Pannon Egyetem

Műszaki Informatikai Kar

Rendszer- és Számítástudományi Tanszék

Üzemmérnök-informatikus BSc

**SZAK-/DIPLOMADOLGOZAT**

**Logopédia időpont foglaló webalkalmazás**

**Csáktornyai Ádám József**

Témavezető: Baumgartner János

2023



Hallgatói nyilatkozat

Alulírott Csáktornyai Ádám József hallgató kijelentem, hogy a dolgozatot a Pannon Egyetem Rendszer- és Számítástechnikai Tanszékén készítettem az Üzemmérnök informatikus BSc végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozatban lévő érdemi rész saját munkám eredménye, az érdemi részen kívül csak a hivatkozott forrásokat (szakirodalom, eszközök stb.) használtam fel.

Tudomásul veszem, hogy a dolgozatban foglalt eredményeket a Pannon Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Dátum: Veszprém, [év hónap nap]

*<<hallgató neve>>*

Témavezetői nyilatkozat

Alulírott <<témavezető neve>> témavezető kijelentem, hogy a dolgozatot *<<*hallgató neve*>>* a Pannon Egyetem <<tanszék neve>>én készítette <<végzettség>> végzettség megszerzése érdekében.

Kijelentem, hogy a dolgozat védésre bocsátását engedélyezem.

Dátum: Veszprém, [év hónap nap]

*<<témavezető neve>>*

Köszönetnyilvánítás

A hallgató köszönetet nyilvánít mindazoknak, akiktől (elméleti, gyakorlati, erkölcsi stb.) segítséget kapott.

Tartalmi összefoglaló

Tartalmi összefoglaló magyarul. Az összefoglalónak tartalmaznia kell (rövid, velős és összefüggő megfogalmazásban) a következőket:

* téma megnevezése,
* megoldott feladat megfogalmazása,
* megoldási mód,
* elért eredmények,
* kulcsszavak (4-6 darab)
* terjedelme nem lehet több 1 A4-es oldalnál.

Az összefoglalót magyar és angol nyelven kell készíteni. Sorrendben a dolgozat nyelvével megegyező kerül előrébb. A cím Title stílusú, formázása: Times New Roman, nagybetű, 14 pt, félkövér, középre igazított; az összefoglaló Normál stílusú, formázása: Times New Roman, 12 pt, sorkizárt, 1.5-ös sortávolság.

**Kulcsszavak:** [4-6 kulcsszó felsorolása, vesszővel elválasztva]

Abstract

Tartalomjegyzék

[Jelölésjegyzék 9](#_Toc149748196)

[1. Irodalmi áttekintés 10](#_Toc149748197)

[1.1. Logopédiai folyamatok 10](#_Toc149748198)

[1.2. IT eszközök és operációs rendszerek fejlődése 10](#_Toc149748199)

[1.3. Mobil alkalmazások fejlődése 10](#_Toc149748200)

[1.4. PWA technológia 10](#_Toc149748201)

[1.5. Mobil operációs rendszerek és sajátosságaik 11](#_Toc149748202)

[1.6. Logopédia és gyógypedagógia területén használt alkalmazások felmérése 12](#_Toc149748203)

[2. Feladat és követelmények 13](#_Toc149748204)

[2.1. Feladat leírása 13](#_Toc149748205)

[2.2. Funkcionális követelmények 13](#_Toc149748206)

[3. Felhasznált technológiák 13](#_Toc149748207)

[3.1. Verziókezelő rendszer 13](#_Toc149748208)

[3.1.1. Git 13](#_Toc149748209)

[3.1.2. GitHub 13](#_Toc149748210)

[3.2. Visual Studio Code 13](#_Toc149748211)

[3.3. Futtatókörnyezet és csomagkezelés 13](#_Toc149748212)

[3.3.1. Node.js 13](#_Toc149748213)

[3.3.2. npm 13](#_Toc149748214)

[3.3.3. Bun 13](#_Toc149748215)

[3.4. Programnyelvek 13](#_Toc149748216)

[3.4.1. JavaScript 13](#_Toc149748217)

[3.4.2. TypeScript 13](#_Toc149748218)

[3.5. React és Redux 14](#_Toc149748219)

[3.6. Firebase 14](#_Toc149748220)

[3.6.1. Authentication 14](#_Toc149748221)

[3.6.2. Hosting 14](#_Toc149748222)

[3.6.3. Firestore Database 14](#_Toc149748223)

[3.6.4. Realtime Database 14](#_Toc149748224)

[3.6.5. Storage 14](#_Toc149748225)

[3.7. PWA 14](#_Toc149748226)

[3.8. Google Chrome, mint fejlesztői eszköz 14](#_Toc149748227)

[3.8.1. Lighthouse 14](#_Toc149748228)

[3.8.2. React fejleszői bővítmények 14](#_Toc149748229)

[3.9. PWA kompatibilis operációs rendszerek 14](#_Toc149748230)

[3.9.1. Android 14](#_Toc149748231)

[3.9.2. iOS 14](#_Toc149748232)

[3.9.3. Windows 14](#_Toc149748233)

[4. Fejlesztői dokumentáció 15](#_Toc149748234)

[4.1. Fejlesztői környezet beállítása és projekt konfigurálása 15](#_Toc149748235)

[4.1.1. Fejlesztői eszközök telepítése 15](#_Toc149748236)

[4.1.2. Projekt beállítása és build-elés 15](#_Toc149748237)

[4.2. Adatreprezentáció 15](#_Toc149748238)

[4.3. Időpont foglalás működése 15](#_Toc149748239)

[4.4. Tárgy kölcsönzés működése 15](#_Toc149748240)

[4.5. Tesztelés 15](#_Toc149748241)

[5. Felhasználói dokumentáció 15](#_Toc149748242)

[5.1. Felhasználói felület 15](#_Toc149748243)

[5.2. Webalkalmazás elérése és telepítés 15](#_Toc149748244)

[5.3. Főbb funkciók menete 15](#_Toc149748245)

[5.3.1. Időpont foglalás 15](#_Toc149748246)

[5.3.2. Tárgy kölcsönzés 15](#_Toc149748247)

[Irodalomjegyzék 16](#_Toc149748248)

[Mellékletek 17](#_Toc149748249)

Jelölésjegyzék

|  |  |
| --- | --- |
| AI: | Artificial Intelligence (Mesterséges Intelligencia) |
| GPU: | Graphical Processing Unit (Grafikus Processzor / Grafikus Feldolgozó Egység) |
| API: | Application Programming Interface (Alkalmazásprogramozási Felület) |
| CPU: | Central Processing Unit (Központi Feldolgozó Egység / Processzor) |
| GUI: | Graphical User Interface (Grafikus Felhasználói Felület) |
| HCI: | Human Computer Interaction (Ember-gép kapcsolat) |
| CIS: | Cognitive Information System (Kognitív információs rendszer) |

# Irodalmi áttekintés

## Logopédiai folyamatok

## IT eszközök és operációs rendszerek fejlődése

## Mobil alkalmazások fejlődése

A mobil eszközök és az azokon futó alkalmazások terén az elmúlt években látványos fejlődés tapasztalható. Az okostelefonok és tabletek terjedése és elterjedése lehetővé tette az emberek számára, hogy bármikor és bárhol hozzáférjenek az internethez és az online szolgáltatásokhoz. Az alkalmazásfejlesztés terén is számos változás történt. Korábban az alkalmazásokat külön-külön kellett fejleszteni az Android és az iOS rendszerekhez, ami nagyobb költségekkel és idővel járt. Azonban az utóbbi években az ún. cross-platform megoldások lehetővé teszik az alkalmazások egyszerre történő fejlesztését mindkét platformra, csökkentve a fejlesztési költségeket és időtartamot.

Ezzel párhuzamosan, a webtechnológiák terén is számos fejlesztés történt. A progresszív webalkalmazások (PWA-k) lehetővé teszik, hogy egy weboldal alkalmazás-szerűen működjön, offline üzemmódot és értesítéseket biztosítva a felhasználók számára. Az ilyen típusú alkalmazások előnye, hogy egyetlen kódbázisból indulnak, és azonnal futtathatóak a böngészőben, azonban számos olyan funkció is rendelkezésre áll, amelyeket korábban csak a natív alkalmazások biztosítottak.

Ezen trendek figyelembevételével a logopédiai folyamatok támogatását megvalósító alkalmazásom is PWA-ként fejlesztem, így lehetővé téve a felhasználók számára, hogy akár telefonjuk böngészőjéből is hozzáférjenek az alkalmazás szolgáltatásaihoz, illetve admin felülethez.

## PWA technológia

Az utóbbi években egyre népszerűbbé váltak a PWA-k (Progressive Web Applications), amelyek egyesítik a webalkalmazások és a natív alkalmazások előnyeit. A PWA-k az internet böngészőjében futnak, de telepíthetőek és olyan funkcionalitást kínálnak, amely hasonló a natív alkalmazásokhoz. A PWA-k az utóbbi időben egyre elterjedtebbek lettek a vállalatok és az üzleti felhasználók körében, mert könnyen fejleszthetők és frissíthetők, valamint a felhasználók is hasonló élményt tapasztalhatnak általuk, mint a natív alkalmazások esetén.

A PWA-knak számos előnyük van a hagyományos webalkalmazásokkal és a natív alkalmazásokkal szemben. Egyik legnagyobb előnye, hogy a felhasználók a böngészőjükből telepíthetik a PWA-kat, így nem szükséges az alkalmazások letöltése és telepítése a készülékekre, ami időt takarít meg és nem foglal helyet a felhasználói eszközökön. A PWA-k gyorsak és reszponzívak, és a felhasználók offline módban is használhatják őket, amely azok számára előnyös, akik korlátozott hozzáféréssel rendelkeznek az internethez. A PWA-k használata nem igényel különösebb technikai tudást, és a fejlesztőknek nem szükséges több platformra külön-külön alkalmazásokat fejleszteniük, ami időt takarít meg és egyszerűsíti a fejlesztési folyamatot.

Az előnyök mellett azonban vannak korlátjai is a PWA-knak, például korlátozottabb hozzáféréssel rendelkeznek a készülék hardveres erőforrásaihoz, mint a natív alkalmazások.

Az alkalmazás elkészítése mellett az egyik legfőbb motiváció az volt, hogy a logopédiai területen tevékenykedő szakemberek munkáját szeretném támogatni egy olyan alkalmazással, amely a legmodernebb technológiákat használja. A PWA technológia választása pedig nem véletlen, hiszen számos előnye miatt könnyen hozzáférhető és használható. Egyrészt minden platformon elérhető, így nem szükséges külön natív alkalmazásokat fejleszteni különböző platformokra. Emellett a PWA alkalmazások használata szinte azonnal elkezdhető, nem igényel telepítést és használatuk során a felhasználói élmény is közelítheti a natív alkalmazások szintjét.

## Mobil operációs rendszerek és sajátosságaik

Az okostelefonok már szinte minden embernél megtalálhatóak. A két leghasználtabb rendszer ezeken az eszközökön pedig az Android és az iOS. Bár sok hasonlóság van a kettő között, egy natív alkalmazás fejlesztése különböző módszerekkel, eszközökkel és lépésekkel történik.

Android esetén bármilyen operációs rendszeren lehet natív alkalmazást készíteni (akár magán Android-on is) Java vagy Kotlin (vagy alternatív környezetekben C#, JavaScipt, TypeScipt vagy C++) nyelven. iOS-nél pedig szükség van egy Macbook-ra, hogy build-elni tudjuk az alkalmazást, mindegy milyen környezetben vagy nyelven fejlesztettük.

Ezek a különbségek megnehezítik az egyéni fejlesztők és cégek dolgát. Bár már vannak technológiák arra, hogy egy kóddal 2 platformra egyaránt lehessen build-et készíteni, például a React Native, Flutter, Kotlin Multiplatform Mobile és sok más, de még így is rengeteg rendszerbéli különbségre kell odafigyelni a fejlesztés során.

A fentebb említett PWA technológiánál viszont nem kell annyira aggódni, hogy a kód máshogy működik a különböző operációs rendszereken vagy máshogy néz ki különböző kijelzőkön. De ez nem jelenti azt, hogy nincsenek különbségek.

Egy progresszív webalkalmazás működéséhez a felhasználónak szüksége van egy böngészőre. Minden rendszeren létezik alapértelmezett böngésző, de ezek nem mindig ugyanazok. Például Android (és Windows) operációs rendszereken a chromium motor jeleníti meg a PWA-kat, de iOS (és MacOS) rendszerek esetén a Safari webböngésző látja el ezt a feladatot. Ha párosítjuk a PWA-t egy modern keretrendszerrel, ami támogatja ezeket a böngészőket, akkor szinte semmilyen különbség nincs az Android-on és iOS-en (és minden máson) futó progresszív webalkalmazás működésében és megjelenítésében.

## Logopédia és gyógypedagógia területén használt alkalmazások felmérése

Bár az informatika gyorsan fejlődik, a magyar oktatásban alig látható ez a fejlődés az általános- és középiskolák körül. Viszont vannak törekvések a haladásra. Néhány helyen a logopédiai és gyógypedagógiai foglalkozásokon tableteket használnak oktató, fejlesztő játékokkal. De ezen kívül nem igazán látható más alkalmazása az informatikának, tehát a szakdolgozatom egy informatika szempontból még teljesen új területen fog működni.

# Feladat és követelmények

## Feladat leírása

## Funkcionális követelmények

# Felhasznált technológiák

## Verziókezelő rendszer

### Git

### GitHub

## Visual Studio Code

## Futtatókörnyezet és csomagkezelés

### Node.js

### npm

### Bun

## Programnyelvek

### JavaScript

### TypeScript

## React és Redux

## Firebase

### Authentication

### Hosting

### Firestore Database

### Realtime Database

### Storage

## PWA

## Google Chrome, mint fejlesztői eszköz

### Lighthouse

### React fejleszői bővítmények

## PWA kompatibilis operációs rendszerek

### Android

### iOS

### Windows

# Fejlesztői dokumentáció

## Fejlesztői környezet beállítása és projekt konfigurálása

### Fejlesztői eszközök telepítése

### Projekt beállítása és build-elés

## Adatreprezentáció

## Időpont foglalás működése

## Tárgy kölcsönzés működése

## Tesztelés

# Felhasználói dokumentáció

## Felhasználói felület

## Webalkalmazás elérése és telepítés

## Főbb funkciók menete

### Időpont foglalás

### Tárgy kölcsönzés

Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | T. De Smedt and W. Daelemans, “Pattern for python.,” *The Journal of Machine Learning Research,* vol. 13, no. 1, pp. 2063-2067, 2012. |
| [2] | „Záróvizsga információk,” [Online]. Available: https://mik.uni-pannon.hu/index.php/hu/oktatas/zarovizsga.html. [Hozzáférés dátuma: 04 03 2022]. |
| [3] | D. J. Wetherall és A. S. Tanenbaum, Computer networks, Pearson Education, 2013. |

Mellékletek

Mappaszerkezet

+logopedia-dashboard

| fájlok ide

|

| backen.bat

| backend.py

| files.doc

| fixedlinks.json

| ipcheck.py

| linkek.json

| linkfix.py

| log.txt

| sqlwriter.py

| textprocessor.py

|

+---backend

| pyvenv.cfg

+---lara

| | entities.py

| | nlp.py

| | parser.py

| | stemmer.py

+---static

| +---css

| | chat.css

| |

| \---js

| chat.js

|

+---templates

| index.html