



Jelenlét követő alkalmazás fejlesztése

Készítette

Györkis Tamás

Programtervező informatikus BSc

Témavezető

Dr. Király Roland

Egyetemi docens

EGER, 2024

Tartalomjegyzék

Bevezetés	4
1. Alkalmazás bemutatása	5
1.1. Adatbázis	5
1.2. Általános funkciók, információk	6
1.3. Adminisztrátori nézet, funkciók	7
1.4. Tanári nézet, funkciók	7
1.5. Hallgatói nézet, funkciók	7
2. Felhasznált technológiák, csomagok	8
2.1. Laravel	9
2.1.1. Keretrendszer alapjai	9
2.1.2. MVC architektúra	9
2.1.3. Eloquent ORM	9
2.1.4. Blade templating engine	9
2.1.5. Laravel Fortify	9
2.1.6. Események, queue	9
2.1.7. Email küldés	9
2.2. Tailwind	9
2.2.1. DaisyUI	9
2.3. LiveWire	9
2.3.1. Ismertető, a csomag működése	9
2.3.2. LiveWire és SPA	9
2.3.3. Komponensek és data binding	9
2.3.4. Események, lapozás	9
2.3.5. Alpine.js: LiveWire és JavaScript kapcsolata	9
2.3.6. WireToast: LiveWire alapú értesítések	9
2.4. Websocket és Pusher szerviz	9
2.4.1. Websocket jelentősége	9
2.4.2. Pusher	9
2.4.3. Websocket integrálása a keretrendszerbe	9
2.4.4. Csatornák, események létrehozása	9

2.4.5.	Események fogadása JavaScript, LiveWire esetén	9
2.5.	QR-kód generálás	9
2.6.	Naptár - FullCalendar.js	9
2.6.1.	Beállításai	9
2.6.2.	Kapcsolat a keretrendszerrel LiveWire segítségével	9
2.7.	Diagramok - Chart.js	9
2.7.1.	Beállításai	9
2.7.2.	Kapcsolat a keretrendszerrel LiveWire segítségével	9
2.8.	Progressive Web Apps	9
2.8.1.	A PWA jelentése, jelentősége	9
2.8.2.	Laravel PWA csomag	9
2.8.3.	A manifest fájl	9
2.8.4.	Service Worker	9
3.	Az alkalmazás tesztelése	10
3.1.	Manuális tesztelés	10
3.2.	Automatizált tesztelés	10
3.3.	Terheléses tesztelés	10
3.4.	Laravel Pint és Github actions	10
4.	Alkalmazás telepítése	11
4.1.	Laravel Forge	11
4.2.	Kézi telepítés lépései	11
	Összegzés	12
	Irodalomjegyzék	13

Bevezetés

Amikor szakdolgozati témaválasztás előtt álltam, sokat gondolkoztam a témán. Mindenképpen egy olyan alkalmazást szerettem volna elkészíteni, mely ténylegesen hasznos is lehet. Egyetemi éveim alatt demonstrátorként tanítottam több féléven át az egyetemen, és innen jött a felismerés, hogy jó lenne, ha a hallgatók jelenlétét, hiányzásait ne táblázatokban kelljen vezetni, hanem egy külön eszköz legyen rá készítve. Innen származik az alkalmazás ötlete.

A megvalósítás során törekedtem arra, hogy egy jól átlátható, ergonomikus weboldalt készítsek, amit – esetleges kisebb módosításokkal – ne csak egyetemi környezetben lehessen használni. Ezekből kifolyólag egy webalkalmazást készítettem, mivel ezt a leg-egyszerűbb elérni, hiszen csak egy webböngésző szükséges hozzá. A tanulók és tanárok által használt oldalak reszponzívan lettek elkészítve, ezáltal biztosítva, hogy mobiltelefonon is használni lehessen. Illetve az alkalmazás PWA funkciókkal rendelkezik, aminek jelentőségére később térek ki, de előjáróban annyit érdemes megemlíteni róla, hogy lehetővé teszi a weboldal alkalmazásként való telepítését számítógép és telefon esetén is, ezáltal egy rendes alkalmazás érzését keltve.

A megvalósításhoz a Laravel keretrendszert használtam, mely egy PHP alapú, MVC keretrendszer, amivel tanulmányaim során találkoztam, és egyből megkedveltem. Nem titkolt célom a szakdolgozatommal, hogy bemutassam, hogy a mai, JavaScript preferált világban, továbbra is lehetséges modern, a mai kort kielégítő weboldalt készíteni PHP segítségével, amihez különböző csomagokat alkalmaztam, amiknek működését, illetve egymással való működését a későbbiekben fogom kifejteni.

1. fejezet

Alkalmazás bemutatása

1.1. Adatbázis

Az 1.1 ábrán látható az alkalmazás által használt adatbázis, egyed-kapcsolat diagrammal ábrázolva, melyről szeretnék néhány szót ejteni.



1.1. ábra. Az alkalmazás által használt adatbázis.

- *users*: ez a tábla tárolja a felhasználók adatait. Minden felhasználóról tárolunk egy azonosítót, egy Neptun kódot, nevet, email címet, egy titkosított jelszót,

egy profilképet, a felhasználó által használt nyelvet, illetve egy UUID-t, azaz egy egyedi azonosítót, amit az órarend exportálásához lehet használni, erről később. Az azonosító oszlopra azért van szükség, mivel az alkalmazás fel van készítve arra, hogy Neptun kód nélkül rendelkező felhasználókat is tudjon kezelni.

- *terms, places*: a félévek és termek (helyek) tárolására szolgáló táblák.
- *subjects, courses, course_classe, class_login_links*: ezen táblák tárolják a tantárgyakat (amik többek között rendelkeznek egy azonosítóval, névvel, leírással, kreditértékkel, és egy tantárgyfelelőssel), a tantárgyakhoz tartozó kurzusokat (amihhez tárolok egy azonosítót, ami minden esetben egyedi, egy kurzus azonosítót, ami minden félév és tantárgy esetében kell egyedinek lennie, tartozik hozzá egy félév, illetve egy létszám limit), illetve kurzusokhoz pedig órák (amiknek van egy kezdete, vége, illetve egy terem, ahol tartják). Az utolsó tábla pedig az órákhoz tartozó bejelentkező linkeket tartalmazza, ennek jelentőségéről kicsit később.
- *attendances*: itt tárolom az egyes kurzusok résztvevőinek az órai jelenléteit, melyek lehetnek: nincs kitöltve, jelen, késés (mely esetben percben lehet tárolni a mértékét), hiányzás, igazolt hiányzás.
- *justification, justification_acceptances, justification_pictures*: ezek a táblák tárolják az igazolásokat, az igazoláshoz tartozó feltöltött képeket, illetve az igazolásban érintett tanárok válaszát, hogy elfogadják-e ó, vagy sem.
- Az egyéb, nem említett táblák inkább kapcsolótábla funkciót töltenek be, vagy a keretrendszernek vannak rá szükségei. Például a *jobs* tábla a háttérben végrehajtandó feladatokat (job) tartalmazza.

Itt érdemes még megemlíteni, hogy a fejlesztés során a MySQL nevezetű, relációs adatbázist használtam, viszont a Laravel keretrendszerből adódóan sok más típusú adatbázis szoftverrel használható az alkalmazás, mivel a táblák felépítése a keretrendszerben van elkészítve.

1.2. Általános funkciók, információk

Az alkalmazás elérhető a <https://szakdolgozat.ddns.net> címen, ahol ki lehet próbálni az alkalmazást. Pár alapértelmezett felhasználó elérhető:

- Szuper adminisztrátor:
 - Felhasználónév: *SADMIN*
 - Jelszó: *superadmin*

- Adminisztrátor:
 - Felhasználónév: *ADMIN0*
 - Jelszó: *admin*
- Tanár:
 - Felhasználónév: *TEACHE*
 - Jelszó: *teacher*
- Hallgató:
 - Felhasználónév: *STUDEN*
 - Jelszó: *student*

1.3. Adminisztrátori nézet, funkciók

1.4. Tanári nézet, funkciók

1.5. Hallgatói nézet, funkciók

2. fejezet

Felhasznált technológiák, csomagok

2.1. Laravel

2.1.1. Keretrendszer alapjai

2.1.2. MVC architektúra

2.1.3. Eloquent ORM

2.1.4. Blade templating engine

2.1.5. Laravel Fortify

2.1.6. Események, queue

2.1.7. Email küldés

2.2. Tailwind

2.2.1. DaisyUI

2.3. LiveWire

2.3.1. Ismertető, a csomag működése

2.3.2. LiveWire és SPA

2.3.3. Komponensek és data binding

2.3.4. Események, lapozás

2.3.5. Alpine.js: LiveWire és JavaScript kapcsolata

2.3.6. WireToast: LiveWire alapú értesítések

2.4. Websocket és Pusher szerviz

2.4.1. Websocket jelentősége

2.4.2. Pusher

2.4.3. Websocket integrálása a keretrendszerbe

2.4.4. Csatornák, események létrehozása

2.4.5. Események fogadása JavaScript LiveWire esetén

3. fejezet

Az alkalmazás tesztelése

3.1. Manuális tesztelés

3.2. Automatizált tesztelés

3.3. Terheléses tesztelés

3.4. Laravel Pint és Github actions

4. fejezet

Alkalmazás telepítése

4.1. Laravel Forge

4.2. Kézi telepítés lépései

Összegzés

Irodalomjegyzék

- [1] FAZEKAS ISTVÁN: *Valószínűességszámítás*, Debreceni Egyetem, Debrecen, 2004.
- [2] TÓMÁCS TIBOR: *A valószínűességszámítás alapjai*, Líceum Kiadó, Eger, 2005.