# **A 2021/22-es tanév második félévének Webprogramozás I. tárgyához készített dokumentáció**

A feladat szerint egy nonprofit szervezet, közhasznú alapítvány honlapját kellett feldolgozni, majd a megadott szempontok alapján elkészíteni a saját verziónkat.

Ezt a feladatot két fő készítette:

* Kovács Ádám Lajos
* Nagy Gábor István

Az általunk választott szervezet a Tetovált Állatmentők Állatvédelmi Egyesület honlapja: [www.allatmentok.hu](http://www.allatmentok.hu)

A megoldás során használt szoftverek és technológiák:

* Visual Studio Code 1.66.0
* XAMPP v3.3.0
* Google Chrome 100.0.4896.75
* Git 2.28.0
* Bootstrap 4.3.0
* Sass CSS

A kiírásban a következő szempontoknak kellett megfelni.

* A weboldalt Front Controller dinamikus tervezési mintával kellett megvalósítani.
* A főoldal tartalmát két videóval kellett bővíteni.
* Létre kellett hozni egy galériát, ahová a felhasználó képeket is feltölthet.
* A weblapnak rendelkeznie kell egy „Kapcsolat” oldallal, ahol a felhasználó üzenetet írhat.
* A feladat szerint az előző pontban tárgyaltakat egy új, üzenet oldalon kell táblázatos formában megjeleníteni.
* Regisztráció, Kilépés, Belépés megvalósítása.
* Reszponzív tervezés Bootstrap használatával.
* Az elkészült weblap feltöltése egy webszerverre.
* Az oldal fejlesztésének dokumentálása GitHubon keresztül.

A feladat elkészítése során használt github link: <https://github.com/bigbugsteve/allatmentok>

A feladatban szereplő leírás szerint, feltöltés egy webszerverre, melynek címe: <https://itexpress.hu/allatmentok>

Mielőtt belekezdtünk volna a projekt megvalósításába, közösen felmértük a szervezet eredeti honlapját. Megállapítottuk együtt a formai követelményeket, képernyőképeket készítettünk, ezek segítségével megállapítottuk az oldalon használt pontos színeket.

Megállapodtunk abban, hogy a weboldal elkészítése során csak angol változóneveket használunk, és igyekszünk a lehető legsűrűbben egyeztetni a feladatok állapotáról.

Minden egyes megbeszélés során együtt kitaláltuk a feladatokat, szétosztottuk egymás között, elkezdtük a megvalósítást, ha egyikünk elakadt akkor közösen próbáltuk meg megoldani. Amikor az adott napi feladatokkal végeztünk, akkor megbeszéltük a következő megbeszélés időpontját, illetőleg azt, hogy mik lesznek a következő részfeladatok amiket meg szeretnénk valósítani, így volt időnk felkészülni az adott részből.

A github fiók Gábor fiókja, Ádám ehhez kapott hozzáférést. Több branch is fellelhető benne, egy ADEV és egy GDEV, ezekben dolgoztuk ki a saját részfeladatainkat, majd merge-eltük. Így a fejlesztés folyamatosan egymás mellett haladt, az utolsó simításokat már a DEV branchben végeztük, a végleges pedig a MASTER branch-be került.

Jelen dokumentumban fogjuk tárgyalni, a feladat minden pontját, sorrendben.

Mivel minden feladatot szétosztottunk egymás között (még ha olykor egyet-egyet közösen is oldottunk meg), így a feladatok végén zárójelben feltüntetjük, hogy az adott részért alapvetően ki felelt.

A fejlesztés során egy .env (példa.: .env.example) nevű fájlban tároltuk az adatbázishoz tartozó változókat, amelyet az /environment/envParser.php fájl kezel, így az érzékeny adatok nem kerültek nyilvánosságra.

A bootstrap mellett saját CSS osztályokat is írtunk, ezt igyekeztünk a BEM módszertan szerint elnevezni. A CSS-hez Sass CSS preprocesszort használtunk, ezt egy Live Sass Compiler pluginnal alakítottuk CSS-sé. Kisebb egységekre bontottuk a weboldal részeit, és mindet a main.sccs („/assets/scss/main.scss) fájlban gyűjtöttük. A Live Sass Compiler végig figyelte a változásokat, és minden mentés után át ezt a main.scss fájlt a böngészők által is értelmezhető CSS („/assets/css”) fájllá alakította. Ezt a main.css fájlt használja a weboldal.

Javascript-et („/assets/js/”) használtunk kliens oldalon a mezők validálására. Az itt megírt függvények a mezők kitöltése során minden egyes billentyűleütésre lefutnak, valamint a beviteli mezők úgynevezett. „blur” eseményére. A változóneveket, a logikát, illetve az eseményeket külön fájlban írtuk meg (variables.js, validation.js, events.js), és ezeket a main.js-be importáltuk, a weboldal ezt a fájlt használja.

A dokumentáció további részében a feladatok bemutatását tárgyaljuk.

1. **feladat:** *„Valósítsa meg a weboldalt a Front Controller dinamikus tervezési minta (Kötelező elem!) alkalmazásával.*

*- minden menüpont esetén az index.php fájlt hívja meg*

*- menüneveket és közös adatokat konfigurációs fájlból olvassa be egy TÖMB-ből (array).”*

Ahogy a feladat is kéri, a Front Controller minta szerint mindig egy index.php oldalt hívunk meg, majd ez tölti be saját magába a kívánt aloldalt.

Első lépésként elkészítettük az oldal statikus részeit, mint példáult a footer, navbar, és a header-t.

A menüpontokat egy config.php-ban tároltuk array listként.

*index.php – body része*

1. **feladat:** *„Bővítse a választott főoldalt 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”* - ÁdámA főoldalon megtalálható a kért két videó. A Bootstramp adta lehetőséggel élve két csempébe került a két videó, így ha a webböngésző ablak mérete nem engedi, akkor a videók tudnak egymás alá is kerülni.  
   A Youtube-on van lehetőség a videót megosztani és ezen belül is kiválasztani, hogy egy html iframe beágyazást készítsen nekünk. Ezt a kódrészletet a megfelelő helyre bemásolva, illetőleg a videó lejátszó ablak méretét a kívántra beállítva már működött is.  
   A saját könyvtáras videó lejátszást a html5-nek köszönhetően egy egyszerű video tag-el megvalósítható a forrást és a média típusát megadva.

home.php – A saját és egy beágyazott videó

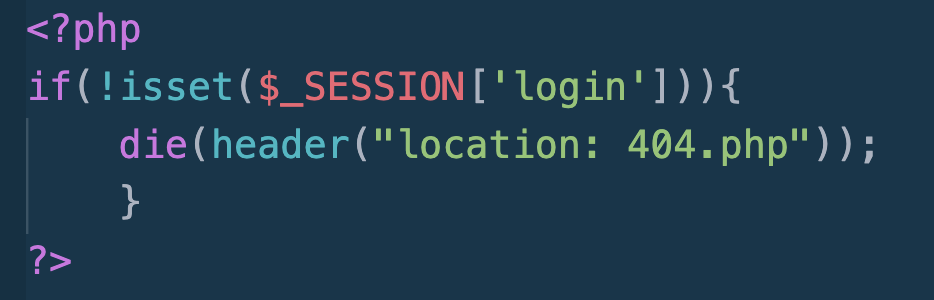
1. **feladat:** *„Készítsen egy harmadik oldalt képek, képgaléria tárolására. Legyen lehetőség új képek feltöltésére. Képfeltöltést csak bejelentkezett felhasználó tehet meg.”* - Ádám

A képgalériát a gallery.php oldalon valósítottuk meg. Ezen az oldalon jelennek meg az assets/images/gallery mappában található képek. Itt vannak alapból a projekthez mellékelt képek, de lehetőség van a későbbiekben tárgyalt mód szerint újakat is hozzáadni. Itt a képek táblázatos módon jelennek meg, mindig a legfrissebb kép van elől, és a „csempékre” a file nevek fel vannak tűntetve. A CSS-nek köszönhetően amennyiben rávisszük az egerünket egy ilyen csempére, bele is nagyítódik a képbe egy kicsit, így adva az oldalnak dinamikát. Amennyiben rá is kattintunk a képre, a böngésző ugyanabban az ablakban / tabon meg fogja nyitni az eredeti képet.

A feltöltés funkciót az upload.php végzi jelen esetben. Ennek elkészítéséhez a 11-PHP – képek, galéria, képfeltöltés-ben található kódot vettem alapul.

Az oldalra egyszerre maximum 3 darab kép (csak jpeg vagy png formátumot fogad el) tölthető fel, de nem kötelező, lehetőség van akár csak egy darabot is feltölteni.

Van fileméret illetve képfelbontás korlát is. Ezek jól látszódnak a mellékelt képeken is.

A feltöltés menüpont annak függvényében jelenik meg a menüsávon, hogy be van e jelentkezve felhasználó. Amennyiben nincs bejelentkezve felhasználó, de mégis valamiért tudja a felhasználó az upload.php elérési útvonalát, nem fog tudni feltölteni, mivel a php-ban először ellenőrzésre kerül, hogy van e bejelentkezett felhasználó.

Amennyiben nincs bejelentkezve, átirányítja a 404.php oldalra.

1. **feladat:** „*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.”*

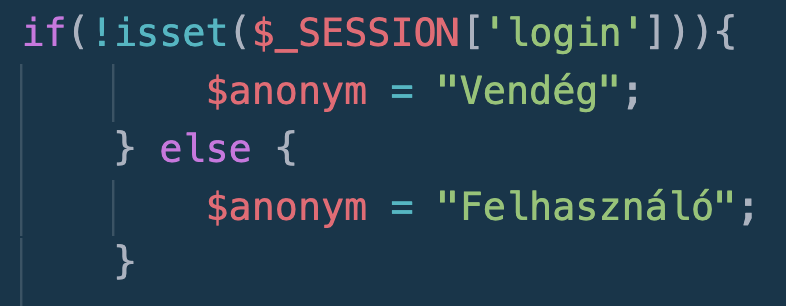
Gábor része

1. **feladat:** *Tegye lehetővé megtekinteni egy hatodik oldalon táblázatban az előző pontban elküldött üzeneteket az adatbázisból fordított időrend szerint (a legfrissebb legyen elől). Írják minden üzenethez a küldés idejét és az üzenetküldő nevét. Ha nem bejelentkezett felhasználó írta, akkor: "Vendég"* - Ádám

Ennél a feladatnál a contact tábláját hívtuk meg az adatbázisnak egy lekérdezéssel:

SELECT csillag FROM contact ORDER BY DESC

Ennek köszönhetően alapból csökkenő időrendi sorrendben jelennek meg a táblázatban az adatok.

A contact táblában szerepel egy oszlop, melynek neve anonym. Ez egy varchar típusú adatot vár, legfeljebb 50 karaktereset. Amennyiben az üzenetet egy bejelentkezett felhasználó írta, akkor „Felhasználó”, ha viszont nem bejelentkezett felhasználó írta, akkor „Vendég” kerül be. Ezt egy HA függvény valósítja meg amely megvizsgálja a Sessiont, hogy login-e. A messageSent.php ezen része:

1. **feladat:** *„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)”*

a-b) A menüpontok array listben vannak tárolva, így minden menüpontnak két értéke van. Ez a két érték határozza meg, hogy egyáltalán látszik e a menüben, látja e a még be nem jelentkezett felhasználó és végül látja e a bejelentkezett felhasználó.

*(1, 1) – mindig látszik*

*(0, 1) – bejelentkezés után látszik*

*(1, 0) – bejelentkezés előtt látszik*

*(0, 0) – soha nem látszik*

c) A „Belépés” menüre kattintva, megváltozik az oldal tartalma, megjelenik a bejelentkezéshez szükséges űrlap, melynek alsó részén megkérdezzük a felhasználót, hogy van e már fiókja. Amennyiben nincs, rá tud kattintani a „Regisztráció” gombra, és az oldal tartalma átalakul a regisztrációhoz.

d) Sikeres regisztráció esetén meghívódik egy regSuccess.php, mely kiírja, hogy sikerült a regisztráció. Sikertelenség esetére nem kellett létrehozni külön oldalt, mert a Form kitöltése nem lehetséges rosszul.

e) Az index.php-ban készült egy HA függvény, mely kiírja a bejelentkezett felhasználó adatait, amennyiben van ilyen.

*csn – családi név*

*un – utónév*

*login – bejelentkezett felhasználó e-mail címe*

1. **feladat:** *„Alkalmazásában használja a Reszponzív tervezést Bootstrap keretrendszerrel.”*

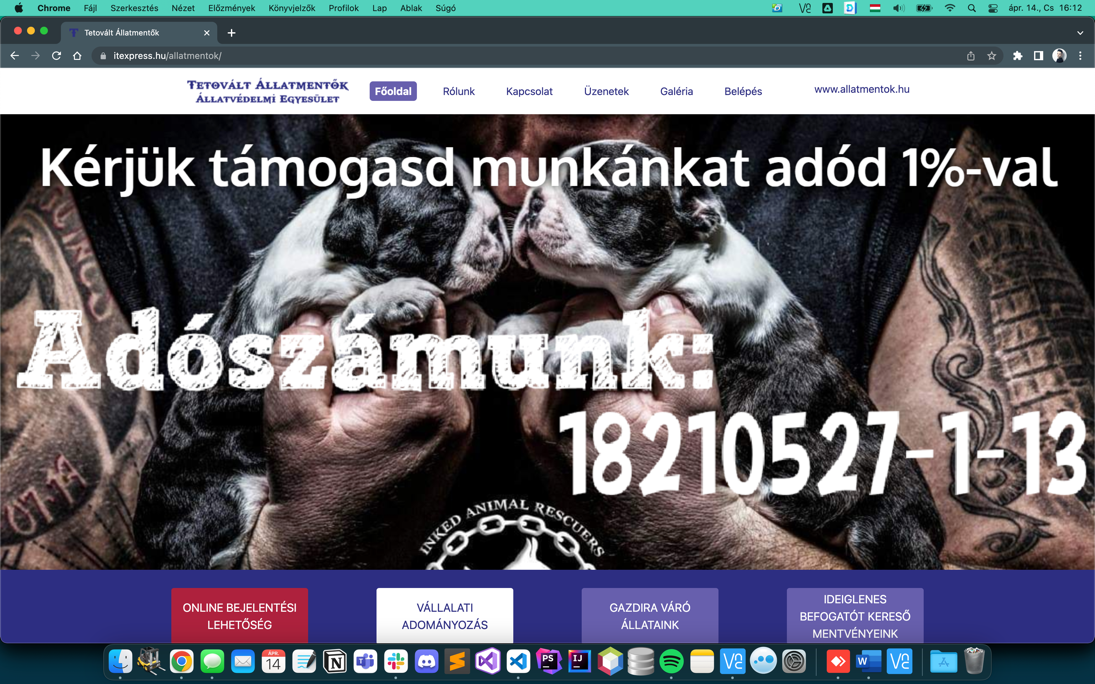
Rögtön a projekt elején beimportálásra került a Bootstramp. Viszonylag sok stílusát használtuk a weboldal elkészítése során, leginkább az üzenetküldő, a belépés / regisztráció menüben, természetesen kiegészítve a saját stíluselemeinkkel.

A Bootstramp a gyökérben található ./index.php head részébe lett beillesztve, így minden template-re érvényesül.

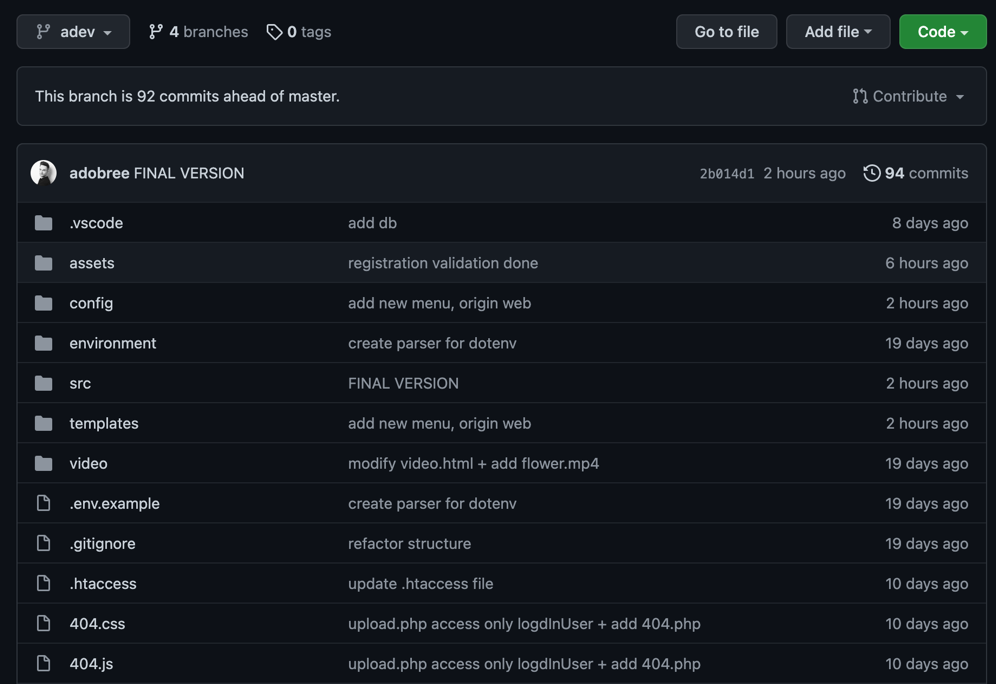
1. **feladat:** *„Az alkalmazását töltse fel és valósítsa meg Internetes tárhelyen is.”*

Mivel Ádám rendelkezett saját weboldallal melyhez tartozott adatbázis is, így célszerű volt ezt a tárhelyet használni. Létrehoztunk egy új adatbázist, azon belül is két táblát, a contact és a user táblát. Ha és amennyiben lokálisan szeretné használni, illetőleg tesztelni a weboldalt, akkor a projekthez mellékelt adatbázis beimportálása után és egy új mestervi\_allatmentok felhasználó (jelszo: Allatt2000) felvétele után helyileg is használható.

Az adatbázis alapértelmezetten localhost-ra néz, mi Xampp-ot használtunk, azzal műkött.

A weboldal elérhető: [www.itexpress.hu/allatmentok/](http://www.itexpress.hu/allatmentok/)

1. **feladat:** *„Használja a GitHub verziókövető rendszert…”*

Ez a része a feladatnak nem jelentett különösebb kihívást, Gábor munkája során napiszinten használja a github-ot, mint verziókezelőt. Ádám egy YouTube-on is ingyenesen fellelhető videós tanfolyam után, és Gábor segítségével könnyen elsajátította a github alapjait, illetőleg magát a git-et. A github link: <https://github.com/bigbugsteve/allatmentok>