SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK

SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Cryptorithm

DIPLOMADOLGOZAT

Témavezető:
Dr. Márton Gyöngyvér,
Egyetemi tanár

Végzős hallgató:
Füzi Zalán

2023

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

UNIVERSITATEA
SAPIENTIA

Cryptorithm

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Márton Gyöngyvér,
Profesor universitar

Absolvent:
Füzi Zalán

2023

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

SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Cryptorithm

BACHELOR THESIS

Scientific advisor:
Dr. Márton Gyöngyvér,
Full Professor

Student:
Füzi Zalán

2023

Declaraţie

Subsemnatul/a FÜZI ZALÁN, absolvent(ă) al/a specializării INFORMATICĂ, promoţia 2023 cunoscând prevederile Legii Educaţiei Naţionale 1/2011 şi a Codului de etică şi deontologie profesională a Universităţii Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licenţă/proiect de diplomă/disertaţie se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informaţiile şi datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, TÂRGU MUREȘ
Data: 2023.06.08.

Absolvent

Semnătura.....

Kivonat

A Cryptorithm egy online platform, ahol a felhasználók kriptográfiai rendszereket próbálhatnak ki, és egy kiterjedt tudásbázist kínál, hogy szabadon tanulhassanak azok, eredetéről, működéséről és felhasználásáról.

A szoftver választékot kínál a népszerű és ismert eszközökből az adatok titkosításához, visszafejtéséhez és transzformációjához, lehetővé téve a felhasználóknak, hogy szabadon válogathassanak és kipróbáljanak különböző lehetőségeket. Emellett rendelkezik egy tanulói felülettel, amely hozzájárul a felhasználók számára a kriptográfiai rendszerek mélyebb megértésében. Itt információkat olvashatnak, és gyakorlati példákon keresztül sajátíthatják el ezeket a rendszereket.

Kiemelt figyelmet fordítottam a fejlesztés során a felhasználói élmény biztosítására, így a szoftver könnyedén használható felhasználói felülettel rendelkezik. Emellett fontos szempont volt az, hogy az oldal támogassa a többnyelvűséget is, így az oldal elérhetővé válik idegen nyelű felhasználók számára is.

A diplomadolgozatom további részében részletesen bemutatom a szoftver architektúráját, a megvalósított funkciókat, valamint a hozzá tartozó tudnivalókat. Emellett a felhasználói dokumentáció is részét képezi a dolgozatnak.

Rezumat

Cryptorithm este o platformă online unde utilizatorii pot încerca sisteme criptografice și oferă o bază de cunoștințe extinsă pentru a învăța în mod liber despre originea, funcționarea și utilizarea acestora.

Software-ul oferă o selecție de instrumente populare și bine cunoscute pentru criptarea, decriptarea și transformarea datelor, permițând utilizatorilor să aleagă și să încerce în mod liber diferite opțiuni. De asemenea, dispune de o interfață de învățare pentru a-i ajuta pe utilizatori să dobândească o înțelegere mai profundă a sistemelor criptografice. Aici aceștia pot citi informații și pot învăța despre aceste sisteme prin exemple practice.

Am acordat o atenție deosebită asigurării unei experiențe de utilizare în timpul dezvoltării, astfel încât software-ul are o interfață ușor de utilizat. În plus, a fost important ca site-ul să suporte multilingvismul, astfel încât să fie accesibil utilizatorilor de limbi străine.

În restul tezei mele voi descrie în detaliu arhitectura software-ului, caracteristicile implementate și cunoștințele aferente. În plus, documentația utilizatorului va face parte din teză.

Abstract

Cryptorithm is an online platform where users can try out cryptographic systems and offer an extensive knowledge base to freely learn about their origin, operation and use.

The software offers a selection of popular and well-known tools for encrypting, decrypting and transforming data, allowing users to freely choose and try out different options. It also has a learning interface to help users gain a deeper understanding of cryptographic systems. Here they can read information and learn about these systems through practical examples.

I have paid particular attention to ensuring a user experience during development, so the software has an easy-to-use interface. In addition, it was important that the site supports multilingualism so that it is accessible to foreign language users.

In the rest of my thesis I will describe in detail the architecture of the software, the implemented features and the related knowledge. In addition, the user documentation will be part of the thesis.

Tartalomjegyzék

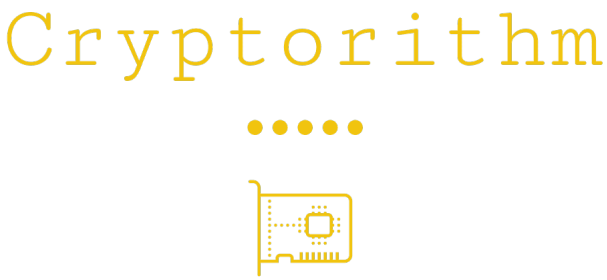
1. fejezet

Bevezető

A digitális korban a kriptográfia egyre növekvő jelentőséggel bír, hiszen az adatbiztonság és az adatvédelem kulcsfontosságú tényezők az információtechnológia területén. A kriptográfiai rendszerek megértése és hatékony használata elengedhetetlen az adatok titkosításához, visszafejtéséhez és biztonságos átviteléhez.

Célja a dolgozatomnak, hogy bemutassa a Cryptorithm nevű webes alkalmazást, amely egy kiterjedt platformot kínál a kriptográfiai rendszerek gyakorlati kipróbálására és tanulására. Az alkalmazás lehetővé teszi a felhasználók számára, hogy interaktív módon felfedezzék a különböző kriptográfiai algoritmusokat és azok alkalmazási területeit. Emellett a Cryptorithm egy részletes tudásbázist is nyújt, amely segíti a felhasználókat az algoritmusok eredetének, működésének és felhasználásának megértésében.

A Cryptorithm alkalmazás a Flask, egy Python alapú könnyűsúlyú webes keretrendszerre épül, amely lehetővé teszi a gyors és hatékony fejlesztést. Az alkalmazás intuitív és felhasználóbarát felülettel rendelkezik, hogy könnyű legyen a navigáció és a kriptográfiai rendszerek gyakorlati alkalmazása. Az alkalmazás többnyelvű támogatást is biztosít, így bárki, függetlenül az anyanyelvétől, elérheti és használhatja a platformot.



1.1. ábra. Cryptorithm logó

1.1. Motiváció és Célkitűzések

Kriptográfiai tudatosság növelése: Az alkalmazás célja, hogy növelje a felhasználók kriptográfiai tudatosságát és ismereteit. A projekt fő motivációja a felhasználók tájékozottságának növelése a digitális biztonság terén, segítve őket a kriptográfiai rendszerek megértésében és alkalmazásában.

Gyakorlati tapasztalat nyújtása: Lehetőséget kapnak a felhasználók, hogy gyakorlatban is kipróbálják és megtapasztalják a kriptográfiai rendszerek működését.

Tanulás és oktatás támogatása: Egy tanulói felületet van biztosítva, ahol a felhasználók elmélyülhetnek a kriptográfia terén. Motivációs tényező a tudásátadás és az oktatás támogatása, amely segít a felhasználóknak a kriptográfiai rendszerek megértésében és elsajátításában.

Nyitottság és közösség támogatása: Az alkalmazás lehetőséget nyújt az idegen anyanyelvű felhasználóknak is a használatra, hogy bővítsék tudásukat a kriptográfiai rendszerekről.

2. fejezet

Programok, technológiák bemutatása

2.1. Az alkalmazás felépítése

2.1.1. Frontend

A frontend fejlesztéséhez HTML-t, CSS-t és JavaScriptet használtam, amelyek lehetővé teszik az interaktív felhasználói felület létrehozását és a dinamikus funkcionalitás implementálását. Ezek a technológiák biztosítják a felhasználók számára az intuitív böngészési élményt és a kriptográfiai rendszerek kipróbálását.

HTML: Egy szabványosított jelölőnyelv, amelyet a weboldal strukturálására és tartalma megjelenítésére használtam. Segítségével definiáltam az elemeket, például címeket, szövegeket, képeket, hivatkozásokat stb.

CSS: Egy stílusleíró nyelv, amelyet a weboldal megjelenítési formázására használtam. Azáltal, hogy különböző stílusokat és tulajdonságokat adtam meg az elemeknek, mint például a szín, a betűtípus, a méret, a margók, a pozíció stb., a CSS lehetővé tette a weboldal testreszabását és a vizuális vonzereje növelését.

JavaScript: Egy programozási nyelv, amelyet az alkalmazás interaktív funkcióinak megvalósítására használtam. A JavaScript lehetővé teszi a weboldalam dinamikus működését, a felhasználói interakciókat, az adatok feldolgozását és a weboldalhoz kapcsolódó események kezelését.

2.1. ábra. HTML - CSS - JS logók

2.1.2. Backend

Backend oldalon a szerveroldali programozáshoz a Python programozási nyelvet választottam, amely nagy népszerűségnek örvend a könnyűsúlyú és rugalmas Flask keretrendszerrel kombinálva. A Python és a Flask együtt lehetővé teszik a hatékony és gyors fejlesztést, valamint az API-k létrehozását és a kérések kezelését.

Az adatbázis kezeléséhez az SQLAlchemy nevű Python könyvtárat használtam, melynek segítségével könnyedén lehet adatmodelleket definiálni és adatbázisműveleteket végezni. Ezáltal hatékony és strukturált adatkezelést biztosít az alkalmazásban.

Ezen technológiák kombinációja lehetővé teszi az alkalmazás teljeskörű működését, az interaktív felhasználói felülettől kezdve az adatok kezeléséig és tárolásáig. Az alkalmazás fejlesztése során a modern frontend és backend technológiák összekapcsolása révén egy hatékony és felhasználóbarát környezetet hoztam létre a kriptográfiai rendszerek kipróbálásához és tanulásához.

Python: A Python egy magas szintű programozási nyelv, amelyet egyszerű és olvasható szintaxisa jellemez. A Python nagyon rugalmas és sokoldalú, és széles körben használják a webfejlesztés, adatelemzés, mesterséges intelligencia, gépi tanulás és sok más területen. A Pythonban kifejezőerő és kényelem találkozik, ami lehetővé teszi a gyors és hatékony fejlesztést. Emellett a Python gazdag könyvtárökoszisztémával rendelkezik, amely számos előre elkészített funkcióval és modullal bővíti a fejlesztési lehetőségeket.

Flask: A Flask egy könnyűsúlyú és rugalmas webes keretrendszer Pythonban. A Flask lehetővé teszi a webalkalmazások gyors és hatékony fejlesztését, minimális konfigurációval és egyszerű szintaxisával. A Flask kiváló választás olyan projektekhez, amelyek kisebb méretűek vagy kevesebb komplexitást igényelnek, ugyanakkor rugalmasságot és skálázhatóságot biztosítanak. A Flask lehetőséget nyújt a beépített funkciók, mint például útvonalak, nézetek, sablonok és adatbázis kezelése egyszerű implementálására.

SQLAlchemy: Az SQLAlchemy egy Python alapú ORM (Object-Relational Mapping) keretrendszer, amely lehetővé teszi az adatbázisokhoz való könnyű és hatékony hozzáférést. Az ORM segítségével a fejlesztők Python objektumokat tudnak kezelni és manipulálni, miközben azokat az adatbázisban tárolják. Ezáltal az SQLAlchemy elrejti az adatbázis-specifikus részleteket és absztrakciós réteget biztosít a programozó számára.

2.2. ábra. Python - Flask - SQLAlchemy logók

2.2. Hasonló platformok, szoftverek összehasonlítása

Cryptii: A Cryptii egy online titkosítási és dekódolási weboldal, amely algoritmusok széles skáláját támogatja. Különböző szimmetrikus és aszimmetrikus titkosítási algoritmusok közül választhat. A weboldal emellett lehetőséget biztosít a kódolásra, a hashelésre és a tömörítésre is. A bevitel és kimenet között lehetőség van különböző rétegeket beiktatni, így a két végpont között több algoritmus is alkalmazható.

Encode-Decode: Az Encode-Decode egy olyan weboldal, amely különböző algoritmusokhoz kínál titkosítási és dekódolási szolgáltatásokat. A weboldal egy egyszerű felületet biztosít, ahol beírhatja a szöveget, és kiválaszthatja a kívánt titkosítási vagy visszafejtési módszert.

A Cryptii és Encode-Decode platform sem ad leírást a használt algoritmusokról, így ezeket csak úgy lehet hatékonyan használni amennyiben ismerjük azokat, vagy más forrásokból tájékozódunk róluk.

CyberChef: A CyberChef egy hatékony online eszköz az adatmanipulációhoz és a titkosításhoz/dekódoláshoz. A műveletek széles skáláját kínálja, beleértve az olyan titkosítási algoritmusokat, mint az AES, RSA, XOR és még sok más. Támogatja továbbá a kódolást, a hashelést, a tömörítést és más adattranszformációkat.

Rumkin: A Rumkin egy olyan weboldal, amely kriptográfiai eszközök széles gyűjteményét kínálja. Különböző algoritmusokhoz kínál lehetőségeket, mint például a Caesar rejtjelezés, a Rail Fence rejtjelezés, a Playfair rejtjelezés és még sok más. Emellett kódolásra, hashelésre és steganográfiára szolgáló eszközöket is tartalmaz.

dCode: A dCode egy olyan online platform, amely titkosítási és dekódolási eszközök hatalmas gyűjteményét kínálja. Algoritmusok, titkosítások és kódolások széles skáláját fedi le, beleértve a klasszikus titkosításokat, modern titkosítási módszereket és speciális kódolásokat. Emellett kriptoanalízishez is kínál eszközöket.

Ezeken az oldalakon már több, részletesebb leírást lehet kapni az algoritmusokról, bár már kevésbé felhasználóbarátok, mint az első két platform.

3. fejezet

Kriptográfiai alapok és rendszerek

3.1. Kriptográfiai alapfogalmak

A Cryptorithm rendszer használata során néhány alapvető kriptográfiai fogalom ismerete előnyös lehet. Az alábbiakban néhány ilyen fogalmat tisztázok, amelyek segíthetnek a rendszer hatékony használatában.

3.1.1. Titkosítás

A titkosítás olyan folyamat, amely során az eredeti üzenetet (nyílt szöveget) átalakítjuk egy titkosított formává, hogy csak a jogosultak tudják elolvasni. Az alkalmazásban található kriptográfiai rendszerek segítségével a felhasználók titkosíthatnak és visszafejthetnek üzeneteket.

3.1.2. Hash függvények

A hash-függvények olyan matematikai algoritmusok, amelyek egy tetszőleges hosszúságú bemenetet (pl. üzenetet) átalakítanak egy fix hosszúságú, látszólag véletlenszerű kimenetbe. Az alkalmazás SHA hashelésének megértése segít a felhasználóknak az adatok hitelességének ellenőrzésében.

3.1.3. Titkosítási módok

A titkosítási módok meghatározzák, hogy a titkosítás hogyan történik az üzenetek blokkjainak kezelése során. Az AES titkosító CTR módban való használata azt jelenti, hogy az üzenetek blokkonkénti titkosítása történik, amely nagyobb szabadságot ad a felhasználóknak az adatok kezelésében.

3.1.4. Rejtjelezések

Az Affin és Caesar rejtjelezések olyan egyszerű eltolásos módszerek, amelyeket a szövegek titkosítására használnak. A rendszerben ezek a rejtjelezések lehetővé teszik a felhasználók számára, hogy megértsék az ilyen egyszerű rejtjelezési módszerek működését és alkalmazását.

3.2. Caesar rejtjelezés

A Caesar rejtjelezés egy egyszerű eltolási titkosítási módszer, amelyben az összes betűt egy adott számmal, a kulcsként használt eltolással helyettesítik. Például, ha a kulcs 3, akkor az "A" betűt a "D" betűre cserélik, a "B" betűt az "E" betűre stb. A Caesar rejtjelezés könnyen feltörhető, mivel csak 26 lehetséges eltolási kulcs létezik, amelyeket egyszerűen végig lehet próbálni.

3.3. Affin rejtjelezés

Az Affin rejtjelezés egy egyszerű szubsztitúciós rejtjelezési módszer, amely a Caesar rejtjelezésre épül. Az Affin rejtjelezés egy lineáris transzformációt alkalmaz a betűkön, amely egy egyenlet alapján helyezi át azokat. Az Affin rejtjelezés a kulcsként használt két paraméter segítségével végez transzformációt az üzeneten. Az Affin rejtjelezés gyenge pontja, hogy az egyszerű frekvenciaanalízis módszerekkel feltörhető lehet, különösen kis méretű kulcsok esetén.

Fontos megjegyezni, hogy az Affin és Caesar rejtjelezések gyakorlati alkalmazásban már nem számítanak biztonságosnak, mivel könnyen feltörhetők. Az SHA és AES viszont biztonságos kriptográfiai algoritmusok, amelyek széles körben használatosak a valós világban.

3.4. SHA (Secure Hash Algorithm)

Az SHA egy hash függvény család, amelyeket a digitális adatok integritásának ellenőrzésére és az adatok egyedi azonosítására használnak. Az SHA hash függvények, például az SHA-1, SHA-256 stb., egy adott bemeneti üzenetet átalakítanak egy fix hosszú hash kóddá. Az SHA algoritmusok irreverzibilisek, vagyis a hash értékből nem lehet visszaállítani az eredeti üzenetet. Ez a tulajdonságuk hasznos a jelszavak, digitális aláírások és az üzenetek integritásának védelmében.

3.5. AES (Advanced Encryption Standard)

Az AES egy szimmetrikus blokk titkosítási algoritmus, amelyet a biztonságos adatátvitel és tárolás céljából használnak. Az AES algoritmus blokkokat titkosít a bemeneti üzenetből, és ezeket a titkosított blokkokat kombinálja a kimeneti titkosított üzenet létrehozásához. Az AES-t az Egyesült Államok Kormánya ajánlja a kormányzati és ipari alkalmazásokban. Az AES több különböző kulcsmérettel (128, 192, 256 bites) és különböző üzemmódokkal (pl. CTR, CBC, ECB) használható.

4. fejezet

Szoftver

4.1. Felhasználói követelmények

Regisztráció és bejelentkezés: Lehetőséget van regisztrációra az alkalmazásban, amely után a felhasználók bejelentkezhetnek a személyes fiókjukba. Ez lehetővé teszi számukra a mentett adatokhoz való hozzáférést.

Kriptográfiai rendszerek kiválasztása: Különböző kriptográfiai rendszerek elérhetőek, amelyeket bejelentkezés után lehet kipróbálni. Ehhez megjelenik egy menüt, ahol kényelmesen navigálhatnak a rendszerek között.

Rendszerek kipróbálása és tesztelése: A felhasználóknak lehetősége van arra, hogy kipróbálják és teszteljék a választott kriptográfiai rendszereket. Ehhez felhasználóbarát és interaktív felületet van biztosítva, ahol megadhatják a bemeneti adatokat, és láthatják a kimeneti eredményeket.

Tanulási anyagok és információk elérhetősége: Az alkalmazásnak tartalmaz egy tanulói felületet , ahol a felhasználók elérhetik a kriptográfiai rendszerekhez kapcsolódó részletes információkat és példafeladatokat. Ennek az oldalnak az eléréséhez nem szükséges a bejelentkezés.

Nyelvi támogatás: Az alkalmazás lehetőséget ad a felhasználóknak arra, hogy különböző nyelveken használják az alkalmazást. Ehhez egy egyszerű nyelv kiválasztó van biztosítva, amely lehetővé teszi a felhasználóknak a kívánt nyelv kiválasztását. A nyelvek rövidítve, ISO 639.1-es kódjuk szerint vannak feltüntetve.

4.2. Rendszerkövetelmény

A Cryptorithm rendszerkövetelménye funkcionális és nem funkcionális részekre oszlik. A funkcionális része tartalmazza az alkalmazás fő céljait és funkcionalitását, hogy hogyan kéne működjön a rendszer és milyen lehetőségei vannak a felhasználónak, míg a nem funkcionális rész kitér a rendszerrel szemben támasztott követelményekre, mint például a felhasználói élmény, többnyelvűség és a rendszer architektúrája.

4.2.1. Funkcionális

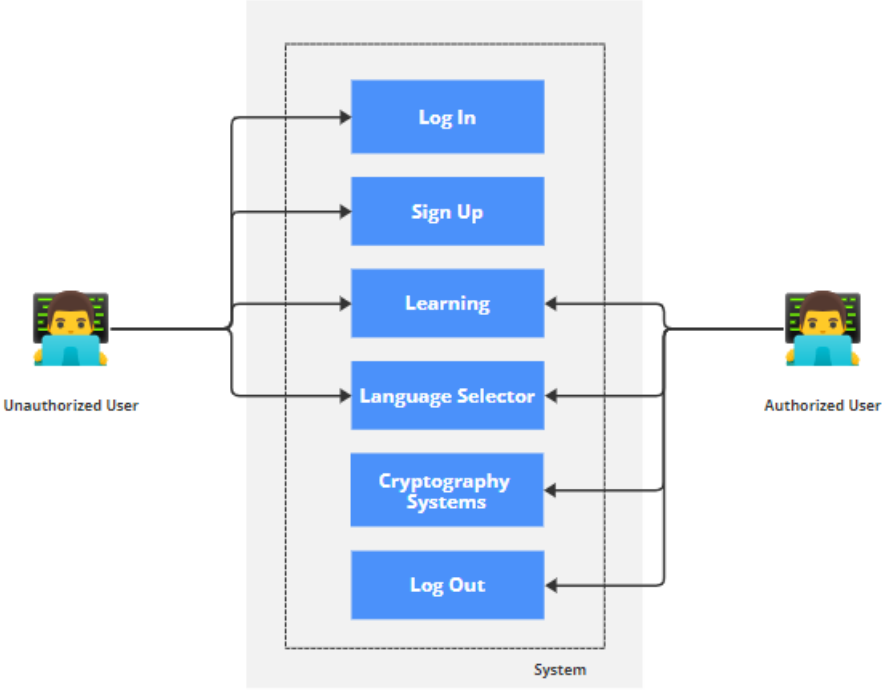
Kriptográfiai eszközök: Az alkalmazás választékot kínál a népszerű és ismert eszközökből az adatok titkosításához, visszafejtéséhez és transzformációjához. A felhasználók szabadon válogathatnak és kipróbálhatnak különböző lehetőségeket.

Tanulói felület: Az alkalmazás rendelkezik egy tanulói felülettel, ahol a felhasználók elmélyülhetnek a kriptográfiai rendszerek megértésében. Itt információkat olvashatnak és gyakorlati példákon keresztül elsajátíthatják ezeket a rendszereket.

Többnyelvűség: Az alkalmazás támogatja a többnyelvűséget, így elérhetővé válik idegen nyelvű felhasználók számára is. A felhasználók kiválaszthatják az anyanyelvüket vagy egyébb preferált nyelvet az alkalmazás használatához.

Visszajelzés: Amennyiben ismert kivétel keletkezik a rendszserben ezt felugró üzeneten keresztül van leközölve, ami ismerteti a hibát. Kivétel keletkezhet nem megfelelő beviteli adatok nyomán, például rossz vagy nem megfelelő kulcsméret megadása, vagy olyan oldalra való navigáció ahova a felhasználónak nincsen jogosultsága. Az előző esetet a hibának megfelelő üzenettel közlődik, míg utóbbi egy az érvényben levő jogosultságnak megfelelő oldalra navigálja a felhasználót.

Szerepkörök: A rendszert két fajta felhasználó számára elérhető, bejelentkezett és nem bejelentkezett. A bejelentkezett felhasználónak elérhető a weboldal összes funkcionalitása, ezzel szemben a nem bejelentkezett felhasználó nem jogosult a kriptográfia rendszerek használatára.



4.1. ábra. Use case diagram

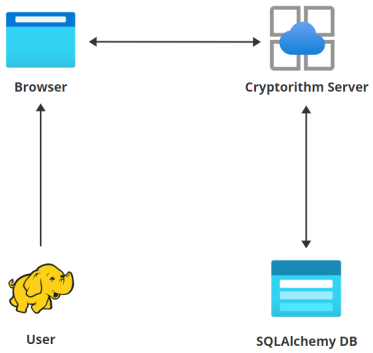
4.2.2. Nem funkcionális

- **Felhasználói élmény:** Az alkalmazás könnyen használható felhasználói felülettel rendelkezik. Az intuitív navigáció és a felhasználóbarát tervezés lehetővé teszi a felhasználók számára a könnyű kezelést és a zökkenőmentes interakciót az alkalmazással. Továbbá megfelelő felugró üzenetekkel jelez vissza a rendszer minden művelet elvégzése után.
- **Teljesítmény:** Az alkalmazás működése gyors és hatékony.
- **Biztonság:** Felhasználói adatok védve vannak, megfelelően vannak tárolva.
- **Skálázhatóság:** Az alkalmazás képes kezelni a megnövekedett felhasználói forgalmat és rugalmasan skálázódni.
- **Hibatűrés:** Hibakezelés és hibajavítás mechanizmusainak megléte, az alkalmazás ellenálló képessége van a hibákhoz. Redszer leállás esetén az újraindítás nem időigényes.
- **Reszponzívitás:** A weboldal felülete főként asztali számítógépekre, laptopokra van optimalizálva, továbbá támogatja a népszerű böngészőket.

4.2. ábra. Népszerűbb Böngészők

4.3. A rendszer architektúrája

Az alkalmazás architektúrája a következő komponenseket kombinálja, hogy a felhasználók kényelmesen használhassák a Cryptorithm alkalmazást. A felhasználói interfész réteg lehetővé teszi a felhasználók interakcióját az alkalmazással, az üzleti logika réteg hajtja végre a szükséges műveleteket és feldolgozza a kéréseket, míg az adatbázis réteg biztosítja az adatok tartós tárolását és kezelését. Az architektúra segít a rendszer komponenseinek szétválasztásában és a fejlesztés hatékonyságának növelésében.



4.3. ábra. Rendszer Architektúra

Felhasználói interfész: Ez a rész felelős az alkalmazás felhasználói felületének megjelenítéséért és a felhasználóval való interakcióért. Megvalósítása az alkalmazásban HTML, CSS és JavaScript segítségével van kivitelezve. Ez a réteg jeleníti meg a kriptográfiai rendszerek kiválasztására, tesztelésére és a tanulói anyagokhoz való hozzáférésre szolgáló felületeket.

Üzleti logika: Az üzleti logika réteg felelős az alkalmazás működéséért, a felhasználók által végrehajtott műveletek feldolgozásáért és az eredmények előállításáért. Itt találhatóak a szerveroldali Python fájlok, amelyek a kriptográfiai műveletek, adatfeldolgozás és adatbázis-interakciók végrehajtásáért felelősek. Flask keretrendszert használva ez a réteg kezeli a HTTP kéréseket és válaszokat.

Adatbázis: Az alkalmazás adatbázisában tárolódnak a felhasználói adatok, például a regisztrált felhasználók adatai és előzményei. Az adatbázis kezelésére az SQLAlchemy Python könyvtárat használtam, amely lehetővé teszi a könnyű adatbázis-műveletek végrehajtását és az adatmodell definiálását.

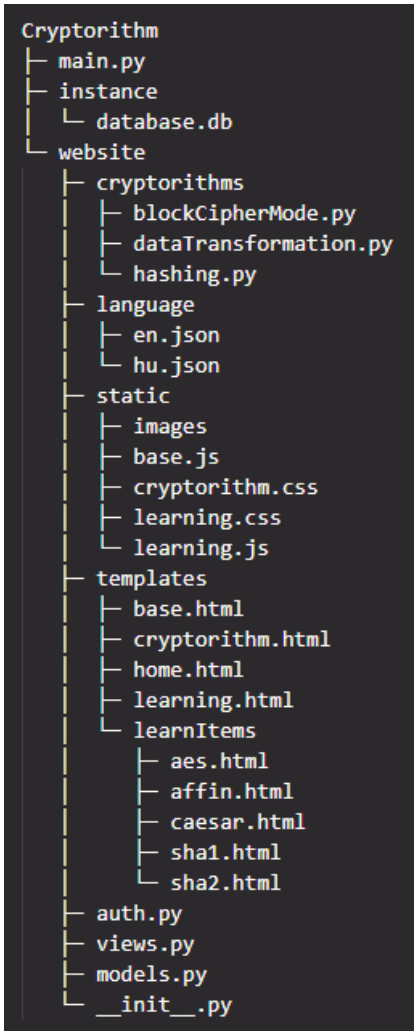
5. fejezet

Tervezés és megvalósítás

A projektek a Visual Studio Code segítségével fejlesztettem, amely egy ingyenes és nyílt forráskódú fejlesztői környezet. Bár alapvetően könnyűsúlyú, ennek ellenére erőteljes szövegszerkesztő, mivel számos programozási nyelvet és technológiát kínál. A telepítése után nagyban testreszabhatjuk saját igényeink szerint és számos egyébb funkciókhoz juthatunk a bővítmények telepítésével.

6 részre bontottam a projektet: instance, cryptorithms, languages, static, templates, cryptorithm

- **Instance**: Az instance mappában van eltárolva az adatbázis lokálisan, melynek modelje a models.py fájlban van definiálva.
- **Cryptorithms**: A cryptorithms mappában levő fájlokban kerültek lettek kivitelezve és implementálva a kriptográfiai rendszerek.
- **Languages**: A nyelvekért felelős .json kiterjesztésű fájlok itt vannak tárolva, itt van egy meghatározott struktúrában lefordított tartalma az oldalnak.
- **Static**: A static alatt a .css és .js fájlok kerültek létrehozásra, melyek a weboldal dizájnért felelősek.
- **Templates**: A template mappába kerültek az alap .html fájlok amelyek a tartalom megjelenitéséért felelősek.
- **Cryptorithm**: A fő mappa a Cryptorithm amely magába foglalja az előbb említett részeket, valamint az API kéréseket és válaszokat, a szessziókezeléshez, a projekt inicializálásához szükséges fájlokat, továbbá az adatbázis adatmodelljét és a projektet indító main fájlt.



5.1. ábra. Rendszer fájl struktúrája

5.1. Könyvtárak

- **Cryptography:** A Cryptography egy Python könyvtár, amely kriptográfiai funkciókat és algoritmusokat kínál. Ez a könyvtár ideális választás a projektben, mivel számos kriptográfiai műveletet valósíthatunk meg vele, például hashelést, titkosítást és visszafejtést. A Cryptography megbízható és jól dokumentált eszköz a kriptográfiai műveletek biztonságos végrehajtásához. Jelen esetben ennek a könyvtárnak a segítségével lettek kivitelezve a modernebb rendszerek, például az SHA és az AES.

- **Flask:** A Flask egy könnyű súlyú, de erőteljes webes alkalmazások fejlesztésére szolgáló Python mikrokeretrendszer. A projektben a Flask keretrendszert használom a webes alkalmazás felépítéséhez és a kérések kezeléséhez. A Flask könnyen tanulható és rendkívül rugalmas, ami lehetővé tette, hogy könnyedén kialakítsam a vágyott funkciókat és testreszabhasamk az alkalmazást.

- **FlaskSQLAlchemy:** A FlaskSQLAlchemy egy könnyen használható és hatékony ORM (Object-Relational Mapping) könyvtár, amely lehetővé teszi az adatbázis műveletek kezelését a Flask alkalmazásban. Az SQLAlchemy révén a FlaskSQLAlchemy segít az adatbázis kapcsolatok kezelésében, az adatmodell osztályok definiálásában és az adatbázis műveletek végrehajtásában. A FlaskSQLAlchemy használata átlátható és hatékony adatbázis-interakciókat tesz lehetővé a projektben.

- **FlaskLogin:** A FlaskLogin egy hasznos kiegészítő a Flask keretrendszerhez, amely segít az autentikáció és az azonosítás kezelésében a webes alkalmazásban. A FlaskLogin segítségével egyszerűen implementálhattam a felhasználói regisztrációt, bejelentkezést és kijelentkezést a rendszerben. Ez a könyvtár nagyban megkönnyíti a felhasználói munkamenetek kezelését és a hozzáférési jogosultságok ellenőrzését.

- **Json:** A JSON (JavaScript Object Notation) egy könnyen olvasható és írható adatformátum, amely széles körben használatos az adatok strukturált tárolására és átvitelére. A JSON könyvtárat használva a projektben könnyedén kezelhettem a JSON adatokat, például ennek segítségével valósítottam meg a többnyelvűséget.

- **Base64:** A base64 egy olyan kódolási formátum, amely lehetővé teszi bináris adatok átalakítását olvasható szöveggé. A base64 könyvtárat használva a projektben a bináris adatokat konvertálhatjuk base64 formátumba, ami könnyen kezelhető és továbbítható a webes alkalmazásban. Ennek használata szükséges volt, hogy hatékonyan lehessen alkalmazni a Cryptography könyvtárból használt fügvényeket.

- **Glob:** A glob egy Python könyvtár, amely lehetővé teszi a fájl- és mappaútvonalak kezelését. A glob könyvtár segítségével könnyedén kezelhettem a fájlok vagy mappák neveinek gyűjteményét. Bizonyos fájlok helyzetmeghatározásához használtam, például a .json fájlok eléréséhez.

- **Os:** Az os (operating system) egy Python könyvtár, amely lehetővé teszi az operációs rendszerrel kapcsolatos műveletek végrehajtását. Az os könyvtárat használva a projektben könnyedén manipulálhattam a fájlokat és könyvtárak, például ezek létrehozását, törlését vagy módosítását.

- **Math:** A math egy beépített Python könyvtár, amely matematikai függvényeket és konstansokat kínál. A math könyvtárat használva a projektben matematikai műveleteket végezhetünk, például számításokat végezhetünk, véletlenszámokat generálhatunk vagy trigonometriai műveleteket hajthatunk végre. Jelen esetben a Caesar és Affin rejtjelezéseknek implementálásában használtam.

- **Hashlib:** A hashlib egy Python könyvtár, amely hash-függvényeket kínál. A hashlib segítségével a projektben különböző hashelési műveleteket végezhettem, például az SHA hashelést, amelyek a jelszavak biztonságos tárolásában és a megfelelő műveletek elvégézésére használtam.
- **Random:** A random egy beépített Python könyvtár, amely véletlenszámok generálására szolgál. A random könyvtárat használva a projektben véletlenszerű adatokat generálhatunk, például az egyedi azonosítókat a szükséges rendszerekhez.

5.2. Felhasználói dokumentáció

6. fejezet

Továbbfejlesztési lehetőségek

Bővített kriptográfiai rendszerek: Az alkalmazásba további kriptográfiai rendszerek, mint például Salsa20, ElGamal vagy ECC (Elliptic Curve Cryptography) integrálása lehetőséget adna a felhasználóknak a különböző algoritmusok kipróbálására és tanulására.

Interaktív gyakorló feladatok: Interaktív gyakorló feladatok készítése, amelyek segítségével a felhasználók a tanulói felületen gyakorolhatják a különböző kriptográfiai rendszerek alkalmazását. Ez javítaná a gyakorlati tapasztalatok szerzését és a tanulás hatékonyságát.

Felhasználói profilok és eredmények nyomon követése: Egy felhasználói profil rendszer létrehozása, ahol a felhasználók modosíthatják adataikat és nyomon követhetik a teljesítményt, például az elért eredményeket vagy az elvégzett feladatokat. Ez segít a felhasználóknak a fejlődésük nyomon követésében és motivációjuk fokozásában.

További nyelvi támogatás: Az alkalmazást további nyelvek támogatásával való kibővítése, hogy a nem angol vagy magyar anyanyelvű felhasználók is könnyen használhassák és megértsék a rendszert. Ez növelné a felhasználói kört és a felhasználói elégedettséget.

Reszponzív dizájn: Az alkalamzás biztosítása a különböző eszközökön történő használathoz.

Ezek a továbbfejlesztési lehetőségek hozzájárulhatnak az alkalmazás funkcionalitásának és felhasználói élményének javításához, valamint a felhasználók kriptográfiai ismereteinek bővítéséhez.

Összefoglaló

A Cryptorithm egy webes alkalmazás, amely lehetővé teszi a felhasználók számára különböző kriptográfiai rendszerek kipróbálását és tanulását. Az alkalmazásban számos beépített kriptográfiai eszköz áll rendelkezésre, mint például az SHA hashelés, az AES titkosítás CTR módban, valamint az Affin és Caesar rejtjelezések. A felhasználók a felületen keresztül kiválaszthatják, hogy melyik rendszert szeretnék alkalmazni, és gyakorolhatnak a megértéséhez. Az alkalmazás intuitív felhasználói felülettel rendelkezik és támogatja a többnyelvűséget. A Cryptorithm célja, hogy segítse a felhasználókat a kriptográfiai ismeretek bővítésében, miközben egy interaktív és tanulást támogató környezetet biztosít.

Köszönetnyilvánítás

Szeretném kifejezni hálámat a családomnak és barátaimnak, akik támogattak és bátorítotak engem ezen az úton. Köszönet illeti a vezető tanáromat is, aki értékes útmutatást és iránymutatást nyújtott a projekt fejlesztése során.

Ábrák jegyzéke

Táblázatok jegyzéke

Irodalomjegyzék

[Ant07] Margit Antal. Toward a simple phoneme based speech recognition system. *Stud. Univ. Babeş-Bolyai Inform.*, 52(2):33–48, 2007.

[Kó8] Zoltán Kátai. Dynamic programming as optimal path problem in weighted digraphs. *Acta Math. Acad. Paedagog. Nyházi. (N.S.)*, 24(2):201–208, 2008.

[Knu11] Donald E. Knuth. *The Art of Computer Programming: Combinatorial Algorithms, Part 1.* Addison-Wesley Professional, 1st edition, 2011.

[LS15] László Lovász and Balázs Szegedy. The automorphism group of a graphon. *J. Algebra*, 421:136–166, 2015.

Függelék