 A szoftver elindításához szükséges lépések 9](#_Toc101808812)

[3.3 Egyéb információ a szoftverrel és fejlesztővel kapcsolatban 10](#_Toc101808813)

[4 Felhasznált források: 10](#_Toc101808814)

# Bevezetés

Ezen dokumentáció bemutatja részletesen a Web-Programozás I. tárgyhoz kapcsolódó beadandó feladatot és annak fejlesztői leírását, fejlesztési folyamatát. A tervezés során a megkapott feltételek mentén készült a weboldal, s feladatról feladatra kerül a bemutatása az oldal a program kódokon keresztül.

# Fejlesztő dokumentáció

## Webes felület megvalósításának kezdeti lépései

A webhely célja, hogy egy követelményrendszernek megfelelő, alapítvány weboldala készüljön el. A beadandó alapjául a [A szülők háza alapítvány](https://tegyeljot.hu/szulok-haza-program/) oldalt használtuk fel. Az általunk módosított weboldalon az eredeti oldalon látható szövegek és pár kép is megtalálható, s követve a minimalista, de még is megfelelő színvilágú dizájnt.

Egy gyakorlati anyag keretén belül kapott Front Controllert használó minta került felhasználásra. Az eredeti weboldal rengeteg menüponttal rendelkezik, ezt leszűkítettük a beadandó követelményeinek megfelelően. További nagy előnye az imént említett technológiának, hogy külső szemlélő semmit nem lát a könyvtárak és fájlok elhelyezkedéséről így nem, vagy csak nehezen hozzáférhető a weboldalunk tartalma egy esetleges támadás esetén.

## A webes felület megvalósítása Front-Controller minta segítségével

Gyakran előfordul, hogy egy nagyobb méretű weblap fejlesztésénél ugyanazok elemeket kell felhasználni az oldal felületén. Ebben a fejezetben bemutatott tervezési mintát ezért fejlesztették ki, hogy modulárisan felhasználható részek legyenek. Ez azt is jelenti, hogy egy adott elemet (pl.: Footer) ne kelljen minden HTML oldalon külön megvalósítani, újból írni a teljes kódrészt. A beadandó során ezt a lehetőséget kihasználtuk.

Főmappa tartalma: index.php

* images mappa: jpg/png képek
* styles mappa: stilus.css, tablazat.css
* includes mappa: config.inc.php
* js mappa: main.js
* kepek: feltöltött képek helye
* logicals mappa: belep.php, kilepes.php, regisztral.php
* php: kapcsolat.php
* templates mappa: index.tpl.php
* templates/pages mappa: cimlap.tpl.php (címlap oldal), kapcsolat.tpl.php (kapcsolat oldal), tablazat.tpl.php (táblázat oldal), 404.tpl.php (hiba oldal), belepes.tpl.php belep.tpl.php, kilepes.tpl.php regisztral.tpl.php, képgaleria.tpl.php, kepfeltoltes.tpl.php, regisztral.tpl.php, kapcsolatFeldolgoz.tpl.php ,tegyelJotProgram.tpl.php

A forrás könyvtárban található egy inludes nevezetű mappa, abban pedig egy config.inc.php fájl. Ez a fájl tartalmazza a paramétereket és változókat, melyek beállítására különböző tömböket használunk. A 2.2.1-es ábrán látható egy ilyen fejléc nevezetű tömb, ahol a fejlécben szereplő képhez rendelt változónevek láthatók. Itt tároljuk, hogy mi a betöltendő kép neve, mi legyen az alternatív felirat, ha nem jelenne meg, továbbá a fejléc címe is itt változtatható meg.

A képen szöveg látható

Automatikusan generált leírás

*2.2.1) A config fájl egy részlete*

A config fájlban még több ilyen tömb található, mint például a fejléc, lábléc és oldalak nevű tömb.

A styles mappa tartalmazza a formázással kapcsolatos, beállított tulajdonságokat.

A js mappa tartalmazza aJavascript fájlt, ami a kliens oldali ellenőrzésért felel (függvényeket tartalmaz, eseményeket tudunk rendelni, lekezelni a weboldalon általa).

További magyarázatot igénylő mappa a templates mappa, amely a weboldalak „törzsét”/tartalmát foglalja magába. Részletesebb kifejtés ezekről a dokumentáció további részében lesz.

## Az Index.php

Az Index.php valósítja meg az oldal elindulását követően, mely oldalt indítsa el a böngésző. Ha a $GET tömb nem tartalmaz adatot (az adatok hiányában), akkor automatikusan a 404-es hiba „oldal” töltődik be. Alapesetben a címlap.tpl-t töltjük be az oldal „keretébe”, az index.php-ba. Ilyenkor az URL sávban nem jelenik meg fent a betöltött php oldal neve címlap esetén. A menüsávon váltogatva pár megjelennek a betöltött template-k nevei.

## A felület bemutatása



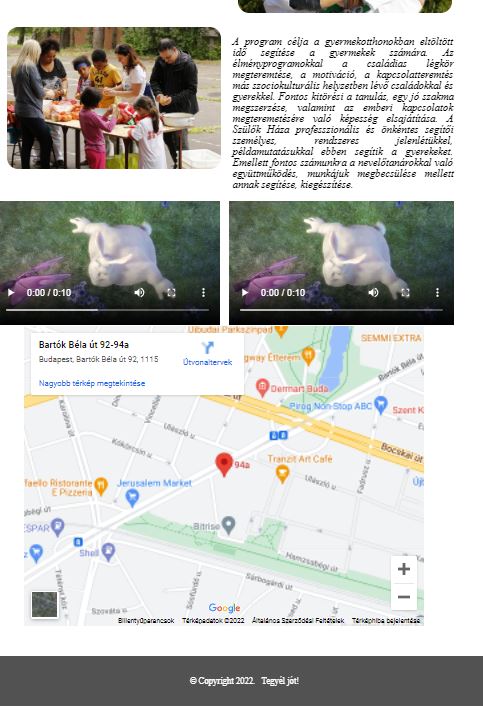
*2.4.1. A betöltődött menüsor látható a képen*

A fent látható fejléc címet és menüsort láthatjuk a az oldal betöltődése után. Ez az az alapértelmezett eset, mikor az index.tpl.php-t tölti be az index.php nevü fájl. A menüsor segítségével tudjuk elérni azt, hogy a az index.php-be -azaz a keretbe- töltődjenek be a külön megírt tartalmi részek (templatek).

A címlapon szerepel egy <h1></h1> sor közé épített címsor. Ezt követően külön bekezdésekre szedve jelennk meg a cikkek, melyek <h2></h2> címsorok között helyezkednek el. A bemutatkozással kapcsolatos oldal nem hordoz magában bővebb információt.

## Tegyél jót program menüpont (template)

Ezen menüpont már több érdekességet rejt magában számunkra (lásd 2.5.1 ábra).



*2.5.1. A Tegyél Jót oldal alsó része*

*része*

Itt láthatunk képekkel teletűzdelt szövegrészt, 2 videóablakot és egy elérhetőséget térképnézettel.

Az oldal szövegét és képeit 4 nagyobb div blokkba készítettük el, melyekben definiáltunk további kisebb div dobozt. Az id=”left” osztályba sorolással ellátott DIV dobozok a bal oldalon található dobozokra vonatkozik a másik id nélkül levő doboz pedig a jobb oldalt.

Tovább görgetve az oldal alja felé láthatjuk a két, témától eltérő , beágyazott DEMO videót. A mozgóképeket ismét egy-egy div blokkba helyeztük el vidLeft és vidRight id-val ellátva. Az alábbi kódsor segítségével jelenik meg a felületünkön a videó:

*<video width="100%" height="auto" controls>*

*<source src="https://www.w3schools.com/html/mov\_bbb.mp4" type="video/mp4">*

*</video>*

Végezetül a térkép elemet helyeztük el az oldalon, melyet iframe forrás deklarációjával oldottunk meg.

## Képgaléria és képfeltöltés

A 3. menüpont alatt a képgaléria és képfeltöltés funkcióját érhetjük el. A megjelenített/feltöltött képek alatt láthatjuk a kép nevét és módosítási dátumát. Kettő darab php fájlra van szükségünk, hogy ezen eredményeket kapjuk az oldalon. A config.inc.php-ban megadjuk a $MAPPA változó segítségével az elérési utat a böngésző számára, ahol a képek vannak. Ezenkívül van 2db tömb definiálva a kiterjesztéseknek és a média típusának. Itt határozzuk még meg a kép maximálisan megjelenített méretét. A képgaléria.tpl részben egy php kódot találunk. A while feltételben megnyitjuk a megadott elérési útvonalat, kiválogatjuk a számunkra megfelelő fájl típusokat s lekérjük a megtalálható fájlok neveit. A fájl neve már megvan a kiíráshoz már csak a filetime() függvénnyel kell lekérnünk a képek módosítás idejét. Amint ezzel elkészültünk egy html-php kódrész segítségével megjelenítjük a képeket. Ennek a kódnak a „magja” egy foreach ciklus, amivel megjelenítjük a képeket külön div dobozba és kiírjuk a hozzá tartozó információkat.

A weboldalunknak van egy képfeltöltés funkcióval rendelkező része. Ezen funkciót a kepfeltoltes.tpl.php fájlban találjuk. A működése az alábbiakban lesz leírva. Először is include paranccsal csatoljuk a config.inc.php-t a kódunkhoz, s létrehozunk egy üzenet nevezetű tömböt. Ezt követően egy IF feltételben lekérdezzük a POST metódussal kapott tömbünket, hogy üres-e (első betöltésnél átugorja a program). Foreach ciklussal végigmegyünk a fájlokon. Következő lépésben IF-ELSEIF elágazás segítségével lekérdezzük a fájl típusát ($fajl[’type’]..) egy másik feltétellel pár sorral lentebb a méretét is lekérdezzük ($fajl[’size’]..). Legvégül még azt kell leellenőriznünk, hogy a kép már felvan-e töltve. Ezt úgy tesszük, hogy a fájlnevet összehasonlítjuk egy file\_exist() függvény segítségével, s ha ez is rendben van, akkor feltölthetjük a képet! Különben hibaüzenet segítéségével tájékoztatjuk a felhasználót a sikertelen műveletről. Legvégezetül a megjelenítésért felelős html résznél elkészítjük az űrlapot a képfeltöltés.tpl.php meghívásával.

# Felhasználói dokumentáció

.

## A futtatáshoz szükséges gépigény

## A szoftver elindításához szükséges lépések

## Egyéb információ a szoftverrel és fejlesztővel kapcsolatban

# Felhasznált források:

* [1] dinamikus struktúra tömb létrehozása: https://slideplayer.hu/slide/2098622/ [ 27. dia ]
* karakterek keresés stringben:

[2] Pánczél István: A C++ programozási nyelv középiskolásoknak – 8 fejezete

[3] <https://www.webotlet.hu/?p=1841><https://gyires.inf.unideb.hu/EFOP344/PythonHTML/strings.html>

[Egy kereses függvény bekezdés]

* [4] Programozás II. gyakorlati anyagok