

Széchenyi István Egyetem
Gépészmérnöki, Informatikai és Villamosmérnöki Kar
Informatika Tanszék

SZAKDOLGOZATKÉSZÍTÉS I.

Tóth Benjámín

Mérnök informatikus Bsc szak

2021



SZÉCHENYI
ISTVÁN
EGYETEM



SZAKDOLGOZATKÉSZÍTÉS I-II.

Óvodapedagógusokat segítő nyilvántartó
rendszer

Tóth Benjámín

Mérnök Informatikus Bsc szak

2021

Nyilatkozat

Alulírott, Tóth Benjámín (D0JXB2), Mérnök Informatikus Bsc szakos hallgató kijelentem, hogy a Óvodapedagógusokat segítő nyilvántartó rendszer című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, 2019.....

hallgató

FELADATKIÍRÁS

Szakdolgozat címe: Óvodapedagógusokat segítő nyilvántartó rendszer

Hallgató neve: Tóth Benjámin

Szak: Mérnök Informatikus

Képzési szint: BSc

Típus: nyilvános

Feladat:

Egy olyan komplett rendszer elemzése, megtervezése és kivitelezése, ami a mindennapokban megkönnyítheti az óvodapedagógusok adminisztratív munkáját.

Részfeladatok:

a) A szakdolgozat elkészítéséhez szükséges résztvékenységek:

Feltáró elemzés egy óvodában a dolgozók és vezető óvónő közreműködésével. A nyilvántartó rendszer adatbázisának megtervezése és létrehozása. Az egyes részfeladatokat ellátó szoftvermodulok megtervezése és elkészítése. Főbb adatok: óvodapedagógusok, dolgozók, gyerekek, eszközök. Főbb funkciók: teljeskörű adatkarbantartás; eszköznyilvántartás; utazási költségtérítés, étkeztetés, szabadságok adminisztrációja; az óvónők minősítéséhez szükséges PDF fájlok kezelése.

b) A szakdolgozat főbb részei:

Az adott óvoda jelenlegi informatikai „rendszerének” rövid bemutatása. A feladat megoldására készített szoftver tervezési dokumentációja.

Belső konzulens: Pusztai Pál egyetemi adjunktus, Matematika és Számítástudomány Tanszék

Külső konzulens: Csala Mátyás, Kockázatkezelési elemző, Budapest Bank Zrt.

Győr, 2019.01.25.

tanszékvezető

belső konzulens

Tartalomjegyzék

Tartalomjegyzék	5
BEVEZETÉS	7
Az első fázis (előkészítés):	7
A második fázis (tervezés):	7
A harmadik fázis (fejlesztés):	7
Technológiák története	8
PHP (Hypertext Preprocessor):	8
Php történelme:	8
Php felépítése:	9
PHP keretrendszer (framework):	9
Php alkalmazásának területei:	11
HTML történelme	12
HTML felépítése:	12
CSS (Cascading Style Sheets):	14
A CSS története:	14
A CSS elemek:	14
JavaScript:	16
JavaScript történelme:	16
MySQL:	17
MySQL történelme:	17
Web alkalmazás (WEB application):	18
Yii2 keretrendszer	18
Fejlesztői segédeszközök, szoftverek	19
Laragon:	19
PHPStorm:	20
Lightshot:	20
Óvodapedagógusokat segítő nyilvántartó rendszer	21
Irodalomjegyzék	22
Mellékletek	24

● BEVEZETÉS

Szakdolgozatom célja egy közeli óvoda informatikai rendszerének használhatóbbá, átláthatóbbá tétele. Az óvónők és óvodapedagógusok hétköznapijainak megkönnyítése, a munkájukhoz elengedhetetlen dokumentumok gyorsabb elérése, rendszerezése. Ennek érdekében egy új rendszer fejlesztését tűztem ki célul, amit végül szakdolgozati témának is sikeresen elfogadtak. A feltáró elemzésben megvizsgáltam az eddig használt eszközöket, programokat, valamint azt, hogy az egyes dolgozók mihez férhetnek hozzá. A program moduljaihoz összegyűjtöttem a kellő információkat, amiket később valós adatként, valamint a fejlesztés során tesztadatként is fel tudok használni. Ilyen például az eszközökről készült leltár, aminek egy részét Excel táblában kaptam meg.

Szakdolgozatom első fele a kiválasztott technológiai háttér bemutatásáról, jellemzéséről fog szólni, valamint a választott adatbázis és programozási nyelv fejlődését is bemutatom.

A második fele a kutatómunka, az egyetemi tanulmányok és a munkahelyen szerzett tapasztalatok felhasználásával az óvodapedagógusokat segítő rendszer fejlesztése, használható, az óvoda dolgozóinak is átlátható program átadása.

Szakdolgozatom fejlesztését három fázisra bontom.

Az első fázis (előkészítés):

A vezetőóvónővel és az óvodapedagógusokkal való megbeszélések lebonyolítása, az aktuális rendszer feltérképezése, ez alapján a fejleszteni kívánt rendszer alapvető funkcionalitásainak összegyűjtése.

A második fázis (tervezés):

Az adatbázis megtervezése, a folyamatok, funkciók megfogalmazása, ez által modulok kialakítása. A modulok közötti kapcsolatok, adatbáziskapcsolatok összegzése, amik végül a rendszer egészét alkotják.

A rendszer alapjainak lefektetése során a megfelelő programozási nyelv, keretrendszer megválasztása is meghatározó szerepet töltött be a tervezési időszakban. A kompatibilitás és elérhetőség miatt PHP nyelv mellett döntöttem, valamint ebben programozási nyelvben eddig megszerzett tapasztalataim alapján és a moduláris felépítés miatt a Yii2 Framework-öt választottam.

A harmadik fázis (fejlesztés):

A tervek, drótvázlatok és megbeszélések alapján a munka megkezdése, és a folyamatok végig vitele. A modulok leprogramozása, összeillesztése, a meghatározott kapcsolatok alapján.

Technológiák története

PHP (Hypertext Preprocessor):

Php történelme:

A PHP programozási nyelv megalkotója Rasmus Lerdorf, a nyelv „őst” 1994-ben készítette, ami csak szkriptek összessége volt, saját munkája megkönnyítésére, 1995-ben Personal Home Page Tools néven tette közzé az első publikus változatot. A PHP-t még abban az évben kibővítette egy HTML-űrlap-feldolgozóval (Form Interpreter) és adatbázis-kezelő rendszert kezelő függvényekkel. 1997-ben a programkód szinte teljesen megváltozott, és hivatalos neve ezzel egy időben PHP: Hypertext Preprocessor-ra változott, önálló, elsősorban webalkalmazások készítése alkalmas, szerveroldali programozásnyelvé alakult.

Ezek után, 1998-ban megjelent a PHP 2, 1999-ben a PHP 3, 2000-ben a PHP 4, 2004-ben a PHP 5, ami sokáig az egyik legelterjedtebb nyelv volt. Jelenleg a PHP 8-as verzió a legújabb, ami számos hasznos újítással kecsegtet.

Az PHP-t többször átirták, nyelv funkciói egyre bővültek, egyre hatékonyabb lett, jelenleg a szerveroldali webprogramozás legelterjedtebb programozási nyelve.

A nyelv minden platformon egységes, alkalmas platformfüggetlen alkalmazások készítésére, futtatható Linux, Windows, HP-UX, Solaris és OpenBSD operációs rendszereken is. A PHP programozási nyelv implementációja a PHP-interpreter, az a futási környezet, amely képes értelmezni és végrehajtani a kód utasításait.

Jellemző szoftverkönyezet az Apache webservert, és a MySQL adatbázis-kezelő rendszer, melyet túlnyomó többségben LINUX operációs rendszeren (LAMP – Linux + Apache + MySQL + PHP) használnak, de a PHP számos webserverral és adatbázis-kezelő rendszerrel képes együttműködni. Windows rendszeren (WAMP – Windows + Apache + MySQL + PHP) a megfelelő alkalmazások és megfelelő beállítások mellett könnyedén futtatható, illetve fejlesztésnek is nekikezdhethünk ezen a rendszeren.

Több programcsomag is ingyenesen elérhető, ami a technológiák összehangolását oldja meg helyettünk, ilyen az EasyPHP, a WAMP, vagy a XAMPP, de természetesen az Apache, a MySQL és a PHP különálló telepítésére is van lehetőség

Php felépítése:

A Php kódokat a HTML tag-ek közé ágyazzuk be, amik lehetnek a HTML kódok előtt, után, vagy közben. Illetve a Php kódrészlet is lehet külön-külön, vagy írhatjuk az összes kódot egybe. A Php nyelvben nincsenek erre vonatkozó megkötések, a lényeg, hogy a php kódrészletek mindig egy Php nyitó tag-gel kezdődnek és egy Php záró tag-gel végződnek. Ez a szintaktika nagyon fontos, mivel a webszerver ebből fogja tudni, hogy az adott kódrészlet, Php kódhoz, amihez indítani kell az értelmezőt.

A Php kódhoz tartozó, rendszeresített tag-ek a következők:

- nyitó tag: „<?php”
- záró tag: „?>”

Rövidített formájának használata függ a rendszerbeállításoktól, illetve a verziótól, ezek szintaktikája pedig

- Rövidített nyitó tag: „<?”
- Rövidített záró tag: „?>”

▪ PHP keretrendszer (framework):

Az alkalmazás gyors fejlesztését, karbantarthatóságát, valamint átláthatóságát csak megfelelő szabályok bevezetésével biztosíthatjuk, alapvetően ezekre szolgálnak a keretrendszerek. A keretrendszerek lényegében szabályok összessége, melyek kódkönyvtárakat biztosítanak az általánosan használt funkciókhoz, csökkentve a fejlesztéssel járó kódolás mennyiségét.

Keretrendszer használatának előnyei:

1. **Gyorsabb fejlesztés:** Mivel a PHP keretrendszerek beépített könyvtárakkal és eszközökkel rendelkeznek, a fejlesztéshez szükséges idő kevesebb. Ilyenek például azok a parancssori eszközök, amik a készítenő funkciók, migrációk, szkriptek, html-lapok keretét kvázi legenerálják. Nekünk szinte csak a logikát kell a keretbe írunk. Az általam választott Yii2 keretrendszerrel ilyen például a migráció készítés (yii migrate/create), vagy az MVC generátor (Gii).
2. **Kevesebb kódolás:** A legtöbb feladat, amit a webfejlesztőknek kódolniuk kell, általános, ezért érdemes használni az előre megírt (keretrendszerbe foglalt)

függvényeket. Például a tömbkezelés (array) műveletei, keresés (find/search) és sorbarendezés (sort) műveletei.

3. **Alapfeladatok (common tasks) kódjai:** Mivel a feladatok alapjai részben, esetenként teljesen megegyeznek, ezért közös könyvtárakat hoztak létre. Így ahelyett, hogy az általa megírt kódot „copy-paste”-elnie kéne, elég csak saját feladataival, magának a logikának a kódolásával foglalkoznia.

Ilyen kódsorok lehetnek:

- űrlapok (formok) validálása (típusellenőrzés, méret ellenőrzés)
- CRUD (create, read, update, delete) műveletek előtöltése, generálása
- MVC (Model, View, Controller) generálás és előtöltés

Kódolási gyakorlatok, kódolási minták alkalmazása: Egy keretrendszer használata lehetőséget ad, illetve megkönnyíti az egyes bevett kódolási gyakorlatok alkalmazását.

- Kódolási minták (design pattern), és anti-minták (anti patterns), tartalmazzák azokat a use-case-eket, amiket kezdő és tapasztalt programozóknak egyaránt be kellene tartaniuk.

Pattern lehet többek között:

- abstract factory, builder, prototype, singleton
- adapter, bridge, facade, proxy
- interpreter, iterator, observer

Antipattern lehet:

- spaghetti code, the blob
- the feud, fire drill
- Kódolási gyakorlat, felépítés: Keretrendszer választása során, az egyik fő szempont, hogy az adott rendszer milyen könyvtárstruktúrát (kódolási gyakorlatot) alkalmaz, továbbá, elnevezési konvenciókat is tartalmaz.

Ezzel átláthatóbb, karbantarthatóbb, rendezettebb kódot kapunk.

4. **Karbantarthatóság:** Egy keretrendszer lehetővé teszi, az előző pontok betartása

mellett a megírt kódok újrafelhasználhatóságát. (DRY – Don't Repeat Yourself).

5. **Biztonságosság:** PHP programozás során a szerver felé rengetek kérés érkezik, ezeket a kéréseket egy keretrendszer automatikusan validálja, ellenőrzi, jól átgondolt módon, a lehető legkevesebb kikaput hagyva a hackerek, és biztonsági rések feltérképezésére fejlesztett szoftverek, vírusok ellen.

Ilyen biztonsági rések lehetnek: Cross site scripting (XSS), Injections (SQL, directory, command, code), session hijacking, fájlok feltöltése, PHP konfigurációk módosítása stb.

A keretrendszerek használata nem helyettesíti a biztonságért felelős kódrészletek helyes megírását, de minimalizálja a rendszert érő, a már ismert támadások sikerességét.

6. **Csapatmunka segítése:** Keretrendszerbe foglaltak szerint kell meghozni a tervezési döntéseket, be kell tartani a keretrendszer által preferált szabályokat. Valamint nem utolsó sorban, meghatározza a dokumentáció írásának módját, stílusát, irányelveit.

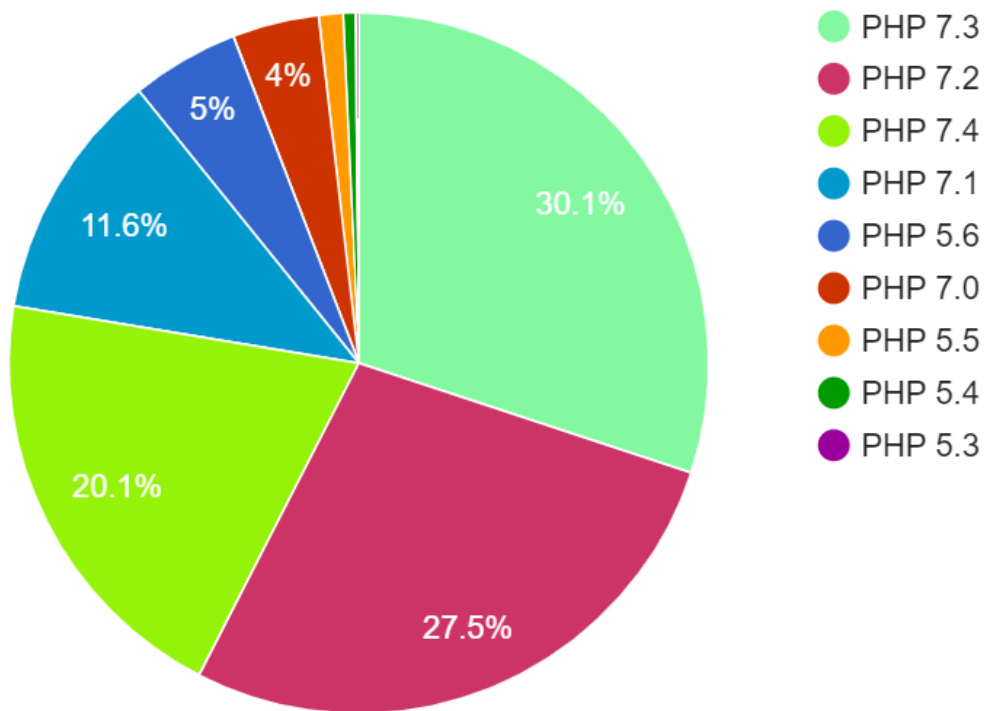
Egy keretrendszer használata egyértelmű alapszabályokat határoz meg a projekthez, Amennyiben egy másik fejlesztő nem ismeri a projektet, képesnek kell lennie arra, hogy a keretrendszer által gyorsan megtanulja az összefüggéseket és együttműködve tudjon dolgozni a csapattöbbi tagjával.

Php alkalmazásának területei:

Mivel a php jelenlegi tudása és sokoldalúsága miatt, minden a weben elérhető, azzal kapcsolatos feladatot gyorsan és biztonságosan képes elvégezni, ezért szinte mindenre kiterjedő, általánosan felhasználható programnyelvről beszélünk. A nyelvet scriptek formájában alapvető feladatok elvégzésétől, portálokon (webshopok, fórumok, egyéb weblapok) keresztül, komplex backend és frontend (vagy akár még több réteggel) ellátott web alkalmazások kiszolgálásáig bármire használható. Ezek függhetnek valamilyen adatbáziskezelő rendszertől.

Ezek alapján nem meglepő, hogy a jelenlegi világháló több mint 80%-át ezen a nyelven fejlesztik, tartják karban, valamint

PHP Versions Grouped 2020-05



Php verziók elterjedtsége (2020.05)

HTML (Hypertext Markup Language):

HTML történelme

A Sir Tim Berners-Lee hozta létre 1991 végén, azzal a szándékkal, hogy webböngészőkön keresztül olvasható és elérhető információkat osszon meg.

A HTML 1.0 1993-ban jelent meg, de kiforratlansága miatt nem terjedt el igazán, nem sokan vettek részt a fejlesztésében, így a nyelv lassan fejlődött.

Ezután következik az 1995-ben megjelent HTML 2.0, amely tartalmazza a HTML 1.0 összes funkcióját és azt a néhány további funkciót, amely 1997 januárjáig a weboldalak tervezésének és létrehozásának standard jelölőnyelveként maradt meg. Ezután jön a HTML 3.0, továbbfejlesztett új szolgáltatásokkal, amelyek hatékonyabb és könnyebben fejleszthető környezetet adott weboldalak tervezéséhez, de az új szolgáltatások nagy mértékben lassították a böngészőt.

Ezután jön a HTML 4.01, amely végül széles körben elterjedt, és a HTML sikeres verziója lett, a HTML 5.0 előtt. A HTML 5 2012-ben jelent meg, amely jelenleg is a legelterjedtebb jelölőnyelv a weboldalak és webes alkalmazásokhoz.

Sajnos lehetőségei nagyon korlátozottak a felhasználói interakcióra (hivatkozások, gombok és űrlapok), ezen hátrányokat kiküszöböli az AJAX (Asynchronous JavaScript and XML), azaz aszinkron JavaScript és XML.

HTML felépítése:

A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak, mint formázást használják, ugyanis ezen lapok teljes funkcionalitása a php-re épül, ami lehet statikus vagy dinamikus.

Statikus HTML lap-ról akkor beszélhetünk, ha szövegek, táblázatok, linkek, és képek beszárási pontjait, elrendezését és stílusát írja le, ezeket tag-eknek nevezzük. Nem tartalmaz olyan adatot, ami időben megváltozhat, az információk kvázi „bele vannak égetve”. Az ilyen fájlok kiterjesztése a „.html” vagy „.htm”.

Tag-ek lehetnek például:

- bekezdések,
- táblázatok,
- címsorok,
- tárolók,
- linkek,
- listák
- képek,

Dinamikus HTML lap ugyanazokat az elemeket tartalmazza, mint a statikus HTML lap. Azonban ennek elemei valamilyen program futása közben kerül generálásra. Általában egy félkész, keretet adó „.html” vagy „.htm” formátumú fájl hiányzó adatait / változóit tölti ki. Ilyen program ok nyelve lehet például a PHP, JAVA, Microsoft.NET.

A PHP egy úgynevezett interpretert (értelmezőt) használ, amely általában egy külső modulja a webservernek. Az interpreter feldolgozza a PHP utasításokat, és csak a kész (HTML) kimenetet küldi el a böngészőnek.

A fejleszteni kívánt alkalmazást vagy weblapot elég egyszer lefejleszteni, és az szinte minden platformon működőképes lesz, tehát független a kliens gép operációs rendszerétől.

Azonban nem független a böngészőtől, és annak verziójától, ami különböző problémákat okozhat. A HTML, CSS, DOM implementációk, illetve a szkriptek futtatása vagy azok futásának tiltása, illetve a szkriptnyelv verziója, a verziók támogatottsága, valamint az eltérő felhasználói böngésző-beállítások szintén zavaró hibákat okozhatnak.

```
<!DOCTYPE html>
<html lang="<?= Yii::$app->language ?>">
<head>
    <meta charset="<?= Yii::$app->charset ?>">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <?php $this->registerCsrfMetaTags() ?>
    <title><?= Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>
<body>
    <?php $this->beginBody() ?>

    <div class="wrap">
        <?php NavBar::begin([
            'brandLabel' => Yii::$app->name,
            'brandUrl' => Yii::$app->homeUrl,
```

1. Html tag példa, php kóddal (saját projekt)

CSS (Cascading Style Sheets):

A CSS története:

A HTML-lel együtt, 1990 óta van jelen. Stíluslapok megjelenését tudják módosítani, eredetileg ezeket a felhasználók használták, mivel a HTML korai verziók kevés attribútumot tartalmaztak, így a felhasználóra volt bízva, hogy a webes dokumentumok hogyan jelenjenek meg. A fejlesztés során egyre több stilisztikai lehetőségre volt szükség, így a HTML nyelvbe egyre több elem került

A CSS eredetileg Håkon Wium Lie ötlete volt 1994-ben, majd Bert Bos-al közösen döntöttek a CSS kifejlesztése mellett. Ekkor már több stílusleíró nyelv is létezett, de a CSS volt az első, ami több különböző stíluslapból tevődhetett össze, így lehetővé téve azt, hogy a felhasználó által megadott stílus bizonyos esetekben felülírja a szerző stílusát. A stíluslapok így a szerző és a felhasználó számára rugalmas vezérlést biztosított.

Ebben az időben alakult meg a World Wide Web Consortium is, mely később fellépett a CSS érdekében, és megalapított egy bizottságot a részletes kidolgozására

1996 decemberében a CSS level 1 hivatalosan is megjelent. 1997 elején a W3C egyik csoportjához került, ami azokkal a problémákkal foglalkozott, melyeket a CSS Level 1 kihagyott. A CSS Level 2, 1998 májusában jelent meg.

A CSS segítségével különböző stíluslapokat hozhatunk létre és ágyazhatunk be HTML honlapjainkba, ezek a stíluslapok befolyásolják az oldal megjelenését. Azt, hogy az egyes HTML tagek hogyan jelenjenek meg, meghatározhatjuk a méretüket, színüket, stílusukat, típusukat stb.

A tervezése során az egyik legfontosabb szempont az volt, hogy a dokumentumok struktúrája (pl.: HTML) el legyen különítve a dokumentum megjelenésétől (melyet CSS-sel lehet megadni).

Az ilyen elkülönítésnek több haszna is van:

- növeli a weblapok használhatóságát, rugalmasságát
- növeli a megjelenítés kezelhetőségét
- csökkenti a dokumentum struktúrájának komplexitását
- átláthatóbb, tisztább kódot eredményez

A CSS a dokumentum stílusát a megjelenítési módszer függvényében is meg tudja adni. Ilyen lehet a képernyő, nyomtatási lap vagy akár hangos böngésző, esetleg braille készülék.

A CSS elemek:

Egy CSS 2 fő része a szelektor, és a deklaráció.

A szelektor kiválasztja a megfelelő elemcsoportot, amit amit formázni szeretnénk.

CSS szelektorok kiválasztása lehet:

- Minden elem: *
- Az elem neve: pl.: „span”, „p” vagy „h2” (minden azonos HTML elemhez)
- Leszármazottak: pl.: „li a”, olyan 'a' elemekre, melyek egy 'li' elem alatt vannak, annak részei
- attribútumok alapján: pl.: class, id: class="osztály" vagy id="azonosító"

A deklaráció leírja a szelektorban megadott HTML tagek formázását. Ennek általában több része van (paraméter, érték), mivel meg kell adni, hogy mit/miket akarunk a szelektorban formázni. Pl.: class = „custom zöld class”, vagy style = „width:128px; height:128px”

```
17  .footer {
18      height: 60px;
19      background-color: #f5f5f5;
20      border-top: 1px solid #ddd;
21      padding-top: 20px;
22  }
23
24  .jumbotron {
25      text-align: center;
26      background-color: transparent;
27  }
28
29  .jumbotron .btn {
30      font-size: 21px;
31      padding: 14px 24px;
32  }
33
34  .not-set {
35      color: #c55;
36      font-style: italic;
37  }
```


JavaScript:

JavaScript történelme:

JavaScript eredetileg Brendan Eich, a Netscape Communications mérnöke fejlesztette ki, az 1990-es évek elején. A nyelvet először a Netscape 2.0-ás verziója támogatta.

A JavaScript egy szkriptnyelv, amely azon három alapvető nyelv egyike, amelyet weboldalak fejlesztésére használnak. Míg a HTML és a CSS a webhely struktúráját és stílusát adja, a JavaScript lehetővé teszi további funkcionalitás hozzáadását a webhelyéhez. ezzel lehetővé téve a webhely látogatóinak, hogy interaktív módon lépjenek kapcsolatba a tartalommal.

JavaScript segítségével futásidőben dolgozhatjuk fel a felhasználó által létrehozott eseményeket (Pl.: Gombra kattintás, Űrlapmező kitöltése, Űrlap elküldése, Böngésző bezárása, stb.).

A JavaScript elsősorban a kliens számítógépen fut, a böngészőben, a Node.js bevezetése lehetővé tette a szakemberek számára, hogy a szervereken is futtassanak JavaScript kódot. Megjelenése óta a JavaScript felülmúlta a Java, a Flash és más nyelveket.

JavaScript főbb tulajdonságai:

- Interaktivitás: A felhasználó által létrehozott események kezelhetősége.
- Biztonság: A JavaScriptben nincs olyan utasítás vagy függvény arra, amivel elérhetővé válnának a kliens fájljai.
- Böngészőfüggetlenség: Már minden nagyobb böngésző támogatja a JavaScript használatát.
- Tehermentesíti a kiszolgálót: Már az űrlapok elküldésekor leellenőrizheti, hogy minden űrlapmező ki van-e töltve és ha nincs akkor már kliens oldalon figyelmezteti a Felhasználót. (Viszont nem helyettesíti a szerveroldali ellenőrzést!)

```

function inspect( dataType ) {
    var selected;
    inspected[ dataType ] = true;
    jQuery.each( obj: structure[ dataType ] || [], callback: function( _, prefilterOrFactory ) {
        var dataTypeOrTransport = prefilterOrFactory( options, originalOptions, jqXHR );
        if ( typeof dataTypeOrTransport === "string" &&
            !seekingTransport && !inspected[ dataTypeOrTransport ] ) {

            options.dataTypes.unshift( dataTypeOrTransport );
            inspect( dataTypeOrTransport );
            return false;
        } else if ( seekingTransport ) {
            return !( selected = dataTypeOrTransport );
        }
    } );
    return selected;
}

```

3. JQuery (javaSript) példa kép

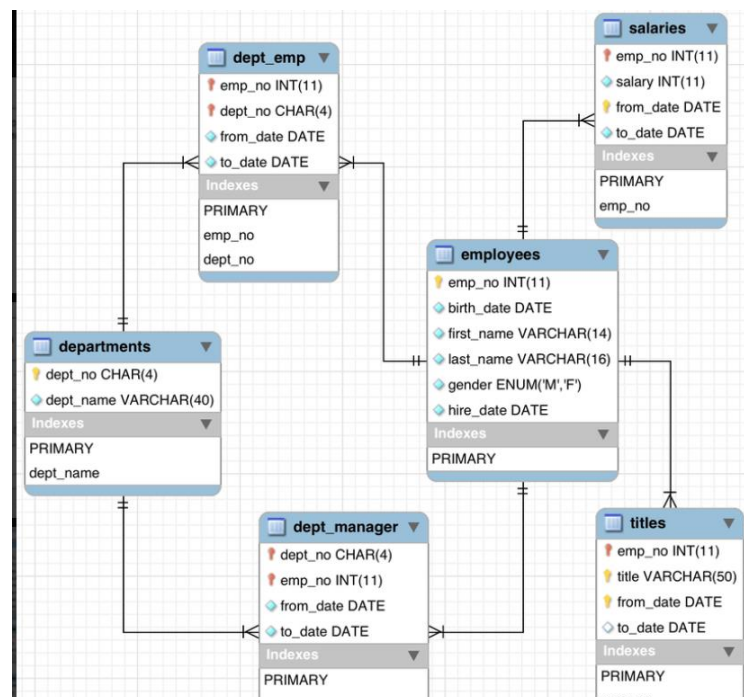
MySQL:

MySQL történelme:

A relációs adatmodellt Edgar Frank Codd dolgozta ki, az 1970-es évek elején létrehozták a SEQUEL (Structured English Query Language) nevű, relációs adatbázisok kezelésére, illetve az adatbázis-kezelő rendszer vezérlésére alkalmas nyelvet, melynek szavai értelmes, az adatkezelő műveletek jelentését kifejező angol szavakból állnak. A szavak felhasználásával pontosan definiált nyelvi szabályok alapján mondatok, az adatkezelő műveleteket precízen leíró utasítások voltak létrehozhatók.

A MySQL-t a svéd MySQL AB cég hozta létre 1995-ben. A platform fejlesztői Michael Widenius (Monty), David Axmark és Allan Larsson voltak. A MySQL célja az volt, hogy hatékony és megbízható adatkezelési lehetőségeket nyújtson otthoni és professzionális felhasználásra. 2000-re több mint féltucat verzió jelent meg amik kompatibilisek voltak az összes főbb platformmal, ekkor nyílt forráskódúvá vált, ezzel nyitottá téve harmadik fél fejlesztői hozzájárulását. Sokak szerint ennek köszönheti azt, hogy végül széles körben el tudott terjedni, a cégek, valamint fejlesztők körében.

2001-re több mint 2 millió aktív MySQL szerver üzemelt, 2002-ben megnyílt a cég amerikai központja, ezzel egy időben pedig 3 millióra bővült az aktív felhasználók száma.



4. Mysql adatbázis struktúra példa

Web alkalmazás (WEB application):

A web-es alkalmazás egy olyan speciális program, amelyet egy web szerver futtat. A felhasználó (user) egy másik számítógépen ül, és onnan aktiválja a program funkcióit egy browser-en keresztül. A program (nyelv) nem kezel billentyűzetet, sem egér eseményeket, mivel a program kezelője nem ugyanezen a gépen dolgozik. A feldolgozandó adatokat a web szerver kapja és továbbítja a program felé, ami a kód alapján válaszul adatokat állít elő, melyet egy HTML szabvány szerint felépített weblapba írja bele. A web szerver egy HTML lapot visszaküld vissza a kliens browserébe, aminek a feladata annyi, hogy az megjelenítse.

Yii2 keretrendszer

A Yii egy teljesítménycentrikus, komponens alapú keretrendszer, modern webalkalmazások gyors fejlesztéshez. Felhasználható a jelenleg ismert webalkalmazások fejlesztéséhez. Komponens alapú architektúrája és gyorsítótár-támogatása miatt különösen alkalmas nagyméretű alkalmazások fejlesztésére, mint például portálok, e-kereskedelmi projektek, restful webszolgáltatások, vagy az általam is választott tartalomkezelő rendszerekhez (CMS).

Yii verziók A Yii jelenleg két fő verzióval rendelkezik: 1.1 és 2.0. Az 1.1-es verzió a régi generáció, és most karbantartási módban van. A 2.0 verzió a Yii teljes átírása, amely a legújabb technológiákat és protokollokat alkalmazza, beleértve a Composert, a PSR-t, a névtereket, a tulajdonságokat és így tovább. A 2.0 verzió a keret jelenlegi generációját képviseli, és a következő néhány évben megkapja a fő fejlesztési erőfeszítéseket. Ez az útmutató elsősorban a 2.0 verzióról szól.

Fejlesztői segédeszközök, szoftverek

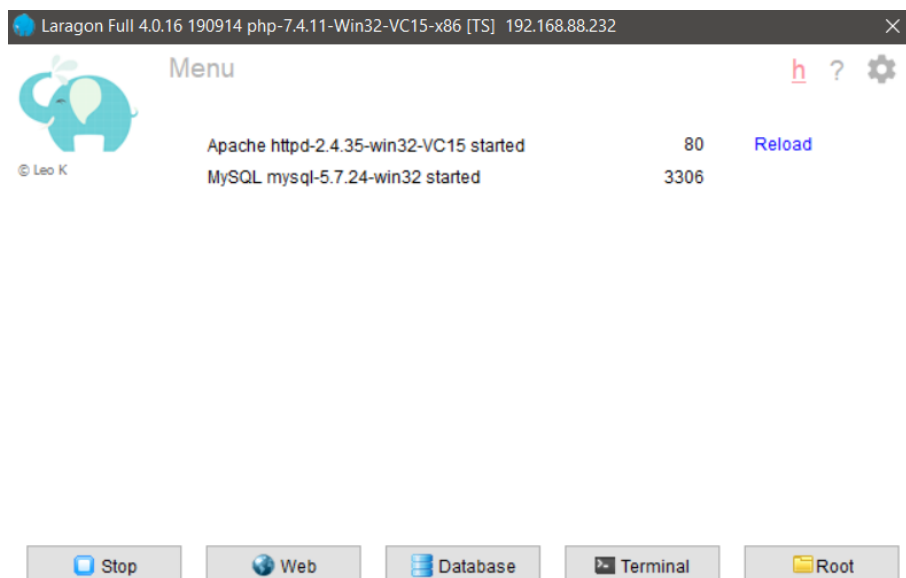
Laragon:

A Laragon egy hordozható, gyors és hatékony univerzális fejlesztői integráció. Kiválóan alkalmas modern webes alkalmazások kezelésére. Fejlesztése során a teljesítményre, a stabilitásra, az egyszerűsége és a rugalmasságra összpontosítottak. Nem használja a Windows szolgáltatásait, saját szolgáltatáskezelője van, amely aszinkron és blokkolatlanul kezeli a szolgáltatásokat, így azok gyorsan és zökkenőmentesen működnek.

A fejlesztői környezetébe integrálva megtaláljuk többek között a PHP-t, a Node.js-t, a Python-t, a Java-t, a Go-t és a Ruby-t, valamint a fejlesztők által leginkább használt programokat, mint a terminal, WinSCP, NotePad++, Putty valamint a netstat. Telepítésüket, használhatóságukat a könnyű kezelés és a gyors elérés miatt nagy mértékben megkönnyíti.

A teljesség igénye nélkül sorolok fel, néhány általam használt fejlesztői integrációt:

- **PHP:** ahol a verziók közötti gyors, és zökkenőmentes váltás könnyen elvégezhető, egyszerűen beállíthatóak a megfelelő PHP extension-ök, valamint könnyen elérhető az error log, a mailer valamint a xdebug beállításai.
- **Apache:** verziókezelés, error és access log, HTTPD konfigurációs fájl és beállításai
- **MySQL:** verziókezelés, integrált phpMyAdmin és HeidiSQL



PHPStorm:

A JetBrains, a világ egyik legelterjedtebb, integrált fejlesztői környezeteket (IDE) „gyártó” cége, az IDE-k kereskedelmi és közösségi kiadásban is megjelennek, valamint hallgatók (a világ bármely pontjáról) ingyenes verziót is igényelhetnek. Első verzió a JAVA-hoz készült 2001 januárjában, és akkoriban egyike volt azoknak az elsőként elérhető Java IDE-knek, amelyek továbbfejlesztett kód navigációval és kód refaktorálási képességekkel voltak felvértezve. Azóta rengeteg nyelvhez nyújtja az egyik legjobb forráskódszerkesztőt, automatizálási eszközöket és hibakeresőt. Ezeknek alapvető szerepe az alkalmazásfejlesztés lehető leggyorsabbá tétele.

A PHPStorm, első sorban PHP nyelv fejlesztését biztosító IDE. Alapvető funkciói közé tartozik a folyamatos hibamegelőzés, automatikus kiegészítés és kódjavítás. Kiterjesztett HTML, CSS és JavaScript szerkesztőt biztosít, valamint hatékony beépített eszközöket kínál az alkalmazások hibakereséséhez, teszteléséhez és profilozásához.

Lightshot:

Nagyon hasznos képernyőkép készítő, módosító eszköz. Rendkívül könnye használható, egyszerűen és gyorsan editálhatóak vele a képernyőképek. Egy kattintással megosztható az elkészített képernyőkép (<https://prnt.sc/> link), valamint „hasonló kép” keresőt is tartalmaz.

Az alkalmazás bemutatása

Óvodapedagógusokat segítő nyilvántartó rendszer

A következőkben a fejlesztett rendszert fogom bemutatni felhasználói, illetve fejlesztői oldalról.

Felhasználói szemszögéből első sorban a felületet, annak működését és kinézetét és mutatom be, képekkel illusztrálva a használatát. Az alkalmazás eléréhez csupán egy böngészőre van szükség.

Felhasználói oldal bemutatása:

Az alkalmazás főoldala egy üdvözlő szöveget tartalmaz. A felhasználót az URL cím beírása után egy egyszerű/letisztult index képernyő fogadja. A jobb felső sarokban található menüpontokon lehetőség van a bejelentkezésre, valamint a regisztrációra.

Bejelentkezés nélkül különösebb funkciók nem érhetőek el! Bejelentkezés után elérhetővé válik a bal oldali menüsáv, amin felett „Hamburger ikon” jelzi, a menüsáv elrejtését.

Regisztráció, Bejelentkezés, Jelszó visszaállítása:

Regisztráció esetén:

Felhasználónév, email cím és jelszó megadása kötelező, a megfelelő formai követelmények betartásával:

- Felhasználónév - 6-20 karakter hosszú egyedi azonosító
- Jelszó - legalább 6 karakter hosszú, néhány egyszerű jelszó nem használható, pl.: jelszó, 123456, abcabc
- Jelszó újra: egyeznie kell az előző jelszóval
- kiegészítő “Mutasd a jelszót funkcióval”

Sikeres regisztráció után, a rendszer automatikusan beléptet, alapértelmezett jogosultságokat megadnja, ezzel a munka megkezdődhet.

Bejelentkezés esetén:

Már létező felhasználó, felhasználónév és jelszó párossal be tud jelentkezni. A bejelentkezési form-on szintén “Mutasd a jelszót”, valamint emlékezz rám funkcióval. A bejelentkezési oldalon lehetőség van az elfelejtett jelszó visszaállítására is.

Bejelentkezés!

A bejelentkezéshez írja be a felhasználónevét és jelszavát!

Felhasználó

Password

☐ Mutasd a jelszót

☒ Emlékezz rám

Elfelejtett jelszó visszaállítása [itt!](#)

Bejelentkezés




Elfelejtett jelszó visszaállítása:

Az elfelejtett jelszó visszaállítását a bejelentkezés során, az “elfelejtett jelszó visszaállítása itt” szövegre kattintva tehetjük meg. Ehhez az regisztrált email címre lesz szükség, megerősítésre, az új jelszó generáláshoz, valamint az új jelszóra!

Felhasználói fiókok

Mindenki, aki használni akarja a rendszert regisztrációra, vagy ez már megtörtént, bejelentkezésre van szüksége. A felhasználói fiókokat, a bal oldali menüsávban találjuk, a „Felhasználók” menüpont alatt.

Lehetőségeink a következők (az adott felhasználó sorában lévő ikonok):

- Adatok vizsgálata –  - „szem” ikonra kattintva láthatjuk a felhasználóhoz tartozó adatokat
- Módosítás –  - „ceruza” ikonra kattintva lehetőségünk van a felhasználónév valamint az emailcím módosítására
- Törlés -  - „kuka” ikonra kattintva törölhető bármely felhasználó!

Óvónők, Szülők, Gyerekek

Ennek a három modulnak a felépítése nagy mértékben megegyezik. Mindhárom modulban lehetőségünk van létrehozni, megvizsgálni, módosítani, illetve törölni. A menüpontokra kattintva alapértelmezetten egy összesítő oldal jelenik meg, ahol az összes Óvónő, szülő vagy gyermek megtalálható.

Óvónők

Az óvónőkhöz tartozó minden adat itt található.

Szülők és Óvodások

A szülőkhöz és a gyermekekhez tartozó minden adat itt található, kiegészítve a szülő-gyermek kapcsolattal, amit az adott szülő/gyermek melletti vizsgálat (👁) ikonra kattintva érünk el. Amennyiben a szülőkhöz hozzá van rendelve gyermek, akkor automatikusan, azok nevével ellátott gombok jelennek meg, amivel azokat a lehető legegyszerűbben elérjük. A gombok a hozzájuk tartozó feliratok által könnyen beazonosíthatóak!

Gombok:

- Szülő adatainak elérése - Szülők nevével ellátott, zöld háttérű gombok
- Szülő – óvoda összerendelése – „Összerendelés” feliratú, sárga háttérű gomb
- Módosítás/ törlés - „Módosítás” és „Törlés” feliratú kék, illetve piros háttérű gomb

The screenshot shows the Ovi App interface. On the left is a dark sidebar with navigation options: 'Felhasználók', 'Óvónők', 'Szülők & gyerekek' (selected), and 'Eszközök'. The main area displays the profile of 'Horváth Pista'. At the top of the profile are five colored buttons: 'Horváth Szabolcs' (green), 'Nagy Katinka' (green), 'Összerendelés' (yellow), 'Módosítás' (blue), and 'Törlés' (red). Below these is a table with the following data:

Név	Horváth Pista
Születési idő	2021-04-06
Csoport	Kiscsoport
Caregivers	Horváth Szabolcs, Nagy Katinka

Fejlesztői oldal bemutatása:

A webserver (lokális szerver), az URL beírásakor, automatikusan a SiteController Index fájlját futtatja.

Regisztráció, Bejelentkezés, Jelszó visszaállítása:

A felhasználót az URL cím beírása után a siteController és annak funkciói irányítják. Az adott folyamatokhoz 5 modellt hoztam létre, mind más-más attribútum szükséges ellenőrzését végzi.

Regisztráció és bejelentkezés során szinte azonos a háttérműködés. Bejelentkezéshez a LoginForm, míg regisztráláshoz a SignupForm” modell teszi lehetővé a szükséges felhasználói adatok validálását. Ezek az adatok a következők:

- felhasználónév,
- emailcím (csak regisztrálásnál és jelszó visszaállításnál),
- jelszó

Az ehhez tartozó rule-ok a következő képen láthatóak.

```

/**
 *
 */
public function rules()
{
    return [
        ['username', 'filter', 'filter' => 'trim'],
        ['username', 'required'],
        ['username', 'unique', 'targetClass' => User::class, 'message' => 'Ez a felhasználónév már foglalt!'],
        ['username', 'string', 'min' => 6, 'max' => 20],

        ['email', 'filter', 'filter' => 'trim'],
        ['email', 'required'],
        ['email', 'email'],
        ['email', 'unique', 'targetClass' => User::class, 'message' => 'Ez az emailcím már foglalt!'],

        [['password', 'password_repeat'], 'required'],
        [['password', 'password_repeat'], 'string', 'min' => 6],
        [['password'], 'in', 'range' => PasswordEnum::getLabels(), 'not' => true, 'message' => 'Nem megfelelő erősségű jelszó!'],
        ['password_repeat', 'compare', 'compareAttribute' => 'password', 'message' => 'A jelszavaknak egyezniük kell!'],
    ];
}

```

1. ábra SignupForm rules

Felhasználói fiókok

Az előző pontban bemutatott validálások után, az a felhasználó, akinek „access token”-je valid, a bal oldali menüt eléri.

● Irodalomjegyzék

“HTML History.” 2012. W3schools.in. 2012.

<https://www.w3schools.in/html-tutorial/history/>

“JavaScript History.” 2015. W3schools.com. 2015.

https://www.w3schools.com/js/js_history.asp

“Webprogramozás I.|Digitális Tankönyvtár.” 2014. Tankonyvtar.hu. 2014.

https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0021_40_webprogramozas_i/32_a_php_trtnete.html

“Debreceni Egyetem Informatikai Kar Web Alapú Alkalmazás Fejlesztése Debrecen 2009.” n.d. Accessed March 13, 2021.

<https://dea.lib.unideb.hu/dea/bitstream/handle/2437/90621/Webes%20alap%FA%20a%20alkalmaz%El%fejleszt%E9s.pdf?sequence=1>

“Mi Az a Web Alkalmazás | Informatika 9-12. Évfolyam | Sulinet Tudásbázis.” 2016. Sulinet.hu. 2016.

<https://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-9-12-evfolyam/programozas-c-kornyezetben/webes-alkalmazasok-keszitese>

“CSS Introduction.” 2021. W3schools.com. 2021.

https://www.w3schools.com/css/css_intro.asp

Saunders, Stephen; Fields, Duane K. & Belayev, Eugene (March 1, 2006), IntelliJ IDEA in Action (1st ed.), Manning, p. 450, ISBN 1-932394-44-3

Davydov, S. & Efimov, A. (May 2005), IntelliJ IDEA. Professional'noe programmirovaniye na Java (V podlinnike) (1st ed.), BHV, p. 800, ISBN 5-94157-607-2,

<<http://www.bhv.ru/books/book.php?id=9910>> Archiválva 2013. december 9-i dátummal a Wayback Machine-ben

“About PhpStorm.” 2017. Componentsource.com. 2017.

<https://www.componentsource.com/product/phpstorm/about>

“Bevezetés a CSS Alapjaiba.” March 19, 2005. Prog.Hu. 2021.

<https://prog.hu/cikkek/907/Bevezetes+a+CSS+alapjaiba.html>

“HTML 5 Alapismerek Bevezetés a HTML 5-Be Mi Is Az a HTML?” n.d. Accessed March 21, 2021. <http://webdesignsuli.hu/wp-content/uploads/2015/11/3.lecke-mindent-a-html-r%C5%91.pdf>

“Lightshot — Screenshot Tool for Mac & Win.” 2021. Prntscr.com. 2021.

<https://app.prntscr.com/en/index.html>

“Webadatbázis-Programozás.” 2021. Elte.hu. 2021.

http://ade.web.elte.hu/wabp/lecke7_lap1.html

Brotherton, Claire. 2020. “The Most Popular PHP Frameworks to Use in [Year].” Kinsta.

Kinsta. September 28, 2020. <https://kinsta.com/blog/php-frameworks>

“Design Patterns and Refactoring.” 2021. Sourcemaking.com. 2021.

https://sourcemaking.com/design_patterns

“How to Secure PHP Web Applications and Prevent Attacks?” 2018. PHP.earth. 2018.

<https://docs.php.earth/security/intro>

“MySQL.” n.d. Accessed April 4, 2021.

https://pub.cateye.hu/uni/adatb1/2/01_Bevazeto.pdf

“History of MySQL.” 2011. Datasciencecentral.com. 2011.

<https://www.datasciencecentral.com/profiles/blogs/history-of-mysql>

“Adatbázis Fejlesztés És Üzemeltetés I.|Digitális Tankönyvtár.” 2014. Tankonyvtar.hu.

2014. <https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011->

[0021_30_adatbazis_fejlesztés_es_üzemeltetés_i/731_az_sql_nyelv_trtete.html](https://regi.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0021_30_adatbazis_fejlesztés_es_üzemeltetés_i/731_az_sql_nyelv_trtete.html)

Mellékletek

[A dolgozat mellékletei, ha vannak]