



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR



SZAKDOLGOZAT

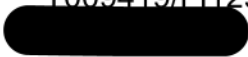
OE-NIK
2024

Hallgató neve:
Hallgató törzskönyvi száma:

Szalai István
T009419/FI12904/N

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertervezés és -fejlesztés Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Szalai István Dávid**
Törzskönyvi száma: T009419/FI12904/N
Neptun kódja: 

A dolgozat címe:

**Angular-ban létrehozott hallgatói fórum OTP-alapú biztonságos
belépéssel**

Student forum created in Angular with OTP-based secure access

Intézményi konzulens: Sipos Miklós László
Külső konzulens:

Beadási határidő: 2024. december 15.

A záróvizsga tárgyai: Szoftvertechnológia és grafikus felhasználói interfész
tervezése
Szoftvertervezés és -fejlesztés specializáció

A feladat

A feladat egy reszponzív és felhasználóbarát webalkalmazás létrehozása Angular és ASP.NET eszközök segítségével, ahol diákok tudnak tárgyakat, képzéseket véleményezni, valamint pontozni bizonyos előre meghatározott szempontok alapján. Emellett hozzon létre egy mobilalkalmazást a bejelentkezés hitelesítésre QR kód és kétféle hitelesítéssel. A rendszer biztosítson szokásos felhasználói profiltípust és ehhez tartozó regisztrációs felületet, illetve működjön egy fórumként a hallgatók számára, ahol hasznos információkat tudnak cserélni. Legyen lehetőség új tárgyakat felvenni az adatbázisba, amit egy adminisztrátor el kell fogadjon, mielőtt mindenki számára látható lenne. Az adatbázisban szereplő tárgyak között legyen lehetőség szűrni név, kredit, értékelések és egyéb paraméterek alapján. A bejelentkezés a felhasználó név és jelszó beírása után egy QR kód segítségével működik, telefonos applikációval történő leolvasás után, a rendszer azonnal beenged a felületre. Amennyiben a QR kód bármilyen okból nem működik, van lehetőségünk egy 5 karakter hosszú kódot generálni és ezt a webalkalmazás felületén megadni, ezután a rendszer automatikusan továbbít a felületre.

A dolgozatnak tartalmaznia kell:

- a megoldandó feladat pontos leírását és részletes elemzését,
- magas szintű szerkezeti mintát az alkalmazásról,
- az egyes funkciók megvalósításához felhasználható lehetséges technológiákat,
- az elkészített rendszer tervét, blokkdiagramját,
- a megvalósítás menetét, felmerülő problémákat és az azokra adott megoldást,
- a tesztelési tervet, a tesztelés módszertanát és a tesztelés eredményeit,
- az elkészült szoftver felhasználási lehetőségeit,
- az eredmények bemutatását és értékelését a teljes munkára vonatkozóan.



Vámossy Zoltán
.....
Dr. habil. Vámossy Zoltán
intézetigazgató

A szakdolgozat elévülésének határideje: **2026. december 15.**
(ÓE HKR 54.§ (10) bekezdés szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

Sipos L.
.....
intézményi konzulens

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2024. december 13.

Sealai István.....
hallgató aláírása



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve:

SZALAI ISTVÁN DÁVID

Neptun Kód:

Tagozat:

NAPPALI TAGOZAT

Telefon:

Levelezési cím (pl.: lakcím):

Szakdolgozat címe magyarul:

Angular-ban létrehozott hallgatói fórum OTP-alapú biztonságos belépéssel

Szakdolgozat címe angolul:

Student forum created in Angular with OTP-based secure access

Intézményi konzulens:

SIPOS MIKLÓS LÁSZLÓ

Külső konzulens:

-

Alk.	Dátum	Tartalom	Aláírás
1.	2024.02.19.	Félév teendőinek, menetrendjének és mérföldköveinek ismertetése.	SS
2.	2024.03.11.	Hasonló rendszerek elemzése.	SS
3.	2024.04.15.	Irodalomkutatás, annak összegzése valamint konzekvenciák meghatározása.	SS
4.	2024.05.06.	Tervezés, követelmények specifikációja. Véglegesítés. Leadással kapcsolatos teendők egyeztetése.	SS

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. tantárgy követelményét teljesítette, beszámolóra bocsátható.

A konzulens által javasolt érdemjegy: 3



Intézményi konzulens

Budapest, 2024.05.17.



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve:

SZALAI ISTVÁN DÁVID

Neptun Kód:



Tagozat:

NAPPALI

Telefon:

Levelezési cím (pl.: lakcím):



Szakdolgozat címe magyarul:

ANGULAR-BAN LÉTREHOZOTT HALLGATÓI FÓRUM OTP-ALAPÚ BIZTONSÁGOS
BELÉPÉSSSEL

Szakdolgozat címe angolul:

STUDENT FORUM CREATED IN ANGULAR WITH OTP-BASED SECURE ACCESS

Intézményi konzulens:

SIPOS MIKLÓS LÁSZLÓ

Külső konzulens:

-

Alk.	Dátum	Tartalom	Aláírás
1.	2024.09.29.	Fejlesztés tervezett menetének revíziója, szakmai egyeztetés, fejlesztési teendők megkezdése.	
2.	2024.10.20.	Fejlesztés állapotának és haladásának áttekintése. Technikai problémák és módosítások egyeztetése.	
3.	2024.11.17.	Felmerülő problémák egyeztetése és megoldása. Tesztelési fázis megindítása.	
4.	2024.12.01.	Dokumentáció véglegesítése. Leadással kapcsolatos teendők egyeztetése.	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat II. tantárgy követelményét teljesítette, védelemre bocsájtható.

A konzulens által javasolt érdemjegy: 5

Intézményi konzulens

Budapest, 2024. 12. 06.

Absztrakt

Szakedolgozatom egy olyan hallgató fórum létrehozása, amelynek a célja a hatékony információcsere a diákok között. Webalkalmazásomban kurzusokat és tárgyakat lehet majd értékelni előre meghatározott szempontok alapján.

Dolgozatom jelen álláspont szerint regisztrációs felülettel nem rendelkezik, csak belépési felülettel, ahol a megfelelően beküldött belépési adatok után egy megerősítő oldal jelenik meg, amelyen egy belépési QR kód található és egy gomb, amennyiben nem QR kóddal szeretnénk belépni, hanem egy 5 számjegyű kóddal. A QR kódnál maradvá, ezt egy telefonos alkalmazás segítségével leolvasás után, a webalkalmazás beenged minket az oldalra, a hitelesítés megtörtént. Telefonos applikációmban, lesz lehetőségünk egy 5 számjegyű kódot generálni, aminek megadásával az oldal hasonlóképpen beenged minket.

Belépés után a főoldalon találjuk magunkat, ahol felsorolja azokat a tárgyakat és kurzusokat, amelyeken a legtöbb értékelés található. Rendelkezésünkre áll egy kereső mező is, amennyiben konkrét tárgyra, vagy kurzusra szeretnénk rákeresni és értékelni azt, vagy éppen mások értékelései érdekelnek minket.

Abstract

My thesis revolves around creating a student forum aimed at facilitating effective information exchange among students. Within my web application, courses and subjects can be evaluated based on predefined criteria.

As of the current stage, my thesis does not feature a registration interface; it only includes a login interface. Upon submitting the appropriate login credentials, users are directed to a confirmation page displaying a login QR code and a button. Alternatively, users can opt for a 5-digit code instead of the QR code for login. Continuing with the QR code, after scanning it using a mobile application, the web application grants access to the page, confirming authentication. In my mobile application, users will have the ability to generate a 5-digit code for login, providing similar access to the site.

After logging in, users find themselves on the homepage, listing the subjects and courses with the most ratings. A search field is also available for specific subject or course searches, enabling users to rate them or explore others rates.

TARTALOMJEGYZÉK

1. Bevezetés	1
2. Irodalomkutatás	2
2.1. Hasonló rendszerek.....	2
2.1.1. Bevezetés	2
2.1.2. Egyetem központú oldalak.....	2
2.1.3. Egyetemek közötti kurzusok.....	3
2.1.4. Professzor értékelő rendszer	5
2.1.5. Külföldi professzor értékelő rendszer	6
2.1.6. Hasonló rendszerek összehasonlításának összegzése	8
2.2. Kapcsolódó témakörök	9
2.2.1. Szoftverfejlesztési módszerek és modellek	9
2.2.2. TypeScript	13
2.2.3. Angular.....	14
2.2.4. .NET.....	15
2.2.5. .NET MAUI	16
2.2.6. QR-kód.....	17
2.2.7. Git / GitHub	19
2.2.8. API hívások.....	22
2.2.9. JWT Authentikáció	23
2.2.10. Összegzés.....	24
3. Követelmény Specifikáció	25
4. Tervezés	26
4.1. Rendszerterv	27
4.2. Funkciólista.....	32
4.2.1. Bejelentkezés menete.....	34
4.2.2. Értékelések folyamata.....	34
4.2.3. Telefonos applikáció használatának folyamata.....	35
4.3. Fejlesztés tervezett menete	36
5. Fejlesztés.....	38
5.1. Backend	38

5.1.1. Modellek.....	38
5.1.2. Rétegek felépítése	39
5.1.3. Kétfaktoros hitelesítés	42
5.2. .Net Maui telefonos applikáció	43
5.3. Frontend	44
5.3.1. Felépítés és kényelmi funkciók	44
5.3.2. Főoldal.....	45
5.3.3. Authentikáció	46
6. Tesztelés	47
6.1. Backend.....	47
6.1.1. Comment Modell Tesztelése	47
6.1.2. Rating Modell Tesztelése	48
6.1.3. Egyéb Modellek Tesztelése	49
6.2. Frontend	49
6.2.1. Selenium tesztesetek	49
6.3. Összegzés	51
7. Értékelés.....	54
7.1. Eredmények bemutatása.....	54
7.2. Az elvárásokhoz mért teljesítmény	57
7.3. Összegzés	58
8. Továbbfejlesztési lehetőségek	59
8.1. Valós idejű értesítési rendszer	59
8.2. Offline mód	59
8.3. Gamifikáció.....	59
9. Összegzés	60
Irodalomjegyzék.....	61
Ábrajegyzék	64
Táblajegyzék	65

1. BEVEZETÉS

A mai világban a diákoknak a tárgyfelveételről és egyéb kurzusokról igazán kevés információ lelhető fel az interneten. Kevés fórum található, ahol megoszthatják eddigi tapasztalataikat és akár kifejezhetik nemtetszésüket, vagy éppen dicsérhetik azt. Ezen információk megszerzése azonban gyakran kihívást jelenthet, különösen azoknak a hallgatóknak, akik először találkoznak egy-egy tantárggyal vagy oktatóval.

A digitális korszakban azonban egyre több hallgató számára válik elérhetővé az az lehetőség, hogy az interneten keresztül megismerje mások tapasztalatait és értékeléseit az egyetemi tárgyakról és kurzusokról. Ebben a kontextusban egyre népszerűbbé válnak az olyan online fórumok és platformok, ahol a diákok megoszthatják véleményüket és tapasztalataikat az egyes tantárgyakról.

Szakedolgozatom célja, hogy létrehozzak egy ilyen biztonságos környezetet, ahol a hallgatók ezen gondolataikat széleskörben képesek megosztani. A másik nagyon fontos téma pedig a biztonság, mert az internet radikális fejlődése sajnos az adathalászatot is sokkal gyakoribbá tette. Véleményem szerint a webalkalmazásom hasznosnak fog bizonyulni és segíthet a hallgatóknak abban, hogy jobban navigáljanak az egyetemi tanulmányaik során, mindezt olyan biztonságos környezetben, ahol a diákok értékelései és véleményei profilukhoz kötve a robotok és adathalászok ellen védve van.

A következő fejezetekben érinteni fogom a ma elérhető megoldásokat, azok előnyeit és hibáit. Összehasonlítást végzem a vizsgált rendszereken és azokat pontozással értékelem szempontok alapján. Az irodalomkutatás során szerzett tapasztalatom alapján bemutatom az alkalmazásom terveit, majd a választott technológiákról és a rendszer felépítéséről, annak fontosabb részeiről fogok írni.

2. IRODALOMKUTATÁS

2.1. Hasonló rendszerek

2.1.1. Bevezetés

Hasonló rendszereket tekintve Magyarországon nem sok ilyen oldalt találtam, amely pontosan ezt a fajta témát feszegetné. Külföldön viszont, már nem ismeretlen a probléma, amelyre számtalan példát találtam és így tudtam elemezni is. Pontos és részletes megfigyelés után saját webalkalmazásom ezek a hiányosságokból fog még pontosabb és felhasználóbarátabb lenni, amelyeket a hasonló rendszerekben is fejtek.

2.1.2. Egyetem központú oldalak

Bizonyos weboldalak inkább egy-egy adott egyetemnek a tárgyaira fókuszál, ilyen például a RateMyCourse.org. [1] Ezen az oldalon regisztráció után lehet saját véleményt formálni, az adott egyetemen tanított tárgyról, amelyek csoportokba vannak sorolva, ezzel is könnyebbé téve a felhasználó számára egy adott tárgy megtalálását.



1. ábra a Rate my course weboldal főoldala [1]

Ezt a projektet kettő darab hallgató hozta létre és tartja azóta is karban. Az oldal az egyetem színeit használja és pár egyszerű funkcióval van ellátva. Fő menüpontok például a főoldal, a kurzusok, információ a készítőkről és a bejelentkezés ahogy ezt a 1. ábra a Rate my course weboldal főoldala is mutatja. Végig navigálva ezeken a menüpontokon, a kurzusokról túl sok információ nem található, értékelésekből pedig sajnos igazán kevés van. [2]

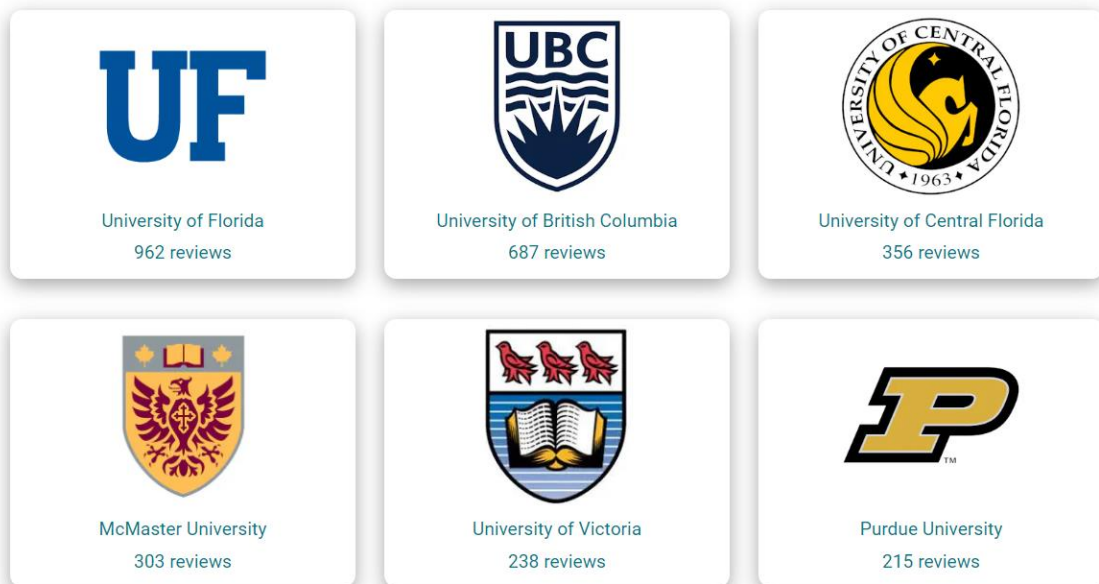
Az oldalon tárgyakat felvenni sajnos nem lehet, tehát a felhasználók által nem bővíthető az adatbázis. Az oldalon elérhető tárgyakat csak regisztráció után lehet értékelni és a bejelentkezés pedig csak Google fiókkal tehető meg. Regisztráció fül nem található, mert amennyiben még sosem jelentkezünk be, olyankor bejelentkezés előtt a regisztrációt is elvégzi nekünk a Google hitelesítés.

Az oldalon sajnos nincs elég szűrési lehetőség véleményem szerint, ami nagyban megnehezíti, hogy elérjük például a legfelkapottabb tárgyakat, a legtöbbet értékelt kurzusokat, vagy akár a legfrissebben felvetteket. Véleményem szerint az oldal legnagyobb hiányossága a felhasználó bázis csekély mérete és a pontosabb szűrésnek a hiánya.

2.1.3. Egyetemek közötti kurzusok

Hasonló témával rendelkező alkalmazások között akad olyan is, amely nem szorítja be magát egy kisebb térbe, amire a példa egy adott egyetem. Ehelyett számtalan egyetemnek vannak fent a kurzusai és tárgyai. Az oldalon van lehetőség kérelmet benyújtani további létesítmény hozzáadásáról, tehát amennyiben a felsőoktatási intézmény, amelyben folytatjuk tanulmányainkat még nincs benne a rendszerben, ilyenkor is van lehetőség az adatbázis bővítésére.

Popular Schools



2. ábra a rate my courses io oldal legtöbbet értékelt egyetemei [3]

Az oldalon regisztráció és bejelentkezésre is van lehetőség, aminél lehetőségünk van a szokásos email és jelszó párosításával bejelentkeznünk, vagy Google fiókkal is egyszerűen, egy kattintással szintúgy regisztrálhatunk és utána be is jelentkezhetünk. [3]

Az oldal átláthatóság szempontjából sokkal jobban strukturált, példa erre a 2. ábra a rate my courses io oldal legtöbbet értékelt egyetemei, amelyeket már a főoldalon láthatunk és az egyetemek alatt a pontos értékelismennyiség is fel van tüntetve. A kiválasztott egyetemre rákattintva a legtöbbet értékelt tárgyakat listába sorolva jeleníti meg, egy ilyen sorban pedig az előre meghatározott szempontoknak az átlagát jeleníti meg, ahogy ezt a 3. ábra a rate my courses io tárgyak felsorolása mutatja.

Code	Name	Overall	Easiness	Interest	Usefulness	↓ Reviews
CHM 2045	General Chemistry 1	3	2.2	2.7	2.9	31
MAC 2313	Analytic Geometry and Calculus 3	3.8	2.7	3.4	3.9	23
MAC 2311	Analytic Geometry and Calculus 1	3.6	2.8	3.1	4.2	19
MAC 2312	Analytic Geometry and Calculus 2	3.8	2.7	2.9	3.5	16
IDS 1161	What is the Good Life	3.2	3.9	2.4	2	14
STA 2023	Introduction to Statistics 1	4	3.5	3.5	4.2	13

3. ábra a rate my courses io tárgyak felsorolása [4]

Egy tárgyat kiválasztva megtekinthetjük az értékeléseket. Egy értékelés több szempontot foglal magába, köztük az összesítést, amelyet a könnyűségből, érdekeségből és hasznosságból átlagol. Egy értékelés felépítése a 4. ábra rate my courses io tárgy értékelése alapján látható.

Class Ratings

4

Good Class

4

Easy

4

Interesting

4

Useful

Professor Rating

5

Amazing Prof

Prof: Robert Fischer / Spring 2022

Jan 7, 2023

Comments on the course

I found this class quite intriguing and broadened my perspective on things.

Advice

Come in with an open mind willing to learn.

Delivery: Hybrid

Grade: A

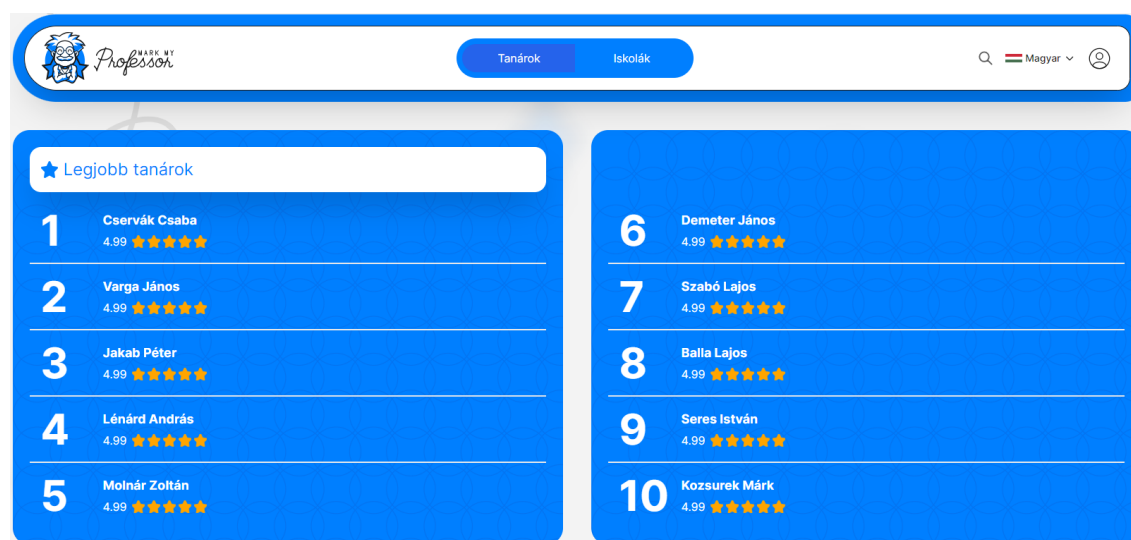
Textbook Use: Optional

4. ábra rate my courses io tárgy értékelése [5]

2.1.4. Professzor értékelő rendszer

A Mark My Professor egy erősen hasonló rendszer, amelyben az értékelések nem a kurzusokra értetendők, hanem a kurzusokat tartó egyetemi professzorokra. Az oldal 2009-ben jött létre azzal a céllal, hogy a tanárok és diákok véleményt cserélhessenek. A diákok észrevételei és véleményei fontosak lehetnek a tanárok számára, mert az egyetem keretein belül nem feltétlen kapnak elég visszajelzést. [6]

Jelenleg az oldal csak a magyar egyetemek és annak Professzorait tartalmazza, egyelőre nem terveznek terjeszkedni a külföldi piac felé. Jelenleg az adatbázisban kilencvenkettő darab egyetem sorakozik fel, közöttük az Óbudai Egyetem is. A főoldal tartalmazza a legtöbbet és legjobban értékelt professzorokat, előadókat és az egyetemi karokat is. Hiányosságát érzem annak, hogy a nevek alatt esetleg Egyetemet, vagy legalább egyetemi kart is megjelenítsen, hiszen névről azért egy első féléves diák nem sok mindent tud leszűrni. [7]



5. ábra Mark My Professor főoldala a legjobban értékelt professzorokkal [7]

Értékelések részletessége öt darab szempontból áll, amelyek csak az oldal tetején találhatóak meg, így lejjebb görgetés esetén könnyen el lehet felejtetni, hogy egy adott értékelés mire is vonatkozik pontosan. Egy ilyen értékelést lehet az oldalon Likeolni és Dislikeolni, amennyiben a maga az értékelés esetleg gyűlöletből jött és nincs mögötte semmilyen valós érv. Ezen felül jelenteni is lehet, gyűlölködés, uszítás miatt, amit egy felugró ablakban maximum 350 karakter terjedelemben meg is kell indokolni. [8]

Saját értékelés írásánál az oldal nem kötelez regisztrációra vagy bejelentkezésre. Ki kell választani a tárgyat, amit felvettünk az adott oktatónál, de van lehetőségünk új tárgyat is felvenni, hogyha az adott tárgy új lenne. Értékelni csillagokkal lehet, itt meg kell adni egytől ötig, hogy mi a véleményünk a követelmény teljesíthetőségéről, tárgy hasznosságáról, segítőkézségről, felkészültségről és az előadásmódról. Plusz

információként az összes eddigi értékelés számát is megmutatja az oldal. Csillagokkal való értékelés után egy ötszáz karakteres mezőben kifejtethetjük a véleményünk a komment szekciójában. Véleményünk beküldése előtt azonban el kell fogadni a felhasználási feltételeket, amiben külön kiemelésben részesül az, hogy véleményem nem sértheti az oktató jó hírnevét. A beküldésről a 6. ábra Mark My Professor értékelés írási felület mutat példát egy ilyen vélemény beküldésének folyamatáról. [9]

Értékelések összes értékelés: 33

Követelmények teljesíthetősége	Tárgy hasznossága	Segítőkézség	Felkészültség	Előadásmód
0 ★★★★★	0 ★★★★★	0 ★★★★★	0 ★★★★★	3 ★★★★★

Komment (maximum 500 karakter)

Szöveges értékelés

☐ Büntetőjogi felelősségem tudatában kijelentem, hogy az általam megfogalmazott vélemény nem sérti az oktató jó hírnevét, valamint nem ütközik a MarkMyProfessor felhasználási feltételeibe.

[Értékelem](#)

6. ábra Mark My Professor értékelés írási felület [9]

2.1.5. Külföldi professzor értékelő rendszer

Nem egyedi téma a hazai piacon ez a fajta véleménynyilvánítás, mert külföldön is számtalan hasonló webalkalmazás formálódott már meg. Kutatásom során rábukkantam a RateMyProfessors.com-ra aminek ötlete már 1999 májusában létrejött [10]. A weboldal több mint 8000 iskolát, 1,7 millió professzort és több mint 19 millió értékelést tartalmaz. Értékelések összesítése képpen egy úgynevezett „Overall Quality” értékelést kapunk, amely lehet jó, rossz, átlagos és emellett egy smileyt is hozzá generál.

Saját leírásuk szerint az oldal gyengesége abban lelhető fel, hogy az értékelő személynek nem szükséges regisztrálni és így nem garantálható, hogy valóban egy ott tanuló diák értékelt. Emiatt sajnos az oldal elmondása szerint bizonyos professzorok olyan dolgok miatt részesültek hátrányban, mint a nemük, bőrszínük és gyakran amennyiben politikai tárgyról van szó, gyakran a nézeteik miatt is előfordult már ilyen [10].

4 / 5

Overall Quality Based on 5 ratings

AS Andrews

Professor in the Biology department at Columbus State Community College

N/A

Would take again

2.2

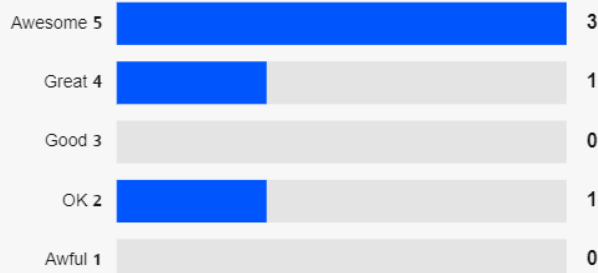
Level of Difficulty

Rate →

Compare

I'm Professor Andrews

Rating Distribution



Check out Similar Professors in the Biology Department

5.00 Aaron Alford **5.00** Caleb Evans **5.00** Chad Saathoff

7. ábra Rate My Professor-on egy Professzor értékelése [11]

Összesítő értékelést jól ábrázolja a 7. ábra Rate My Professor-on egy Professzor értékelése és annak összesítése. További professzorokkal van lehetőségünk összehasonlítani és így akár tisztább képet kapni arról, hogy választásunk melyik kurzusra essen. Bizonyos értékeléseket van lehetőségünk értékelni, de ez csak regisztráció után lesz elérhető. Az értékelések rövidek és lényegre törők, elsősorban, hogy van-e szükség füzetre alatta egy rövid leírás a személyes tapasztalatokról, bal oldalt egy sávban egy értékelés a tárgy minőségéről és egy másik értékelés a nehézségéről. Egy értékelés bemutatásának a 7. ábra Rate My Professor-on egy Professzor értékelése remekül illusztrálja a fentebb leírtakat [11].

QUALITY
5.0

DIFFICULTY
1.0

BIO100
Textbook: **No**
Mr.Andrews is a good teacher. I received a 85% in his class. Fairly decent class, not too hard. Just study the presentation notes and you should do fine in class. Attendance is not required but missing the notes will not do you any help.

Mar 18th, 2012

Helpful 0 0

Reviewed: Aug 16th, 2016

8. ábra Rate My Professors értékelés felépítése [11]

2.1.6. Hasonló rendszerek összehasonlításának összegzése

Az alábbi táblázatot azon szempontok alapján készítettem, amely közös pontok alapján a hasonló rendszerek és szakdolgozatom összehasonlíthatóak. Ilyen szempontok a kezelhetőség, a modern felület, bejelentkezés, regisztráció és biztonság, adatbázis mérete és az értékelések részletessége.

Név	Kezelhetőség	Felület	Biztonság	Adatbázis	Értékelések
Rate My Course Org	8/10	5/10	6/10	2/10	4/10
Rate My Courses Io	9/10	4/10	7/10	7/10	8/10
Mark My Professor	7/10	6/10	9/10	8/10	5/10
Rate My Professors	7/10	9/10	8/10	10/10	9/10
Saját rendszer	8/10	9/10	10/10	5/10	9/10

1. táblázat a hasonló rendszerek összehasonlításáról

Táblázatomban félkövér betűstílussal jelöltem az adott szempont szerinti legmagasabb értékelést. Saját rendszeremben törekszem a hasznos funkciók implementálását a legjobb felhasználói élmény érdekében.

2.2. Kapcsolódó témakörök

2.2.1. Szoftverfejlesztési módszerek és modellek

A szoftverfejlesztési módszerek és modellek olyan eljárások és szabályok, amelyek segítenek a szoftvertervezőknek és programozóknak a szoftverek létrehozásában, tesztelésében, karbantartásában és dokumentálásában. A szoftverfejlesztési módszerek közé tartoznak például a vízesés modell, a spirál modell, az agilis módszerek, a prototípuskészítés és a radikális modell. A szoftverfejlesztési modellek célja, hogy meghatározzák a szoftverfejlesztési folyamat lépéseit, fázisait, tevékenységeit és eredményeit. A szoftverfejlesztési módszerek és modellek kiválasztása függ a szoftver jellegétől, méretétől, követelményeitől, minőségétől, határidejétől és költségvetésétől [12].

2.2.1.1. Agilis szoftverfejlesztés

Az Agilis módszertan egy iteratív és inkrementális fejlesztési megközelítés, amely a rugalmasság, a gyors reagálás és a folyamatos visszajelzés elveire épül. A módszertan célja a hatékony és gyors szoftverfejlesztés, amely alkalmazkodik a változó követelményekhez és az ügyfél visszajelzéseire. Előnyei közé tartozik, hogy lehetővé teszi a csapatok számára, hogy gyorsan alkalmazkodjanak az ügyfélkövetelmények vagy a piaci feltételek változásaihoz anélkül, hogy vissza kellene menniük és a nulláról kezdeniük a terveiket. Az agilis módszerek jobb minőségellenőrzést is kínálnak, mivel lehetővé teszik a csapatok számára, hogy az érdekelt felektől érkező visszajelzések alapján folyamatosan javítsák munkájukat. Az agilis projektmenedzsment általában a szállítási határidőket is csökkenti azáltal, hogy a feladatokat gyorsan elvégezhető kis részekre bontja. A S.O.L.I.D alapelvek adják az agilis szoftverfejlesztés alapját. [13]

Története 1957-ig vezethető vissza, de érdembeli fejlődés 1974-ben történt, amikor E.A. Edmonds az alkalmazkodó szoftverfejlesztésről fejezte be dolgozatát. Párhuzamosan ezzel Dan Gielan igazgatása alatt a New York Telefon Társaság Rendszerfejlesztési Központja is ezen dolgozott. Az 1970-es évek közepétől pedig Gielan előadásokat tartott az Agilis módszertan elődjének gyakorlatáról és előnyeiről. A vízesés-orientált „nehéz” módszerekre 10 válaszul írtak 1990-es évek közepétől könnyedebb, „mikromenedzselt” szoftverfejlesztési módszereket, amelyekre ma már kollektívan agilis módszerekként hivatkozunk. Ilyenek például 1994-ből a Unified Process és a DSDM (dinamikus rendszerfejlesztési módszer), 1995-ből a Scrum, 1996-ból a Crystal Clear és az XP (extrém programozás). 2001 februárjában Utah-ban összeültek a szoftverfejlesztési szakértők, majd megfogalmazták és kiadták a „Kiáltvány az agilis szoftverfejlesztésért” nevű kiáltványt. Néhány szerző pedig megalakította az Agilis szövetséget, egy nonprofit szervezetet, amely a kiáltvány alapelvei szerinti szoftverfejlesztést segíti. [14]

Az agilis fejlesztési módszer folyamata

Az agilis módszerek általában lebontják az adott feladatot több kisebb feladatra. Egy fejlesztési ciklus, vagy másnéven sprint általában 4 hétig tart, ami alatt az iterációban megfogalmazott követelményeket teljesítik, a tervezéstől, a megvalósításon át a tesztelés és végül az átvételig. Az agilis módszerek folyamata az alapelvekre épül, aminek fontos alapja a vevő elégedettsége, vagyis a rövid idő alatt szállított jól működő szoftver. Gyakran kell működő verziókat szállítani a megrendelőnek (legalább havonta), így minden észrevételét és vagy változtatását minél könnyebben tudjuk implementálni a szoftverbe. Fontos a fenntartható fejlesztés, az egyenletes tempó. A személyes kommunikáció az információ leghatékonyabb módja, ezért fontos az üzleti szakemberek, a megrendelő(k) és a fejlesztők napi együttműködése. [15] A fentebb említett agilis módszerek közül szeretném kiemelni a Scrum keretrendszert, ami alapján építem fel a webalkalmazásomat is.

Scrum lényege és felhasználása

A Scrum megmutatja a termékmenedzsment és a fejlesztési gyakorlatok relatív hatékonyságát, ezáltal lehetőséget teremt a folyamatok továbbfejlesztésére. Felépítése Scrum csapatokból, hozzájuk rendelt szerepekből, eseményekből, munkaanyagokból és szabályokból áll. A keretrendszeren belül minden egyes komponens meghatározott célt szolgál, és mindegyik alapvetően szükséges a Scrum sikeréhez és hatékony alkalmazásához. A Scrum szabályai összekapcsolják az eseményeket, szerepeket és munkaelemeket, meghatározva a közöttük lévő kapcsolatokat és kölcsönhatásokat. A Scrum rendszer egyszerű és könnyen érthető, ugyanakkor a mesteri szintre fejlesztése kihívásokkal teli lehet. [16]

Scrum felépítése és alapfogalmak

Alapvetően három fontos szerepcsoport van, előre sorolnám a Product Owner-t, aki bár nevéből adódóan terméktulajdonos, azonban ez mégsem a megrendelőt jelenti. A Project Owner, továbbiakban PO, a fejlesztői cég része és feladata a megrendelői igény felmérése, a termék feladat listájának létrehozása és karbantartása. Fontos ismét kiemelni, hogy ez a szerep a felhasználói igényeket reprezentálja, és prioritást ad a feladat leírásának. Második ilyen szerepkör a Scrum Master. Ez egy támogató vezetői szerepkör, melynek fő célja a csapat teljesítményének növelése és a munkát akadályozó tényezők elhárítása, amelyek gátolják a csapatot abban, hogy a sprinteken hozott célokat megvalósítsák. A Scrum Master nem a csapat vezetője (hiszen a csapat önszerveződő), hanem inkább a csapat és a külső tényezők közötti közvetítői szerepet tölt be. Feladata, hogy ellenőrizze, hogy a Scrum által előírt folyamatokat megfelelően alkalmazzák, és betartsák a Scrum szabályait. Kulcsfontosságú feladata a csapat védelme és annak biztosítása, hogy a csapat az elvégzendő feladatokra koncentráljon. A Scrum Master a csapatot lendületben tartja, és kiemelt szerepet játszik az események lebonyolításában.

Termék követelménylista (Product backlog)

A "termék követelménylistája" magas szintű követelményleírásokat tartalmaz az egész termékkel kapcsolatban. Ezek a követelmények funkcióleírásokat, kívánságokat, ötleteket és hasonlókat foglalhatnak magukban, amelyeket prioritás szerint rendeztek. Alapvetően ez a lista definiálja a fejlesztés célját. A lista nyitott a szerkesztésre mindenki számára, és becsléseket tartalmaz a bejegyzések üzleti értékére és a velük járó ráfordításokra vonatkozóan. A becslések segítik a terméktulajdonost a feladatok sorrendjének és prioritásának meghatározásában. A termék követelménylistáját a terméktulajdonos kezeli. Az üzleti értéket a terméktulajdonos határozza meg, míg a fejlesztési ráfordításigényt a fejlesztők. A követelmények gyakran az úgy nevezett "user story" formátumot követik. [16]

Sprint teendőlista (Sprint backlog)

Meghatározza, hogyan fogja a csapat elvégezni a sprint során tervezett funkciókat, amiket kisebb részfeladatokra bontanak le, amelyek általában 4-16 órás időtartamra vannak darabolva. Előnye, hogy ezen részletezettségi szint mellett minden csapattag pontosan megérti, hogy milyen feladatokat kell végrehajtani, és mindenki kiválaszthatja a számára legmegfelelőbb részfeladatokat. Fontos, hogy a feladatokat a fejlesztők választják ki és nem más rendeli ki őket adott fejlesztőkhöz, mert így saját képességeikhez mérten tudnak feladatokat végezni, ami boldogabb csapatot és gyorsabb munkaelőrehaladást is eredményez. A 9. ábra Scrum teendőlista felépítése jól bemutatja egy sprint alatt megbeszélte feladatok eredményes szétosztását és aktív használatát. [16]



9. ábra Scrum teendőlista felépítése [17]

Növekmény (Increment)

A leszállított termék backlog elemeinek és az összes azelőtt lefolytatott sprint során keletkezett elemeknek a növekmény értékének összessége. Ennek az értéknek a sprint befejezésével „kész” értéket kell felvegyen, vagyis felhasználható állapotúnak kell legyen attól függetlenül, hogy a terméktulajdonos végül kibocsájtja-e azt.

Scrum folyamata

Minden projekt először egy sprint tervezéssel kezdődik, amely megbeszélések az elvégzendő feladatok kijelölésével kezdődik a termék teendőlistájáról, ebben a Project Owner (terméktulajdonos) is részt vesz. A fejlesztők és a teljes csapat is részletezi észrevételeit az egyes részfeladatok időszükségleteiről. Megtervezik az első sprint során elvégzendő feladatokat és a fejlesztőcsapat pedig eldönti azoknak szétosztását. Egy ilyen megbeszélés 4 hetes sprintek esetén akár 8 óra hosszúságú is lehet.

Napi Scrum-megbeszélés

Másnéven a Daily Standup a tevékenységek összehangolására lett kitalálva. Általában reggel, nap indító meetingként szolgál. Általában 15 percre le van limitálva az ideje és a résztvevők az előző napi fejlődésről, problémákról, és az aznapi tervekről konzultálnak egymással. Rengeteg munkahelyen ezek a meetingek állva történnek, hogy serkentsék a rövid és fókuszált beszélgetést. Gyakorisága napi, a találkozót minden nap ugyanabban az időben tartják, és általában a csapat munkaterületén vagy egy virtuális platformon zajlik. A Daily Standup egy hatékony eszköz a napi előrehaladás monitorozására, a csapatösszetartozás erősítésére és az akadályok gyors azonosítására a Scrum projektben. A rövid és fókuszált formátum biztosítja, hogy mindenki gyorsan és hatékonyan megossza a fontos információkat. [16]

Sprint áttekintés / Demo

Minden sprint végén tartanak egy ilyen meetinget és célja az eddig elkészült termék bemutatása a terméktulajdonosnak. Remek lehetőség a visszajelzések begyűjtésére a csapatot, vagy az irányt illetően. A Project Owner ellenőrzi, hogy az eddig elkészült alkalmazás meddig jutott el és hogy a termék megfelel-e a követelményeknek az eredeti elképzelések alapján. Ilyenkor szokás a termék backlog felülvizsgálata és a prioritások újragondolása a feladatoknál.

Sprint visszatekintés (Retrospective)

Célja, hogy a fejlesztőcsapat és az érintettek körében egy nyílt és konstruktív fórumot biztosítson, ahol lehetőség van a csapat teljesítményének értékelésére és a folyamatos fejlesztési lehetőségek azonosítására. Fontos, hogy nem csak a negatívumokról van szó, egy külön pont a pozitívumok megjegyzése, így a sikereket is megosztjuk és elismerjük a jól elvégzett munkát. [16]

2.2.2. TypeScript

A TypeScript egy ingyenes és nyílt forráskódú magasszintű programozási nyelv, amit a Microsoft fejlesztett ki 2012-ben kettő évnyi fejlesztés után és célja a JavaScript nyelv kiterjesztése olyan funkciókkal, amelyek segítik a nagyobb és összetettebb alkalmazások fejlesztését. Mivel a TypeScript a JavaScript egy „bővebb” változata, így az előzőekben megírt JavaScript típusú programjaink tudnak TypeScript környezetben is futni, esetekben viszont biztonsági problémák, vagy típus ellenőrzés miatt a fejlesztői környezet vissza is dobhatja. Ez a nyelv használható arra is, hogy JavaScript alkalmazásokat fejlesszünk mind kliens, mind szerveroldalra például úgy, ahogy ezt Node.js-ben, vagy mint Deno-ban tudjuk. [18]

Különbségek a JavaScript nyelvtől

A TypeScript és a JavaScript közötti különbségek közül az egyik legfontosabb a típusrendszer. A TypeScript egy szigorúan típusos nyelv, amely lehetővé teszi a fejlesztők számára, hogy a kód biztonságosabb legyen és a hibák számát is csökkentik ezzel. A JavaScript viszont egy gyengén típusos nyelv, amely nem követeli meg a változók típusának deklarálását. A kód olvashatóbbá és karbantarthatóbbá is válik ezáltal. Fontos megemlíteni, hogy a TypeScript továbbá támogatja a generikus típusokat is, amelyek lehetővé teszik az általánosan használható függvények és osztályok írását anélkül, hogy meg kellene ismételni ugyanazt a kódot különböző helyeken, különböző típusokra. Ezt egy rövid példában is jól be tudom mutatni, ami a 9. ábra TypeScript generikus metódus hivatkozáson is jól látszódik. [19]

```
1 function arrayElements<T>(array: T[]): void{
2     for(let item of array){
3         console.log(item);
4     }
5 }
6 arrayElements([1, 2, 3]);           //Console: 1, 2, 3
7 arrayElements(["a", "b", "c"]);     //Console: a, b, c
8
```

10. ábra TypeScript generikus metódus

Leghasznosabb funkciók közé tartozik még az előzetes típusellenőrzés, mivel a fejlesztői környezet, ebben az esetben a Microsoft saját szoftverét hoznám be példának, tehát a Visual Studio Code előzetesen lefordítja a kódot és már fordulási időben jelzi az esetleges problémákat, ezzel segít elkerülni őket a futási idő előtt. [19]

Összességében a TypeScript egy nagyon elterjedt nyelv és nagyon széles körben támogatott is. Rengeteg fejlesztési platformon használják, amelyek közül a leghíresebbek a Node.js, az Angular, a React és a Vue.js is.

2.2.3. Angular

Alapvetően egy nyílt forráskódú front-end webkeretrendszerrel beszélünk, amelyet a Google fejlesztett ki és azóta is rendszeres frissítéseket kap. Dinamikus és korszerű webalkalmazások készítésére lett kitalálva. Az Angular egy TypeScript alapú keretrendszer, amely nem összetévesztendő az Angular JS-el. Általában a keretrendszerek növelik a webfejlesztés hatékonyságát és teljesítményét, mivel egy következetes struktúrát biztosítanak, így a fejlesztőknek nem kell mindig nulláról újra felépíteniük a kódot. A keretrendszerek időt takarítanak meg, és további számos plusz funkciót kínálnak még a fejlesztőknek. [20]

2.2.3.1. Dokumentum Objektum Modell

A Dokumentum Objektum Modell, vagy másnéven DOM egy olyan objektummodell, amely teljesen nyelv és platformfüggetlen. A HTML, XML és egyéb rokonságban álló formátumoknak szerkezetét és objektumaikkal történő interakciókat modellezi. A dokumentum minden tartalmát és összetevőjét tartalmazza.

Az Angular nem közvetlenül a DOM-ot változtatja, vagy manipulálja, hanem egy virtuális DOM-on történnek a műveletek, ezzel minimalizálva a valós DOM-on történő változásokat és így optimalizálja azt. Ilyen művelet például az adatkötés, vagy az eseménykezelés is. Az adatkötés egy olyan folyamat, amely a modell és a nézet közötti automatikus adatfrissítést foglalja magában. Az Angular kétirányú adatkötést használ. A modell állapota tükrözi azokat a változásokat, amelyeket a UI elemekben végeztek és emellett a UI állapota tükrözi azokat a változásokat, amelyek a modell állapotában történtek. Ezáltal az alkalmazás képes az adatok dinamikus megjelenítésére a felhasználói felületen. Az eseményvezérelt programozásra is lehetőséget ad az Angular, ilyenek például a gombnyomások, vagy az egérgattintás és ezek hatására változik az alkalmazás állapota, de sok esetben a DOM elemei is. [20]

2.2.3.2. Angular modulok

A következetes struktúra magában foglalja a moduláris szerkezetet is. Ezek az alkalmazások modulokra vannak osztva, amik segítik a kód szervezését és annak újrafelhasználását. Az Angular moduljait gyűjtőnéven NgModuloknak hívjuk, ezek lényegében olyan konténerek, amelyek összefüggő kódblokkokat tartalmaznak.

```
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  @NgModule({
5    imports:      [ BrowserModule ],
6    providers:    [ Logger ],
7    declarations: [ AppComponent ],
8    exports:      [ AppComponent ],
9    bootstrap:    [ AppComponent ]
10  })
11  export class AppModule { }
```

11. ábra Angular modulok importálása

Tartalmazhatnak komponenseket, szolgáltatókat és olyan kódrészleteket is, amelynek kiterjedését az adott NgModul határozza meg. Importálhatnak olyan funkcionálisokat is, amik másik NgModulokból származnak és ezeket exportálni is tudja más moduloknak. Példa erre a 12. ábra Angularban a routing beállítása hivatkozás, ami jól bemutatja a modulok importálását. [20]

Routing

Az egy oldalas alkalmazásoknál a fejlesztő úgy változtatja a látott tartalmat, hogy kisebb porciókat jelenít meg, vagy tüntet el a felhasználó előtt modulokkal és komponensekkel, ahelyett, hogy új oldalt kérne le a szerverről.

```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { HomeComponent } from './home.component';
4 import { AboutComponent } from './about.component';
5
6 const routes: Routes = [
7   { path: '', component: HomeComponent },
8   { path: 'about', component: AboutComponent },
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule]
14 })
15 export class AppRoutingModule { }
16
```

12. ábra Angularban a routing beállítása

Az oldalak közötti navigációt az Angular „Router” végzi amit be kell először importálni, majd ahogy a 12. ábra Angularban a routing beállítása is mutatja, ebben a példában kettő útvonalat definiálunk. [21]

2.2.4. .NET

A dotnet egy keretrendszer, amelyet a Microsoft fejlesztett ki és lehetővé teszi különböző típusú alkalmazások létrehozását és futtatását. Maga a keretrendszer több programozási nyelvet is támogat, szám szerint 40 darabot, köztük a C#, F#, Visual Basic-et is. Ezzel a keretrendszerrel készült alkalmazások platformfüggetlenek, így a .NET 5 és újabb verzióval készült alkalmazások futtathatóak Windows, Linux és macOS rendszereken is. [22] Ezek a platformfüggetlen alkalmazások általában asztali, játék, vagy mobilalkalmazásokat jelentenek. Ezeken kívül számos másik környezetben is használatos a .NET, ilyenek a felhőalkalmazások, az ügyfélalkalmazások és a külön Windows-ra dedikált alkalmazások.

Funkciókat tekintve a .NET lehetővé teszi a fejlesztők számára, hogy megbízható és hatékony kódot írjanak. Ilyenek például az aszinkron kód, az attribútumok, a kódelemzők, a lambda kifejezések, eventek és a párhuzamos programozás is a rengeteg sok másik funkció mellett.

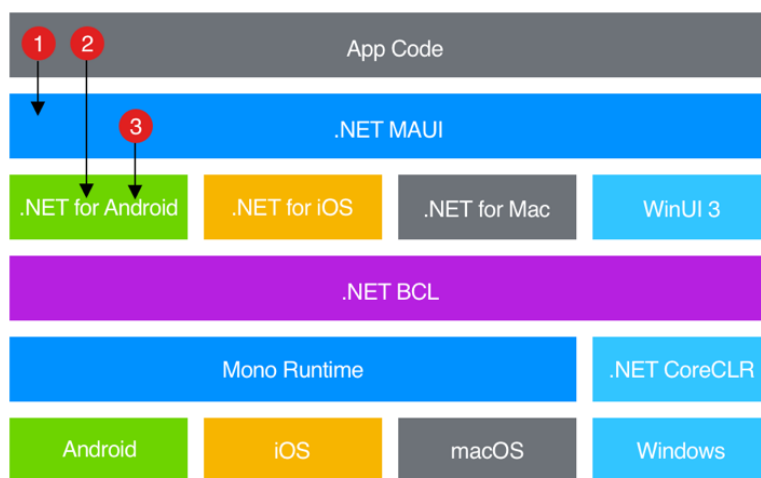
A .NET új verziói minden év novemberében kerülnek kiadásra. A páratlan számú évek kiadásai hosszú távú támogatást (LTS) biztosítanak, és három évig élvezhetik azt. A páros számú évek alatt kiadott verziók standard távú támogatással (STS) rendelkeznek, és 18 hónapon át kapják meg a rendszeres frissítéseket és rendelkeznek támogatással. [23]

2.2.5. .NET MAUI

A .NET MAUI (Multi-platform App UI) egy olyan keretrendszer, amely lehetővé teszi, hogy C# és XAML kódbázisból hoz létre natív mobil- és asztali alkalmazásokat. A .NET MAUI a legújabb technológiákat használja fel a Windows, macOS, iOS és Android natív alkalmazások fejlesztéséhez, absztrahálva ezeket egy közös .NET alapú keretrendszerbe. [24]

2.2.5.1. .NET MAUI működése

A .NET MAUI alapját a .NET 6 jelenti, ami egy korszerű, nyílt forráskódú keretrendszer a szoftverfejlesztéshez. Az elődje, a Xamarin.Forms, továbbfejlesztése és modernizálása. A .NET MAUI lehetővé teszi a fejlesztők számára, hogy egységes kódbázissal készítsenek platformfüggetlen alkalmazásokat, és azokat az egyes célplatformokhoz optimalizált felhasználói élménnyel lássák el. [24]



13. ábra .NET MAUI architektúrája [24]

Az architektúra felépítésének szemléltetésére a 13. ábra .NET MAUI architektúrája hivatkozás remekül bemutatja, hogy a .NET MAUI alkalmazás kódja elsődlegesen az egyes számmal megjelölt réteggel veszi fel a kapcsolatot, fordítja le a kódot és csak ezután kezd neki a kettes és hármas számmal megjelölt API-k hoz. Ehhez fűzném hozzá, hogy bizonyos esetekben, a leírt kódot egyből fordíthatja akár a kettes számmal megjelölt API-ra is. [24]

2.2.5.2. .NET MAUI alkalmazások egyéb rendszerekre való fordítása

- A lefejlesztett alkalmazás androidra való fordítása úgy működik, hogy az általunk leírt C# kódot a .NET MAUI először egy közös nyelvre (intermediate language), amit aztán valós időben (JIT) fordít le egy natív assembly nyelvre, amikor az alkalmazás elindul. [24]
- IOS alkalmazások tekintetében már előre lefordítja az általunk írt C# kódot egy natív ARM assembly nyelvre, tehát nem valós idejű a fordítás. [24]
- macOS alkalmazásoknál a MAUI a Mact Catalyst segítségével végzi el a szükséges átalakításokat, ami egyébként az Apple saját megoldása és célja az IOS-re írt alkalmazások asztali megjelenítése UIKit segítségével. [24]
- A .NET MAUI-ban írt Windows alkalmazások a Windows UI 3 (WinUI 3) könyvtárat használják, hogy natív alkalmazásokat hozzanak létre, amelyek a Windows asztali platformra lettek tervezve. [24]

2.2.6. QR-kód

A QR (Quick Response) kód egy kétdimenziós vonalkód, amelyet 1994-ben találtak fel Japánban autó alkatrészek nyilvántartására. Ezek a kódok fekete és fehér blokkokból állnak, amelyet négyzetes alakban foglalnak helyet. Beolvasásuk kamerával lehetséges és a bennük található információt az eszközök képesek értelmezni és megjeleníteni azt. [25]

2023-ra a QR-kódokat sokkal szélesebb körben alkalmazzák, ideértve a kényelem orientált alkalmazásokat. A QR-kódokat használhatjuk szövegek megjelenítésére a felhasználónak, weboldal megnyitására, névjegy hozzáadására, Uniform Resource Identifier (URI) megnyitására, vezeték nélküli hálózathoz való csatlakozásra és vagy e-mail, vagy szöveges üzenet írására is. Számos QR-kódgenerátor áll rendelkezésre alkalmazás formájában vagy online akár online is, amelyek általában ingyenesek. A QR-kód az egyik leggyakrabban használt típusú kétdimenziós kód lett jelenleg a világon. [25]

2022-ben az Egyesült Államokban 89 millió ember szkennelt QR-kódot mobil eszközükkel, ami 26%-kal több, mint 2020-ban. A QR-kódok többségét a felhasználók fizetés céljából olvasták be, vagy termék- és menüinformációkat kértek le vele. [25]

2.2.6.1 QR-kód generálás

QR-kód generáláshoz a legtöbb ember által használt és megbízható nyílt forráskódú „QRCoder” nevű könyvtárat tanulmányoztam, amely C# .NET alapokon fekszik és könnyedén elérhető a Visual Studio-nak a Nuget csomagjai között. Nem szükséges semmilyen egyéb könyvtár letöltése, elég az „Install-Package QRCoder” kódot beírni a Package Manager konzolba, vagy kézzel manuálisan letölteni a Nuget csomagok közül. [26]

```
1 QRCoder qrGenerator = new QRCoder();  
2 QRCodeData qrCodeData = qrGenerator.CreateQRCode("Text which should be encoded.",  
3   QRCodeGenerator.ECCLevel.Q);  
4 QRCode qrCode = new QRCode(qrCodeData);  
5 Bitmap qrCodeImage = qrCode.GetGraphic(20);  
6
```

14. ábra QR-kód generálása C# nyelven

Ahogy a 14. ábra QR-kód generálása C# nyelven bemutatja, összesen négy sor kódból képesek vagyunk QR-kódot generálni és megjeleníteni azt. Rengeteg opcionális paramétert tudunk még megadni, akár azt is, hogy milyen színekből generálja nekünk, vagy akár, hogy egy saját logó-t rakjon a közepére. [26]

A QRCoder könyvtár azonban nem merül ki az előbb leírt tulajdonságokban, összesen tizenegy féle kódot képes nekünk generálni, amelyekből kilenc darabhoz nem szükséges további könyvtár telepítése. Ilyen kódok például a [26]

- QRCode
- ArtQRCode
- AsciiQRCode
- Base64QRCode
- BitmapByteQRCode
- PdfByteQRCode
- PostscriptQRCode
- SvgQRCode
- UnityQRCode (QRCoder.Unity segítségével)
- XamlQRCode (QRCoder.Xaml segítségével)

Technikailag a QR-kód csak egy szöveges karakterláncnak a vizuális ábrázolása. Mindazonáltal a legtöbb QR-kódozoló képes "speciális" QR-kódok olvasására, amelyek különböző műveleteket képesek végrehajtani. Ilyen például a WiFi-QR-kódok, aminek köszönhetően beolvasás után a telefon automatikusan csatlakozik a vezeték nélküli hálózathoz. Ezek a "speciális" QR-kódok speciálisan strukturált adatcsomagokból, karakterláncokból generálódnak. [26]

Ezekből a karakterláncokból az alábbi „különleges” utasításokat lehet létrehozni: [26]

- Könyvjelző beszúrás
- Naptár esemény létrehozása
- SIM kártya adatok mentése
- Geolokáció
- Email küldése
- Egyszerhasználatos jelszó mentése
- Skype hívás indítása
- SMS küldése
- URL megnyitása
- Wifi hálózathoz való csatlakozás

2.2.7. Git / GitHub

A git egy nyílt forráskódú verziókezelő szoftver, amelyet eredetileg a Linux kernelhez fejlesztettek ki. Használata széleskörben elterjedt a fejlesztők és a szoftvertervezők között és ma már alapvető eszköz a csoportos kódolásoknál. Eredetileg egy alacsony szintű rendszernek tervezték felhasználói felület nélkül, mindazonáltal a Git projekt azóta teljes értékű revíziókövető rendszerré szélesedett, amelyet már közvetlenül is lehet használni. Minden fejlesztő a teljes projekt történetét tölti le a szerverről és tárolja a saját gépén. [27]

2.2.7.1. Alapvető fogalmak

A repository (tároló vagy raktár) egy olyan hely, ahol a Git vagy más verziókezelő rendszer tárolja a projekt forráskódját és az annak előzményeit. Ez a hely lehet helyi gépen, a fejlesztő számítógépén, vagy távoli szerveren, például egy online szolgáltatón, mint például a GitHub, GitLab vagy Bitbucket. Tartalmazza a forráskódot, a commitokat, az ágakat (brancheket) és a történetét is beleértve minden változást. [28]

A commit egy kisebb egység, ami tárolja a kódban írt változtatásokat, ezeknek a tulajdonosát, a változtatás időpontját és egy rövid üzenetet, ami elmagyarázza a változások lényegét.

A branchek, vagy másnéven ágak a párhuzamos munkát teszik lehetővé. Különböző funkciókon tudunk egyszerre dolgozni, amiket aztán össze kell fésülni, ezeket hívjuk „merge” -nek.

A merge, vagy összefésülés a különböző ágakon végzett változtatások egyesítése, ezzel tehát a párhuzamosan végzett funkciókat segít egyesíteni.

A clone egy klónozási művelet, amivel a fejlesztők a teljes repository tartalmát tudják lemásolni a saját számítógépükre, így a fejlesztők a saját, helyi példányukon tudnak dolgozni, amit aztán fel tudnak majd pusholni.

A pusholás egy olyan művelet, ami a saját számítógépen végzett változtatásokat felküldi a távoli repositoryba, így tudják mások le cloneozni a megírt funkcióinkat és további változtatásokat.

Egy conflict akkor jelenhet meg, amikor A és B push között átfedés történik a kódban, ilyenkor a Git nem tudja eldönteni melyiket tartsa meg és ezeket manuálisan kézzel kell kijavítani. Ezekre megoldásokat a Git workflow-k adnak. [28]

2.2.7.2. Git workflow

Egy Git workflow egy „használati utasítás” a Git használatához, segít a folyamatos és produktív kódolásban. Erre azért van szükség, mert a Git nagyon a flexibilitásra fókuszál és ezért nincs megszabva a konkrét használata. Fontos, hogy egy projekt indításánál, a fejlesztők egyet értsenek a workflow kiválasztásában. Ilyen döntéseknél a csapat mérete és az esetleges hibák kijavítása lehet a fő szempont. [29]

Centralized workflow

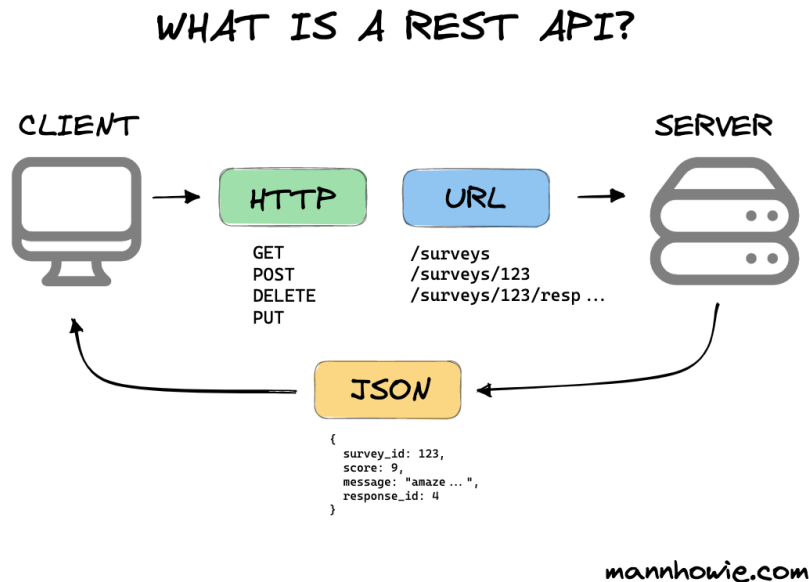
Csapatok számára nem ideális, mert nem igényel brancheket. Egy darab főágunk van, nevezetesen a Master és mindenki ide pusholja fel a változtatásokat. Ez a legegyszerűbb módja a git használatának, mert amikor végeztünk egy adott funkcióval, csak feltöltjük és leírjuk a változásokat. Problémák azonban akkor jelennek meg, amikor egy projekten már többen dolgozunk. A kódban átfedések jelenhetnek meg, ami conflictokhoz vezetnek és a pusholást nem engedi majd a Git. Ezeket, ahogy a Git fogalmaknál írtam, sajnos kézzel kell megoldani és eldönteni melyik kód maradjon és melyiket töröljük. [30]

Feature branch workflow

Bár ez a workflow továbbra is használja a master branch-et és arra építkezik, de a commitolások viszont különféle branchekre történnek. Minden egyes funkciónál egy új ágat hozunk létre, amin történnek a változások a kódban. Ezeknek a brancheknek az elnevezésére is van egy szabvány, ami általában úgy néz ki, hogy „feature / login-template-header”. Öt darab kategória van egy branch típusának, ilyen a hotfix, a bugfix, a feature, a test és a wip (work in progress). Amikor tehát egy funkció elkészült, akkor egy pull requestet nyújtunk be, amit a többi fejlesztőnek kell megerősíteni, hogy a kód és a funkció merge-eljen hozzá a master branch-hez, ami az alkalmazás egy „kész”, vagy legalább is működő állapotát reprezentálja. [31]

2.2.8. API hívások

Információk lekéréséhez a GET típusú hívást fogom használni, ami a backendemből vissza tudja szolgáltatni az adatot objektumként, vagy tömbként is. Ezeket az adatokat vagy XML, vagy pedig JSON formátumban adja vissza nekünk a backend egy http 200-as OK kóddal, ahogy ezt a 16. ábra Rest API hívások működése is mutatja. [34]



16. ábra Rest API hívások működése [34]

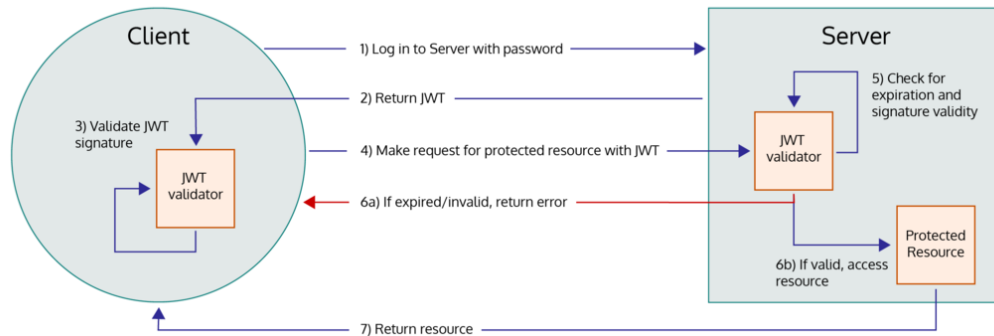
POST metódust a kliens által küldött adatok továbbítására használják a szerver felé. A POST kérés az erőforrás létrehozását vagy módosítását célozza, és a kérés törzsében hordozza az adatokat. A szerver válaszként visszaküldi a létrehozott felhasználó adatait. A POST API hívások tervezésekor és használatakor mindig fontos a biztonság, az adatintegritás és a megfelelő hiba kezelés. [35]

Az PUT és a PATCH metódus, a már létező entitásokra értelmezett változtatásokat küldi a szerver felé. Különbségeikre mindjárt kitérek. Fontos, hogy a hívásban meg kell címezni, a már létező objektum azonosítóját és a PUT esetében a frissítendő objektum minden adatnak szerepelnie kell, még azoknak is, amelyek nem változtak. A PATCH esetében ezekre már nem kell figyelniük, mert a kliensünk csak a frissítendő adatokat küldi el a szervernek, természetesen az azonosítót továbbra is csatolnunk kell.

A DELETE metódust a kliens által küldött adatok törlésére vagy eltávolítására használják a szerver felé. A DELETE kérés segítségével a kliens egy meglévő erőforrást kérhet a szervertől eltávolítani. Itt is tartalmaznia kell a kérésnek az azonosítót és általában kell egy visszajelzést kapni a szervertől a törlés sikerességéről. [36]

2.2.9. JWT Authentikáció

JWT, vagy JSON Web Token, egy olyan nyílt szabvány, amely biztonságosan átadható információkat tartalmaz egy JSON formátumú tokenben. Általában autentikáció és információ-cserék alkalmazására használják webes alkalmazásokban.



17. ábra JWT Authentikáció folyamata

Egy ilyen token három részből áll, ez a három rész pedig a fejléc (header), adattartalom (payload) és az aláírás (signature). A fejléc tartalmazza az algoritmus és a token típusát, az adattartalom nevéből adódóan a való tartalmat foglalja magában, ilyen például a név, az id és ehhez hasonló. Az aláírás pedig a fejléc és az adattartalom alapján generálódik, ez egyfajta biztosíték, hogy a tokenünk érvényes és ami még fontosabb, hogy sértetlen.

Egy JWT azonosítás menete a következő, Amikor egy felhasználó bejelentkezik, a szerver ellenőrzi a hitelesítő adatokat. Ha a hitelesítés sikeres, a szerver generál egy JWT-t. A szerver létrehozza a JWT-t a felhasználó adataival, például a felhasználó nevével és jogosultságaival. Ezután kiszámolja az aláírást a titkos kulcs segítségével, majd összeállítja a teljes JWT-t. Ezután a szerver visszaküldi a JWT-t a kliens felé, amit a kliens elment a böngésző lokális tárhelyében, vagy a memóriában. A kliens innentől minden kérés után elküldi a szervernek a JWT-t, hiszen ebben van az azonosítónk és a jogosultságaink is, a szerver pedig mindig feldolgozza ezt és ellenőrzi az aláírást.

2.2.10. Összegzés

Szakdolgozatom ez előzőekben is bemutatott technológiákból épül fel. Alkalmazásom front-end-je az Angular és .NET MAUI keretrendszert használ, mert kiváló választás lehet, ha szükségünk van egy teljeskörű és erős frontend keretrendszerre, amely lehetőséget biztosít a hatékony fejlesztésre, skálázhatóságra és karbantarthatóságra. Funkciólistája széleskörű és az integrált tesztelés és a form validáció is hasznosnak bizonyul.

Webalkalmazásom back-endje ASP.NET keretrendszer lesz, mert hatékony és rugalmas fejlesztést tesz lehetővé, különféle projektméretekhez és követelményekhez alkalmazkodva. Cross-platform fejlesztési támogatása miatt pedig a további rendszereken is jól használható.

Verziókövető platformnak a GitHubot választottam, mert remek kódmegosztási platform és a feature branch workflow-t követve egy jól átlátható rendszert tudok kiépíteni.

Szakdolgozatomban használt technológiák és keretrendszerek szorosan kapcsolódnak az eddigi tanulmányaimhoz, így hatékonyan, felhasználásukkal egy professzionális Full Stack webalkalmazást és egy könnyedén használható mobilapplikációt tudok megvalósítani.

3. KÖVETELMÉNY SPECIFIKÁCIÓ

Webalkalmazásom platformfüggetlen lesz, így bármilyen felületen, ahol rendelkezésünkre áll egy modern webböngésző és stabil internetkapcsolat, használható lesz. Dinamikus megjelenítését a front-end keretrendszeremnek a Bootstrap és az Angular Material adja. Regisztrált és regisztrálatlan felhasználók kezelését különböző funkciók elérésével biztosítom. Vendégek számára az oldal elérhető lesz és a kurzusok és tárgyak böngészése is, viszont ezek értékelése már bejelentkezéshez kötött annak érdekében, hogy egy adott dühös személy ne tudja elárasztani egy professzor profilját bármilyen okból negatív, vagy pozitív értékeléssel.

Egy felhasználói regisztráció alapvető információk megadásával történik, ilyen például a név, az email cím és a jelszó. A bejelentkezés ezekkel az információkkal történik, viszont ahhoz, hogy az oldal elfogadja az adatokat, egy plusz 5 számjegyű kódot generál, amelyet QR-kód formájában is kihelyez az oldalra.

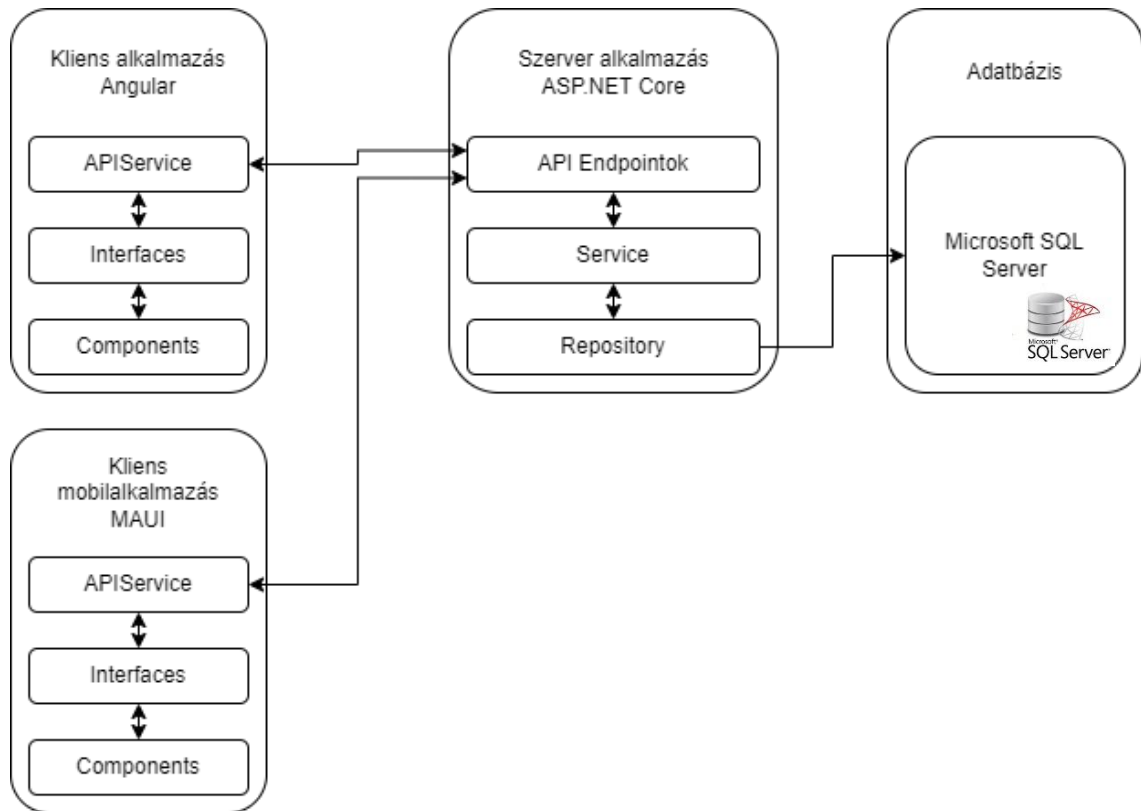
Az oldalon elérhetővé válik regisztráció után a kurzusok és egyebek értékelése és azok tetszikelése vagy jelentése.

4. TERVEZÉS

Az irodalomkutatás és a követelmény specifikáció elvégzésével megbizonyosodtam döntéseimről a keretrendszereket és a funkcionalitást illetően. A zökkenőmentesség érdekében a már eddig elsajátított nyelvekkel fogok dolgozni, ami az Angulart, a .NET MAUI-t és az ASP.NET-et foglalja magában. Az Angular és ASP.NET könnyen együttműködhetnek, például RESTful API-k használatával. Az ASP.NET Web API lehetővé teszi az egyszerű adatkommunikációt, amit az Angular könnyen feldolgoz és használ. Az Angular és az ASP.NET együttvéve egy erős technológiai csomagot alkotnak, amely lehetővé teszi a modern, hatékony és biztonságos webalkalmazások kialakítását. A .NET MAUI-ról ezek szint úgy elmondhatóak, de ebben az alkalmazásban mobil applikáció céljából került választásra, mert a .NET-es tudásomból adódóan ezzel fogok tudni a leghatékonyabban dolgozni és a minőséget tekintve is a maximumot kihozni magamból. A fejlesztőknek lehetőségük van a teljes körű alkalmazásfejlesztésre, mind frontend, mind backend oldalon.

4.1. Rendszerterv

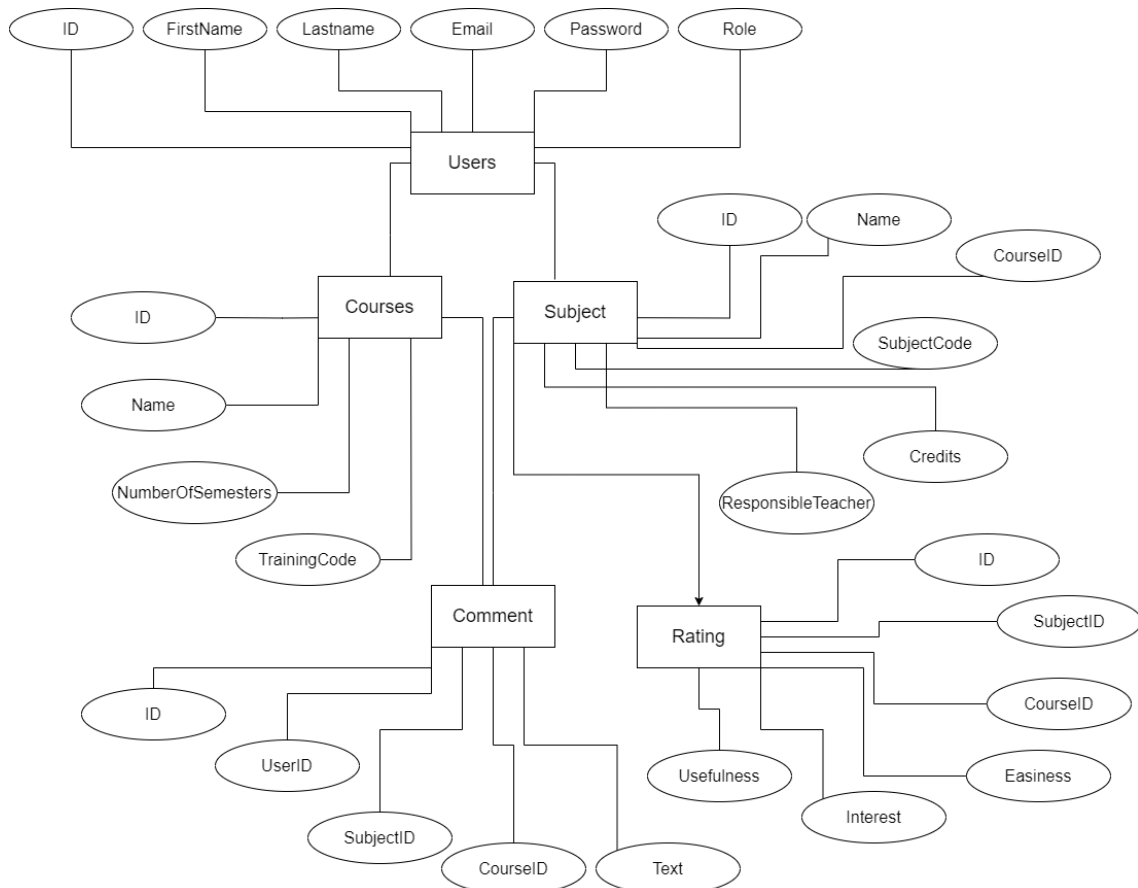
Rendszertervem egy teljes rendszertervvvel ábrázolnám, ami látható a 18. ábra Teljes rendszertervnél. Az elkészült projekt 4 fő részre bontható le, ami a Backendet, a Frontendet, az Adatbázist és a Mobil applikációt foglalja magába. Frontendem és a telefonos applikáció a backenddel kommunikál, ameddig a backend az adatbázisból kéri le a szükséges adatokat.



18. ábra Teljes rendszerterv

EK modell

Alkalmazásom egy háromrétegű architektúrát követ, ami a felhasználói interfészt (UI) réteget, az üzleti logika (backend) réteget és az Adatbázis réteget tartalmazza. A felhasználói felület vagy interfész magában foglalja az összes olyan elemet, amellyel a User találkozhat. Az UI réteg feladata az, hogy a rendszer funkcionalitását könnyen érthető, intuitív és esztétikus módon prezentálja a felhasználók számára. Ilyen elemek például a kurzusok megjelenítése, keresése, tanárok / professzorok szűrése és értékelése is.



19. ábra Rendszer egyed-kapcsolat modell

Az egyed-kapcsolat modell úgynevezett EKM az adatmodellezésben segíti a fejlesztőket. Általánosságban jól bevált módszer, amely széleskörben el is van terjedve kiváló grafikai adatok ábrázolásával és ezeknek kapcsolatainak bemutatásával.

A 19. ábra Rendszer egyed-kapcsolat modellremekül bemutatja entitásaim kapcsolatát. Teljesen pontosan 4 táblát köt össze, amelyekhez attribútumaik vannak hozzárendelve ovális alakzatban.

Adatmodell leírása

Adatbázisom táblázatos bemutatására táblázatos megjelenítést használok, amely az EKM diagramnál több információt mutat be. Táblázataim fejlécében az oszlopok neve, és típusa. Ezen felül még hozzátenném, hogy a táblázatban a tartalom sorban egy leírás is helyet fog kapni annak érdekében, hogy tisztább legyen a szerepe az attribútumnak.

User tábla

A User táblám a felhasználók adatait tartalmazza, amiben benne van az ID, FirstName, LastName, Email, Password és a Role. Az ID természetesen magától növekszik minden beszúrással.

Oszlop név	Típus	Tartalom	Hossz
ID	numeric	Elsődleges kulcs	-
FirstName	varchar	Keresztnév	50
LastName	varchar	Vezetéknév	50
Email	varchar	Email cím	180
Password	varchar	Jelszó	50
Role	varchar	Szerepkör	30

2. táblázat - Felhasználók szerkezete

Subject tábla

A subject táblám a tárgyak attribútumait tartalmazza, amely magába foglalja az ID, a Name, a SubjectCode, a Credit és a ResponsibleTeacher-t.

Oszlop név	Típus	Tartalom	Hossz
ID	numeric	Elsődleges kulcs	-
Name	varchar	Név	50
SubjectCode	varchar	Tárgy kódja	50
Credit	numeric	Megszerzendő kreditek száma	-
ResponsibleTeacher	varchar	Oktató	50

3. táblázat - Tárgyak szerkezete

Course tábla

Tartalmazza a kurzusok attribútumait, ebből a táblából fogom könnyedén elérni azokat a tárgyakat például, amelyek bizonyos kurzusokban található.

Oszlop név	Típus	Tartalom	Hossz
ID	numeric	Elsődleges kulcs	-
Name	varchar	Név	50
NumberOfSemesters	numeric	Szemeszterek száma	-
TrainingCode	varchar	Kurzus kódja	50

4. táblázat - Kurzus szerkezete

Comment tábla

Tartalmazza azokat a hozzászólásokat, amelyeket vagy kurzusokhoz, vagy tárgyakhoz fűztek.

Oszlop név	Típus	Tartalom	Hossz
ID	numeric	Elsődleges kulcs	-
UserID	numeric	Idegen kulcs a User táblához	-
SubjectID	numeric	Idegen kulcs a Subject táblához	-
CourseID	numeric	Idegen kulcs a Courses táblához	-
Text	varchar	Komment szövege	320

5. táblázat - Komment szerkezete

Rating tábla

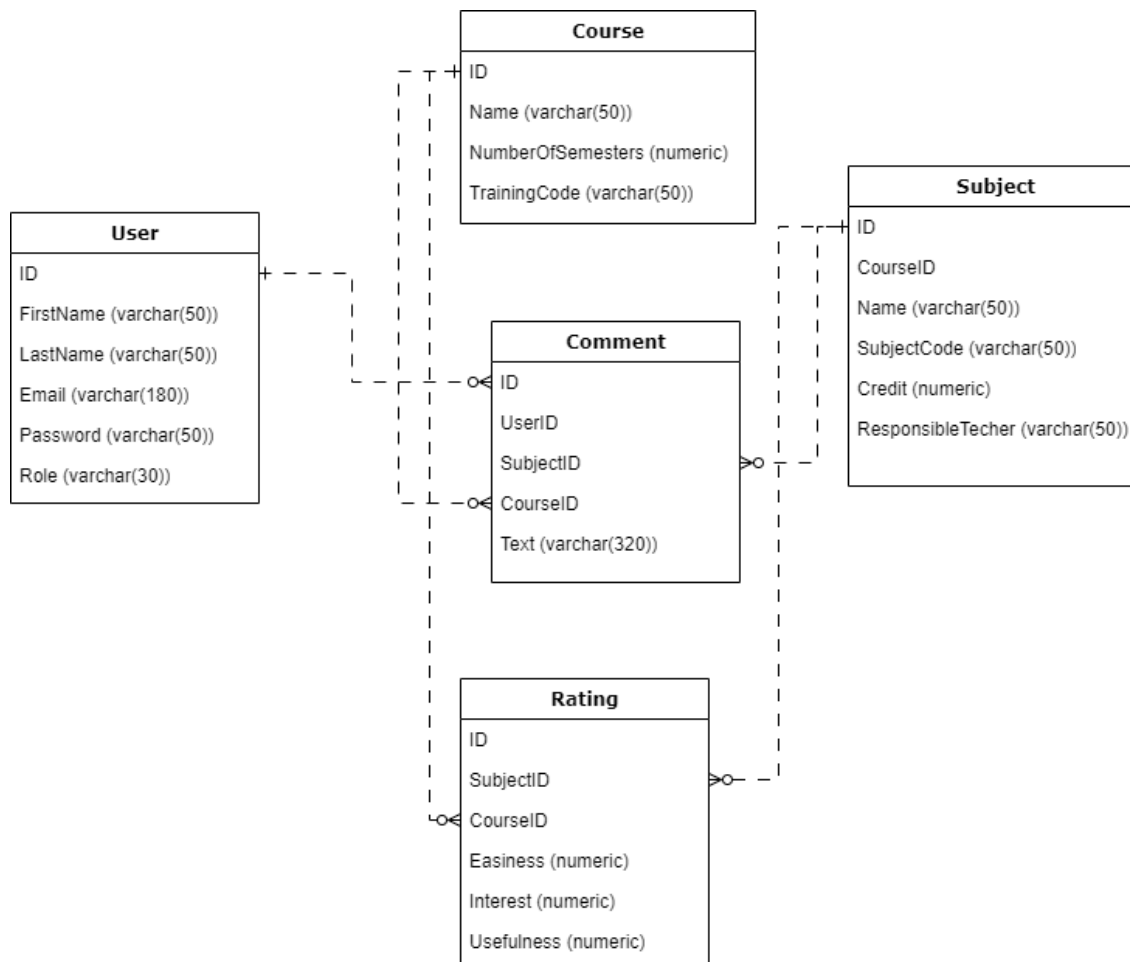
Értékeléseket tartalmazza, amelyből majd következtetni lehet, hogy egy adott kurzus, vagy tárgy mennyire lehet hasznos, érdekes, könnyű a többi hallgató szerint.

Oszlop név	Típus	Tartalom	Hossz
ID	numeric	Elsődleges kulcs	-
SubjectID	numeric	Idegen kulcs a Subject táblához	-
CourseID	numeric	Idegen kulcs a Course táblához	-
Easiness	numeric	Float érték a tárgy, vagy kurzus könnyűségének	-
Interest	numeric	Float érték a tárgy, vagy kurzus érdekességére	-
Usefulness	numeric	Float érték a tárgy, vagy kurzus hasznosságának	-

6. táblázat - Értékelések szerkezete

Adatbázis táblák kapcsolata

Adatbázisom tábláit grafikusán is megjeleníttem, hogy a fejlesztés közben innen, már könnyedén le tudjam olvasni, hogy melyik táblában milyen attribútumok vannak és ezzel felgyorsítom a fejlesztésem menetét is.



20. ábra Adatbázis táblák és kapcsolatai

4.2. Funkciólista

Szakedolgozatom ezen része az alkalmazás főbb funkcióit sorolja fel, amelyet a felhasználók az oldal használatával alkalmazni fognak. A funkciólista kidolgozása fontos lépés a szoftverfejlesztési folyamatban, mert biztosítja, hogy a fejlesztők a funkciók tekintetében ugyan azt gondolják a félreértések elkerülése végett.

Webalkalmazásom funkciói

- Bejelentkezés QR kód használatával
- Bejelentkezés ideiglenes 5 karakter beírásával
- Kurzusok listázása, keresése és szűrése
- Tárgyak listázása, keresése és szűrése
- Kurzusok értékelése
- Tárgyak értékelése
- Kurzusokhoz komment írása
- Tárgyakhoz komment írása
- Kurzusok hozzáadása
- Tárgyak hozzáadása
- Kurzusok módosítása
- Tárgyak módosítása
- Kurzusok törlése
- Tárgyak törlése
- Értékelések törlése
- Kommentek törlése
- Felhasználók törlése
- Kurzusok hozzáadása
- Tárgyak hozzáadása
- Átlagok számítása kurzusonként
- Átlagok számítása tárgyanként

Telefonos applikációm funkciói

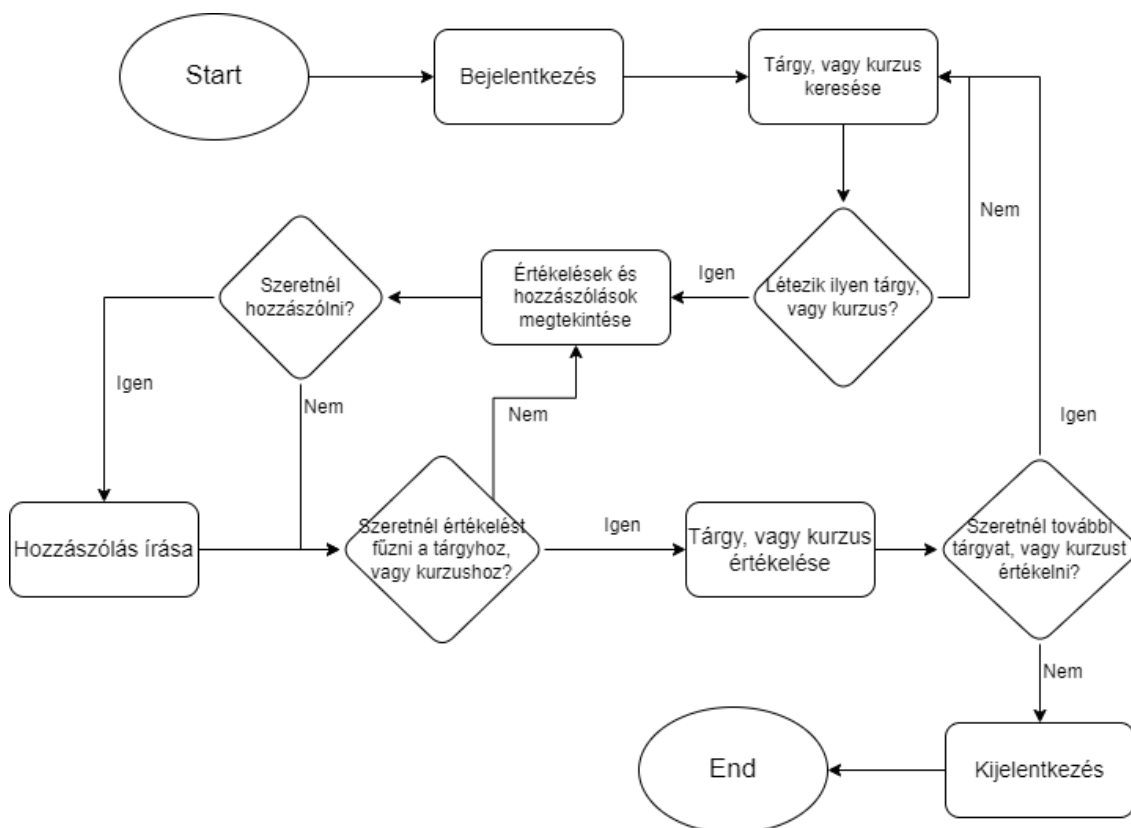
- Bejelentkezés
- QR-kód leolvasás kamerával
- 5 karakter hosszú egyszerhasználatos biztonsági kód generálása

4.2.1. Bejelentkezés menete

Bejelentkezésem az egyik legfontosabb és komplexebb funkcióm, amely elengedhetetlen a biztonságos bejelentkezés miatt. Első lépésként a felhasználó beírja az Email címét és jelszavát a bejelentkezési fülön, majd rányom a belépésre. Ezután a rendszer generál egy 5 jegyű ideiglenes kódot, amit a mobil applikációból lehet lekérni és beírni az oldalon, vagy szintűgy az applikáció segítségével le tudjuk olvasni a generált QR-kódot, ami után a rendszer beléptet minket.

4.2.2. Értékelések folyamata

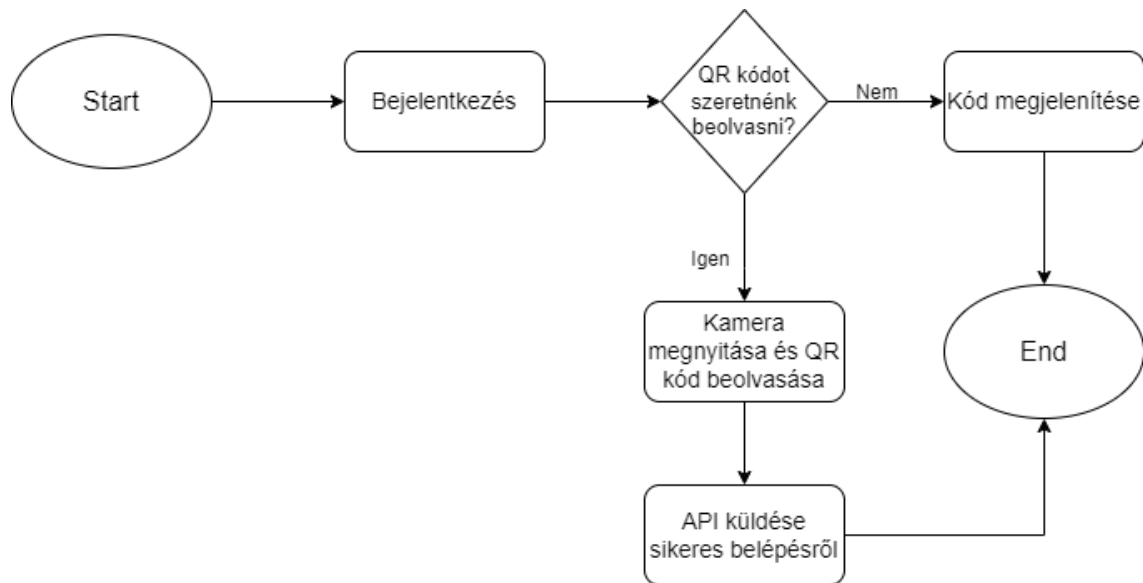
A tárgyak, vagy kurzusok értékelése alkalmazásom talán egyik lefőbb folyamata, hiszen az alkalmazás alapját képezi. Felhasználónk bejelentkezés után tárgyak és kurzusok között tud keresni és kiválasztás után lehetősége van hozzászólást írni, vagy éppen értékelni bizonyos szempontok alapján, amelyek az adatbázis ábrában megtalálhatóak a Rating táblában. Folyamatát az értékelésnek a 21. ábra Értékelés, vagy hozzászólás megadásának folyamata mutatja be.



21. ábra Értékelés, vagy hozzászólás megadásának folyamata

4.2.3. Telefonos applikáció használatának folyamata

Telefonos applikációm működése egy bejelentkezési felülettel indul, amely szintűgy várja a felhasználó email címét és jelszavát. Sikeres bejelentkezés után kettő darab gomb közül választhatunk, az első a QR kód beolvasása gomb, amely megnyitja a kamerát és a QR kód tartalmát megnyitja egy böngészőben, ami egy API hívást indít a szerver felé, ezáltal tudja majd a backend, hogy sikeres volt a hitelesítés. A másik lehetőségünk, hogy fedje fel nekünk az ideiglenes kódot, amit kiír nagyban a kijelzőre és ezt begépelve az oldalon tudjuk hitelesíteni magunkat. Ezt a folyamatot mutatja be a 22. ábra Telefonos applikációval történő bejelentkezés.



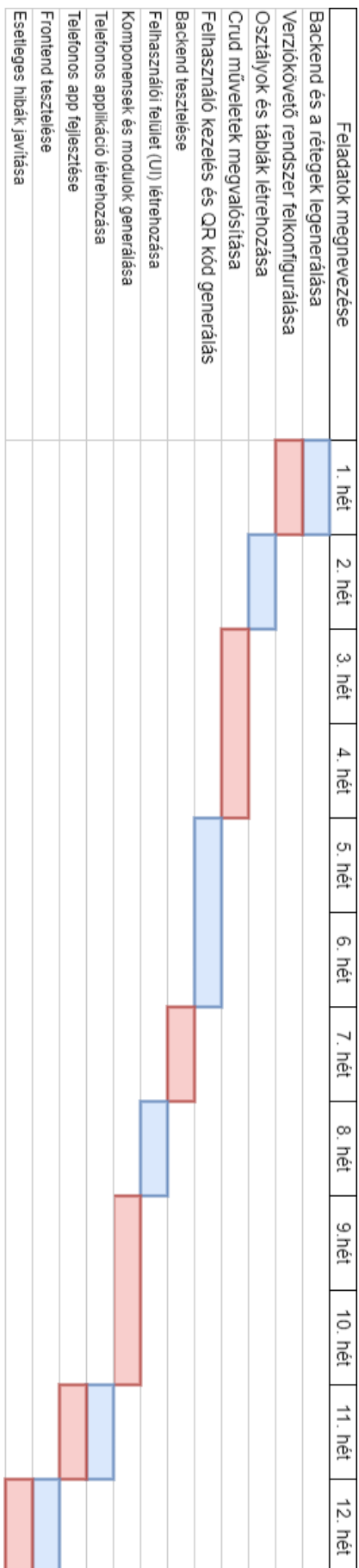
22. ábra Telefonos applikációval történő bejelentkezés

4.3. Fejlesztés tervezett menete

Fejlesztésem menete az alábbiak szerint lesz kivitelezve. Tizenkettő hét alatt kell egy kész terméket leadnom, így a felépítést ez nagyban befolyásolja. Tervezésem már kész, így fejlesztéssel kezdeném, először a backend-et szeretném felépíteni. Ez magában foglalja az entitások elkészítését, és az üzleti logikában megfogalmazott táblarendszerek megvalósítását, lekérését, módosítását, tehát röviden a CRUD-ot (create, read, update, delete). A végpontjaim tesztelését, mivel ebben az esetben még nincs interfész, amin tudnám tesztelni, a Postman nevű programmal fogom megvalósítani. Tehát ezekre a funkciókra szeretnék előreláthatólag négy-öt hetet felhasználni.

Frontendem bár a backend közben elkezdem, lényeges haladást a legfontosabb API hívások megírása és sikeres működése után tudom megvalósítani. Mivel reszponzív alkalmazásom lefejlesztése picit több időt vesz igénybe, így próbálom öt-hat hét alatt befejezni, ami a tesztelést még nem tartalmazza.

A tesztelés része pedig az utolsó simításokkal együtt fog megtörténni, az utolsó egy-két hetet már csak az alkalmazás csiszolására tartogatom, ahol a lehetséges hibákat mind frontend, mind backend oldalon fogom tesztelni, ez magában foglalja a funkcionális tesztelést, a UI/UX tesztelést, a teljesítmény felmérést, optimalizálást és egy pár automatizált tesztet is. Az automata teszteket Seleniumnal fogom megvalósítani, mert ezt már magabiztosan tudom kezelni előző munkahelyem miatt. Ezen tesztek segítenek meggyőződni arról, hogy az alkalmazás megfelelően működik, és minimalizálja a hibák és problémák esélyét. Az automatizált tesztek használata hatékonyabbá és skálázhatóbbá teszi a tesztelési folyamatot, különösen a funkcionális és regressziós tesztek terén. Tervezett fejlesztésem Gantt diagrammját a 23. ábra Gantt diagramm a tervezett fejlesztésről mutatja be.



23. ábra Gantt diagramm a tervezett fejlesztésről

5. FEJLESZTÉS

A projekt kivitelezése során több váratlan kihívással is találkoztam, amelyek próbára tették a fejlesztői képességeimet és a problémamegoldó készségemet. Az első nehézségek az Angular és .NET technológiák összekapcsolásából adódtak: míg az Angular kliensoldali keretrendszerként gyors és interaktív felhasználói felületet biztosít, a .NET backend komplex üzleti logika kezelésére és adatkezelésre szolgál. Az ezen rendszerek közötti kommunikáció és adatcsere biztosítása számos különféle elakadás elé állított, a megfelelő API struktúra kialakításától kezdve az autentikáció és jogosultságkezelés kérdésein át a teljesítmény optimalizálásáig.

Minden egyes elakadás és probléma megoldása azonban segített elmélyíteni a szakmai tudásomat, és betekintést nyertem a használt rendszerekbe, az alkalmazások tervezésébe és fejlesztésébe.

5.1. Backend

A backend fejlesztése során a fő célom az volt, hogy egy megbízható, rugalmas és biztonságos rendszert hozzak létre, amely megfelelően kiszolgálja a frontend kéréseit és támogatja a webalkalmazás teljes működését. Ehhez a .NET keretrendszert választottam, amely stabil alapot biztosít komplex alkalmazások számára, különösen az adatkezelési és üzleti logikai funkciók tekintetében.

A backend tervezésénél az első lépés a modellek, az adatstruktúra, az üzleti logika és a frontenddel való kommunikáció érdekében az API hívások megtervezése volt. Ezek tervezéséről már a szakdolgozatom korábbi fejezeteiben írtam, tehát nem most kellett elkezdenem elmélkedni ezek felépítéséről.

5.1.1. Modellek

Modelljeim alapvetően az adatok strukturált kezelésére szolgálnak, és ezek képezik a backend réteg egyik legfontosabb alapját. Egy jó modell felépítése biztosítja az adatok konzisztens kezelését, valamint könnyen karbantartható és skálázható architektúrát eredményez.

Modellek áttekintése

Comment

Ez a modell az egyes kurzusokhoz és tárgyakhoz kapcsolódó felhasználói hozzászólásokat tárolja. Az attribútumai között megtalálható a komment szövege, a felhasználó és a kurzus vagy tárgy azonosítója, amelyhez a komment tartozik.

Course

A kurzusokat leíró modell, amely tartalmazza az egyes kurzusok nevét, kódját, az oktatót, valamint az egyéb kapcsolódó adatokat, például a hozzá tartozó félévek számát és a tantárgyat. A Ratings navigációs tulajdonság lehetővé teszi az értékelésekhez való hozzáférést.

LoginModel

Ez a modell kizárólag a bejelentkezéshez szükséges adatokat tartalmazza, azaz a username és password mezőket. Ennek a modellnek egyszerű a felépítése, mivel célja a felhasználói hitelesítés alapadatainak kezelése.

Rating

Az értékelések tárolására szolgáló modell, amely lehetővé teszi a felhasználók számára, hogy egy kurzust különböző szempontok (könnyedség, érdekesség, hasznosság) szerint értékeljenek. Minden értékelés egy adott kurzushoz és felhasználóhoz kötődik.

SiteUser

Ez a modell a felhasználói adatok kezelésére szolgál. Az UserId, Email és további alapvető adatok itt kerülnek tárolásra. Ez lehetővé teszi a rendszer számára, hogy kezelje a felhasználókhoz kapcsolódó adatokat és beállításokat.

Subject

A tantárgyakat leíró modell, amely tartalmazza a tantárgy nevét és egyedi azonosítóját. A tantárgyak a kurzusok szerves részei, így a Subject modell összekapcsolódik a Course modellel.

TwoFactorModel

Ez a modell a kétlépcsős azonosítás során szükséges verificationCode mezőt tartalmazza. A LoginModel kiegészítéseként működik, és segít a biztonságosabb bejelentkezés megvalósításában, mivel a felhasználónak egy további hitelesítési kódot is meg kell adnia.

5.1.2. Rétegek felépítése

A .NET alkalmazás rétegezése egy elterjedt tervezési minta, amely a különböző felelősségi köröket különálló rétegekbe szervezi. Ez a struktúra segít átláthatóbbá tenni a kódot, és lehetővé teszi a könnyebb karbantartást és tesztelést. Az alábbiakban bemutatom az egyes rétegeket és azok szerepét, illetve néhány nehézséget, amelyekkel a fejlesztés során találkoztam.

Controllerek

Ez a réteg felelős az API végpontok megvalósításáért és a HTTP kérések fogadásáért. A controllerek a beérkező kéréseket a megfelelő logikai réteghez irányítják, majd visszaadják a választ az ügyfélnek. Controllereim hasonlóan épülnek fel egymáshoz. Az adott controller meghívja a hozzá tartozó üzleti logika (logic) interfészét, majd az abban található metódusokat hívja a különböző kérések alatt. Jelenleg egy általános controller felépítését a 24. ábra Comment controller felépítése ábrázolja megfelelően.

```
namespace studentforum_backend.Controllers
{
    [Route("[controller]")]
    [ApiController]
    1 reference
    public class CommentController : ControllerBase
    {
        private readonly ICommentLogic service;

        0 references
        public CommentController(ICommentLogic service)
        {
            this.service = service;
        }

        // GET: api/<CommentController>
        [HttpGet]
        0 references
        public IEnumerable<Comment> ListComments()
        {
            var list = service.GetAll();
            return list;
        }

        // GET api/<CommentController>/5
        [HttpGet("{id}")]
        0 references
        public Comment GetCommentById(int id)
        {
            return service.GetById(id);
        }

        [HttpGet("GetSubjectComments/{id}")]
        0 references
        public IEnumerable<Comment> GetSubjectComments(int id)
        {
            return service.GetSubjectComments(id);
        }
    }
}
```

24. ábra Comment controller felépítése

Nehézségek:

Legtöbbször elkövetett hiba, amibe beleütköztem, hogy ne keverjem a kódom helyét. Rendszeresen a controllerekbe írtam olyan üzleti logikát, amelyeket aztán commitok előtt refaktoráltam és a megfelelő helyre (logikai rétegbe) helyeztem át.

Üzleti logika réteg (Logic)

Ebben a rétegben található az alkalmazás üzleti logikája. Itt történnek a különféle adatfeldolgozások, számítások és döntéshozatalok, amelyek az alkalmazás működéséhez szükségesek. A logikai réteg interfészeket használhat, hogy a különböző logikai műveletekhez könnyen cserélhető implementációkat biztosítson.

Nehézségek:

Saját problémáim, amelyeket az üzleti logikák írása közben tapasztaltam általában a validációval voltak. El kellett dönteni, hogy amennyiben beérkezik egy objektum, akkor milyen esetben fogadjam azt el és hogy milyen buktatói lehetnek az ilyen hiányos objektumoknak. Amennyiben például valamely attribútuma hiányzik, úgy azt hogyan kezeli a modell és a program további része. Rengeteg hibába ütköztem ezek során, amely megmutatta számomra, hogy mennyire át kell gondolnom egy modell felépítését és az üzleti logikába beérkező és kimenő objektumok felépítését és validációját.

Adatbázis (Data)

A Data réteg felelős az adatbázis-kapcsolat kezeléséért és itt található a ApplicationDbContext, amely az adatbázis-táblák modellezését végzi. Ebben a részben ütköztem talán a legtöbb és számomra legnehezebb hibákba. Általánosságban nézve amennyiben a modellek jól fel vannak építve, úgy ennek a résznek a nehézsége is jelentősen csökken.

Nehézségek:

Az adatbáziskapcsolat helyes kezelése kihívás lehet, különösen akkor, ha sok kapcsolódó entitást kell kezelni. Ezen felül egy sajátos hiba is fenn állt számomra, amiben a számítógépem rendszere nem volt képes elindítani az SQL szervereket. A probléma megoldásának kutatása során több cikket is felfedeztem arról, hogy amennyiben Windows 11-es rendszert használunk, úgy ez a hiba nem is annyira egyedi, mint gondoljuk. Minél többet olvastam utána, úgy azt véltem felfedezni, hogy akiknek a számítógépükben Samsung 980 SSD van, azoknál ez még gyakoribb. Sajnos mind a kettő hibát kiváltó ok fennáll nálam, így szakdolgozatom fejlesztése során mindvégig memóriában tárolt adatbázist használtam. A hiba megoldása számomra végül egy másik számítógépről való fejlesztés volt. Összességében vagy a Windows 11-et, vagy az SSD-t kellen kicserélnem a számítógépemben a hiba megoldása érdekében.

```

0 references
public static void Main(string[] args)
{
    var builder = WebApplication.CreateBuilder(args);

    builder.Services.AddCors(options =>
    {
        options.AddPolicy("AllowSpecificOrigins",
            builder =>
            {
                builder.WithOrigins("http://localhost:4200") // Az Angular alkalmazás URL-je
                    .AllowAnyHeader()
                    .AllowAnyMethod()
                    .AllowAnyOrigin();
            });
    });

    var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");

    builder.Services.AddDbContext<ApplicationDbContext>(options =>
    {
        //options.UseInMemoryDatabase("Studentforumdb");
        options.UseSqlServer(connectionString);
        options.EnableSensitiveDataLogging(); // Adatérzékeny hibakeresés
        options.UseLazyLoadingProxies();
    });
}

```

25. ábra Adatbázisom a szervizeimmel

5.1.3. Kétfaktoros hitelesítés

Alkalmazásom biztonsága érdekében egy kétfaktoros hitelesítést implementáltam, amely egy OTP alapú QR kóddal történő azonosítást jelent. A backendem először egy normál, csupán felhasználónév jelszóval történő bejelentkezéssel szereltem fel, hogy a frontendem is tudjon haladni, majd a későbbiekben forgattam át a már megírt részeket.

Létrehoztam egy AuthService nevezetű szervízt, amely a controllerről leveszi azokat a részeket, amelyek nem tartoznak oda. Ilyenek például az olyan metódusok, amelyek az adatbázissal kommunikálnak, mert a controllerekben csupán interfészek és service-ek lehetnek importálva. Ilyen metódusok például a biztonsági kód adatbázisban való tárolása, biztonsági kód érvényesítése, jwt token generálása, biztonsági kód lekérése felhasználó alapján, kód felhasználtsági állapotának lekérdezése és használatra állítása.

Nagyobb nehézségek a kétfaktoros hitelesítés során az volt, hogy a biztonsági kódot az adatbázisban megfelelően tároljam és hogy a .NET Maui telefonos applikációmmal a QR kód leolvasása után hogyan történjen a bejelentkezés. Mérlegeltem az előttem lévő lehetőségeket és végül amellet döntöttem, hogy a frontend alkalmazásom kérdezzen rá a backendnél a bejelentkezés állapotára. Gondoltam arra is, hogy a backend hívjon vissza a frontend felé, de a rendelkezésemre álló idő alapján végül az első megoldást választottam, amely tökéletesen elvégzi feladatát.

```

[HttpPost("PreLogin")]
0 references
public async Task<IActionResult> LoginStep1([FromBody] LoginModel login)
{
    var user = await _userManager.FindByNameAsync(login.UserName);

    if (user == null)
        return Unauthorized("Invalid credentials");

    // Generating a random 5 character code
    var code = GenerateRandomCode(5);

    // Store the code temporarily (e.g., in the session or DB with a timestamp)
    _authService.StoreVerificationCode(user.Id, code);

    // Return the generated code to the frontend
    return Ok(new { VerificationCode = code });
}

```

26. ábra felhasználónév és jelszó beírása után történő backend metódus

5.2. .Net Maui telefonos applikáció

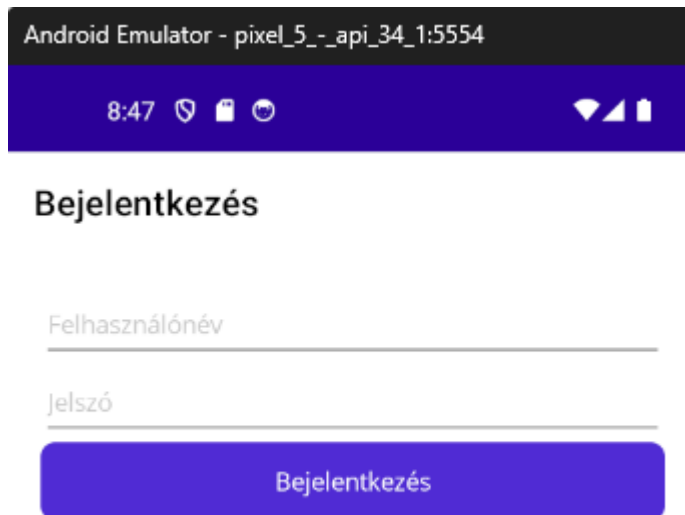
Telefonos applikációm egy olyan mobilalkalmazás, amely különféle platformokon, például Androidon és iOS-en is működik. A felhasználóbarát, egyszerű bejelentkezési felület mellett az alkalmazás két különböző hitelesítési lehetőséget is biztosít: a QR kód beolvasását és egy öt számjegyű titkos kód megjelenítését. Az alábbiakban bemutatom az alkalmazás tervezését és működését.

Alkalmazásom összesen négy darab felületből áll, egy bejelentkezési képernyőből, egy választó oldalból, ahol eldönthetjük, hogy a QR kóddal, vagy a titkos kód megjelenítésével szeretnénk tovább haladni, egy oldalból, amely a kameránkat nyitja meg és olvassa le a felületen elhelyezett QR kódot és egy felület, ahol csak középben nagyban megjelenítjük a kódot. Ezek az oldalak biztosítanak mindent, amire szükségünk van / lehet.

Problémám a megfelelő entitások importálásánál kezdődött, amikor is rengeteg lehetőség közül kellett választanom. Kutatásom során kettő darab nagyobb és megfelelő könyvtár állt rendelkezésre. Az egyik ilyen (amelyet utólag is használtam) a „ZXing.Net.Maui” nevezetű. Erről részletes leírást találtam és rengeteg hasznos oktató videót, amelyekben el tudtam mélyíteni tudásom a célom érdekében.

A másik ilyen, amely talán ugyan úgy célravezető lehetett volna a „BarcodeScanning.Native.Maui”. Erről érdemben kevesebb információ állt rendelkezésemre és a későbbiekben ez segített döntést hozni a megfelelő könyvtár használatában.

Felületeim végül egyszerűen, középre igazítva dizájnosan élénk színekkel készültek annak tudatában, hogy a modernitása minél tovább legyen releváns a mai és jövőbeli alapokhoz képest. A könyvtár használata nagyon egyszerű és érthető volt azért is, mert ekkor már több ideje tanulmányoztam a .Net MAUI alkalmazások felépítését és a 2023/2024 első félévében szerzett tapasztalatok miatt a „Web fejlesztés” nevezetű tárgy miatt.



5.3. Frontend

Angular alkalmazásom az Angular CLI (Command Line Interface) segítségével gyorsan létrehoztam. Az inicializálás után az Angular számos alapértelmezett beállítással, struktúrával és fájllal segíti a fejlesztés indulását.

- `ng new project-name`
- `cd project-name`
- `ng serve`

Ez létrehoz egy alap Angular alkalmazást, amely azonnal futtatható, és böngészőben megtekinthető. Az inicializálás során beállíthatók különböző konfigurációk, mint például a stílusokhoz használt CSS vagy SCSS fájlok, valamint az alapértelmezett routing (oldalak közötti navigáció) támogatás.

A navigációs felület kitalálása sok fejtöréssel járt, nem tudtam eldönteni, hogy használjak beépített komponenseket, amiket a PrimeNg biztosít, vagy szabjam személyre egyszerű Html Scss és Typescript fájlok segítségével. Összességében remek dolgokat kínál a PrimeNg és nagyon kényelmesen lehet használni beépített komponenseit, így általánosságban a fejlesztés során arra törekedtem, hogy azokat használjam, mert ez egy fajta egységes kinézetet is ad az oldalnak, tehát különböző elemei nem fognak elütni egymástól.

5.3.1. Felépítés és kényelmi funkciók

Navigációs felületem után következett az oldalsó navigációs menü, amely az oldalakat jeleníti meg. Itt mindenképpen egy olyan komponenst szerettem volna létrehozni, amely ki és behúzható biztosítva ezzel azt, hogy minden körülmény között elég hely legyen az oldalon megjelenített információknak. Az `app.component.html`-ben pedig felépítettem az

alkalmazás alapjait a már előbb említett komponensekből. A navbar-ban van egy `onLanguageChange()` metódus, amely a többnyelvűség állításánál fut le. Amikor a navbarban átállítjuk a nyelvet (jelenleg a magyar, angol, német érhető el), olyankor az oldal azonnal frissíti az oldalon elhelyezett szöveget anélkül, hogy az oldalt valóban újra kellene tölteni.

Frontendem építése közben próbáltam rengeteg kényelmi funkcióra gondolni, amely inkább segíti a felhasználót, mintsem hátráltatja. Ilyen például, hogy a biztonsági tokenünk, amely lejár 1 óra után vizsgálva van 5 percenként, hogy még biztosan jó-e. Rengetegszer futottam már bele olyanba, hogy szerettem volna módosítani valamit, vagy elmenteni egy adott objektumot, de a tokenem már nem volt érvényes így csak egy hibaüzenetet kaptam, éppen ezért itt nem szerettem volna hasonlót produkálni.

Fontos és véleményem szerint kiemelendő funkció még a nyelv kiválasztásának mentése, háttérszín átállítása és akár a betűméret állítása és annak mentése is. Szerettem volna, hogy a felhasználó a sajátjának érezhesse az oldalt, így készítettem egy ablakot, amelyben átállítható a háttér színe, a betűméret átállítsa és akár egy kép feltöltése is, amelyet szintűgy lementek a `localStorage`-ba és megtartok.

Táblázataim tetejére elhelyeztem egy univerzális keresőmezőt, egy új sor hozzáadása gombot, egy változott sorok mentése gombot és egy Excelbe exportálás gombot. Véleményem szerint bármikor szükséges lehet, hogy Excel fájlba mentjük a táblázatunk listás formáját, mert amennyiben esetleg meg akarjuk osztani vagy egyebek, akkor ez könnyedén megtehető és letölthető. Ehhez kapcsolódóan az oldalt elláttam éjszakai móddal is, hogy legyen lehetőségünk éjszaka szemkímélő módot beállítani.

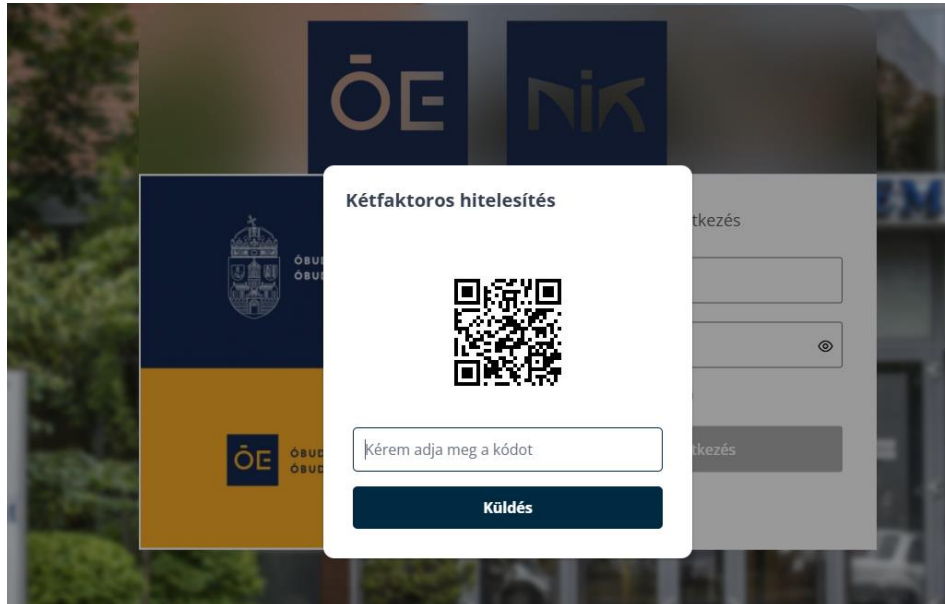
5.3.2. Főoldal

Főoldalam a belépés után, vagy a bejelentkezés kihagyása után kerül betöltésre. Szemünk elé tárul egy középre igazított ablak, amelyben megtalálható egy mező, amelyben az összes kurzus és tárgy neve szerepel. Amennyiben elkezdünk beírni bármit, úgy ajánlatokat sorol fel, hogy vajon melyikre is gondolhattunk és amennyiben megtaláljuk az általunk keresettet és rákattintunk, úgy egyből az adott tárgy vagy kurzus oldala jön fel, ahol láthatjuk értékeléseiket és azokhoz fűzött kommenteket.

Az oldalon ezen felül felsorolom a három utolsónak hozzáadott kommentet, ahhoz tartozó értékelést és a tárgy, vagy kurzus nevét is. Az értékeléseket egységesítettem egy csillaggal és átlagot számolok a háromfajta értékelési lehetőségből. Ez a panel három darab értékelést foglal magába, amely három másodpercenként automatikusan átcsúszik a következőre elemre.

5.3.3. Authentikáció

Webalkalmazásomban nem kötelező a felhasználóval történő bejelentkezés, mert az én véleményem szerint az eddigi értékelések megtekintése nem kellene, hogy felhasználóhoz kötött legyen. Ezáltal a bejelentkezési oldalon, ahova navigál minket az oldal alapból megtekinthető egy „Tovább az oldalra” felirat, amire rákattintva tudjuk használni a bejelentkezés nélküli funkciókat. Amennyiben mégis szeretnénk



28. ábra Kétfaktoros hitelesítés ablaka

bejelentkezni, úgy a felhasználónév jelszó párosítás beírása után egy új ablak ugrik be középre, amely egy QR kódot tartalmaz és egy szöveg beírására alkalmas input mezőt.

A QR kódot leolvasva a telefonos applikációm a böngészőben egy Post API hívást küld a backend rendszerem felé, amely ezt regisztrálja, majd ezeket a függőben lévő belépési kísérleteket tovább engedi. Beírt kód esetén az előbb említett kérést mi magunk küldjük a backendnek és összességében a végeredmény ugyan az.

6. TESZTELÉS

A szoftverfejlesztési ciklus egyik kiemelten fontos része a tesztelés, amely biztosítja az alkalmazás funkcionalitásának, teljesítményének és megbízhatóságának ellenőrzését. A szakdolgozatom során a .NET backend és az Angular frontend fejlesztésekor is nagy hangsúlyt fektettem a megfelelő tesztelési gyakorlatokra. A backend oldalon unit tesztek készítését mock repository-k használatával, míg a frontend teszteléséhez Selenium alapú automatizált tesztek alkalmaztam.

6.1. Backend

A .NET backend teszteléséhez unit tesztek készítését, amelyek célja, hogy az egyes komponensek (elsősorban a logikai réteg metódusainak) helyes működését izoláltan ellenőrizzék. Mivel a logikai réteg gyakran támaszkodik adatbázisműveletekre, mock repository-kat használtam az adatok szimulálására. Ezek az adatbázist szimuláló úgynevezett fals adatbázisok lehetővé teszik, hogy a valódi adatbázis elérése nélkül ellenőrizzem a logikai rétegem helyes működését.

A backend tesztelése során kiemelten fontos volt az üzleti logika, a modellek és a különböző műveletek helyességének validálása. Az alábbiakban részletezem a tesztelési folyamatokat és a főbb teszteseteket. A tesztek célja, hogy az adatbázissal kapcsolatos CRUD (Create, Read, Update, Delete) műveletek és az összetettebb logikai funkciók hibamentesen működjenek.

Tesztek szerkezete

A tesztek pozitív és negatív forgatókönyveket is tartalmaznak:

- Pozitív esetek: Sikeres CRUD műveletek és logikai funkciók validálása.
- Negatív esetek: Hibakezelés ellenőrzése hibás adatok, nem létező entitások esetén.

6.1.1. Comment Modell Tesztelése

A Comment modellnél az alap CRUD műveletek mellett több összetett logikai lekérdezés tesztelésére is sor került:

GetSubjectCommentsSuccess

Az adott tantárgyhoz tartozó kommentek sikeres lekérdezése.

Ellenőrzés:

- Az eredmény nem üres, és a várt kommentek számával egyezik meg.

GetSubjectCommentsEmpty

Nem létező tantárgyhoz tartozó kommentek lekérdezése.

Ellenőrzés:

- Az eredmény üres lista.

GetCourseCommentsSuccess

Az adott kurzushoz tartozó kommentek sikeres lekérdezése.

Ellenőrzés:

- Az elvárt mennyiségű komment érkezik vissza pontosan.

Get3LastCommentsSuccess

Az utolsó három komment lekérdezése.

Ellenőrzés:

- Az eredmény nem üres, és pontosan három kommentet tartalmaz.

Get3LastCommentsLessThan3

Kevesebb mint három komment esetén az összes komment visszaadása.

Ellenőrzés:

- Sikeresen visszaadja az utolsó kettő kommentet, mert nem létezik három

Get3LastCommentsEmpty

Üres kommentlista visszaadása.

6.1.2. Rating Modell Tesztelése

A **Rating** modell esetén szintén a CRUD műveletek tesztelése mellett összetett lekérdezések kerültek tesztelésre:

GetRatingsForCourseSuccess

Az adott kurzushoz tartozó értékelések lekérdezése.

Ellenőrzés:

- Az eredmény a várttal megegyező értékeléseket tartalmazza.

GetRatingsForSubjectSuccess

Az adott tantárgyhoz tartozó értékelések lekérdezése.

6.1.3. Egyéb Modellek Tesztelése

Az összes többi modell (pl. **Subject, Course**) esetén alapvető CRUD teszteseteket írtunk, amelyek sikeresen lefedik az adatbázis műveleteket. Minden metódus esetén biztosítottuk a sikeres és hibás esetek kezelését.

6.2. Frontend

Az Angular frontend teszteléséhez Selenium-teszteket készítettem, amelyek a felhasználói felület automatizált tesztelésére szolgálnak. Ezek a tesztek a böngésző automatizálásán keresztül biztosítják, hogy az alkalmazás különböző felhasználói forgatókönyvek szerint helyesen működjön.

A Selenium-tesztek fő célja:

- End-to-End tesztelés: Az alkalmazás teljes működési folyamatának ellenőrzése a felhasználói interakcióktól kezdve a backend hívásokig.
- Funkcionális hibák észlelése: Biztosítja, hogy az oldalak és funkciók megfelelően működnek különböző böngészők és eszközök alatt.

6.2.1. Selenium tesztesetek

Oldalcím ellenőrzése (TestPageTitle)

Ez a teszteset az oldal betöltése után ellenőrzi, hogy a böngésző címsorában megjelenő oldal cím megegyezik az elvárttal. Ez alapvető teszt, amely biztosítja, hogy az alkalmazás megfelelően betöltődött.

Ellenőrzés:

- Az oldal címe "StudentForum" legyen.

Fő komponens betöltésének ellenőrzése (TestMainContentLoad)

Ellenőrzi, hogy az alkalmazás fő komponense, az app-root, betöltődött-e. Ez kulcsfontosságú annak biztosításához, hogy az alkalmazás alapvető részei, mint például a menü vagy a navigációs fejléc elérhetőek legyenek.

Ellenőrzés:

- Az app-root komponens létezik-e.

Oldalsó menü tesztelése (TestSidenavMenuItems)

Ez a tesztet ellenőrzi az oldalsó navigációs menü meglétét, és azt, hogy a menüpontok helyesen jelennek meg. Két menüpontot várunk: "Kurzusok" és "Tantárgyak".

Ellenőrzés:

- Oldalsó menü megléte.
- Két menüpont létezik.
- A menüpontok nevei helyesek.

Navigáció a menüpontok között (TestMenuNavigation)

A teszt azt vizsgálja, hogy az oldalsó menü elemei megfelelően működnek, azaz kattintásra az elvárt oldalakra navigálnak.

Ellenőrzés:

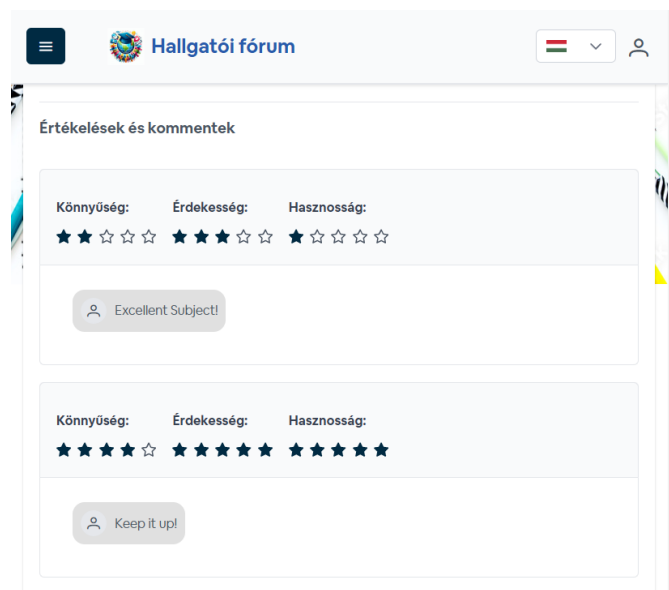
- Az első menüpontra kattintva a "/course-list" oldal töltődik be.
- A második menüpontra kattintva a "/subject-list" oldal töltődik be.

Első kurzus gomb és kártya címének ellenőrzése (TestFirstCourseButtonAndCardTitle)

Ez a tesztet ellenőrzi, hogy a kurzuslistában az első gombra kattintva a megfelelő kurzus adatai jelennek meg. A megjelenített kurzuskártyán a címnek "Kurzus_1"-nek kell lennie.

Ellenőrzés:

- Az első gomb létezik és kattintható.
- A kártya címe "Kurzus_1".



29. ábra Egy tantárgy alatti értékelések példája

Nyelvválasztó ellenőrzése zászlókkal (TestLanguageSelectorWithFlags)

Ez a teszt a navbarban található nyelvválasztót vizsgálja, amely különböző országok zászlóit jeleníti meg. Az elérhető nyelvek: magyar, angol és német.

Ellenőrzés:

- A dropdown lenyíló menü 3 elemet tartalmaz.
- Az img elemek alt attribútumai helyesen tükrözik az országokat: "Hungary", "England", "Germany".
- A kiválasztott zászló megjelenik a dropdown tetején a kiválasztás után.

Tesztek struktúrája

Az összes tesztet az NUnit keretrendszeren alapul, és a Selenium WebDriver segítségével automatizáljuk a tesztelést. A tesztek az alkalmazás különböző funkcionális elemeit célozzák meg, mint például az oldalcím, navigációs menü, tartalom megjelenítése, valamint a nyelvválasztás.

Setup és TearDown: Minden teszt előtt elindul a ChromeDriver, és betölti az alkalmazást. A tesztek futása után a WebDriver-t mindig bezárjuk.

6.3. Összegzés

A tesztelés folyamata során szerzett tapasztalatok segítettek az alkalmazás minőségének javításában. A különböző tesztelési technikák kombinálása révén biztosítottam, hogy a rendszer mind backend, mind frontend szinten megbízhatóan működjön, és ellenálló legyen a potenciális hibákkal szemben.

A könnyebb áttekinthetőség érdekében két külön táblázatot csatolok a frontend és a backend tesztek eredményeiről. Az első táblázat tartalmazza a frontend oldali tesztelési eredményeket, míg a második a backend tesztek eredményeit. Ezek a táblázatok segítenek az alkalmazás különböző rétegeiben végzett tesztelési folyamatok könnyű értelmezésében és értékelésében.

Teszt eset neve	Elvárt eredmény	Eredmény
Oldalcím ellenőrzése (TestPageTitle)	Az oldal címe "StudentForum"	✓
Fő komponens betöltésének ellenőrzése (TestMainContentLoad)	Az app-root komponens betöltődött és létezik.	✓
Oldalsó menü tesztelése (TestSidenavMenuItems)	Két menüpont legyen: "Kurzusok" és "Tantárgyak". Az oldalsó menü és a menüpontok megfelelően megjelennek.	✓
Navigáció a menüpontok között (TestMenuNavigation)	Az első menüpont a "/course-list" oldalra navigál, a második a "/subject-list" oldalra.	✓
Első kurzus gomb és kártya címének ellenőrzése (TestFirstCourseButtonAndCardTitle)	Az első kurzus kattintása után a kártya címe "Kurzus_1".	✓
Nyelvválasztó ellenőrzése zászlókkal (TestLanguageSelectorWithFlags)	A nyelvválasztó három zászlót tartalmaz: Magyarország (Hungary), Anglia (England), Németország (Germany). A kiválasztott zászló megfelelően jelenik meg.	✓

7. táblázat Frontend tesztek eredménye

Modell	Teszt Csoport	Eredmény	Megjegyzés
Comment	CRUD - Success	✓	Minden CRUD metódus sikeres esetei tesztelve.
	CRUD - Error	✓	Minden CRUD metódus hibás esetei tesztelve.
	GetSubjectCommentsSuccess	✓	Tantárgyhoz tartozó kommentek lekérdezése.
	GetSubjectCommentsEmpty	✓	Üres kommentlista lekérdezése tantárgynál.
	GetCourseCommentsSuccess	✓	Kurzushoz tartozó kommentek lekérdezése.
	GetCourseCommentsEmpty	✓	Üres kommentlista kurzusnál.
	Get3LastCommentsSuccess	✓	Utolsó 3 komment lekérdezése.
	Get3LastCommentsEmpty	✓	Nincs komment lekérhető eset.
Rating	CRUD - Success	✓	Minden CRUD metódus sikeres esetei tesztelve.
	CRUD - Error	✓	Minden CRUD metódus hibás esetei tesztelve.
	GetRatingsForCourseSuccess	✓	Kurzushoz tartozó értékelések lekérdezése.
	GetRatingsForSubjectSuccess	✓	Tantárgyhoz tartozó értékelések lekérdezése.
Subject	CRUD - Success	✓	Minden CRUD metódus sikeres esetei tesztelve.
	CRUD - Error	✓	Minden CRUD metódus hibás esetei tesztelve.
	CRUD - Success	✓	Minden CRUD metódus sikeres esetei tesztelve.
	CRUD - Error	✓	Minden CRUD metódus hibás esetei tesztelve.

8. táblázat Backend tesztéseim táblázata

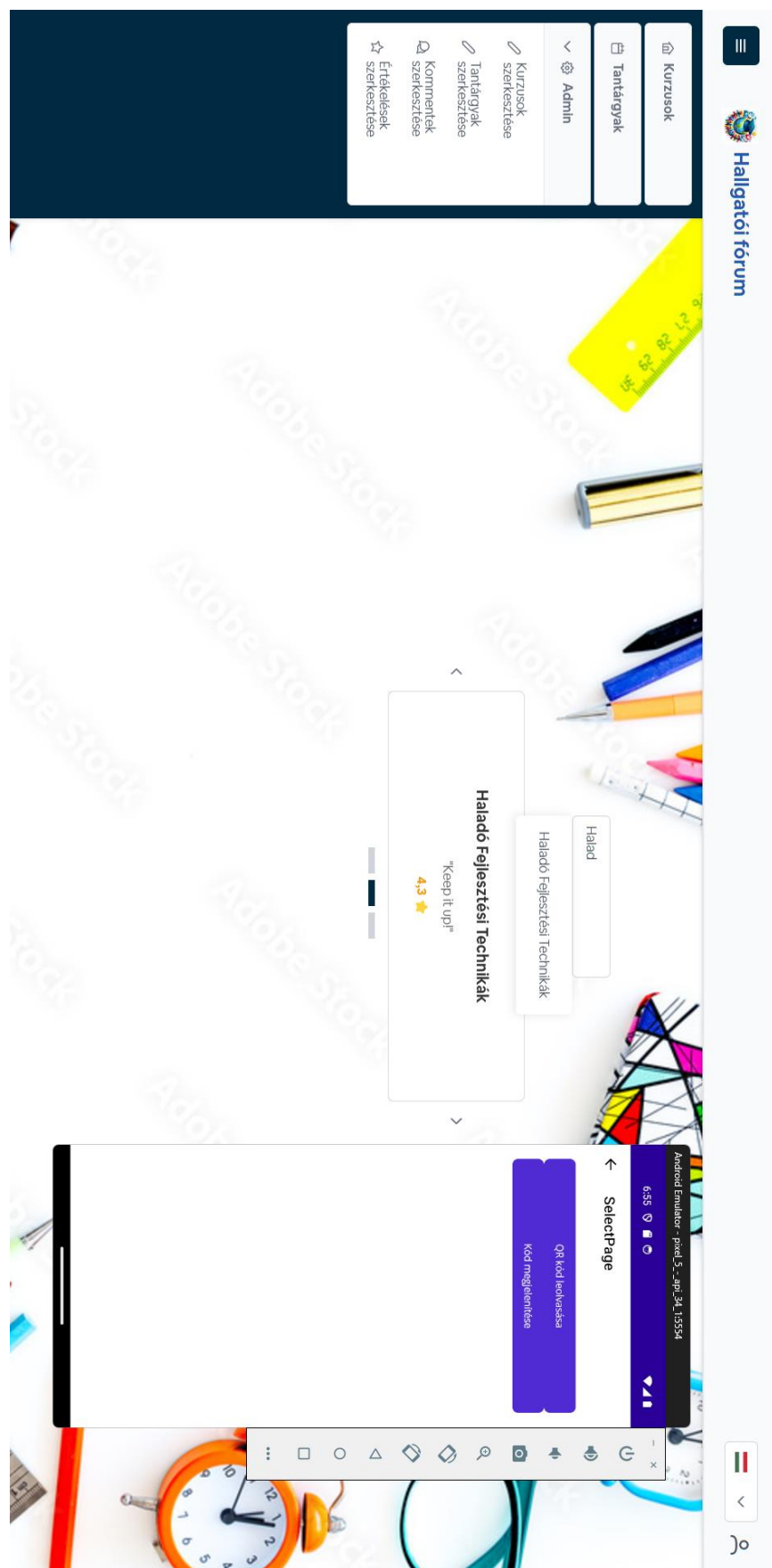
7. ÉRTÉKELÉS

Miután elkészítettem az alkalmazást és elvégeztem a szükséges teszteléseket, az alábbiakban értékelem a fejlesztés eredményeit az eredeti elvárásokhoz képest.

7.1. Eredmények bemutatása

A projekt célja egy Angular, egy .NET és egy .NET MAUI alapú webalkalmazás fejlesztése volt, amely egy online hallgatói értékelő platformként szolgál. Az alkalmazás fő funkciói közé tartozik a kurzusok, tárgyak értékelése, valamint az értékelések és hozzászólások írása és megtekintése. Az alábbi eredményeket értem el a fejlesztés során:

- **Felhasználói beléptetés és hitelesítés**
A felhasználók bejelentkezési rendszere megfelelően működik, beleértve a kétfaktoros hitelesítést. Az elvárásoknak megfelelően csak a helyes belépési adatok és a validált titkos kód megadása után kapják meg a felhasználók a hozzáférési tokenet.
- **Telefonos applikáció a hitelesítési kód lekérdezéséhez és beolvasásához**
Cross-platform telefonos applikációm képes a biztonsági kód lekérdezésére, vagy az oldalon szereplő QR-kód leolvasására és a felhasználó beengedésére.
- **CRUD műveletek implementálása**
Minden modell (kurzusok, tárgyak, értékelések, hozzászólások) esetében a CRUD műveletek sikeresen implementálva és tesztelve lettek. A tesztek során az összes metódus sikeresen végrehajtotta a kívánt műveleteket.
- **Keresési funkció és rangsorolás**
Az alkalmazás lehetővé teszi a kurzusok és tárgyak közötti keresést, valamint az értékelések alapján a legjobban értékelt tárgyak megjelenítését. A top 3 tárgy dobogós elrendezése esztétikusan jelenik meg, és az adatok helyesen frissülnek.
- **Automatizált tesztek**
Az automatizált frontend és backend tesztek biztosítják, hogy az alkalmazás minden kritikus funkciója megfelelően működjön, minimalizálva a hibalehetőségeket.



30. ábra Webalkalmazásom főoldala a telefonos applikációmmal

Studentenforum

Kurse

Fächer

Admin

Kurse bearbeiten

Fächer bearbeiten

Kommentare bearbeiten

Bewertungen bearbeiten

Haladó Fejlesztési Technikák

Durchschnittsbewertung: ★★☆☆☆

3,33 / 5

Verantwortlicher Lehrer: Kakas János

Fachcode: SBJ_1

Kredit: 3

Bewertung hinzufügen

Leichtigkeit: ☆☆☆☆☆

Interesse: ☆☆☆☆☆

Nützlichkeit: ☆☆☆☆☆

Schreibe eine Bewertung!

Einreichen

Bewertungen und Kommentare

Leichtigkeit: ★☆☆☆☆

Interesse: ★★☆☆☆

Nützlichkeit: ★☆☆☆☆

Bearbeiten

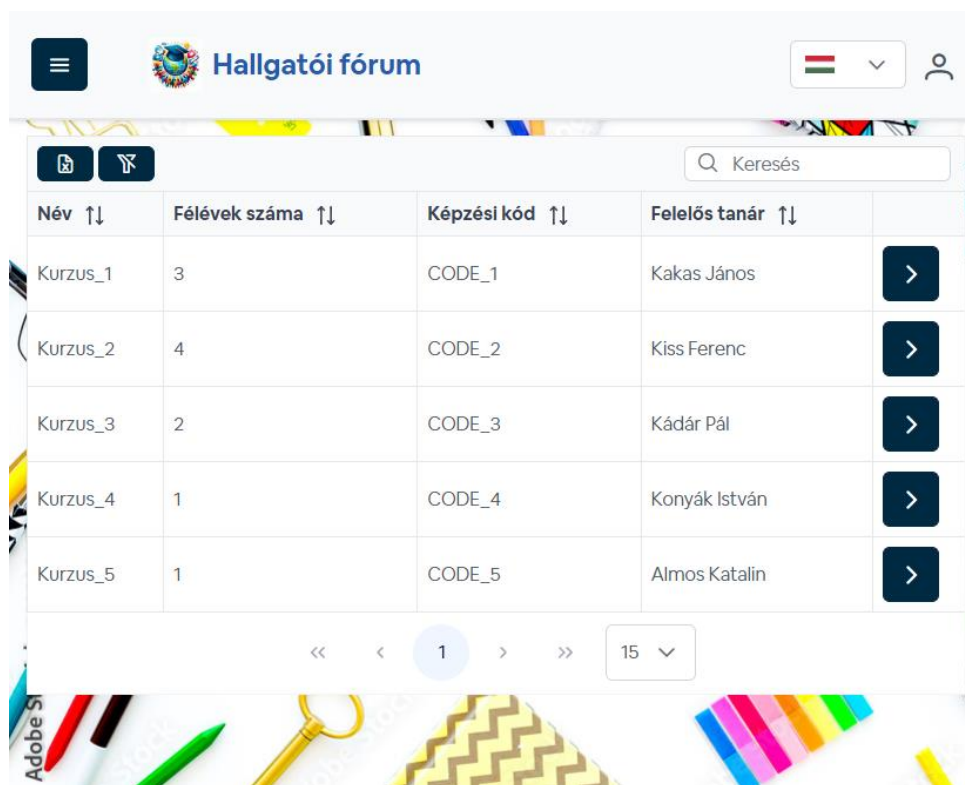
Löschen

31. ábra Haladó Fejlesztési Technikák tantárgy részletes leírása és értékelései német nyelven

7.2. Az elvárásokhoz mért teljesítmény

Pozitívumok:

- Az alkalmazás minden alapvető funkcióját sikerült lefejleszteni és megfelelően tesztelni.
- Sikeres kétfaktoros hitelesítés ahogyan a 28. ábra Kétfaktoros hitelesítés ablaka bemutatja.
- Az UI reszponzív és felhasználóbarát, különösen a keresőmező és a rangsorolt listák. Ezt remekül bemutatja a 32. ábra Kurzusok felsorolása összehúzott kijelzőn
- A backend hatékonyan kezeli a komplexebb LINQ lekérdezéseket, amelyek lehetővé teszik a gyors adatszűrést és rendezést.



Név ↑↓	Félévek száma ↑↓	Képzési kód ↑↓	Felelős tanár ↑↓	
Kurzus_1	3	CODE_1	Kakas János	>
Kurzus_2	4	CODE_2	Kiss Ferenc	>
Kurzus_3	2	CODE_3	Kádár Pál	>
Kurzus_4	1	CODE_4	Konyák István	>
Kurzus_5	1	CODE_5	Almos Katalin	>

32. ábra Kurzusok felsorolása összehúzott kijelzőn

Hiányosságok:

- Az adminisztrációs funkciók (pl. felhasználók moderálása) nem kerültek teljes implementálásra.
- Értékelések likeolása, jelenleg sajnos ez a funkció nem áll rendelkezésre, de a „Rating” tábla bővítésével ez is implementálható lenne.

Többletfunkciók:

- Minden táblázat az oldalon el van látva Excel export gombbal, amellyel könnyedén kimenthetőek az adatok a könnyebb megoszthatóság érdekében.

- Az oldal támogatja a többnyelvűséget és fel van készítve három darab nyelvre, amelyek bármikor változtathatóak, bővíthetők és könnyedén továbbfejleszthetők. A nyelvválasztásunkat az oldal eltárolja és a böngésző bezárása után is képes megjegyezni.
- Admin felületeken a táblázatokban egyszerre több sor is szerkeszthető és elmenthető.
- Az oldal kinézetének állítási lehetősége, amiben megváltoztathatjuk a betűméretet és az oldal háttérét bármilyen általunk feltöltött képpel, amelyet az alkalmazás eltárol és megjegyez.

7.3. Összegzés

Az alkalmazás az elvárásokhoz képest jól teljesített, és bizonyos területeken túl is szárnyalta azokat. A felhasználói élmény megfelelő, a rendszer stabil, és a funkciók széles köre biztosítja az alapvető működést. A jövőbeli fejlesztések során érdemes lenne a hiányosságokra koncentrálni.

8. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az alkalmazás jelenlegi verziója számos hasznos funkciót tartalmaz, azonban a fejlesztési folyamat során több olyan potenciális fejlesztési irány is felmerült, amelyek tovább növelhetnék az alkalmazás értékét és felhasználói élményét. Az alábbiakban ezek közül néhányat részletezek.

8.1. Valós idejű értesítési rendszer

Egy valós idejű értesítési rendszer bevezetése jelentősen javítaná a felhasználói élményt. Ez lehetővé tenné, hogy a felhasználók azonnal értesüljenek az új hozzászólásokról, értékelésekről vagy a kétfaktoros hitelesítési próbálkozásokról. A SignalR technológia alkalmazása ideális megoldás lehetne erre, mivel lehetővé teszi a valós idejű kommunikációt a backend és a frontend között.

8.2. Offline mód

A .NET MAUI telefonos applikáció offline módjának fejlesztése szintén hasznos lenne, különösen olyan esetekben, amikor a felhasználók internetkapcsolat nélkül szeretnék használni az alkalmazást. Az offline mód lehetővé tenné az értékelések és kommentek helyi mentését, amelyeket az applikáció később szinkronizálna a szerverrel.

8.3. Gamifikáció

A felhasználói elköteleződés növelése érdekében az alkalmazás gamifikációs elemekkel is bővíthető lenne. Például az aktív felhasználók jelvényeket és pontokat gyűjthetnének a hozzászólásokért és értékelésekért, amelyeket később különböző előnyökre válthatnának be az alkalmazásban.

Összegzés

Ezek a továbbfejlesztési lehetőségek nemcsak a felhasználói élményt növelnék, hanem hozzájárulnának az alkalmazás szélesebb körű elfogadásához és versenyképességéhez is. Ezek megvalósítása a jövőben fontos lépés lehet az alkalmazás értékének növelése érdekében.

9. ÖSSZEGZÉS

Dolgozatomban egy hallgatói fórum web- és mobilalkalmazásának fejlesztését valósítottam meg, amely a diákok közötti információcsere hatékonyságát hivatott növelni. Az alkalmazás lehetőséget nyújt a kurzusok és tárgyak értékelésére különböző szempontok alapján, miközben modern biztonsági megoldásokat alkalmaz, mint például a kétfaktoros hitelesítés.

A fejlesztés során több kulcsfontosságú eredményt sikerült elérnem. Az elkészült rendszer a webes és mobilplatformokat egyaránt támogatja, lehetővé téve a felhasználók számára a biztonságos belépést QR-kód vagy a rendszer által generált kód segítségével. A belépés után a főoldalon azonnal elérhetők a legnépszerűbb tárgyak és kurzusok, valamint egy keresőfunkció, amely lehetővé teszi a felhasználók számára az adott kurzusok vagy tárgyak gyors elérését és értékelését.

A dolgozat során bemutattam a rendszer tervezési és megvalósítási folyamatát, beleértve a .NET, Angular és .NET MAUI technológiák alkalmazását, valamint a kétfaktoros hitelesítés integrációját. Az elvégzett tesztek alapján az alkalmazás funkciói megfelelnek a kitűzött céloknak: a rendszer stabilan működik, és teljesíti az alapvető követelményeket. Bár bizonyos továbbfejlesztési lehetőségek – mint például a valós idejű értesítések vagy az offline, internet nélküli azonosítás – még megvalósításra várnak, az eddig elkészült funkcionalitás is már jelentős értéket képvisel.

A projekt során szerzett tapasztalatok nemcsak szakmai tudásomat mélyítették el, hanem rávilágítottak arra is, hogy a felhasználói visszajelzések és az iteratív fejlesztési folyamatok mennyire fontosak egy sikeres alkalmazás létrehozásában. Külön köszönet illeti konzulensemét, név szerint Sipos Miklós Egyetemi Tanársegédet és a fejlesztési folyamatot támogató online közösséget, akik eddig szerzett tapasztalataik megosztásával és szakmai tanácsaikkal hozzájárultak a projekt sikeréhez.

Összességében a dolgozat nemcsak egy konkrét rendszer megvalósításának leírása, hanem egy olyan tanulási folyamat dokumentációja is, amely meghatározó volt szakmai fejlődésem szempontjából. Reményeim szerint az elkészült alkalmazás a gyakorlatban is hasznosnak bizonyul, és hozzájárul a diákok közötti hatékony információáramláshoz.

IRODALOMJEGYZÉK

- [1] „RateMyCourse ORG,” [Online]. Available: <https://www.ratemycourse.org/>.
[Hozzáférés dátuma: 03. 03. 2024.].
- [2] „Rate my course org courses,” [Online]. Available: <https://www.ratemycourse.org/courses>.
[Hozzáférés dátuma: 03. 03. 2024.].
- [3] „RateMyCourses IO,” [Online]. Available: <https://www.ratemycourses.io/>.
[Hozzáférés dátuma: 04. 03. 2024.].
- [4] „Rate my courses io tárgyak,” [Online]. Available: <https://www.ratemycourses.io/uf>.
[Hozzáférés dátuma: 04. 03. 2024.].
- [5] „Rate my courses io értékelés,” [Online]. Available: <https://www.ratemycourses.io/uf/course/chm2045>.
[Hozzáférés dátuma: 04. 03. 2024.].
- [6] „Mark My Professor Információk,” [Online].
Available: <https://www.markmyprofessor.com/rolunk>.
[Hozzáférés dátuma: 04. 03. 2024.].
- [7] „Mark My Professor főoldal,” [Online]. Available: <https://www.markmyprofessor.com/top10>.
[Hozzáférés dátuma: 12. 03. 2024.].
- [8] „Mark My Professor Tanári adatlap,” [Online]. Available: <https://www.markmyprofessor.com/tanar/jakab-peter-23157>.
[Hozzáférés dátuma: 12. 03. 2024.].
- [9] „Mark My Professor Értékelés hozzáadása,” [Online]. Available: <https://www.markmyprofessor.com/tanar/jakab-peter-23157/velemeney-hozzaadasa>.
[Hozzáférés dátuma: 12. 03. 2024.].
- [10] „Rate My Professor Wiki,” [Online]. Available: <https://en.wikipedia.org/wiki/RateMyProfessors.com>.
[Hozzáférés dátuma: 12. 03. 2024.].
- [11] „Rate My Professor Overall Quality,” [Online]. Available: <https://www.ratemyprofessors.com/professor/1592532>.
[Hozzáférés dátuma: 19. 03. 2024.].

- [12] S. Miklós, „Szoftverfejlesztési módszerek és modellek,” [Online]. Available: <https://nikprog.hu/>. [Hozzáférés dátuma: 19. 03. 2024.].
- [13] S. Miklós, „Agilis szoftverfejlesztés,” [Online]. Available: <https://nikprog.hu/>. [Hozzáférés dátuma: 19. 03. 2024.].
- [14] „Agilis szoftverfejlesztés wikipedia,” [Online]. Available: https://hu.wikipedia.org/wiki/Agilis_szoftverfejleszt%C3%A9s. [Hozzáférés dátuma: 19. 03. 2024.].
- [15] S. Miklós, „Agilis szoftverfejlesztési alapelvek,” [Online]. Available: <https://nikprog.hu/>. [Hozzáférés dátuma: 19. 03. 2024.].
- [16] „Scrum keretrendszer,” [Online]. Available: <https://hu.wikipedia.org/wiki/Scrum>. [Hozzáférés dátuma: 19. 03. 2024.].
- [17] „Scrum teendőlista,” [Online]. Available: <https://www.visual-paradigm.com/scrum/what-is-sprint-backlog-in-scrum/>. [Hozzáférés dátuma: 19. 03. 2024.].
- [18] „TypeScript Története,” [Online]. Available: <https://en.wikipedia.org/wiki/TypeScript>. [Hozzáférés dátuma: 19. 03. 2024.].
- [19] „Typescript és javascript különbségek,” [Online]. Available: <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/>. [Hozzáférés dátuma: 19. 03. 2024.].
- [20] „Angular funkciói és felépítése,” [Online]. Available: <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>. [Hozzáférés dátuma: 28. 03. 2024.].
- [21] „Angular Routing,” [Online]. Available: <https://angular.io/guide/routing-overview>. [Hozzáférés dátuma: 28. 03. 2024.].
- [22] „.NET keretrendszer,” [Online]. Available: <https://synerinsoft.com/blog/oktatas/mi-az-a-net>. [Hozzáférés dátuma: 28. 03. 2024.].
- [23] „Microsoft .NET,” [Online]. Available: <https://learn.microsoft.com/hu-hu/dotnet/core/introduction>. [Hozzáférés dátuma: 28. 03. 2024.].
- [24] „.NET Maui,” [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0>. [Hozzáférés dátuma: 28. 03. 2024.].
- [25] „QRCode Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/QR_code.

- [Hozzáférés dátuma: 03. 04. 2024.].
- [26] „QRCoder Github Docs,” [Online]. Available: <https://github.com/codebude/QRCoder>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [27] „Git jellemzői,” [Online]. Available: <https://hu.wikipedia.org/wiki/Git>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [28] K. András, „Git alapvető fogalmak,” [Online]. Available: <https://nikprog.hu/wp-content/uploads/2022/01/modul06.pdf>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [29] „Git Workflowk,” [Online]. Available: <https://www.atlassian.com/git/tutorials/comparing-workflows>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [30] „Git Workflow példák,” [Online]. Available: <https://medium.com/jamalsoliva/git-workflows-centralized-featured-branch-and-git-flow-52ebb4efd3db>. [Hozzáférés dátuma: 03. 04. 2024.].
- [31] K. András, „Branch kategóriák,” [Online]. Available: <https://nikprog.hu/wp-content/uploads/2022/01/modul06.pdf>. [Hozzáférés dátuma: 03. 04. 2024.].
- [32] „GitHub alapok,” [Online]. Available: <https://dev.to/mahithchigurupati/github-everything-you-need-to-know-2ija>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [33] „GitHub Wikipedia,” [Online]. Available: <https://hu.wikipedia.org/wiki/GitHub>.
[Hozzáférés dátuma: 03. 04. 2024.].
- [34] K. András, „Rest API menete,” [Online]. Available: <https://nikprog.hu/wp-content/uploads/2022/01/modul08.pdf>.
[Hozzáférés dátuma: 07. 04. 2024.].
- [35] „POST API,” [Online]. Available: <https://rapidapi.com/blog/api-glossary/post/>.
[Hozzáférés dátuma: 07. 04. 2024.].
- [36] „PUT és PATCH API,” [Online]. Available: <https://www.abstractapi.com/guides/put-vs-patch>.
[Hozzáférés dátuma: 07. 04. 2024.].
- [37] „Postman,” [Online]. Available: <https://www.postman.com/downloads/>.
[Hozzáférés dátuma: 04. 12. 2023.].

Ábrajegyzék

1. ábra a Rate my course weboldal főoldala [1]	2
2. ábra a rate my courses io oldal legtöbbet értékelt egyetemei [3]	3
3. ábra a rate my courses io tárgyak felsorolása [4]	4
4. ábra rate my courses io tárgy értékelése [5]	4
5. ábra Mark My Professor főoldala a legjobban értékelt professzorokkal [7].....	5
6. ábra Mark My Professor értékelés írási felület [9].....	6
7. ábra Rate My Professor-on egy Professzor értékelése [11].....	7
8. ábra Rate My Professors értékelés felépítése [11]	7
9. ábra Scrum teendőlista felépítése [17]	11
10. ábra TypeScript generikus metódus	13
11. ábra Angular modulok importálása	14
12. ábra Angularban a routing beállítása	15
13. ábra .NET MAUI architektúrája [24]	16
14. ábra QR-kód generálása C# nyelven	18
15. ábra Feature branch workflow illusztrálása [29].....	21
16. ábra Rest API hívások működése [34]	22
17. ábra JWT Authentikáció folyamata	23
18. ábra Teljes rendszerterv	27
19. ábra Rendszer egyed-kapcsolat modell	28
20. ábra Adatbázis táblák és kapcsolatai	32
21. ábra Értékelés, vagy hozzászólás megadásának folyamata.....	34
22. ábra Telefonos applikációval történő bejelentkezés	35
23. ábra Gantt diagramm a tervezett fejlesztésről	37
24. ábra Comment controller felépítése	40
25. ábra Adatbázisom a szervizeimmel	42
26. ábra felhasználónév és jelszó beírása után történő backend metódus	43
27. ábra .NET MAUI telefonos applikáció bejelentkezési oldala	44
28. ábra Kétfaktoros hitelesítés ablaka.....	46
29. ábra Egy tantárgy alatti értékelések példája	50
30. ábra Webalkalmazásom főoldala a telefonos applikációmmal.....	55
31. ábra Haladó Fejlesztési Technikák tantárgy részletes leírása és értékelései német nyelven	56
32. ábra Kurzusok felsorolása összehúzott kijelzőn.....	57

Táblajegyzék

1. táblázat a hasonló rendszerek összehasonlításáról.....	8
2. táblázat - Felhasználók szerkezete	29
3. táblázat - Tárgyak szerkezete	29
4. táblázat - Kurzus szerkezete	30
5. táblázat - Komment szerkezete	30
6. táblázat - Értékelések szerkezete	31
7. táblázat Frontend tesztek eredménye	52
8. táblázat Backend teszteseteim táblázata	53