# *Országos Állatvédőrség Alapítvány*

# *fejlesztői dokumentációja*

***Tárhely URL cím:*** <http://webprog.geekdoktor.hu/>

***GitHub projektjének URL cím:*** <https://github.com/NagypalMarton/Webprog_beadando2020>

# Feladat pontos megfogalmazása

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.

1. Valósítsa meg a weboldalt a Front Controller tervezési minta alkalmazásával. A forráskód a lehető legjobban használja ki a HTML5 lehetőségeit. Legyen a fejlesztett weboldalon egy link a választott szervezet weboldalára, amelyről elég a főoldalt és még egy aloldalt átvenni.
2. Bővítse a választott főoldalt vagy aloldalt két videóval, egyet saját könyvtárából, és egyet szolgáltatótól (pl. Youtube), és a saját könyvtárában lévő video ne legyen több mint 5 mp, a méretkorlát miatt.

A további oldalakat a következő pontok szerint alakítsa ki:

1. Készítsen egy harmadik oldalt képek, képgaléria tárolására. Legyen lehetőség új képek feltöltésére. Képfeltöltést csak bejelentkezett felhasználó tehet meg.
2. A negyedik oldalon legyen egy kapcsolat űrlap, amelynek segítségével e-mailt lehet küldeni az oldal tulajdonosa számára, de az e-mail küldése helyett jelenítse meg az adatokat egy új (ötödik) oldal tartalmaként. Ellenőrizze megfelelően az űrlap helyes kitöltését. Az ellenőrzést végezze el kliens- és szerveroldalon is. A kliens oldali ellenőrzésben használja a HTML5 lehetőségeit és JavaScript-et is.  
   Az elküldött Űrlap adatokat ezen kívül mentse le adatbázisba is.
3. Tegye lehetővé megtekinteni egy hatodik oldalon táblázatban az előző pontban elküldött üzeneteket az adatbázisból fordított időrend szerint (a legfrissebb legyen elől).
4. Regisztráció, Belépés, Kilépés. Bővítsük a honlapot „Belépés” és „Kilépés” menüponttal a következők szerint:

a) A „Belépés” menüpont akkor látható, ha nincs bejelentkezve a felhasználó.

b) A „Kilépés” menüpont akkor látható, ha be van jelentkezve a felhasználó.

c) A „Belépés” menüpontra kattintva feljön egy oldal, ahol lehet bejelentkezni vagy regisztrálni.

d) Regisztrációkor nem léptetjük be automatikusan a felhasználót.

e) A rendszer fejlécen jelenítse meg a bejelentkezett felhasználót, ha be van lépve, a következő formában: Bejelentkezett: Családi\_név Utónév (Login\_név)

1. Alkalmazását töltse fel és valósítsa meg Internetes tárhelyen is. Bármelyik tárhely-szolgáltatót használhatja.
2. Használja a **GitHub** verziókövető rendszert *(*[*https://github.com/*](https://github.com/)*)*. Ne csak a kész alkalmazást töltse fel egy lépésben, hanem a részállapotokat is még legalább 2 lépésben.

# Gépi és nyelvi igényeket

A fejlesztőnek rendelkeznie kell minden modern böngésző legalább 5.0-s verziójával. Ha a forráskódot is módosítani akarja, akkor szerezze az ingyenes *Notepad++* vagy a *Visual Studio Code* vagy a fizetős *PHPstorm* programot, illetve ha futtatni szeretné gépén a PHP-t tartalmazó oldalakat, akkor szerezz be a *XAMPP* nevű ingyenes programot és a fejlesztendő/kipróbálandó weboldalt másolja a program hddocs mappába *(vagy egy ön által létrehozott mappába)* kell bemásolnia. Módosításhoz ismernie kell a következő webes programozási nyelveket: *HTML, CSS, PHP, Javascript*.

Jelenleg még nincs reszponzívvá téve az oldal, így kisebb kijelzőjű eszközökön, mint például mobil telefonokon és tableteken nem jelenik meg megfelelően (esetlegesen nagyobbként jeleníti meg a képeket és a videókat). Laptopokon, asztali gépeken és TV-ken tökéletesen jelenti meg, minden gond nélkül.

# A mappa és fájlszerkezet bemutatását név és funkció szerint

A weblap megnyitásakor a szerver megnyitja az ***index.php*** fájlt, amely meghívja először a *include* mappában lévő ***config.inc.php*** fájlt és a végén a *temp* mappában lévő ***index.tpl.php***-t. A fájl megnyitja az *oldalak* nevű tömböt, amelyben megkeresi a kívánt oldalakat, ha nem találja, akkor 404es hiba kódot ír ki. A ***config.inc.php*** fájl tartalmazza az oldal sablonos tartalmi elemeit, mint például a cím, fejléc információkat, léceket vagy éppen a 404es hibaüzenetet. Az ***index.tpl.php*** építi fel konkrétan a weboldalt, amelyeket minden oldalon meg kell jelenteni és ezzel együtt nyitódik meg a *styles* mappában lévő ***index.css***, amely a stílus beállításokat lehet módosítani. És csak a tartalmi elemek változik, ami az oldalakra való kattintáskor változik meg.

A *temp* mappán belül a *pages* mappában vannak. Innen nyitja meg a *temp* mappában lévő ***index.tpl.php*** fájl az aloldalak tartalmát. A ***blog.tpl.php*** fájl megérkezésekor betölti még pluszban egy oldaldobozt, ahonnan további két aloldalra (***projektjeink.tpl.php***-t és ***kapcsolat.tpl.php***-t fogja megnyitni). A *Kapcsolat* oldalon küldhet be az közvetlenül üzenetet, de jelenleg alatta jeleníti meg a bekért adatokat.

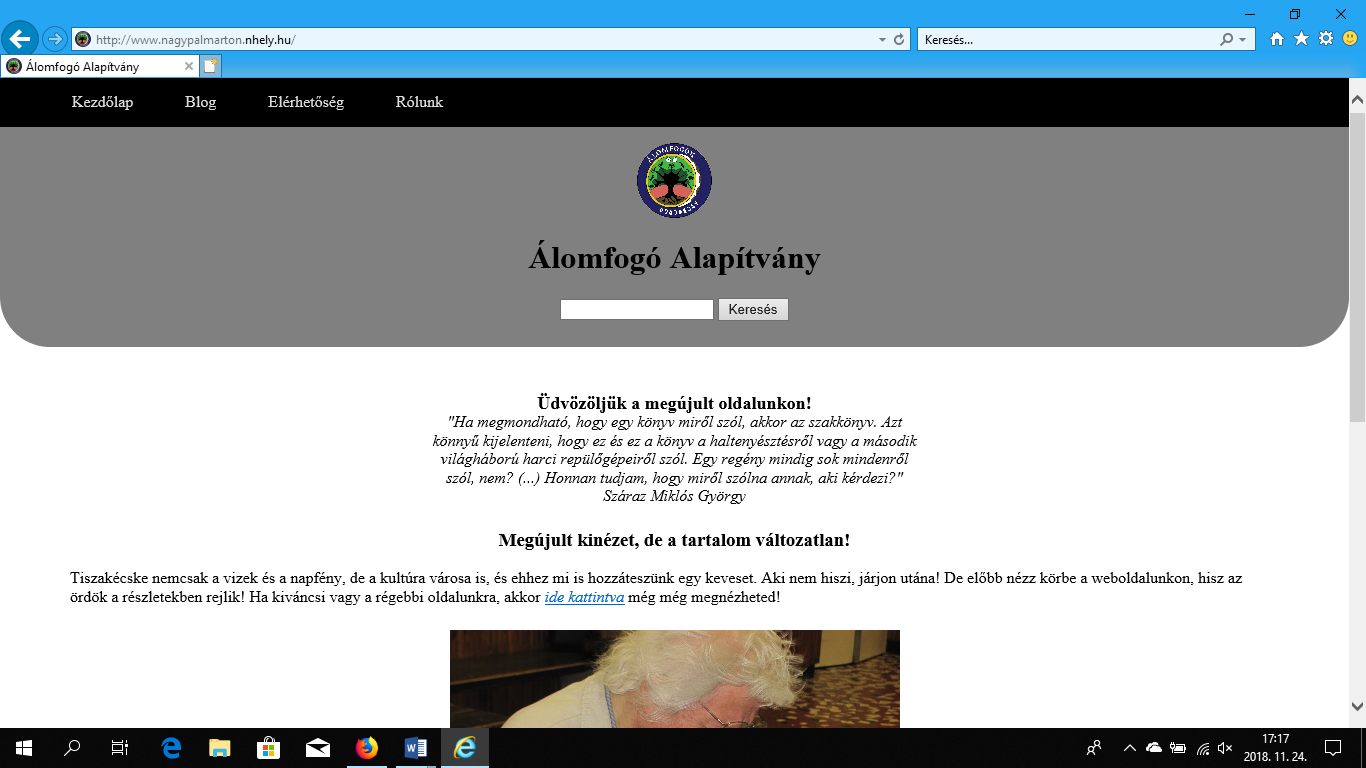
|  |  |
| --- | --- |
| ***Mappanév*** | ***Tartalom*** |
| docs | letölthető dokumentumok  (pl.: Fejlesztői dokumentáció.docx) |
| imagines | a weboldalon megjelenő képek  (pl.: Eszterhazy.jpg) |
| video | saját videók, ahonnan beilleszthetőek saját tárhelyről |

# A tesztelési tervet és a tesztelés eredményeit

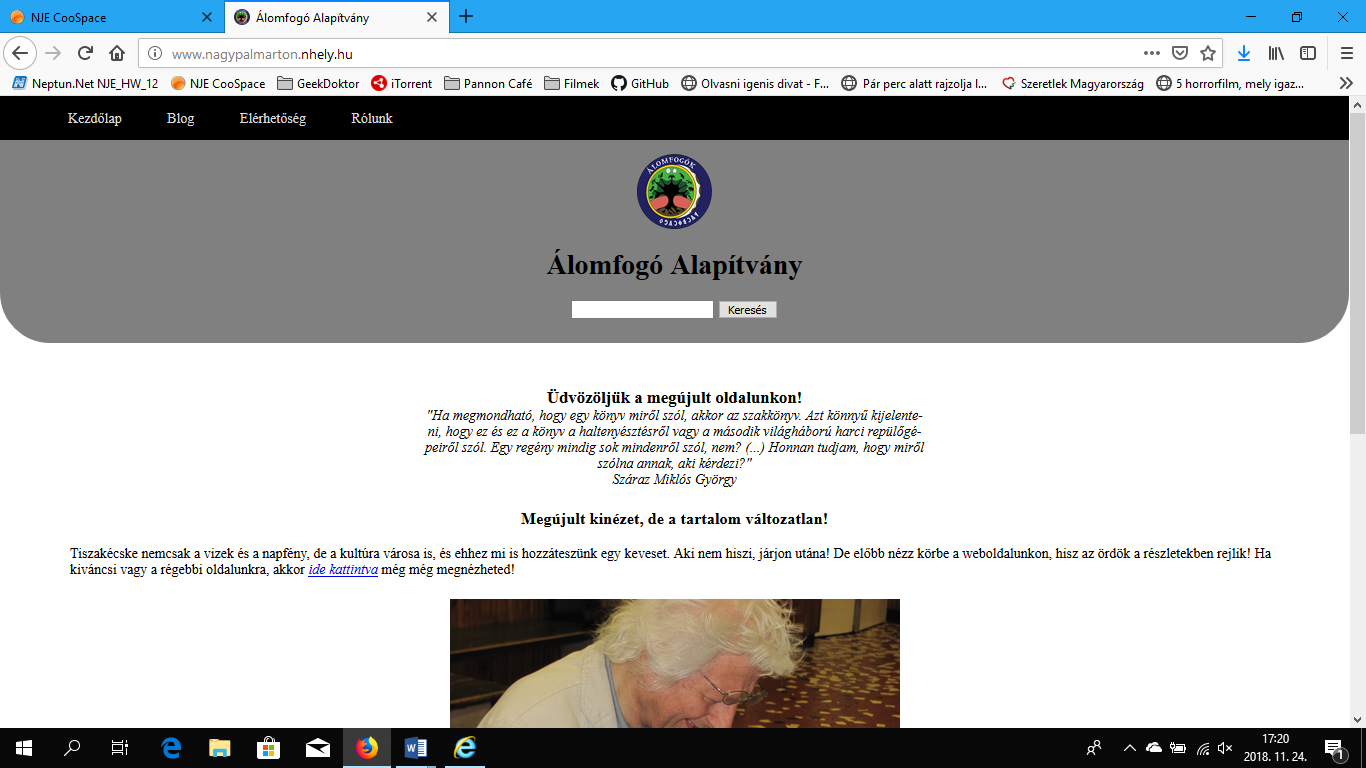
mappa szerkezet

Következő böngészőkre lett kipróbálva: *Google Chrome, Microsoft Edge, Internet Explorer Mozilla Firefox, Opera, Safari*

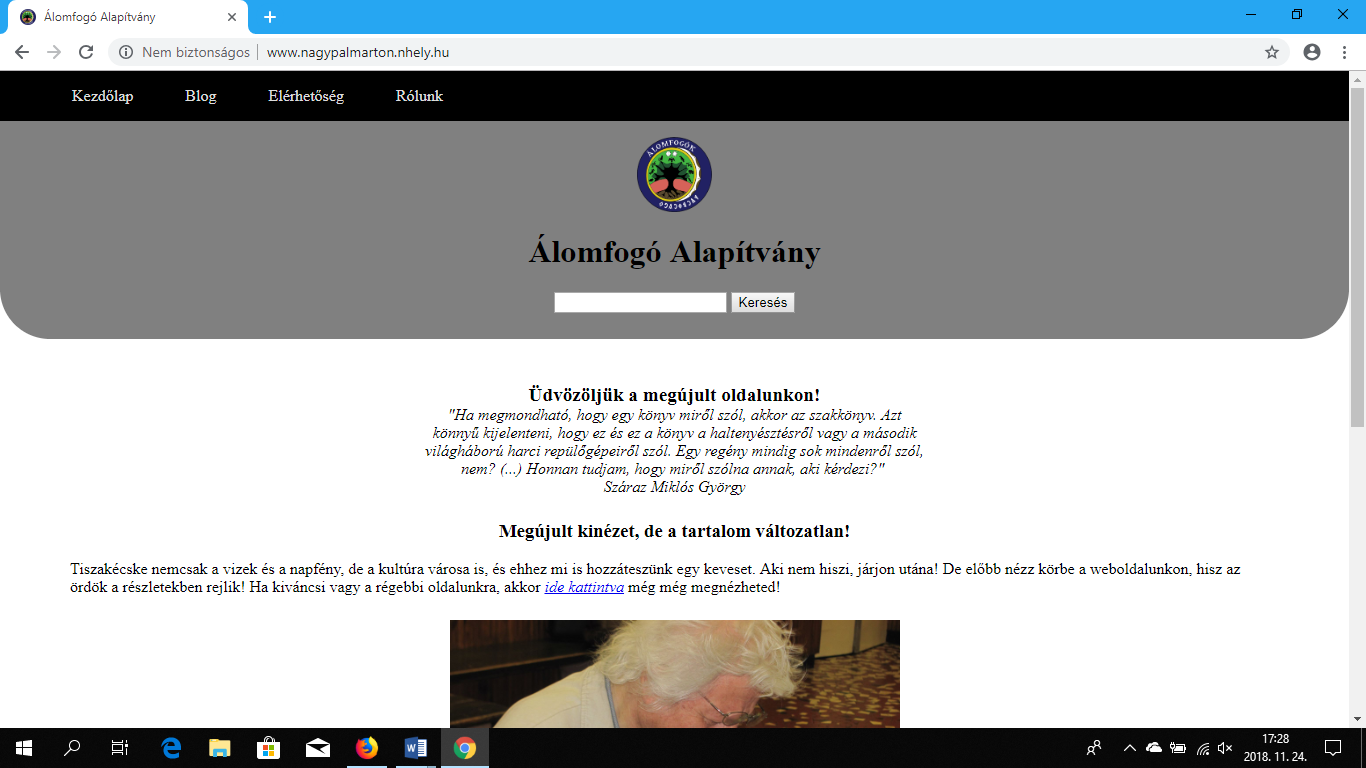
Mind az öt böngészőn hibátlanul lefutott és semmilyen kellemetlen elváltozást nem tapasztaltam. Helyenként esetlegesen lassabban tovább tart betölteni a *Google Maps*-os térképet és fejlécképet. Előzmények törlése nélkül előfordulhat, hogy a kapcsolati lapon hibásnak érzékeli a bevitt adatkat.

**Internet Explorer**

**Mozilla Firefox**



**Google Chrome**



# A potenciális fejlesztési lehetőségeket

1. Optimalizálás kisebb kijelzőkre (reszponzívvá tenni)
2. SEO (-> keresőoptimalizálás)
3. Legördülő, áttetsző menükkel további tartalmakkal bővíteni
4. További oldaldobozokban lehetne widgitekkel*(=minialkalmazásokkal)* bővíteni, ahol további tartalmakat lehet betenni, mint például Archívum, Facebook Oldal doboz.
5. Egyedi üzenet minden nap, mikor egy látogató érkezik a főoldalra.
6. Stílus fejlesztés
7. Reklámoknak való hely, bevétel gyanánt

# Használt külső állományok/keretrendszerek/sablonok bemutatását

A weboldalam a Front Controller tervezési rendszerrel van megoldva. A **front vezérlő** programtervezési minta több minta-katalógusban is szerepel, és a WEB alkalmazások tervezéséhez, felépítéséhez kapcsolódik. A minta „egy központosított belépési pontot biztosít a kérések kezeléséhez.

A front vezérlők gyakran használatosak WEB alkalmazásokban, a működés folyamatainak implementálására. Bár nem feltétlenül szükséges, de sokkal könnyebb kontrollálni a navigációt (az oldalváltások sorát) az egymással összefüggésben lévő oldalak között (például egy online vásárláshoz használt lapok/„lépések“ között) egy front vezérlőből, mint megoldani, hogy az oldalak külön-külön feleljenek a navigációért.

A front vezérlő implementálható egy Java osztályként, vagy egy szkriptben egy szkript nyelven, mint például a *PHP*, *Python* vagy a *Ruby*, amelyben egy webes munkamenet minden egyes kérésénél meghívnak egy szkriptet. Ez a szkript, például egy *index.php*, kezeli majd az alkalmazás vagy a keretrendszer minden általános feladatát, mint például a munkamenet (session) kezelés, a gyorsítótárazás (cache) vagy a bejövő adatok szűrése. A konkrét kérés alapján, a front vezérlő további objektumokat hoz létre, és eljárásokat hív meg, hogy a ténylegesen megkívánt feladat/feladatok el legyenek végezve.

A front vezérlő alternatívája egyedi szkriptek sora lenne, mint a *bejelentkezes.php*, *megrendeles.php* stb., minden egyes kérés típushoz külön-külön. Ezeknek a szkripteknek duplikált kódrészletei vagy objektumai lennének, a minden kérésnél előforduló rész-feladatok elvégzésére. Mindazonáltal az egyes szkriptek nagyobb rugalmassággal rendelkezhetnének az egyes konkrét feladatok elvégzésére.