[Bevezetés 1](#_Toc100424390)

[Fejlesztői Dokumentáció 2](#_Toc100424391)

[**Alkalmazott fejlesztői környezetek és technológiák** 2](#_Toc100424392)

[Fejlesztői környezet 2](#_Toc100424393)

[Programozási és egyéb nyelvek 2](#_Toc100424394)

[Ütemterv 3](#_Toc100424395)

[Adatbázis tervezés. 4](#_Toc100424396)

[Megvalósítás 5](#_Toc100424397)

[index.php 5](#_Toc100424398)

[header.php 5](#_Toc100424399)

[menu.php 7](#_Toc100424400)

[footer.php 7](#_Toc100424401)

[database.php 8](#_Toc100424402)

[db\_function.php 8](#_Toc100424403)

[function.php 9](#_Toc100424404)

[language.php 11](#_Toc100424405)

[index.js 12](#_Toc100424406)

# **Bevezetés**

A jövőmet az informatika világában képzeltem el ezért elengedhetetlennek érzem, hogy tudásomat a gyakorlatban is képes legyek prezentálni. A témakiválasztás idején fontos szempont volt a számomra, hogy az elkészített weboldal felhasználható legyen az álláskeresés esetén. Ezek feltételek mentén haladva és e probléma orvoslására készítettem egy Portfólió weboldalt.

A weboldal célja, hogy álláskeresés esetén egy általam készített az önéletrajz mellé egy portfoliót is feltudjak mutatni. A portfolió egy esetlegesen alapított magánvállalkozás esetén is alkalmazható, hogy a jövőbeli ügyfeleimet tájékoztathassam a személyemről, szakmai tudásomról és az aktuálisan elkészített munkásságaimról.

Megoldást nyújt abban, hogy ne keljen több felületet kezelnem és fenntartanom egyszerre, hanem összegyűjthettem egy oldalra és ott kezelhetek minden szükséges információt és repository-t.

A weboldal magyarul és angolul is elérhető, hogy az itthoni és a külföldi piacon is versenyképes maradjon. Több nyelv beiktatását nem éreztem szükségesnek, mivel az IT szektor mindig is angol nyelv központú volt.

Fontos szempont volt a részemről, hogy az oldalt teljesen mértékben én készítettem, kódoltam és nem egy előre elkészített sablonnal dolgoztam. Amely teljes kontrolt enged az oldalam felett továbbá, nem függ más cégektől, sem a publikálás, sem a testre szabhatóság tekintetében.

# **Fejlesztői Dokumentáció**

## **Alkalmazott fejlesztői környezetek és technológiák**

### Fejlesztői környezet

**Visual Studio Code:** Ezen felületen készítettem el a weboldal kódolásának a túlnyomó részét.

**Notepad ++:** Kisebb és gyors korrigálások esetén, továbbá előző munkáim kódjainak vissza tekintéséhez alkalmaztam.

**XAMMP:** E szoftver gondoskodik a php számára szükséges szerver környezetéről. Továbbá biztosítja az oldalamhoz szükséges adatbázist.

### Programozási és egyéb nyelvek

**HTML:** A HTML(Hypertext Markup Language) nyújtja a weboldal alapját. E jelölőnyelv segítségével állítottam elő az oldal szerkezetét és felépítését.

**CSS:** A CSS(Cascading Style Sheets) az oldal stílus elemeinek és oldal megjelenésének beállítási szempontjából elengedhetetlen elem. Továbbá ez biztosítja az weboldalon található háttérképeket megjelenítését és minden design elemet.

**PHP:** A PHP(Hypertext Preprocessor) scripting nyelv segítségével rengeteg funkció és dinamikus elem megjelenítés fűződik. Ezen nyelv segítségével rengeteg programozási eszközzel dolgozhattam, amely számtalan kulcsfontosságú funkcióért felel.

**JavaScript:** A JavaScript felel az weboldalamon található gombok és egyéb kliens oldali funkciók a felhasználó számára zökkenő mentes működéséről.

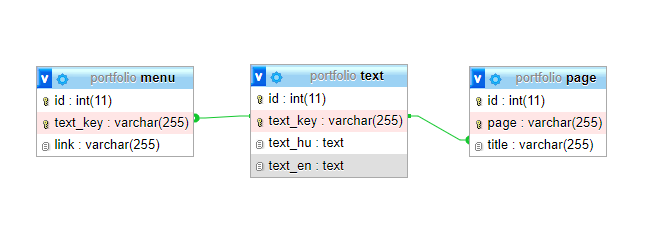
**MySQL:** A MySQL adatbázis kezelési környezetett alkalmaztam a munkáim folyamán. Itt kerültek tárolásra például a fordítási kulcsok, és a szöveg több nyelven.

## Ütemterv

|  |  |
| --- | --- |
| 2021.10 | A weboldallal kapcsolatos első gondolatok és tervek elkészítése. |
| 2021.11 | Végleges tervek és azok prezentálása. |
| 2022.02.15-17 | Az adatbázis terv elkészítése. |
| 2022.02.25 | A weboldal alap mappái, fájlszerkezetének létrehozása, kiépítése. |
| 2022.02.27 | A weboldal alapkódjainak lefektetése. |
| 2022.03.09 | A főoldal, navigációs sáv első változatának elkészítése. A stílus alapjainak felállítása.  Egy olyan gomb elkészítése, amely az oldal tetejére görget. |
| 2022.03.10 | Az adatbázis első verziójának elkészítése és össze kötése a weboldallal. A menu kiírása az adatbázisból. A menu(menu.php) rendszer elkészítése. |
| 2022.03.16 | A menürendszer tovább fejlesztése. A főoldal javítása, bővítése. |
| 2022.03.18-21 | A főoldal stílusának finomítása, fejlesztése. |
| 2022.03.22 | A főoldal háttérbeállítása és egyéb funkciók implementálása. |
| 2022.03.23 | Az oldalra elhelyezésre került még egy gomb, mely a tartalomhoz történő görgetést szolgálja. |
| 2022.03.24-26 | Cookie beépítése. Nyelv választasi opció kiépítése. Görgetések letiltása mobil nézetből, ha a navigációs menü megnyílik. |
| 2022.03.28 | Az adatbázis végleges verziójának elkészítése, majd összekötése a weboldallal. Minden dinamikus szöveg kiírása az adatbázisból. |
| 2022.03.29-31 | Hibajavítások és a mellék oldalak fejlesztése. |

## Adatbázis tervezés.

Adatbázisom arra épül, hogy a weboldalon lévő szövegeket, menüpontokat és az adott oldal title-jét dinamikusan az adatbázisból jelenítse meg. Erre azért van szükség, hogy a felhasználó által beállított nyelven jelenítse meg az oldalt.



**menu** tábla felel a navigációs sávon található menüpontok betöltéséért.

* id: Egy elsődleges kulcs, amely az egyedi azonosításért felel.
* text\_key: Idegen kulcs, amely text tábla azonos nevű oszlopára hivatkozik. Emberi olvasásra alkalmas. Meghatározza a menüpontok szövegét.
* link: Az aktív állapot ellenőrzését szolgája, továbbá a navigálásért is felel.

**text** tábla felel az oldal szövegeinek megjelenítéséről. Itt tárolódik minden az oldalon megjelenítésre kerülő szöveg.

* id: Egy elsődleges kulcs, amely az egyedi azonosításért felel.
* text\_key: Fordítási kulcs, amely a fejlesztési munkálatokat gyorsítja fel. Emberi olvasásra alkalmas.
* text\_hu: A fordítási kulcshoz tartozó magyar szöveg.
* text\_en: A fordítási kulcshoz tartozó angol szöveg.

**page** tábla tárolja az adott menüponthoz a title-t.

* id: Egy elsődleges kulcs, amely az egyedi azonosításért felel.
* page: Az aktív állapot ellenőrzését szolgája.
* title: Idegen kulcs, amely text tábla azonos nevű oszlopára hivatkozik. Emberi olvasásra alkalmas. Meghatározza a menüpontok szövegét.

## Megvalósítás

### index.php

Az index.php a legfőbb fájl az egész oldalon. Feladata, hogy minden kulcsfontoságú php fájlt meghívjon.

Itt történik a db\_function.php, function.php, header.php, footer.php fáljok, továbbá az oldal kulcsainak lefordításáért felelős függvények meghívása.

### header.php

Ezt a fájt úgy készítettem el, hogy minden oldalon fix legyen. Itt helyezkedik el a <head> beállításai és importálások, mint a Style.css, vagy a betűtípus. Ebben a fájlban található meg a navigációs sáv kódja is.

A benne található rövid php kódban kerül meghívásra a menu() függvény melyet a $menu változó fog megkapni. Meghívásra kerül a language.php és a menu.php. Továbbá megvizsgálja az oldal, hogy milyen nyelvre van beállítja és az a $htmllang megkapja értéken.

A <head>-ben megadom az az oldal kódolását, ami UTF-8. Meghívásra került a style.css és emellett a betűtípus és annak változati, és egy icon csomag. A title egy függvényben kapja meg az értéket attól függően, hogy mely oldalon tartózkodik a felhasználó.

Egy menuflex nevű flexkonténerbe elhelyezésre került a logo, a menü navigációs sáv és egy icon, amely mobil nézetben jelenik meg.

A menü sávok megjelenítése egy php kóddal lett megoldva, amely megvizsgálja, hogy a felhasználó melyik oldat tartózkodik és azt a menü pontot teszi current class-be, amely kiemelésre kerül.

Alatta a menü navigációs sáv kódja található. Célja, hogy egy foreach -el kiírva legyen minden menü pont. Ezáltal bármikor bővíthető vagy eltávolítható egy-egy menüpont. Olyan nyelven kerülnek kiírásra a menüpontok, amely beállításra került.

### menu.php

A menu.php az oldalak közti átjárást biztosítja.

Alapértelmezetten a főoldalra vezet az oldal, ha az url-ben egy nem létező vagy helytelen értéket kapna. Ha mégis teljesül a megfelelő értkékadás akkor egy belső if -ben megvizsgálásra kerül, hogy az érték megfelelő formátumú, ha feltétel igaz akkor átadja a $menuitem-nek. Tovább adja majd $page változónak végül egy meghívási útvonalként beépül egy stringbe. A string a index.php fájlban kerül felhasználásra.

Ezt követően a weboldal title-je kerül terítékre. Az éppen aktuális oldalhoz tartozó fordítókulcsot ki kéri az adatbázisból, majd lefordíttatja egy wraptrans konstruktorral.

### footer.php

A header-höz hasonlóan minden oldalon fix elem.

Ide került elhelyezésre a gomb, amely megjelenik akkor, ha felhasználó legörget az oldalon és eltűnik, ha a teteére görget a felhasználó. Ezt a funkciót JavaScriptben oldottam meg.

Kiírásra kerül a nevem és a dátum, emellett itt kerül meghívásra az index.js JavaScript fájl.

Továbbá itt helyezkednek el a záró tagek is.

### database.php

Itt történik az adatbázishoz történő csatlakozás.

Ez a rövid függvény megkapja az adatbázis elérhetőségének útvonalát és a felcsatlakozáshoz szükséges adatokat, mint a user, password.

### db\_function.php

Ebben az fájlban az adatbázisnak szánt utasítások találhatóak.

function menu() lekérdezésnek az a célja, hogy lekérdezésre kerüljön a navigációs sávhoz szükséges fordítási kulcsok az adatbázisból.

function page($page) esetén hasonlóan a menu function-hoz lekérdezésre kerül a fordítási kulcs az adatbázishoz. Ám itt a title-hez kerül lekérdezésre.

Meghívásra kerül a $lang változó, amely tartalmazza oldal aktuális nyelvi beállításait. A $qmarks változó a dinamikus elemek lekéréséhez kérdő jeleké alakítja az aktuális tömb értékeit. A lekérdezés az aktuális nyelvi beállítások alapján kérdezi le a szöveget az adatbázisból.

### function.php

A function, a nevéhez hűen function-öket tartalmaz. Ezek a konstruktorok az weboldalon belül számtalan felhasználásra/meghívásra kerülnek. Viszont az itt található konstruktorok mindegyike össze dolgozik egy fő célért, ami nem más, mint az adatbázisból való kulcsokat meghívja, lefordítsa és kiírassa az aktuális nyelvi beállításokhoz igazodva.

A $regx egy olyan változó, mely egy regex szöveg ellenőrző stringet kap értéknek. Gyakran kerül felhasználásra.

A function wraptrans($m) az adatbázisból kapott fordítókulcsot string # hastag-el egészíti ki, hogy amikor a weboldal kódja áttekintésre került egy másik function által, akkor regex segítségével kiszűrhetőek lesznek a kód rengeteg közül.

A function extractkeys($m) konstruktor fogja megkapni a weboldal teljes kódját, amit a $m változó tartalmazz magában. Azok az elemek a kódban mely 3-3 # közé került a regex segítségével kiválogatásra kerülnek és megszabadítja őket a #-től, majd egy tömbbe helyezi őket.

A function maptrans($pagetrans) a megkapott tömb értékeihez rekordokat társít. A kulcsukhoz hozzárendelésre kerülnek a hozzá tartozó szövegek az adatbázisból.

A function finaltrans($m, $n) legvégső lépéskent megvizsgálja az oldal kódját, majd a match-elt értékeket átalakítja az adott nyelven beállított rekordban levő értékkel, ha nem match-el akkor vissza adja a $m-t(azaz az oldal eredi kódját).

### language.php

A language.php feladata, a nyelvi beállítások elvégzése.

Ha az oldal url-ben az aktuális nyelvi beállítások megjelennek akkor, az egy cookie-ba kerül eltarolásra, mely a későbbiekben felhasználásra kerül.

A nyelvi beállítások különböző értékekből kerülhetnek beállításra.

* Első soron az oldal url-ből kerül kiolvasásra.
* Másodsoron a cookie-ból. A cookie eltárolja a felhasználó beállításait, amelyet elvégzett az oldalon. Ez azért hasznos mivel, ha a felhasználó elhagyja az oldalt a későbbi visszatérése esetén megmaradnak a beállítások.
* Harmadsoron a default érték kerül érvényesítésre. Ha a felhasználó frissen belép az oldalra, akkor az url nem fog tartalmazni értéket. Ha cookie, még nem készült vagy az törölve lett akkor onnan sem lehet adatott felvenni tehát az érték null lesz.

$temp\_lang fontossági sorendben megkapja annak a beállításának az értékét, amely a legrelevánsabb.

Egy switch segítségével elbírálásra kerül, hogy a $lang változó mely beállítási értéket kapja meg, melyet a weboldal más területeken felhasznosít majd. Ha az érték vagy "hu" vagy "en" akkor értelemszerűen, azt a stringet kapja meg. Viszont, ha az érték null vagy az előbb felsoroltak közül egy feltétel sem teljesül, akkor a default érték fog megvalósulni, ami a "hu".

### index.js

Az index.js ben rejlenek a JavaScriptes kódok, melyek a kliens oldali funkciókért felelnek. Feladatuk, hogy a weboldal futása közben oldal újra töltések szükségességén kívül is működtessen funkciókat.

class ScrollToTop

A class a footerben lévő gomb elrejtéséről és megjelenítéséről gondoskodik.

A constructor() felel a footerben elhejeszkedő gomb meghívásáról.

Megvizsgálásra kerül, hogy a oldal top értéke nagyobb-e mint 20. Ha ez a feltétel igaz akkor a Page bool változó true értéket kap. A változott továbbadásra kerül Pagebool2 függvénynek.

A bool érték megvizsgálásra kerül. Ha az érték true akkor a gomb style-ja block lesz, azaz megjelenik. Ha az érték false akkor a gomb style-ja none lesz, tehát nem jelenik meg.

Ha a gomb lenyomásra kerül, akkor az oldal felgördül a top 0-ik értékre. Ezt egy smooth behavior-el van kivitelezve, amely segítségével nem felugrik az oldal tettéére, hanem ízlésesen felcsúszik.   
  
A window.addEventListener folyamatosan felügyeli, hogy történt-e sor az oldalon görgetés és folyamatosan lefuttatja a checkPagePosition() függvényt.

class ScrollBlock

A weboldal készítése közben felmerültek problémák. Mobil nézetből az oldal máshogy jeleníti meg a menü navigációs sávot. Gomb lenyomásra a teljes képernyőre rányílik az menü sáv. A probléma azaz, hogy ilyenkor ki lehetett görgetni a menü sávból. Ez a class ezt a hibát hivatott kijavítani úgy, hogy teljesen le tiltja a görgetés lehetőséget, ha a menü sáv megnyílik.

disable() tartalmazza az utasítás, mely le tiltja a görgetést az oldalon.

enable() Az egyel feletti utasítással ellentétesen engedélyezi a görgetés az oldalon.

A konstruktorban kerül meghívása #check id-val rendelkező gombót és egy check nevű bool változónak átadásra kerül. Ha lenyomásra kerül akkor egy if elágazással ellenőrzése kerül és a kapott értéktől függően a görgetés lehetősége ki- vagy bekapcsolásra kerül.

class ScrollDown

A főoldalon elhelyezésre került egy gomb, amely célja, hogy az oldalt az tartalomhoz görgesse. Ehhez a kód, úgy lett megírva, hogy a flexkonténerhez görgessen. Azaz, ha az oldal később változtatásokon vagy bővítésen esik át akkor ne keljen a koordinátákat módosítani, hanem minden esetben a tartalomhoz görgessen.

A konstruktorban kerül meghívásra a gomb és a flexkonténer meghívása későbbi felhasználásra.

buttonPush() függvényben maga az utasítás található. Ha megkapja a parancsot a lefutásra, akkor a flexkonténerhez görgeti az oldalt, a kiszámolt értékek alapján. A számított adatot az oldal aktuális pozíciójából és a konténer helyzetéből szerzi összeadással.