Veszprémi Szakképzési Centrum Ipari Technikum

**Szakképzés neve: Szoftverfejlesztő**

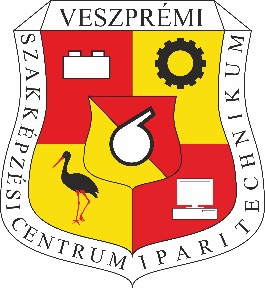
**OKJ száma: 54 213 05**

SZAKDOLGOZAT

Iskolai helyettesítések

Gyurik Pál Tóth Roland

témavezető 13.C



Veszprém, 2024

**Tartalomjegyzék**

[**Bevezető, a probléma rövid ismertetése** 3](#_Toc157952794)

[**Felhasználói dokumentáció** 4](#_Toc157952795)

[**Hardver követelmények** 4](#_Toc157952796)

[**Szoftver követelmények** 4](#_Toc157952797)

[**Fejlesztői dokumentáció** 5](#_Toc157952798)

[**Felhasznált programozási nyelvek, keretrendszerek** 5](#_Toc157952799)

[**HTML (HyperText Markup Language)** 5](#_Toc157952800)

[**CSS (Cascading Style Sheets)** 6](#_Toc157952801)

[**JS (JavaScript)** 7](#_Toc157952802)

[**PHP (Hypertext Preprocessor)** 7](#_Toc157952803)

[**Bootstrap** 8](#_Toc157952804)

[**Az adatbázis terve:** 8](#_Toc157952805)

[**Feladatspecifikáció:** 8](#_Toc157952806)

[**Adatigények:** 8](#_Toc157952807)

[**Műveleti igények:** 9](#_Toc157952808)

[**Az adatbázis felépítése:** 9](#_Toc157952809)

[**Összegzés** 11](#_Toc157952810)

[**Források** 11](#_Toc157952811)

# **Bevezető, a probléma rövid ismertetése**

A Veszprémi Szakképzési Centrum Ipari Technikum által jelenleg használt helyettesítések megjelenítésére szolgáló weboldal nem képes értesítéseket küldeni a diákoknak és tanároknak az elkövetkezendő helyettesítésekről. A problémára a Kréta rendszere sem nyújt megoldást, mivel ez a rendszer sem megbízható az értesítések küldésében. Az általam fejlesztett új helyettesítések megjelenítésére, valamint azokról értesítések küldésére szolgáló weboldal ezen problémák kiküszöbölésére nyújt megoldást. A weboldalon belépést követően a felhasználó akár e-mailen, akár Google naptáron keresztül történő értesítésre tud feliratkozni az őt éríntő helyettesítésekről.

# **Felhasználói dokumentáció**

## **Hardver követelmények**

A weboldal localhostról futtatható (valamint nincs domain bérlés), ezért az eszköznek egy laptopnak vagy egy asztali számítógépnek kell lennie. A minimális és ajánlott hardverkövetelmény a fejlesztői szerverkörnyezet minimális és ajánlott, a XAMPP minimális és ajánlott, valamint a böngésző memóriahasználata alapján alakul.

Minimális hardverkövetelmény

* Operációs rendszer: Windows 10 vagy OS X 10.14 vagy az előbb említett rendszereknél frissebb verzió
* Memória: legalább 2 GB
* Lemezterület: legalább 500 MB szabad lemezterület
* Processzor: legalább 1 GHz-es processzor

Ajánlott hardverkövetelmény

* Operációs rendszer: Windows 10 vagy OS X 10.14 vagy az előbb említett rendszereknél frissebb verzió
* Memória: legalább 8 GB
* Lemezterület: legalább 1 GB szabad lemezterület
* Processzor: legalább 2 GHz-es processzor

## **Szoftver követelmények**

Ajánlott szoftverek:

Böngészők:

* Google Chrome: Aktuális verzió [2023.12.03.]: 119.0.6045.200
* Mozilla Firefox: Aktuális verzió [2023.12.03.]: 120.0.1
* Microsoft Edge: Aktuális verzió [2023.12.03.]: 119.0.2151.97

XAMPP: Aktuális verzió [2023.12.03.]: 8.2.12

# **Fejlesztői dokumentáció**

## **Felhasznált programozási nyelvek, keretrendszerek**

### **HTML (HyperText Markup Language)**

A HTML egy leíró nyelv, amely az internetes tartalmak strukturálására és formázására szolgál. Első változata 1990-ben jelent meg, azóta nagyon sok újításon esett át az alapok megtartásával, ezért napjainkban már a HTML5-ös szabványt használjuk.

A weboldalon megjelenő tartalmak leírása tag-ek (címkék) között található. Ezek a címkék szabják meg azt, hogy a böngésző hogyan értelmezze a különböző tartalmi részeket. A tag-ek relációs (< és >) jelek között helyezkednek el, valamint nem érzékenyek kis és nagybetűre, de célszerű, vagy csak kisbetűkkel, vagy csak nagybetűkkel használni őket. A tag-ek többsége páros, azaz van egy nyitó és záró része az utóbbi annyiban különbözik a nyitó címkétől, hogy a relációs jelek közé egy / jel is kerül. A különböző elemek lehetnek inline és blokk szintű elemek. A böngésző a blokk szintű elemeket egy tömbként jeleníti meg, azaz nem kerülhet az elem mellé másik elem. Ezzel szemben az inline elemek kisebb szintű szerkezeti elemek és több is kerülhet belőlük egymás mellé. Inline elemek például: hivatkozások, képek. Blokk elemek például: címsorok, bekezdések, táblázatok, listák.

A HTML dokumentum fő részei:

* <!DOCTYPE html> (Dokumentum típus meghatározás): A weboldal nyelvtanát adja meg a böngésző számára. Ha üresen kerül megadásra a böngésző trükköző üzemmódban próbálja meg értelmezni a kódolást.
* <html>...</html>: Az e közötti rész alkotja a böngésző által értelmezett és megjelenített weblapot.
* <head>...</head> (Fejrész): A HTML oldalra vonatkozó meta-adatokat tartalmazza. A meta-adatok a fájllal kapcsolatos különféle jellemzők és utasítások tárolására szolgálnak.
* <body>...</body> (Törzs): A HTML dokumentumnak azon része, amit a böngésző megjelenít, illetve a stílus meghatározások is erre a részre érvényesek.

### **CSS (Cascading Style Sheets)**

Stíluslapnyelv, amely segítségével a HTML dokumentum formázása, elrendezése valósítható meg. Korábban a HTML nyelvben található címkék voltak arra használva, hogy a dokumentum elemeit formázzák, de ez jelentősen megnövelte a kód hosszát és nehezen átláthatóvá tette. A stílusnyelvnek köszönhetően teljesen külön lehet választani a tartalmi és a megjelenítési részt.

Három fajta módon használható a stíluslap megadása:

* Küldő stíluslap: Egy külön CSS állományban adjuk meg a HTML elemeire vonatkozó stílus beállításokat. Ez a CSS állomány akár több HTML állományhoz is kapcsolható, közösen szerkeszthető. Segítségével a HTML és CSS kód teljesen külön választható, ezzel jól áttekinthetővé válik mindkét kód.
* Beágyazott stíluslap: Csak az adott HTML dokumentum elemeit formázhatjuk. Nem kell külön állományt létrehozni és letölteni, a CSS kód a HTML oldal elején, a fejrészben látható.
* Inline stílus: Közvetlenül egy elemhez rendelhetünk stílust. A HTML kód átláthatatlanná válhat, illetve több esetben sem hatékony. (Például, ha több elemnek is ugyan azt a stílust szeretnénk megadni.)

A stílusok megadása egy kijelölőből és egy meghatározásblokkból állnak. Például: H1 { color: red; }

A kijelölők típusai:

* elem (például: h1)
* csoport (például: h1,h2)
* osztály (például: .osztaly)
* azonosító (például: #azonosito)
* ál-osztály (például: :hover)
* kombinátorok (például: child>)

A reszponzív weboldal tökéletesen alkalmazkodik különböző multimédiás eszközök (mobil, tablet, laptop, monitor) képernyő vagy ablakméretéhez azok felbontásától függetlenül. A reszponzív weboldal kulcsfontosságú elemei rugalmas méretűek, így a különböző képernyő méreteken dinamikusan tud csökkeni, vagy nőni a szélességük. A viewport (nézetablak) meta egy nagyon fontos része a reszponzív tervezésnek. A viewport meta elem segítségével felülírhatjuk a böngésző által előre meghatározott nézetablakot.

### **JS (JavaScript)**

A JavaScript egy könnyűsúlyú, interpretált nyelv (értelmező program szükséges a futtatására, ilyen például egy böngésző), amely lehetővé teszi a dinamikus, interaktív weboldalak létrehozását. A Netscape által lett 1995-ben bevezetve, azóta folyamatosan fejlődött és az internetes alkalmazások hajtómotorává vált. A JavaScript nemcsak a kliensoldalon használatos, hanem a Node.js révén a szerveroldalon is alkalmazzák. Ez lehetővé teszi a fejlesztők számára, hogy a kliens- és szerveroldali logikát egységesen fejlesszék, ami hatékonyabb és összehangolt alkalmazásokhoz vezet. A JavaScript kiterjeszthetőségének köszönhetően számos keretrendszert és könyvtárat használhatunk, például: React, Angular vagy Vue.js.

Főbb lehetőségeink a JavaScripttel:

* Futási időben módosíthatjuk a weboldalt vagy annak egy részét
* Futási időben hozhatunk létre új HTML elemeket
* Futási időben módosíthatjuk a weboldal stílusát
* Számításokat végezhetünk kliens oldalon
* Űrlap adatait ellenőrizhetjük már kliens oldalon
* Ajax/aszinkron kéréseket küldhetünk és a kérésből érkező választ feldolgozhatjuk
* Az oldalon történő eseményekre reagálhatunk. Például: click, load
* Valós idejű kommunikációt folytathatunk a szerverrel

### **PHP (Hypertext Preprocessor)**

Szerveroldali szkriptnyelv, amely kifejezetten webfejlesztésre lett tervezve. Rasmus Lerdorf által 1994-ben lett létrehozva, a PHP azóta fejlődött és vált az egyik legelterjedtebb és népszerűbb szerveroldali webprogramozási nyelvek közé. A PHP kódokat a szerveren futtatják, és a generált HTML-t küldik el a böngészőnek, így a felhasználók a feldolgozott eredményt láthatják. A PHP együttműködik más technológiákkal, például az Apache vagy a Nginx webszerverekkel, és könnyen integrálható más nyelvekkel, például HTML, CSS, JavaScript.

A PHP erőssége a dinamikus weboldalak készítése. Az interaktivitás és a dinamizmus érdekében a PHP lehetővé teszi a változók használatát, a felhasználói bemenetek feldolgozását, az adatbázisokkal való kommunikációt, és más szerveroldali feladatokat.

Fontos figyelembe venni a biztonsági kérdéseket, mint például az SQL injection vagy a cross-site scripting. A fejlesztőknek rendszeresen frissíteniük kell a PHP verziót, hogy kihasználhassák a legújabb fejlesztéseket és biztonsági javításokat.

### **Bootstrap**

Bootstrap egy népszerű, nyílt forráskódú front-end keretrendszer, amelyet a Twitter fejlesztett ki. Célja a gyors és egyszerű webfejlesztés elősegítése, különösen a reszponzív és mobilbarát tervezések létrehozásában. Bootstrap számos előre elkészített stíluslapot (CSS) és JavaScript komponenst kínál, amelyek lehetővé teszik a fejlesztők számára, hogy könnyedén és egységesen alakítsák ki weboldalaikat és alkalmazásaikat.

Az egyik kiemelkedő tulajdonsága a reszponzivitás támogatása. A Bootstrap segítségével készült weboldalak és alkalmazások automatikusan alkalmazkodnak a különböző eszközök méreteihez és képességeihez, például asztali számítógépek, táblagépek és mobiltelefonok. Ezáltal egyetlen kódalap segítségével valósítható meg a többplatformos kompatibilitás.

A Bootstrap előnye a könnyen használható és testreszabható komponensekben rejlik. A fejlesztők számára elérhetőek előre elkészített gombok, navigációs sávok, űrlapok, modálok és egyéb vizuális elemek, amelyek segítik a gyors és egyszerű felhasználói felületek kialakítását. Emellett a keretrendszer lehetővé teszi a saját stílusok és komponensek könnyű hozzáadását vagy testreszabását is.

## **Az adatbázis terve:**

### **Feladatspecifikáció:**

A helyettesítések oldal számára olyan adatbázis készítése, amely a helyettesítések kiírásának, és ezek által érintett diákok, valamint tanárok adatainak nyilvántartására, tárolására és megfelelő kezelésére alkalmas.

Az adatbázis adminisztrációja nagyrészt helyi (localhost) hálózatról zajlik majd, az adminisztrációt várhatóan egy rendszergazda, és néhány felelős személy végzi majd, illetve bizonyos adminisztrációs feladatok Web-felületen történő végrehajthatóságára is igény van.

### **Adatigények:**

Az adatbázisnak tartalmaznia kell a következő adatokat:

Diák felhasználó esetén:

* A diák felhasználóneve, jelszava, email címe, valamint az osztálya

Tanár felhasználó esetén:

* A tanár felhasználóneve, jelszava, email címe, valamint a tanár neve

A helyettesítések eltárolásához szükséges adatigények:

* Az iskola összes osztálya
* Az iskola összes tanárának a neve
* Az elkövetkezendő helyettesítések eltárolásához szükséges adatigény:
  + Az óra száma
  + A helyettesítendő, valamint a helyére kiírt tanár neve
  + A helyettesítendő, valamint a helyére kiírt tantárgy neve
  + A helyettesítendő, valamint a helyére kiírt terem száma

### **Műveleti igények:**

* Új felhasználó adatainak felvitele
* Felhasználó adatainak módosítása
* Felhasználó adatainak törlése
* Új helyettesítés felvitele
* Helyettesítés módosítása
* Osztályok módosítása
* Tanárok módosítása

### **Az adatbázis felépítése:**

felhasznalok:

* id (int, 5): Azonosító
* felhasznalonev (varchar, 30): Felhasználónév
* email (varchar, 100): E-mail cím
* jelszo (varchar, 60): Jelszó
* osztalyId (int, 2): Az osztály azonosítója, idegen kulcs az osztalyok táblára
* tanarId (int, 2): A tanár azonosítója, idegen kulcs a tanarok táblára
* ertesites (tinyint, 1): Értesítés beállítása

helyettesites:

* ora (int, 1): Az óra sorszáma
* osztalyId (int, 2): Az osztály azonosítója, idegen kulcs a osztalyok táblára
* tanarId1 (int, 2): Az helyettesítendő tanár azonosítója, idegen kulcs a tanarok táblára
* tanarId2 (int, 2): A helyettesített tanár azonosítója, idegen kulcs a tanarok táblára
* tantargy1 (varchar, 30): A helyettesítendő tantárgy
* tantargy2 (varchar, 30): A helyettesített tantárgy
* terem1 (varchar, 5): A helyettesítendő terem
* terem2 (varchar, 5): A helyettesített terem

osztalyok:

* id (int, 2): Azonosító
* osztalynev (varchar, 5): Az osztály neve

tanarok:

* id (int, 2): Azonosító
* tanarnev (varchar, 50): A tanár neve

Backend, fontos kódrészletek:

# **Összegzés**

# **Források**

<https://hu.wikipedia.org/wiki/Google_Chrome> [megtekintve: 2023.11.28.]

<https://hu.wikipedia.org/wiki/Mozilla_Firefox> [megtekintve: 2023.11.28.]

<https://hu.wikipedia.org/wiki/Microsoft_Edge> [megtekintve: 2023.11.28.]

<https://webiskola.hu/html-ismeretek/html-oldal-felepitese-strukturaja/> [megtekintve: 2023.11.30.]

<https://hu.wikipedia.org/wiki/JavaScript> [megtekintve: 2023.12.02.]

<https://hu.wikipedia.org/wiki/PHP> [megtekintve: 2023.12.02.]

<https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)> [megtekintve: 2023.12.03.]