

ide jön majd a cím

# Látogatói Kártya

Készítette és tervezte: Csikós Norbert

Konzulens tanár: Bende Attila

CSZC Mihály Dénes Szakgimnázium és Szakközépiskola OKJ: 54 213 05 – Szoftver fejlesztő  
esti

2020.04.01.

## Tartalomjegyzék

<b>Bevezetés.....</b>	<b>3</b>
<b>Fejlesztői dokumentáció.....</b>	<b>4</b>
<i>Alkalmazás indítása.....</i>	<i>4</i>
<i>Alkalmazott technológiák.....</i>	<i>4</i>
Fejlesztői környezet .....	4
Kliens oldalon használt technológiák .....	5
<i>Szerver oldalon használt technológiák .....</i>	<i>7</i>
<i>Adabázisleírása.....</i>	<i>8</i>
<i>Programfunkciók ismertetése .....</i>	<i>8</i>
<b>Felhasználói dokumentáció.....</b>	<b>8</b>
<b>Testelés.....</b>	<b>9</b>
<b>Továbbfejlesztési lehetőségek .....</b>	<b>9</b>
<b>Összegzés .....</b>	<b>9</b>
<b>Felhasznált irodalom .....</b>	<b>9</b>

## Bevezetés

A Látogatói Kártya program ötlete Bende Attila tanár úrtól származik. A tanár úr, a Verebély László Szakgimnáziuma és Szakközépiskolában is tanít, ahol rendvédelmi oktatás is folyik. Ez az intézmény is részt vesz az áprilisban tartandó budapesti rendészeti napokon, nem csak az iskola területén hanem külső helyszíneken is. A látogatói kártya a külső rendezvényhelyszínek való belépésre jogosít fel.

A látogatói kártya egy fényképpel ellátott, személyek azonosítására alkalmas, bankkártya méretű, papírra nyomtatott kártya. A fényképen kívül a kártya tartalmazza a tulajdonosa teljes nevét, feltüntethető rajta az intézmény nevét, ahonnan érkezett és akár a rendfokozatát is.

Végig gondolva azt, hogy a programot mire fogják majd felhasználni és feltételezéseim szerint milyen különböző helyzetek adódhatnak a felhasználás során, sok olyan kérdés merült fel bennem és arra jutottam, hogy egy egyszerű webalkalmazásban oldom meg a feladatot. A választás mellett szolt, hogy:

- Ha az iskola egyszerre több látogatót fogad egy időben és egyszerre több kártyát szeretne készíteni, akkor egyszerre több eszközre van szüksége. Ezt a problémát egyszerűbb megoldani (eszközparktól függően), hanem kell azon gondolkodni mennyi számítógépet telepítsük az alkalmazást. Tehát a telepítés webalkalmazás esetén egyszerűbb mert csak egyszer kell feltenni a szerverre, szemben egy java vagy a C# asztali alkalmazással minden olyan számítógépre kell feltelepíteni amelyiken használni szeretnénk.
- Ez a program nem mindennap lesz használva, így olyan helyen tároljuk amit nem telepítünk újra akár félévente vagy évente mint egy használatban lévő asztali számítógépet vagy laptopot. Ebből a szempontból is inkább webalkalmazásban gondolkodtam.
- Úgy gondoltam hogy szempont lehet az is hogy a program elérhető legyen akár mobil eszközről is. Egy olyan alkalmazásban gondolkodtam ami elérhető webböngészőből és képes használni a mobil eszközök webkameráját.

## **Fejlesztői dokumentáció**

### **Alkalmazás indítása**

A XAMPP alkalmazás htdocs mappájába csomagolja ki a norbertcsikos.zip file-t. A norbertcsikos/db/latogatoi.sql file-t importálja a phpMyAdmin adatbázis szerverén. A file tartalmazza az adatbázis elkészítését tartalmazó scriptet is. A böngésző címsorába az Apache szerver elérési útvonala után írja be: /norbertcsikos/ (pl.: <http://localhost/norbertcsikos/>)

### **Alkalmazott technológiák**

#### **Fejlesztői környezet**

operációs rendszer: macOS Mojave

processzor: 2,4 GHz Intel Core i5

memória: 16 GB

Visual Studio Code: 1.44.0

Szerver:

XAMPP 7.1.29 Rev. 0

Adatbázis-kiszolgáló

Kiszolgáló típusa: MariaDB

Kiszolgáló verziója: 10.1.39-MariaDB

Protokoll verzió: 10

Webkiszolgáló:

Apache/2.4.39(Unix)

PHP verzió: 7.1.29

phpMyAdmin verziószám: 4.8.5

Kliens:

Node verzió:13.10.1

npm verzió: 6.13.7

angular cli: 9.1.0

angular cdk: 9.2.0

angular-devkit/build-angular: 0.901.0

angular material: 9.2.0

nrwl workspace": "9.2.1

rxjs: 6.5.4

TypeScript

## **Kliens oldalon használt technológiák**

Már több olyan framework-ről hallottam, olvastam amelyek olyan kódok építésére készültek amik egyszerre felhasználhatóak asztali alkalmazásokba, weboldalak készítéséhez illetve akár nativ mobilalkalmazásokhoz is. Ilyen lehetőséget biztosít nekünk a teljesség igénye nélkül, az Angular, a Flutter, Xojó. Választásom az Angular framework-re esett

„Az Angular egy Google által fejlesztett, nyílt forráskódú TypeScript keretrendszer dinamikus webes alkalmazásokhoz. Segítségével nagyban egyszerűsödik a webes alkalmazások frontend fejlesztése. Használatával a HTML eszköztára kibővül és az alkalmazások komponensei még egyértelműen elkülönülnek. Az Angular adatkapcsolásának, illetve függőség injektálásának köszönhetően, rengeteg felesleges boilerplate kód elhagyható.”

<http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=27>

az Angular jól dokumentált, sokan használják, ezért nagyon sok problémára kerestek már megoldást és ezeket a problémákat meg is oldották. Folyamatosan fejlesztik. Nativ megoldásokat is kínál és ez ebben a programban előnyt is jelent a kamera használatnál. Több előre megírt desing gyűjteményből lehet választani mint a BootStrap, Angular Material.

Angular Material desing komponens gyűjteményből választottam a program elemeinek megjelenítéséhez.

Az Angular keretrendszer alapja a NodeJs.

A Node.js (vagy más néven Node) egy rendszer, melyben JavaScriptben írhatunk szerver oldali alkalmazásokat. Maga a rendszer C/C++-ban íródott, és egy esemény alapú I/O rendszert takar a Google V8 JavaScript motorja felett. A Node célja, hogy egy egyszerű felületet adjon skálázható hálózati alkalmazások írására. Azt tudjuk, hogy a JavaScript a világon a jelenleg legnépszerűbb programozási nyelv, többek között ezért esett erre a nyelvre a választás. A JavaScript nyelvnek vannak más sajátosságai, amelyek szintén előtérbe kerülnek a rendszer használata során. Ahogy a böngészőben, úgy Node alatt is egyetlen esemény hurokban (event loop) fut a program, és ez adja az egyik legfontosabb előnyt.

A Node.js-t úgy írták meg, hogy (szinte) minden esemény aszinkron legyen, ezért a program sosem blokkolódik, azaz nem kell várni, hogy egy művelet befejeződjön, vele párhuzamosan futtathatunk további műveleteket. Ez az alkalmazás folyamatosabb futását, több alkalmazás párhuzamosítását, valamint az egész rendszer optimálisabb működését teszi lehetővé.

[http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=16#section\\_4](http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=16#section_4)

„A Node hivatalos csomagkezelője az npm, amellyel kényelmesen listázhatjuk, telepíthetjük, frissíthetjük és eltávolíthatjuk az elérhető modulokat, valamint mi magunk is publikálhatjuk az általunk fejlesztett modulokat az npm rendszerébe.”

[http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=16#section\\_4](http://nyelvek.inf.elte.hu/leirasok/JavaScript/index.php?chapter=16#section_4)

Az Angular-hoz szorosan hozzá tartozik a TypeScript nyelv. A program kliens oldali kódokjai készültek TypeScript-ben.

„A TypeScript egy objektum-orientált script nyelv, amit a Microsoft készített. Legfőbb filozófiája nyelvnek az, hogy "legyen a TypeScript bővebb halmaza a Javascript-nek". A Javascript-től eltérően, legnagyobb újdonsága, hogy statikusan típusos a nyelv.

Cél nem egy újabb Javascript-gyilkos nyelv megalkotása volt, hanem egy olyan eszköz elkészítése, mely segíti a fejlesztőket az igazán nagy projektek elkészítésében is. A projekt fontosságát mutatja az is, hogy a Windows 8 fejlesztésénél a WinRT egy részét is ezen a nyelven írták.

A nyelv teljesen nyílt forráskódú, illetve operációsrendszer független. Mivel a fordító a TypeScript forráskódból Javascript kódot generál, így böngészőfüggetlen is a nyelv. Nem szükséges semmiféle külső program vagy plugin telepítése sem.

Nagy előnye, hogy felülről kompatibilis a Javascript nyelvvel, így a már meglévő Javascript-es projektjeinket (pl.: jQuery, Raphael, stb.) is változtatások nélkül fel tudjuk használni a TypeScript-ben. Ezenfelül, hogy a statikusan típusos nyelvi támogatást is megkaphassuk a Javascript-es projekteknel lehetőség van definíciós fájlok segítségével TypeScript kompatibilissé tenni őket.

Támogatott fejlesztőeszközök a Visual Studio Code, Visual Studio 2012, Emacs, Vim és Sublime Text, de ez a jövőben még bővíthet.”

<http://nyelvek.inf.elte.hu/leirasok/TypeScript/index.php?chapter=3>

## **Szerver oldalon használt technológiák**

Az alkalmazás szerver környezetének az oktatás során is használt XAMPP keretrendszert hívtam segítségül.

„A XAMPP egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webszerver, a MariaDB (korábban a MySQL[2]) adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei). Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.”

<https://hu.wikipedia.org/wiki/XAMPP>

A szerveroldali kódok PHP-ban lettek implementálva.

„A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakot hozva létre. Rasmus Lerdorf 1995-ben indította útjára. Ma a The PHP Group tartja fenn és fejleszti. A PHP szabad szoftver, de licence nem csereszabatos a GNU licenccel, mivel megkötéseket tartalmaz a PHP név használatára.”

„A hagyományos HTML lapokkal ellentétben a kiszolgáló a PHP-kódot nem küldi el az ügyfélnek, hanem a kiszolgáló oldalán a PHP-értelmező motor dolgozza fel azt. A programokban lévő HTML elemek érintetlenül maradnak, de a PHP kódok lefutnak. A kódok végezhetnek adatbázis-lekérdezéseket, létrehozhatnak képeket, fájlokat olvashatnak és írhatnak, kapcsolatot létesíthetnek távoli kiszolgálókkal. A PHP-kódok kimenete a megadott HTML elemekkel együtt kerül az ügyfélhez.”

Az adatbázist, a XAMPP által biztosított MySQL szerverrel valósítottam meg.

A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver.

## **Adabázisleírása**

### **Programfunkciók ismertetése**

## **Felhasználói dokumentáció**

A Látogatói Kártya program a Verebély László Szakgimnáziuma és Szakközépiskola áprilisban tartandó budapesti rendészeti napok külső rendezvényhelyszínek való belépésre jogosító személyek azonosítására alkalmas fényképpel ellátott

ahol rendvédelmi oktatás is folyik. Ez az intézmény is részt vesz az áprilisban tartandó budapesti rendészeti napokon, nem csak az iskola területén hanem külső helyszíneken is. A látogatói kártya a külső rendezvényhelyszínek való belépésre jogosít fel.

A látogatói kártya egy fényképpel ellátott, személyek azonosítására alkalmas, bankkártya méretű, papírra nyomtatott kártya. A fényképen kívül a kártya tartalmazza a tulajdonosa teljes nevét. feltüntethető rajta az intézmény nevét, ahonnan érkezett és akár a rendfokozatát is.

A kártya elkészítése során felhasználónak lehetősége van megadni a kártyán megjelenítendő keresztnévet, vezetéknévet. Kiválaszthatja egy legördülő menüből a látogató nevének eltagját(pl: dr if, id), a látogatók csoportját(pl: tanulo, szervezo, vip ), a látogató rangját, rendfokozatát illetve, fényképet adhat a kártyához. Felhasználónak lehetősége van a meglévő



titulus(név előtagja),- latogatoi csoport,- rang listájához új elemet hozzáadni, a lista elemeit törölni, a latogatok csoporthoz rendelt, a fénykép körül található színes keret színét színpalettából kiválasztani.

A fényképet, webkamera segítségével készítheti el vagy saját eszközéről importálhatja és addhatja hozzá a kártyához. A program automatikusan átméretezi a képet(Kép szélesség: W: 30mm, Kép magasság: H: 32mm. A képek szerkesztésére a programban nincs lehetőség. A kártyákat a felhasználó mentheti és nyomtathatja.

## **Testelés**

## **Továbbfejlesztési lehetőségek**

## **Összegzés**

## **Felhasznált irodalom**