# *Álomfogó Alapítvány fejlesztői dokumentációja*

***Tárhely URL cím:*** <www.nagypalmarton.nhely.hu/>

***GitHub projektjének URL cím:*** <github.com/NagypalMarton/B3081T.git>

# 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 legalább 5 oldallal a Front Controller minta alkalmazásával (később tárgyaljuk, bővebb leírása a http://web.progtanulo.hu/web-programozas-alapismeretek/3-szerver-oldali-mukodes/36-allomanykezeles/362-egyszeru-front weboldalon).   
   A forráskód a lehető legjobban használja ki a HTML5 lehetőségeit. Az oldalak elsősorban szöveget tartalmazzanak, de illusztrációként 2-3 képet is célszerű elhelyezni oldalanként.

Legyen a fejlesztett weboldalon egy link a választott szervezet weboldalára!

1. Ágyazzon be legalább két videót, legalább egy saját könyvtárából, és legalább egy szolgáltatótól (pl. Youtube, Ustream, Scribd). A két video szerepelhet ugyanazon az oldalon és nem lényeges, hogy milyen hosszúak.
2. Legyen egy keresés űrlap, amely (pl. a Weblaborhoz hasonlóan) a Google keresését kéri a weboldalon belül. Helyezze az űrlapot az oldalak közös részébe (minden oldalon legyen elérhető).
3. 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 6. oldal tartalmaként. Ellenőrizze megfelelően az űrlap helyes kitöltését.
4. Tesztelje a weboldalt a különböző népszerűbb modern böngészőkben.
5. Alkalmazását töltse fel és valósítsa meg Internetes tárhelyen is.
6. Használja a GitHub verziókövető rendszert.

# 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 PHPstorm programot, illetve ha futtatni szeretné gépén a PHP-t tartalmazó oldalakat, akkor szerezz be a XAMPP nevű ingyenes programot és a fejlesztendő weboldalt másolja a program hddocs mappába kell bemásolnia. Módosításhoz ismernie kell a következő webnyelveket: *HTML, CSS, PHP*.

Jelenleg még nincs reszponzívvá téve, így kisebb kijelzőjű eszközökön, mint például mobilokon 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.

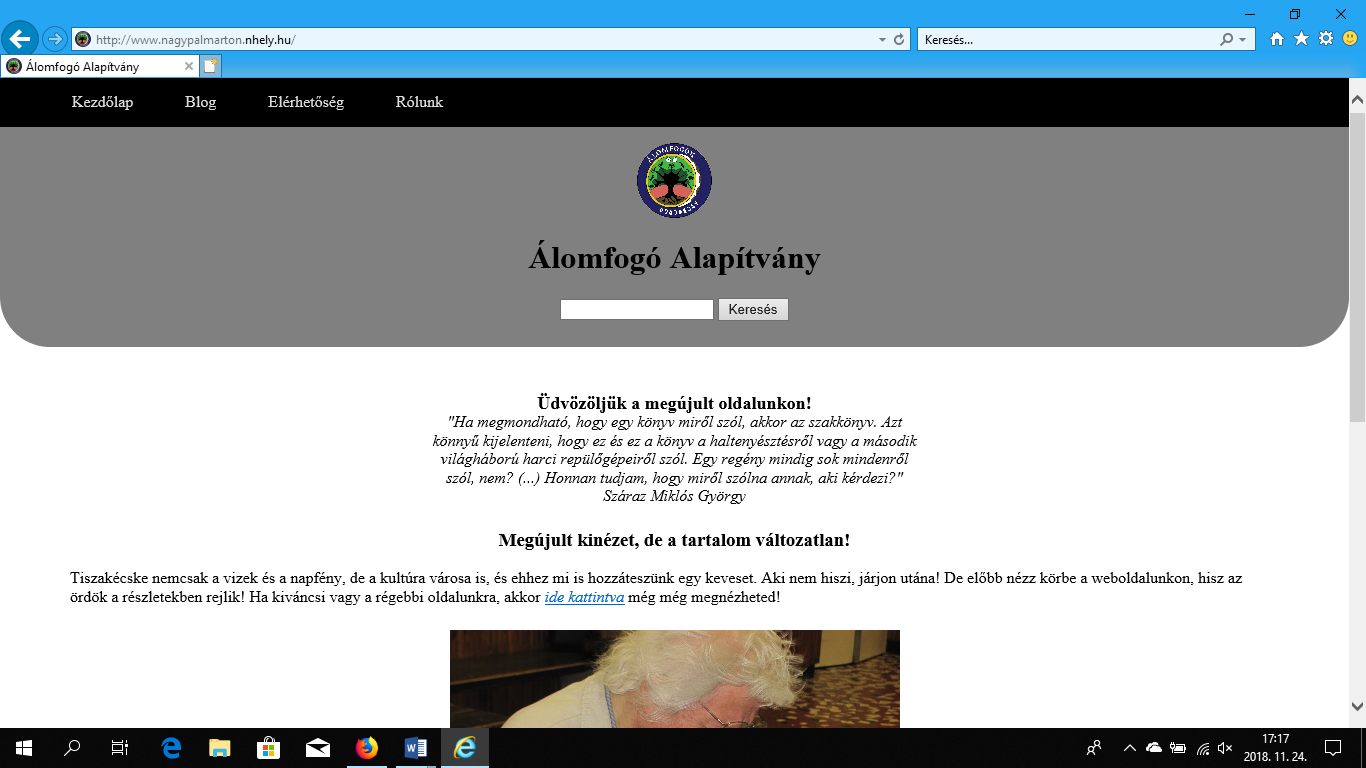
|  |  |
| --- | --- |
| ***Mappanév*** | ***Tartalom*** |
| docs | letölthető dokumentumok  (pl.: Fejlesztői dokumentáció.docx) |
| imagines | a weboldalon megjelenő képek  (pl.: Eszterhazy.jpg) |
| video | saját videók, ahonnan beilleszthetőek saját tárhelyről |

# A tesztelési tervet és a tesztelés eredményeit

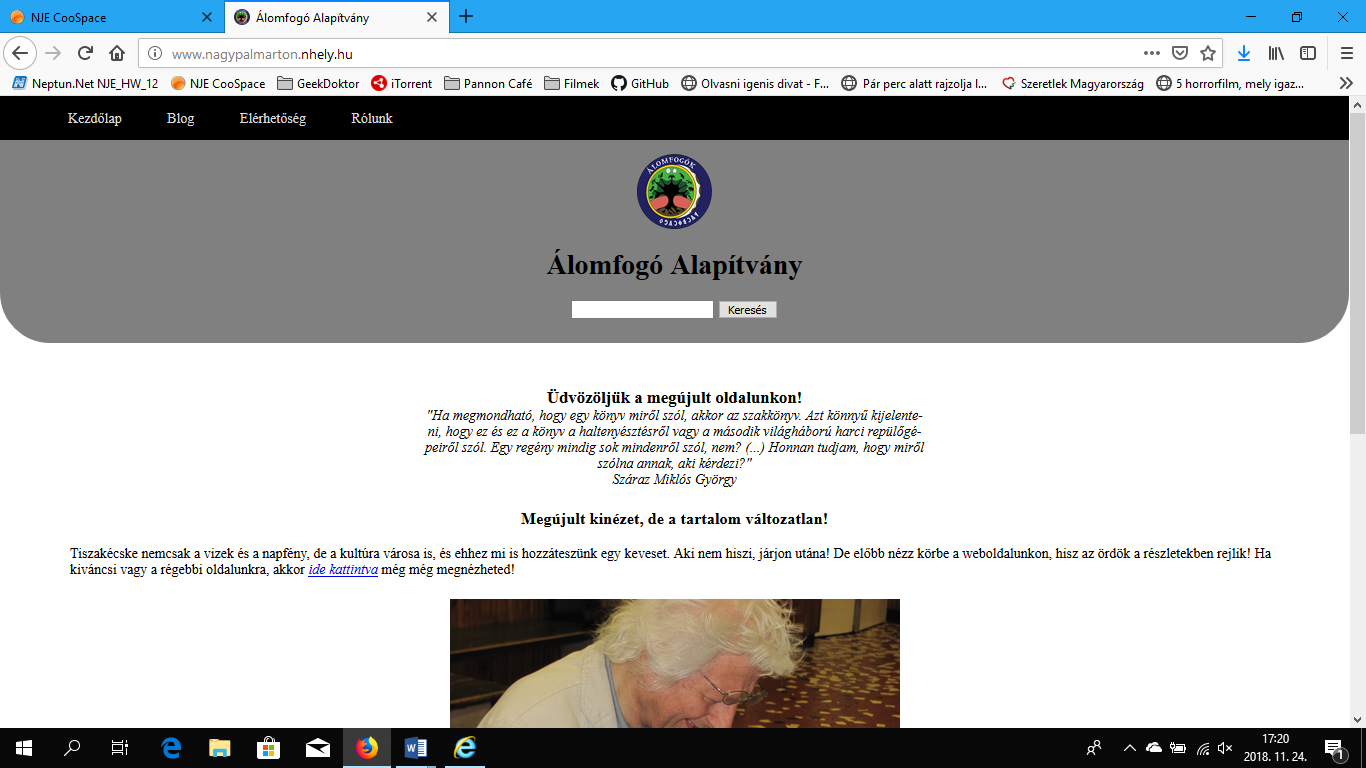
mappa szerkezet

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

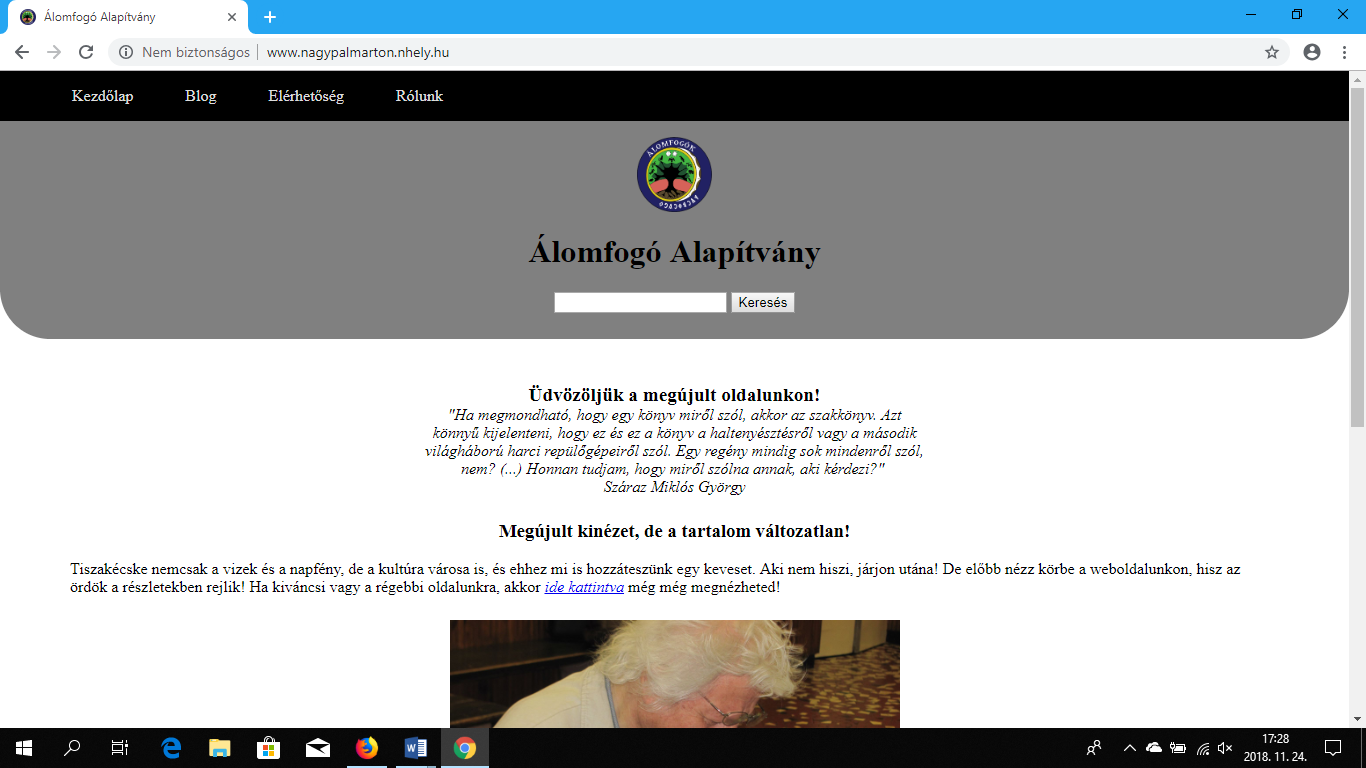
Mind az öt böngészőn hibátlanul lefutott és semmilyen kellemetlen elváltozást nem tapasztaltam. Helyenként esetlegesen lassabban tovább tart betölteni a *Google Maps*-os térképet és fejlécképet. Előzmények törlése nélkül előfordulhat, hogy a kapcsolati lapon hibásnak érzékeli a bevitt adatkat.

**Internet Explorer**

**Mozilla Firefox**



**Google Chrome**



# A potenciális fejlesztési lehetőségeket

1. Optimalizálás kisebb kijelzőkre (reszponzívvá tenni)
2. Galéria létrehozása, ahol vissza lehetne nézni az eddigi rendezvényekre
3. Kommentelési lehetőség a Blog szekcióban
4. SEO (-> keresőoptimalizálás)
5. Legördülő, áttetsző menükkel további tartalmakkal bővíteni
6. További oldaldobozokban lehetne widgitekkel*(=minialkalmazásokkal)* bővíteni, ahol további tartalmakat lehet betenni, mint például Archívum, Facebook Oldal doboz.
7. Egyedi üzenet minden nap, mikor egy látogató érkezik a főoldalra.
8. Stílus fejlesztés
9. 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.