

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



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Számítógépes gondolkodás fejlesztése: programozási technikák.
Feladatok és ezek automatikus javítását biztosító szoftver.

DIPLOMADOLGOZAT

Témavezető:
Dr. Osztián Erika,
Adjunktus

Végzős hallgató:
Lukács Krisztina-Beáta

2023

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



UNIVERSITATEA
SAPIENTIA

Dezvoltarea gândirii computaționale: tehnici de programare.
Software pentru rezolvarea problemelor și corectarea automată a
acestora.

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Osztián Erika,
Lector universitar

Absolvent:
Lukács Krisztina-Beáta

2023

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



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Developing computer thinking: programming techniques.
Tasks and their automatic correction assurance software.

BACHELOR THESIS

Scientific advisor:
Dr. Osztián Erika,
Phd. lecturer

Student:
Lukács Krisztina-Beáta

2023

LUCRARE DE DIPLOMĂ

Coordonator științific:
dr. Osztián Erika

Candidat: **Lukács Krisztina-Beáta**
Anul absolvirii: 2022

a) Tema lucrării de licență: Dezvoltarea gândirii computaționale: tehnici de programare. Software pentru rezolvarea problemelor și corectarea automată a acestora.

b) Problemele principale tratate:

- Abilități algoritmice: înțelegerea și aplicarea conceptelor de bază ale algoritmilor
- Gândire logică și rațională
- Abilități de rezolvare a problemelor
- Corecția automată a problemelor

c) Desene obligatorii:

- Diagrama de utilizare a interfeței
- Arhitectura sistemului

d) Softuri obligatorii:

- Aplicația web, bazată pe tehnologiile PHP și JAVASCRIPT, ajută utilizatorii să exerseze cele cinci tehnici de programare: divide et impera, backtracking, greedy, programare dinamică și branch and bound. Platforma oferă o funcționalitate automată de corectare pentru sarcinile programate de utilizatori. Aceștia pot codifica și rula exercițiile cu intrările corespunzătoare, iar apoi le pot trimite pentru evaluare. În urma evaluării, utilizatorilor li se acordă punctaje și feedback pentru exercițiile rezolvate, iar rezultatele sunt stocate într-un tabel.

e) Bibliografia recomandată:

- Why php is the best choice for web development? <https://semidotinfotech.ae/blog/why-php-is-the-best-choice-for-web-development/>.
- How do html, css and javascript work together? <https://www.itonlinelearning.com/blog/how-do-html-css-and-javascript-work-together/>.
- Html vs. css vs. javascript: What's difference? <https://brytdesigns.com/html-css-javascript-whats-the-difference/>.

f) Termene obligatorii de consultații:

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca,
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, laboratorul 414

Primit tema la data de: 10.10.2021

Termen de predare: 15.06.2023

Semnătura Director Departament



Semnătura responsabilului
programului de studiu



Semnătura coordonatorului



Semnătura candidatului



Declarație

Subsemnatul/a LUKACS KRISTINA-BEATA absolvent(ă) al/a specializării
INFORMATICA, promoția 2019-2022 cunoscând
prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a
Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta
lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală,
cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de
specialitate sunt citate în mod corespunzător.

Localitatea,

Data: 06.06.2023.

Absolvent

Semnătura... Lukacs

Kivonat

A dolgozatommal segítséget szeretnék nyújtani azoknak a programozóknak, elsősorban diákoknak, akik számára a programozási technikák nehézséget okoznak. Az „Öt technikával gyakorolj” egy lokális platform, amely elsősorban magyar nyelvű és különböző programozási technikával megoldható gyakorlatokat tartalmaz. A felhasználónak szükséges a regisztráció majd a bejelentkezés a feladatok megtekintéséhez és kiválasztásához egyaránt.

Automatikus javítását biztosítja azoknak a feladatoknak, amelyeket a felhasználó lekódol, lefuttatja a megadott bementekkel, majd elküldeni javításra. Egy végső minősítésben, pontszámban nyilvánul meg a felhasználónak szánt visszajelzés a megoldott gyakorlatairól mely egy táblázatban van összefoglalva.

A felület célja, hogy segítséget nyújtson azon személyek számára, akik gyakorolni szeretnének s fejleszteni kívánják logikájukat és számítógépes gondolkodásmódjukat. Inkább azoknak a középiskolás tanulóknak vagy egyetemi hallgatóknak ajánlom, akik már rendelkeznek előzetes ismeretekkel a programozási technikákról.

Rezumat

Platforma „Exersează cu cinci tehnici” este o resursă locală care oferă exerciții și suport pentru programatori, în special pentru studenți care întâmpină dificultăți în tehnicile de programare. Această platformă conține o varietate de exerciții în limba maghiară și poate fi rezolvată utilizând diferite tehnici de programare.

Pentru a utiliza platforma, utilizatorul trebuie să se înregistreze și să se conecteze. După autentificare, utilizatorul poate vizualiza și selecta sarcinile disponibile. Platforma oferă și o funcționalitate de corectare automată a exercițiilor. Utilizatorul poate programa și codifica soluții, le poate rula cu intrările corespunzătoare și apoi le poate trimite pentru corectare.

În funcție de feedback-ul primit pentru exercițiile rezolvate, utilizatorul primește un punctaj care este înregistrat într-un tabel de evaluare. Scopul principal al platformei este de a ajuta oamenii care doresc să-și exerseze abilitățile de programare, să-și îmbunătățească logica și să-și consolideze gândirea computerizată. Această platformă este recomandată în special liceenilor sau studenților care deja au cunoștințe anterioare despre tehnici de programare.

Abstract

With this thesis I would like to help programmers, especially students, who have difficulties with programming techniques. „Practice with Five Techniques” is a local platform, which contains exercises mainly in Hungarian language and with different programming techniques. The user needs to register and log in to both view and select exercises.

It provides automatic correction of exercises that the user solves, runs with the given inputs and then sends for correction. A final rating, a score, is the feedback to the user on the exercises he has solved and is summarised in a table.

The interface is designed to help people who want to practice and improve their logic and computational thinking. It is more suitable for high school or university students who already have some knowledge of programming techniques.

Tartalomjegyzék

1. Bevezető	10
1.1. Célkitűzések	11
2. Előzetes névtelen felmérés	12
3. Létező weboldalak felkutatása	16
3.1. LightOj	16
3.2. HackerEarth	17
3.3. Coderbyte	17
4. Szoftverkövetelmények	18
4.1. Felhasználói követelmények	18
4.2. Rendszerkövetelmények	19
4.2.1. Funkcionális követelmények	19
4.2.2. Nem funkcionális követelmények	20
5. Weblap felépítése	23
5.1. A rendszer architektúrája	23
5.1.1. Adatbázis leírása	24
5.1.2. Az adatbázis struktúrája	26
5.2. Felhasznált technológiák	27
5.2.1. PHP	27
5.2.2. HTML, CSS és JavaScript	27
5.2.3. Bootstrap	27
5.3. Felhasznált fejlesztői környezet	28
5.3.1. Visual Studio Code	28
5.4. Felhasználói felület bemutatása	28
5.5. Implementáció	37
5.5.1. Automatikus javítás	37
Összefoglaló	40
5.5.2. További fejlesztések	40
Ábrák jegyzéke	41
Irodalomjegyzék	42

1. fejezet

Bevezető

A mai világban az internet használata elengedhetetlen az ember életében mivel a technikai eszközök körülvesznek minket, amit nap mint nap használunk. Az emberek nem csak információkeresésre, maguk szórakoztatására, kapcsolattartásra, s kommunikációra használják az internetet, hanem az új dolgok elsajátítására is. Gyakran előfordul, korosztálytól és szakmai beállítottságtól függetlenül, hogy egy új információ megkereséséhez a világhálót veszik igénybe. A tanulás a diákok számára ma már szinte teljes egészében megoldható az ezekre szolgáló platformok segítségével, mint például:

- [Marketing](#) [1]
- [Zana.tv](#) [7]

Az iskolai és az egyetemi tudás elsajátításához, a diák számos tantárgy követelményeit kell megismerje. Egy olyan alapismeretre kell szert tennie, amely ezáltal az általános műveltséget biztosítja számára. Ez magával hozza, hogy mivel a diákoknak, hallgatóknak egyre több anyagot kell egyszerre megérteniük, és begyakorolniuk, túlterhelődnek.

Az informatika az egyik olyan tantárgy a sok közül, amely sokoldalú, szerteágazó s rengeteg információval rendelkező tudományág egyesítése. Ezért a tantervekben is számos fejezet tárul a diák elé, amellyel meg kell küzdenie. Az informatikában az algoritmusokra kell építkezzünk. Egy algoritmus utasítások sorozata, amelyre nincs egy recept, amit követnünk kell. Pont ez a sokszínűség, ez a sokféle megoldás adja meg e tudományág kivételességét, egyediségét, s ettől lesz érdekfeszítő és szórakozgató. Azonban csak azok a diákok látják meg egy tantárgy szépségét, akiknek sikerül elsajátítaniuk ezt az algoritmikus gondolkodást.

Számos kutatás bizonyítja, hogy a diákok algoritmikus gondolkodásának fejlesztése a gyakorlat nyújt megoldást. A tanulók azonban egyre kevesebb időt szánnak erre. A gyakorlatoknak az lenne a céljuk, hogy rögzüljön a diák elméjében az új tananyag. Gyakorlás közben szoktak a diákok ráeszmélni, hogy bár úgy gondolták, hogy megértették a tananyagot ez mégsem így történt. Ezzel különösen az informatika tantárgynál, a programok megírásakor szembesül a diák.

A programozási technikák gyakorlati része is temérdek sok nehézséget, problémát okoz a tanulóknak. Mivel nincs mindig idő az órán begyakorolni, ezért sok esetben a diáknak otthon kell befejeznie a feladatát. A diák azonban gyakran elhanyagolja a házi feladatát, idő hiányában vagy sokszor a kellő tudás mennyiség, meg nem értés hiányában.

A dolgozatomban ezeknek a problémáknak megoldására segít. Egy olyan szoftvert építettem fel, mely a tanult programozási stratégiákkal megoldható feladatokat szemlélteti, lehetőséget nyújt ezek megoldására, automatikus javítását biztosítja a megoldásoknak, pontozza ezeket és segítséget is nyújt, biztosítva az algoritmus pszeudokód változatának elérhetőségét.

1.1. Célkitűzések

Szakedolgozatomban központi témája a programozási technikák s ezek automatikus javítását biztosító szoftver.

A célja egy magyar nyelvű platform megvalósítása, amiben az öt stratégiával megoldható feladatok leprogramozhatóak s ennek következtében fejlődhet az algoritmikus gondolkodásmódjuk is.

A magyar nyelvű weboldal rengeteg segítséget nyújtana, hiszen hatékonyan, könnyedén, szinte játékosan lehetne elsajátítani egy tananyagot, mindenki a saját anyanyelvén tud a legjobban. Sikerült megvalósítani annak lehetőségét, hogy a diák kiválaszthassa azt a programozási nyelvet, amelyet tanulmányai során elsajátított, ami a legközelebb áll hozzá:

- C++
- C
- JAVA

A weboldal erőssége abban is rejlik, hogy lehetőség van keresni az oldalon, szűrni a feladatokat szakszó szerint, nehézségi szint szerint, sőt ezek megjelenítések nagysága is állítható.

Az automatikus javítás visszajelzést biztosítana a tanuló számára, ami segítené az általa megírt kód véglegesítésében.

Egy újabb erőssége a weboldalnak, hogy a megírt kódját le tudja tölteni a saját gépére, fel tud tölteni egy előre megírt kódot, tudja módosítani ezt. Mindezek és a kapott pontszám motiválja a további tanulásban.

Megszeretném megemlíteni, hogy a weboldal különböző böngészőkben is megnyitható.

2. fejezet

Előzetes névtelen felmérés

A szakdolgozatom létrehozása elején készítettem egy felmérést középiskolások és egyetemisták körében, ennek az volt a szerepe, hogy megerősítse a weboldalam szükségességét, hasznosságát. (2.2 ábra) A felmérés névtelenül történt s mindenikük egyetértett a válaszaik névtelen feldolgozásában.(2.1 ábra)

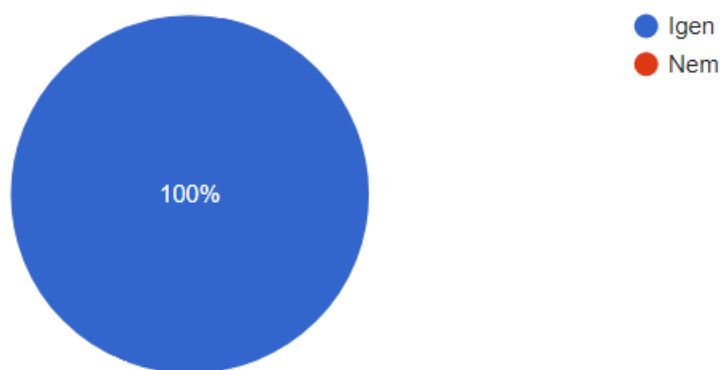
Azt is szerettem volna felmérni, hogy mely programozási technikákat ismerik inkább (2.3 ábra), melyik programozási nyelvekben kódolnak szívesen és többet (2.4 ábra). Ismernek-e olyan felületet, amely automatikus javítást biztosít s ha igen az milyen nyelvű volt.(2.5 ábra)

A kérdésekre adott válaszaik meggyőztek abban, hogy segítene-e a gyakorlásban egy online magyar nyelvű weboldal, amely biztosítja a feladatok automatikus javítást s ezek pontozását is.(lásd a 2.6 ábrán)

Ezen felmérésen 2.7 ábrán figyelhető meg a válaszadók eloszlása fiú-lány felbontásban.

Beleegyezel-e, hogy névtelenül felhasználjam ezen a felmérésen a válaszaidat?

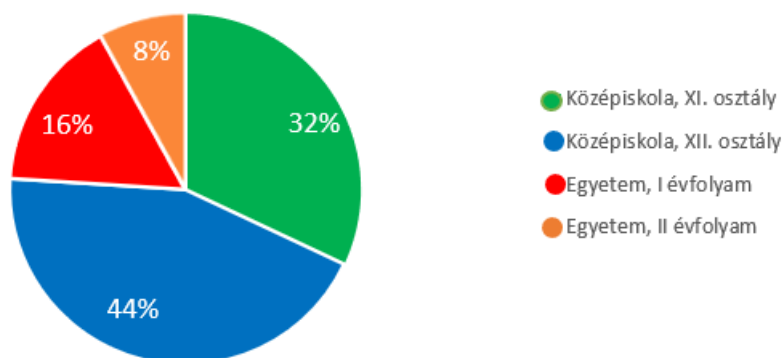
50 válasz



2.1. ábra. Az adatok névtelen feldolgozása

Hol tanulsz?

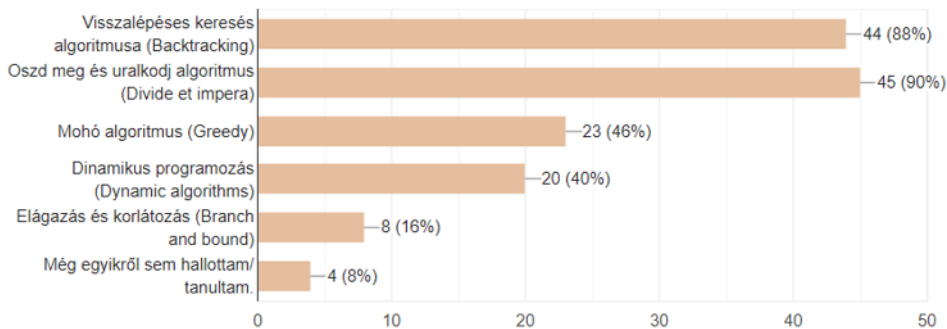
50 válasz



2.2. ábra. Célcsoport

Jelöld be milyen programozási technikákról hallottál/ esetleg tanultál?

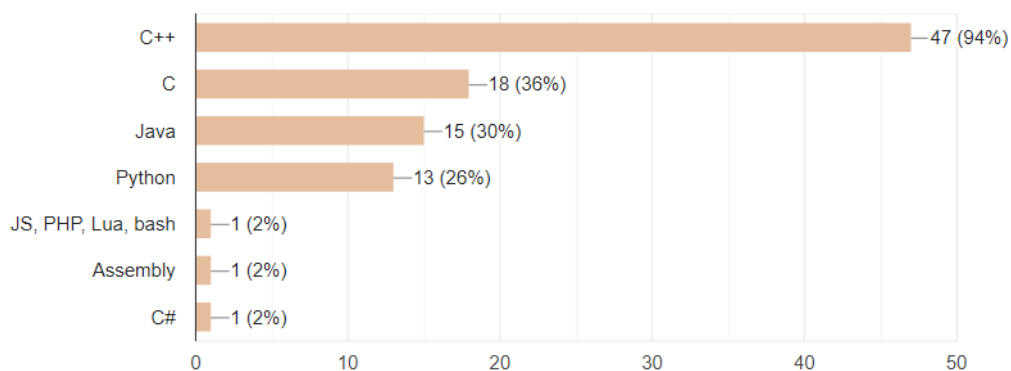
50 válasz



2.3. ábra. Öt programozási technika felismerése

Melyik programozási nyelvet ismered?

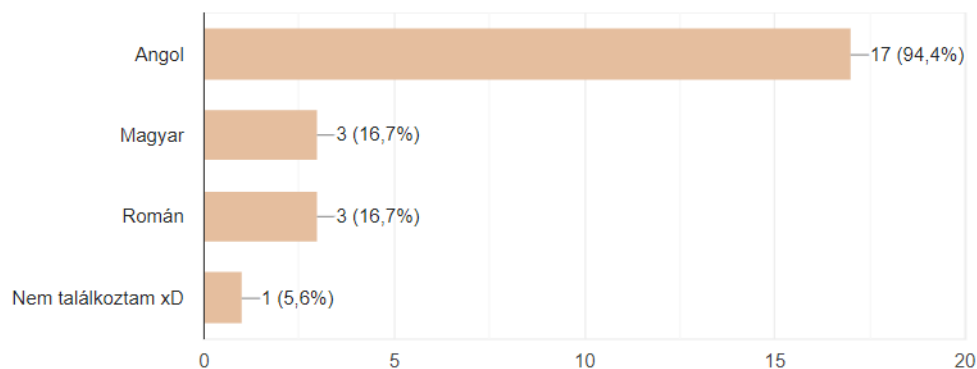
50 válasz



2.4. ábra. Programozási nyelv használata

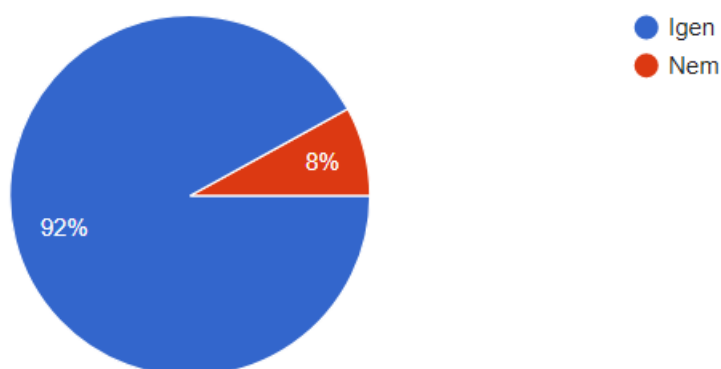
Ha igen akkor milyen nyelvű volt?

18 válasz



2.5. ábra. Automatikus javítást biztosított szoftverek

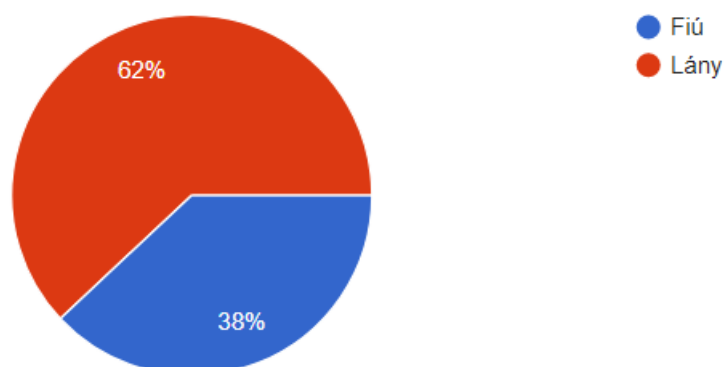
50 válasz



2.6. ábra. Magyar nyelvű automatikus javítást biztosított szoftver

Nemed?

50 válasz



2.7. ábra. Nemek szerinti felbontás

Az előzőleg említett felmérés megerősített abban, hogy szükségszerű egy ilyen felület megtervezni, létrehozni s megvalósítani. Útmutatást s irányt adott ennek a platform fejlesztésében és végrehajtásában is.

Ez a szoftver a [2.2](#) ábrán látható célcsoport hasznára, segítségére válna. Nemcsak feladat lekódolási lehetőséget biztosít, hanem azonnali kód lefuttatást is, és egy olyan környezetet, ami visszajelzést ad a felhasználónak . (lásd [2.5](#) ábrát)

3. fejezet

Létező weboldalak felkutatása

Mielőtt nekifogtam volna ennek az új oktatási környezet megvalósításának, a felmérés elemzése után magam is felkerestem azokat a webalkalmazásokat, amelyek a fő funkciója hasonló ahhoz, amit megszeretnék valósítani, fejleszteni. Többnyire angol nyelvű webalkalmazást találtam, ezeket fogom bemutatni a következő alfejezetekben.

3.1. LightOj

A [LightOj](#) egy online elérhető platform, amely a feladatok automatikus javítását is biztosítja [8] [9] .

Jane Alam Jan hozta létre 2011-ben, majd továbbfejlesztette s így 2017-ben jelent meg a végső változata, amely egy méretezhető s jobb felhasználói felülettel rendelkezett. A háttér rendszerét Node.JS, Go, és Pythonnal fejlesztették.

Kategorizálva vannak a feladatok, megtalálható az öt stratégia közül négy s még e mellett más feladatok témakörei is elérhetőek, mint például adatstruktúra, adatbázis, matematikai, mátrix, gráfok s még pár kategória. Minden feladat kijavításához be kell jelentkezni s a feladatok leprogramozása is több nyelven történhet. A bejelentkezett felhasználónak meg van jelenítve hány feladat van összesen mindegyik kategóriából, s hányat oldott meg.

A felhasználó egy statisztikai összeggést lát egy kör diagramon, amin ki van írva, hogy hány százalékban adott rossz választ, volt-e futás idejű hibája s hány volt elfogadott. Az oldalon megtekinthető a közelgő rendezvények, versenyek, amelyeken részt vehet.

Figyelemre méltó, hogy rangsorolva vannak a bejelentkezett felhasználók a megoldott feladatokra kapott pontszámok szerint és ott található a felhasználó neve is.

A hátránya az én weboldalammal szemben az, hogy itt ezen a felületen a feladatoknak meg van jelenítve a megoldása is, míg az enyémnél csak bizonyos idő után érhető el, mivel a tanulónak kell először egyedül megpróbálnia.

3.2. HackerEarth

A [HackerEarth](#) egy fejlesztői szoftver, amely segítséget nyújt a vállalatoknak az online kódolási teszteken keresztül megmérni a fejlesztők készségeit, ismereteit.[10]

A HackerEarth 2012-ben indult el az Egyesült Államokban és olyan szoftvermegoldásokat kínál, amelyek segítenek a munkaadóknak abban, hogy azonosítani, átvizsgálni és kinevezni tudják az IT-fejlesztőket.[11]

A felületen témákba vannak sorolva az oktatóanyag a gyakorlati problémával együttesen, mint például az alapvető programozás, adatstruktúra, algoritmusok, amiben az öt technikából két technika megtalálható, gépi tanulás, Python. Mindegyik téma változó feladatszámú.

A felület biztosítja a feladatok automatikus javítását, de csak a bejelentkezés után. Minden feladathoz megjelennek a hasonló de más problémák is s több mint tíz programozási nyelven le is lehet őket programozni. Más felhasználók által beküldött feladatmegoldások tekintetűek meg egy-egy feladatnál. A felhasználó, a feladat helyes megoldása után száz pontot fog kapni, ez a legnagyobb, legmagasabb pontszám. Ezen a felületen is vannak kihívási versenyek s ranglisták.

A feladatok a bejelentkezett felhasználók számára ingyenes. A hátránya a felülettel szemben az, hogy nincs minden feladatnak egy megoldása feltüntetve, míg az enyémben ez három nyelven történik s mindegyiket megkaphatja egy bizonyos idő után.

3.3. Coderbyte

A [Coderbyte](#) egy olyan weboldal, ami segít a felhasználónak a sok valós probléma, feladat megoldásában, amelyekkel más programozók is szembesültek, s szembesülnek. Több mint háromszáz feladattal rendelkezik, amely tizenhat különböző nyelven leprogramozható. [12] [13]

Kétmilliónál több felhasználó által beküldött megoldást is megtekinthet a webhelyen. Külön címek által vannak felsorakoztatva a feladat témakörei s e szerint lehet kiszűrni is őket miután regisztrált s belépett az oldalra.

Számtalan témakör látható, megtekinthető s ezek közül megemlíteném a titkosítást, a tömböket, a mátrixokat és az öt módszer közül csak a dinamikus programozás található meg. Programozási nyelveken, mint például a HTML is megtalálható egy-egy feladat.

A feladat beküldése során, ha a helyes eredménye lett akkor egy kedves üzenetet, dicséretet, valamint tíz pontot, s érmet is szerez.

Felhasználóként kevés feladatot lehet lekódolni, leimplementálni, mert prémiumos, azaz kell fizetni érte. Ebben különbözik az általam tervezett s megvalósított weboldaltól, mert az enyém ingyenes s költségmentes. A „programozási nyelveket is meg lehet tanulni egy hét” alatt ezzel a névvel ellátott kihívások, videók és feladatok is megtalálhatóak ezen weblapon, de ez is korlátozva van a fizetés miatt, ahogyan az interjú kérdések s a hozzá tartozó feladatok is.

4. fejezet

Szoftverkövetelmények

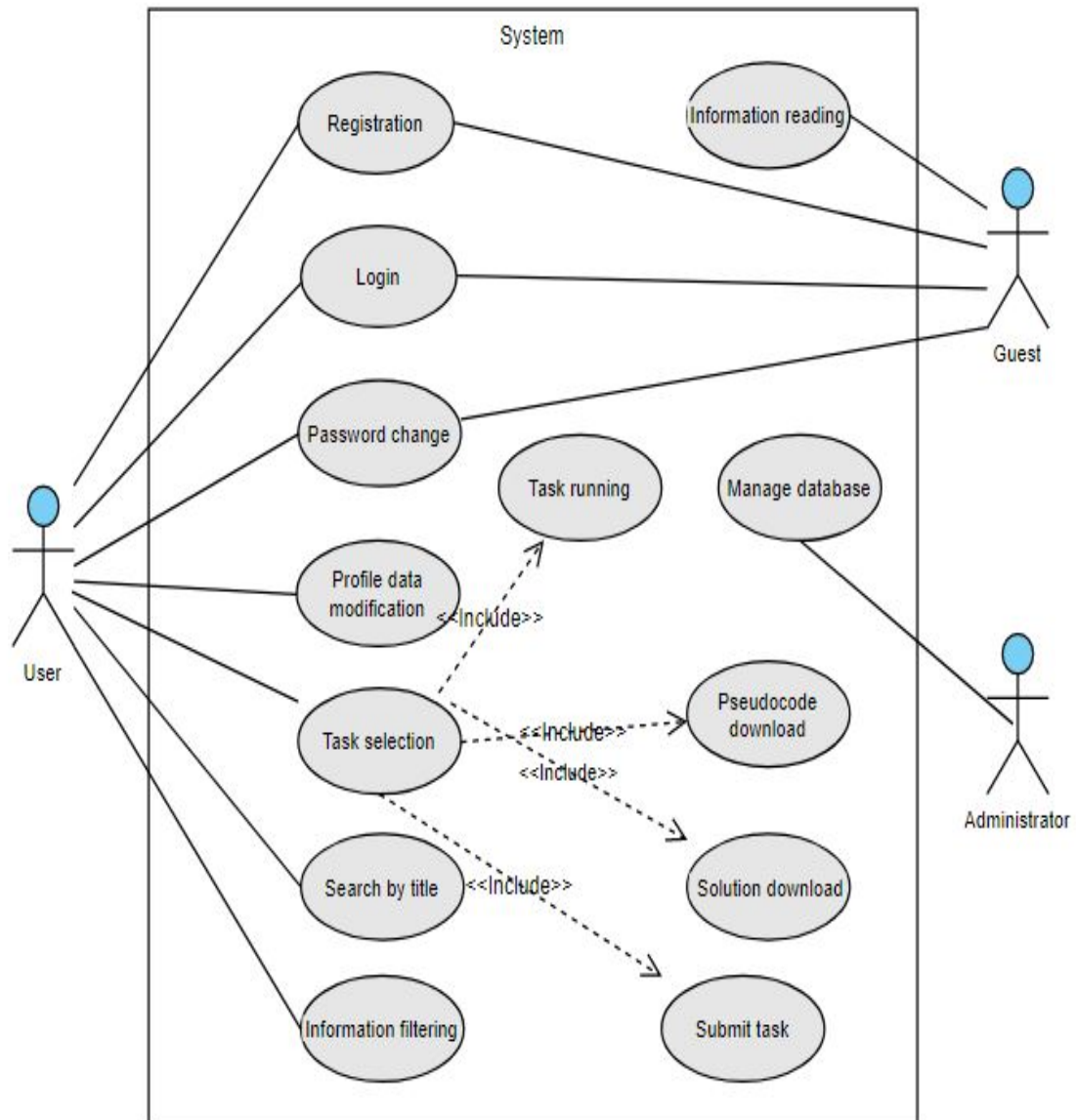
4.1. Felhasználói követelmények

A felhasználói követelmény során készítettem egy használati esetdiagramot (use case diagram) s ez a 4.1 ábrán látható megjelenítve. A felületnek három felhasználója van: a vendég, a felhasználó és az adminisztrátor.

A vendég csak a főoldalt látja, amely arról tájékoztatja, hogy milyen programozási nyelveket használ a kitűzött feladatok kódolásakor. Lehetősége van regisztrálni, mely során szerepköre megváltozik: felhasználóként dolgozhat tovább.

A felhasználónak lehetősége van megtekinteni a feladatokat, választani ezek közül vagy szűrni őket. Joga van bizonyos fájlok letöltéséhez is. Természetesen elérhető számára az általa kiválasztott programozási környezet, amelyben megoldhatja a feladatot, elküldheti ellenőrzésre, javításra, melynek eredményét pontszámokban is megtekintheti. Minden felhasználó csak a saját pontszámát láthatja.

Az adminisztrátor csak az adatbázishoz fér hozzá s azt tudja létrehozni, módosítani, törölni.



4.1. ábra. A felület használati esetdiagramja

4.2. Rendszerkövetelmények

4.2.1. Funkcionális követelmények

A vendég elolvashatja a főoldalon található információkat, választási lehetősége adódik, hogy ő is felhasználó szerepörbe kerüljön.

A felhasználónak a regisztráció, bejelentkezés, és az elfelejtett jelszó során a beírt adatai ellenőrzésre kerülnek az adott mező megnevezése szerint. Meg kell adni egy vezetéknév, keresztnév, e-mail címet, jelszót, ami 8 karakterből kell álljon, s be kell jelölje, hogy elfogadja a feltételeket. Továbbá a „Regisztráció” nyomógombra kattintva az adatok feltöltődnek az adatbázisba.

A belépés oldalon megjelenik az „Elfelejtett jelszó” gomb is, amivel megváltoztathatja a jelszót. Ezen a weblapon azzal a regisztrációnál használt e-mail címet és jelszót kell beírni a mezőkbe s a „Bejelentkezés” nyomógombra klikkelve azonnal látható a kezdőlap (lásd a 4.2 ábrán, ahol ábrázoltam a bejelentkezésnek a menetét), amin üdvözlí a felhasználót.

A menün megjelenik a „Technikák”, és a „Pofilom” oldalak a „Regisztráció” és a „Bejelentkezés” pedig eltűnik. A stratégiák közül kiválasztható egy, ami a menüben tekinthető meg. Ezen oldalon feladatok láthatóak, melyekből lehet válogatni, de keresés által, s szűrés során is megoldható egy feladat kiválasztása.

A felhasználó a „Megoldás most” nyomógommbal eljut a feladat oldalára, ahol a „Pszudokód letöltése” -re kattintva lementhető, „C/C++/Java kód letöltése” -el pedig a feladvány kódolt megoldása tölthető le. A „Futtatás” -ra klikkelve lefuttatható. A kód feltölthető, letölthető, s beküldhető a „Beküldés” -el ellenőrzésre és javításra is.

A „Profilom” -ra kattintva a felhasználó megtekintheti a saját, regisztrációnál használt adatait és a „Módosítás” gombbal módosíthatja őket. Ennek a tevékenységnek az ábrázolása látható a 4.3 ábrán. „Feladatok pontszámai” pedig maga a pontszámokat tartalmazza feladatokra, programozási fájlkiterjesztéssel és feltöltési dátummal kiegészítve.

A „Kijelentkezés” -re kattintva pedig kikerül arra az oldalra, amit a vendég lát.

Az adminisztrátor az adatbázisban dolgozik, feltölthet új adatokat s módosíthatja s törölheti a meglévőket.

4.2.2. Nem funkcionális követelmények

A weboldal igénybevételéhez jól kell működjön az internet kapcsolat. Mivel ez egy platform ezért fontos, hogy egy laptopon vagy számítógépen használják. Ezeken a technikai eszközökön Windows operációs rendszer szükséges, és egy bármilyen böngésző például Chrome, Microsoft Edge, amivel megnyitható.

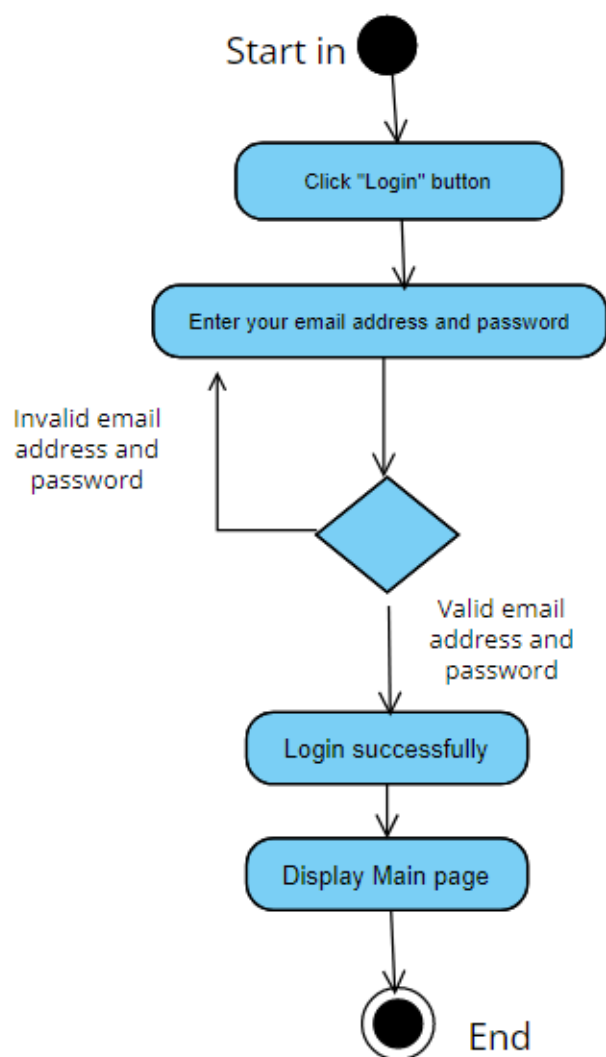
A felhasználók adatainak az eltárolására a MySQL adatbázis a felelős. Mivel ez a weblap még lokálisan működik ezért szükséges egy webszerver az XAMPP is, ami lehetővé teszi a felület lefuttatását s frissítését a böngészőben.

A feladatoknál megnyilvánul egy fordítóprogram is, ami ugyancsak lokálisan működik mivel ehhez szükséges a cygwin, jdk programozási és futtatási környezetek.

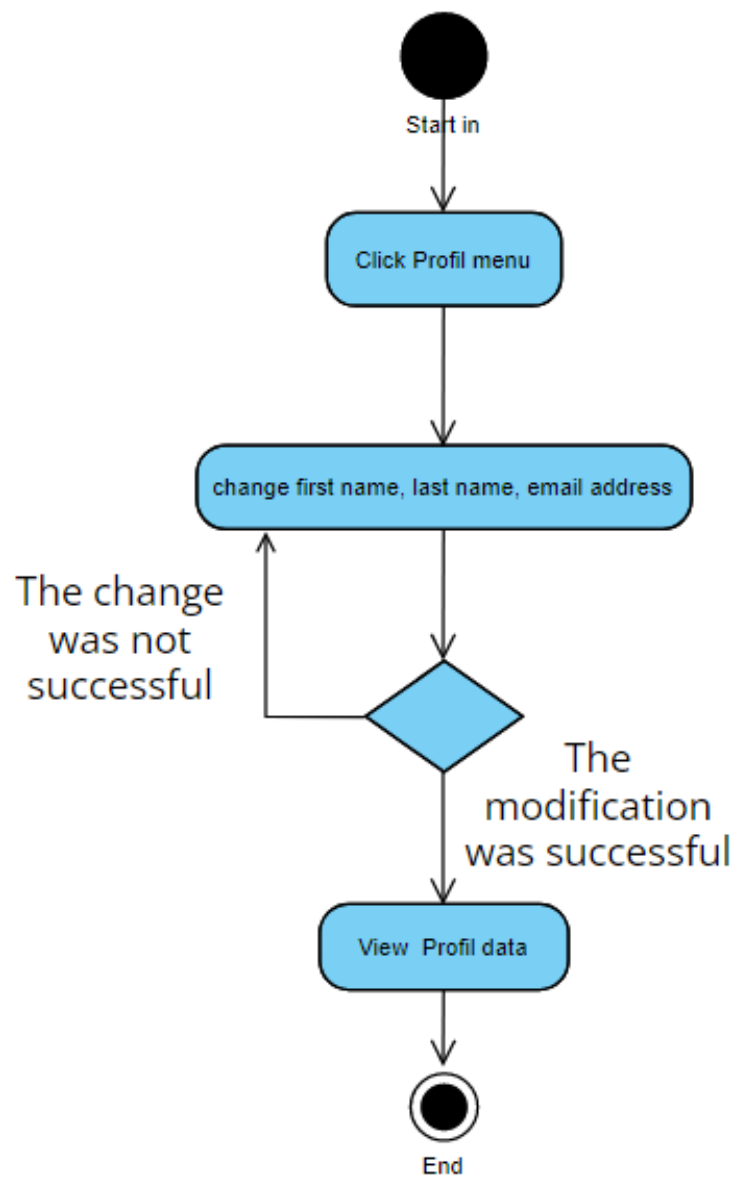
A felületen a letöltött fájlok megnyitásához, ami a kódolást teszi egyszerűbbé egy pdf olvasóra lesz szükség, ez akár lehet az Adobe Acrobat DC.

A feladvány három nyelven van lekódolva és egy összetett fájlként van becsomagolva, aminek megnyitására pedig célszerű egy szövegszerkesztő például a Notepad++.

Az állomány, ha egy .cpp, .c kiterjesztésű fájl, akkor a Visual Studio 2022-vel is megnyitható, ha pedig .java akkor IntelliJ IDEA Community Edition-val, ezáltal olvashatóbbá válik a kód a programozó számára.



4.2. ábra. A bejelentkezés tevékenységi diagramja



4.3. ábra. Felhasználói adatok módosításának tevékenységi diagramja

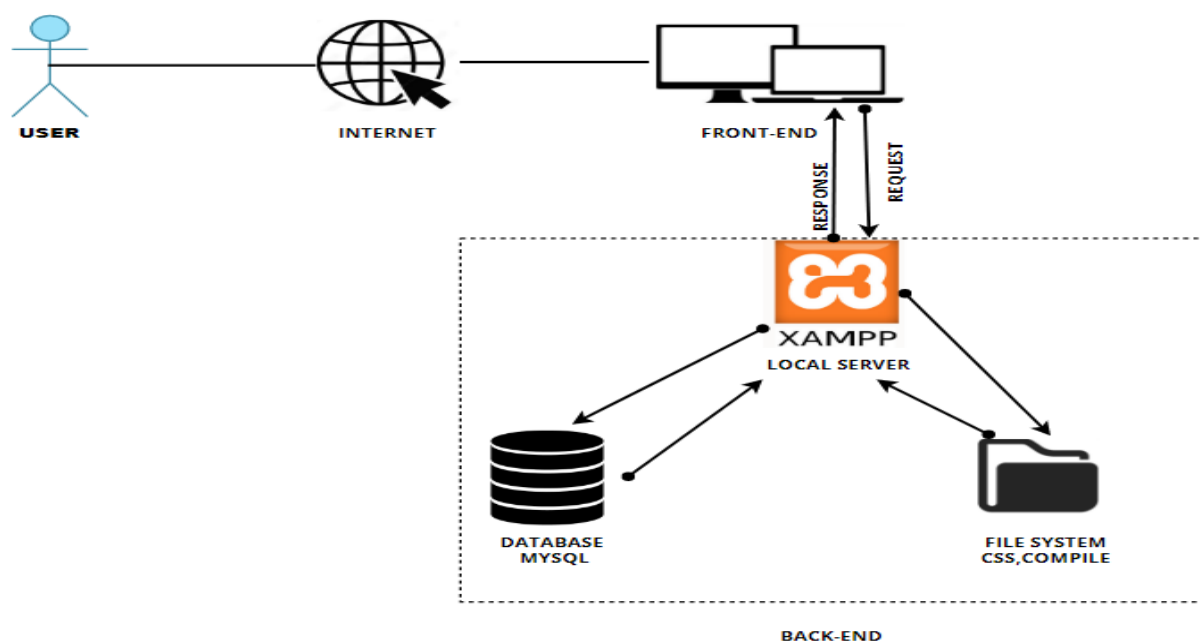
5. fejezet

Weblap felépítése

5.1. A rendszer architektúrája

A rendszer architektúrája két nagy komponensre bontható le (ami a 5.1 ábrán tekinthető meg):

- Front-end
- Back-end



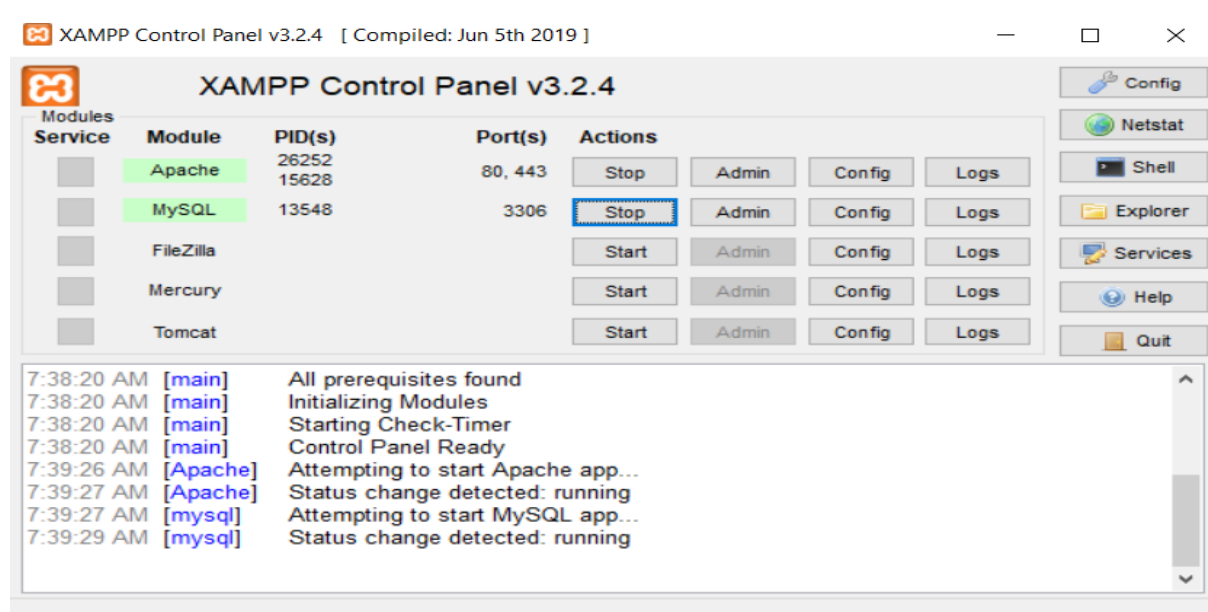
5.1. ábra. A rendszer architektúrája

A Back-end komponens pedig két részből áll az adatbázisból és a rendszerfájlból. Ahogy az ábra is mutatja a felhasználó igényel internetet a Front-end eléréséhez, ami nem más, mint maga a felület. A Front-end a kéréseket (requests) a szervertől kéri. A válaszokat (response) a szoftvertől kapja vissza. A lokális szerverről és az adatbázisról a következő alfejezetben bővebben lesz szó.

A rendszerfájl azon fájlokat tartalmazza, amelyek lokálisan vannak eltárolva mint például a weblap stílusai és a lokális futtatási környezetek.

5.1.1. Adatbázis leírása

A szakdolgozatom témájának fejlesztéséhez és működéséhez használom az *XAMPP* lokális szervert, ami biztosít egy *MySQL* adatbázis megvalósítást s ez a *phpMyAdmin* segítségével történik. Ahhoz, hogy használni tudjam az *XAMPP*-ot, ami egy ingyenes alkalmazás (látható a 5.2 ábrán) le kellett töltenem a személyes laptopomra, amin tervezem és fejlesztem a felületet.

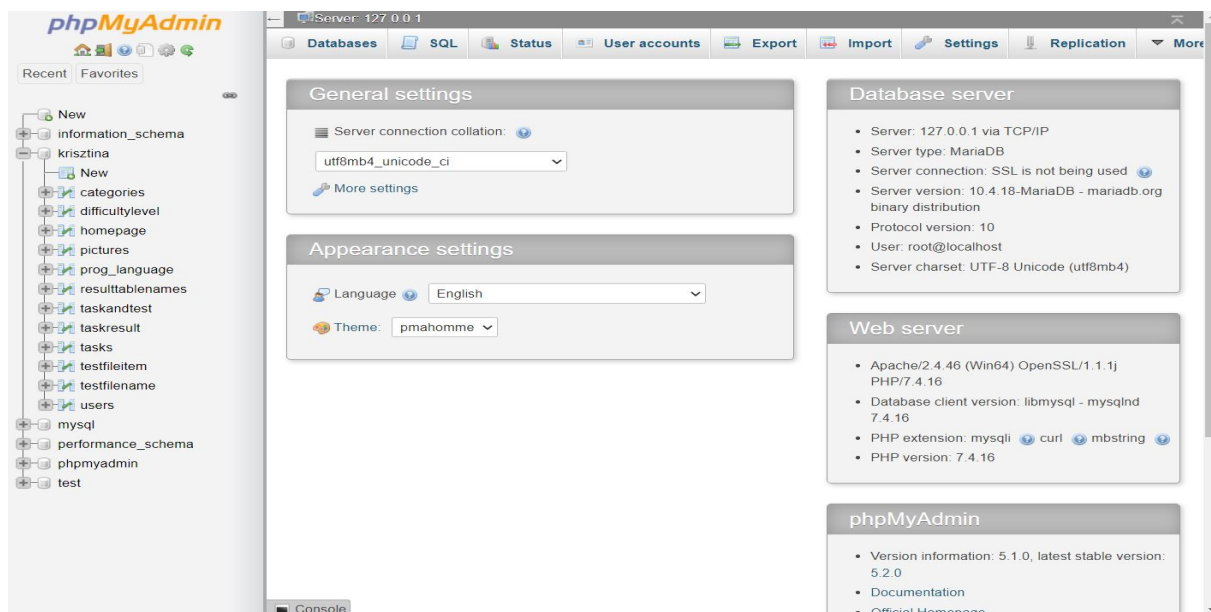


5.2. ábra. A lokális szerver

A *phpMyAdmin* egy költségmentes szoftver, amely megkönnyíti a felhasználónak az adatbázis használatát. A szoftver egy grafikus felület, ami PHP nyelven készült s íródott. (látható a 5.3 ábrán).

A felhasználó közvetlen SQL lekérdezéseket hajt végre a *phpMyAdmin* felületen keresztül, hozzájárul a műveletek végrehajtásában, megvalósításában, mint a mezők, táblák létrehozásában, módosításában és törésében is. Az adatbázisok kezelésében és kivitelezésében a *phpMyAdmin* létfontosságú szerepet játszik. A *phpMyAdmin* felületén való tevékenységek végrehajtásához először el kell indítani az *XAMPP*-ot s az *Apache* és *MySQL*(5.2 ábrán zöld színnel van kijelölve) modulokon a „Start” gombbal kell klikkelni. Amint ez a *MySQL* működni kezd a mellette lévő *Admin*-ra kell kattintani, ami behozza a *phpMyAdmin* felületet.[14]

Ahhoz, hogy a grafikus felületen tudjon táblákat létrehozni, kell a hozzáférés miatt egy felhasználói fiók, s egy az adatbázisnév is szükségszerű.



5.3. ábra. PhpMyAdmin felülete

Mindezek után egy *database.sql* fájlban létrehoztam a táblákat s feltöltöttem adatokkal majd beimportáltam erre az adatbázisnévbe, amit előzőleg megvalósítottam és létrehoztam.

A 5.1 kódrészleten megtekinthető a feladatok (tasks) létrehozásának egy táblája. Észlelhető egy azonosító is, ami az ábrán az első sor első értéke, amely egy elsődleges kulcs is (primary key). Az azonosító automatikusan növekszik.

A tábla még két idegen kulcsot tartalmaz, amit angolul *foreign key*-nek neveznek. A két idegen kulcs pedig a kategória s nehézségi szint id-kat (categorieId, lavelId) tartalmazza. Az idegen kulcsok nagyon hasznosak, mivel megakadályozzák az ismétlődő adatokat az adatbázisban.

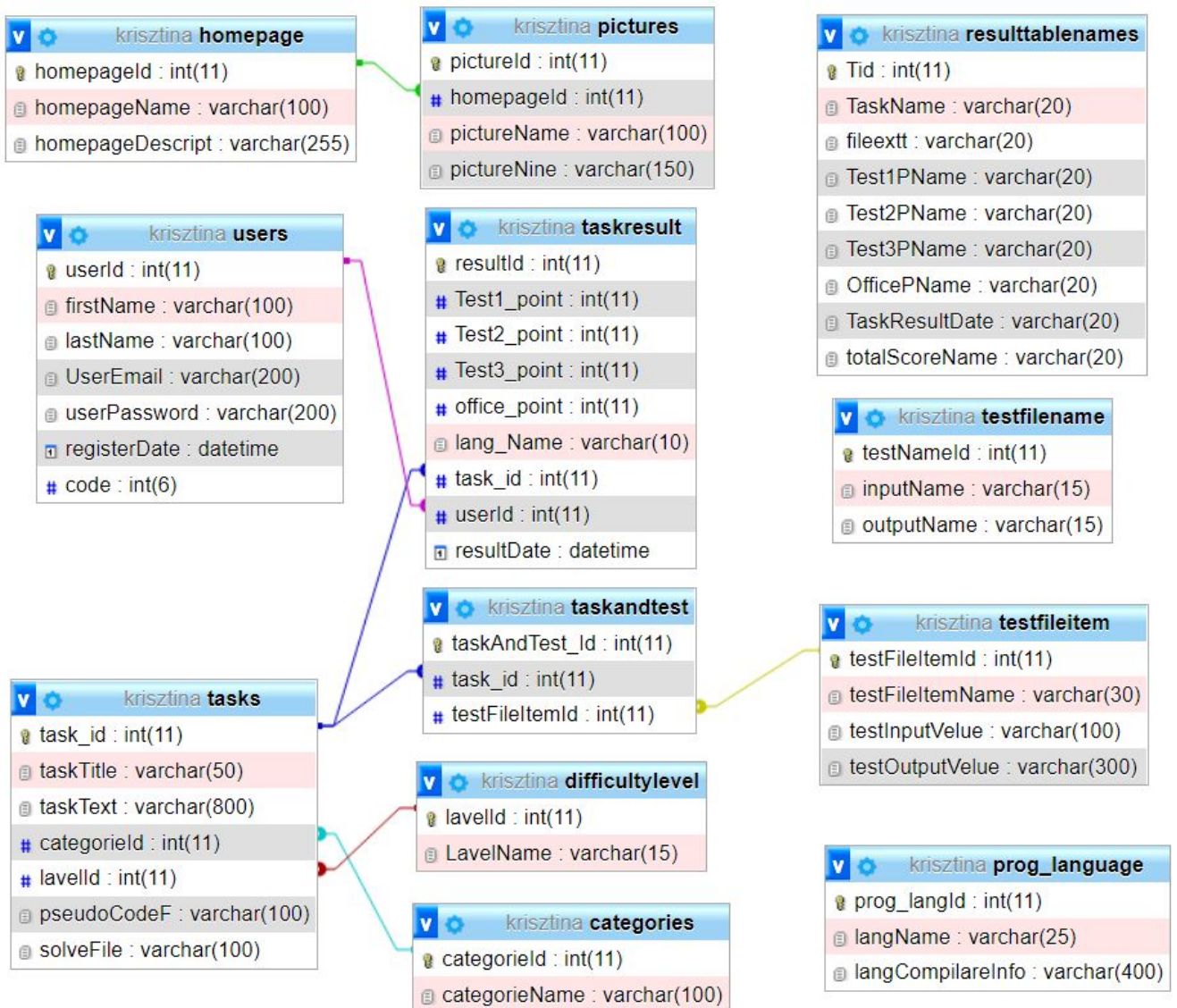
A kulcsokon kívül még más szükséges adatot is el kell eltárolni mint a feladat címét (taskTitle), a kérelmét (taskText), a pszeudokód fájl nevét (pseudoCodeF) és a megoldás fájljának szintúgy a nevét (solveFile).

```
CREATE TABLE tasks(
    task_id int not null AUTO_INCREMENT PRIMARY KEY,
    taskTitle varchar(50) not null,
    taskText varchar(800) not null,
    pseudoCodeF varchar(100),
    solveFile varchar(100),
    categorieId int,
    lavelId int,
    FOREIGN KEY (categorieId) REFERENCES categories(categorieId),
    FOREIGN KEY (lavelId) REFERENCES difficultyLevel(lavelId)
);
```

5.1. kódrészlet. Feladat tábla létrehozása

5.1.2. Az adatbázis struktúrája

Az 5.4 ábrát a *phpMyAdmin*-ban a „Designer” nézetre klikkelve valósítottam meg mivel ez kirajzolja az adatbázisom struktúráját.



5.4. ábra. Az adatbázis struktúrája

5.2. Felhasznált technológiák

A felület fejlesztése folyamán számos programozási nyelvet alkalmaztam a fejlesztés és létrehozás során, mivel áttekinthetőbbé, szebbé tették a végrehajtást.

5.2.1. PHP

A php, mint programozási nyelv a leggyakrabban igénybe vett szkriptnyelv mely dinamikus s statikus webalkalmazások létrehozására is felhasználják. Előnye pedig az, amit magam is felhasználok a fejlesztésben, hogy beágyazható a HTML kódba.

Ez a szkriptnyelv költségmentesen elérhető. A webhelyek javára gyors betöltéssel rendelkezik, amivel még inkább ajánlatos az igénybevétele. Világos, egyszerű, áttekinthető jellege miatt könnyebb a használatbevétele. [2]

5.2.2. HTML, CSS és JavaScript

A HTML felosztható hiperszövegre (HyperText), melyben a szöveghez a hivatkozások révén fér hozzá, és jelölés-re (Markup), ami a szöveg megjelenítését vázolja fel. Ennek következtében azt jelenti, hogy a HTML meghatározza a webhely szerkezetét, struktúráját.[3]

A CSS egy stíluslapnyelv, ami szinte minden HTML-ben használt elemnek a korrigálásra összpontosít. A HTML kódba kell csatolni ahhoz, hogy észrevegye a változtatást az elemeken. A CSS egy statikus programozási nyelv, amely segítségével a webhely egy letisztult, stílusos, modern, s tetszetősnek tűnik.

A JavaScript egy nagyon közismert s felkapott nyelv, amit gyakran használnak a fejlesztők. Akkor használják inkább, amikor interaktivitást, több viselkedést kell hozzáadniuk a weblaphoz. Ez a nyelv, mint Front-end és Back-end-re is használható.[4]

A fejlesztésben felhasználtam ezt a három nyelvet mivel nagyon elterjedt, népszerű és az eddigi tapasztalatim is azt igazolták, hogy könnyebbé, egyszerűbbé tették ezáltal a munkámat.

5.2.3. Bootstrap

A *Bootstrap* egy CSS keretrendszer, amit a front-endhez használnak mivel a webfejlesztéshez letisztult s gyors. Tartalmaz rengeteg sablont a gombokhoz, a navigációhoz, a kérdőívekhez s még sok egyébhez is. Reszponzív weboldalakot lehet vele létrehozni.

Ezeket építettem bele, mivel ennek használatával több időt spóroltam, aminek következtében több időm jutott a fontosabb részre a fejlesztés során.[5]

5.3. Felhasznált fejlesztői környezet

5.3.1. Visual Studio Code

A diplomamunkám fejlesztéséhez a Visual Studio Code-t használtam ebben dolgoztam s programoztam. A Visual Studio Code egy ingyenes környezet, mindenki számára hozzáférhető, használható. [6]

Azért választottam ezt a kódszerkesztőt, mivel ehhez könnyen lehet hozzáadni bővítményeket. Mivel több nyelvben is fejlesztettem, s programoztam, ezért szükségem volt egy olyan alkalmazásra, amiben hatékonyan tudok dolgozni.

A bővítmények arra szolgáltak, hogy kiegészítsék az azonnali beírt kódrészletet, ami-re adnék egy példát is, hogy szemléltessem ennek a fontosságát. Egy HTML elemet pötyögtettem be a szerkesztőbe pontosabban a `<div>`-et akkor abban a pillanatban amikor a „>” karaktert beírtam megjelenik az elemnek a bezárása is, a `</div>`.

Mivel sokszor használtam a tanulmányaim során ezt a környezetet, ezért még inkább szerettem ebben fejleszteni. Könnyebben s felszabadultabban tudtam leimplementálni az elképzeléseimet, gondolataimat.

5.4. Felhasználói felület bemutatása



5.5. ábra. Főoldal

A felhasználó megnyitja a böngészőben a platformot, s a főoldalon találja magát, ami az 5.5 ábrán tekinthető meg. A főoldalon lévő első kép szemlélteti, hogyan néz ki a regisztrációs weblap. A „Regisztrálj” egy link, egy webcím, amire rákattintva a regisztrációs oldalra kalauzolja el a felhasználót. Nemcsak innen tudja elérni, megnyitni, hanem még a főoldalnak a tetejében lévő vízszintesen elhelyezkedő menüből is a Regisztráció névvel.

Ha lejjebb görget, akkor megfigyelheti a három ikont, amelyen a programozási nyelvek vannak feltüntetve, megjelenítve. A menüben észrevehető a főoldal mellett egy „Bejelentkezés” menüpont és előtte egy kisebb ikon is. Az ikonok a név előtt arra szolgálnak, hogy azonnal kitűnjön, feltűnjön a felhasználónak a főbb menüpontok.

A lapon felfedezhető a bal felső sarokban egy logó is (5 Technikával gyakorolj), aminek az a célja, a szerepe, hogy azonosítsa a webhelynek a fő arculatát.

A lap alján észlelhető egy horizontálisan lévő lábléc, melynek nincsen nagyobb jelentősége, csak egy nevet tartalmaz, mégpedig azét, aki tervezte és fejlesztette ezt a platformot.

A regisztráció weblapján mindig elő fog bukkanni egy felugró ablak, mely egy előre meghatározott információval rendelkezik, egy „OK” gombbal, ami a szöveg alatt található. Értesítést ad arról, hogy ne használja a felhasználó ugyan azt az e-mail címet, amit már ezen a felületen igénybe vett. Ha mégis ez használta, akkor megint visszakerül erre a lapra s újból ki kell tölteni a megadott mezőket. Továbbá szükséges a „Regisztráció” gombra kikelnie ahhoz, hogy ismételten most már biztosan regisztrálva legyen a platformon.

5.6. ábra. Regisztrációs oldal

Az 5.6 ábrán megfigyelhető, hogy a kitöltendő mezők körvonalai piros színűek, ez azt jelenti, hogy üresek. Ha így hagyja a cellákat s mégis megnyomja a „Regisztráció” gombot akkor az első mezőtől kezdődően kis üzenetekkel figyelmezteti a felhasználót, hogy oda mit kellene írnia.

Amit kitölti, s ha újból megnyomja a gombot a rákövetkező cella, ami nincs kitöltve ahhoz is ad egy jelzést, s ez így megy tovább addig, amíg az egészt ki nem tölti. Két jelölő négyzet található ezen az oldalon, az első a „Jelszó megtekintése”, amivel a pontozott karakterek olvasható karakterekké válnak. A második pedig a feltételek elfogadása. A gomb alatt egy link is megjelenik a „Jelentkezz be most”, ami a jelentkezés oldalára irányítja.

A regisztráció weblapon észrevehető, ha minden mezőt megfelelően kitölt, akkor a cellák körvonala zöld színre változnak s a rubrika alatt pedig ugyancsak zöld színnel értesíti a felhasználót, hogy rendben van a mező, már ki van töltve.

A jelszó és a megerősített jelszónál megfigyelhető, ha megegyezik ez a kettő csakugyan zölddel kiírja a cella alá, hogy „Egyezik a két jelszó”, ha pedig nem akkor pirossal „Nem egyezik a két jelszó”.

Sikeres regisztráció után a felhasználó a bejelentkezés oldalra érkezik. De ha már volt regisztrációja, akkor nem kell a fent említetteket végig csinálnia, hanem egyenesen a „Bejelentkezés” gomra kell kattintania a menüből.

A bejelentkezés oldalán két mezőt kell kitölteni az e-mail címet és a jelszót. Ismételten egy felugró ablakot lát először a felhasználó ez arra figyelmezteti, hogy ha azzal az email címmel és jelszóval, amit begépett, ha még nem regisztrált ne tudjon bejelentkezni, s ezért újból kell próbálkoznia.

Ahogy a regisztrációnál is megjelent a piros és a zöld körvonal s a segítségül kapott mondatok itt is ugyanúgy megjelennek annak függvényében, hogy a cella üres vagy üresen a gombra klikkelt, vagy már nem üres. (lásd a 5.7 ábrát)

Megfigyelhető, ha az e-mail cím és a jelszó rubrikákba nem írja be a megfelelő alakjukat addig nem vált zöldre a cella, hanem piros marad.

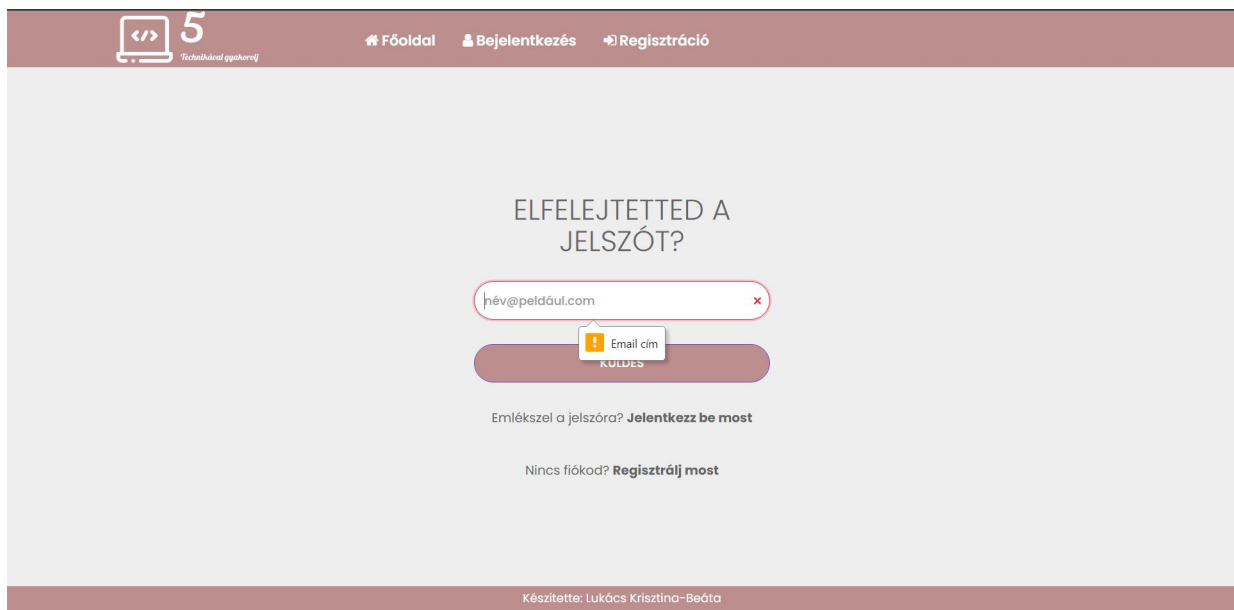
Két link is észlelhető az egyik a „Regisztrálj most” ez visszavisz a regisztráció weblapra s a másik az „Elfelejtett jelszó?” -val megváltoztathatod a jelszót. (5.8)

Amint a cellák körvonalai zöldre színeződnek s a „Bejelentkezés” -re kattintva átugrik a kezdőlapra, ami azt jelenti sikeresen be tudott jelentkezni.

The screenshot shows a web page with a dark red header. On the left is a logo with a computer icon and the number '5'. In the center of the header are three navigation links: 'Főoldal', 'Bejelentkezés', and 'Regisztráció'. The main content area is light gray and contains a registration form. The form has two input fields: the first is for an email address, containing 'hév@peldaul.com', and the second is for a password, containing 'Jelszó'. A tooltip with an exclamation mark and the text 'Email cím' is visible over the password field. Below the fields is a checkbox labeled 'Jelszó megtekintése' and a link 'Elfelejtett jelszó?'. A large red button labeled 'BEJELENTKEZÉS' is centered below the form. At the bottom of the form area is a link 'Nincs fiókod? Regisztrálj most'. The footer is dark red and contains the text 'Készítette: Lukács Krisztina-Beáta'.

5.7. ábra. A bejelentkezés oldal

A felhasználónak van lehetősége a jelszavát megváltoztatni az e-mail címére küldött kód segítségével. Ahogy a bejelentkezésnél itt is felűink a piros és zöld jelzés a mezőkön, és egy komment szerűség, ami segítségül lehet a felhasználónak a helyes kitöltésben. Az elfelejtett jelszó oldalon a webcímek által a felhasználó vissza tud lépni a bejelentkezés



5.8. ábra. Elfelejtett jelszó weblap

oldalra a “Jelentkezz be most” -al s a regisztrálásra a „Regisztrálj most” -al. (látható a 5.8 ábrán)

Az e-mail cím elküldése után, átkerül a kódellenőrző oldalra, ahol be kell pötyögtetnie a hat darab véletlen sorrendű számot, amit az e-mailben kapott a felhasználó. Ezután egy újabb oldalra kerül, ahol két mező várja az új s a megerősített jelszóval. Ha ez sikeresen megváltozott visszakerül a főoldalra, ahol bejelentkezhetsz a már megújított jelszóval.



5.9. ábra. Kezdőlap

A kezdőoldalra éve köszönti a felhasználót a teljes nevén s sok sikert kíván a kitűzött feladatok megoldáshoz. A menüben a „Főoldal” az utolsó helyen áll, mivel nem ez a legfontosabb, legérdekesebb oldala a szoftvernek.

Főoldal mellett az oldal széléhez közeledve látható a keresés gomb, amivel a feladatokat lehet a címük szerint megtalálni. (5.9 ábrán)

A már megszokott s használt menüben megjelent a „Profilom” nevű oldal is, ami tartalmazza a bejelentkezett információját, ami megtekintő, ha rákattintott s betöltődött az oldal, 5.10 ábrán figyelhető meg.



5.10. ábra. A felhasználó adatainak a megtekintése

Lefelé görgetve láthatóvá válik három kitöltött mező: a vezetéknév, keresztnév, s az e-mail cím, amiben a saját adatai vannak megjelenítve, majd a „Módosítás” -ra klikkelve átalakítja őket. (5.11 ábrán) Egyszerre lehet átszerkeszteni, átírni a már meglévőket valami másra, újra. Sikeres módosítás esetén egy felugró ablak jelez a felhasználónak. Majd ismételten betöltőik az oldal, ahol észrevehető, megfigyelhető a módosított érték vagy értékek. A menüben észlelhető, hogy ennek az oldálnak két aloldala van, mégpedig a „Feladatok pontszámai” és a „Kijelentkezés”. A felhasználó a „Kijelentkezés”-t választva kijelentkezik az oldalról.

A menüt nézve jobbról az első főmenüpont az a „Technikák”. Lenyílik hozzá egy öt tagú almenü is, s ez az öt stratégia, amik a következők:

- Oszd meg és uralkodj
- Visszalépéses keresés (5.12 ábra)
- Mohó módszer
- Dinamikus programozás
- Elágazás és korlátozás

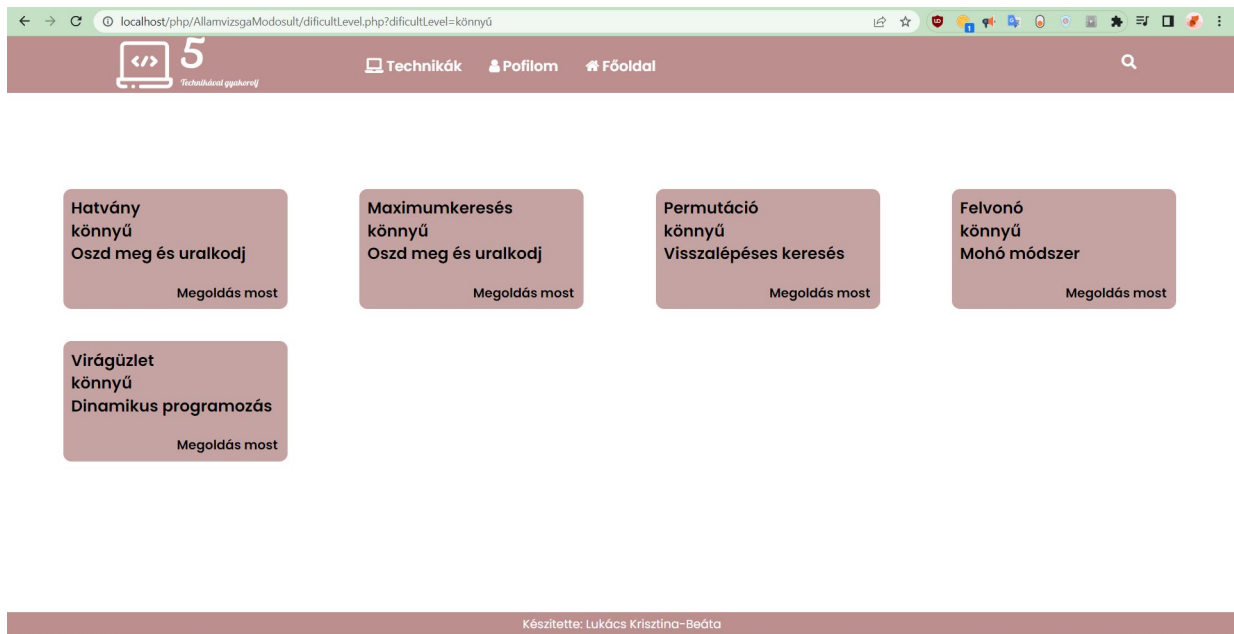
5.11. ábra. A felhasználó adatainak a módosítása

5.12. ábra. Visszalépés keresés oldala

A stratégiák egy-egy weblapok, amin különböző nehézségű feladatok tekinthető meg a címeikkel s egy linkkel „Megoldás most” -al együtt. Ezekénél a weblapoknál a felhasználónak lehetősége van szűrést végezni, mégpedig a könnyű, közép, nehéz kulcsszavakra, fogalmakra kattintva megjelenítődnek azok a feladatok is, amelyek nem csak egy stratégiába sorolhatók. (lásd 5.13 ábrán)

Összesen tíz darab programozási stratégiával megoldható feladat van jelen a weblapon, három feladat van az első almenüpontban, két feladat a második, harmadik és negyedikben, s egy feladat az elágazás és korlátozásnál. A tíz feladatból néhány feladatnak

a megjelenítése a Dr. Katái Zoltán „Algoritmusok felülnézetből” és „Algoritmustervezési stratégiák” című könyveiből inspirálódtam.[15] [16]



5.13. ábra. Könnyű nehézségi szintű feladatok

Minden feltöltött feladatnak van külön saját weblapja. (5.15 ábra) A megszokott módon itt is feltűnik egy felugró ablak, ami jelzi a felhasználónak a visszaszámolt időnek az újratekintését amint a weblap újból betöltődik. Ez az idő egy harminc perces (1800 másodperces) visszaszámlálás, ami következtében egy összecsomagolt fájl kerül letöltésre. Az a személy, aki a feladatot oldja sok hasznát veszi, mivel ez a feladatnak egy lehetséges megoldását, leprogramozását tartalmazza három programozási nyelven.

A felette lévő gomb, aminek letöltése nincs időre megszabva, rögtön le lehet tölteni s felhasználni. Ennek a gombnak a neve a „Pseudokód letöltése”, egy pdf fájl, amiben a feladatnak a vázlatos kódolása található, magyarázatokkal, kommentekkel együtt.

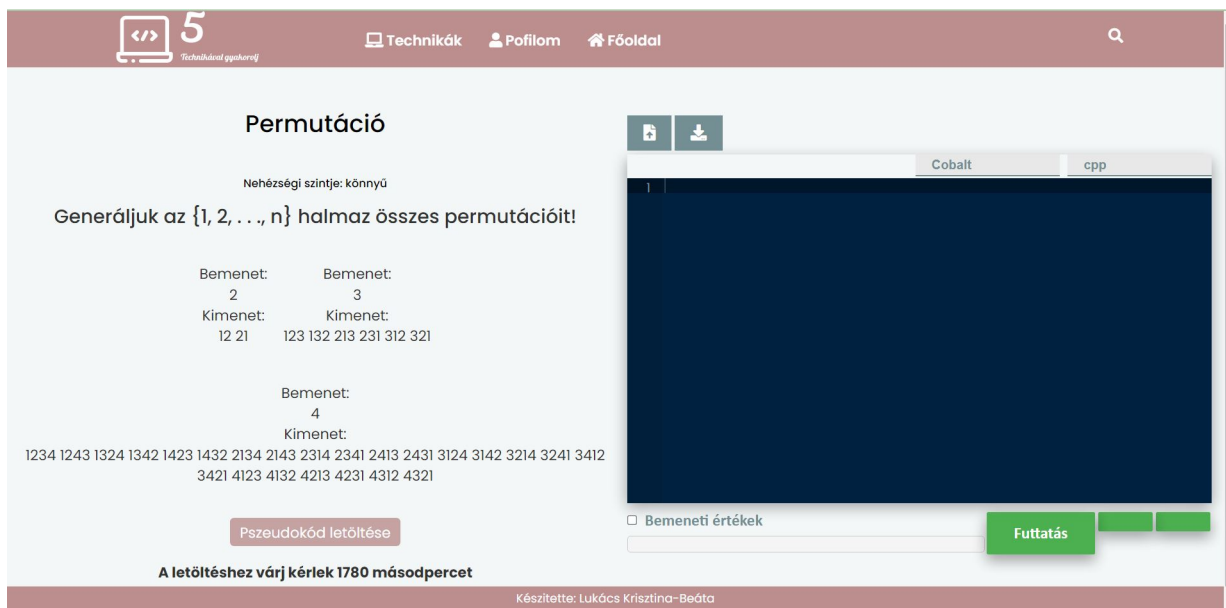
A feladat a holnapon megjelenített fordítóprogramra is figyelmezteti a rendszert a felhasználót. Ezen oldalak újbóli újratöltődése után a már a fordítóprogramba bekerült algoritmus, kód letölthető onnan s újból kell küldenie.

Az oldal tartalmazza a gyakorlat címét, a hozzá tartozó nehézségi szintet, és a feladat szövegét is. Három bemeneti és kimeneti értéket jelenít meg aminek fontos szerepe van a javításban. (5.15 ábra)

A fordítóprogramon három programozási nyelvnek a kiterjesztéséből lehet választani, s magának a szerkesztőnek a dizájnját is meg lehet változtatni. A téma megváltoztatása bármikor történhet lehet a kódolás előtt vagy közben is.

Két ikon látható, amiknek különböző szerepük van. Az elsőre kattintva feltölthető a fordítóprogramba egy leprogramozott algoritmus, ami a „Futtatá”-ra klikkelve kiírja az eredményt. A másodikra klikkelve pedig letölthető a begépelt kód a szerkesztőből.

A program alatt egy jelölő négyzet mutatkozik meg a „Bemeneti értékek”-el amit be kell jelölni, ha használni akarja a felhasználó, amihez feltűnik az értékeknek egy szöveg-doboz. Az „Elküldés” csak akkor jelenik meg a felhasználónak a „Futtatás” gomb mellett,



5.14. ábra. Permutációs feladat

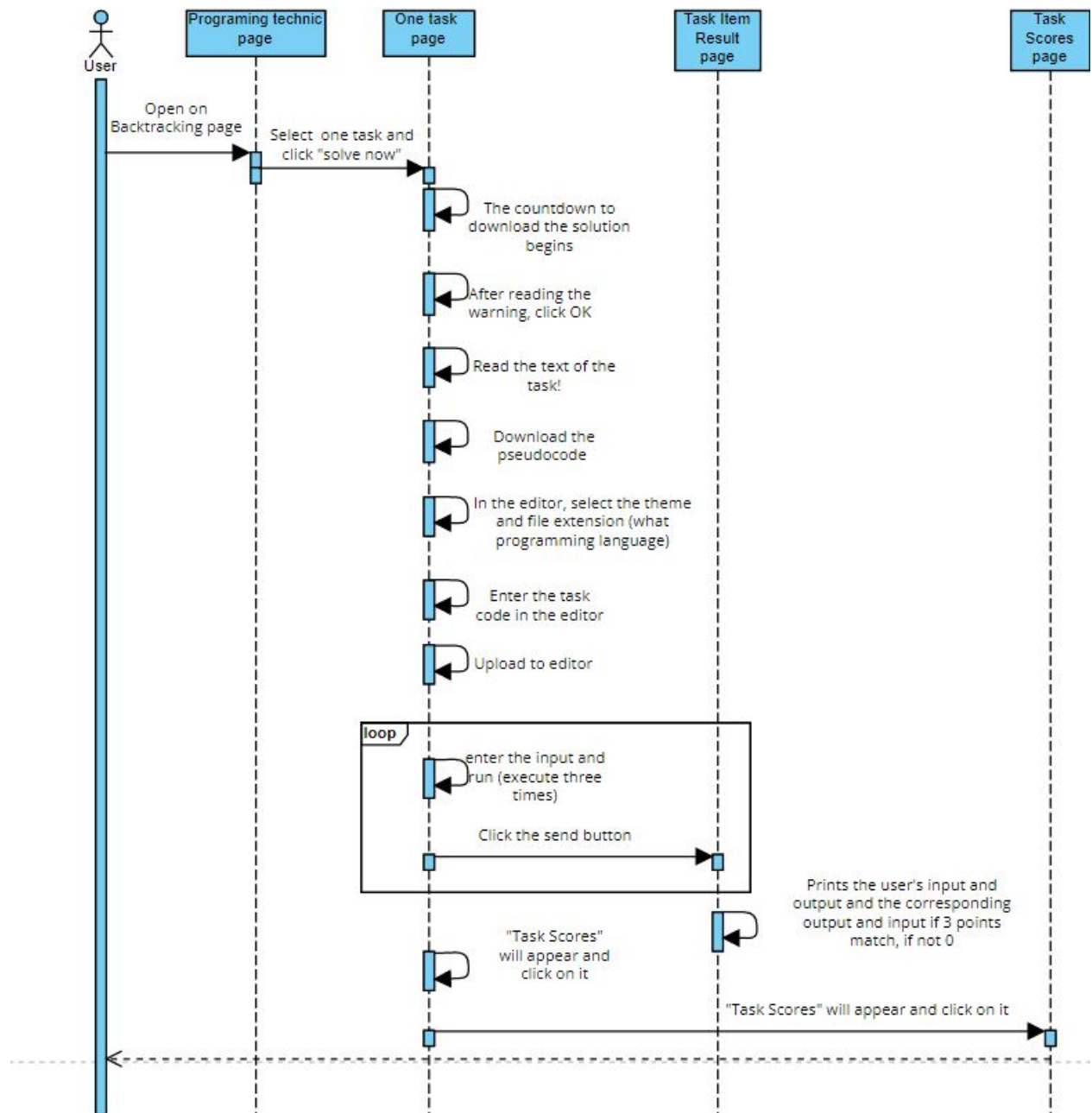
amikor lekódolta, ezután az egyik bemenettel együtt kipróbálta s lefuttatta akkor jelenik csak meg az a gomb. Csakis háromszor kattinthat erre az „Elküldés”-re. Minden küldés után megnyílik egy új oldal, ahol a felhasználó és a rendszer bemenetének és kimenetének értékei megjelennek, de csak öt másodpercre aztán eltűnik. Tehát minden bemenetre csak egy lehetősége van beküldeni javításra. Miután teljes egészében az „Elküldés” gomb eltűnik helyette megjelenik a „Feladatok pontszámai” ami egy táblázatot jelenít meg, amiben fel van tüntetve a három teszt fájlnak a pontszámai, a hivatalból kapott pont, a fájlki-terjesztése, egy időpontot, a feladat nevét, s az összesített pontszámot. A pontszámok táblázatának a megtekintése a főmenüből is elérhető és megtekinthető.

FELADAT CÍME	FÁJLKITERJESZTÉS	TESZTFÁJL1 PONTSZÁM	TESZTFÁJL2 PONTSZÁM	TESZTFÁJL3 PONTSZÁM	HIVATALBÓL	DÁTUM	ÖSSZESÍTETT PONTSZÁM
Hatvány	java	3	3	3	1	2022-06-16 16:57:15	10
Hatvány	cpp	3	3	3	1	2022-06-16 16:58:21	10
Hatvány	c	3	3	0	1	2022-06-16 17:00:57	7
Maximumkeresés	cpp	3	3	3	1	2022-06-16 17:12:01	10
Maximumkeresés	c	3	3	0	1	2022-06-16 17:33:35	7
Felvonó	c	3	3	3	1	2022-06-17 10:21:05	10
Felvonó	cpp	3	3	0	1	2022-06-17 10:30:07	7

Készítette: Lukács Krisztina-Beáta

5.15. ábra. Összesített pontszámok táblázata

Megfigyelhető az 5.16 ábrán egy visszalépéses keresés feladatának a szekvencia diagramja.



5.16. ábra. Szekvencia diagram

5.5. Implementáció

5.5.1. Automatikus javítás

A felhasználó bármikor le tudja futtatni a már begépet, vagy feltöltött kódját. Futtatás során fel kell használnia az összes bementet, ami meg van jelenítve a feladat oldalon, de külön-külön futtatva. A futtatás után, ha nem ad ki szintaktikai hibát, hanem egy bizonyos eredményt, akkor elküldheti az algoritmusát a javításra.

A felhasználót a rendszer egy új oldalra irányítja, az ellenőrzésre, aminek a linkje tartalmazza az addig oldott feladatának a bemeneti és kimeneti értékeit.

Innen fogom leszedni s hasonlítom össze a feladatnak megfelelő be-és kimeneteivel, ami az adatbázisban van eltárolva. Mivel három tesztelési fájlt tartalmaz a platform, ezért három alkalommal kell beküldeni egy feladat megoldását, hogy megkapja a végső pontszámot.

Egy függvényben implementáltam le. Elsőként egy változóba lekértem az adatbázisból az összes bemeneti, kimeneti értékeket, ami ahhoz a feladathoz tartozik, amit éppen programozott, s beküldött javításra a felhasználó.

Külön tömbbe teszem ezeket az értéket. Lementem őket külön-külön változóba mivel, nem együtt használom fel őket a későbbiekben.

Létrehoztam egy táblát a pontszámoknak. Ez az a táblázat lesz, ami elérhető a „Feladatok pontszámai”-ra kattintva. Ha létezik már a feladat ebben a táblában, amit a felhasználó programozott, abban a programozási nyelvben, s az a személy oldotta-e meg aki bejelentkezve van a honlapon, akkor lementem ennek a gyakorlatnak az azonosítóját és programozási nyelvét is.

A 5.2 kódrészleten megtekinthető az automatikus javítás egy fontosabb része. Ezen a kódrészleten megfigyelhető az első sorban lévő „if” aminek segítségével összehasonlítom a két bemenetet és kimenetet, amit felhasználó küldött be és az adatbázisban lévőét. Ha megegyezik azt jelenti, hogy helyes s megfelelő a leprogramozott feladat, ezért kiírom arra az ellenőrzés weblapra, ami megjelent a felhasználónak új lapként.

A második „if”-ben azt vizsgálom, hogy az az azonosító, amit már említettem, létezik-e, ha igen lementem id-ját és programozási nyelvét is.

A harmadik „if”-ben pedig azt elemzem, hogy ugyanaz a kódolási nyelvben van feltöltve a már beküldött feladat. Ha a belső „if” igaz akkor módosítja, hogy abba az előzőleg megkapott azonosítójú mezőben az első tesztelési fájlhoz három pont kerüljön.

Ha az ág hamis, vagyis nem abban a nyelvben volt lekódolva hanem egy másikban, akkor egy új mezőbe töltse fel s csakugyan az első bemenethez a három pont és a hivatalból kapott egy pont kerüljön fel a táblának a mezőjébe. Ez a hamis ág azt jelenti, hogy ugyan az a feladat, de más nyelvben oldotta meg.

Az első „if”-nek, ha a másik ága teljesül akkor az jelenti, hogy a kimenet nem egyezik meg a megadottéval. Csak nulla pontot kap erre a tesztfájltra. Ezzel az értékkel tölti a táblának az első bemeneti celláját.

Az „elseif”-ben az az eset van megírva, mely során a felhasználónak bemenete megegyezik a rendszerével, de a kimenetük teljesen eltérő.

Ez a két eset, ami ebben a kód részletben van bemutatva ez az első bemeneti tesztelési fájlra van leimplementálva s így néznek ki a másik kettőre is csak más adatokkal.

```
if($input === $testInput[0] && $output === $testOutput[0]){
```

```

echo '<div id="hideDiv">'. $taskName. '<p>A te
    bemeneted:'. $input. '<br> A te kimeneted:'. $output. '<br>A helyes
    bemenet:
'. $testInput[0]. '<br>A helyes
    kimenet:'. $testOutput[0]. '</p><br>Megegyzeik! Kapsz 3 pontot
</div>';
if(mysqli_num_rows($result) > 0){
while($row = mysqli_fetch_assoc($result)){
    $resultId=$row['resultId'];
    $resultLang = $row['lang_Name'];
}
if( $resultLang === $langU ){
$query = "UPDATE taskresult SET Test1_point=3 WHERE
    resultId='$resultId'";
}
else{
    $query = "INSERT INTO taskresult
        (Test1_point,Test2_point,Test3_point,office_point,
        lang_Name,task_id,userId,resultDate)
        VALUES(3,0,0,1,'$langU',$pageId, $user_id, NOW())";
}
}
elseif($input === $testInput[0] && $output != $testOutput[0]){
echo '<div id="hideDiv">'. $taskName. '<p>A te
    bemeneted:'. $input. '<br> A te kimeneted:'. $output. '<br>A helyes
    bemenet:
'. $testInput[0]. '<br>A helyes kimenet:'. $testOutput[0]. '</p><br>A
    kimenet nem egyezik.<br>Kapsz 0 pontot!</div>';

if(mysqli_num_rows($result) > 0){
    while($row = mysqli_fetch_assoc($result)){
        $resultId=$row['resultId'] ;
        $resultLang = $row['lang_Name'];
    }
    if( $resultLang === $langU ){
        $query = "UPDATE taskresult SET Test1_point=0 WHERE
            resultId='$resultId'";
    }
}

```

```

else{
    $query = "INSERT INTO taskresult
        (Test1_point,Test2_point,Test3_point,office_point,
        lang_Name,task_id,userId,resultDate)
        VALUES(0,0,0,1,'$langU',$pageId, $user_id, NOW())";

    }
}
else{

    $query = "INSERT INTO taskresult
        (Test1_point,Test2_point,Test3_point,office_point,
        lang_Name,task_id,userId,resultDate)
        VALUES(0,0,0,1,'$langU',$pageId, $user_id, NOW())";

    }
    $result2=mysqli_query($link,$query);

}

}

```

5.2. kódrészlet. Automatikus javítás kódszinten

Összefoglaló

Dolgozatomban a programozási stratégiákkal megoldható feladatok automatikus javítását biztosító szoftver megvalósításával foglalkoztam, melyet különböző s többféle programozási nyelvekkel, technológiák segítségével valósítottam meg. Először egy felmérést végeztem, ami iránymutatásként szolgált a szoftver megtervezésében, s megalkotásában. Továbbá utánanéztem a hasonló platformoknak is, ami automatikus javítást biztosított. A webhelyen a felhasználó és rendszer által használt funkciók ismertetése következett, ahol bővebben s részletezve betekintést enged ezeknek a használatába. Diagramokkal ábrázolom eme pár megvalósulást. A továbbiakban ismertettem s ábrázolom a rendszer architektúráját és a szoftver vázát. A szoftver felépítése során megemlítem azokat a felhasznált technológiákat és fejlesztői környezetet, amit használtam és alaposan, aprólékosan bemutatom a felhasználói felületet képek segítségével. Majd ezután az implementáció fontosabb részét, az automatikus javítás megvalósítását ismertetem kódrészlettel s magyarázattal.

<https://github.com/kriszta99/Allamvizsga.git>

5.5.2. További fejlesztések

Jövőbeli terveimben szerepel elsősorban a platformom használatának a letesztelése a líceum és egyetem diákjai körében. Mennyire sikerült a fő funkció megvalósítása s, az a tény, hogy tényleges s valós adatok által a segítségükre lehet-e ez a szoftver a feladatok gyakorlásában, és ezen öt stratégia elmélyülésében.

Szeretném, ha ez a weboldal nemcsak lokálisan működne, hanem elérhető legyen bárki számára, akinek szüksége lenne erre a tanulása során.

Több feladattal, nehezebb szintű feladványokkal bővíteni mindegyik módszer oldalt. S nemcsak feladatokkal, hanem magát a programozási környezet részét is akár több programozási nyelvel mint például:

- Python
- Kotlin
- JavaScript
- PHP

A felhasználónak jelenítse meg összesen hány feladatot oldott meg, bejelölhesse a kedvenc feladatait, jelezze a feladatokon, ha már megoldott egyet, legyen a feladatok lekódolása során egy statisztika diagramja. Visszatérve a fordítóprogramra, ahol kellene egy *debug* része is amin a felhasználó lépésenként értheti meg az algoritmust is amiből sokat tanulhatna.

Ábrák jegyzéke

2.1. Az adatok névtelen feldolgozása	12
2.2. Célcsoport	13
2.3. Öt programozási technika felismerése	13
2.4. Programozási nyelv használata	13
2.5. Automatikus javítást biztosított szoftverek	14
2.6. Magyar nyelvű automatikus javítást biztosított szoftver	14
2.7. Nemek szerinti felbontás	14
4.1. A felület használati esetdiagramja	19
4.2. A bejelentkezés tevékenységi diagramja	21
4.3. Felhasználói adatok módosításának tevékenységi diagramja	22
5.1. A rendszer architektúrája	23
5.2. A lokális szerver	24
5.3. PhpMyAdmin felülete	25
5.4. Az adatbázis struktúrája	26
5.5. Főoldal	28
5.6. Regisztrációs oldal	29
5.7. A bejelentkezés oldal	30
5.8. Elfelejtett jelszó weblap	31
5.9. Kezdőlap	31
5.10. A felhasználó adatainak a megtekintése	32
5.11. A felhasználó adatainak a módosítása	33
5.12. Visszalépéses keresés oldala	33
5.13. Könnyű nehézségi szintű feladatok	34
5.14. Permutációs feladat	35
5.15. Összesített pontszámok táblázata	35
5.16. Szekvencia diagram	36

Irodalomjegyzék

- [1] Matek egyszerűen. <https://www.mateking.hu/>.
- [2] Why php is the best choice for web development? <https://semidotinfotech.ae/blog/why-php-is-the-best-choice-for-web-development/>. (2022.sep.22).
- [3] How do html, css and javascript work together? <https://www.itonlinelearning.com/blog/how-do-html-css-and-javascript-work-together/>.
- [4] Html vs. css vs. javascript: What's the difference? <https://brytdesigns.com/html-css-javascript-whats-the-difference/>. (2019.dec.9).
- [5] What is bootstrap? <https://www.javatpoint.com/what-is-bootstrap>.
- [6] Visual studio code hivatalos oldala. <https://code.visualstudio.com/>.
- [7] Tantargyakra lebontva tartalmaz videokat, gyakorlatokat. <https://zanza.tv/>.
- [8] Lightoj hivatalos weboldala. <https://lightoj.com/>.
- [9] The journey of building the new lightoj website. <https://medium.com/swlh/the-journey-of-building-the-new-lightoj-33f66b6a265>. (2021.jan.14).
- [10] Hackerearth hivatalos weboldala. <https://www.hackerearth.com/>.
- [11] Hackerearth review with pricing, hiring instructions, key information, and faqs. <https://www.betterteam.com/hackerearth>.
- [12] Coderbyte hivatalos oldala. <https://coderbyte.com/>.
- [13] About coderbyte. <https://coderbyte.com/about>.
- [14] Creating mysql database with xampp. <https://www.javatpoint.com/creating-mysql-database-with-xampp>.
- [15] D. K. Zoltán. *Algoritmusaok felülnézetből*. Scientia, Kolozsvár, 2007.
- [16] D. K. Zoltán. *Algoritmustervezési stratégiák*. Scientia, Kolozsvár, 2020.