

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM  
MAROSVÁSÁRHELYI KAR,  
INFORMATIKA SZAK**



**SAPIENTIA  
ERDÉLYI MAGYAR  
TUDOMÁNYEGYETEM**

AttendEase: Biztonságos jelenlét-nyilvántartó rendszer

**DIPLOMADOLGOZAT**

Témavezető:  
Dr. Iclánzan David,  
Egyetemi docens

Végzős hallgató:  
Buda Imola

**2023**

**UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA  
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,  
SPECIALIZAREA INFORMATICĂ**



**UNIVERSITATEA  
SAPIENTIA**

AttendEase: Sistem securizat de înregistrare a prezenței

**LUCRARE DE DIPLOMĂ**

Coordonator științific:  
Dr. Iclănanz David,  
Conferențiar universitar

Absolvent:  
Buda Imola

**2023**

**SAPIENTIA HUNGARIAN UNIVERSITY OF  
TRANSYLVANIA**  
**FACULTY OF TECHNICAL AND HUMAN SCIENCES**  
**COMPUTER SCIENCE SPECIALIZATION**



**SAPIENTIA**  
HUNGARIAN UNIVERSITY  
OF TRANSYLVANIA

AttendEase: Secure attendance manage system

**BACHELOR THESIS**

Scientific advisor:  
Dr. Iclănanz David,  
Associate professor

Student:  
Buda Imola

**2023**

**LUCRARE DE DIPLOMĂ**

Coordonator științific:  
**dr. Iclanzan David**      Candidat: **Buda Imola**  
Anul absolvirii: 2023

**a) Tema lucrării de licență:** Sistem securizat de înregistrare a prezenței bazat pe utilizarea codurilor QR generate dinamic de unică folosință.

**b) Problemele principale tratate:**

- Studiu bibliografic privind codurile QR și codurile de unică folosință.
- Proiectarea unui protocol de comunicare client-backend cu elemente multiple de siguranță, anti-falsificare și anti-fraudă (generarea de coduri conform unui protocol nepredictibil de către client, folosirea coordonatelor GPS și date SSID ale rețelei la nivel de client și transmitera acestora către backend, autentificare biometrică cu maximum 30 de secunde înainte de marcarea prezenței, operații prin tranzacții, stocare blockchain etc.).
- Proiectarea și realizarea unor aplicații pentru client, backend și a dashboardului de management (vizualizare prezență, configurare cursuri).
- Documentarea adecvată a stadiilor de proiectare, implementare și testare a aplicațiilor.

**c) Desene obligatorii:**

- Schema bloc a sistemului
- Diagrame de proiectare pentru aplicațiile software realizate.

**d) Softuri obligatorii:**

- Aplicație mobilă client pentru participanții la curs.
- Aplicație backend ce generează codurile și procesează cererile de înregistrare a prezenței într-un mod sigur, evaluând și verificând elementele de siguranță.
- Aplicație web de management a cursurilor și vizionare, exportare de statistici privind prezența.

**e) Bibliografia recomandată:**

WATERS, Joe. QR Codes for Dummies. John Wiley & Sons, 2012.

KIESEBERG, Peter, et al. QR code security. In: *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*. 2010. p. 430-435.

FAUZI, Ahmad Fahmi Mohd, et al. Development of web-based smart security door using qr code system. In: *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE, 2020. p. 13-17.

Krombholz, K., Frühwirt, P., Kieseberg, P., Kapsalis, I., Huber, M., & Weippl, E. (2014). QR code security: A survey of attacks and challenges for usable security. In *Human Aspects of Information Security, Privacy, and Trust: Second International Conference, HAS 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings* 2 (pp. 79-90). Springer International Publishing.

Wahsheh, H. A., & Luccio, F. L. (2020). Security and privacy of QR code applications: a comprehensive study, general guidelines and solutions. *Information*, 11(4), 217.

Focardi, R., Luccio, F. L., & Wahsheh, H. A. (2019). Usable security for QR code. *Journal of Information Security and Applications*, 48, 102369.

**f) Termene obligatorii de consultații:** săptămânal, preponderent online  
**g) Locul și durata practicii:** Universitatea „Sapientia” din Cluj-Napoca,  
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 413  
Primit tema la data de: 25.04.2022  
Termen de predare: 06.07.2023

Semnătura Director Departament

Semnătura responsabilului  
programului de studiu

Semnătura coordonatorului

Semnătura candidatului

## Declarație

Subsemnatul/a ..... Buda Ioana....., absolvent(ă) al/a specializării ..... informatică....., promovația 2020/2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, Corunca  
Data: 2023.06.15.

Absolvent

Semnătura.... Buda.....

# Kivonat

Az államvizsgadolgozatom témája egy biztonságos jelenlét nyilvántartó rendszer kidolgozása és bemutatása, melyet egy egyetem vagy iskola használhatna. Ennek célja, hogy a diákok jelenlétéét egy QR-kód alapú megoldás segítségével rögzítse, és hogy kizárja a csalást.

Az államvizsgadolgozatomban részletesen bemutatom a rendszer működését és előnyeit, kiemelem a rendszer biztonságát, amely a tanárok és diákok számára egyaránt nyugalmat és bizalmat biztosíthat. Ez a rendszer hatékony megoldást biztosít az órákon való jelenlét nyilvántartására, illetve segít fenntartani az órák rendjét.

Következtetéseimben összegzem az implementált rendszer előnyeit, korlátait és felvetem javaslataimat az esetleges továbbfejlesztési lehetőségekre.

A dolgozatomnak az a célja, hogy egy olyan jelenlét nyilvántartó rendszert biztosítson, amely gyors és Amit egyszerű használni, ami garantálja a diákok jelenlétének pontos, biztonságos nyilvántartását, és ami kivédi a csalás lehetőségét.

A jelenlét nyilvántartó rendszer bevezetése egy egyetemen vagy egy iskolában elősegítheti a hatékonyabb, biztonságosabb oktatást, hozzájárulva ezzel a diákok és tanárok eredményességéhez és hatékonyságához.

# Rezumat

Tema tezei mele de examen de stat este dezvoltarea și prezentarea unui sistem sigur de înregistrare a prezenței care ar putea fi utilizat de o universitate sau școală. Scopul sistemului este de a înregistra prezența studenților folosind o soluție bazată pe coduri QR și de a exclude frauda.

În teza de examen de stat, prezint în detaliu funcționarea și avantajele sistemului. Am subliniat securitatea sistemului, ceea ce înseamnă încredere și liniște sufletească pentru elevi și profesori. Sistemul oferă o soluție eficientă pentru înregistrarea prezenței și ajută la menținerea ordinii și fiabilității cursurilor.

În concluziile mele, rezum avantajele și limitările sistemului dezvoltat, precum și sugestiile mele pentru posibile dezvoltări ulterioare. Scopul tezei mele de examen de stat este de a oferi un sistem de înregistrare a prezenței care este rapid și ușor de utilizat, garantează înregistrarea corectă și sigură a prezenței studenților și previne posibilitatea fraudei.

Introducerea sistemului la o universitate sau o școală poate promova o educație mai eficientă și mai sigură, contribuind la satisfacția și eficacitatea elevilor și profesorilor.

# Abstract

The topic of my state exam thesis is the development and presentation of a secure attendance registration system that could be used by a university or school. The purpose of the system is to record students' attendance using a QR code-based solution and exclude fraud.

In my state exam thesis, I present the operation and advantages of the system in detail. I emphasized the security of the system, which means trust and peace of mind for students and teachers. The system provides an efficient solution for attendance registration and helps maintain the order and reliability of classes.

In my conclusions, I summarize the advantages and limitations of the developed system, as well as my suggestions for possible further developments. The purpose of my state exam thesis is to provide an attendance registration system that is quick and easy to use, guarantees accurate and safe registration of students' attendance, and prevents the possibility of fraud.

The introduction of the system at a university or a school can promote more efficient and safer education, contributing to the satisfaction and effectiveness of students and teachers.

# Tartalomjegyzék

<b>1. Bevezető</b>	<b>11</b>
<b>2. Elméleti háttér</b>	<b>12</b>
2.1. QR kódok . . . . .	12
2.2. QR kódos beléptető rendszerek . . . . .	14
<b>3. Követelmény-specifikációk</b>	<b>16</b>
3.1. Felhasználói követelmények . . . . .	16
3.2. Rendszer követelmények . . . . .	17
3.2.1. Funkcionális követelmények . . . . .	17
3.2.2. Nem-funkcionális követelmények . . . . .	17
<b>4. A rendszer részletes leírása</b>	<b>18</b>
4.1. A rendszer architektúrája . . . . .	18
4.2. Átjátszhatóság elkerülése . . . . .	19
4.2.1. Tranzakciókezelés . . . . .	19
4.2.2. A két eszköz távolsága egymástól . . . . .	20
4.3. Szekvenciadiagramok . . . . .	21
4.3.1. Regisztrációk szekvenciadiagramja . . . . .	21
4.3.2. Jelenlét bekerülésének szekvenciadiagramja . . . . .	22
4.4. Főbb komponensek . . . . .	23
4.4.1. A weboldal (tanár alkalmazás) főbb komponensei . . . . .	23
4.4.2. A diákok alkalmazás főbb komponensei . . . . .	27
<b>5. Használt technológiák</b>	<b>34</b>
5.1. React . . . . .	34
5.2. Vite . . . . .	35
5.3. Chakra UI . . . . .	35
5.4. React Native . . . . .	36
5.5. Firebase . . . . .	37
5.6. GitHub . . . . .	37
5.7. Trello . . . . .	39
5.8. LaTeX . . . . .	41
<b>6. A felhasználói felület bemutatása</b>	<b>42</b>
6.1. Weboldal (tanárok alkalmazása) . . . . .	42
6.1.1. Regisztráció . . . . .	42

6.1.2. Bejelentkezés . . . . .	43
6.1.3. Főoldal . . . . .	43
6.1.4. Jelenléti ív vezetés (QR kód generáló) . . . . .	44
6.2. Mobilapp (diákok alkalmazása) . . . . .	45
6.2.1. Regisztráció . . . . .	45
6.2.2. Bejelentkezés . . . . .	47
6.2.3. Jelenlét bevezetése (QR kód olvasó) . . . . .	48
6.2.4. Sikeres jelenlét bevezetése . . . . .	49
<b>7. Továbbfejlesztési lehetőségek</b>	<b>50</b>
7.1. Control panel . . . . .	50
7.2. Hashelt tokenek (blockchain) . . . . .	50
7.3. Jelenléti adatok exportálása . . . . .	50
7.4. Multiplatform támogatottság . . . . .	50
7.5. Több intézmény számára elérhetőség . . . . .	51
7.6. Diák alkalmazás kibővítése . . . . .	51
7.7. Biometrikus azonosítás . . . . .	51
7.8. Órapótlás lehetősége . . . . .	51
7.9. Regisztráció és belépés Google fiókkal . . . . .	51
7.10. A felhasználói felület javítása . . . . .	51
<b>Összefoglaló</b>	<b>52</b>
<b>Köszönetnyilvánítás</b>	<b>54</b>
<b>Ábrák jegyzéke</b>	<b>56</b>
<b>Irodalomjegyzék</b>	<b>57</b>

# **1. fejezet**

## **Bevezető**

Az oktatási intézményekben, legyen szó egyetemekről vagy iskolákról, a diákok jelenléte és részvételére az órákon kulcsfontosságú tényező. Az egyértelmű és pontos jelenlét nyilvántartása nem csupán az intézmények adminisztratív feladata, hanem alapvető fontosságú a hatékony oktatási folyamatok szempontjából. A hagyományos módszerek, például a papír-alapú jelenléti ívek vagy manuális bejelentkezési eljárások gyakran hibásak, időigényesek és csalásra is lehetőséget adnak.

Ebben a dolgozatban egy olyan biztonságos jelenlét nyilvántartó rendszer kidolgozására és bemutatására fókuszálok, amely megoldást nyújt a fent említett problémákra. A rendszer alapját a QR-kód technológia jelenti, amely könnyen alkalmazható és megbízható eredményeket nyújt a jelenlét nyilvántartásában.

A QR-kód alapú megoldás lehetővé teszi a diákok számára, hogy egyedi QR-kódokkal rendelkezzen a jelenlétéük, amelyeket a tanárok digitális eszközeikről könnyedén leolvashatnak. Ezáltal a diákok jelenléte az adott órákon egyszerűen és pontosan rögzíthető. A rendszer folyamatosan figyeli a két eszköz távolságát, így kiküszöböli a csalási kísérleteket, amikor a diákok és a tanár eszköze túl messze vannak egymástól.

A dolgozatban bemutatom a rendszer tervezését és implementálását, valamint leírom a használt technológiákat és annak alkalmazási lehetőségeit. Megvizsgálom a rendszer teljesítményét, hatékonyságát és az esetleges hiányosságokat, fejlesztési, optimalizálási lehetőségeket.

A biztonságos jelenlét nyilvántartó rendszer kidolgozása és alkalmazása hozzájárulhat a hatékonyabb oktatáshoz és az intézmények adminisztratív terheinek csökkentéséhez. A dolgozat célja, hogy a QR-kód alapú megoldást bemutatva segítse az oktatási intézményeket a hatékonyabb és megbízhatóbb jelenlét nyilvántartás elérésében. [?]

## 2. fejezet

### Elméleti háttér

#### 2.1. QR kódok

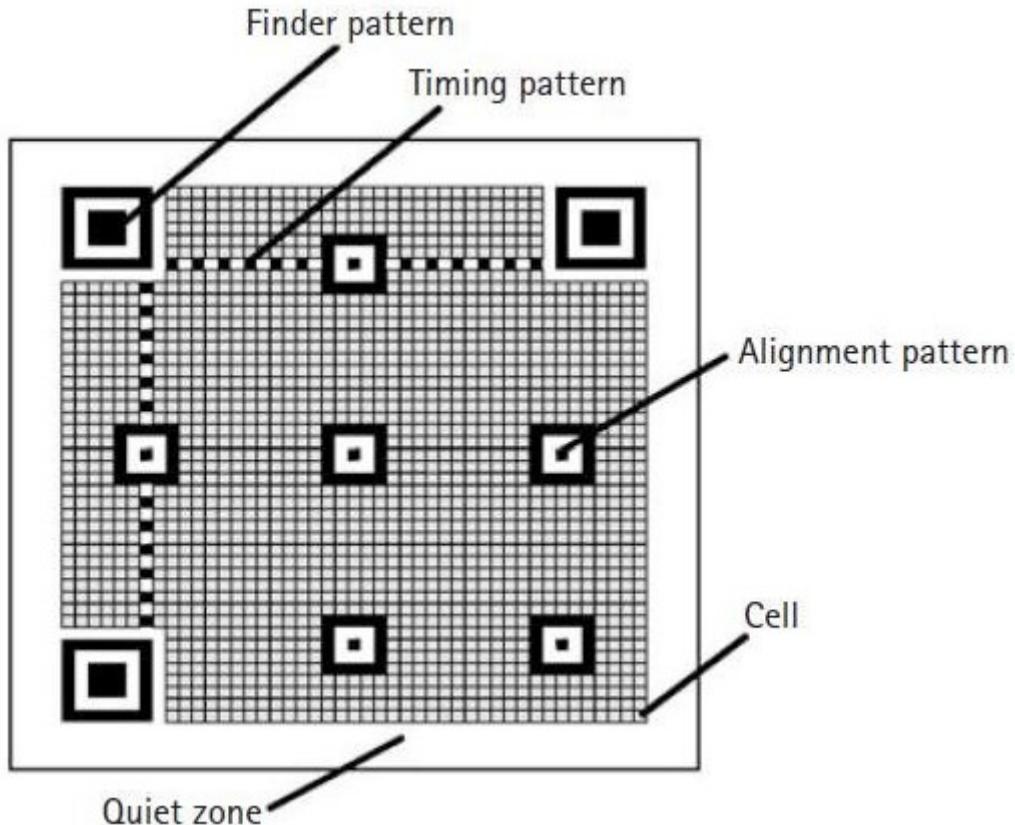
[1]

A QR kódok (Quick Response Codes) népszerű vonalkód típusok, amelyeket gyakran használnak információk tárolására és megosztására. Ezek a kódok kétdimenziós és négyzet alakú mintázatokból állnak, amelyekben különböző adatok kódolhatók, például szövegek, webhivatkozások, kontaktinformációk vagy egyedi azonosítók.

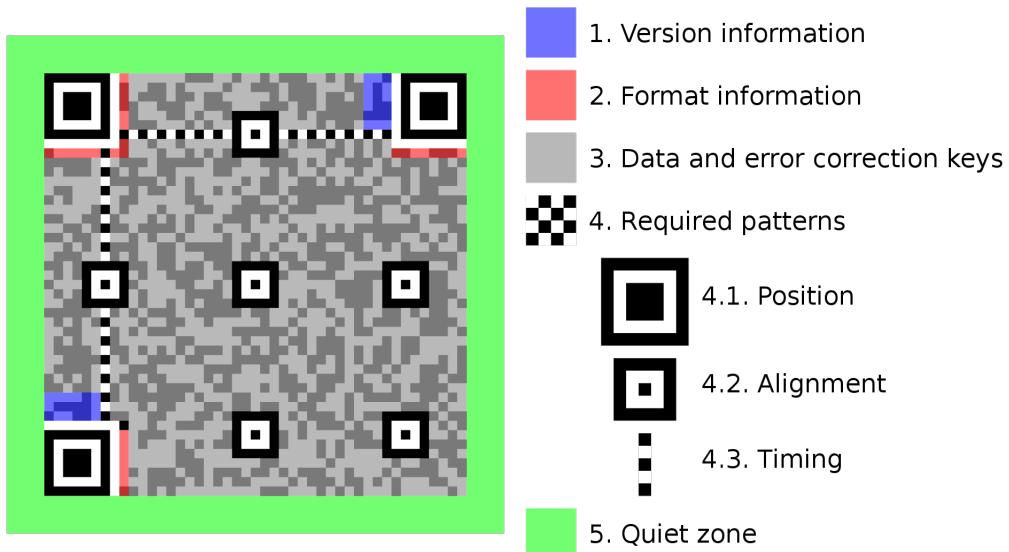
A QR kódok előnye, hogy nagy mennyiségű információt képesek tárolni és könnyen leolvashatók a megfelelő QR kód olvasó alkalmazásokkal vagy eszközökkel. Ez a technológia széles körben elterjedt az üzleti, reklám- és marketingterületeken, valamint az oktatásban és az események szervezésében.

A QR kódok alkalmazása számos előnyt kínál. Kényelmes és gyors módszert nyújtanak információk megosztására, mivel egyetlen leolvasással könnyedén hozzáférhetővé válnak az adatok. Továbbá, a QR kódok hordozhatók és könnyen nyomtathatók különböző felületekre, például plakátokra, címkkre vagy névjegyekre.

A QR kódok felépítése összetett, és tartalmazza az adott információk bináris reprezentációját. A kódok egy speciális mintázatból állnak, amelyeket a leolvasó eszköz érzékel és értelmez. A kódok tartalmazhatnak szöveget, számokat, hivatkozásokat vagy más adatokat, amelyeket a rendszer feldolgozhat. [2]



**2.1. ábra.** QR kódok felépítése



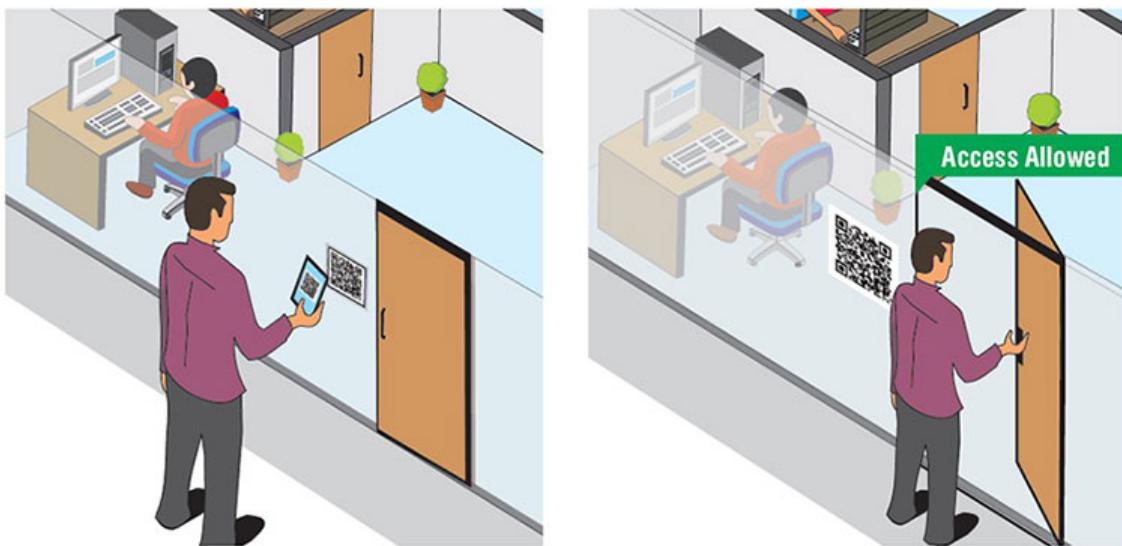
**2.2. ábra.** QR kódok felépítése

A QR kódok technológiája folyamatos fejlődésen megy keresztül, és újabb lehetőségek nyílnak meg a funkcionális és a kreatív felhasználás terén. A kutatások arra összpontosítanak, hogy optimalizálják a QR kódok felépítését és leolvasását.

A könnyű felhasználhatóság és az adatok sokfélesége lehetővé teszi széles körű alkalmazásukat különböző területeken, beleértve a beléptető rendszerek fejlesztését és a jelenlét nyomon követését is.

## 2.2. QR kódos beléptető rendszerek

[3]



**2.3. ábra.** QR kódos beléptető rendszerek

A QR kódokat gyakran használják az elektronikus beléptető rendszerekben, mivel biztonságos és hatékony módszert nyújtanak a személyek azonosítására és jelenlétének rögzítésére.

Az elektronikus beléptető rendszerekben a QR kódokat használják a személyek azonosítására. A QR kódokkal működő beléptető rendszerek számos megoldást kínálnak a biztonságos és hatékony működéshez.

Az ilyen rendszerek előnyei közé tartozik a gyors és hatékony beléptetési folyamat, az adatok pontos rögzítése és könnyű visszakeresése, valamint a csalások és a hibák minimalizálása. A QR kódok biztonságosak és nehéz másolni vagy manipulálni őket, ezáltal erősítve a rendszer megbízhatóságát. Továbbá, a QR kódok alkalmazása a felhasználók számára is egyszerű és kényelmes, mivel csak egy eszközre van szükségük a kód leolvasásához.

Számos kutatási eredmény és technikai megoldás található a QR kódok felhasználásával kapcsolatban a beléptető rendszerekben.

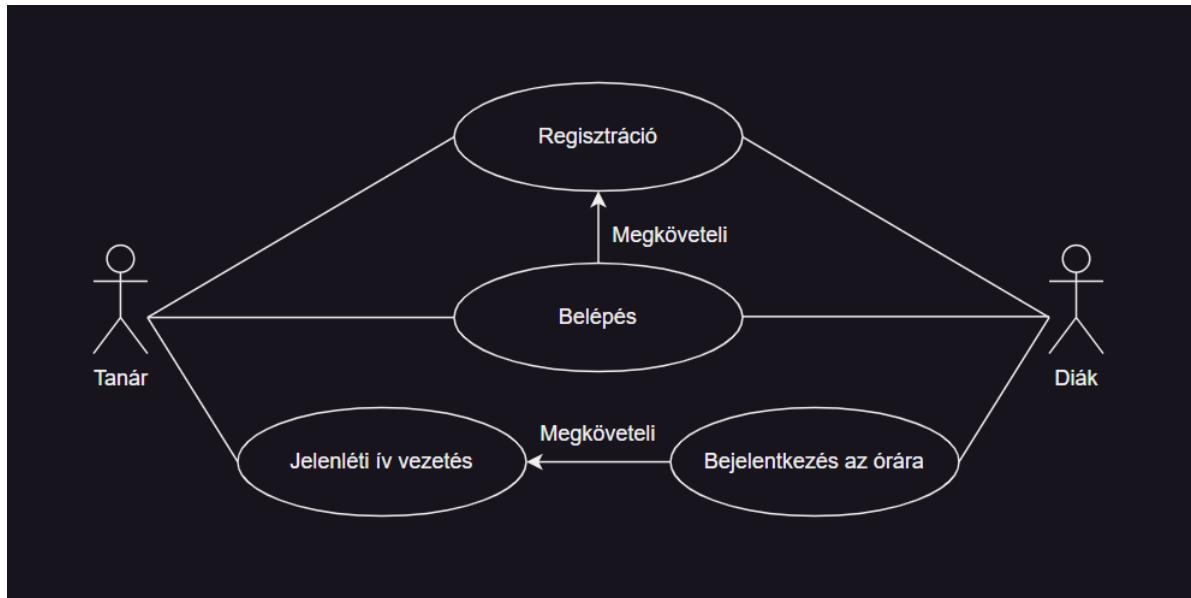
A QR kódos beléptető rendszerek széles körben alkalmazhatók, és számos előnyt kínálnak a személyazonosság azonosításában és a különböző információk rögzítésében. Az áttekintett szakirodalom alapján megállapítható, hogy ezek a rendszerek biztonságosak, megbízhatóak és könnyen használhatók különböző környezetekben, mint például oktatási intézmények, vállalatok vagy egyéb létesítmények. A további kutatások és fej-

lesztések segíthetnek a QR kód alapú beléptető rendszerek további hatékonyságának és funkcionalitásának javításában.

Az ilyen beléptető rendszerek továbbfejlesztése és testreszabása számos lehetőséget kínál. Például, a rendszerhez kiegészítő funkciók, mint például a GPS alapú helyzetellenőrzés, biometrikus azonosítás vagy távoli hozzáférési lehetőség is implementálható. Az eszközök és technológiák folyamatos fejlődése további innovatív megoldásokat és biztonsági intézkedéseket eredményezhet a QR kódos beléptető rendszerek területén.

## 3. fejezet

# Követelmény-specifikációk



3.1. ábra. Use Case diagram

### 3.1. Felhasználói követelmények

A felhasználói követelmények figyelembevételével lehet megtervezni és kifejleszteni egy olyan biztonságos jelenlét nyilvántartó rendszert, ami megfelel az oktatási intézmények igényeinek, és ami elősegíti a rendszer hatékony működését. Ezek a következők:

- A diákok és a tanárok alkalmazásának a felhasználói felülete egyaránt áttekinthető, érthető és könnyen kezelhető kell legyen.
- A rendszernek tudnia kell, hogy a tanárnak az aktuális időpontban éppen milyen órája van, ez által megkönnyítve, jobb felhasználói élményt nyújtva a tanárnak.
- A rendszernek helyesen kell kezelnie a diákok jelenlétének bejegyzését, a diákokat a megfelelő órára kell megjelentetni bejegyezni és ezt azonnal rögzítenie kell, hogy ne legyenek felesleges várakozási idők.

- A rendszernek maximálisan megbízhatónak kell lennie, nem szabad lehetőséget hagyjon a csalásra.
- Lehetővé kell tennie, hogy a tanár az alkalmazásból követhesse a statisztikákat, az órák jelenléti ívét.
- A rendszernek ellenálló képességgel kell rendelkeznie a kiberbiztonsági fenyegetésekkel szemben, például védelemmel az adatbányászat, adatlopás vagy adatvesztés ellen.

## **3.2. Rendszer követelmények**

A rendszerkövetelmények azokat a technikai és infrastrukturális követelményeket foglalják magukban, amelyeket a biztonságos jelenlét nyilvántartó rendszernek teljesítenie kell ahhoz, hogy megfelelően működjön. Ezek a funkcionális és nem-funkcionális követelmények.

### **3.2.1. Funkcionális követelmények**

A funkcionális követelmények meghatározzák a rendszer elvárt működését és funkcionalitását, és biztosítják, hogy a rendszer megfeleljen a felhasználók elvárásainak és igényeinek a jelenlétnyilvántartás terén. Ezek a követelmények a következők:

- A diákoknak és a tanároknak lehetőségük kell legyen regisztrálni a rendszerbe.
- A felhasználói azonosításnak biztonságos és megbízható módon kell történnie, például felhasználónév és jelszó módszer használatával.
- A rendszernek képesnek kell lennie generálni egyedi QR-kódokat minden órán, minden diákk számára.
- A tanároknak lehetőségük kell legyen, hogy az eszközeiken megjelenítsék a diákok által leolvasandó QR-kódokat.
- A rendszernek azonnal rögzítenie kell a diákok jelenlétét, amint a QR-kód leolvasásra került.
- A rögzített jelenléti adatoknak pontosnak és megbízhatónak kell lenniük, és könnyen hozzáférhetőknek kell maradniuk az tanárok számára.
- A rendszernek megfelelő adatvédelmi intézkedéseket kell alkalmaznia a felhasználók adatainak védelme érdekében.
- A felhasználók személyes adatainak biztonságos tárolása és kezelése elengedhetetlen.

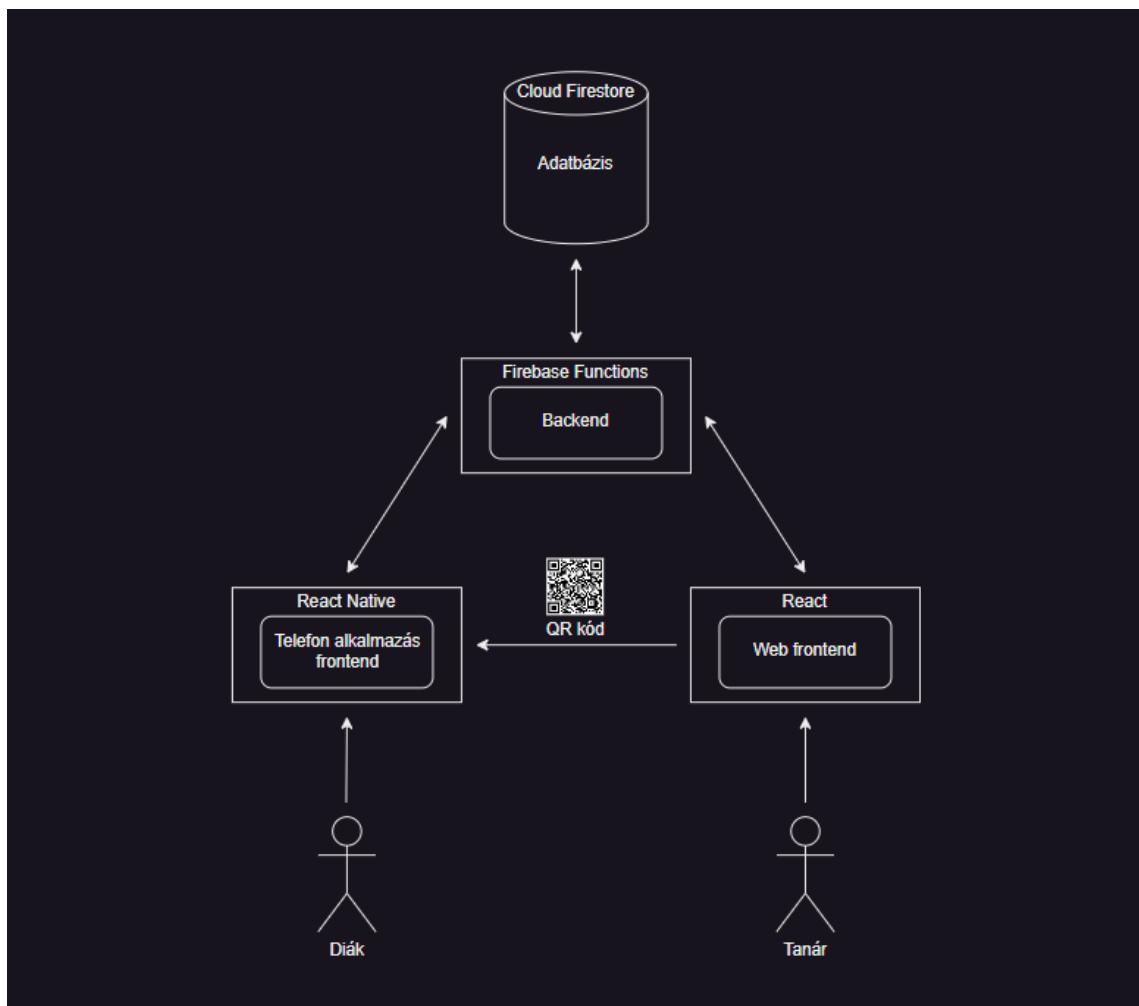
### **3.2.2. Nem-funkcionális követelmények**

- Szükség van egy olyan eszközre, ami képes böngészőt futtatni, kapcsolódik az internethez és billentyűzet és egér segítségével tud interaktálni a weboldallal.
- Szükség van egy Android operációs rendszert támogató eszközre, ami kapcsolódik az internethez.
- A felhasználók, mind a tanárok, mind a diákok regisztrált adatainak tárolása egy valós idejű adatbázisban történik, ehhez Firebase szükséges.
- Firebase funkciók szükségesek az azonosításhoz és a diákok jelenlétének beírásához az adatbázisba az adott órára.

## 4. fejezet

# A rendszer részletes leírása

### 4.1. A rendszer architektúrája



4.1. ábra. Architektúra

A rendszer tervezésénél első sorban próbáltam arra összpontosítani, hogy jól különálló és egyenként tesztelhető komponensekből álljon össze, ez által biztonságosabbá téve

a rendszert és megakadályozva, hogy a felhasználók csalásnak áldozatai legyenek, vagy hogy ők maguk csaljanak.

A rendszer architektúrája az alkalmazás kliens- és szerveroldalát elkülöníti egymástól, ami biztosítja a megbízhatóságot és biztonságot. A telefonos alkalmazás és a weboldal frontendje a felhasználói interfészt nyújtja, míg a Firebase backend szerver oldali funkciókat biztosít, például az adatok kezelését és az üzleti logika végrehajtását.

Az ábrán jól látható, hogy teljesen elkülönül a telefon alkalmazás frontendje és a weboldal frontendje is a backendtől. A felhasználók, mind a tanár és a diák, kizárolag a frontenddel kommunikálnak, ezáltal semmilyen hozzáférésük nincs az adatbázishoz.

## 4.2. Átjátszhatóság elkerülése

Dolgozatom során célom az, hogy ez a rendszer minél inkább megbízható legyen, ennek érdekében két módszert vezetek be. Először is, a tokenek felhasználása során tranzakciókezelést alkalmazok. Másodszor pedig a rendszer figyelemmel kíséri a diákok helyzetét a tanár eszközének heylzetéhez képest GPS koordináták alapján.

Ezek a módszerek biztosítják, hogy a jelenlét nyilvántartó rendszerben ne lehessen átadni vagy manipulálni a jelenléti adatokat. A tranzakciókezelés és a helyzetellenőrzés kombinációja garantálja az adatok integritását és a rendszer megbízhatóságát, tovább csökkentve a csalási lehetőségeket.

### 4.2.1. Tranzakciókezelés

A tokenek elfogyasztása során a jelenlét nyilvántartó rendszerben bizonyos intézkedéseket hozok annak érdekében, hogy elkerüljem az átjátszhatóságot és a csalásokat, itt arra gondolok, hogy a diák által generált tokeneket csak egyszer használhatóvá tessük, és a rendszer rögzíti és ellenőrzi ezeket a felhasználásokat.

Mielőtt a diák beolvassa a QR kódot, a rendszer generál egy egyedi tokenet számára. Ez a token lehetőséget ad a diáknak a jelenlét regisztrálására az adott órán vagy kurzuson. Azonban a rendszer azt is figyeli, hogy a tokeneket csak egyszer lehessen felhasználni, tehát egyszerűen elfogyasztjuk azokat a felhasználás során.

A token elfogyasztása a tranzakciókezelés során történik. A tranzakciókezelés során az adatbázisban rögzítem és kezelem a tokenekkel kapcsolatos tranzakciókat. Amikor a diák beolvassa a QR kódot, a rendszer generál egy egyedi token-t, és ezt rögzíti a tranzakciós adatokban. A rendszer leellenőrzi, hogy a diák által szkennelt token még érvényes-e és nem került-e már felhasználásra. Ha a token érvényes, akkor elfogyasztjuk azt a tranzakció rögzítése során, és rögzítjük, hogy a diák jelen volt az adott órán vagy kurzuson. Ezután a rendszer nem fogadja el újra ugyanazt a token-t a jövőben, így megakadályozza az átjátszhatóságot és a csalásokat.

A tranzakciókezelési folyamat során az adatok integritását és hitelességét ellenőrzöm.

A tokenek elfogyasztásának módszere garantálja, hogy a jelenlét nyilvántartás megbízható és manipulációmentes legyen. A felhasznált tokeneket rögzítem, így nyomon követhetővé válik a diák jelenléte, és megakadályozom, hogy ugyanazt a token-t többször használhassák fel. Ez biztosítja az adatok integritását és a rendszer megbízhatóságát a jelenlét nyilvántartás során.

#### **4.2.2. A két eszköz távolsága egymástól**

Az eszközök GPS-koordinátáinak hasonlítása egy további intézkedés a jelenlét nyilvántartó rendszerben, amely hozzájárul az átjátszhatóság és csalások elkerüléséhez. Ennek lényege, hogy a rendszer figyelemmel kíséri és összehasonlítja a diák által küldött GPS-koordinátákat a tanár eszközének helyzetéhez képest.

Amikor a diák beolvassa a QR kódot, az eszköze GPS-koordinátákat küld a rendszernek, azaz a rendszer rögzíti a diák aktuális tartózkodási helyét. Ugyanígy a tanár eszközének is lekéri a rendszer a helyzetét.

Amikor a rendszer ellenőrzi a beolvasott tokent, akkor a rendszer összehasonlítja a diák aktuális helyzetét a tanár helyzetével, és csak akkor fogadja el a jelenlétet, ha a diák ténylegesen az egyetem vagy iskola területén tartózkodik.

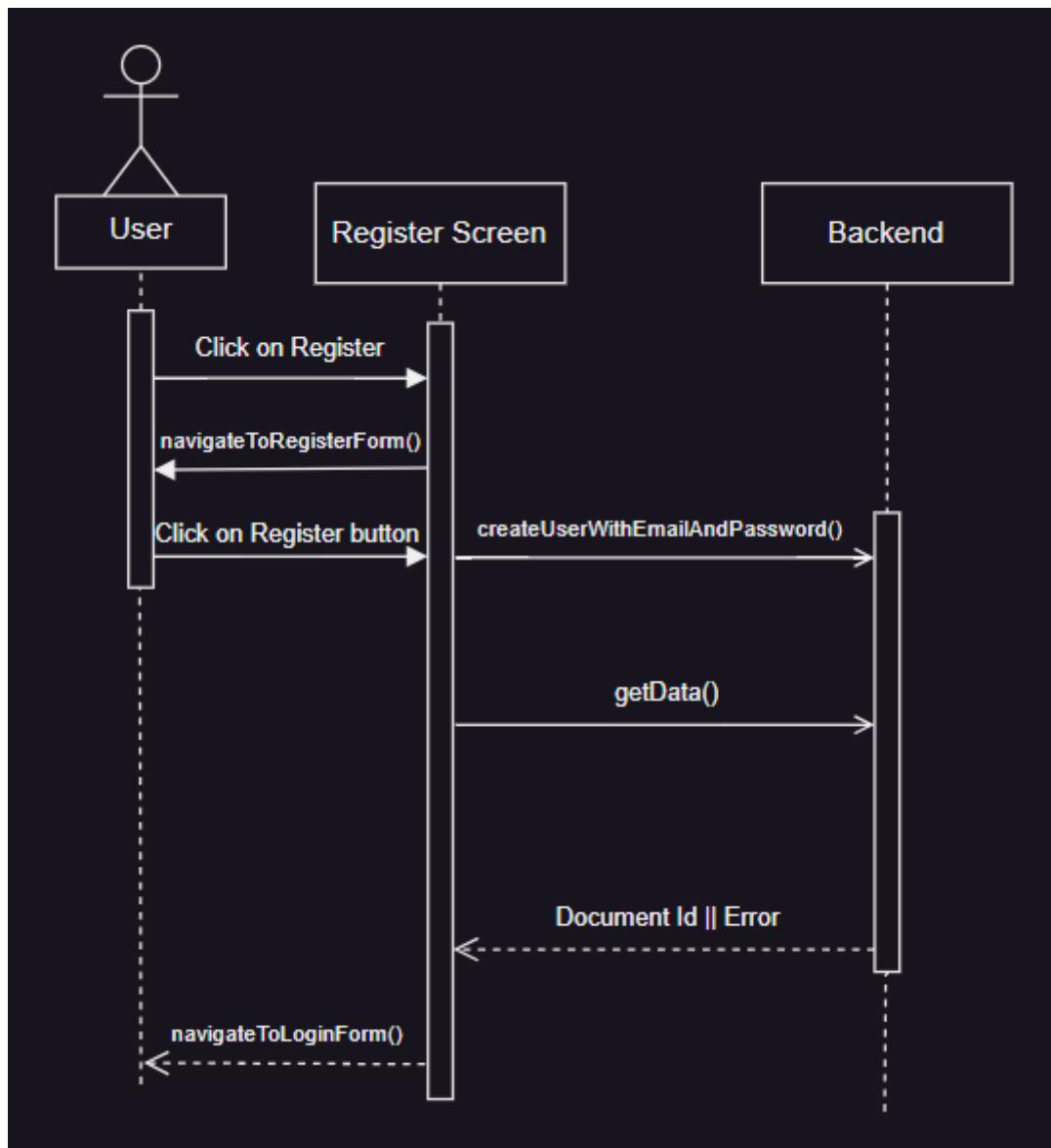
Az eszközök GPS-koordinátáinak hasonlítása során összehasonlítom a diák által küldött koordinátákat a tanár eszköze által küldött koordinátákkal. Ha a koordináták közel vannak egymáshoz vagy a meghatározott területen belül vannak, akkor elfogadom a jelenléti információt és rögzítem, hogy a diák valóban ott tartózkodott. Ha viszont a koordináták jelentősen eltérnek vagy a meghatározott területen kívül vannak, akkor kétség merülhet fel a diák jelenlétével kapcsolatban.

Az eszközök GPS-koordinátáinak hasonlítása lehetővé teszi a rendszer számára, hogy ellenőrizze és összevesse a diák helyzetét az elvárt tartózkodási hellyel. Az eszközök GPS-koordinátáinak hasonlítása tehát egy további biztonsági intézkedés, amely megerősíti a jelenlét nyilvántartó rendszer megbízhatóságát és manipulációmentességét.

## 4.3. Szekvenciadiagramok

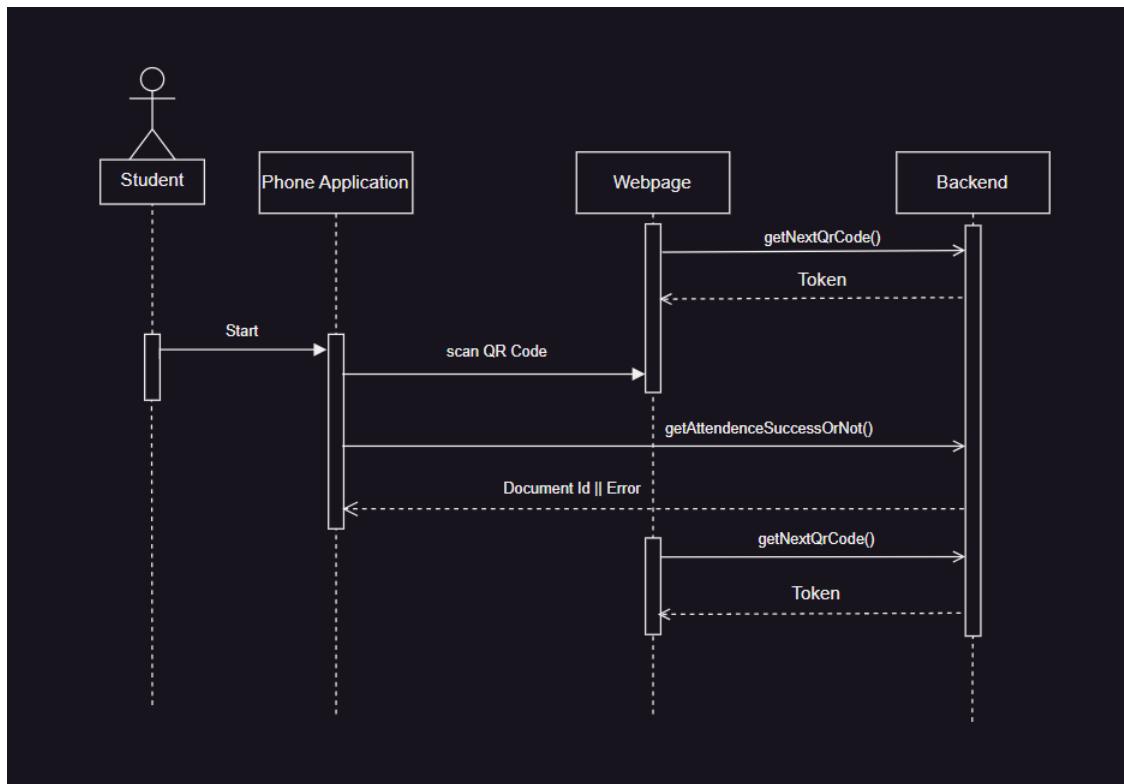
### 4.3.1. Regisztrációk szekvenciadiagramja

A telefonos és a webes regisztráció nagyon hasonló.



4.2. ábra. Regisztrációk szekvenciadiagramja

#### 4.3.2. Jelenlét bekerülésének szekvenciadiagramja



4.3. ábra. Jelenlét bekerülésének szekvenciadiagramja

## 4.4. Főbb komponensek

### 4.4.1. A weboldal (tanár alkalmazás) főbb komponensei

#### Regisztráció

```
const register = async () => { //async - külön fut és nem "fagy ki", lehet interakálni a kinezettel
  try{
    const user = await createUserWithEmailAndPassword(auth, registerEmail, registerPassword); //auth - firebase-configból
    console.log(user);
    getData();
    navigateToLoginForm();
  } catch (error) {
    console.log(error.message);
  }
}
```

4.4. ábra. Tanár regisztráció

A tanár regisztrálni tud az email címével és a jelszavával, ehhez egy firebase beépített függvényt hívtam meg, a createUserWithEmailAndPassword-ot.

```
const getData = async () => {
  try {
    const document = await addDoc(collection(db, "Teachers"), {
      dateOfBirth: calendar,
      teacherEmail: registerEmail,
      teacherName: name,
    });
    console.log("Document written with ID: ", document.id);
  } catch (e) {
    console.error("Error adding document: ", e);
  }
}
```

4.5. ábra. Tanár adatainak elmentése regisztrációkor

Hasonlóan a diákok regisztrációjához, amikor a tanár regisztrál, nemcsak az email címét és a jelszavát, hanem a teljes nevét és a születési dátumát is kéri megadni az alkalmazás. Amikor a felhasználó regisztrál, akkor nemcsak regisztrál, hanem emellett még beíródik az adatbázisba is, létrejön egy dokument a Teachers táblában a megfelelő információkkal róla.

## Bejelentkezés

```
const login = async () => { //async - külön fut és nem "fagy ki", lehet interaktálni a kinezettel
  try{
    const user = await signInWithEmailAndPassword(auth, loginEmail, loginPassword); //auth - firebase-configbol
    console.log(user);
    navigateToHomepage();
  } catch (error) {
    console.log(error.message);
  }
}
```

4.6. ábra. Tanár bejelentkezés

A tanár is természetesen a regisztrált adatokkal, a megfelelő email címmel és a megfelelő jelszóval tud belépni a weboldalra, ahol a jelenlétet vezetni tudja.

## Qr kód generátor, azaz jelenléti ív vezetése

```
useEffect(() => [
  const fetchQrCode = async () => {
    try {
      const user = auth.currentUser;
      const functions = getFunctions();
      const getNextQrCode = httpsCallable(functions, 'getNextQrCode');

      await Promise.all([auth, user, functions, getNextQrCode]);

      getNextQrCode({ email: user.email, token:qrCode, webLatitude: latitude, webLongitude: longitude })
        .then(result) => {
          setQrCode(result.data);
          console.log(result.data);
        });
    } catch(error) {
      showAlert = true;
      console.log('Error fetching: ' + error);
    }
  }

  onAuthStateChanged(auth, (user) => {
    if (user) {
      // User is signed in
      fetchQrCode();
    } else {
      // User is signed out
      console.log('not logged in');
    }
  });
], [checking]);
```

4.7. ábra. Qr kód megjelenítése

A frontend kéri a következő QR kódot, figyeli a változást, ha nem érvényes már a jelenlegi QR kód, akkor frissíti azt.

```

//GET nextQRCode
exports.getNextQRCode = functions.https.onCall(async (data, context) => {
  const documentId = data.email;
  const token = data.token;
  const webLat = data.webLatitude;
  const webLong = data.webLongitude;

  // console.log("webLong: " + webLong);
  let valid = "";
  if (token != ""){
    valid = await checkValidToken(token)
    .then((result) => {
      console.log(result);
      return result;
    })
    .catch((error) => {
      console.error(error);
      throw new Error('An error occurred while retrieving the document.');
    });
    console.log(valid);
    if (valid == "valid"){
      return token;
    }
  }else{
    console.log("ures a token");
  }

  let teacherID = "";
  let courseID = "";
  let classID = "";
  let randomToken = "";

  teacherID = await getTeacherIDByEmail(documentId)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });

  courseID = await getCourseIDByTeacherID(teacherID)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });

  classID = await getClassIDByCourseID(courseID)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });

  randomToken = generateRandomString(10);
  await createQRCodeByClassID(classID, randomToken, webLat, webLong);
  return randomToken;
});

```

4.8. ábra. Qr kód generálása

Először ellenőrzi a backend, hogy a leszkennelt token érvényes-e, azaz létezik-e ilyen tokennel document a QrCodes táblában. Ha létezik, akkor ezzel a tokennel kell majd beíródjon a jelenlét a diáknak az Attendence táblába.

Szükséges tudnunk a tanár ID-ját, amit az email címe alapján keres meg a backend, ugyanígy a tanár ID szerint megkeresi a tanár által tanított tantárgyat, ezután pedig megnézi a Courses táblában, hogy melyik az a megfelelő tantárgyú óra, aminek megtartására az adott időben (jelenleg) kerül sor.

Ez alapján megkaptam az órát, amihez kell generálni a tokeneket a QrCodes táblában, ez után ezt generálja majd a backend minden egyes alkalommal, amikor a frontend ezt a kérést küldi felé.

```
function success(position) {
    const latitude = position.coords.latitude;
    const longitude = position.coords.longitude;
    setLongitude(longitude);
    setLatitude(latitude);
    console.log(`Latitude: ${latitude}, Longitude: ${longitude}`);
}

function error() {
    console.log("Unable to retrieve your location");
}

useEffect(() => {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(success, error);
    } else {
        console.log("Geolocation not supported");
    }
}, []);
```

4.9. ábra. Tanár eszközének a helyének lekérése

Szükséges minden eszköz helyzetét lekérni ahhoz, hogy össze tudjam hasonlítni őket és meg tudjam határozni, hogy a diákok valóban ott tarózkodik-e a tanár eszközének közelében, azaz az egyetemen vagy iskolában.

#### 4.4.2. A diák alkalmazás főbb komponensei

##### Regisztráció

```
const RegisterUser = () => {
  createUserWithEmailAndPassword(authentication, email, password)
    .then((re) => [
      console.log(re),
      console.log(email),
      getData(),
      onPress(),
    ])
    .catch((err) => {
      console.log(err);
    })
};
```

4.10. ábra. Diák regisztráció

A diákok regisztrálni tud az email címével és a jelszavával, ehhez egy firebase beépített függvényt hívtam meg, a createUserWithEmailAndPassword-ot.

```
const getData = async () => {
  try {
    const document = await addDoc(collection(db, "Students"), {
      dateOfBirth: dateOfBirth,
      studentEmail: email,
      studentName: name,
    });
    console.log("Document written with ID: ", document.id);
  } catch (e) {
    console.error("Error adding document: ", e);
  }
}
```

4.11. ábra. Diák adatainak elmentése regisztráció során

Amikor a diákok regisztrálják az alkalmazást, nemcsak az email címét és a jelszavát, hanem a teljes nevét és a születési dátumát is kéri megadni az alkalmazás. Amikor a felhasználó regisztrál, akkor nemcsak regisztrál, hanem emellett még beíródik az adatbázisba is, létrejön egy új document a Students táblában a megfelelő információkkal róla.

## Bejelentkezés

```
const SignInUser = () => {
  signInWithEmailAndPassword(authentication, email, password)
    .then((re) => {
      setIsSignedIn(true);
      console.log(email);
      onPress2();
      authentication.isSignedIn
    })
    .catch((err) => {
      console.log(err);
    })
}
```

4.12. ábra. Diák bejelentkezés

A diákok természetesen a regisztrált adatokkal, a megfelelő email címmel és a megfelelő jelszóval tud belépní az alkalmazásba, ahol a jelenlétet igazoltatni tudja az adott órára.

## Kijelentkezés

```
const SignOutUser = () => {
  signOut(authentication)
    .then((re) => {
      setIsSignedIn(false);
    })
    .catch((err) => {
      console.log(err);
    })
}
```

4.13. ábra. Diák kijelentkezés

A diákok ki is tud jelentkezni az alkalmazásból.

## Qr kód szkenner

```
//request camera permission
useEffect( () => {
  askForCameraPermission();
}, []);

const askForCameraPermission = () => {
  (async () =>{
    const {status} = await BarCodeScanner.requestPermissionsAsync();
    setHasPermission(status == 'granted')
  })()
}
```

4.14. ábra. Kamera hozzáférési engedély kérése

Az alkalmazás engedélyt kér a kamerához való hozzáféréshez, ennek segítségével tudja majd leolvasni a QR kódot.

```
const [location_longitude, setLocation_longitude] = useState(0);
const [location_latitude, setLocation_latitude] = useState(0);

useEffect (() => {
  (async () => {
    let {status} = await Location.requestForegroundPermissionsAsync();
    if(status == 'granted'){
      console.log("Permission granted");
    }else{
      console.log("Permission not granted");
    }

    const loc = await Location.getCurrentPositionAsync();
    setLocation_latitude(loc.coords.latitude);
    setLocation_longitude(loc.coords.longitude);
    console.log(location_latitude);
    console.log(location_longitude);
  })()
},[]);
```

4.15. ábra. Helymeghatározás hozzáférési engedély kérése

Az alkalmazás engedélyt kér a helymeghatározáshoz való hozzáféréshez, ennek segít-ségével tudja meghatározni a diákok helyzetét és majd összehasonlítani a tanár eszközének helyzetével.

```

//what happens when we scan qrcode
const handleQRCodeScanned = ({type, data}) => {
    setScanned(true);
    setToken(data);
    console.log('Type: ' + type + '\nData' + data);

    try{
        const user = auth.currentUser;
        const functions = getFunctions();
        const getAttendanceSuccessOrNot = httpsCallable(functions, 'getAttendanceSuccessOrNot');

        let document = getAttendanceSuccessOrNot({email: user.email, token: data, latitude: location_latitude, longitude: location_longitude})
            .then((result) => {
                console.log("courseName: " + result.data.courseName);
                setClass_(result.data.courseName);
                setTime_(result.data.time);
                setToggle(true);
                return result;
            });
    } catch(error){
        console.log(error);
    }
}

```

**4.16. ábra.** QR kód szkennelés lekezelése

Amikor a diák leszkenneli a QR kódot, akkor a rendszer az email címe, a leszkennelt QR kód (azaz token) és a fizikai helyzetéhez megfelelően fogja beírni vagy nem írni be a diáknak a jelenlétet a megfelelő órára.

```

// ATTENDENCE for the student
exports.getAttendanceSuccessOrNot = functions.https.onCall(async (data, context) => {
  let studentID = "";
  let classID = "";
  let token = "";

  token = data.token;

  classID = await getClassIDByToken(data.token,data.latitude,data.longitude)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });

  studentID = await getStudentIDByEmail(data.email)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });

  const docRef = admin.firestore().collection('Classes').doc(classID);
  const document = await docRef.get();

  let time = document.data().time;
  const docRef2 = admin.firestore().collection('Courses').doc(document.data().courseID);
  const document2 = await docRef2.get();

  console.log("A talált ora neve: " + document2.data().courseName);
  console.log("A talált ora ideje: " + time);

  let text = await createAttendanceByEmailAndToken(token, classID, studentID);

  const attendanceInfo = {
    courseName: document2.data().courseName,
    time : time
  }
  return attendanceInfo;
});

```

4.17. ábra. A diák jelenlétének kezelése

A backend megkeresi, hogy melyik óra tartozik a megfelelő tokenhez a QrCodes táblában, illetve azt, hogy melyik studentID tartozik a bejelentkezett felhasználó email címéhez, ez után pedig meghívódik a jelenlét beírásának függvénye.

```

async function createAttendanceByEmailAndToken(token, classID, studentID){
  const newAttendance = {
    attendanceToken : token,
    classID : classID,
    studentID : studentID
  }

  let has = await studentHasAttendance(classID,studentID)
  .then((result) => {
    console.log(result);
    return result;
  })
  .catch((error) => {
    console.error(error);
    throw new Error('An error occurred while retrieving the document.');
  });
  console.log("van jelenlet vagy nincs: " + has);

  if (has == "hasAttendence"){
    return "Has Attandence already";
  }

  await admin.firestore().runTransaction(async (req,res) => {
  try {
    const doc = await admin.firestore().collection('QrCodes').get();
    let docID;
    doc.forEach(document => {
      if (document.data().qrCode == token) {
        docID = document.id;
        console.log(document.id, '=>', document.data());
      }
    });
    console.log("most fogok torolni : "+docID);
    const result = await admin.firestore().collection('QrCodes').doc(docID).delete();
    console.log(result);
    return result;
  }catch (errrror){
    return errrror.message;
  });
}

await admin.firestore().collection('Attendance').add(newAttendance)
  .then((doc) => {
    return 'document' + doc.id + 'created successfully';
  })
  .catch((err) => {
    return 'something went wrong';
    console.error(err);
  });
}

```

**4.18. ábra.** A diákok jelenlétének beírása

Itt először ellenőrzésre kerül az, hogy a diáknak ahhoz az órához van-e már beírva jelenléte, és ha nincs, akkor tranzakciókezeléssel megoldva íródik be a diáknak a jelenlét.

A token szerint kitörlöm a QrCodes táblából azt a dokumentet, ahol a token megegyezik, és csak ez után írom be a jelenlétet a diáknak, ezzel kivédve azt, hogy egy QR kódot akár két diák is használjon.

Ha a diák egy olyan QR kódot szkennel le, ami létezik a QR codes táblában, akkor megkeresi a rendszer, hogy az milyen tantárgyhoz tartozik, és ezekkel a információkkal létrehozza számára a jelenlétet az Attendence táblában.

```
function checkDistance(webLat,webLong,appLat,appLong){  
    let a = Math.abs(webLat - appLat);  
    let b = Math.abs(webLong - appLong);  
    a=a*1000;  
    b=b*1000;  
    // console.log("a: "+a+"b: "+b);  
    if ((a<2) && (b<2)){  
        return true;  
    }  
    return false;  
}
```

4.19. ábra. A két eszköz helyzetének összehasonlítása

A két eszköz helyzete is összehasonlításra kerül, és ha távolabb vannak egymástól, mint 20 méter, akkor nem írja be a jelenlétet a rendszer a diáknak. (Azért van ez a 20 méteres intervallum véve, mert nem lehetünk biztosak benne, hogy a mérések teljesen pontosak.)

## 5. fejezet

# Használt technológiák

### 5.1. React

[4]

A React egy rendkívül népszerű és hatékony JavaScript keretrendszer, amelyet a weboldalak és alkalmazások fejlesztésére használnak.

A React a komponensek felhasználására épül. Ez azt jelenti, hogy az alkalmazást vagy weboldalt kisebb, önállóan működő komponensekre bonthatjuk, amelyek könnyen újrahasznosíthatók és újra felhasználhatók más részekben is. Ez nagyban segíti a kód olvashatóságát, karbantarthatóságát és skálázhatóságát.

Emellett egy virtuális DOM-ot használ, amely a böngésző saját DOM-jával szinkronizálódik. Ez azt jelenti, hogy a React csak azokat a komponenseket frissíti, amelyekben történtek változások, és csak ezeket rendereli újra a böngészőben. Ez jelentős teljesítménynövekedést eredményez, mivel a virtuális DOM hatékonyan kezeli a változásokat és minimalizálja a tényleges DOM-manipulációkat.

A React lehetővé teszi a közvetlen irányítást a felhasználói felület felett a komponensek állapotának kezelésével. Az alkalmazás vagy weboldal állapota változhat, és a React egyszerűvé teszi a változások kezelését és a megjelenítés frissítését az alkalmazásban. Ez kényelmes és hatékony módon teszi lehetővé az interaktív és dinamikus felhasználói élmények létrehozását.

Jól integrálható más JavaScript könyvtárakkal és keretrendszerrel, így rugalmasan kombinálható más technológiákkal is. Például könnyedén integrálhatjuk a React Router-t a navigációhoz. Emellett számos olyan könyvtár és keretrendszer is létezik, amelyek támogatják a Reactet, például a Chakra UI, amely segítségével könnyen és elegánsan testreszabhatjuk a felhasználói felületet.

Végül pedig számos segédanyagot, dokumentációt, könyvtárat és eszközt találunk, amelyek megkönnyítik a fejlesztést és gyorsítják az időt az eredmények eléréséhez. Emellett rengeteg vállalat és nagy projekt használja a Reactet, így könnyű segítséget találni, ha problémába ütközünk.

Választásom a React mellett továbbá hosszú távú előnyöket is hozhat. A React egy stabil és folyamatosan fejlődő keretrendszer, amelyet a Facebook támogat. Ez azt jelenti, hogy a Reactet frissítik és javítják, valamint új funkciókat is hozzáadnak a rendszerhez. Ez biztosítja, hogy weboldalam minden naprakész legyen és kihasználja az újításokat és fejlesztéseket.

Összességében azért választottam ezt a web frontend keretrendszeret, mert a React egy hatékony, rugalmas és jól támogatott keretrendszer, amely lehetővé teszi a gyors és hatékony weboldal fejlesztést.

## 5.2. Vite

[5]

A Vite egy gyors, korszerű, eszközöket nyújtó fejlesztői környezet (build tool), amely a React projektekhez hasznos lehet, mert számos előnnyel jár a hagyományos build toolokhoz képest.

Gyorsabb fejlesztési folyamatot biztosít, olyan funkciókat nyújt, amelyek javítják a webalkalmazás teljesítményét. A Vite beépített támogatást nyújt az aszinkron betöltéshez, amely segít csökkenteni az oldal betöltési időt, és így az alkalmazások gyorsabbak lehetnek.

Ezen kívül elősegíti a jobb kódminőséget, mivel számos szabályt és ellenőrzést használ a kód minőségének javítása érdekében. Például a Vite figyelmeztetéseket ad, ha az alkalmazásban szerepel egy nem használt modul, ami segít az alkalmazás méretének csökkenésében.

Végül, de nem utolsó sorban nagyon rugalmas és konfigurálható. Az alkalmazás különböző részeit, például a szövegeket, képeket, stílusokat, külön-külön lehet konfigurálni. Ez lehetővé teszi, hogy az alkalmazás mérete és teljesítménye jobban ellenőrzés alatt tartható legyen.

## 5.3. Chakra UI

[6]

A Chakra UI komponenskönyvtárat azért választottam a React projekt számára, mert számos előnyös tulajdonsággal rendelkezik, amelyek segítik a hatékony és könnyű felhasználói felület (UI) fejlesztést.

Például könnyen tanulható és alkalmazható. Az intuitív API és a jól dokumentált komponensek lehetővé teszik a gyors és hatékony fejlesztést. A konfiguráció minimális, és a beállításokat egyszerűen testreszabhatjuk az alkalmazásunk igényei szerint.

Lehetővé teszi a komponensek könnyű testreszabását és stílusozását. A rendszer alapvetően a CSS-in-JS megközelítést használja, és támogatja a tematizálást, így könnyedén alkalmazhatunk globális vagy komponensszintű stílusokat. A komponensek rendelkeznek előre definiált stílusvariációkkal, amelyek segítségével egyszerűen és következetesen hozhatunk létre vonzó felhasználói felületeket.

A Chakra UI egy komponensközpontú könyvtár, amelyben az összes UI elemet különálló komponensek formájában kezelhetjük. Ez a megközelítés nagyban elősegíti a kód újra felhasználhatóságát, a fejlesztés gyorsaságát és a karbantarthatóságát. Emellett a komponensek egyszerűen kombinálhatók és átrendezhetők, ami rugalmasságot és könnyű skálázhatóságot eredményez.

Mindemellett egy teljes körű komponenskönyvtár, amely számos előre elkészített és jól tesztelt UI komponensem kínál, például gombok, űrlapok, modulok stb. Ezek a komponensek nagyban felgyorsítják a fejlesztési folyamatot, mivel nem kell minden elemet

nulláról létrehoznunk. Ezen felül, a komponensek reszponzívak és a hozzájuk tartozó felhasználói interakciók is kezelve vannak, ami segíti a kiváló felhasználói élmény kialakítását.

Végül pedig egy népszerű és aktív fejlesztői közösséggel rendelkezik, ami biztosítja a frissítéseket, hibajavításokat és új fejlesztéseket. A hivatalos dokumentáció és a közösség által készített példák és útmutatók széles választékban segítenek a Chakra UI hatékony használatában.

Ezek az előnyök meggyőzővé tették számomra, hogy a Chakra UI-t válasszam a React projekt számára. Az egyszerű használat, a rugalmas stílusozás, a komponens alapú fejlesztés, a készített funkciók és a támogató közösség mind hozzájárulnak ahhoz, hogy hatékonyan és gyorsan fejleszthessek felhasználóbarát és vonzó felületeket.

## 5.4. React Native

[7]

A React Native-t választottam az Android platformra történő mobilalkalmazás fejlesztéséhez, mert számos előnyvel rendelkezik, amelyek segítenek hatékony és rugalmas alkalmazások létrehozásában.

Az egyik fő előnye a React Native-nek, hogy lehetővé teszi az alkalmazások keresztpatformos fejlesztését. Ez azt jelenti, hogy ugyanazt a kódot használhatjuk az Android és az iOS alkalmazásokhoz, minimalizálva ezzel a fejlesztési időt és erőforrásokat. Ez különösen előnyös, ha több platformra szeretnénk kiadni az alkalmazást, mert csak egyszer kell fejlesztenünk és karbantartanunk a kódot. Így később multiplatform támogatottá lehető az alkalmazás.

Gyors és hatékony fejlesztési folyamatot biztosít. A React alapú architektúra lehetővé teszi a komponens alapú fejlesztést és az újrafelhasználható kódok használatát. Ezáltal gyorsabban és hatékonyabban hozhatunk létre új funkciókat és interakciókat az alkalmazásban.

Lehetővé teszi a natív felhasználói élmény kialakítását. Az alkalmazás közvetlenül a natív platform komponenseit és funkcióit használja, így az alkalmazás sima és gyors lesz, és a felhasználók számára természetes érzetet nyújt. Emellett a React Native lehetővé teszi a platformspecifikus optimalizációkat és testreszabásokat is.

A React Native egy nagyon népszerű fejlesztői platform, így egy aktív és kiterjedt közösséggel rendelkezik. Ez azt jelenti, hogy rengeteg dokumentáció, példa és könyvtár áll rendelkezésre, valamint sok fórum és közösségi platform, ahol kérdezhetünk és kapcsolatba léphetünk más fejlesztőkkel. Ez a támogatás nagy segítséget nyújt a problémák megoldásában és az új ismeretek megszerzésében.

Ezen kívül nagy skálázhatóságot biztosít, így az alkalmazásunkat könnyedén bővíthetjük és alkalmazkodhatunk a növekvő felhasználói igényekhez. Emellett a React Native alkalmazások általában nagyon jó teljesítményt nyújtanak, mivel közvetlenül a natív platform funkcióit használják, és a React Virtual DOM rendszere hatékonyan kezeli az UI frissítéseket.

Ezen előnyök alapján választottam a React Native-et az Android alkalmazás implementálásához. A keresztpatformos kompatibilitás, a gyors fejlesztési folyamat, a natív felhasználói élmény, a nagy fejlesztői közösség, valamint a skálázhatóság és teljesítmény

mind hozzájárulnak ahhoz, hogy hatékony és sikeres mobilalkalmazásokat hozzak létre az Android platformra.

## 5.5. Firebase

[8]

A Firebase-t választottam a backend és adatbázis megoldásnak számos előnye miatt, amelyek segítenek gyors és hatékony fejlesztést megvalósítani.

A Firebase-t rendkívül könnyű beállítani és elindítani. Az alapvető infrastruktúra, például a szerverek, a konfigurációk és a skálázhatóság, már előre be van állítva, így nem kell időt és erőforrásokat fordítanunk az alaprendszer kiépítésére és karbantartására.

Biztosítja a felhasználókezelés és hitelesítés funkcióit, amelyek segítségével könnyedén kezelhetjük a felhasználói fiókokat, bejelentkezést és hozzáférések kezelését.

A Firebase további funkciókat és szolgáltatásokat kínál, amelyek segíthetnek az alkalmazásunk fejlesztésében és kiegészítésében. Ilyen például a felhőalapú tárolás, amelyet én is használok a projektben, az analitikák és a tesztelési eszközök, amelyek minden hozzájárulnak az alkalmazásunk funkcionalitásának és teljesítményének javításához.

A Firebase Functions-t választottam a projektemhez megoldásképp, mert lehetővé teszi a háttérben futó szerveroldali logika könnyű implementálását és skálázhatóságát. A beépített integráció a Firebase adatbázissal és más szolgáltatásokkal lehetővé teszi az egyszerű adatmanipulációt és valós idejű frissítések kezelését. Emellett a Firebase Functions automatikusan skálázódik és kezeli a terhelést, így hatékonyan és megbízhatóan fut a projektem részeként. [9]

Adatbázisnak a Cloud Firestore-t választottam, mert egy skálázható, valós idejű dokumentum alapú adatbázist nyújt, amely ideális a felhasználói adatok tárolására és kezelésére. Az egyszerű API-k és a valós idejű események lehetővé teszik az adatok könnyű szinkronizálását és frissítését a felhasználók között. A Firestore biztonságos, gyors és megbízható adatbázis megoldást nyújt a projektem számára. [10]

Ezek az előnyök és a Firebase kiterjedt funkciókészlete alapján választottam ezt az adatbázis és egyben teljes backend megoldást.

## 5.6. GitHub

[11]

A GitHub-ot választottam a verziókezelési és kódtárolási platformnak, mert egy kiválló verziókezelési eszköz, amely lehetővé teszi a kódmezők változásainak nyomon követését, git alapú struktúrája révén könnyen kezelhetjük a különböző változtatásokat, visszaállíthatjuk a korábbi verziókat, ágakat hozhatunk létre és összefűsölhetjük a kódváltozásokat, ezáltal hatékonyan tudunk dolgozni a kódon, több fejlesztő együttműködhet, és biztosított a kód biztonsága és integritása.

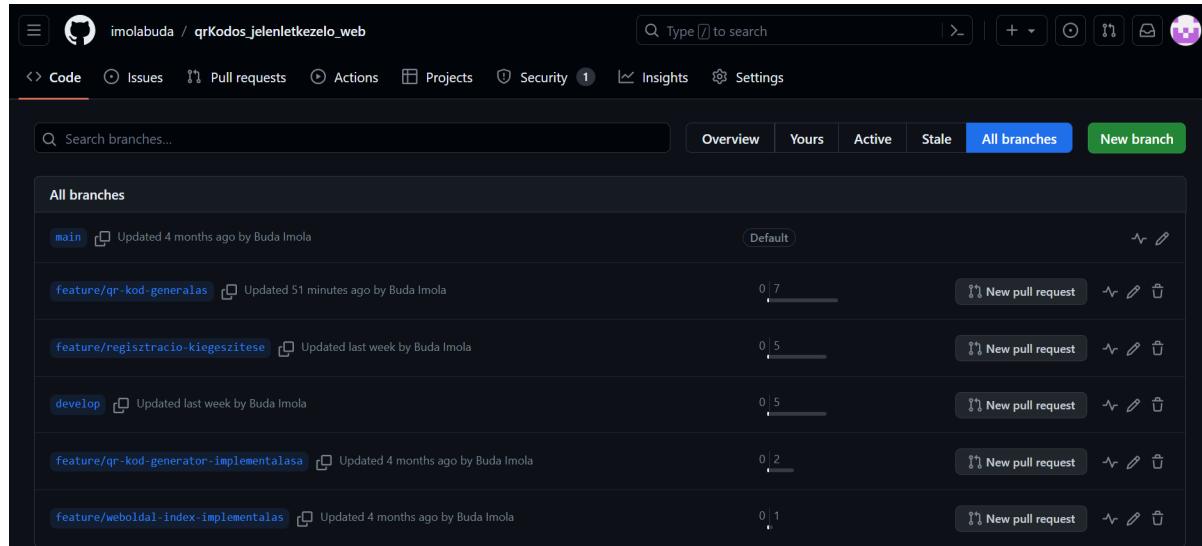
A GitHub egy közösségi alapú platform, amely lehetővé teszi a fejlesztők számára a közös munkát és az együttműködést. Más fejlesztőkkel könnyedén megoszthatjuk a kódunkat, kollaboratívan dolgozhatunk projekteken, áttekinthetjük mások munkáját is és sok nyílt forráskódú projekt található, amelyekből inspirációt nyerhetünk és tanulhatunk.

Ez a verziókezelő platform biztosítja a központi kódtárolást, amely könnyen hozzáférhető és elérhető a fejlesztők számára, minden kódváltozás, dokumentáció és egyéb projekt erőforrás egy helyen van tárolva, így könnyen visszakereshető és kezelhető. Ezek mellett még könnyen integrálható más fejlesztői eszközökkel és folyamatokkal, például CI/CD rendszerekkel vagy projektmenedzsment eszközökkel.

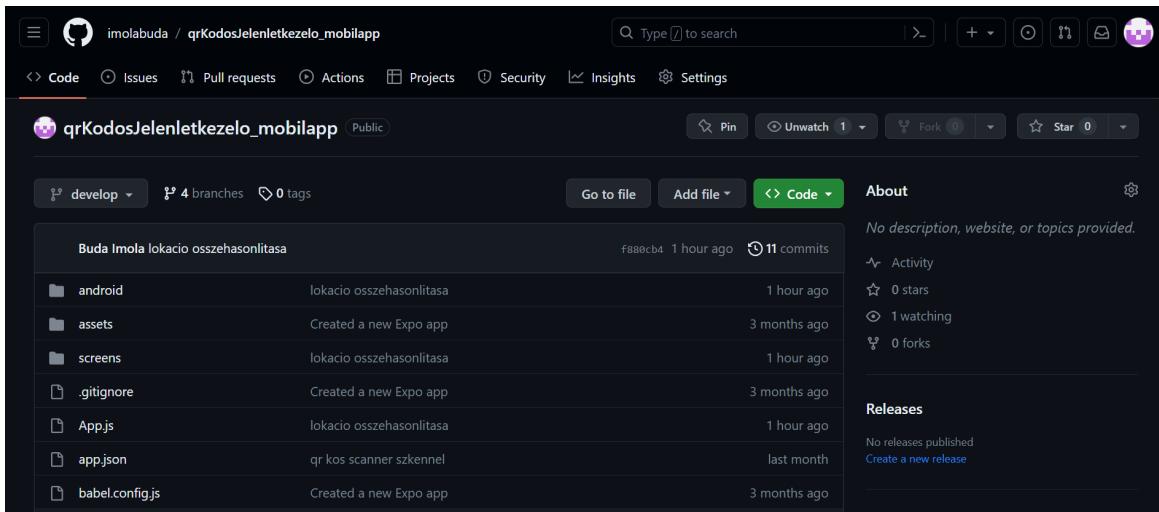
A GitHub lehetővé teszi a problémák és hibák nyomon követését az integrált problémajegyzék funkcióval, fejlesztőként könnyedén jelenthetünk hibákat, nyomon követhetjük a feladatokat és a fejlesztési folyamatot, ami segít a projekt irányításában és a problémák hatékony kezelésében.

A GitHub számos integrációs lehetőséget kínál más szolgáltatásokkal és eszközökkel. Például könnyen csatlakoztatható Continuous Integration és Continuous Deployment (CI/CD) rendszerekhez, automatizált tesztelő eszközökhöz vagy kódvizsgáló szoftverekhez. Ezáltal egyszerűen beállíthatjuk a fejlesztési folyamatot és automatizálhatjuk bizonatos feladatokat.

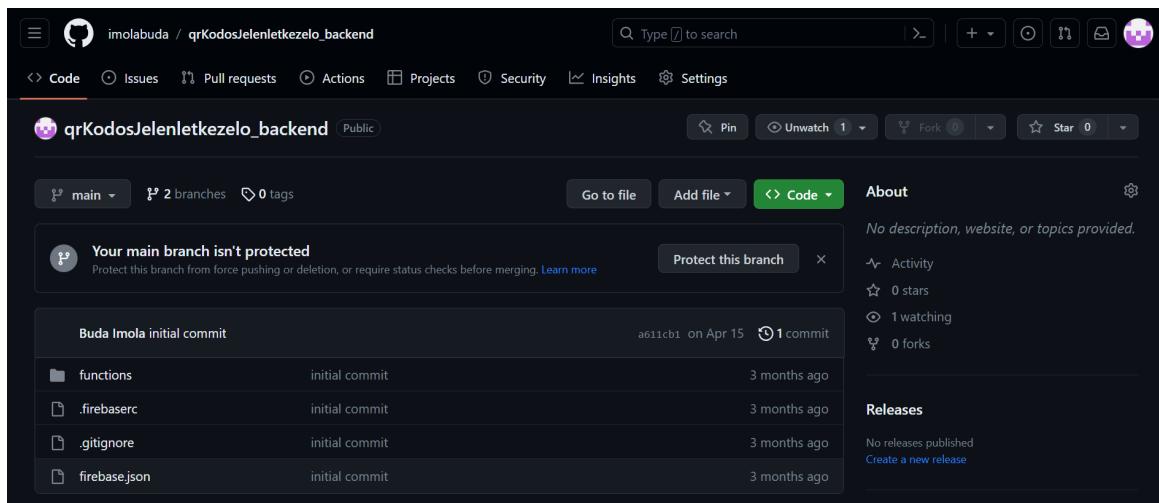
Ezen előnyök és lehetőségek miatt választottam a GitHubot, amely segít hatékonyabban dolgozni, támogatja az együttműköést és nyomon követhetőséget, valamint biztosítja a kód tárolásának és verziókezelésének megbízható keretrendszerét.



**5.1. ábra.** Weboldal frontend GitHub repoja



**5.2. ábra.** Mobilapp frontend GitHub repoja



**5.3. ábra.** Backend GitHub repoja

## 5.7. Trello

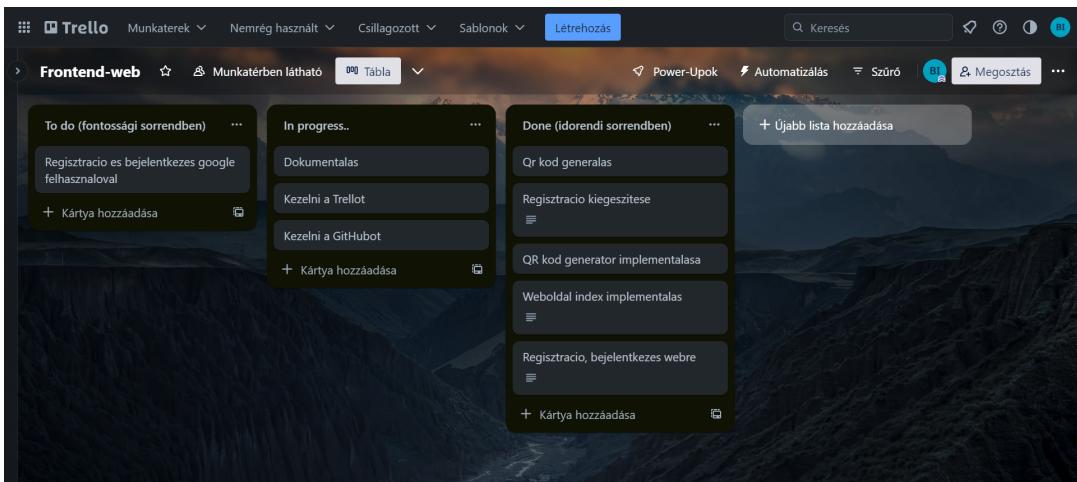
[12]

A Trello-t választottam a feladatkezelési eszköznek, mert nagyon könnyen kezelhető és felhasználóbarát felülettel rendelkezik, a feladatakat táblákon és kártyákon lehet rendezni, amelyek könnyen mozgathatók és módosíthatók. A feladatok hozzáadása, cím-kézése és határidők beállítása egyszerű és gyors.

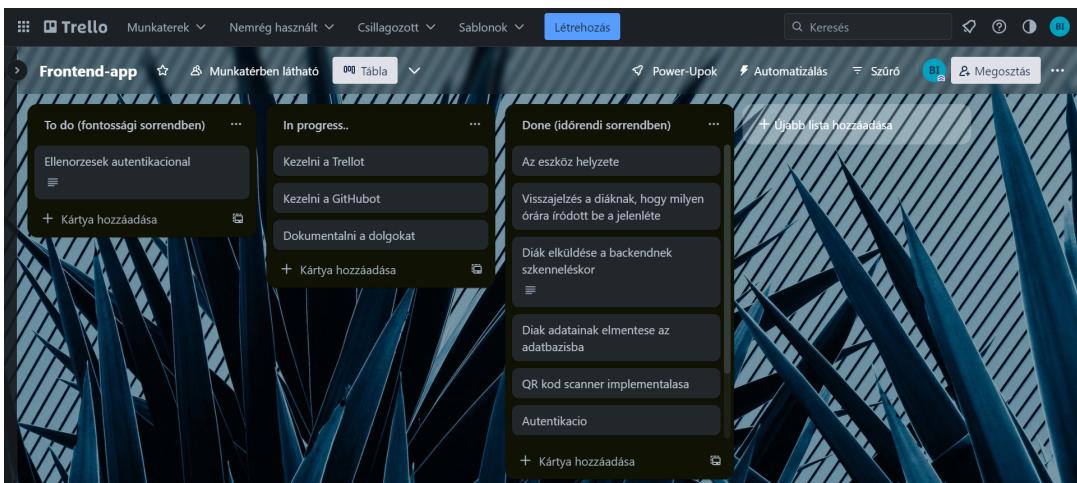
Ez a felület segít átláthatóvá tenni a feladatokat és a projektet. A táblák és kártyák rendszere könnyen érhető és áttekinthető, így könnyebb lehet követni az aktuális állapotokat és priorizálni a teendőket. Emellett lehetővé teszi a csapatmunkát és az együttműködést, mert különböző felhasználók hozzáférhetnek a táblákhoz és kártyákhoz, így könnyen megoszthatják a feladatokat és követhetik a munkafolyamatot, illetve a kommentek és értesítések segítenek a kommunikációban és a feladatokról való folyamatos információáramlásban.

A Trello rugalmasan testre szabható a projekt igényeinek megfelelően, mert a táblák, listák és kártyák saját preferenciák szerint alakíthatók ki, címkékkel, határidőkkel, leírás-sal és más adatokkal egészíthetők ki, mindenellett bővítményeket is kínál, amik további funkcionálitást nyújtanak.

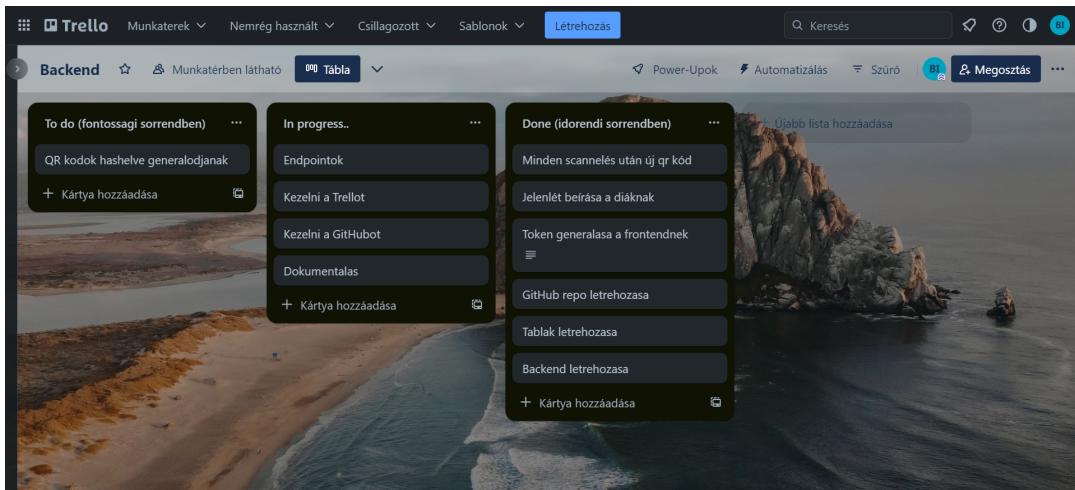
Ezen alapján választottam a Trello-t, ami segít a feladatok hatékony kezelésében, a munkafolyamatok átláthatóvá tételeben, a felhasználóbarát felület, a testreszabhatóság lehetővé teszi, hogy a Trello-t a saját igényeimre szabjam, és hatékonyan használjam a projektem nyomon követésére.



5.4. ábra. Weboldal frontend trello táblája



5.5. ábra. Mobilapp frontend trello táblája



**5.6. ábra.** Backend trello táblája

## 5.8. LaTeX

A LaTeX egy dokumentumfeldolgozási rendszer, amelyet gyakran használnak tudományos és matematikai dokumentumok, cikkek, könyvek, prezentációk és tudományos jelentések készítésére.

A LaTeX kiválóan alkalmas tudományos dokumentumok írására, mivel könnyedén kezelhetők benne a matematikai formulák, szimbólumok és táblázatok, a beépített matematikai módszerek és az egyszerű jegyzetrendszer segít a professzionális és esztétikus tudományos publikációk létrehozásában.

A LaTeX rendkívül népszerű a könyvek és dokumentációk szerkesztésére, mert ebben könnyen kezelhetők a fejezetek, alfejezetek és tartalomjegyzék, valamint a lábjegyzetek, idézetek és képek beillesztése, illetve a professzionális tipográfiai lehetőségek segítenek kiváló minőségű dokumentumok készítésében.

A LaTeX rendkívül rugalmas és erőteljes dokumentumfeldolgozási rendszer, amely lehetővé teszi a professzionális és esztétikus dokumentumok létrehozását. Használata különösen ajánlott tudományos, matematikai és műszaki területeken, ahol pontos formázás és matematikai kifejezések kezelése szükséges.

## 6. fejezet

# A felhasználói felület bemutatása

### 6.1. Weboldal (tanárok alkalmazása)

#### 6.1.1. Regisztráció

The screenshot shows a registration form titled "Regisztráció". It contains the following fields:

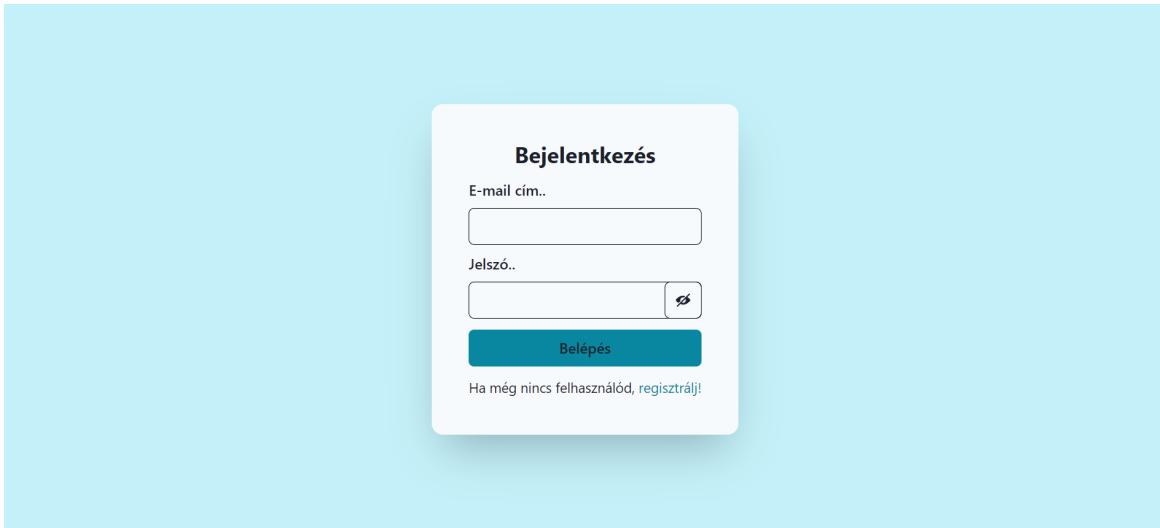
- E-mail cím.. (Email address) - An empty input field.
- Teljes név.. (Full name) - An empty input field.
- Születési dátum.. (Date of birth) - A date input field showing "07/02/2023".
- Jelszó.. (Password) - An empty input field with a visibility toggle icon.
- A large teal-colored "Regisztrálok" (Register) button at the bottom.
- Text at the bottom: "Ha van már felhasználód, lépj be!" (If you already have an account, log in!).

6.1. ábra. Webes regisztráció

The screenshot shows a registration form with a date picker overlay. The date shown is "07/02/2023". The calendar grid for July 2023 is displayed, with the 2nd highlighted in blue. Other dates are shown in a light gray background. Below the calendar, there is a password input field and a "Regisztrálok" button, identical to the one in the previous screenshot. At the bottom of the form, there is the text "Ha van már felhasználód, lépj be!".

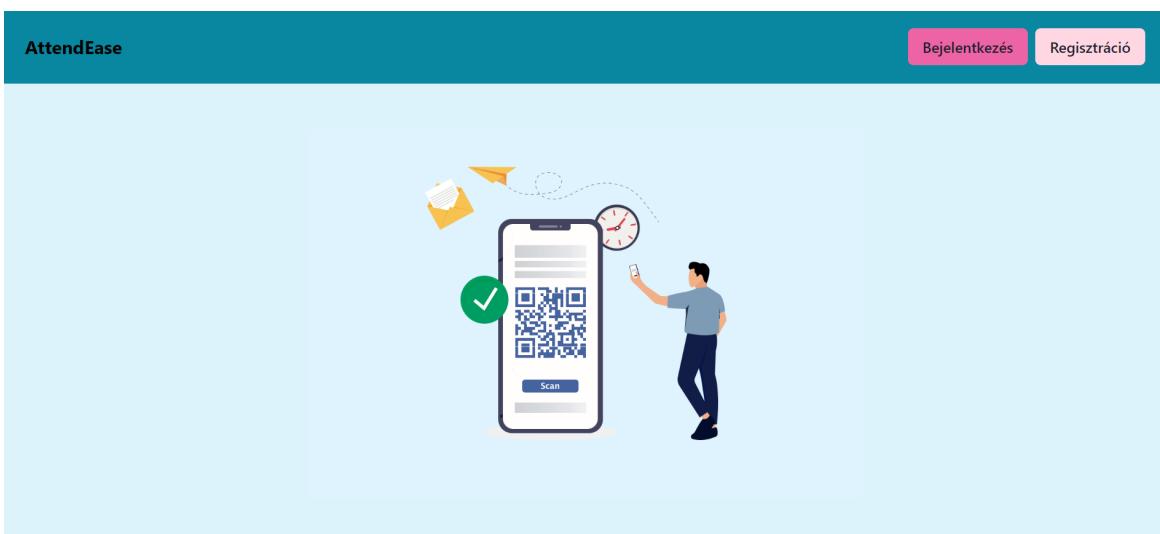
6.2. ábra. Webes regisztráció: születési dátum kiválasztása

### 6.1.2. Bejelentkezés

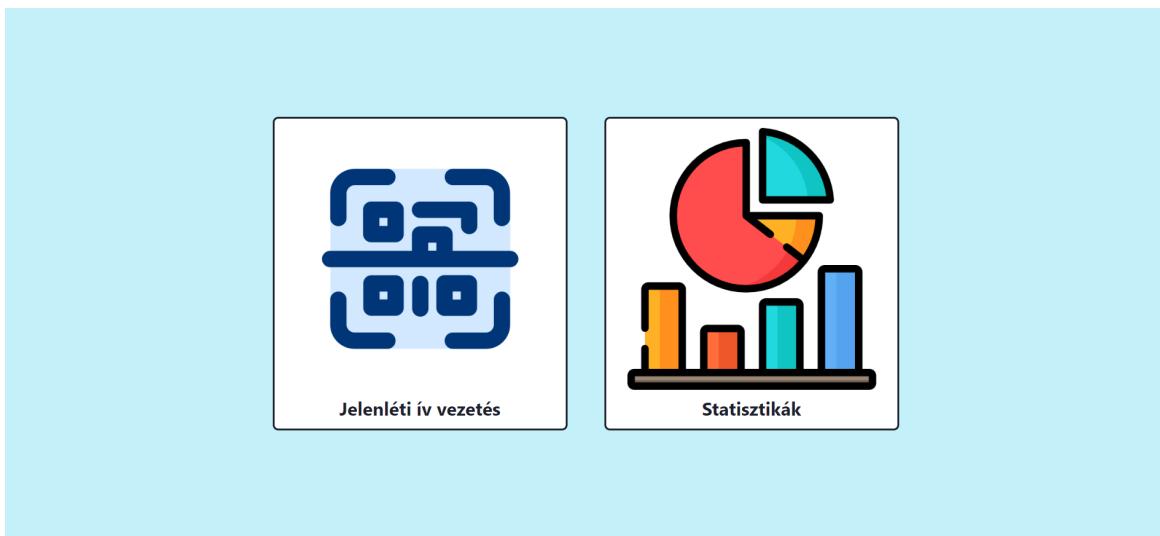


6.3. ábra. Webes bejelentkezés

### 6.1.3. Főoldal



6.4. ábra. Főoldal bejelentkezés előtt



6.5. ábra. Főoldal bejelentkezés után

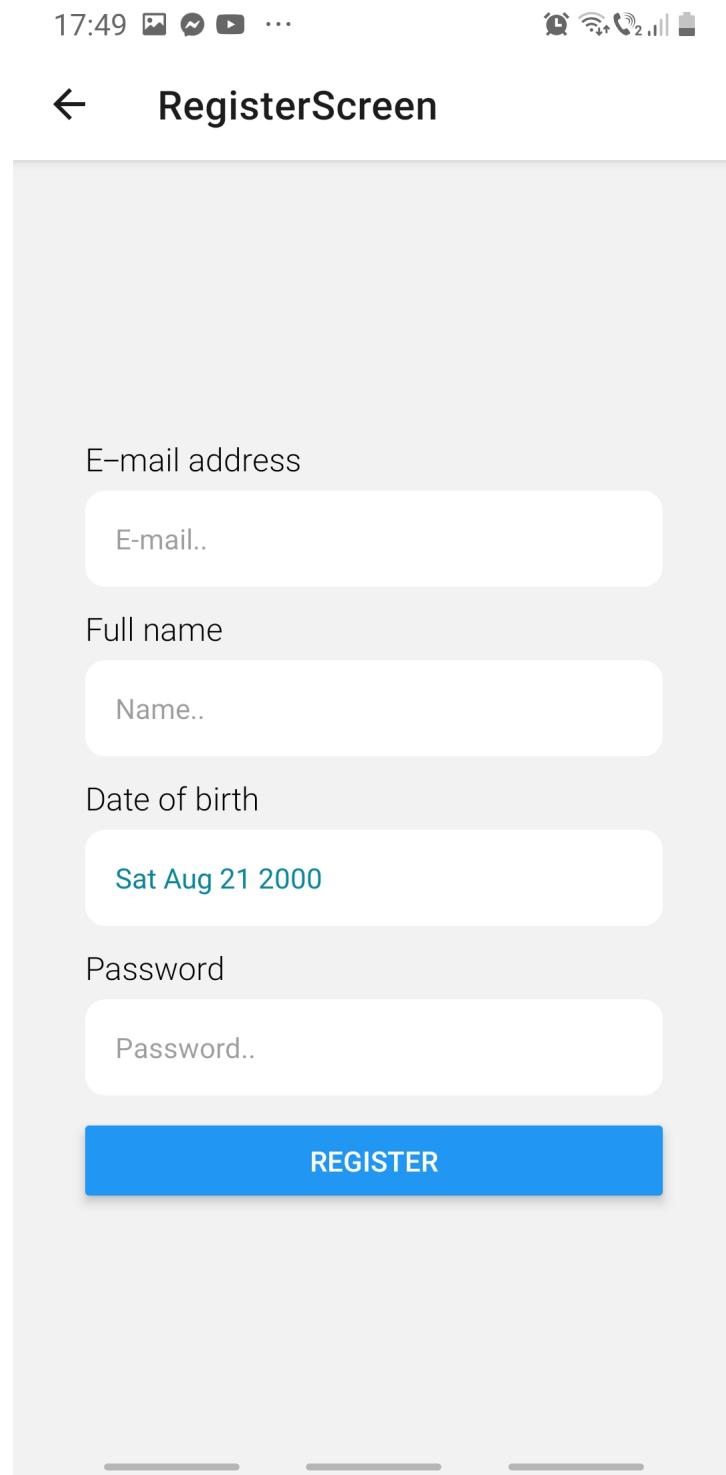
#### 6.1.4. Jelenléti ív vezetés (QR kód generáló)



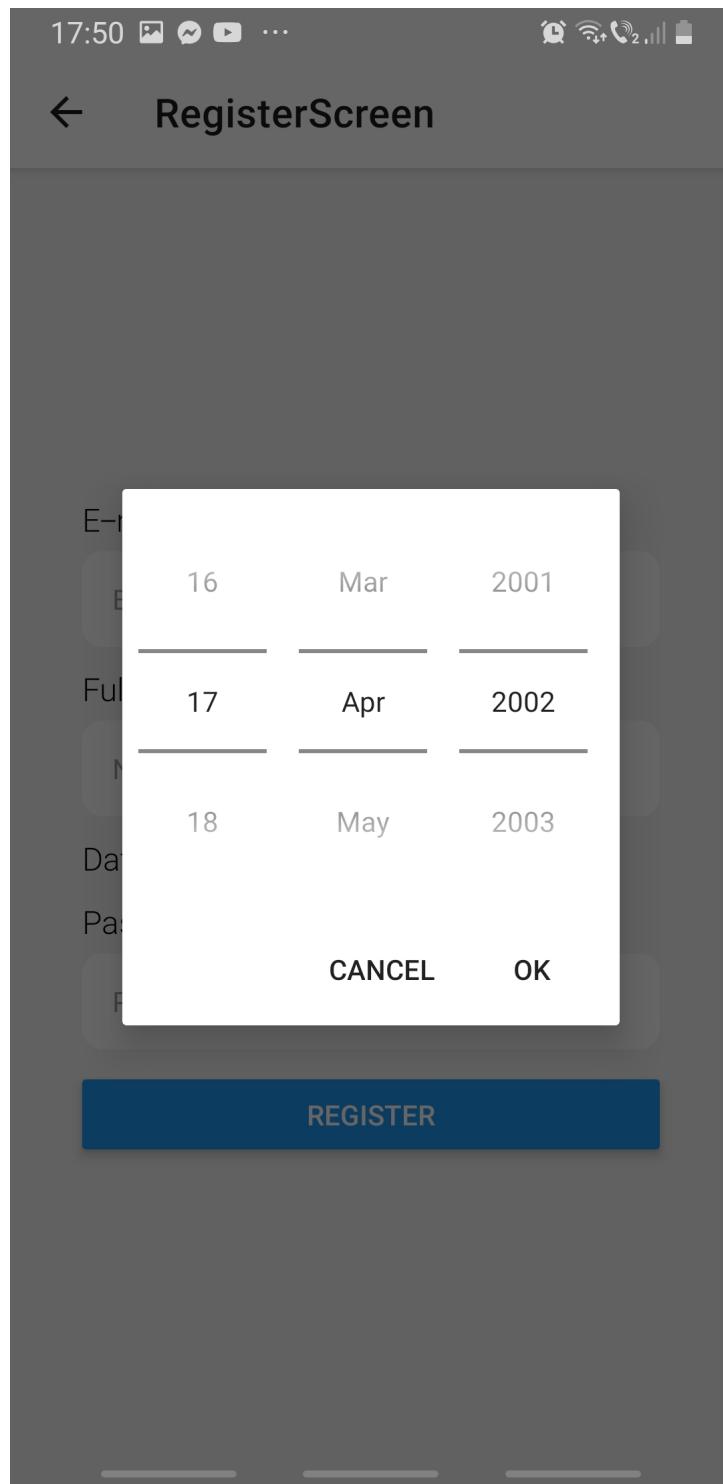
6.6. ábra. Jelenléti ív vezetés (QR kód generáló)

## 6.2. Mobilapp (diákok alkalmazása)

### 6.2.1. Regisztráció

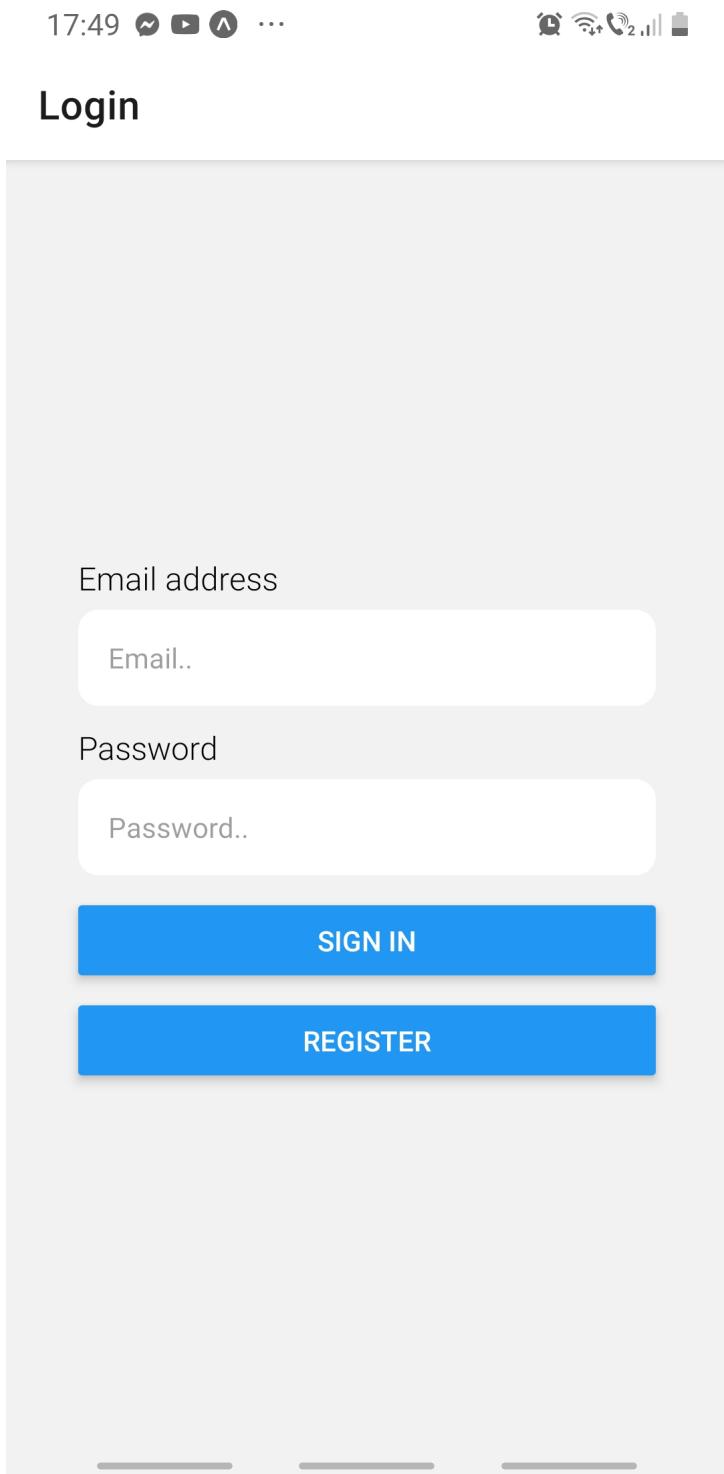


6.7. ábra. Telefonos regisztráció



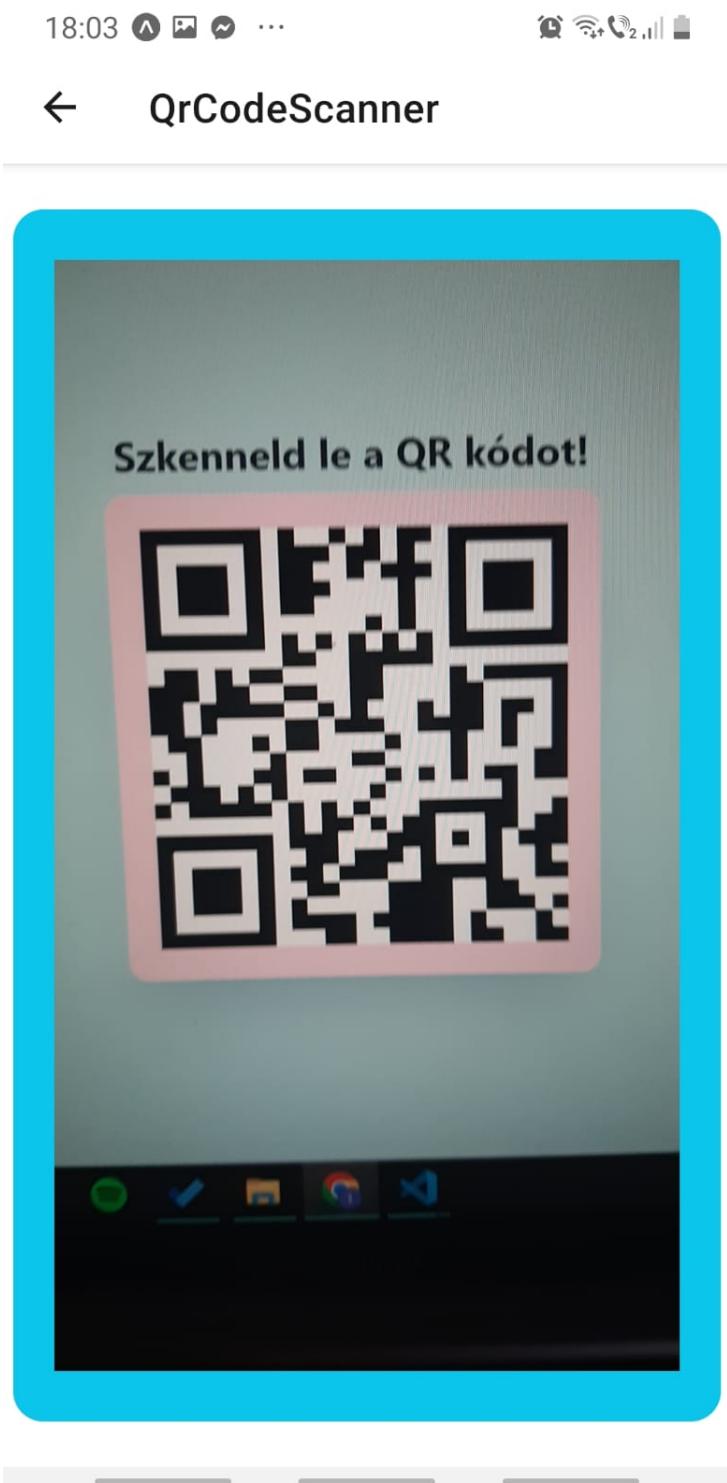
**6.8. ábra.** Telefonos regisztráció: születési dátum kiválasztása

## 6.2.2. Bejelentkezés



6.9. ábra. Telefonos bejelentkezés

### 6.2.3. Jelenlét bevezetése (QR kód olvasó)



6.10. ábra. QR kód olvasó

#### 6.2.4. Sikeres jelenlét bevezetése



**6.11. ábra.** Sikeres jelenlét bevezetve

## **7. fejezet**

# **Továbbfejlesztési lehetőségek**

A jelenlét nyilvántartó rendszer továbbfejlesztése számos lehetőséget kínál a funkcionalitás és a felhasználói élmény javítására, az én ötleteim ezekre a következők:

### **7.1. Control panel**

Egy control panel létrehozása egy fejlesztési irány lehet, amely lehetővé teszi a tanároknak, hogy könnyen és gyorsan rögzíthessék a kurzusaikat és az óráikat a rendszerben.

### **7.2. Hashelt tokenek (blockchain)**

Az adatbiztonság további erősítése érdekében érdemes lenne a jelenléthez tartozó tokeneket hashelве generálni, mert ez megakadályozná a tokenek illetéktelen felhasználását vagy manipulálását, tovább növelve a rendszer megbízhatóságát, és még inkább kizárra ezzel a csalás lehetőségét.

### **7.3. Jelenléti adatok exportálása**

A jelenleti adatok exportálása révén a tanárok könnyedén hozzáférhetnek a jelenléti adatokhoz és vizsgálhatják az időbeli tendenciákat, amelyek segíthetnek a diákok jelenlétének nyomon követésében. Ezáltal a tanárok hatékonyabban kezelhetik az órákat, valamint a diákokkal való kapcsolatot.

### **7.4. Multiplatform támogatottság**

A multiplatform támogatottság szintén fontos aspektus lehet a rendszer fejlesztésében. Azzal, hogy kiterjesztjük a telefonos alkalmazást más platformokra, mint például iOS-re, még több felhasználó számára lenne elérhetővé téve a jelenlét nyilvántartás lehetősége, és ez nagyon fontos lenne, mivel sokan használnak iOS vagy más operációs rendszert támogató telefont a diákok közül is.

## **7.5. Több intézmény számára elérhetőség**

A rendszer kiterjesztése azzal a céllal, hogy több intézmény is használhassa, nagyobb skálában is alkalmazhatóvá tenné a megoldást, ehhez rugalmas adminisztrációs funkciókra lenne szükség.

## **7.6. Diák alkalmazás kibővítése**

Sokat javítana a működésen és a minőségen, ha a diákok alkalmazása kibővülne egy olyan funkcionalitással, ami által a diákok meg tudják nézni a saját jelenléteiket, hogy milyen órán hányszor vettek részt és esetleg küldene az alkalmazás egy értesítést, ha mondjuk már csak 1 hiányzás megengedett egy adott órán.

## **7.7. Biometrikus azonosítás**

Az előrehaladott biztonsági funkciók közé tartozik a biometrikus azonosítás bevezetése pár másodperccel a szkennelés lehetősége előtt, amely lehetővé tenné a diákok számára, hogy ujjlenyomat vagy arcfelismerés alapján igazolják magukat. Ez tovább növelné a biztonságot és kényelmet, mert így gyorsabban be tudnának lépni az alkalmazásba, és így egyértelmű lenne az, hogy tényleg az adott személy szkennelte le a qr kódot és nem egy más személynek adta oda esetleg a telefonját.

## **7.8. Órapótlás lehetősége**

Ha egy tanárnak éppen nincs órája az órarendben az adott időben, akkor is lehetőséget kell adni számára, hogy órát tartson, azaz egy órát bepótoljon. Erre az lenne a megoldás, hogy ha nincs órája éppen a tanárnak, akkor tudja kiválasztani, hogy milyen órát szeretne pótolni és arra a diákok ugyanúgy "jelentkezni" tudjanak a token szkennelése által.

## **7.9. Regisztráció és belépés Google fiókkal**

Egy másik továbbfejlesztési lehetőség a Google-fiókkal való bejelentkezés integrálása lenne a rendszerbe, ez lehetővé tenné a felhasználók számára, hogy egyszerűen és biztonságosan jelentkezzenek be a Google-fiókjukkal a jelenlét nyilvántartó rendszerbe. A Google Account bejelentkezés segítségével a felhasználók nem kellene újabb felhasználói fiókot hozzanak létre, és az azonosítás folyamata egyszerűbbé és gyorsabbá válna.

## **7.10. A felhasználói felület javítása**

A UI és annak design-jának fejlesztése szintén fontos aspektus lehet a felhasználói élmény javítása érdekében, mert a rendszer felhasználói felületének modernizálása és intuitívabbá tétele segíthet a felhasználóknak könnyebben és gyorsabban navigálni a rendszerben, növelte ezzel az átláthatóságot, illetve a felhasználói elégedettséget.

# Összefoglaló

A dolgozatom célja egy biztonságos jelenlét nyilvántartó rendszer kidolgozása volt egy egyetem számára. A kifejlesztett rendszer lehetővé teszi a diákok jelenlétének nyomon követését egy QR kód alapú megoldással, amelyet a tanár egy digitális eszközéről a diákok a telefonjukon lévő alkalmazás segítségével olvas be, ezen keresztül az adott diák jelenléte az adott órára rögzítésre kerül. A tanár weboldala generálja az egyedi, egyszer használható QR kódot, a diák alkalmazása pedig lehetővé teszi, hogy leolvassa azt, ezáltal beírva a diáknak a jelenlétet a megfelelő órára.

Az alkalmazás működése során a rendszer figyeli a két eszköz távolságát egymástól, és ennek alapján ellenőrzi a jelenlét hitelességét, ez a megközelítés hatékonyan kivédi a potenciális csalásokat, mivel a diáknak minden alkalommal új QR kód generálódik, és az eszközök közötti távolság mérése valós időben történik.

Részletesen dokumentáltam a rendszer felhasználói követelményeit, amelyek a diákok és tanárok igényeit egyaránt figyelembe veszik. Emellett bemutatásra kerülnek a rendszer funkcionális követelményei, amelyek a jelenlét rögzítéséhez, a QR kódok generálásához, illetve az adatok kezeléséhez kapcsolódnak.

A rendszer implementációja során a React Native keretrendszert választottam a telefonos alkalmazás fejlesztéséhez, amely lehetővé teszi az Android platformra történő kódírást, a weboldal fejlesztéséhez frontenden React keretrendszert választottam. A tervezéshez és a feladatkezeléshez a Trello nevű feladatkezelő rendszert választottam, ami nagy segítséget nyújt a projekt feladatak strukturálásában, a verziókezeléshez pedig a GitHubot használtam.

A rendszer backendjeként és adatbázisként a Firebase szolgáltatást választottam, amely egy skálázható és megbízható felhőalapú platform. A Firebase Functions lehetővé teszi a háttérben futó szerveroldali logika implementálását, míg a Firebase Firestore dokumentum alapú adatbázis biztosítja az adatok tárolását és kezelését.

A dolgozat részletesen bemutatja az alkalmazás architektúráját, az átjászthatóság elkerülését, a komponensek működését, az adatok áramlását a rendszerben, illetve bemutatja a diák alkalmazás és a tanár alkalmazás felhasználói felületeit. Emellett kiemeli a választott technológiák előnyeit, mint például a könnyű hordozhatóság, a rugalmasság és a fejlesztési gyorsaság.

A végén összegzésre kerülnek a rendszer által nyújtott előnyök, korlátok és javaslatok a rendszer jövőbeni továbbfejlesztésére, optimalizálására.

Frontend web GitHub repo: [https://github.com/imolabuda/qrKodos\\_jelenletkezelo\\_web](https://github.com/imolabuda/qrKodos_jelenletkezelo_web)  
Frontend mobilapp GitHub repo: [https://github.com/imolabuda/qrKodosJelenletkezelo\\_mobilapp](https://github.com/imolabuda/qrKodosJelenletkezelo_mobilapp)  
Backend GitHub repo: [https://github.com/imolabuda/qrKodosJelenletkezelo\\_backend](https://github.com/imolabuda/qrKodosJelenletkezelo_backend)

# Köszönetnyilvánítás

Szeretnék köszönetet mondani a vezetőtanáromnak, Dr. Iclánzan Dávidnak az odaadó és támogató munkájáért és azért, hogy mindig számíthattam a türelmes és kedves segítségére.

Köszönöm a tanáraimnak a befektetett munkát, hogy ennyit tanulhattam tőlük, hogy példát mutattak úgy a tanuláshoz és munkához, mint az élethez is.

# Ábrák jegyzéke

2.1. QR kódok felépítése . . . . .	13
2.2. QR kódok felépítése . . . . .	13
2.3. QR kódos beléptető rendszerek . . . . .	14
3.1. Use Case diagram . . . . .	16
4.1. Architektúra . . . . .	18
4.2. Regisztrációk szekvenciadiagramja . . . . .	21
4.3. Jelenlét bekerülésének szekvenciadiagramja . . . . .	22
4.4. Tanár regisztráció . . . . .	23
4.5. Tanár adatainak elmentése regisztrációkor . . . . .	23
4.6. Tanár bejelentkezés . . . . .	24
4.7. Qr kód megjelenítése . . . . .	24
4.8. Qr kód generálása . . . . .	25
4.9. Tanár eszközének a helyének lekérése . . . . .	26
4.10. Diák regisztráció . . . . .	27
4.11. Diák adatainak elmentése regisztráció során . . . . .	27
4.12. Diák bejelentkezés . . . . .	28
4.13. Diák kijelentkezés . . . . .	28
4.14. Kamera hozzáférési engedély kérése . . . . .	29
4.15. Helymeghatározás hozzáférési engedély kérése . . . . .	29
4.16. QR kód szkennelés kezelése . . . . .	30
4.17. A diák jelenlétének kezelése . . . . .	31
4.18. A diák jelenlétének beírása . . . . .	32
4.19. A két eszköz helyzetének összehasonlítása . . . . .	33
5.1. Weboldal frontend GitHub repoja . . . . .	38
5.2. Mobilapp frontend GitHub repoja . . . . .	39
5.3. Backend GitHub repoja . . . . .	39
5.4. Weboldal frontend trello táblája . . . . .	40
5.5. Mobilapp frontend trello táblája . . . . .	40
5.6. Backend trello táblája . . . . .	41
6.1. Webes regisztráció . . . . .	42
6.2. Webes regisztráció: születési dátum kiválasztása . . . . .	42
6.3. Webes bejelentkezés . . . . .	43
6.4. Főoldal bejelentkezés előtt . . . . .	43
6.5. Főoldal bejelentkezés után . . . . .	44

6.6. Jelenléti ív vezetés (QR kód generáló) . . . . .	44
6.7. Telefonos regisztráció . . . . .	45
6.8. Telefonos regisztráció: születési dátum kiválasztása . . . . .	46
6.9. Telefonos bejelentkezés . . . . .	47
6.10. QR kód olvasó . . . . .	48
6.11. Sikeres jelenlét bevezetve . . . . .	49

# Irodalomjegyzék

- [1] Wikipedia contributors, „Qr code,” [https://en.wikipedia.org/wiki/QR\\_code](https://en.wikipedia.org/wiki/QR_code), hozzáférés dátuma: 2023. június 28.
- [2] Investopedia, „Qr code,” <https://www.investopedia.com/terms/q/quick-response-qr-code.asp>, hozzáférés dátuma: 2023. június 28.
- [3] Kisi, „Qr codes access control,” <https://www.getkisi.com/guides/qr-codes-access-control>, hozzáférés dátuma: 2023. július 2.
- [4] „React,” <https://react.dev/>, hozzáférés dátuma: 2023. július 2.
- [5] „Vite.js,” <https://vitejs.dev/>, hozzáférés dátuma: 2023. június 23.
- [6] „Chakra ui - components,” <https://chakra-ui.com/docs/components>, hozzáférés dátuma: 2023. július 2.
- [7] „React native,” <https://reactnative.dev/>, hozzáférés dátuma: 2023. július 2.
- [8] „Firebase,” <https://firebase.google.com/>, hozzáférés dátuma: 2023. július 2.
- [9] „Google cloud functions and firebase,” <https://cloud.google.com/functions/docs/concepts/functions-and-firebase>, hozzáférés dátuma: 2023. június 20.
- [10] „Firebase firestore,” <https://firebase.google.com/docs/firestore>, hozzáférés dátuma: 2023. június 20.
- [11] HubSpot, „What is github used for?” <https://blog.hubspot.com/website/what-is-github-used-for>, hozzáférés dátuma: 2023. június 14.
- [12] „Trello tour,” <https://trello.com/tour>, hozzáférés dátuma: 2023. június 23.