

Veszprémi Szakképzési Centrum Ipari Technikum

Szakképzés neve: Szoftverfejlesztő

OKJ száma: 54 213 05

SZAKDOLGOZAT

NavIPARI

Dudás Erika
témavezető

Mátics Dániel
13.C



Veszprém, 2022

Tartalomjegyzék

Felhasználói dokumentáció.....	4
Általános specifikáció	4
Rendszerkövetelmények	4
Hardver követelmények	4
Szoftver követelmények	4
A program telepítése	5
XAMPP telepítése.....	5
Adatbázis betöltése.....	9
Webalkalmazás telepítése	13
A program használatának részletes leírása	14
Alaprajz irányítása	14
Útvonal tervezése	14
Adminisztrációs oldal használatának leírása.....	15
Csomópontok.....	15
Termek	16
Fejlesztői dokumentáció.....	17
Témaválasztás	17
Indoklás	17
Használt nyelvek, keretrendszerek, fejlesztői eszközök.....	17
HTML.....	17
CSS	17
JavaScript.....	18
PHP	18
SQL	18
Bootstrap	18
jQuery	18
Three.js.....	18
XAMPP	19
Visual Studio Code	19
Böngészők.....	19
Inkscape	19
Adatmodell leírása.....	20
„admins” tábla	20
„nodes” tábla.....	20

„adjacency” tábla.....	20
„classrooms” tábla.....	20
„aliases” tábla	21
Az adatbázis	21
Részletes feladat-specifikáció, algoritmusok (index.js)	22
async function init()	22
function loadSVG(url).....	22
async function getNodeList().....	22
async function getRoomList().....	22
function planRoute(e).....	23
function showRoute()	23
function bfs(startNode, endNode)	23
Részletes feladat-specifikáció, algoritmusok (admin.js)	24
function addNewNode()	24
async function saveNewNode().....	24
Részletes feladat-specifikáció, algoritmusok (backend)	25
public function dataINRoute()	25
Tesztelési dokumentáció	26
Továbbfejlesztési lehetőségek.....	26
Irodalomjegyzék, forrásmegjelölés	27

Felhasználói dokumentáció

Általános specifikáció

A NavIPARI egy webes felületen elérhető térképnézegető és útvonal tervező alkalmazás. Az oldal fő célja, hogy reményt nyújtson az iskolában magukat elveszettnek érző felhasználóknak. Az alkalmazás kulcs funkciója, hogy az ember kiválaszthat két termet egy kiindulási termet és egy elérni kívánt cél termet és a program megtervezi majd az alaprajzon vizuálisan megjeleníti a legoptimálisabb útvonalat. Kiinduló- és célpontként nem csak terem, hanem az épület ki/bejáratai is megadhatóak. Továbbá fellelhetőek a listában olyan hasznos helyek is, mint például a büfé, társalgó vagy rendszergazdák irodája is.

A webalkalmazás magába foglal egy adminisztrációs oldalt is, ahol a felület használatára jogosult felhasználók bejelentkezés után lehetőséget kapnak a termék listájának szerkesztésére, valamint ezekhez tartozó egyéb elnevezések, aliasok módosítására. Ez a felület szolgál az útvonal tervezéshez szükséges csomópontok felvételére, szerkesztésére.

Rendszerkövetelmények

Hardver követelmények

Abban az esetben, ha a webalkalmazás valamilyen webszerveren való futtatását megoldottuk az oldal a legtöbb modern böngészőben és szinte bármilyen eszköz használatával biztosan működőképes. Különösebb hardverigény nem mérhető, azonban gyengébb teljesítményű grafikus gyorsítóval rendelkező számítógépek, mobil eszközök esetében a felhasználói élmény romlása léphet fel. Ez viszont nem befolyásolja a működést, csak az animációk folyamatosságán és a térkép vezérlésének reakcióidején vehető észre a romlás.

Szoftver követelmények

A működéshez egy olyan webkiszolgálónak kell kiszolgálni az oldalt, ami képes php kódok értelmezésére, valamint futtatására. Erre a célra a fejlesztésnél 2.4.48 verziójú Apache szervert használtam, amihez modulként volt hozzáadva a php 8.0.10 verziójú értelmezője. A kód működése más fajta kiszolgálóprogram, illetve más verziójú php értelmező használatával is működhet, de nem garantált.

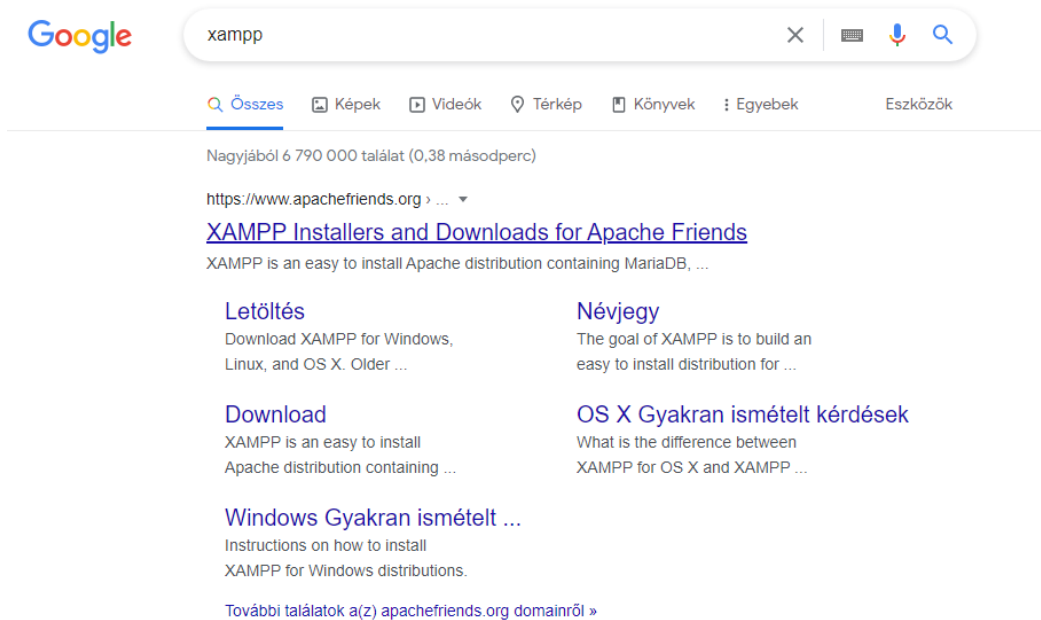
A webkiszolgáló mellett szükséges egy adatbázis szerver is, aminek a kód szerint ugyan azon a gépen kell futnia, mint amin a webszerver is található. A szerveroldali kód a php mysqli kiegészítőjével lett megírva, ebből következik, hogy ezt mindenképp engedélyezni kell a php.ini fájlban. Maga az adatbázis kiszolgáló a fejlesztés alatt a MariaDB 10.4.21 verziója volt.

A program telepítése

XAMPP telepítése

Az előzőleg a szoftver követelményekben felsorolt kiszolgálók mind az XAMPP alkalmazás részeként lettek telepítve mivel ennek segítségével könnyedén és gyorsan megteremthettem a fejlesztéshez kellő feltételeket különösebb háttértudás és hosszas konfigurálgatás nélkül. Ennek használatát tudom javasolni annak is, aki telepíteni kívánja a webalkalmazást.

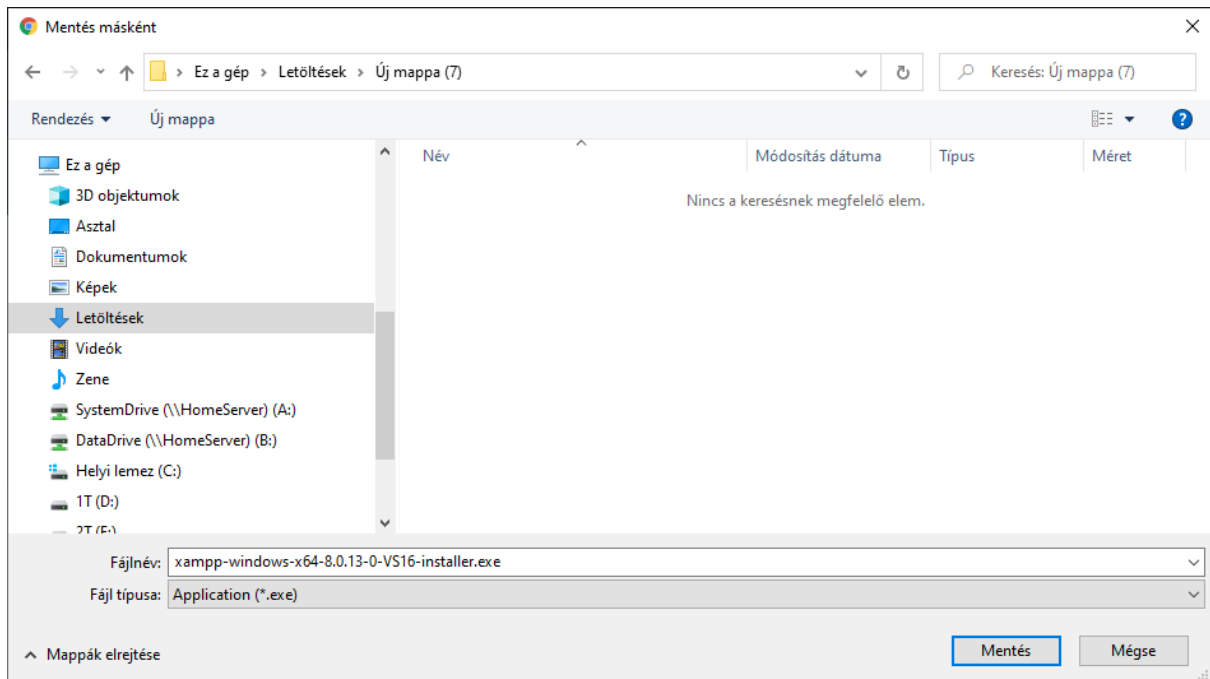
Először is szükség van a program telepítőjére, amit a www.apachefriends.org oldalon egyszerűen ingyen beszerezhető. Elsőnek írjuk be a böngésző keresőmezőjébe, hogy „xampp” és nagy valószínűséggel rögtön az első találat az is lesz, amit keresünk.



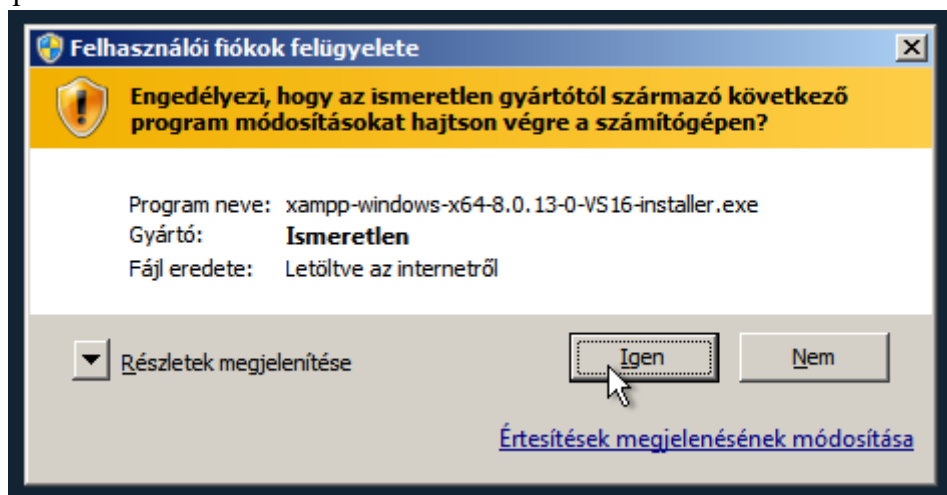
Ezen az oldalon ki kell választani az operációsrendszerünknek megfelelő telepítő fájlt (a továbbiakban én a windows verzió telepítését fogom leírni, de a folyamat menete más rendszereken is hasonló lehet).



A nekünk megfelelő platform kiválasztása után a letöltés el is indul, de az is előfordulhat, hogy előtte meg kell adnunk, hogy a gépünkön hol szeretnénk tárolni a letöltött fájlt.

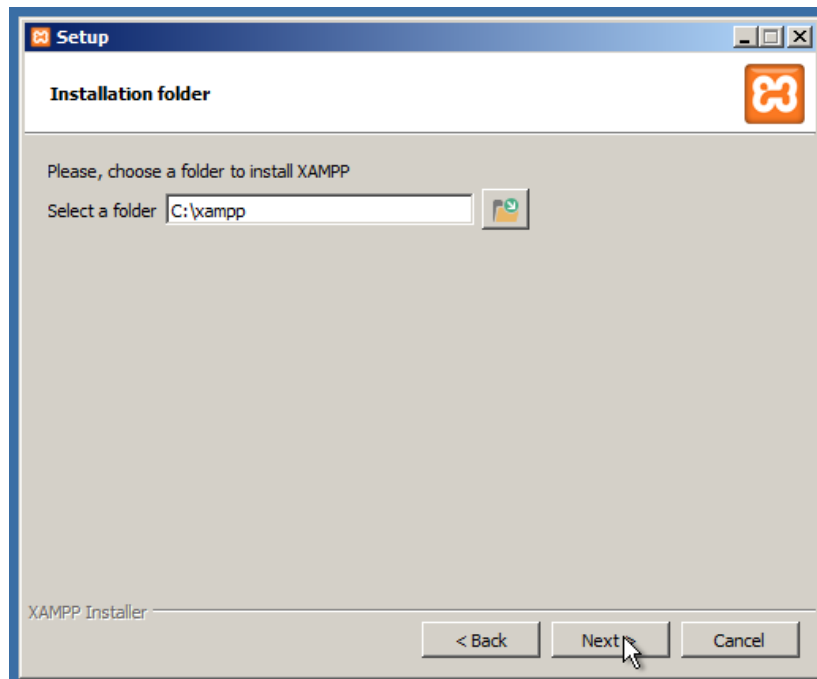


Miután a letöltés befejeződött azonnal el is indíthatjuk a telepítőt. Ha ennél a lépésnél a rendszer megkérdezné, engedélyezzük, hogy a program módosításokat hajtson végre a számítógépen.



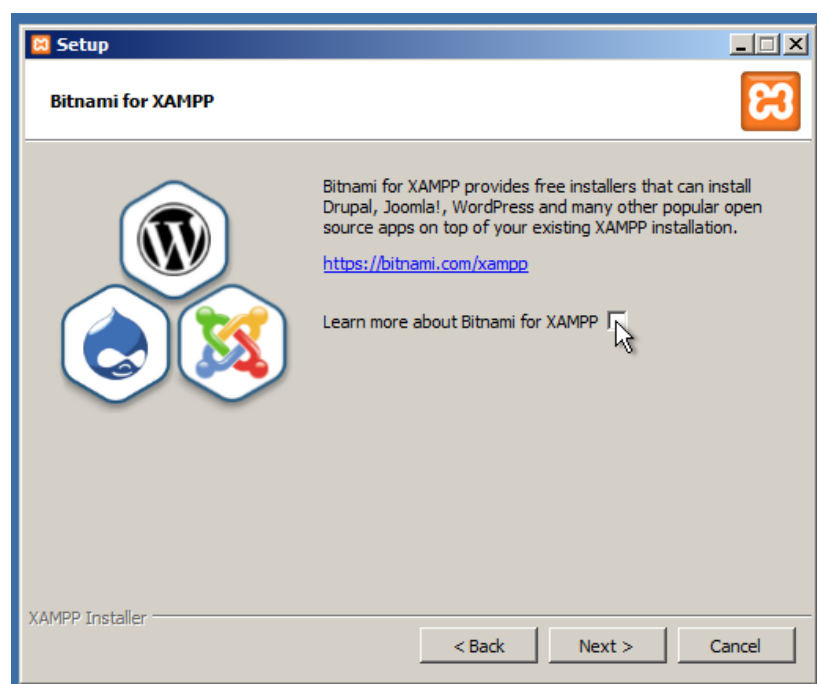
A telepítési folyamat elején a telepítő üdvözl minket, ha a tovább gombra kattintunk kiválaszthatóak a telepíteni kívánt modulok. Itt az alapértelmezett beállítások változtatását csak tapasztalt, hozzáértő felhasználóknak javaslom.

A következő oldalon megadhatjuk hová szeretnénk telepíteni az alkalmazást. Ez még a későbbiekben a weboldal forrásainak felmásolásánál fontos lesz szóval, ha megváltoztatjuk lehetőleg jegyezzük fel.



Az ezt követő panelen választhatunk két telepítési nyelv közül is, név szerint angol és német között.

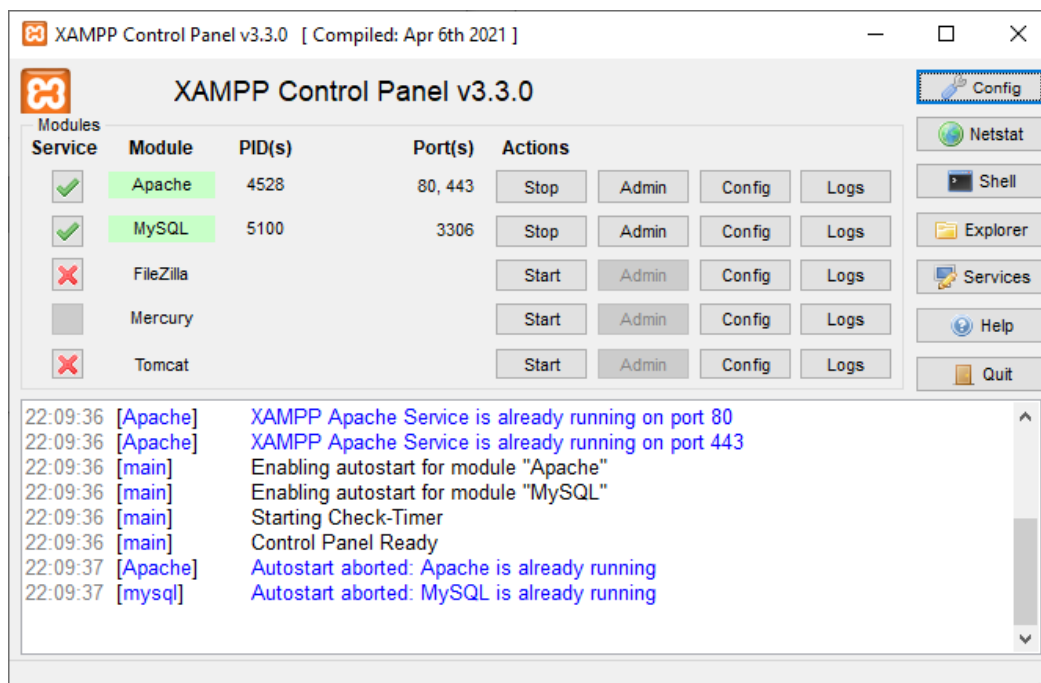
Ha tovább lépünk a telepítő ajánlást tesz pár a XAMPP mellé telepíthető alkalmazásra, valamint felkínálja, hogy a továbblépéskor megnyit a böngészőnkben egy oldalt, ahol részletesebben olvashatunk a témában. Ha erre nem tartunk igényt csak vegyük ki a pipát a jelölőnégyzetből és kattintsunk a tovább gombra.



Ezután a telepítő közli, hogy minden készenáll és ha megint a továbbra kattintunk a telepítés kezdetét is veszi.

A telepítés során, ha esetleg megint engedélyt kérne tőlünk a program engedélyezzük neki a hozzáférést, majd nyomjunk a befejezés gombra a telepítő ablak bezárásához és a xampp vezérlőpult indításához.

Miután elindult a vezérlőpult ellenőrizzük, hogy mind az Apache, mind a MySQL modul sikeresen elindult-e. Ezt a nevük zöld háttere jelzi, valamint az, hogy a velük egy sorban jobbra lévő első gombon a „Stop” felirat szerepel. Amennyiben ez nem mondható el kattintsunk az adott modul nevével egy sorban lévő „Start” gombra.



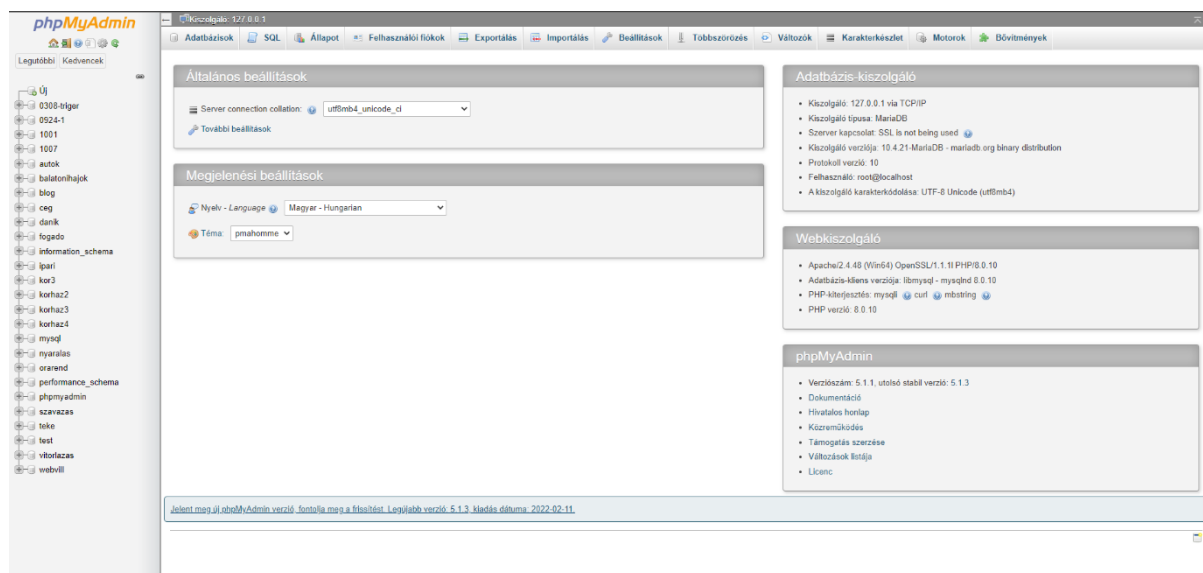
Lehetőségünk van arra is, hogy egy-egy modult az operációs rendszerrel együtt indítsunk. Ennek elérésére a modul nevétől balra található piros „X” gombot nyomjuk meg és a felugró ablakban válasszuk az „igen” opciót. Ha minden jó megy az X egy pipára vált, valamint az ablak alján található szöveges mezőben is a „Successful!” felirat jelenik meg.

Adatbázis betöltése

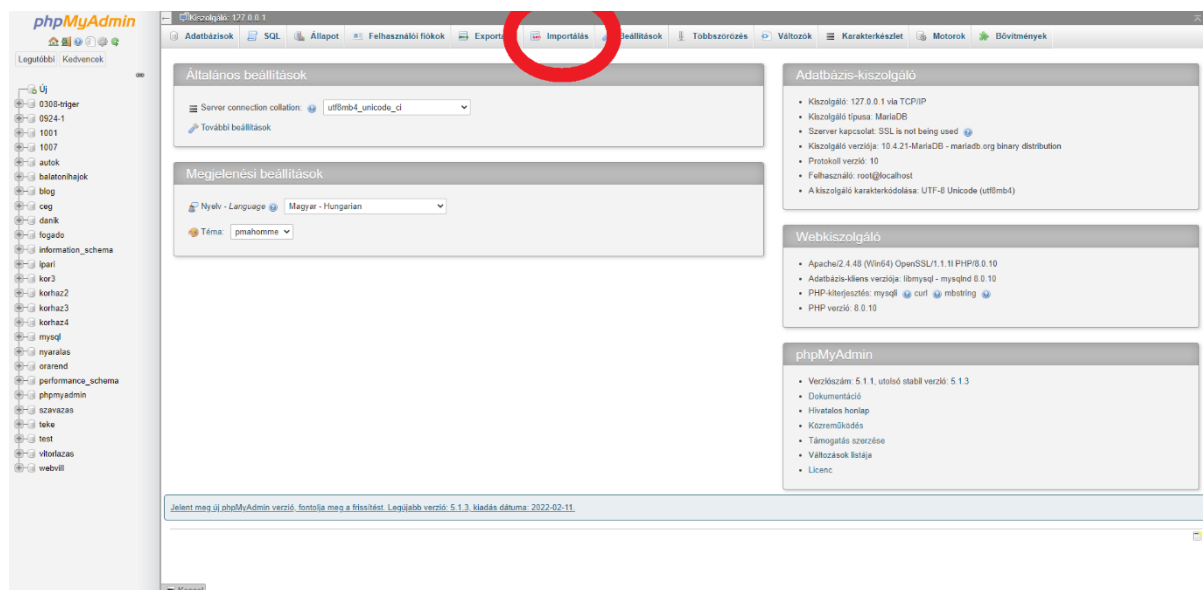
Az alkalmazás működéséhez szükségünk van csomópont és terem adatokra. Ezeket már előre elkészítettem, a kedves felhasználónak csak annyi a dolga, hogy beimportálja az általam biztosított adatbázistfájlt a szerverére. Ehhez a szakdolgozat forrásfájljainak gyökérkönyvtárában található „db_import.sql” fájlra lesz szükség.

Az importáláshoz először is nyissuk meg a böngészőt, majd annak url sávjába írjuk be, hogy „http://localhost/phpmyadmin”

Ha mindent jól csináltunk egy az alábbi képen látható oldalt kell látnunk.

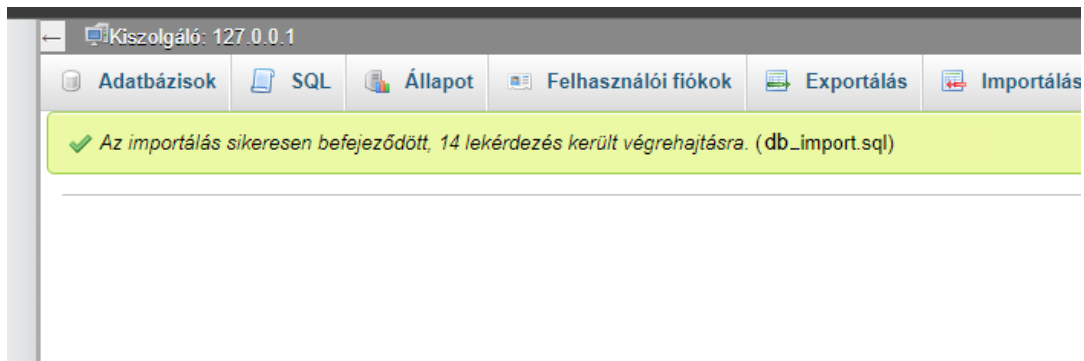


A következő lépésben kattintsunk a felső szalagsávon található „Importálás” gombra

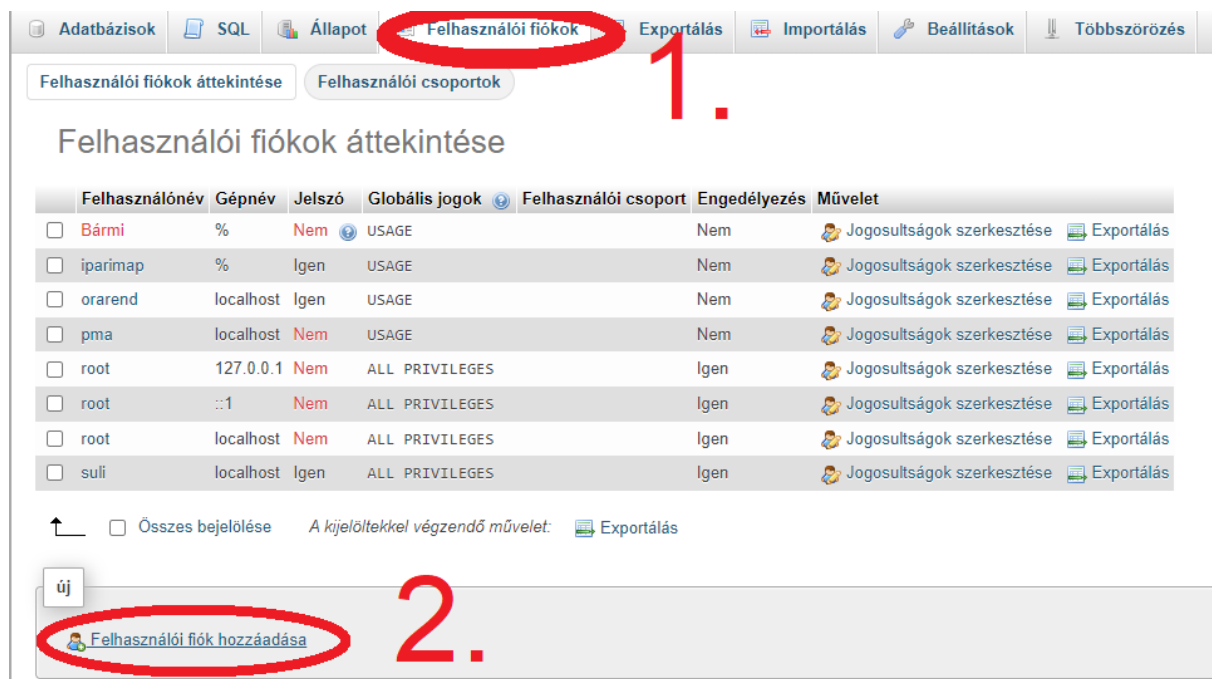


Ezen az oldalon a Fájll kiválasztása gomb segítségével válasszuk ki a feljebb említett „db_import.sql” fájlt, majd az oldal legalján jobb oldalon megtalálható „Indítás” feliratú gomb megnyomásával indítsuk el a folyamatot.

Ha minden jól megy egy zöld mezőben kell látnunk egy feliratot miszerint „Az importálás sikeresen befejeződött”



Ha ezzel megvagyunk még van egy dolgunk ezen a felületen, ugyanis létre kell hoznunk egy felhasználót, amin keresztül majd az alkalmazás hozzáfér az importált adatbázisunkhoz. Ehhez kattintsunk a felső szalagsávon található „Felhasználói fiókok” gombra, az ez után megnyíló oldalon pedig a „Felhasználói fiók hozzáadása” gombra.



Az ez után megjelenő oldalon töltjük ki a bejelentkezési adatok űrlap mezőit az alábbi kép alapján. Felhasználónévnek írjuk be, hogy „iparimap” jelszónak pedig az „1234” számkombinációt adjuk meg. Ne felejtsük el kitölteni a második jelszómezőt se ugyan azzal a jelszóval, amit feljebb megadtunk.

Ha kitöltöttük nyomjunk a lap alján bal oldalon található „Indítása” gombra. Ez után menjünk az „Adatbázis” fülre és állítsuk be, hogy ez a felhasználó hozzáférhessen és kezelhesse a beimportált „ipari” nevű adatbázist.

Ennek kivitelezéséhez válasszuk ki az adatbázis nevét a listából és nyomjunk az „Indítás” gombra. A következő felületen pipáljuk be az „Összes bejelölése” jelölőnégyzetet majd ismét nyomunk az „Indítás” gombra.

Adatbázisok

SQL

Állapot

Felhasználói fiók

Exportálás

Importálás

Beállítások

Többszörözés

Változó

Ön frissítette 'iparimap'@'% ' jogait.

GRANT ALL PRIVILEGES ON `ipari`. * TO 'iparimap'@'% ' WITH GRANT OPTION;

[Szerkesztés helyben] [Módosítás] [PHP-kód létrehozása]

Adatbázis

Tábla

Eljárás

Bejelentkezési adatok

Jogok szerkesztése: Felhasználói fiók 'iparimap'@'% ' - Adatbázis ipari

Adatbázis-specifikus jogok

Összes bejelölése

Megjegyzés: a MySQL jogosultságnevek az angolból származnak.

Adatok

SELECT

INSERT

UPDATE

DELETE

Szerkezet

CREATE

ALTER

INDEX

DROP

CREATE TEMPORARY TABLES

SHOW VIEW

CREATE ROUTINE

ALTER ROUTINE

EXECUTE

CREATE VIEW

EVENT

TRIGGER

Adminisztráció

GRANT

LOCK TABLES

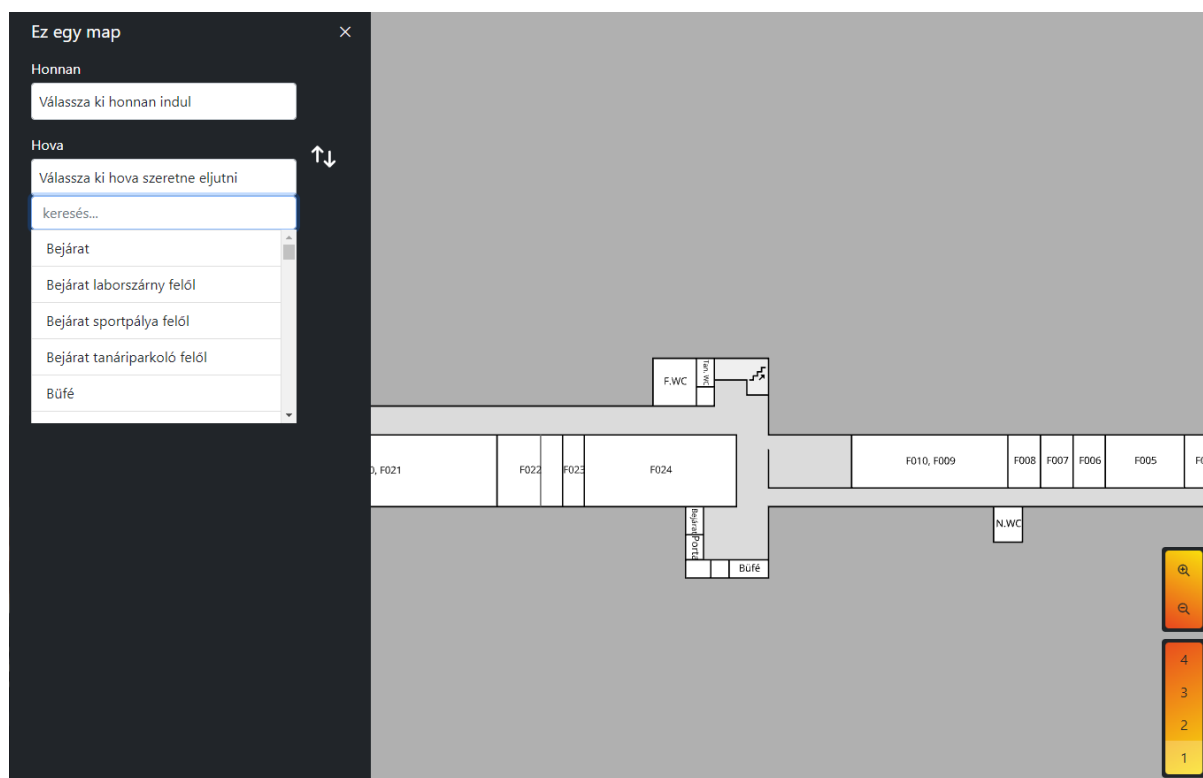
REFERENCES

Indítás

Webalkalmazás telepítése

Ezzel készen is áll az adatbázisunk, tovább léphetünk a webalkalmazás forrás állományainak felmásolására. Ehhez keressük meg a szakdolgozat forrásfájljai között a „htdocs” elnevezésű mappát és az egész tartalmát másoljuk át az XAMPP telepítési könyvtárban található htdocs mappába. Ez a mappa alapértelmezetten, ha nem változtattuk meg a telepítésnél akkor a „C:\xampp\” címen érhető el. A másolás előtt a biztonság kedvéért bizonyosodjunk meg róla, hogy a célmappa üres legyen.

Ha ezt a lépést is elvégeztük a weboldal elméletileg már el is érhető, ennek tesztelésére lépünk vissza a böngészőnkre és az url sávba írjuk be, hogy „http://localhost”. Ha mindent jól csináltunk meg kell, hogy jelenjen az iskolaépület legalsó szintjének alaprajza és a jobb alsó sarokban az aktuális és további szinteket váltó gombok (4 darab gomb: 1,2,3,4 felirattal a nagyítás és kicsinyítés gombok alatt). Továbbá, ha a baloldalt található menüt kinyitjuk a cél vagy kiindulópont mezőkre kattintva az előugró legördülő listákban látnunk kell a termeket.



A program használatának részletes leírása

Alaprajz irányítása

Amikor a felhasználó megérkezik az oldalra rögtön szemben találja magát az iskolaépület viszonylag pontos alaprajzával. Ezen a rajzon fel vannak tüntetve a termek (valamely terem a számával, valamely terem pedig egy könnyebben értelmezhető névvel). Amennyiben valami nem olvasható, esetleg nem fér bele a képbe, akkor lehetőség van a térkép nagyítására és kicsinyítésére is. Ennek módja mobil vagy egyéb érintőképernyős eszközön két ujjal, ezeket egymáshoz közelítve vagy távolítva tehető meg csak úgy, mint a legtöbb alkalmazásban a kicsinyítés/nagyítás gesztusok működnek. Emellett, ha egeret használunk van két további módszer, mindkettő a középső görgethető kerékkel kapcsolatban. Az egyik módszer, hogy a kerék előre vagy hátra görgetésével vezéreljük, a másik pedig, hogy azt lenyomva tartjuk majd közben az egeret felfelé és lefelé mozgatjuk. Arra az esetre, ha bárkinek problémát jelentene az előzőleg felsorolt módok használata az oldal jobb alsó sarkában található két csoportnyi gomb. A felső csoport felelős a nagyítás mértékének állításáért ezeken a gombokon egy nagyító ikon található, amiben vagy egy „+” vagy egy „-” jel szerepel annak megfelelően melyik irányba állítja a nézetet.

Lehetőségünk van a rajz mozgatására is. Ez érintőképernyőn egy ujjas húzással érhető el. Emellett egérrel a bal egérgombot lenyomva tartva az egér mozgatásával érhető el ugyan ez a funkció.

Nagyító/kicsinyítő gombok alatt megtalálhatóak az alaprajz szintjeinek váltását szolgáló gombok. Az aktuálisan kiválasztott szintnek megfelelő gomb háttérszíne mindig egy picit fakóbb, fehéresebb lesz, mint a többi. Ha ezekre kattintunk egy rövid töltőképernyő után az alaprajz átvált az általunk választott szintre.



Útvonal tervezése

Az általunk használt kijelző méretétől függően az útvonaltervező menü előhozására szolgáló gomb pozíciója változhat. Keskenyebb kijelzőkön ez a képernyő tetején helyezkedik el, szélesebb monitoron azonban a képernyő bal oldaláról csúszható elő. Mindkét esetben egy sötét téglalapot kell keresni, benne egy a képernyő közepe felé mutató fehér nyíllal.

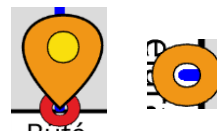
Ha ezt a gombot megnyomjuk ugyan csak a monitor méretétől függően vagy egészben vagy részlegesen az alaprajz fölé fog csúszni az útvonaltervező menü. Itt egy nagyon egyszerű űrlappal találkozhatunk, ahol megadhatjuk, hogy honnan szeretnénk indulni és hová szeretnénk eljutni. A kiindulópont megválasztására jó módszer lehet, ha már az épületben tartózkodunk, hogy a környezetünkben szétnézve keresünk egy teremszámot, amik általában az ajtón, ajtófélfán vagy az ajtó mellett a falon találhatóak.

Ezen két pont kiválasztására használt legördülő lista tartalmaz egy-egy keresőmezőt is, amibe, ha elkezdjük gépelni a keresett terem számának vagy nevének akár részletét az oldal szűri nekünk a találatokat, hogy könnyebben megtaláljuk, amit keresünk.

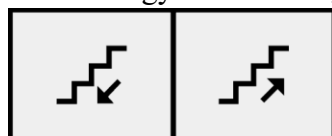
Egy másik hasznos funkció, amit ki tudnánk emelni az a két mező mellett jobbra található két nyilat ábrázoló gomb lenne. Ennek lényege, hogy megcseréli a kiválasztott pontokat, ha esetleg fordított sorrendben jelöltük volna ki elsőre.

Ha kiválasztottuk a két pontot nincs más dolgunk, mint az „Útvonalterv” gombra nyomni. Ez után az oldal megtervezi a legoptimálisabb útvonalat és egy kék vonallal jelzi is azt az alaprajzon.

A kiindulópontot egy narancssárga lyukas karikával jelöli, a célállomást pedig egy hasonló piros karikával és egy narancssárga csepp alakú ikonnal.



Előfordulhat az is, hogy az útvonalat jelző vonal végén nincs semmilyen ikon, ebben az esetben nagy valószínűséggel ez egy lépcsőnél végződik. Ebben az esetben meg kell néznünk, hogy az adott lépcső pontosan milyen irányba visz, felfelé vagy lefelé. Ez a lépcsőikon mellett található nyílból dönthető el, ami mutathat átlósan felfelé és átlósan lefelé is. Ha meggyőződünk róla milyen irányba halad az adott lépcső használjuk a szintek közötti váltásra szolgáló gombokat a jobb alsó sarokban, hogy láthassuk az útvonal további részét.



Az útvonaltervező menü az útvonal megtervezése után automatikusan bezáródik, de tervezés nélkül is bezárható a panel jobb felső sarkában lévő „X” gombbal.

Adminisztrációs oldal használatának leírása

Első sorban ennek elérésére többek között biztonsági okokból sincs hivatkozás elhelyezve a felhasználói oldalon. Elérése úgy történik, hogy a felhasználói oldal elérési címe mögé írjuk, hogy „/admin.php” (ha helyi gépen futtatjuk a telepítési útmutató alapján <http://localhost/admin.php>)

Ha előzőleg még nem voltunk bejelentkezve vagy a munkamenetünk már lejárt az oldal automatikusan átirányít a bejelentkezés oldalra, ahol meg kell adni egy érvényes felhasználónév, jelszó párost. Alap esetben az általam mellékelt adatbázissal ez „admin” és „jelszo”. Sikeres bejelentkezés után visszakerülünk az admin oldalra, ami elsőre nagyon hasonlíthat a felhasználók által látottakra azonban, ha itt nyitjuk meg az „útvonaltervező” menüt, akkor két gombbal találkozunk, illetve a webalkalmazás neve helyett egy „Kijelentkezés” felirat olvasható, amire kattintva kijelentkezhetünk az admin felületről.

Csomópontok

A „Csomópont hozzáadása” gomb megnyomásával kapunk az ablak közepére egy piros pontot, aminek az alaprajzhoz viszonyított pozícióját a rajz húzásával módosíthatjuk.

A pont mellett megjelenik egy kis ablak is a jobb felső sarokban, amin beállíthatjuk, hogy az adott pont lépcsőnek számít-e és azt is megadhatjuk az eddig felvett csomópontok listájából, hogy melyikkel áll szomszédsági kapcsolatban. Az előbbi beállítás helytelen beállításával előfordulhat, hogy más szinten nem fogjuk látni az adott csomópontot. Az utóbbi beállítás pedig nagyon fontos, mivel ezen kapcsolatok alapján generálja az útvonalat az alkalmazás.

Az adott szinthez tartozó előzőleg felvett csomópontok a menüben a két gomb alatt jelennek meg. Ezekre a sárga listabejegyzésekre kattintva szerkesztőmódba léphetünk ahol a kamera újból az adott csomópontra fókuszál és visszkapjuk a felugró tulajdonságablakot is.

A tulajdonságablakon a „Mentés” gombbal véglegesíthetjük beállításainkat, a „Törlés” gombbal törölhetjük a csomópontot, az „X” gombbal pedig elvethetjük a módosításokat.

Mentés után minden pont kap egy egyedi azonosítót, a későbbiekben ez alapján hivatkozhatunk rájuk.

Termek

A „Terem hozzáadása” gomb megnyomásával egy más fajta felugró ablakot hozhatunk elő. Itt meg kell adni a terem nevét, valamint azt, hogy melyik csomóponthoz kapcsolódik az adott terem. Opcionálisan felvehetünk egyéb elnevezéseket, melyek a felhasználóknak segítenek a keresett terem szűrésénél.

256:	53.41866058157 -58.504247361787
90	Igazgatói iroda
91	Titkárság
92	Igazgató igazgatóhelyettes 1

Ha felvettünk egy termet a hozzá tartozó csomópont citromsárgáról narancssárgára vált, ezzel jelezve, hogy az adott csomópontoz tartozik terem.

A felvett termék menüben is megjelennek piros színnel a hozzájuk tartozó csomópont listaelemén belüli al-listaelemként. Termeket ugyan úgy lehet utólag is szerkeszteni, ha rákattintunk a hozzájuk tartozó piros listaelemre.

Egy csomópontoz több terem is rögzíthető azonban egy terem csak egy csomópontoz kapcsolódhat.



Fejlesztői dokumentáció

Témaválasztás

Záródolgozatom témájának egy webes térkép alkalmazás elkészítését választottam. Az alkalmazás célja az lenne, hogy segítséget nyújtson a felhasználó számára eljutni az iskola egyik pontjából az iskola egy másik pontjába. A térképen az Ipari épületének viszonylag pontos alaprajza szerepel szintekre bontva. Meg lehet adni a kiindulási pontot és a célpontot, az alkalmazás pedig segítene megtalálni a keresett termet és megtervezi a legoptimálisabb utat a célig, amit vizuálisan is egy vonallal megjelenít az alaprajzon.

Indoklás

Célközönségnek az Iparis tanárokat, diákokat és látogatókat gondolnám, mivel úgy gondolom tapasztalatból is mondhatom, hogy néha elég nehézkes tud lenni a navigáció az épületen belül. Az ötlet születésének körülményei kissé ironikusak, de röviden az történt, hogy jó pár év után is tudtak olyan terembe küldeni, amiről csak annyit tudtam eldönteni, hogy az épület melyik szárnyában keressem. Ekkor tudatosult bennem, hogy milyen jó lenne egy olyan alkalmazás, ami egy navigációhoz hasonlóan mutatná, hogy merre találok és hogy juthatok el egy adott teremhez.

Használt nyelvek, keretrendszerek, fejlesztői eszközök

HTML

A HTML vagy másnéven HyperText Markup Language egy leíró nyelv, amiben úgynevezett tag-ek segítségével írhatjuk le a weboldalak strukturális felépítését és kinézetét, hogy ezt a böngésző hogyan jelenítse meg. Megkülönböztetünk páros és szimpla tag-eket, előbbieknél onnan a nevük, hogy egy nyitó és egy záró tag közé rakhatunk tartalmat, míg a szimpla tag-nél erre nincs lehetőség. Tag-ek jelölése relációs jelek között a tag neve, a lezáró tag-nél pedig relációs jelek között egy perjel és a tag neve (például: `<div>Ez egy blokk elem</div>`). Jelenleg a szabvány, amit a W3C fejleszt az ötödik verziójánál jár, aminek fő célja, hogy külső beépülő modulok nélkül (adobe flash player és hasonló) hozzassunk létre webalkalmazásokat. ^[1]

CSS

A CSS vagy Cascading Style Sheets egy stílusleíró nyelv, mellyel meghatározható milyen módon jelenjen meg a html oldalunk a böngészőben. Továbbá nem csak HTML vagy XHTML típusú dokumentumoknál használható, hanem gyakorlatilag bármely XML alapú dokumentum stílusát leírhatjuk vele, mint például a szakdolgozatom során széles körben használt SVG fájlokét. ^[2]

JavaScript

A JavaScript egy objektumorientált programozási szkriptnyelv, ami főképp webes felhasználásban elterjedt, de más környezetben is hasznosítható. Segítségével könnyedén hozhatunk létre dinamikus, felhasználói interakcióra reagáló weboldalakat. Futási környezetét tekintve futhat a kliens gépén, de akár szerver oldali kódot is írhatunk a segítségével. Hagyományos felhasználásban azonban csak a kliens oldalon található meg. Ez egy értelmezett nyelv, az az a kód futás időben kerül értelmezésre és végrehajtásra. Típusosságát tekintve gyengén típusos, vagyis a változók típusát az eltárolt érték határozza meg. ^[3]

PHP

A PHP egy széleskörben elterjedt szerveroldali szkriptnyelv. Használatával dinamikus tartalmat vihetünk weboldalunkra. Nagy előnye, hogy a felhasználó böngészője már csak a kód eredményét kapja, így olyan műveleteket is végezhetünk, amik nem lennének célszerűek vagy biztonságosak, ha a felhasználó gépén történnének. A PHP kódok futtatásához a webkiszolgálóhoz php értelmezőt kell kapcsolni, ezt megtehetjük modulként, de megtehetjük CGI vagy másnéven Common Gateway Interface használatával is. ^[4]

SQL

Az SQL vagy Structured Query Language relációsadatbázisokból való lekérdezések leírására szolgál. Segítségével megfogalmazhatjuk milyen adatokat és milyen formában szeretnénk megkapni az adatbázis szervertől. Ez is fontos szerepet játszik a dinamikus weboldalak kialakításában. ^[5]

Bootstrap

A Bootstrap egy ingyenes és szabad felhasználású CSS keretrendszer. Tartalmát főképp a hatalmas css állomány adja, de teljes funkcionalitásához szükség van pár javascript állományra is. Alkalmazásával könnyedén készíthetünk esztétikus és reszponzív weboldalakat. Használata pedig rendkívül egyszerű, a többszáz rendelkezésre álló css osztálykijelölők használatával és kombinációjával érhetjük el a kívánt végeredményt. ^[6]

jQuery

A jQuery egy ingyenes és szabad felhasználású javascript könyvtár. Számtalan függvénnyel járul hozzá a könnyed fejlesztéshez és átlátható kód írásához. Számos ismert cég is használja. ^[7]

Three.js

A Three.js egy ingyenes és szabad felhasználású javascript könyvtár és alkalmazásprogramozási felület (API). Segítségével lehetőség nyílik háromdimenziós grafikák megjelenítésére weboldalunkon. Alkalmazásával hardveresen gyorsított 3D animációk és grafikák készíthetők a böngészőbe egy kis javascript használatával bármilyen beépülő modul vagy plugin nélkül.

XAMPP

A XAMPP egy nyílt forráskódú programcsomag, ami kifejezetten php fejlesztők számára készült, így az alkalmazás tartalmaz webszervert az Apache személyében, előre konfigurált php modullal. Kapunk még egy MariaDB alapú adatbázis szervert és egy webes php alapú vezérlőpultot a kezeléséhez. Be van építve még egy ftp és egy mail szerver is, ha esetleg szükség lenne valamelyikre. Továbbá elérhető egy java alapú webszerver szolgáltatás is. ^[8]

Visual Studio Code

A fejlesztés során a Visual Studio Code fejlesztői környezetet használok, mivel számtalan beépített funkcióval segíti a munkát a kód bizonyos részeinek színezésétől a kiegészítési javaslatokig, valamint a projekt során használt összes nyelvet támogatja így nem volt kérdés, hogy ezt fogom használni.

Böngészők

A fejlesztés során tesztelésre több böngészőt felváltva használtam, hogy megbizonyosodjak róla, hogy mindenhol megfelelően jelenik-e meg a tartalom. Ezek a böngészők számítógépen a Google Chrome, Microsoft Edge és a Firefox mobilkészüléken pedig a Google Chrome mobilváltozata, valamint a Tor Browser.

Inkscape

Az Inkscape egy nyílt forráskódú szabad felhasználású vektorgrafikus képszerkesztő program. Segítségével minőségromlás nélkül méretezhető pixelek helyett vektorokkal operáló grafikákat készíthetünk, ami az én esetemben nagyon fontos volt. Működését tekintve nagyon hasonlít az Adobe Illustrator termékéhez. ^[9]

Adatmodell leírása

A webalkalmazás működéséhez összesen öt tábla szükséges, név szerint ezek a következők:

„admins” tábla

Ez a tábla tárolja az adminisztrációs felület használatára jogosult felhasználók adatait. A táblát alkotó három mezőből az első az „ID”, ami integer típusú számláló (Auto increment) és egyben elsődleges kulcsmező is. Ezen kívül megtalálható még egy „USERNAME” mező, ami varchar típusú. Ennek hosszúsága 100 karakterre van maximalizálva. Végül de nem utolsó sorban van egy jelszó mező „PASSWORD” névvel, ami a jelszavak hasító értékét tartalmazza. Ez is szintén varchar típusú, de ennek maximum hossza csak 65 karakter lehet. Az összes mező úgy van beállítva, hogy NULL értéket ne fogadjon el.

„nodes” tábla

Ez a tábla tárolja a csomópontok összes adatát. Itt is fellelhető egy „ID”, ami elsődleges kulcsként szerepel és számláló típusú mező, a csomópontok azonosítására is használatos. Fontos része a táblának a „POS_X” és „POS_Y” névre hallgató oszlopok, melyek double típusúak. Ezek tartalmazzák az admin felületen elhelyezett pontok scene-en belüli X és Y koordinátáját. A pozíción kívül tárolni kell azt is, hogy az adott csomópont melyik szinthez tartozik. Erre szolgál a „FLOOR” mező, ami integer típusú. A tábla utolsó mezője az „IS_STAIR” elnevezést kapta és feladata, hogy igaz vagy hamis (0 vagy 1) értéket tároljon azzal kapcsolatban, hogy az adott csomópont lépcsőhöz tartozik-e. Ennek a mezőnek a típusa tinyint.

„adjacency” tábla

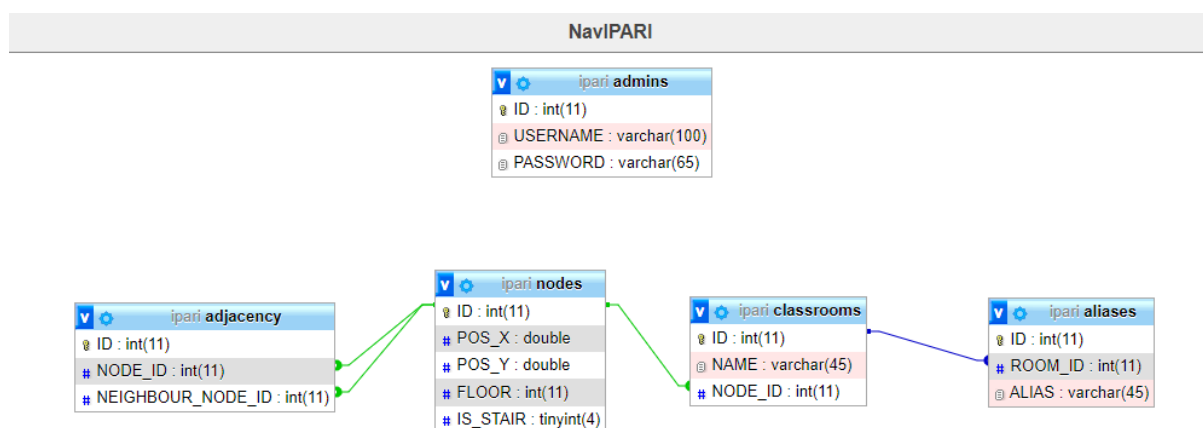
Ez a tábla azért felelős, hogy a csomópontok közötti kapcsolatokat tartsa számon. Elsődleges kulcsmezője az „ID”, ami számláló típusú. Ezen kívül pedig tartalmaz egy „NODE_ID” és egy „NEIGHBOUR_NODE_ID” mezőt, amik a szóban forgó csomópont és egy az azzal kapcsolatban lévő csomópont azonosítóját tárolják. Mindkét mező típusa integer és a NULL értékek nem engedélyezettek. Ez a fajta tárolási struktúra redundanciát eredményez, mivel két csomópont kapcsolatánál két rekord kerül rögzítésre, de ez a módszer nagyban megkönnyíti és rövidíti az alkalmazás kódját.

„classrooms” tábla

A termék adatainak tárolására szolgáló első tábla. Tartalmaz egy számláló típusú elsődleges kulcsmezőt „ID” néven. Továbbá tárolja a terem elnevezését a „NAME” mezőben, ami varchar típusú és maximálisan 45 karakter hosszúságú lehet. Végül rendelkezik egy „NODE_ID” elnevezésű mezővel is, amiben annak a csomópontnak az azonosítóját tárolja, amihez az adott terem tartozik. Ennek a típusa integer és NULL értéket nem fogad el, pont úgy, ahogy a tábla többi mezője sem.

„aliases” tábla

Ez az utolsó tábla, itt vannak tárolva a termék egyéb elnevezései. A tábla rendelkezik egy „ID” mezővel, ami elsődleges kulcs és számláló típusú. Ezen kívül tárolja a terem azonosítóját, amihez tartozik az adott elnevezés. Ez egy integer típusú mező. Maga az elnevezés pedig az „ALIAS” mezőben van elhelyezve, ami varchar típusú és maximálisan 45 karakter hosszú.



Az adatbázis

Az adatok importálása során, ha még nem létezik, akkor létrejön egy „ipari” elnevezésű adatbázis. Minden tábla létrehozása előtt megvizsgálja, hogy létezik-e már az adott tábla és ha igen, akkor törli azt. Ez azért szükséges, hogy ne keletkezessenek duplikált adatok abban az esetben se, ha többször importálnánk be a fájlt. Minden tábla utf8_hungarian_ci karaktertáblára lesz beállítva a létrehozásakor, hogy a magyar ékezetes karakterek tárolásánál véletlen se legyen probléma a tárolással független az adatbázis kiszolgáló beállításaitól.

Részletes feladatspecifikáció, algoritmusok (index.js)

async function init()

Ez a függvény akkor kerül meghívásra, mikor az oldal betöltése befejeződött. Feladata kulcsfontosságú, mivel minden alapbeállítást és folyamatot ez állít be és indít el. Első sorban példányosít és létrehoz egy Three.js scene-t, majd beállítja a kamera méretét és alap pozícióját a képernyő mérettől függően. Ez után létrehoz egy példányt az OrbitControls osztályból, ami a Three.js egyik kiegészítője és a kamera mozgatásáért felelős. Beállítja a _guiData objektum alapadatait, ez a későbbiekben az alaprajz betöltéséhez lesz fontos. Ez után meghívja a loadSVG és a loadIcons függvényeket az alaprajz és az ikonok betöltéséhez. Ha ez lefutott a nagyítás/kicsinyítés gombok kattintás eseményeire állít be, hogy mit tegyenek. Az inicializálás vége felé létrehozza a szintek váltására szolgáló gombokat és az útvonaltervező menü előhívására szolgáló gombokat, továbbá lekérdezi a szerverről a csomópontok és termek listáját a getNodeList és getRoomList függvények meghívásával.

function loadSVG(url)

Ez a függvény rögtön azzal indít, hogy egy új THREE.Scene példányt készít, ezzel alaphelyzetbe állítva a vásznat. Ez után példányosítja a Three.js SVGLoader kiegészítőjének SVGLoader osztályát és meghívja a példány .load metódusát a paraméterként kapott „url” változó értékével, ami a betölteni kívánt .svg fájl elérési útját tárolja. Az url változón kívül második paraméterként egy függvényt ad át még a load metódusnak, amiben az van definiálva, hogy a fájlból beolvasott path, polygon és egyéb svg tag-eket miként jelenítse meg a scene-ben.

async function getNodeList()

Ez a függvény aszinkron POST kérést küld a szerveren található /php/API.php fájlnek és a kérés testében json formátumban átadja, hogy a getNodes funkciót szeretné elérni. Erre a kérésre a szerver egy json formátumban átadott objektum tömbbel válaszol, a választ ideiglenes változóban tárolja a következő lépéshez. A lekérdezés végeztével végigmegy a kapott listán és a listát alkotó objektum ID, POS_X, POS_Y, FLOOR, IS_STAIR és NEIGHBOURS mezőivel egy új példányt hoz létre az általam definiált NODE osztályból, majd ez hozzáadja a globális _Nodes tömbhöz. A _Nodes tömb sikeres feltöltése után bejárja ezt a tömböt is, és minden elemnek beállítja a neighbourNodeList mezőjét egy a megfelelő szomszédos csomópontok mutatóiból álló tömbre az objektum NeighbourIDList mezője alapján.

async function getRoomList()

Ez a függvény aszinkron POST kérést küld a szerveren található /php/API.php fájlnek és a kérés testében json formátumban átadja, hogy a getRooms funkciót szeretné elérni. A kérés eredményét rögtön a _Rooms globális változóba tárolja. A lekérdezés végeztével a _Rooms tömböt abc szerint rendezi.

function planRoute(e)

Ez a függvény paraméterként átveszi a böngésző kattintás eseményének adatait, amit csak annyira használ, hogy megakadályozza az űrlap automatikus elküldését. Második lépésként lekérdezi a kiválasztott kiinduló és cél termék azonosítóját, majd ezek alapján kikeresi ezeket a _Rooms tömbből. Ez után a _Nodes tömbben keresi meg a két teremhez tartozó csomópontot, amiket átad a bfs függvénynek, hogy az megkeresse a legrövidebb útvonalat a két csomópont között. Abban az esetben, ha az útvonal első csomópontja nem az aktuális szinten található a kód betölti a felhasználó számára ezt a szintet, ha pedig erre nincs szükség akkor meghívja a showRoute függvényt. Végezetül pedig bezárja a menüt, hogy egyből látszódjon az elkészült útvonal az alaprajzon.

function showRoute()

Ez a függvény eltávolítja a scene-ből az esetlegesen már jelenlévő ikonokat. Ez után meghívja a drawLine függvényt, aminek átadja a _Path globális változóban tárolt útvonal tömb másolatát, amiből a drawLine függvény az aktuális szintre vonatkozó részeket vizuálisan is megjeleníti a scene-ben. Majd annak feltételében, hogy az útvonal eleje vagy vége az aktuális szinten van elhelyezi az új ikonokat.

function bfs(startNode, endNode)

Ez a függvény a NODE osztály egy-egy példányát várja mindkét paraméteréül, majd a Breadth-first search elnevezésű gráf keresőalgoritmus alkalmazásával meghatározza a startNode paraméterben kapott csomóponttól indulva milyen sorrendben mely más csomópontok meglátogatásával juthatunk el az endNode paraméterben kapott csomópontig.

Először is a _Nodes minden elemének isVisited és prevNode mezőjét alapértékre állítja, majd létrehoz egy tömböt a queue változóban, amibe beletölti a kiinduló csomópontot és isVisited mezőjét igazra állítja. Ezt követően indít egy ciklust, ami addig megy amíg a queue tömb hossza nem nulla. A cikluson belül kiveszi a queue tömb első elemét és végig lépked annak szomszédjain. Ha az aktuális szomszéd még nem volt meglátogatva (az isVisited mező értéke hamis), akkor hozzáadja ezt a queue tömbhöz, valamint az isVisited mezőjét igazra állítja, továbbá előző csomópontként (prevNode) beállítja azt az elemet, aminek éppen a szomszédjainak listáján megy végig.

Miután a queue tömb kiürül a program összeállítja az útvonal csomópontjainak mutatójából álló tömböt. Ehhez egy ciklust futtat, ahol a ciklusváltozót beállítja az endNode változóban kapott csomópontra. A ciklus addig fut amíg a ciklusváltozó értéke nem lesz null érték és minden ciklus végén a ciklusváltozó felveszi értékül a ciklusváltozóban tárolt csomópont prevNode mezőjében tárolt csomópontot. A ciklus testében a path segéd tömbhöz adjuk a ciklusváltozó értékét.

A segédváltozóban tárolt útvonal ezen a ponton visszafelé van szóval meghívásra kerül a reverse metódus a tömbön. Ez után, ha a tömb első eleme az az elem, amit paraméterként a startNode változóban megkapott a függvény akkor visszaadja ez a tömböt, ha ez nem mondható el akkor a keresés nem jutott eredményre, ezért egy üres tömböt ad vissza a függvény. ^[11]

Részletes feladatspecifikáció, algoritmusok (admin.js)

function addNewNode()

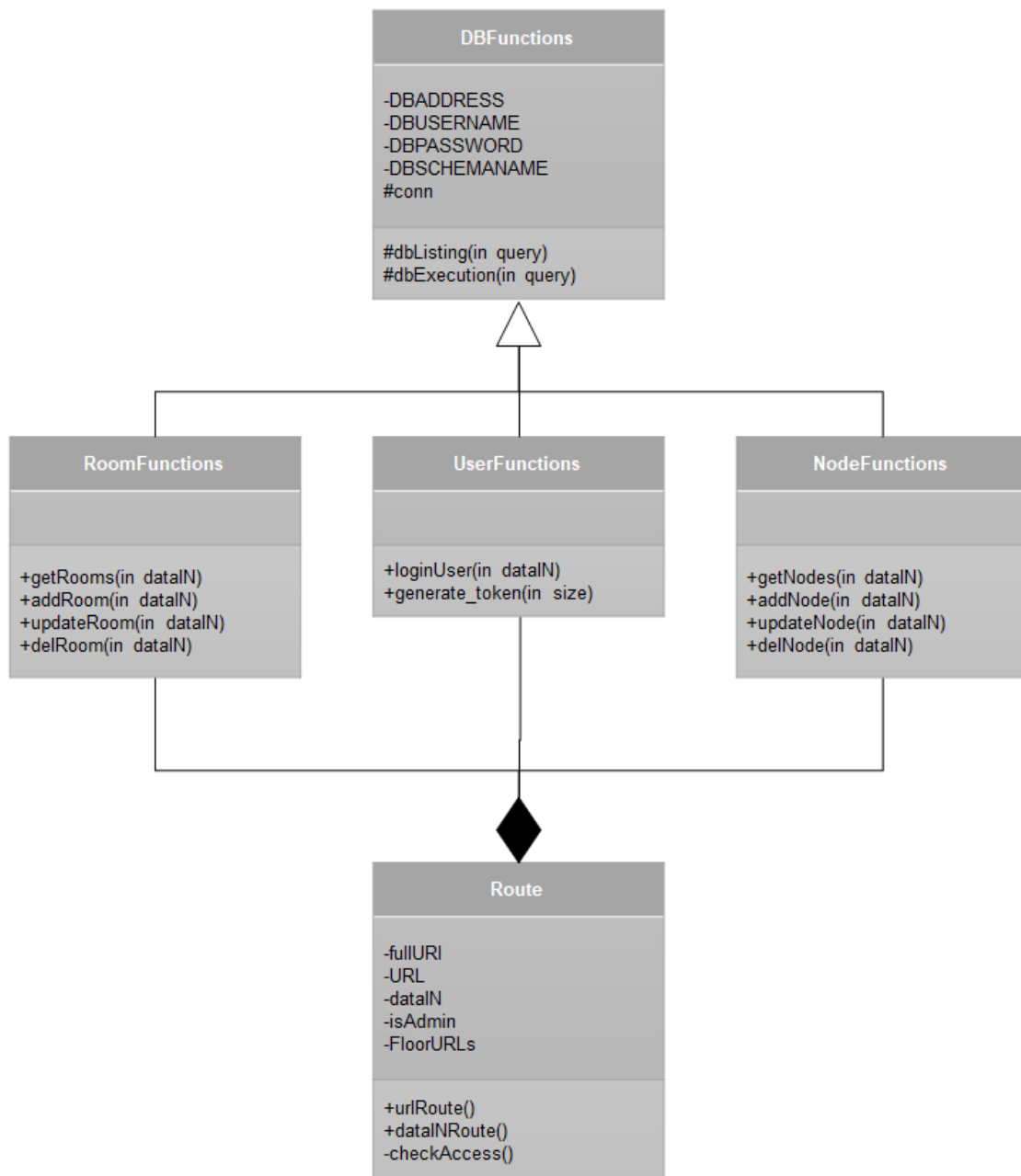
Ez a függvény akkor kerül meghívásra, ha az adminisztrátor felhasználó az admin oldalon a csomópont hozzáadása gombra kattint. A program először megvizsgálja, hogy van-e már éppen módosítás alatt másik csomópont vagy terem, ha nincs akkor alaphelyzetbe állítja a jobboldali tulajdonságpanel mezőit és megjeleníti azt, majd feltölti a csomópontok listáját az adott emelet csomópontjaival. Ez után lekéri a kamera pozícióját és a createNodeCircle függvény segítségével létrehoz egy új piros színű pöttyöt ezen a ponton. Végezetül létrehoz és az EditedNode globális változóba tárolja a NODE osztály (Nem ugyan az, mint amit feljebb részleteztem, az a felhasználói oldalhoz tartozik) egy új példányát -1 azonosítóval, az előzőleg létrehozott pöttyel és az aktuális szint számával.

async function saveNewNode()

Először is a függvény ellenőrzi, hogy éppen új vagy meglévő csomópont mentéséről van-e szó. Amennyiben az isEditMode értéke igaz, az az már létező csomópont módosítása történik, akkor aszinkron POST kérést küld a szerveren található /php/API.php fájlra és a kérés testében json formátumban átadja az azonosítót, pozíciót, hogy lépcső-e, a szomszédok listáját és hogy csomópont módosítási művelet történik. Abban az esetben, ha új csomópont kerül felvételre az előzőhöz hasonló kérést küld annyi különbséggel, hogy nem küld azonosítót és jelzi, hogy csomópont felvétel történik.

A sikeres művelet után mindkét esetben a program frissíti a _Nodes tömbben található csomópontok szomszédjait, ha ez szükséges. Továbbá az EditedNode és EditedNeighbours változók értékét alaphelyzetbe állítja, valamint frissíti a vizuálisan megjelenő kapcsolatokat szimbolizáló piros összeköttetéseket az egyes csomópontok között.

Részletes feladatspecifikáció, algoritmusok (backend)



public function dataINRoute()

A metódus a kérés testében átadott json formátumú adatokat objektummá alakítja, majd az objektum function mezője alapján eldönti melyik műveletet kell végrehajtania és a megfelelő osztályt (felhasználó műveletek, csomópont műveletek vagy terem műveletek) példányosítja, valamint meghívja annak szükséges metódusát.

Tesztelési dokumentáció

A webalkalmazás tesztelését nagy részben én magam végeztem. Egy-egy modul elkészültekor teszteltem annak megfelelő működését majd, ha valamilyen problémát tapasztaltam strukturális teszteléssel kerestem a hiba forrását, hogy elhárítsam azt. Amikor az alkalmazás abba a stádiumba ért barátaim bevonásával bétatesztelést végeztünk, ahol rengeteg apró hibát sikeresen megtaláltunk és elhárítottam. Egyik és legnagyobb ezek közül az, hogy Firefox böngésző használatánál az alaprajz svg fájljának stílusai nem voltak értelmezve és az egész rajz fekete volt.

A bétateszt alatt sok hasznos tapasztalatot gyűjtöttem a módszerrel kapcsolatban és rendkívül hasznosnak találtam. A jövőbeli projektjeimnél is feltétlen alkalmazni fogom.

Továbbfejlesztési lehetőségek

A jelenlegi verzióban nincs lehetőség az adminisztrátor felhasználók egyszerű szerkesztésére, csak az adatbázisban turkálva lehet ezeket módosítani. A jövőben remek fejlesztés lehetne, ha készülne egy menüpont, ahol egy adminisztrátor fel tud venni vagy módosítani tudja az meglévő adminisztrátorokat.

A felhasználói felület fel van készítve arra, hogy dinamikusan töltse be az alaprajzok listáját, azonban ez a szerveroldali kódban egy konstans változóból van kiszolgálva. A jövőben meg lehetne oldani, hogy az adminisztrációs felületen fel lehessen tölteni másik alaprajzokat, ezzel egyszerűbben naprakész maradhatna az alkalmazás az épület változása esetén is.

Az rendelkezésre álló időbe már nem fért bele, hogy egyértelműbbé tegyem a felhasználó számára, hogy az útvonalat jelző vonal melyik irányból melyik irányba tart. Ez akár valami animációval, vagy a statikus szín textúrára cserélésével könnyen orvosolható lehetne.

Irodalomjegyzék, forrásmegjelölés

1. <https://hu.wikipedia.org/wiki/HTML> (megtekintve: 2022. április 08.)
2. https://hu.wikipedia.org/wiki/Cascading_Style_Sheets (megtekintve: 2022. április 08.)
3. <https://hu.wikipedia.org/wiki/JavaScript> (megtekintve: 2022. április 08.)
4. <https://hu.wikipedia.org/wiki/PHP> (megtekintve: 2022. április 08.)
5. <https://hu.wikipedia.org/wiki/SQL> (megtekintve: 2022. április 08.)
6. <https://hu.wikipedia.org/wiki/Bootstrap> (megtekintve: 2022. április 08.)
7. <https://hu.wikipedia.org/wiki/JQuery> (megtekintve: 2022. április 08.)
8. <https://hu.wikipedia.org/wiki/XAMPP> (megtekintve: 2022. április 08.)
9. <https://hu.wikipedia.org/wiki/Inkscape> (megtekintve: 2022. április 08.)
10. A töltőképernyőnél és a böngésző címsorában megjelenő logó az iskola tanár diák közös tárhelyéről származik (megtekintve: 2022. április 08.)
11. A gráfokkal kapcsolatos elméleti tudást és a használt keresőalgoritmust WilliamFiset youtube felhasználó által feltöltött videókból szedtem össze. <https://www.youtube.com/watch?v=DgXR2OWQnLc&list=PLDV1Zeh2NRsDGO4--qE8yH72HFL1Km93P> (megtekintve: 2022. április 08.)