

Kodolányi János Egyetem

TDK dolgozat: Social AI

Budapest

2023

Tartalomjegyzék

RÖVIDÍTÉSEK JEGYZÉKE	4
ÁBRAJEGYZÉK	5
TÁBLÁZATOK JEGYZÉKE	5
KIVONAT.....	6
Bevezetés.....	7
1. A szoftver tervezése	8
1.1. Alkalmazott technológiák és nyelvek.....	8
1.2. Alkalmazott külső könyvtárak (Dependencies)	11
1.3. Támogatott böngészők	12
1.4. Architektúra terv.....	13
1.5. Az API tervezése és dokumentációja	13
1.6. Facebook Access Token.....	16
1.7. Adatszerkezet	16
2. A szoftver kialakítása	17
2.1. Lokális fejlesztői környezet kialakítása	17
2.2. Google Cloud beállítása	18
2.3. Meta for Developers.....	20
2.4. ChatGPT és Pexels.com implementáció	22
2.5. A ChatGPT támogató szerepe	24
2.5.1. Swagger dokumentáció generálás	24
2.5.2. Algoritmus szintaxis hibák javítása.....	24
2.5.3. Programozási könyvtárakkal kapcsolatos tanácsadás.	24
2.5.4. Magyarázattal egybekötött fordítás	24
2.6. Fontend fájlok és komponensek	25
3. Éles környezet üzembe helyezése	27
3.1. Azure CosmosDB létrehozása és beállítása	27

3.2. Azure Static Web Apps létrehozása és beállítása	29
3.3. Azure Functions létrehozása és beállítása	30
3.4. GitHub Actions és Azure DevOps CI/CD folyamatok dokumentációja	31
3.5. Domain vásárlása és beállítása, Azure DNS-zóna	33
3.6. Azure DNS-zóna létrehozása és beállítása	34
4. Az IT biztonság kialakítása	35
4.1. Üzemeltetési felelősségek felosztása.....	36
4.2. Felelősségi körömbé tartozó biztonsági intézkedéseim	37
4.2.1. Frontend védelem - HTTPS tanúsítvány	37
4.2.2. Backend védelem – API Secret	38
4.2.3. Monitorozás, naplózás és riasztás.....	38
4.2.4. Azure portál - Kétlépcsős hitelesítés	38
4.2.5. Felhasználók bejelentkezési adatainak védelme	38
4.2.6. Adatbázis védelem.....	38
4.3. Jogi felelősség, megfelelés, GDPR	39
4.3.1. Adatvédelmi szabályzat.....	39
4.3.2. Általános Szerződési Feltételek.....	39
4.3.3. GDPR megfelelés	39
5. Tesztelés	40
5.1. Új teszt felhasználó felvételének menete	40
5.2. A teszt.....	41
Összegzés	42
HIVATKOZÁSJEGYZÉK.....	43
IRODALOMJEGYZÉK.....	45
MELLÉKLET.....	46

RÖVIDÍTÉSEK JEGYZÉKE

- DevOps: Development and Operations (Fejlesztés és Üzemeltetés)
- CI: Continuous Integration (Folyamatos Integráció)
- CD: Continuous Deployment (Folyamatos Telepítés)
- API: Application Programming Interface (Alkalmazásprogramozási interfész)
- DNS: Domain Name System (Domain Névszerver)
- ID: Identifier (Azonosító)
- OAuth: Open Authorization (Nyílt Hitelesítés)
- CORS: Cross-Origin Resource Sharing (Kereszt eredetű erőforrás megosztás)
- NPM: Node Package Manager (Node Csomagkezelő)
- PIP: Package Installer for Python (Csomag Telepítő Pythonhoz)
- URL: Uniform Resource Locator (Egységes erőforrás-azonosító)
- CSS: Cascading Style Sheets (Lépcsőzetes stíluslapok)
- ENV: Environment (Környezet)
- JS: JavaScript (Programozási nyelv)
- JSX: JavaScript XML (JavaScript XML)
- JSON: JavaScript Object Notation (JavaScript objektum leírás)
- YAML: YAML Ain't Markup Language (YAML nem jelölési nyelv)
- CNAME: Canonical Name (Kanonikus név)
- TTL: Time to Live (Élettartam)
- WWW: World Wide Web (Világháló)
- UUID: Universally Unique Identifier (Egyetemesen egyedi azonosító)
- SDK: Software Development Kit (Szoftverfejlesztő eszközkészlet)
- GSI: Google Sign-In (Google bejelentkezés)
- DB: Database (Adatbázis)
- NS: Name Server (Névszerver)
- HTTP: Hypertext Transfer Protocol (Hipertext átviteli protokoll)
- HTTPS: Hypertext Transfer Protocol Secure (Biztonságos hipertext átviteli protokoll)
- HTML: Hypertext Markup Language (Hipertext jelölési nyelv)
- AI: Artificial intelligence (Mesterséges intelligencia)
- PaaS: Platform as a Service (Szolgáltatásként nyújtott platform)

- On-prem: On-premises (Helyszíni infrastruktúra)
- IaaS: Infrastructure as a Service (Infrastruktúra szolgáltatásként)
- SaaS: Software as a Service (Szoftver szolgáltatásként)
- ÁSZF: Általános Szerződési Feltételek
- GDPR: General Data Protection Regulation (Általános adatvédelmi rendelet)
- ChatGPT: Chat Generative Pre-trained Transformer

ÁBRAJEGYZÉK

1. ábra: Rendszer architektúra	13
2. ábra: Az alkalmazás API dokumentációja.....	15
3. ábra: Facebook Access Token lekérdezésének folyamata.....	16
4. ábra: Google Cloud OAuth consent screen beállítások	18
5. ábra: Google Cloud OAuth 2.0 Client ID beállítások	19
6. ábra: Meta for Developers APP beállítások	20
7. ábra: Meta for Developers APP beállítások	21
8. ábra: Azure Cosmos DB-fiók áttekintés.....	27
9. ábra: Azure Cosmos DB adatkezelő.....	28
10. ábra: Azure Static Web Apps áttekintés	29
11. ábra: Azure Functions áttekintés	30
12. ábra: DevOps Architektúra ábra.....	31
13. ábra: Domain vezérlő felület, Rackhost	33
14. ábra: Azure DNS-zóna beállítások	34
15. ábra: Adatközpont-infrastruktúra	35
16. ábra: Üzemeltetési felelősségi felosztás	36
17. ábra: HTTPS tanúsítvány	37
18. ábra: Meta for Developers új tesztelő felvétele	40

TÁBLÁZATOK JEGYZÉKE

1. táblázat: Alkalmazott technológiák és nyelvek	10
2. táblázat: Alkalmazott külső könyvtárak (Dependencies)	11
3. táblázat: Támogatott böngészők	12
4. táblázat: Azure DNS-zóna, beállított DNS rekordok	34

KIVONAT

A cél egy szoftver fejlesztése és üzembe helyezése felhő (Microsoft Azure), valamint mesterséges intelligencia (ChatGPT) technológiák alkalmazásával, mely a közösségi média (Facebook) oldalakon lévő tartalomgyártás automatizációját valósítja meg részlegesen.

Célcsoportok, hasznosság:

1. A szoftver célcsoportja: Azok a vállalkozások, akik a közösségi média jelenlétüket mesterséges intelligencia alapú szoftverrel szeretnék automatizálni.
2. A szakdolgozat célcsoportja: Informatikai szakemberek, akik érdeklődnek a felhő, valamint a mesterséges intelligencia technológiák gyakorlati alkalmazása iránt.

Feladatok, szerkezet:

- ☐ Bevezetés (terjedelem 1 oldal)
- ☐ A szoftver tervezése (terjedelem a dolgozat ~25%-a)
- ☐ A szoftver kialakítása (terjedelem a dolgozat ~25%-a)
- ☐ A szoftver üzembe helyezése (terjedelem a dolgozat ~15%-a)
- ☐ IT biztonság kialakítása (terjedelem a dolgozat ~10%-a)
- ☐ Tesztelés (terjedelem a dolgozat ~5%-a)
- ☐ Tantárgyak szakdolgozati kapcsolata (terjedelem a dolgozat (~10%-a)
- ☐ Összegzés (terjedelem 1 oldal)

Az üzembe helyezés során DevOps módszertant, azon belül számos (pl.: CI, CD) gyakorlatot, valamint az ehhez szükséges eszközöket is alkalmazom. (pl.: Azure DevOps, GitHub). A DevOps CI/CD folyamatok teljeskörűen megvalósultak, így a további fejlesztések automatizáltan települnek. (lásd 3.4 fejezet)

A biztonságra figyelmet fordítok mind a szoftver fejlesztése, mind az üzembe helyezése során, mivel tanulmányaim során az IT biztonság specializációt választottam. (lásd 4. fejezet). Sikerült a teljes fejlesztés tartalmát (minden komponenst, tehát a Frontend alkalmazást, Backend rendszert, adatbázist) éles (lásd 3. fejezet) és biztonságos (lásd 4. fejezet) környezetbe üzembe helyezni a Microsoft Azure publikus felhő infrastruktúrájában.

Törekedtem arra, hogy az alkalmazás üzemeltetési költségeit alacsonyan (~0 Ft / hó) tartsam, melyet sikerült közel a nullán tartani, melyet számlákkal tudok igazolni (lásd 3., 4., 5., 6. mellékletek) mindamelllett, hogy az alkalmazás képes a skálázásra.

BEVEZETÉS

A szakdolgozatom célja, hogy bemutassam, hogyan lehet közösségi média oldalakat (pl.: facebook) menedzselő, mesterséges intelligencia (ChatGPT) alapú funkciókat is tartalmazó szoftvert fejleszteni, mely részben automatizálja a tartalomgyártást, valamint hogyan lehet üzembe helyezni felhő technológiával (Azure).

Egy webalkalmazást fogok fejleszteni, mivel egy webalkalmazás számos (pl.: Windows, IOS, Android) operációs rendszeren elérhető.

Több Microsoft (pl.: Azure Functions, Azure DevOps), Google (pl.: Identity Platform), Meta (pl.: Meta for Developers) terméket és szolgáltatást is alkalmazok és integrálok a fejlesztés során (lásd 1. táblázat), melyeket a továbbiakban tételesen részletezek. (lásd 1.1 fejezetben) A fejlesztés során számos programozási (pl.: Python, Javascript), leíró (pl.: HTML, CSS) és lekérdező (pl.: SQL) nyelvet fogok alkalmazni, melyet összesítek, lásd 1. táblázat.

A következő fejezetekben részletesen tárgyalom a szoftver tervezését (lásd 1. fejezetben), a szoftver kialakítását (lásd 2. fejezetben), a szoftver üzembe helyezését Microsoft Azure publikus felhő infrastruktúrájában (lásd 3. fejezetben). A biztonságra figyelmet fordítok, mind a szoftver fejlesztése, mind az üzembe helyezése során, mivel tanulmányaim során az IT biztonság specializációt választottam. (lásd 4. fejezetben)

Ezen kívül törekszem arra, hogy az alkalmazás üzemeltetési költségeit alacsonyan (~0 Ft / hó) tartsam.

Az üzembe helyezés során DevOps módszertant, azon belül számos (pl.: CI, CD) gyakorlatot, valamint az ehhez szükséges eszközöket is alkalmazom. (pl.: Azure DevOps, GitHub)

A dokumentáció végén tárgyalom a tesztelési eredményeket (lásd 5. fejezet), valamint felvázolom a további fejlesztési lehetőségeket. (lásd „Összegzés” fejezet)

Köszönetemet szeretném kifejezni mindazoknak, akik támogattak a szakdolgozat elkészítése és a szoftver fejlesztése során, Dr. Rikk Jánosnak és Dr. Pitlik Lászlónak.

1. A SZOFTVER TERVEZÉSE

1.1. Alkalmazott technológiák és nyelvek

Céлом egy modern és költséghatékony megoldás létrehozása, amely nem csak az aktuális igényeket elégíti ki, hanem a jövőbeni növekedést és megnövekedett igényeket is kiszolgálja, miközben stabil és skálázható marad.

„Az évtized legforradalmibb és talán legnépszerűbb IT infrastruktúra megoldása a felhő technológia.” (Rubóczki Edit Szilvia, http://lib.uni-obuda.hu/sites/lib.uni-obuda.hu/files/Ruboczki_Edit_Szilvia_ertekezes.pdf 59. oldal, Letöltve 2023.10.07.), ezért az alkalmazás fejlesztéséhez és üzemeltetéséhez egy felhőalapú infrastruktúrát, a Microsoft Azure-t (<https://azure.microsoft.com>) fogom alkalmazni. Az Azure „Always Free” (<https://azure.microsoft.com/en-us/pricing/free-services/>) szolgáltatásokat és platformokat alkalmazom, melyek lehetővé teszik a kezdeti fejlesztést és a későbbi skálázást a növekvő igényekhez igazodva.

A backend rendszer kialakításához az Azure Functions-t (<https://azure.microsoft.com/hu-hu/products/functions/>) választottam, amely felhőalapú funkciók fejlesztését és üzemeltetését teszi lehetővé. Python nyelvet választottam az Azure Function lefejlesztéséhez, mivel a Microsoft teljeskörűen támogatja a Python nyelvet az Azure Function fejlesztéséhez (<https://learn.microsoft.com/en-us/azure/azure-functions/functions-reference-python?pivots=python-mode-decorators&tabs=asgi%2Capplication-level>), valamint az egyik fő funkciója az alkalmazásnak egy ChatGPT modell (<https://openai.com/blog/chatgpt>) alapú funkció, melyhez szintén elérhető hivatalos OpenAI Python könyvtár (<https://platform.openai.com/docs/libraries>). Az adatbázis kezeléséhez az Azure Cosmos DB-t (<https://azure.microsoft.com/hu-hu/free/cosmos-db/>) választottam, ami egy horizontálisan skálázható NoSQL adatbázis, amely képes integrálódni az Azure Functions-szel. (<https://learn.microsoft.com/en-us/azure/azure-functions/functions-add-output-binding-cosmos-db-vs-code?tabs=in-process%2Cv1&pivots=programming-language-python>).

A frontend részhez (webfelület) a ReactJS (<https://react.dev>) keretrendszert választottam, amely dinamikus felhasználói felületek készítésére és kezelésére szolgál. A JavaScript nyelvet használom a ReactJS-ben, valamint HTML és CSS nyelveket is alkalmazok a

tartalom megjelenítéséhez és stílusozásához. Az Azure Static Web Apps (<https://azure.microsoft.com/en-us/products/app-service/static>) platformot választottam a ReactJS alkalmazás üzembe helyezéséhez, mivel a platform teljeskörűen támogatja a React alkalmazások üzemeltetését (<https://learn.microsoft.com/en-us/azure/static-web-apps/getting-started?tabs=react>).

Az autentikáció és felhasználói hitelesítés terén a Google Identity Platformot (<https://cloud.google.com/identity-platform>) integrálom, mely egyszerű bejelentkezést tesz lehetővé a felhasználóknak a Google fiókjukkal, és nem szükséges a leendő felhasználóknak fiókot regisztrálniuk, ha már rendelkeznek Google fiókkal (gmail címmel).

A verziókezeléshez a Git-et (<https://git-scm.com>) választottam, ami egy elosztott verziókezelő rendszer. A forráskódok tárolására a GitHub (<https://github.com>) és az Azure DevOps (<https://azure.microsoft.com/en-us/products/devops>) platformokat használom, amelyek lehetőséget nyújtanak a közös munkára és a fejlesztési folyamat nyomon követésére, valamint támogatják a CI/CD folyamatokat.

A folyamatos integráció és folyamatos szállítás (CI/CD) folyamatokhoz a GitHub Actions-t (<https://github.com/features/actions>), valamint az Azure Pipelines-t (<https://azure.microsoft.com/en-us/products/devops/pipelines>) használom, amelyek automatizált tesztelést, build folyamatokat, a környezeti változók tárolását, és az alkalmazások gyors és folyamatos szállítását teszik lehetővé. A CI/CD folyamatokat YAML nyelven definiálom.

Az alkalmazás DNS kezeléséhez az Azure DNS-t (<https://azure.microsoft.com/en-us/services/dns/>) választottam. Az Azure DNS egy felhőalapú DNS szolgáltatás, amely lehetővé teszi a domain nevek regisztrációját és kezelését az Azure környezetben. Ez fontos része a felhőalapú infrastruktúrának, mivel biztosítja a domain nevek feloldását az alkalmazás webfelületéhez.

A képkereső motorhoz a Pexels.com (<https://www.pexels.com/>) platformot választottam. Az alkalmazásban API integrációt kell kialakítani, mivel a Pexels API hívásokon keresztül lehetőség van a képek keresésére, letöltésére és megjelenítésére az alkalmazás felületén.

Az alábbi (1. táblázat) összesíti az alkalmazott technológiákat és nyelveket:

1. táblázat: Alkalmazott technológiák és nyelvek

Technológiai komponensek	Alkalmazott rendszerek és nyelvek (URL-ek az 1.1 fejezetben)
Alkalmazott Cloud platform	Microsoft Azure
DNS kezelő rendszer	Azure DNS
Backend keretrendszer	Azure Functions
Backend rendszerénél alkalmazott nyelvek	Python, SQL, JSON
Backend rendszer üzembehelyezésének rendszere	Azure Function App
Frontend keretrendszer	React JS
Frontend rendszerénél alkalmazott nyelvek	Javascript, HTML, CSS
Frontend rendszer üzembehelyezésének rendszere	Azure Static Web Apps
Frontend autentikáció és felhasználói hitelesítés	Google Identity Platform
Verziókezelés	Git
Forráskódok tárolása	GitHUB, Azure DevOps
Backend rendszer CI/CD folyamat platform	GitHUB Actions
Frontend rendszer CI/CD folyamat platform	Azure DevOps
CI/CD folyamatoknál alkalmazott nyelv	YAML
API dokumentációnál alkalmazott nyelv	YAML
Adatbázis	Azure Cosmos DB
Adatbázis kezelésénél alkalmazott nyelvek	SQL, JSON
Közösségi média platform	META (Facebook)
Mesterséges intelligencia motor	OpenAI
Kép keresőmotor	Pexels
Lokális kompatibilis futtató környezetek	Linux, Windows, MacOS
CI/CD folyamatok futtató környezete	Linux

1.2. Alkalmazott külső könyvtárak (Dependencies)

A szoftverfejlesztés során előfordul, hogy különböző (lásd 2. táblázat) külső könyvtárakat és csomagokat használunk, hogy könnyebben (kevesebb fejlesztési idő alatt) elvégezhessük a feladatokat, és elnyerjük a kívánt funkcionalitást.

Az Azure Functions backend rendszer fejlesztése során különböző (lásd 2. táblázat) Python könyvtárakat alkalmazok, ezek a könyvtárak segítik a backend funkcionalitás megvalósítását.

A React frontend rendszer fejlesztése során több (lásd 2. táblázat) Node Package Manager (NPM) csomagot használok, ezek a csomagok segítenek a frontend funkciók és a felhasználói felület kialakításában, megkönnyítik (pl.: felgyorsítják) a React alapú alkalmazás fejlesztést.

Az általam alkalmazott külső könyvtárak és függőségek a következők:

2. táblázat: Alkalmazott külső könyvtárak (Dependencies)

Dependency típusok	Dependencies
Azure Functions Backend rendszerénél alkalmazott függőségek Preferred Installer Program (PIP) csomagok	openai, azure-cosmos, azure-functions, uuid
React Frontend rendszerénél alkalmazott Node Package Manager (NPM) csomagok	axios, bootstrap, jwt-decode, react, react-dom, react-router-dom, react-scripts
React Frontend rendszerénél alkalmazott scriptek	Google GSI Client, Facebook SDK for Javascript

Azure Functions backend rendszer:

- openai: Ez a Python könyvtár lehetővé teszi a GPT-3.5 nyelvi modell használatát.
- azure-cosmos: Ez a könyvtár segíti az adatbázis műveletek kezelését az Azure Cosmos DB-ben.
- azure-functions: Ez a könyvtár lehetővé teszi Azure Functions alkalmazások fejlesztését és kezelését Pythonban.
- uuid: Ez a könyvtár segít az egyedi azonosítók (UUID) generálásában.

React frontend rendszerbe beépített scriptek:

- Google GSI Client: Ez a script a Google Sign-In integrációhoz szükséges, és lehetővé teszi a felhasználók Google-fiókkal történő bejelentkezését.
- Facebook SDK for JavaScript: Ez a script a Facebook integrációjához szükséges, és lehetővé teszi a felhasználók által kezelt facebook oldal adatok beolvasását, valamint a felhasználóktól hiteles jóváhagyás kérését az oldalak kezeléséhez.

React frontend rendszer:

- bootstrap: Ez egy népszerű CSS és JavaScript keretrendszer, amely segít az egyszerű és responszív webes dizájn kialakításában.
- jwt-decode: Ez a könyvtár segíti a JSON Web Token (JWT) dekódolását, amely a felhasználó azonosítására és hitelesítésére szolgál.
- react: Ez a könyvtár az alapja a React alkalmazások fejlesztésének.
- react-dom: Ez a könyvtár segíti a React komponensek renderelésében a böngészőben.
- react-router-dom: Ez a könyvtár segíti a React alapú alkalmazások útvonalkezelését a böngészőben.
- react-scripts: Ez a könyvtár tartalmazza a fejlesztői eszközöket és parancsokat a React alkalmazások futtatásához és buildeléséhez.

1.3. Támogatott böngészők

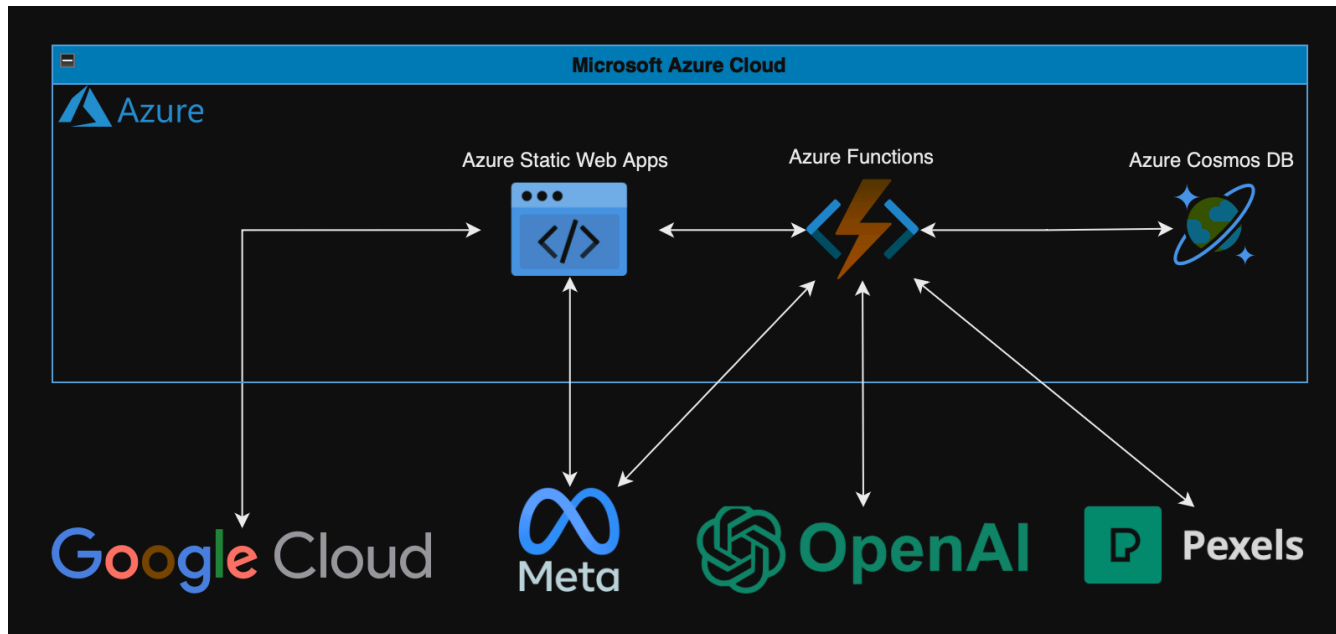
A szakdolgozatban bemutatott webalkalmazást a következő (lásd 3. táblázat) böngészőkkel tervezem kompatibilisnek, így tehát a táblázatban szereplő böngészők kompatibilitására vállalom garanciát:

3. táblázat: Támogatott böngészők

Támogatott böngészők
Safari
Microsoft Edge
Google Chrome

1.4. Architektúra terv

Az alábbi (1. ábrán) látható a rendszer architektúráját és az egyes komponensek közötti kapcsolatokat bemutató ábra:



1. ábra: Rendszer architektúra

Forrás: Saját készítésű ábra

1.5. Az API tervezése és dokumentációja

Az API tervezéséhez és dokumentálásához a Swagger.io-t (<https://editor.swagger.io>) használtam, amely egy nyílt forráskódú eszköz a RESTful API-k leírásához és dokumentálásához. A Swagger.io lehetővé tette számomra, hogy részletesen meghatározzam az API végpontokat, paramétereket, válaszokat és adatmodelleket.

Az alábbiakban részletesebben ismertetem az egyes végpontokat és paramétereiket:

GET /api/ImgGen:

Ez a végpont képet generál a megadott szöveg alapján, és visszaadja a generált kép URL-t.

A végpont a következő paramétereket várja:

- text: A generált kép szövege.
- code: Az API autentikációs kódja. A végpont hitelesítéséhez kötelezően megadandó.

GET /api/UserAPI:

Ez a végpont az email bemeneti paraméter alapján visszaadja a felhasználóhoz társított adatokat az adatbázisból. A felhasználó azonosításához a végpont az alábbi paramétereket várja:

- email: Az email cím, amely alapján lekérdezzük a felhasználó adatait. Ez a paraméter kötelezően megadandó.
- code: Az API autentikációs kódja. Ez a paraméter szintén kötelezően megadandó a végpont hitelesítéséhez.

PATCH /api/UserAPI:

Ez a végpont lehetővé teszi a felhasználó adatainak frissítését az adatbázisban. A végpont a következő paramétereket várja:

- email: A felhasználó email címe.
- billing_name: A számlázási név.
- billing_postalcode: A számlázási irányítószám.
- billing_town: A számlázási város.
- billing_taxnumber: Az adószám.
- billing_address: A számlázási cím.
- accessToken: A Facebook-tól megkapott Short-lived access token, lásd ábra 3.
- code: Az API autentikációs kódja. A végpont hitelesítéséhez kötelezően megadandó.

GET /api/PostGen:

Ez a végpont szöveget generál a megadott paraméterek alapján, és visszaadja a generált szöveget és a hozzá tartozó kép URL-t közösségi média posztokhoz. A végpont a következő paramétereket várja:

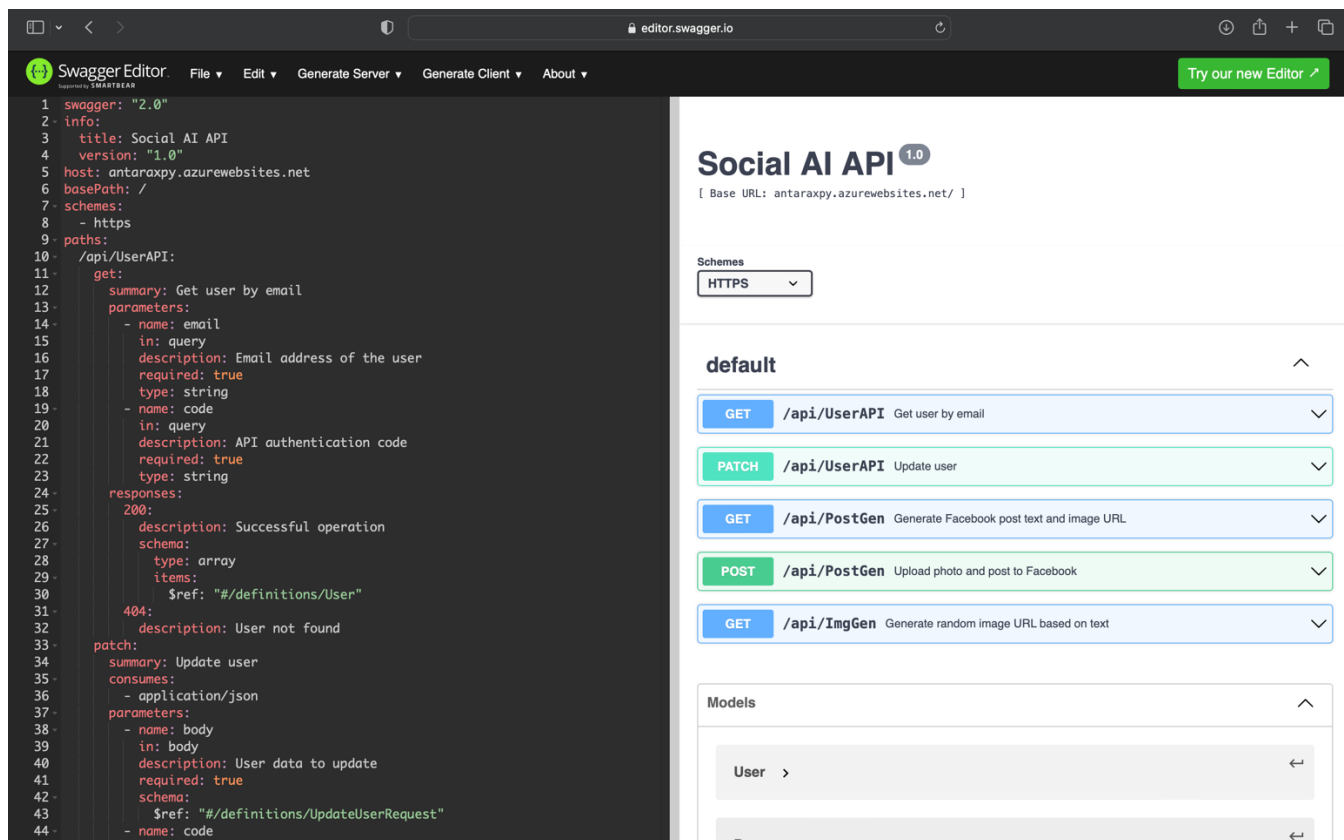
- company: A vállalat neve.
- website: A vállalat weboldala.
- activity: A vállalat tevékenysége.
- about: A vállalat leírása.
- language: A poszt nyelve.
- code: Az API autentikációs kódja. A végpont hitelesítéséhez kötelezően megadandó.

POST /api/PostGen:

Ez a végpont lehetővé teszi a generált szöveg és kép feltöltését közösségi média platformokra. A végpont a következő paramétereket várja:

- body: Az adatokat tartalmazó JSON objektum, amelyben megadjuk a feltöltendő szöveget, az autentikációs token-t, a közösségi média oldal azonosítóját és a kép URL-t. A kötelezően megadandó mezők:
 - textpost: A generált szöveg.
 - accesstoken: Az autentikációs token.
 - pageid: A közösségi média oldal azonosítója.
 - url: A feltöltendő kép URL-je.
 - code: Az API autentikációs kódja. A végpont hitelesítéséhez kötelezően megadandó.

Részletesebb dokumentációt a szakdolgozat melléklete tartalmaz (1. melléklet).



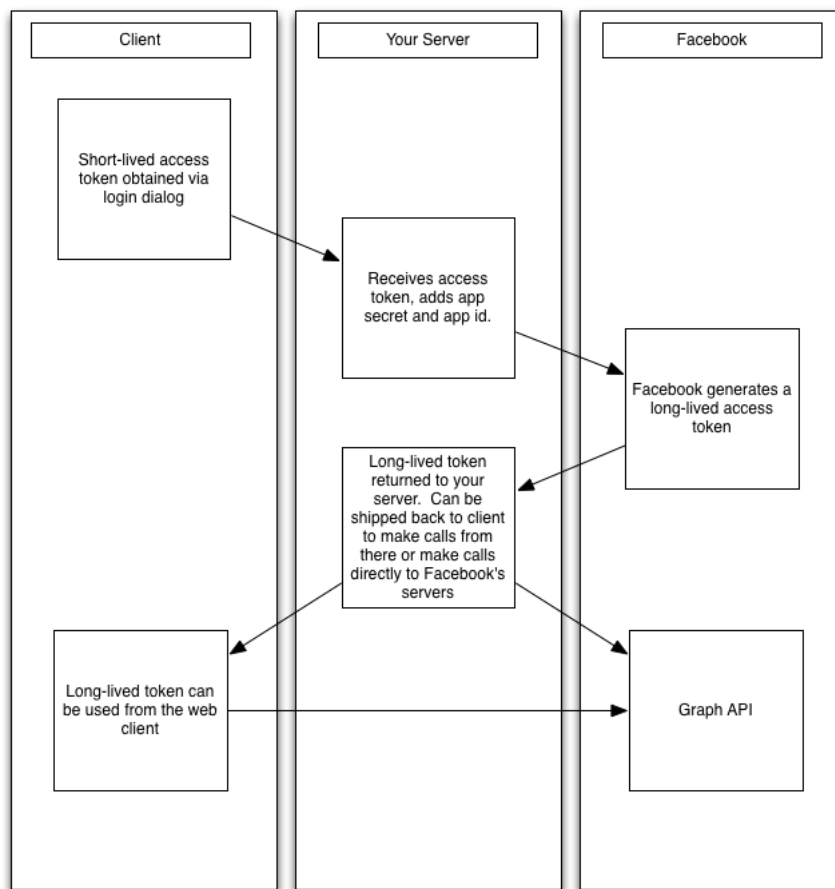
2. ábra: Az alkalmazás API dokumentációja

Forrás: <https://editor.swagger.io>

1.6. Facebook Access Token

Az alábbi dokumentáció részletesen bemutatja az Access Token kérése, megújítása és felhasználása lépéseit, valamint az engedélyek kezelését. Ezek az információk segítettek nekem megérteni és megvalósítani a Facebook integrálásával kapcsolatos funkciókat:

<https://docs.squiz.net/funnelback/docs/latest/build/data-sources/facebook/facebook-page-access-token.html>



3. ábra: Facebook Access Token lekérdezésének folyamata

Forrás: <https://developers.facebook.com/docs/facebook-login/guides/access-tokens/get-long-lived>

1.7. Adatszerkezet

Az Azure Cosmos DB egy NoSQL adatbázis, melynek szerkezete JSON formátumú dokumentumokból épül fel. A megtervezett felhasználói objektum a szakdolgozat mellékletét képezi User.json fájl néven. (lásd 2. melléklet)

2. A SZOFTVER KIALAKÍTÁSA

2.1. Lokális fejlesztői környezet kialakítása

A lokális fejlesztői környezet kialakításának menete Windows 10 operációs rendszer esetében értelmezendő.

Telepíteni kell az operációs rendszerre az alábbi MSI telepítőt, ez a telepítő az Azure Functions Core Tools-t tartalmazza annak érdekében, hogy az Azure Function lokális fejlesztői környezeten képes legyen elindulni:

<https://go.microsoft.com/fwlink/?linkid=2174087>

Telepíteni kell az operációs rendszerre az alábbi oldalon letölthető Python 3.10 verzióját, mely szintén az Azure Function elindításának függősége:

<https://www.python.org/downloads/release/python-3100/>

Telepíteni kell az alábbi oldalon letölthető Node.js v20.2.0 verzióját annak érdekében, hogy a lokális fejlesztői környezeten el tudjuk indítani a React Frontend alkalmazást:

<https://nodejs.org/en>

Le kell klónozni, vagy letölteni az alábbi privát Azure DevOps repo-ból a React Frontend alkalmazás forráskódját:

https://dev.azure.com/antarax/_git/OpenAI%20React

A React Frontend alkalmazás elindításához az alábbi 3 parancsot kell futtatni:

1. npm install
2. npm run build
3. npm run start

Le kell klónozni, vagy letölteni az alábbi privát Github repo-ból a Python Azure Function forráskódját:

<https://github.com/Anttarax/AzureFunction>

A Python Azure Function elindításához az alábbi 2 parancsot kell futtatni:

1. pip install -r requirements.txt
2. func start

Forrás:

<https://learn.microsoft.com/en-us/azure/azure-functions/functions-run-local?tabs=macos%2Cisolated-process%2Cnode-v4%2Cpython-v2%2Chttp-trigger%2Ccontainer-apps&pivots=programming-language-python>

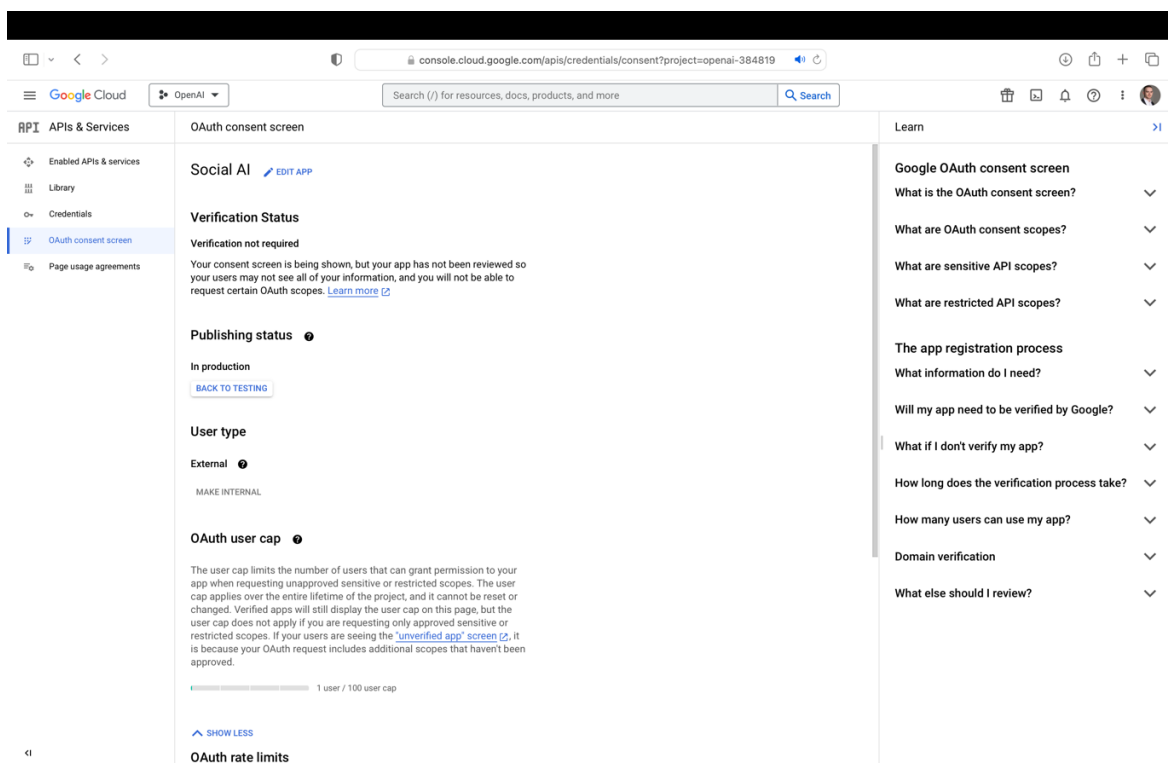
<https://create-react-app.dev/docs/getting-started/>

2.2. Google Cloud beállítása

Annak érdekében, hogy Google autentikációt lehessen implementálni és ezzel a felhasználók be tudjanak jelentkezni a Google fiókjukkal az alkalmazásba, létre kellett hozni a Google Cloud Platformon egy OAuth consent screen-t, majd ehhez létre kellett hozni egy OAuth 2.0 Client ID-t. A Google Cloud legenerálta a Client ID-t, és az ahhoz tartozó Client secret-et a létrehozás során. Ezeket az implementálás során a Frontend React alkalmazás forráskódjában helyeztem el.

Az oldal, ahova implementálásra került a Google autentikáció: <https://socialai.progeurope.hu/login>

Az OAuth consent screen beállításában a Publishing státuszt production státuszra kellett beállítani, mivel ez teszi lehetővé, hogy bárki (nem csak a fejlesztők és a tesztelők) képesek legyenek bejelentkezni. Mivel csak a leggyakoribb, alap információkat (publikus név, publikus email) kérem el a bejelentkezésnél a felhasználóktól, így verifikációs folyamatot nem kellett elvégezni az alkalmazáshoz, mivel a Google ilyen esetben ezt nem követeli meg.



4. ábra: Google Cloud OAuth consent screen beállítások

Forrás: <https://console.cloud.google.com/apis/credentials/consent?project=openai-384819>

Ezt követőleg elkezdhettem létrehozni az előbb létrehozott OAuth consent screen-hez tartozó Credentials-t. A létrehozásnál ki kellett választani az alkalmazás típusát, ami jelen esetben Web application típus.

Mivel Frontend oldali implementálás történt, így a különböző környezetek URL-jét fel kellett venni az alábbi engedélyezési listára:

Authorized JavaScript origins:

- ☐ URIs 1: <https://socialai.progeurope.hu> (Éles környezet)
- ☐ URIs 2: <http://localhost> (Lokális fejlesztői környezet)
- ☐ URIs 2: <http://localhost:3000> (Lokális fejlesztői környezet)

The screenshot shows the Google Cloud Console interface for configuring an OAuth 2.0 Client ID for a web application. The left sidebar shows the navigation menu with 'Credentials' selected. The main content area is titled 'Client ID for Web application' and includes a 'DELETE' button. The 'Name' field is set to 'Social AI'. The 'Client ID' is 50655227462-up5uiminhpan49cv2vfj615puriu372j.apps.googleusercontent.com. The 'Creation date' is April 25, 2023 at 11:13:12 PM GMT+2. The 'Client secrets' section shows a secret and its creation date. The 'Authorized JavaScript origins' section shows three entries: 'https://socialai.progeurope.hu', 'http://localhost', and 'http://localhost:3000'. The 'Authorized redirect URIs' section is empty. The 'SAVE' button is visible at the bottom.

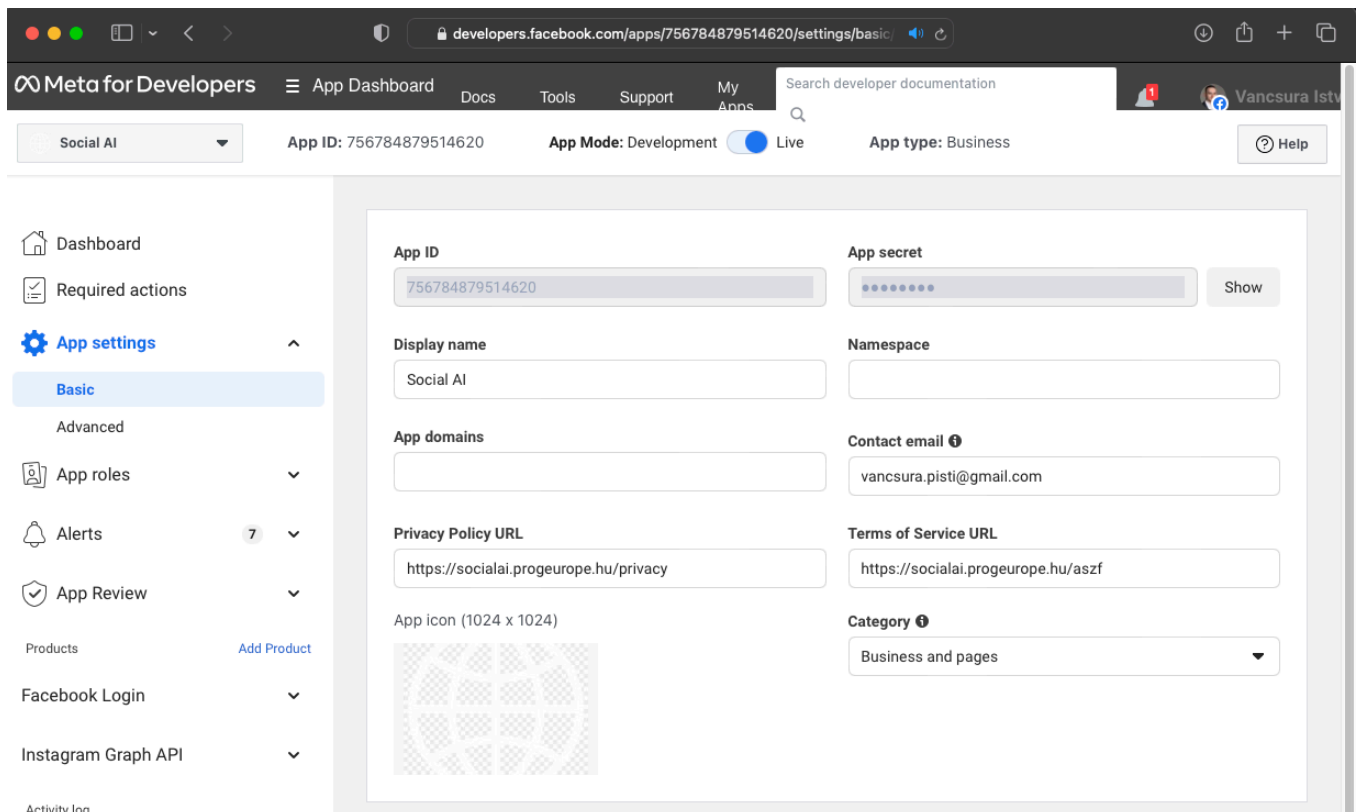
5. ábra: Google Cloud OAuth 2.0 Client ID beállítások

Forrás: <https://console.cloud.google.com/apis/credentials/oauthclient/50655227462-up5uiminhpan49cv2vfj615puriu372j.apps.googleusercontent.com?hl=hu&project=openai-384819>

2.3. Meta for Developers

Annak érdekében, hogy megvalósítsuk a funkciót, mely lehetővé teszi az alkalmazás számára, hogy a felhasználó nevében tartalmat készítsen és tegyen közzé a Facebook-on, a korábban említett Facebook SDK-t implementáltam a Frontend React alkalmazásba. Ahhoz, hogy ez az implementálás megfelelően működjön, a Google autentikációhoz hasonlóan itt is szükség van egy Client (App) ID-ra, és a hozzá tartozó Secret-re, melyeket szintén a forráskódban helyeztem el. A létrehozás során a Business típusú Appot választottam, és a továbbiakban részletezett Facebook Login beállításait végeztem el.

- ☐ Privacy Policy URL: <https://socialai.progeurope.hu/privacy>
- ☐ Terms of Service URL: <https://socialai.progeurope.hu/aszf>
- ☐ Website Site URL: <https://socialai.progeurope.hu/>
- ☐ Contact email és Data Protection Officer contact information: A saját személyes adataimmal töltöttem ki.



6. ábra: Meta for Developers APP beállítások

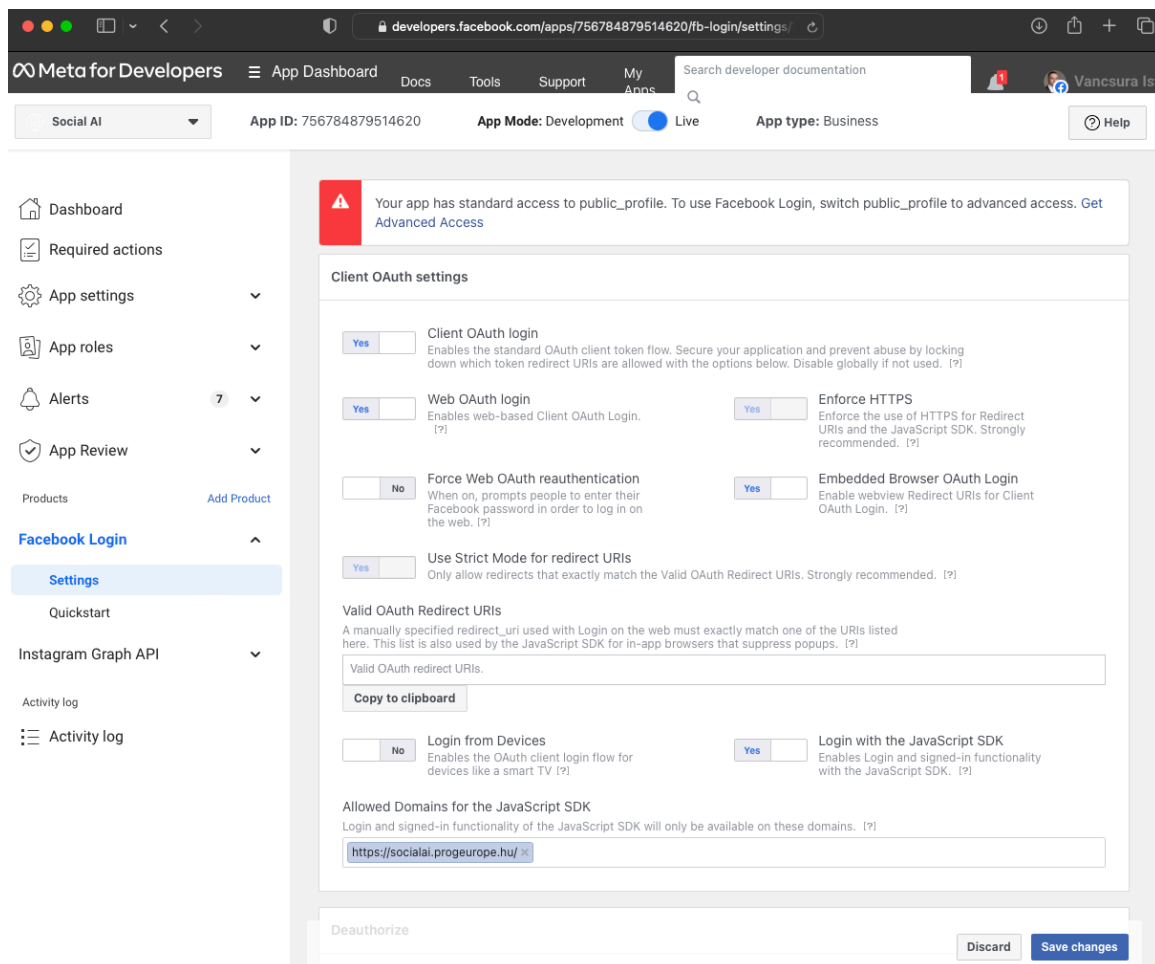
Forrás: https://developers.facebook.com/apps/756784879514620/settings/basic/?business_id=758453229336454

Allowed Domains for the JavaScript SDK:

- <https://socialai.progeurope.hu/> (Éles környezet)

A Google autentikáció beállításával ellentétben itt nem kell, és nem is lehet felvenni a lokális fejlesztői környezetet az engedélyezési listára, mivel kötelező HTTPS (titkosított) URL-eket megadni, azonban, ha az App Mode-ot Development-re állítjuk, működik, viszont ilyenkor az éles környezet nem jó. Potenciális megoldás erre a jövőben 2 létezik:

1. Felvesszük a lokális fejlesztői környezet URL-t https-ként, és kiállítunk lokálisan egy tanúsítványt, majd implementáljuk azt a lokális fejlesztői környezetbe.
2. Létrehozunk a fejlesztői környezet számára egy külön Meta for Developers APP-ot, ami Development mode-ban marad, míg az eredeti pedig Live módban szolgálja ki az éles környezetet.



7. ábra: Meta for Developers APP beállítások

Forrás: https://developers.facebook.com/apps/756784879514620/fb-login/settings/?business_id=758453229336454

2.4. ChatGPT és Pexels.com implementáció

Implementáltam az OpenAI és a Pexels.com API-t az alkalmazásba. A pexels.com olyan képek platformja, melyek jogilag szabadon felhasználhatóak. A platform biztosít egy nyílt adatbázist, hozzá keresőmotort a képek kereséséhez és lekérdezéséhez, az OpenAI pedig egy nyelvi modellt biztosít, melynek a neve ChatGPT, ami lehetővé teszi a szövegek generálását.

Ezek implementációk összessége teszi lehetővé a felhasználók számára, hogy posztokat generáljanak az alkalmazásban, majd ezt az előbbi fejezetben leírt Facebook integrációnak köszönhetően közvetlenül az alkalmazásból publikálják saját oldalukra.

Az alkalmazás alábbi két API-jába került implementálásra az OpenAI és Pexels.com API:

1. PostGen

Az alábbiakban részletesen magyarázom, milyen logika alapján működik:

Ha a HTTP kérés metódusa GET, akkor:

1. A kapott adatok alapján (Facebook integrációból származó adatok a felhasználó inputjaival kiegészítve, mely részletesebben megtekinthető „Az alkalmazás API dokumentációja” fejezetben) az OpenAI API-n keresztül generál egy közösségi média poszt szöveget, valamint a szöveghez tartozó kulcsszavakat a Pexels.com képek kereséséhez.
2. Ezután véletlenszerűen kiválaszt egy képet a Pexels.com API-n keresztül a generált kulcsszavak alapján. (mely kulcsszavak célzottan az előző lépésben generált szöveghez tartoznak, csupán a célzott keresés után választ véletlenszerűen, a találatok közül).
3. A választott kép URL-jét és a generált szöveget JSON formátumban visszaküldi a Frontend alkalmazásnak
4. Az e-mail cím alapján lekérdezi az adatbázisból a felhasználót és növeli az egyenlegét. (melyre később számlázási rendszert lehet építeni, mely szerepel a továbbfejlesztési lehetőségek között)

Ha a HTTP kérés metódusa POST, akkor a beérkezett adatok, melyek tartalmazzák a végleges közösségi média poszt szövegét és a szöveghez tartozó kép URL-jét (teljes adatkör „Az alkalmazás API dokumentációja” fejezetben) publikálja a Facebook oldalra a posztot.

2. ImgGen

Azért volt szükség rá, hogy létrehozzak egy külön ImgGen API-t, mivel előfordulhat olyan eset, hogy a PostGen által generált szöveg elfogadja a felhasználó, viszont a képet nem. Valamint a generált szöveget lehetősége van szerkeszteni is a felhasználónak, ezért létre lett hozva egy „Kép újra generálása” funkció és ezzel együtt az ImgGen API végpont.

Az alábbiakban részletesen magyarázom, milyen logika alapján működik:

1. A kapott adatok alapján (melyek tartalmazzák a végleges közösségi média poszt szövegét, (teljes adatkör „Az alkalmazás API dokumentációja” fejezetben) az OpenAI API-n keresztül generál kapott a szöveghez tartozó kulcsszavakat a Pexels.com képek kereséséhez.
2. Ezután véletlenszerűen kiválaszt egy képet a Pexels.com API-n keresztül a generált kulcsszavak alapján. (mely kulcsszavak célzottan az előző lépésben generált szöveghez tartoznak, csupán a célzott keresés után választ véletlenszerűen, a találatok közül).
3. A választott kép URL-jét JSON formátumban visszaküldi a Frontend alkalmazásnak.

A „Poszt generálása” és a „Kép újra generálása” funkciók ismételhetők a végtelenségig a felhasználó döntése szerint, majd a folyamat végén a „Poszt véglegesítés és küldés” funkcióval zárul, mely az előzőekben részletezett PostGen API végpontjának POST metódus hívását jelenti.

Az OpenAI API kulcsot az alábbi oldalon hoztam létre, majd ezt a Backend rendszer forráskódjában helyeztem el: <https://platform.openai.com/account/api-keys>. Az alábbi oldalon lévő dokumentáció alapján végeztem el az implementálást: <https://platform.openai.com/docs/api-reference>

A Pexels.com API kulcsot az alábbi oldalon hoztam létre, majd ezt a Backend rendszer forráskódjában helyeztem el: <https://www.pexels.com/api/new/>. Az alábbi oldalon lévő dokumentáció alapján végeztem el az implementálást: <https://www.pexels.com/api/documentation/>

2.5. A ChatGPT támogató szerepe

Az alábbi példák mutatják a ChatGPT sokoldalúságát. A gépi intelligencia képességeinek felhasználása sokféle (az alábbi alfejezetek szerinti) módon segíthet a fejlesztőknek. Az alfejezetekről mellékleteket is csatolok a szakdolgozathoz, mely valós, megtörtént kommunikációkról tartalmaz logokat.

2.5.1. *Swagger dokumentáció generálás*

A ChatGPT képes programkódot elemezni. A REST API-k kódjainak értelmezésével felismerte, melyek a végpontok, milyen paramétereket várnak és milyen válaszokat adnak vissza. Ezután képes volt Swagger dokumentációt generálni YAML formátumban a felismert információkból, mely szintaktikailag is megfelelt a Swagger dokumentáció követelményeinek. Kapcsolódó 8. számú melléklet.

2.5.2. *Algoritmus szintaxis hibák javítása*

A ChatGPT képes az algoritmus hibák azonosítására. Amikor a kód hibás eredményeket produkált, a ChatGPT részletesen elmagyarázta, hogy hol lehet a hiba, például hiányzó zárójeleket, vagy helytelen változóneveket mutatva ki a kódban. Ezután javaslatokat tett a hibák kijavítására, például a helyes szintaxis bemutatásával. Kapcsolódó 9. számú melléklet.

2.5.3. *Programozási könyvtárakkal kapcsolatos tanácsadás.*

Amikor a projektnek szüksége volt egy táblázat-kezelő könyvtárra, a ChatGPT segített a megfelelő választásban. A ChatGPT felsorolta a lehetséges táblázat-kezelő könyvtárakat, és ismertette előnyeiket. Kapcsolódó 10. számú melléklet.

2.5.4. *Magyarázattal egybekötött fordítás*

Idegen nyelvű (pl.: angol) szakmai dokumentációk esetében a ChatGPT képes volt nem csak lefordítani az idegen nyelvű szöveget magyarra, de az értelmezésében és összefoglalásában is képes volt segítséget nyújtani, ezzel rövidebbé téve a fejlesztési időt. Kapcsolódó 11. számú melléklet.

2.6. Frontend fájlok és komponensek

- .env: Ez a fájl arra szolgál, hogy olyan környezeti változókat tároljon lokális fejlesztésnél, amelyeket nem írhatunk bele közvetlenül a frontendünk kódjába, például API kulcsok.
- .gitignore: Ebben a fájlban azon fájlok és mappák listája van összegyűjtve, amelyeket ki akarunk (vagy ki kell) hagyni a verziókezelés alól, például a node_modules mappa, amiben a frontend dependenciák forrásfájljai vannak telepítve lokálisan.
- index.html: Ezt a fájlt a public mappában tároljuk és fő feladata, hogy a react applikáció az ebben a fájlban található `<div id="root"></div>` html tagbe bele renderelje a felhasználónak megjeleníteni kívánt tartalmat. Ezenkívül fontos feladata, hogy a kódban alkalmazott `<script>` tagek segítségével csatlakozzunk a Facebook és Google API-okhoz, hogy ezeket a frontend programban használni tudjuk.
- index.js: Ez a komponens a React applikáció indító komponense, fő feladata, hogy betöltsen a többi komponenset. A komponensek által kirenderelt html (amit a felhasználónak a weboldalon megjelenítünk) az index.html fájlban levő `<div id="root"></div>` html tagen belül lesz megjelenítve.
- App.jsx: Ez a komponens felelős a React appban létrehozott Route-ok (útvonalak) meghatározásáért, továbbá az adott útvonalakhoz történő „page” komponensek társításáért.
- index.css: Ebben a fájlban vannak meghatározva azok a css osztályok, amelyeket a frontend applikációnk formázásához a többi komponensben használunk.
- Footer.jsx: Ebben a fájlban a minden aloldal alján megjelenített Footer react komponens van deklarálva, aminek fő funkciója, hogy megjeleníti a linkeket az Általános Szerződési Feltételek, Adatkezelési szabályzat, és a Kapcsolat oldalra.
- Navbar.jsx: Ez a komponens jeleníti meg a navigációs menüt minden aloldal tetején.
- Account.jsx: Ez a react komponens felelős a „/” útvonalon a tartalom megjelenítéséért. Ezen az aloldalon a Google fiókhoz társított fő adatokat (név, email-cím, profilkép) tekintheti meg a felhasználó.
- Aszf.jsx: Ez a komponens jeleníti meg az Általános Szerződési Feltételeket az „/aszf” útvonalon.

- Privacy.jsx: Ez a komponens jeleníti meg az Adatkezelési szabályzatot a „/privacy” útvonalon.
- Billing.jsx: Ez a komponens felelős a később potenciálisan implementálásra kerülő fizetési lehetőségek megjelenítéséért a „/billing” útvonalon.
- Facebook.jsx: Ez a komponens felelős a felhasználó Facebook fiókjába történő bejelentkezésért, továbbá a bejelentkezett felhasználók adatainak és elérhető Facebook oldalainak egy rendezett listában történő megjelenítéséért a „react-table” npm csomag segítségével.
- Login.jsx: Ez a komponens felelős a „/login” útvonalon a felhasználó Google fiókjának segítségével a felhasználó bejelentkeztetéséért. Erre a „/login” útvonalra fog átirányítani minket az applikáció bármelyik más oldala, amennyiben még nem vagyunk bejelentkezve.
- Post.jsx: Ez a komponens felelős a „/post” útvonalon a Facebook posztok generálásához szükséges form megjelenítéséért. Ez az applikáció talán legfontosabb komponense, mivel ez a komponens felelős a mesterséges intelligencia segítségével a konkrét Facebook posztok létrehozásáért. Ezen az aloldalon a felhasználó személyre szabott posztokat generálhat, módosíthatja ezeket, majd egy gomb megnyomásával valóban meg is tudja osztani ebből az appból a Facebook API segítségével.

A navigációs menü, valamint a footer részei és az általuk meghivatkozott útvonalak:

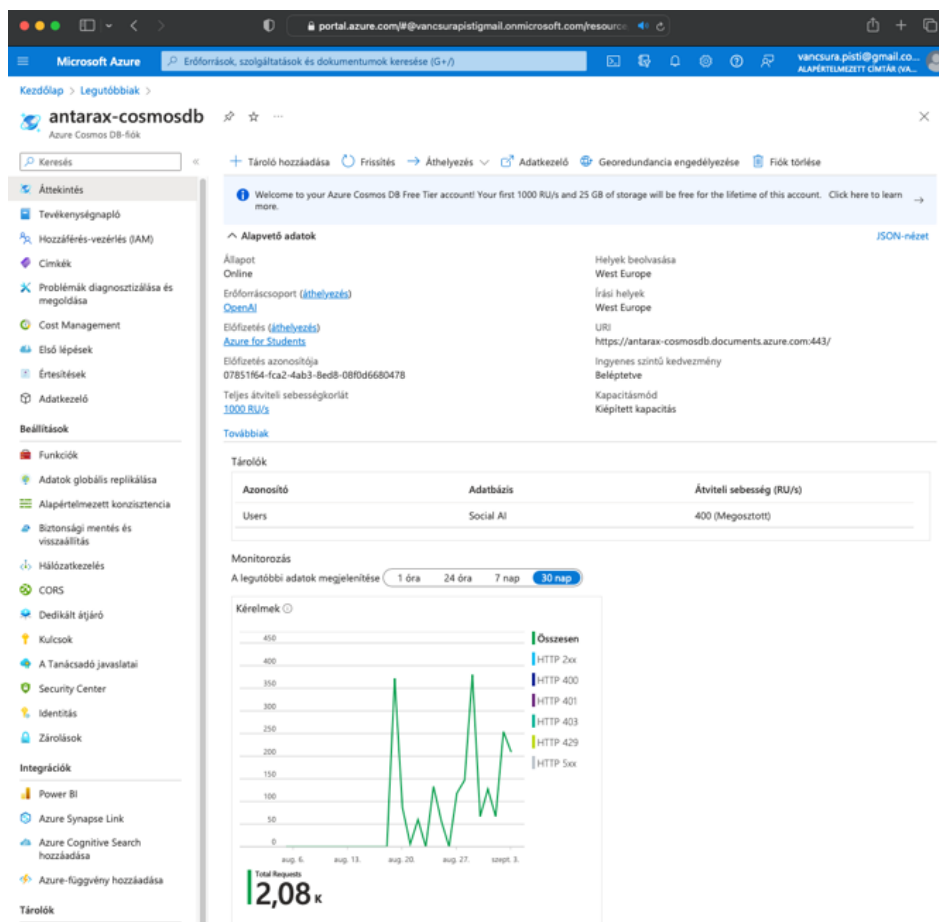
- Fiókom: „/”
- Számlázás: „/billing”
- AI poszt generálás „/post”
- Facebook törzsadatok: „/facebook”
- Adatkezelési szabályzat: „/privacy”
- Általános szerződési feltételek: „/aszf”
- Kapcsolat: „<https://progeurope.hu/hu/contact/with-map/>” (a cég kapcsolat oldala, mely egy ettől független projekt során készült el.)

3. ÉLES KÖRNYEZET ÜZEMBE HELYEZÉSE

3.1. Azure CosmosDB létrehozása és beállítása

Létrehozott Azure Cosmos DB-fiók paraméterei:

- ☐ Hely: West Europe
- ☐ URI: <https://antarax-cosmosdb.documents.azure.com:443/>
- ☐ Teljes átviteli sebességkorlát: 1000 RU/s
- ☐ CORS Engedélyezett forráshelyek: *



8. ábra: Azure Cosmos DB-fiók áttekintés

Forrás:

<https://portal.azure.com/#@vancsurapistigmail.onmicrosoft.com/resource/subscriptions/07851f64-fca2-4ab3-8ed8-08f0d6680478/resourceGroups/OpenAI/providers/Microsoft.DocumentDb/databaseAccounts/antarax-cosmosdb/overview>

Az Azure Cosmos DB-fiókban egy Social AI nevű adatbázis lett létrehozva az alábbi paraméterekkel:

- ❑ Maximális átvételi sebességhatár: 400 RU/s
- ❑ Skálázás: Automatikus

A létrehozott Social AI nevű adatbázisban létre lett hozva egy Users nevű Container (Tároló), mely tárolóba az alkalmazás a felhasználók és a felhasználókhoz tartozó adatokat képes tárolni JSON formátumban. Minden felhasználónak egy külön JSON dokumentum jön létre. A tároló partíció kulcsa a következő attribútum: /email

The screenshot shows the Azure Cosmos DB Data Explorer interface. The left sidebar displays the 'Social AI' database and the 'Users' container. The main pane shows a table of users with columns 'id' and '/email'. The right pane displays the JSON document for a user with 'id' 'b70c5b65-d5f...' and '/email' 'babok200035...'. The JSON document includes fields like 'email', '_rid', '_self', '_etag', '_attachments', 'billing_name', 'billing_postalcode', 'billing_town', 'billing_address', and 'pages'.

9. ábra: Azure Cosmos DB adatkezelő

Forrás:

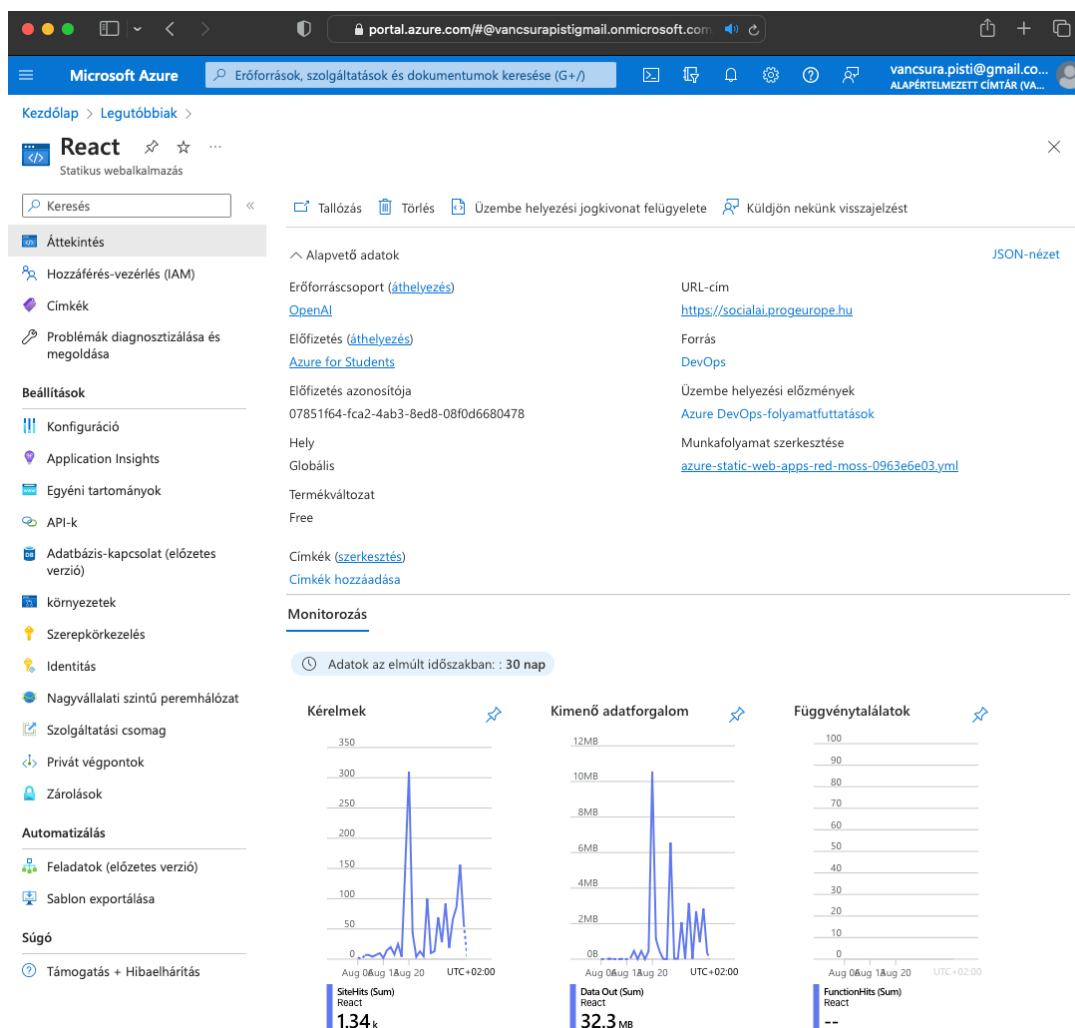
<https://portal.azure.com/#@vancsurapistigmail.onmicrosoft.com/resource/subscriptions/07851f64-fca2-4ab3-8ed8-08f0d6680478/resourceGroups/OpenAI/providers/Microsoft.DocumentDb/databaseAccounts/antarax-cosmosdb/dataExplorer>

3.2. Azure Static Web Apps létrehozása és beállítása

- ☐ Hely: West Europe
- ☐ URL-cím: <https://socialai.progeurope.hu>

Egyéni tartományok:

- ☐ red-moss-0963e6e03.3.azurestaticapps.net (Automatikusan létrehozott)
- ☐ www.socialai.progeurope.hu
- ☐ socialai.progeurope.hu (Alapértelmezett)



10. ábra: Azure Static Web Apps áttekintés

Forrás:

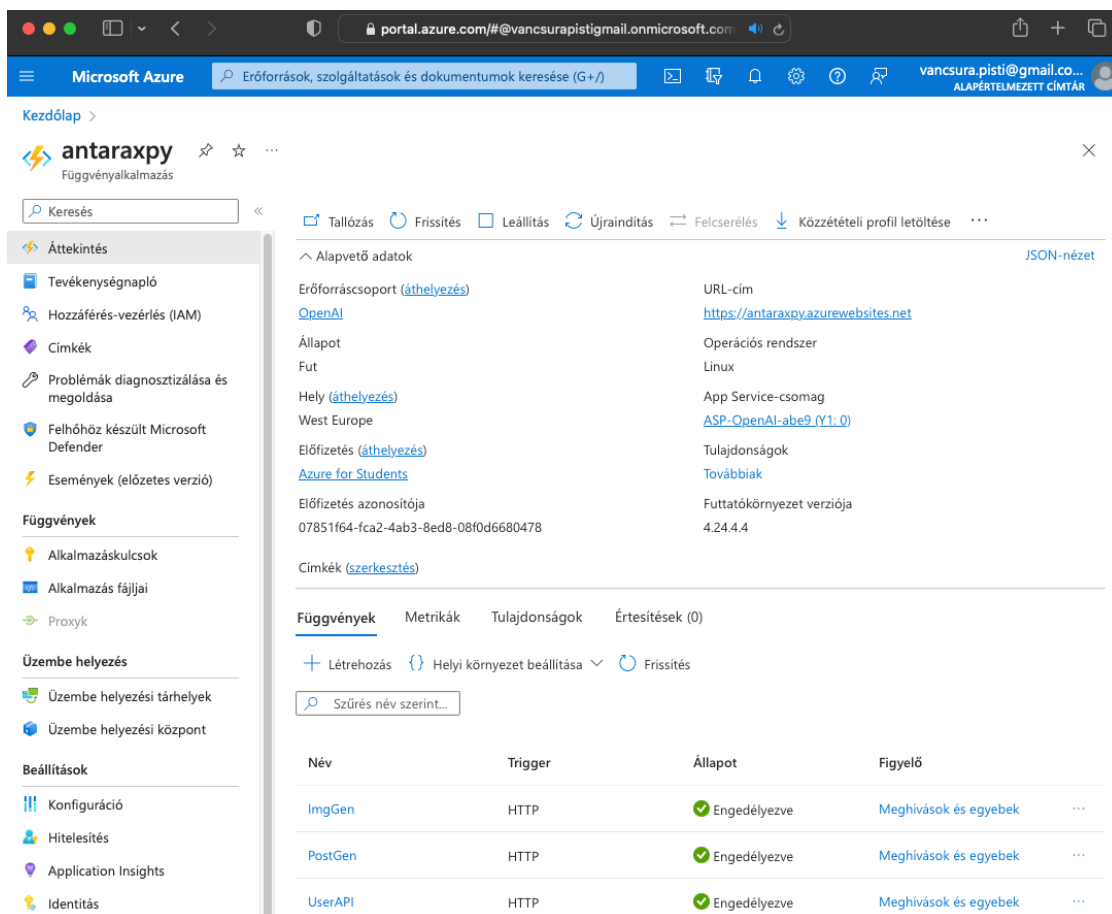
<https://portal.azure.com/#@vancsurapistigmail.onmicrosoft.com/resource/subscriptions/07851f64-fca2-4ab3-8ed8-08f0d6680478/resourceGroups/OpenAI/providers/Microsoft.Web/staticSites/React/staticsite>

3.3. Azure Functions létrehozása és beállítása

- ☐ Hely: West Europe
- ☐ Operációs rendszer: Linux
- ☐ URL-cím: <https://antaraxpy.azurewebsites.net>
- ☐ CORS Engedélyezett eredetek: *

Létrehozott függvények az API terveknek és a Swagger dokumentációnak megfelelően:

- ☐ <https://antaraxpy.azurewebsites.net/api/ImgGen>
- ☐ <https://antaraxpy.azurewebsites.net/api/PostGen>
- ☐ <https://antaraxpy.azurewebsites.net/api/UserAPI>



11. ábra: Azure Functions áttekintés

Forrás:

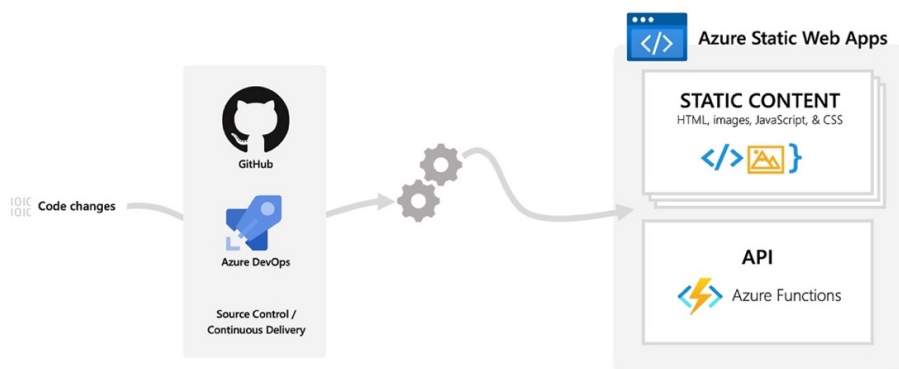
<https://portal.azure.com/#@vancsurapistigmail.onmicrosoft.com/resource/subscriptions/07851f64-fca2-4ab3-8ed8-08f0d6680478/resourceGroups/OpenAI/providers/Microsoft.Web/sites/antaraxpy/appServices>

3.4. GitHub Actions és Azure DevOps CI/CD folyamatok dokumentációja

Ebben az alfejezetben dokumentálom az alkalmazás komponenseinek CI/CD .YAML definíciós fájljait, a teljes .YAML fájlakat pedig a Forráskódok.zip melléklete tartalmazza a szakdolgozatnak.

A definíciós fájlok megtalálhatóak az alábbi eléréseken:

- ☐ Social AI React Frontend/azure-static-web-apps-red-moss-0963e6e03.yml
- ☐ Social AI Functions Python Backend/.github/workflows/main_antaraxpy.yml



12. ábra: DevOps Architektúra ábra

Forrás: <https://azure.microsoft.com/en-us/products/app-service/static>

Minden komponensre vonatkozó CI/CD pipeline paraméterek:

- ☐ Trigger: Main branch
Tehát ez azt jelenti, hogy amikor a fejlesztő a main branch-be felpusholja az adott komponens forráskód változásait, elindul automatikusan a CI/CD pipeline, tehát az automatizált folyamat, mely elkészíti (szakszóval le buildeli) a program új verzióját, majd azt a cél környezetbe automatikusan telepíti az új verziót.
- ☐ Agent: Microsoft Hosted Agent
Tehát ez azt jelenti, hogy az automatizált CI/CD folyamat egy Microsoft által futtatott és üzemeltetett szerveren történik meg. A másik opció az lett volna, ha Self-Hosted Agent-et használok, tehát én biztosítom a szerver környezetet az automatizációk futtatásához, ez azonban költséges lett volna és nem volt indokolt.
- ☐ vmImage: ubuntu-latest
Tehát az automatizált folyamat egy Linux operációs rendszeren fut.

Frontend React alkalmazás CI folyamat lépések:

1. task: NodeTool
 - a. versionspec: '20.x'

Magyarázat: Telepíti a Node.js-t a 20.x verzióval, mivel ez alapvető követelménye egy React típusú alkalmazásnak.
2. script: npm install
Magyarázat: A projekt függőségeinek telepítése, mely a következő lépés függősége.
3. script: npm run build
Magyarázat: A projekt forráskódjának build folyamatát végzi el.
4. task: CopyPublishBuildArtifact
 - a. CopyRoot: 'build'
 - b. Contents: '**'

Magyarázat: A build folyamat után a létrehozott fájlok (build mappában található) másolása és közzététele a CI/CD által használt artifacts-ba.

Frontend React alkalmazás CD folyamat (Deploy):

5. task: Deploy Azure Static Web App
 - a. Working directory: \$(System.DefaultWorkingDirectory)
 - b. App location: _React CI CD/build

Magyarázat: A React alkalmazás buildjét tartalmazó mappából telepíti az alkalmazást.

Azure Functions Python Backend rendszer CI folyamat lépések:

1. Setup Python version
Magyarázat: Beállítja a Python verziót a megadott PYTHON_VERSION-ra, majd létrehoz egy virtuális Python környezetet és aktiválja azt.
2. Install dependencies: pip install -r requirements.txt
Magyarázat: Telepíti a projekt függőségeit a requirements.txt alapján.

Azure Functions Backend rendszer CD folyamat (Deploy):

3. Letölti a build munkafolyamat által létrehozott artifactot.
4. Megtörténik az Azure Functions telepítése.

3.5. Domain vásárlása és beállítása, Azure DNS-zóna

Megvásároltam a progeurope.hu domaint a Rackhost Zrt. (<https://www.rackhost.hu/>) Domain regisztrátornál, melynek vezérlését ezután az Azure DNS-zóna szolgáltatásra delegáltam a névszerverek beállításával, melyeket az Azure portál adott meg számomra.

- ☐ Névszerver#1: ns1-32.azure-dns.com
- ☐ Névszerver#2: ns2-32.azure-dns.net
- ☐ Névszerver#3: ns3-32.azure-dns.org
- ☐ Névszerver#4: ns4-32.azure-dns.info

The screenshot shows the Rackhost domain management interface for the domain 'progeurope.hu'. The top navigation bar includes links for 'Domain', 'Webtárhely', 'WordPress Tárhely', 'cPanel tárhely', 'Üzleti email', 'VPS', and 'Tudásbázis'. The main content area is divided into several sections:

- PROFIL**: A sidebar menu with options like 'Felhasználói adatok', 'Számlázás és kreditek', 'Tennivalók', 'Hibajegyek', and 'Tevékenység napló'.
- Domainek » progeurope.hu**: The main section showing domain details.
 - Státusz**: Regisztrált
 - Előfizetés**: Aktív
 - Lejárat dátuma**: 2025-07-12
 - Névszerverek**: ns1-32.azure-dns.com, ns2-32.azure-dns.net, ns3-32.azure-dns.org, ns4-32.azure-dns.info. A green button 'Névszerverek cseréje' is visible.
- Tulajdonos**: A table with contact information for 'Prog Europe Kft.', including 'Vancsura István' as the contact person, email 'vancsura.pisti@gmail.com', and phone number '+36202769700'.
- Adminisztratív kapcsolattartó**: A table with administrative contact information for 'Prog Europe Kft.', including 'Vancsura István' as the contact person, email 'vancsura.pisti@gmail.com', and phone number '+36202769700'.
- Előfizetés lemondása**: A section with a warning about domain expiration and a red 'Lemondás' button.

13. ábra: Domain vezérlő felület, Rackhost

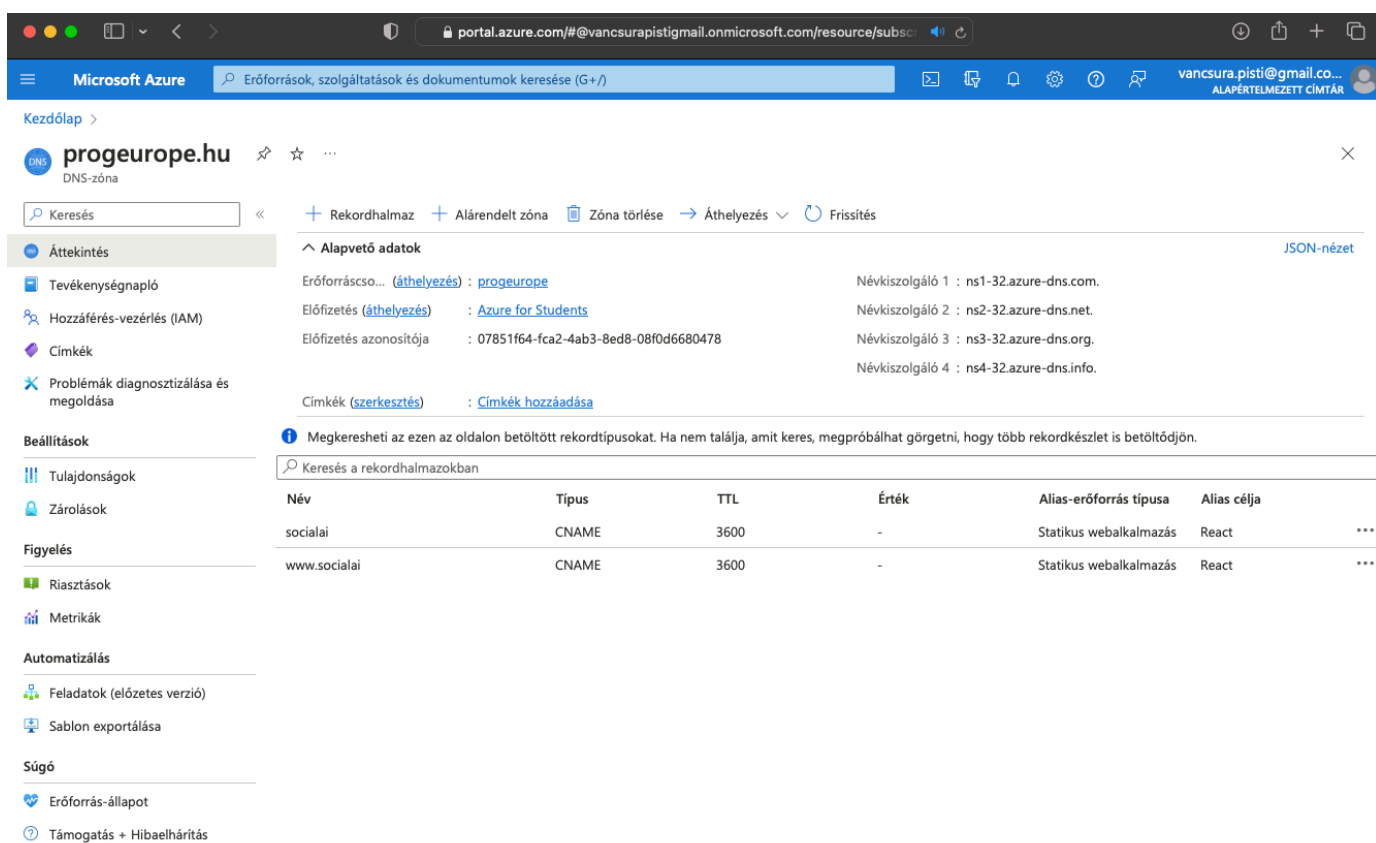
Forrás: <https://www.rackhost.hu/dashboard/domain/127377>

3.6. Azure DNS-zóna létrehozása és beállítása

Az alkalmazás számára egy socialai nevű aldomaint hoztam létre az alábbi (lásd 4. táblázat és 14. ábra) DNS rekordok felvételével:

4. táblázat: Azure DNS-zóna, beállított DNS rekordok

Név	Típus	TTL	Érték	Alias-erőforrás típusa	Alias célja
socialai	CNAME	3600	-	Statikus webalkalmazás	React
www.socialai	CNAME	3600	-	Statikus webalkalmazás	React



14. ábra: Azure DNS-zóna beállítások

Forrás:

<https://portal.azure.com/#@vancsurapistigmail.onmicrosoft.com/resource/subscriptions/07851f64-fca2-4ab3-8ed8-08f0d6680478/resourceGroups/progeurope/providers/Microsoft.Network/dnsZones/progeurope.hu/overview>

4. AZ IT BIZTONSÁG KIALAKÍTÁSA

Mivel az IT biztonság specializációt választottam tanulmányaim során, így figyelmet szenteltem arra, hogy egy biztonságos környezetbe legyen üzembe helyezve a lefejlesztett alkalmazás, éppen azért a Microsoft Azure felhő infrastruktúráját választottam, mely „világszerte több mint 100 rendkívül biztonságos létesítményre terjed ki.” (Microsoft, 2023.03.25.)

<https://learn.microsoft.com/hu-hu/azure/security/fundamentals/physical-security>

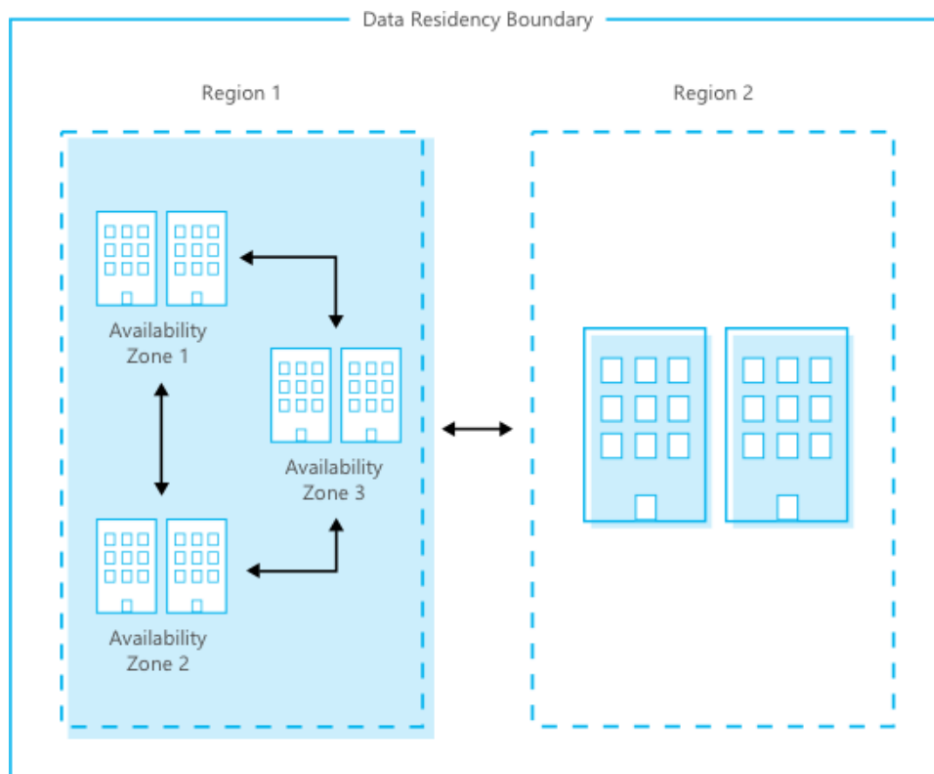
Letöltve 2023.10.07.

„Ezek a földrajzilag szétszórta adatközpontok megfelelnek az olyan kulcsfontosságú iparági szabványoknak, mint az ISO/IEC 27001:2013 és az NIST SP 800-53, a biztonság és a megbízhatóság érdekében.” (Microsoft, 2023.06.01.)

<https://learn.microsoft.com/hu-hu/azure/security/fundamentals/infrastructure>

Letöltve 2023.10.07.

Az alábbi ábra szemlélteti a magas rendelkezésre állás érdekében kialakított felépítést:






15. ábra: Adatközpont-infrastruktúra

Forrás: <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/physical-security>

4.1. Üzemeltetési felelősségek felosztása

Az alkalmazás minden komponensét PaaS szolgáltatásként veszem igénybe, ezáltal a következő (16. ábra) szerinti felelősségfelosztás van köztem és a Microsoft, mint szolgáltató között: (Az ábrán látható „Customer” engem reprezentál)

	Responsibility	SaaS	PaaS	IaaS	On-prem
Responsibility always retained by the customer	Information and data	Customer	Customer	Customer	Customer
	Devices (Mobile and PCs)	Customer	Customer	Customer	Customer
	Accounts and identities	Customer	Customer	Customer	Customer
Responsibility varies by type	Identity and directory infrastructure	Shared	Shared	Customer	Customer
	Applications	Shared	Shared	Customer	Customer
	Network controls	Shared	Shared	Customer	Customer
	Operating system	Shared	Shared	Customer	Customer
Responsibility transfers to cloud provider	Physical hosts	Microsoft	Microsoft	Microsoft	Customer
	Physical network	Microsoft	Microsoft	Microsoft	Customer
	Physical datacenter	Microsoft	Microsoft	Microsoft	Customer

 Microsoft  Customer  Shared

16. ábra: Üzemeltetési felelősségi felosztás

Forrás: <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/shared-responsibility>

A Microsoft felelős az alábbi területekért:

- ☐ Fizikai adatközpont
- ☐ Fizikai hálózat
- ☐ Fizikai kiszolgálók
- ☐ Operációs rendszer

Közös felelősség terhel engem és a Microsoftot az alábbi területekért:

- Hálózati vezérlések
- Alkalmazások
- Identitás és címtár infrastruktúra

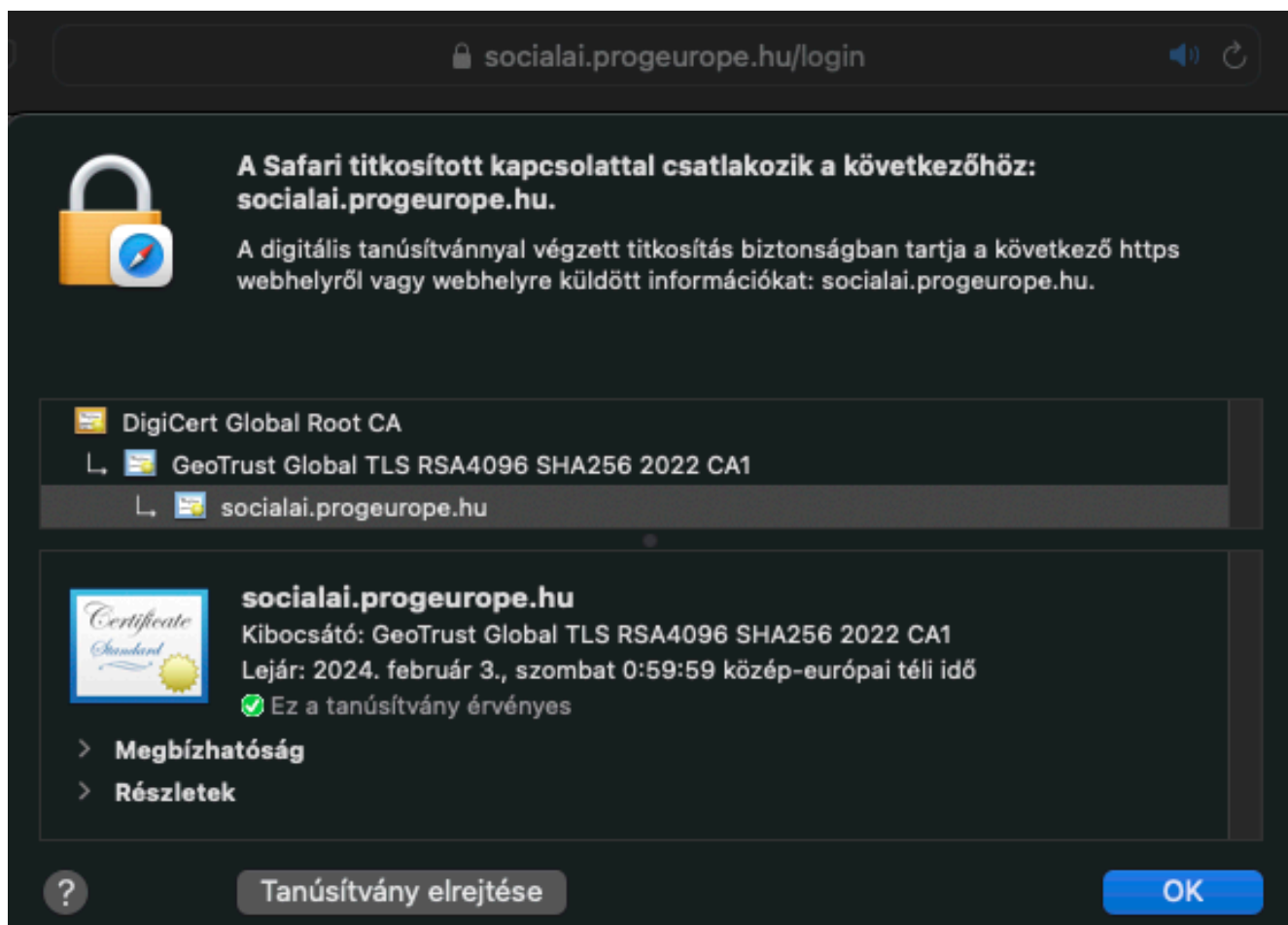
Én vagyok felelős az alábbi területekért:

- Információ és adat
- Eszközök (telefonok és számítógépek)
- Felhasználói fiókok és identitások

4.2. Felelősségi körömbé tartozó biztonsági intézkedéseim

4.2.1. Frontend védelem - HTTPS tanúsítvány

Az alkalmazás HTTPS tanúsítvánnyal lett ellátva annak érdekében, hogy megvédjem az adatokat a továbbítás során. Ez megakadályozza a közvetlen adatlopást és a közvetlen hozzáférést az adatokhoz.



17. ábra: HTTPS tanúsítvány

Forrás: <https://socialai.progeurope.hu/login>

4.2.2. Backend védelem – API Secret

A backend védelme kritikus fontosságú, mivel itt történik az adatok feldolgozása és kezelése, ezért API Secret lett beállítva, mely egy olyan kulcs, amit az alkalmazás használ az API-k hitelesítésére.

4.2.3. Monitorozás, naplózás és riasztás

A korábbi, éles környezet üzembe helyezése fejezetekben már látott ábrakon szereplő grafikonok prezentálják, hogy mind a Frontend, Backend és adatbázis monitorozva és naplózva van az alábbi szempontokból:

- ☐ Kérelmek száma és becsült költsége.
- ☐ Kimenő adatforgalom.
- ☐ Függvény végrehajtások száma (Sikeres, sikertelen, Http 4xx, Http 5xx).

Sikertelen, Http 4xx, Http 5xx hívásokra pedig riasztási szabályok beállítása is tervben van.

4.2.4. Azure portál - Kétlépcsős hitelesítés

Az Azure portálhoz kétlépcsős hitelesítést állítottam be. Ez azt jelenti, hogy az adminisztrátoroknak két különböző azonosítási lépést is teljesíteniük kell annak érdekében, hogy hozzáférjenek az infrastruktúra, alkalmazás és adatbázis rétegekhez.

4.2.5. Felhasználók bejelentkezési adatainak védelme

A korábban már részletezett Google autentikáció implementálása mögött nem csak kényelmi, hanem biztonsági megfontolások is voltak, hiszen ezzel a Google-től a felhasználói adatok biztonságát szolgáltatásként veszem igénybe, ezzel a felhasználói autentikációs adatok kezelésének felelőssége is a Google-t terheli, mivel az én alkalmazásom ilyen jellegű adatokat nem kezel, az adatbázisom pedig ilyen jellegű adatokat nem tárol.

4.2.6. Adatbázis védelem

Az adatbázisként használt Azure Cosmos DB tartalmazza a beépített biztonsági funkciókat, mint például az adatok titkosítását és hozzáférési vezérlést, az adatbázis API pedig titkos kulccsal van ellátva.

4.3. Jogi felelősség, megfelelés, GDPR

4.3.1. Adatvédelmi szabályzat

Elkészítettem és közzétettem egy Adatvédelmi szabályzatot, amely részletesen leírja, hogy milyen személyes adatokat gyűjtünk, hogyan kezeljük őket, és milyen jogokkal rendelkeznek a felhasználók az adataikkal kapcsolatban. Az Adatvédelmi szabályzat elérhető a következő linken: <https://socialai.progeurope.hu/privacy>

4.3.2. Általános Szerződési Feltételek

Létrehoztam és közzétettem Általános Szerződési Feltételeket (ÁSZF), amelyek rögzítik a felhasználók és az alkalmazás közötti szerződéses feltételeket és kötelezettségeket. Az ÁSZF elérhető a következő linken: <https://socialai.progeurope.hu/aszf>

4.3.3. GDPR megfelelés

Biztosítottam, hogy az alkalmazásom megfeleljen az Európai Unió Általános Adatvédelmi Rendeletének (GDPR). Az adatkezelési gyakorlatom és az adatvédelmi elveim összhangban vannak a GDPR előírásaival.

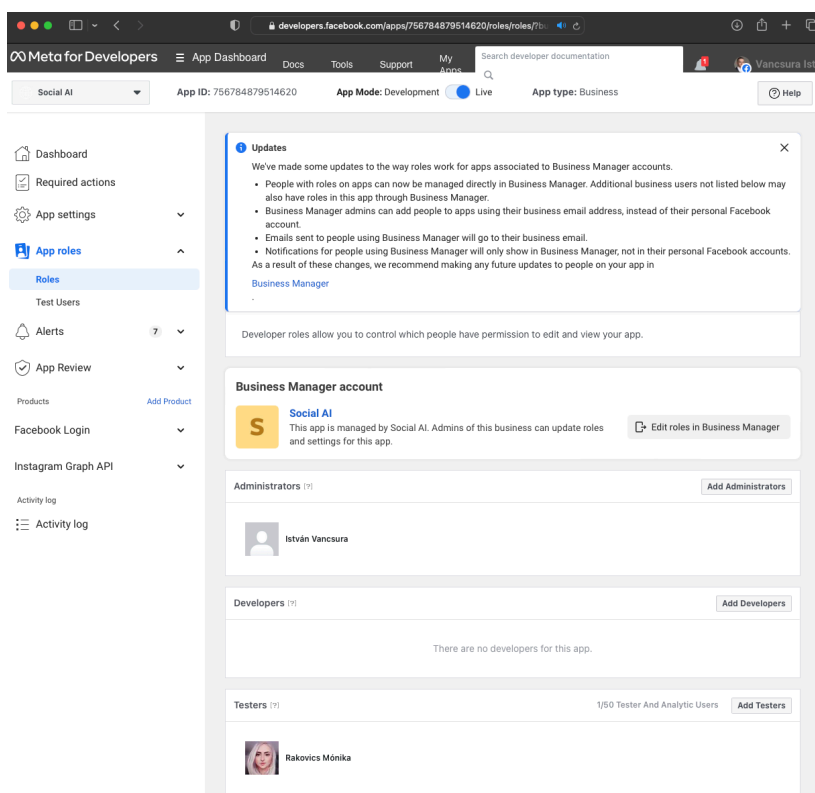
A GDPR alapján a felhasználóknak joguk van az adataik törléséhez, vagy azokhoz való hozzáféréséhez. Lehetővé teszem a felhasználók számára, hogy kérést tegyenek az adataik törlésére a Kapcsolat űrlapon keresztül, és biztosítom az adatokhoz való hozzáférést.

Az alkalmazásomban különféle technikai és szervezési intézkedéseket alkalmaztam az adatvédelem érdekében, például titkosítást, biztonsági mentéseket, melyeket az előző, „Felelősségi körömbé tartozó biztonsági intézkedéseim” fejezetben részleteztem.

5. TESZTELÉS

5.1. Új teszt felhasználó felvételének menete

1. Rendelkezzen a potenciális tesztelő minimum 1 darab saját Facebook oldallal, vagy ha nem rendelkezik hozzon létre egyet az alábbi oldal leírása alapján:
<https://hu-hu.facebook.com/business/help/104002523024878>
2. Regisztráljon az alábbi oldalon Facebook fiókjával annak érdekében, hogy rendelkezzen Facebook Developer fiókkal, mivel ez szükséges a következő lépés elvégzéséhez: <https://developers.facebook.com/async/registration>
3. Az alábbi oldalon fel kell venni a „Testers” listába a potenciális tesztelőt. Ezt követőleg kapni fog egy meghívást és értesítést a meghívásról, amit el kell fogadnia. A meghívás elfogadását követően az alábbi linken elérhető alkalmazás teljeskörűen használható lesz a tesztelő számára <https://socialai.progeurope.hu/>



18. ábra: Meta for Developers új tesztelő felvétele

Forrás:

https://developers.facebook.com/apps/756784879514620/roles/roles/?business_id=758453229336454

5.2. A teszt

Bevontam egy egyetemi társamat a tesztelésbe, Rakovics Mónika személyében, akinek ezúton is köszönöm a közreműködését.

Az alkalmazást rendszer szinten teszteltem, ezen kívül külön teszteltem a backend API végpontjait postman szoftver segítségével. A rendszerszintű tesztelés alatt tudatosan végig próbáltam az applikáció funkcionalitásait (Google autentikáció, Facebook törzsadatok lekérdezése, AI poszt generálás, posztok személyre szabhatósága, posztok utólagos szerkeszthetősége, posztok megosztása az appon keresztül a Facebookra a Facebook API segítségével). Mindezeket manuális teszteléssel végeztem, automatizált tesztek egyelőre nem írtam, de a későbbiekben ez is potenciális fejlesztési lehetőség lehet. A backend API végpontjainak unit- és automatizált tesztelése különböző kimenetek esetén szintén egy potenciális fejlesztési lehetőség a jövőre nézve.

A szakdolgozat mellékletét képezi egy videó (lásd 12. melléklet), amely szemlélteti az alkalmazás éles környezetben lévő, beüzemelt változatát, a funkciók működését és a tesztelést, valamint tartalmaz egy élő céges döntési helyzetet szimulált prezentációt.

ÖSSZEGZÉS

Eredmények értékelése:

Sikeresnek értékelem az alábbiakat:

- ☐ Sikerült a teljes fejlesztés tartalmát (minden komponenst, tehát a Frontend alkalmazást, Backend rendszert, adatbázist) éles (lásd 3. fejezet) és biztonságos (lásd 4. fejezet) környezetbe üzembe helyezni Cloud technológiák alkalmazásával.
- ☐ Az üzembe helyezés költségeit sikerült közel a nullán tartani (lásd 3., 4., 5., 6. mellékletek) mindamellet, hogy az alkalmazás képes a skálázásra.
- ☐ DevOps CI/CD folyamatok teljeskörűen megvalósultak, így ennek köszönhetően a további fejlesztések során elkészült forráskód feltöltése után automatizáltan megtörténik az alkalmazás új verziójának telepítése. (lásd 3.4 fejezet)

Részlegesen sikeresnek értékelem az alábbiakat:

- ☐ Szöveg generálási és kép kiválasztási algoritmus létrehozása. (lásd 2.4 fejezet)
- ☐ Facebook integráció, mivel jelenleg csak teszt felhasználók képesek az alkalmazást használni, amíg a Facebook Business verifikáció és App Review folyamat meg nem történik.

További fejlesztési lehetőségek:

- ☐ További közösségi média platformok integrálása.
- ☐ Bankkártyás fizetés / banki átutalási rendszer implementálása és automatizációja.
- ☐ A Backend forráskódból a Secret információk kiszervezése Azure DevOps/GitHub rendszerbe, ahogyan az a Frontend rendszer esetében is történt.
- ☐ Számlázási rendszer implementálása és automatizációja.
- ☐ Autopilot funkció fejlesztése, mely lehetővé teszi a teljesen automatizált tartalomgyártást.
- ☐ További autentikációs platformok integrálása annak érdekében, hogy ne csak Google fiókkal lehessen bejelentkezni.
- ☐ Oldal kinézetének továbbfejlesztése (pl.: szín, képek, elrendezés, logók)
- ☐ Oldal továbbfejlesztése felhasználóbarát szempontból.
- ☐ Szöveg generálási és kép kiválasztási algoritmus továbbfejlesztése.
- ☐ Facebook Business verifikációs folyamat és App Review folyamat lefolytatása.

HIVATKOZÁSJEGYZÉK

- Microsoft: Ismerje meg az ingyenes Azure-szolgáltatásokat, URL: <https://azure.microsoft.com/en-us/pricing/free-services/> Letöltve 2023.09.15.
- Microsoft: Azure Functions. Eseményvezérelt kiszolgáló nélküli kódfüggvény végrehajtása teljes körű fejlesztői környezettel, URL: <https://azure.microsoft.com/hu-hu/products/functions/> Letöltve 2023.09.15.
- Microsoft: Azure Functions-útmutató Python-fejlesztőknek, 2023.05.25., URL: <https://learn.microsoft.com/hu-hu/azure/azure-functions/functions-reference-python?pivots=python-mode-decorators&tabs=asgi%2Capplication-level> Letöltve 2023.09.15.
- OpenAI: Introducing ChatGPT, URL: <https://openai.com/blog/chatgpt> Letöltve 2023.09.15.
- OpenAI: Libraries, URL: <https://platform.openai.com/docs/libraries> Letöltve 2023.09.15.
- Microsoft: Natív felhőalkalmazások létrehozása egy gyors NoSQL-adatbázissal és egy ingyenes Azure-fiókkal, URL: <https://azure.microsoft.com/hu-hu/free/cosmos-db/> Letöltve 2023.09.15.
- Microsoft: Azure Functions csatlakoztatása az Azure Cosmos DB-hez a Visual Studio Code használatával, 2023.06.24.
<https://learn.microsoft.com/en-us/azure/azure-functions/functions-add-output-binding-cosmos-db-vs-code?tabs=in-process%2Cv1&pivots=programming-language-python> Letöltve 2023.09.15.
- React: The library for web and native user interfaces, URL: <https://react.dev> Letöltve 2023.09.15.
- Microsoft: Static Web Apps, URL: <https://azure.microsoft.com/hu-hu/products/app-service/static> Letöltve 2023.09.15.
- Microsoft: Gyorsmutató: Az első statikus webhely elkészítése az Azure Static Web Apps segítségével. 2023.09.07, URL: <https://learn.microsoft.com/hu-hu/azure/static-web-apps/getting-started?tabs=react> Letöltve 2023.09.15.
- Google: Identity Platform, URL: <https://cloud.google.com/identity-platform> Letöltve 2023.09.15.

- Microsoft: Azure DevOps: Modern fejlesztési eszközökkel jobban tervezhet, hatékonyabban együttműködhet, és gyorsabban szállíthatja az elkészült termékeit, URL: <https://azure.microsoft.com/hu-hu/products/devops> Letöltve 2023.09.15.
- GitHub: GitHub Actions: Automate your workflow from idea to production. <https://github.com/features/actions> Letöltve 2023.09.15.
- Microsoft: Azure Pipelines: Folyamatos buildelés, tesztelés és üzembe helyezés bármilyen platformra és felhőbe, URL: <https://azure.microsoft.com/hu-hu/products/devops/pipelines> Letöltve 2023.09.15.
- Facebook: Long-Lived Access Tokens, URL: <https://developers.facebook.com/docs/facebook-login/guides/access-tokens/get-long-lived> Letöltve 2023.09.15.
- Microsoft: Helyi fejlesztés Azure Functions a Core Tools használatával. 2023.08.28. <https://learn.microsoft.com/hu-hu/azure/azure-functions/functions-run-local?tabs=macos%2Cisolated-process%2Cnode-v4%2Cpython-v2%2Chttp-trigger%2Ccontainer-apps&pivots=programming-language-python> Letöltve 2023.09.15.
- React: Getting started, URL: <https://create-react-app.dev/docs/getting-started/> Letöltve 2023.09.15.
- Microsoft: Azure DNS: Az Azure DNS-sel az Azure-ban üzemeltetheti DNS-tartományait, URL: <https://azure.microsoft.com/hu-hu/products/dns> Letöltve 2023.09.15.
- OpenAI: Welcome to the OpenAI platform, URL: <https://platform.openai.com/overview> Letöltve 2023.09.15.
- Pexels: Start building with the power of Pexels, URL: <https://www.pexels.com/api/> Letöltve 2023.09.15.
- Facebook: Facebook-oldal létrehozása, URL: <https://hu-hu.facebook.com/business/help/104002523024878> Letöltve 2023.09.15.
- Pexels: API Introduction, URL: <https://www.pexels.com/api/documentation/> Letöltve 2023.09.15.
- OpenAI: Introduction, URL: <https://platform.openai.com/docs/api-reference> Letöltve 2023.09.15.

- Microsoft: Megosztott feladatkörök a felhőben, URL: <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/shared-responsibility> Letöltve 2023.09.15.
- Microsoft: Azure hálózati architektúra. 2023.06.01. <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/infrastructure-network> Letöltve 2023.09.15.
- Microsoft: Az Azure-infrastruktúra biztonsága. 2023.06.01., URL: <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/infrastructure> Letöltve 2023.09.15.
- Microsoft: Azure létesítmények, létesítmények és fizikai biztonság. 2023.03.25., URL: <https://learn.microsoft.com/hu-hu/azure/security/fundamentals/physical-security> Letöltve 2023.09.15.
- Rubóczki Edit Szilvia: „Az évtized legforradalmibb és talán legnépszerűbb IT infrastruktúra megoldása a felhő technológia.”, URL: http://lib.uni-obuda.hu/sites/lib.uni-obuda.hu/files/Ruboczki_Edit_Szilvia_ertekezes.pdf 59. oldal Letöltve 2023.10.07.

IRODALOMJEGYZÉK

- Petkovics Imre, Petkovics Ármin: Az informatika jövője a felhőben van, 2010, URL: http://www.vmtt.org.rs/mtn2010/492_502_Petkovics_A.pdf Letöltve: 2023.10.07.
- Váradi József: Adatvédelem a számítási felhőben, 2012, URL: <http://publikaciok.lib.uni-corvinus.hu/publikus/tdk/20120326135701.pdf> Letöltve: 2023.10.07.
- Rubóczki Edit Szilvia: A felhasználók biztonságtudatosságának jelentősége a publikus felhőszolgáltatások nagyvállalatoknál történő bevezetésekor, 2019, URL: http://lib.uni-obuda.hu/sites/lib.uni-obuda.hu/files/Ruboczki_Edit_Szilvia_ertekezes.pdf Letöltve 2023.10.07.
- A Magyar Nemzeti Bank 4/2019. (IV.1) számú ajánlása a közösségi és publikus felhőszolgáltatások igénybevételéről, URL: <https://www.mnb.hu/letoltes/4-2019-felho.pdf> Letöltve: 2023.10.07.

- Katonka Zsolt: A mesterséges intelligencia szerepe a mindennapi életben, 2017, URL: http://dolgozattar.uni-bge.hu/12581/1/katonka_zsolt_2017jun_publikus.pdf. Letöltve 2023.10.07
- Dr. Dudás László: Mesterséges intelligencia, URL: <http://ait.iit.uni-miskolc.hu/~dudas/MIEAok/MIEa1.PDF> Letöltve 2023.10.07.
- Rikk János, Pitlik László: ChatGPT-kísérletek-programozás, 2023, URL: <https://m2.mtmt.hu/gui2/?mode=browse¶ms=publication;34071055> Letöltve 2023.10.08.
- Pitlik László: Token-based chatGPT3 – Example, 2023, URL: <https://m2.mtmt.hu/gui2/?mode=browse¶ms=publication;34071014> Letöltve 2023.10.08.
- Microsoft: Shared responsibility in the cloud, 2023.09.29., URL: <https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility> Letöltve 2023.09.15.
- Microsoft: Azure infrastructure security, 2023.02.01., URL: <https://learn.microsoft.com/en-us/azure/security/fundamentals/infrastructure> Letöltve 2023.09.15.

MELLÉKLET

1. melléklet: Social AI API.yaml
2. melléklet: User.json
3. melléklet: Microsoft számla.pdf
4. melléklet: Google számla.pdf
5. melléklet: Rackhost számla.pdf
6. melléklet: OpenAI számla.pdf
7. melléklet: Forráskódok.zip
8. melléklet: ChatGPT Swagger dokumentáció generálás.html
9. melléklet: ChatGPT algoritmus szintaxis hibák javítása.html
10. melléklet: ChatGPT programozási könyvtárakkal kapcsolatos tanácsadás.html
11. melléklet: ChatGPT magyarázattal egybekötött fordítás.html
12. melléklet: Social AI.mp4 (<https://www.youtube.com/watch?v=zrq5ssPQK5M>)
13. melléklet: Prog Europe Kft. alapító okirat.pdf
14. melléklet: Prog Europe Kft. bejegyző végzés.pdf