SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Szoftver rendszerek modellezése

Szoftverfejlesztés Msc.

I. év

PletykAI

hírek intelligensen csoportosítva.

Tanár: **dr. Szántó Zoltán**

Diák: **Bács Balázs**

Tartalom

Bevezetés

Napjainkban sok helyről hozzáférhetünk a friss hírekhez, rengeteg híroldal működik, és mindegyik arra törekszik, hogy a lehető leggyorsabban értesítse az olvasóit a világban történő eseményekről. Ezeken az oldalakon általában szűrhetjük a híreket téma szerint (politika, sport, közélet stb.), de az olyan híroldal ritka, amely az érzelmi töltet vagy hangulat szerint csoportosítja a tartalmait.

Az alkalmazás célja éppen ezért az, hogy az olvasó hangulat szerint osztályozva lássa a cikkeket, és könnyen választhasson a kategóriák között. Minden cikk rendelkezik egy címkével, amely az adott cikk hangulatát, érzelmi töltetét jelöli, illetve kiválasztható, hogy csak egy bizonyos csoportba tartozó cikkeket jelenítsen meg.

A PletykAI egy olyan webes alkalmazás, amely egy híroldal (www.maszol.ro) főoldalát fésüli át (webscraping), onnan összegyűjti a cikkeket, majd az adatok elemzése és feldolgozása után azokat osztályozva megjeleníti.

Adatok feldolgozása alatt a következő műveleteket kell érteni:

- releváns HTML tagek tartalmának kinyerése az oldal forráskódjából
- cikkek címének lefordítása angol nyelvre
- a cikkek felcímkézése egy mesterséges intelligencia által az szerint, hogy pozitív, negatív vagy semleges hangvételű

A téma áttekintése

A hangulatelemzés (más néven véleménybányászat vagy érzelmi mesterséges intelligencia) természetes nyelvi feldolgozás, szövegelemzés, számítógépes nyelvészet és biometrikus adatok felhasználása az érzelmi állapotok és szubjektív információk szisztematikus azonosítására, kinyerésére, számszerűsítésére és tanulmányozására. A hangulatelemzést széles körben alkalmazzák a vásárlói anyagok, például vélemények és felmérésekre adott válaszok, online és közösségi média, valamint egészségügyi anyagok értelmezésére, amelyek a marketingtől az ügyfélszolgálaton át a klinikai orvoslásig hasznosak lehetnek.

A projekt célja

A projekt célja főként az volt, hogy kicsit jobban megismerkedjek a python nyelvvel, annak webfejlesztést érintő lehetőségeivel, valamint az érzelemfelismeréssel és annak alkalmazási módszereivel. Ez mellett cél volt egy szofver projekt létrehozásában is tapasztalatot szerezni, az ötleteléstől kezdve, a projektmenedzsmenten keresztül egészen a fejlesztésig és tesztelésig.

Felhasznált technológiák

Flask

Az alkalmazás alapját egy Flask webalkalmazás képezi, amely egy Pythonra épülő webes keretrendszer. Mikrokeretrendszerként jellemzik, mert nem igényel különleges eszközöket vegy könyvtárakat. Nincs beépített adatbáziskezelője, sem űrlap validációja, ezért viszonylag egyszerű a megtanulása és gyorsan lehet használható eredményt elérni vele, ugyanakkor ezek a tulajdonságai a negatívumai is lehetnek egyben. Elérhető hozzá sokféle kiegészítő csomag, amelyekkel bővíthetők a funkcionalitások.

Jinja

Ez egy web template engine, amit a Flask használ. Segítségével Python utasításokat adhatunk meg a HTML állományokban, így dinamikus működéssel ruházhatunk fel egy statikus oldalt.

BeautifulSoup

A BeautifulSoup egy Python-csomag HTML- és XML-dokumentumok elemzésére. Létrehoz egy elemzőfát az elemzett oldalak számára, amely felhasználható az adatok HTML-ből való kinyerésére (webscraping). Segítségével kibányászható a HTML tagek tartalma a weboldalak forráskódjából.

Bootstrap

Ingyenes és open-source CSS keretrendszer, amely reszponzív weboldalak megvalósítására van kihegyezve. Tartalmaz HTML, CSS és JS alapú elemeket is, amelyekkel gyorsan és egyszerűen lehet formázni az oldal kinézetét, szerkezetét, betűket, gombokat stb.

googletrans

A Googletrans egy ingyenes Python csomag, amely a Google Fordító API-ját implementálja pythonba. Segítségével a kódon belül használhatjuk a google fordító minden funkcióját.

VADER (Valence Aware Dictionary and sEntiment Reasoner)

Ez egy érzelemfelismerés egy szövegelemző módszer, amely felismeri a szöveg polaritását (pozitív, negatív, semleges), legyen szó egy egész dokumentumról, bekezdésről, mondatról vagy csak egy szóról. Ezt a módszert széleskörűen használják vásárlói termékértékelések felmérésére, internetes bejegyzések vizsgálatára, illetve egyéb más területeken is.

Az érzelemfelismerés nem egy egyszerű feladat, mivel nem elég csak az egyes szavakat megvizsgálni, hanem lehetőleg az egész szövegkörnyezetet kell felmérni, és azt értelmezni. A feladatot tovább nehezítik a nyelvek sajátosságai, illetve az emberek változatos fogalmazási szokásai.

A VADER egy olyan lexikális alapú érzelemfelismerő modell, amely érzékeny a szöveg érzelmének polaritására és intenzitására is. Egy olyan szótárra támaszkodik, amelyben minden szóhoz hozzá van rendelve egy bizonyos pontszám, annak függvényében, hogy az a szó általában milyen hangulatot vagy érzelmet hordoz. Egy teljes szöveg érzelmi töltetét az azt alkotó szavak pontjainak összessége adja meg. Ugyanakkor A VADER alapvető kontextualitást is képes értelmezni (pl.: „did not love” – ezt negatívként értelmezi, annak ellenére, hogy a „love” alapvetően egy pozitívan pontozott szó).

A VADER előnyei más érzelemfelismerő megoldásokkal szemben:

- nincs szükség tanító adatra
- viszonylag jó értelmezi az olyan szövegeket is, amelyek tartalmaznak emotikonokat, szleng kifejezéseket, nagy betűket, írásjeleket
- egyszerűen implementálható Pythonban

A VADER 4 értéket ad vissza, ha egy szövegen lefuttatjuk a vaderSentiment csomag SentimentIntensityAnalyzer() objektumának polarity_scores metódusát.

```
print(sentiment.polarity_scores("This is an excellent car with great mileage"))
{'neg': 0.0, 'neu': 0.435, 'pos': 0.565, 'compound': 0.8316}
```

A negatív, a pozitív és a semleges érzelem is kap egy bizonyos százalékértéket, illetve a compound egy összesített érték. Ez -1 és +1 közötti érték lehet, a -1 a legnegatívabb, míg a +1 a legpozitívabb, illetve 0 a semleges. A felhasználótól függ, hogy milyen intervallumokat tekint pozitívnak, negatívnak vagy semlegesnek. Ebben a projektben ha compound <= -0.05 akkor negatívnak vesszük, pozitívnak, ha compound >= 0.05, egyébként meg semleges.

MongoDB

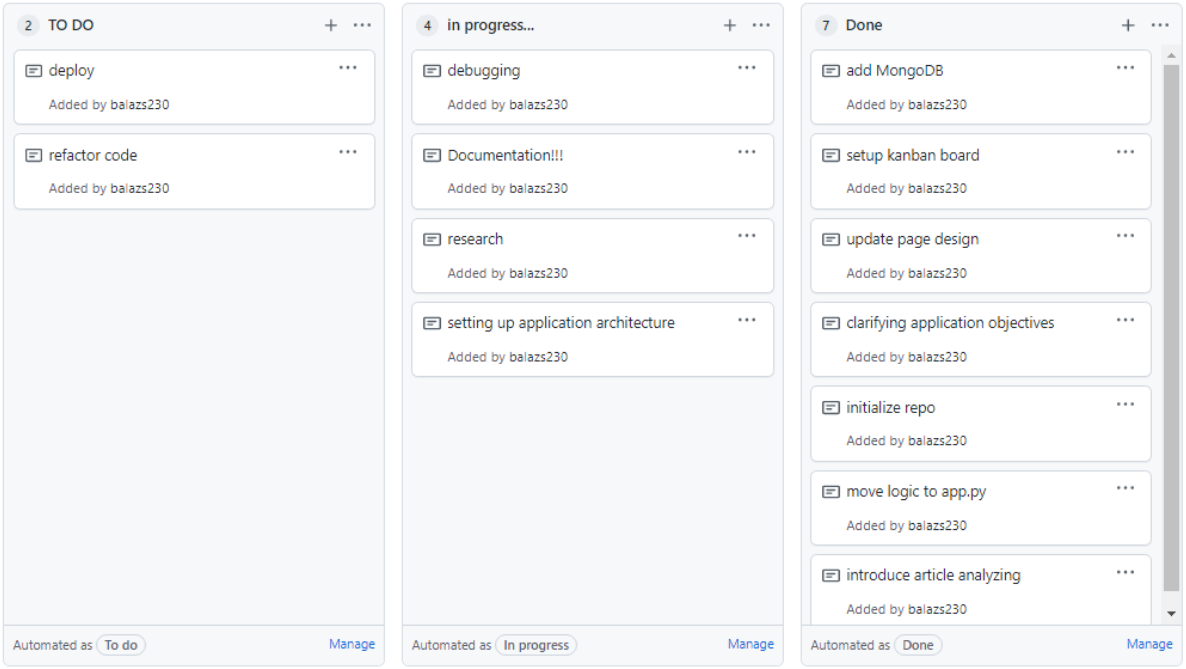
A MongoDB nyílt forráskódú dokumentumorientált adatbázis szoftver, a NoSQL (nem-relációs) adatbázisszerverek közé tartozik. A dokumentumokat JSON-szerű formátumban tárolja (BSON). A MongoDB adatbázis-kezelőben a hagyományos relációs adatbázisokkal ellentétben nem táblák vannak, hanem gyűjtemények (collections), ezeken belül pedig nem sorok és rekordok, hanem dokumentumok (documents). Érdekesség, hogy mindkettő csak akkor jön létre, amikor már adatot is teszünk bele. Minden új dokumentumhoz hozzárendel egy egyedi azonosítót (ID), amivel hivatkozni lehet az adott dokumentumra.

Az adatbázisba lehetőség van elmenteni a cikkeket, abban az esetben, ha később szeretnénk őket elolvasni. Ebben az esetben a mongo dokumentumba bekerül a cikk teljes eredeti HTML struktúrája.

A projektben lokális MongoDB klaszter van alkalmazva, egy lehetséges fejlesztési lehetőség lenne ezt lecserélni egy felhő alapú tárhelyre.

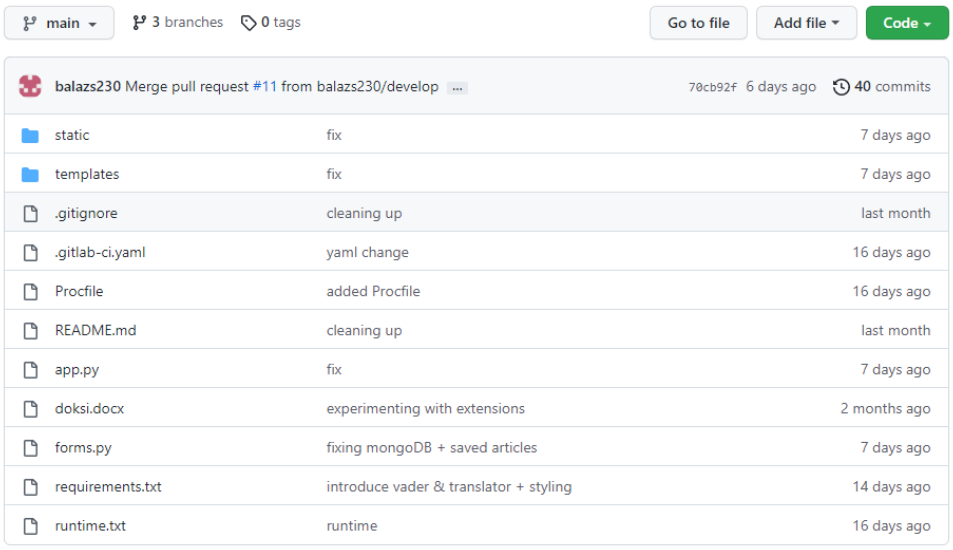
Projektmenedzsment

A projekt menedzseléséhez egy Kanban boardot használtam Github-on. Alább láthatók az elvégzett feladatok, implementált funkciók, a még folyamatban lévő feladatok, illetve néhány lehetőség a program továbbfejlesztésére. Törekedtem arra, hogy az egyes funckciók implementálására fordított idő egyenesen arányos legyen az illető funkció hozzáadott értékével, ez azonban nem minden esetben sikerült.



Verziókezelés

A project egy nyilvános GitHub repositoryban található. Itt két branchet használtam (*develop* és *main*). Minél gyakoribb kommitokkal dolgoztam, annak érdekében, hogy részletesen követhető és visszaállítható legyen.



Követelmények

Felhasználói követelmények

A webalkalmazás képes egy megadott híroldalról begyűjteni az adatokat, képes kinyerni a releváns információkat, feldolgozza és osztályozza a cikkeket hangulat alapján, majd megjeleníti őket emészthető formátumban.

Rendszerkövetelmények

- Windows, Mac, Linux operációs rendszer
- internetkapcsolat
- böngésző
- terminál

Funkcionális követelmények

A program helyi futtatásához arra van szükség, hogy legyen telepítve a Python 3.10+, telepítsük a requirements.txt fájlban levő csomagokat, illetve legyen konfigurálva egy helyi MongoDB adatbázis. Internetelérés szükséges az alkalmazás használatához.

Futtatás:

python app.py – a fő állomány futtatása

mongod – az adatbázis elindítása

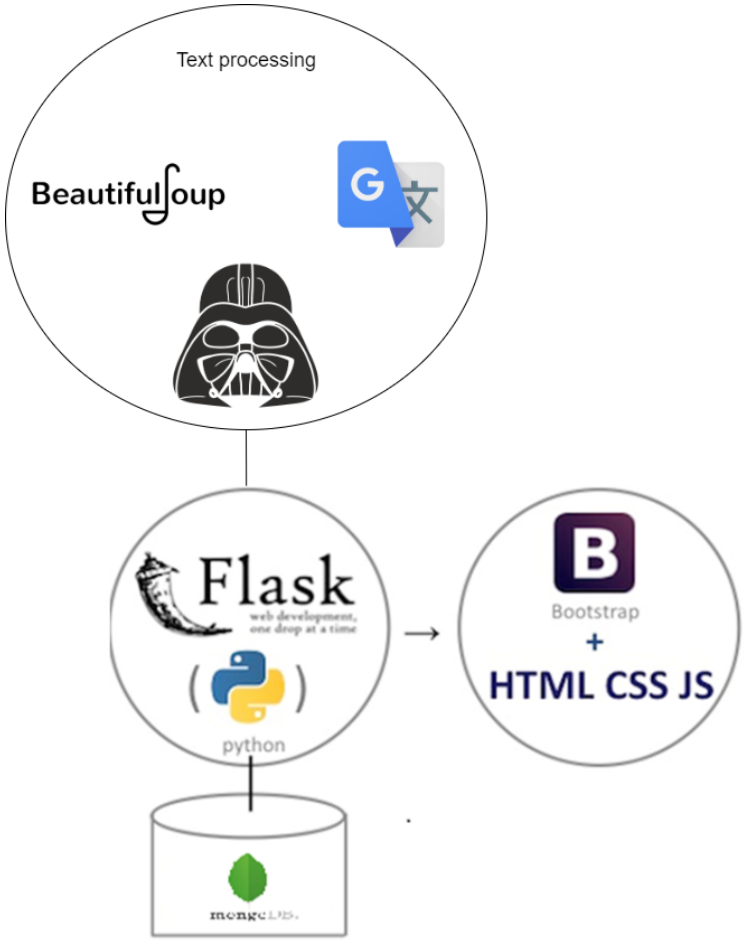
Nem-funkcionális követelmények

- észszerűen osztályozza a híreket
- egyszerű legyen váltani a kategóriák között
- elmentse a menteni kívánt cikkeket

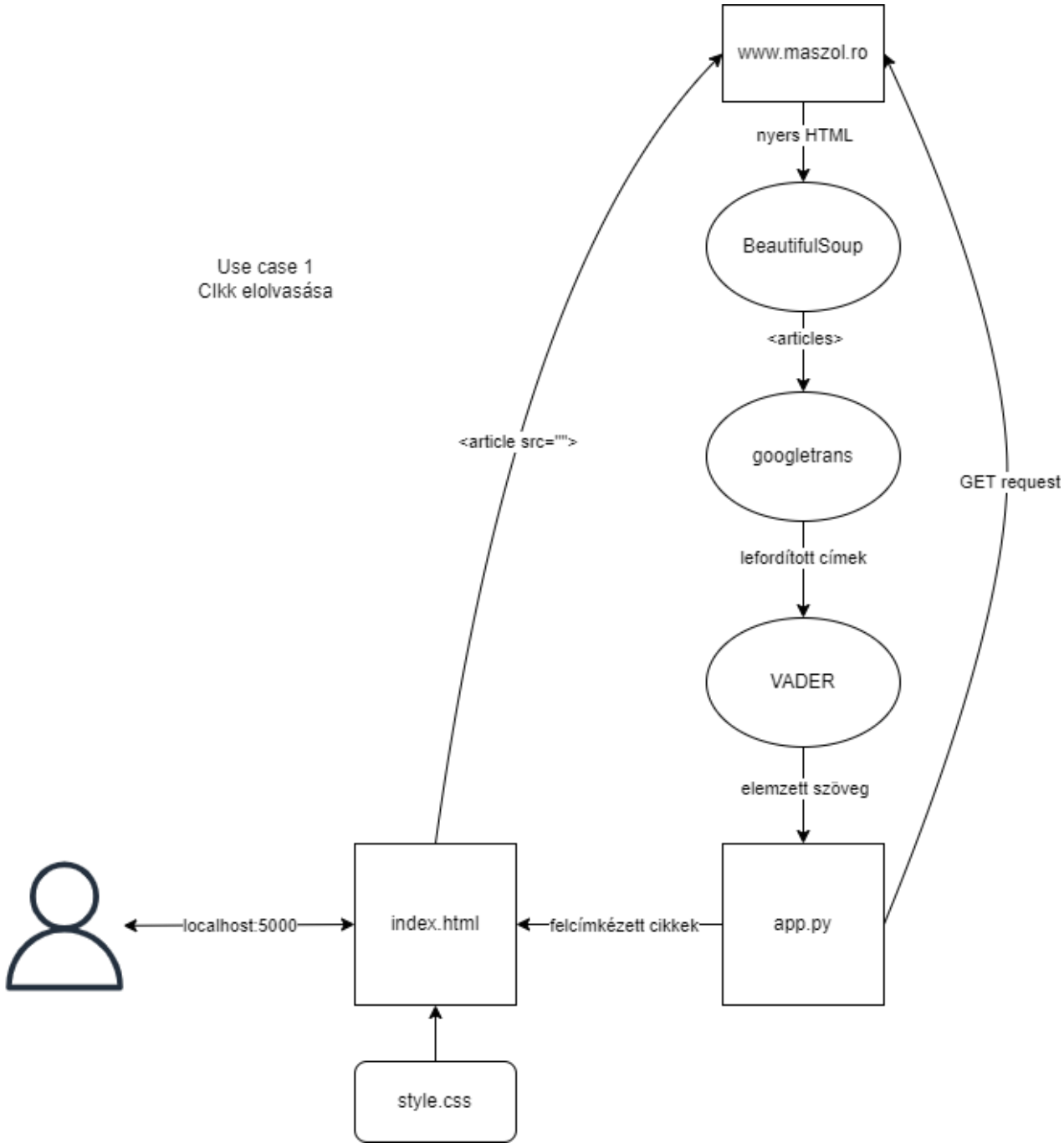
Fejlesztési lehetőségek

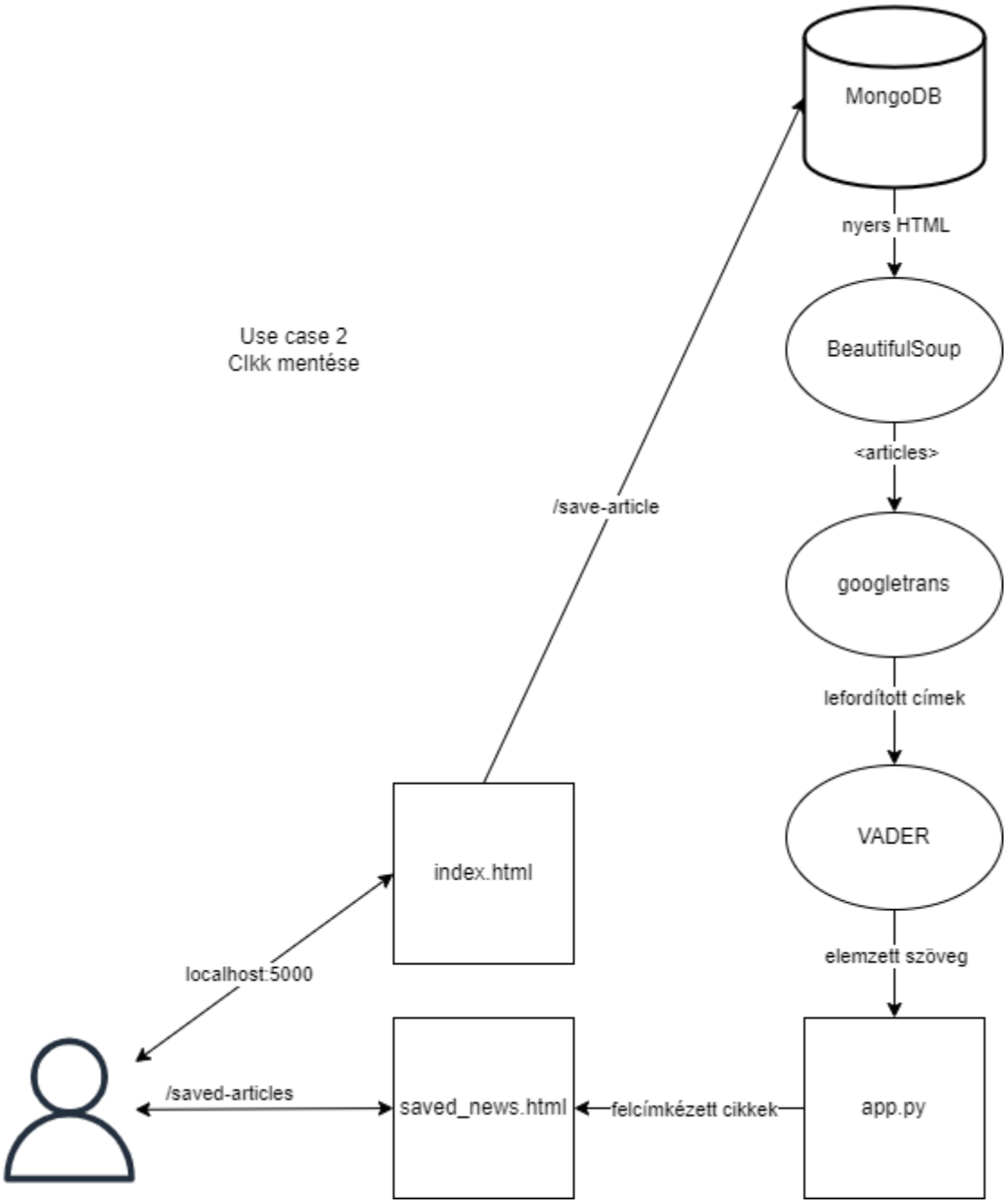
- az alkalmazás felbontása microservicekre (front-end, back-end, adatbázis, szövegfeldolgozás)
- front-end megvalósítása egy dedikált keretrendszerrel (pl.: React)
- kitelepítés valamilyen felhőszolgáltatásba (AWS, Azure, GCP)
- más híroldalak is szolgáljanak forrásként
- a hangulatelemzés finomítása több módszer ötvözésével
- kódminőség javítása

Architektúra

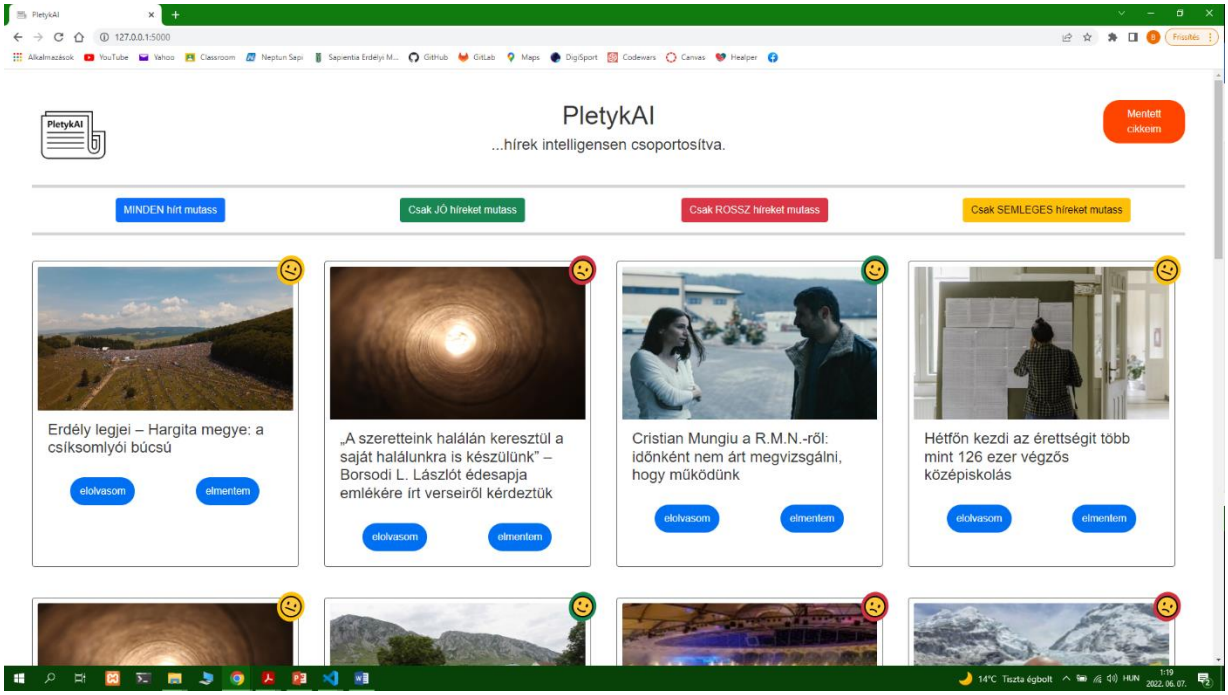


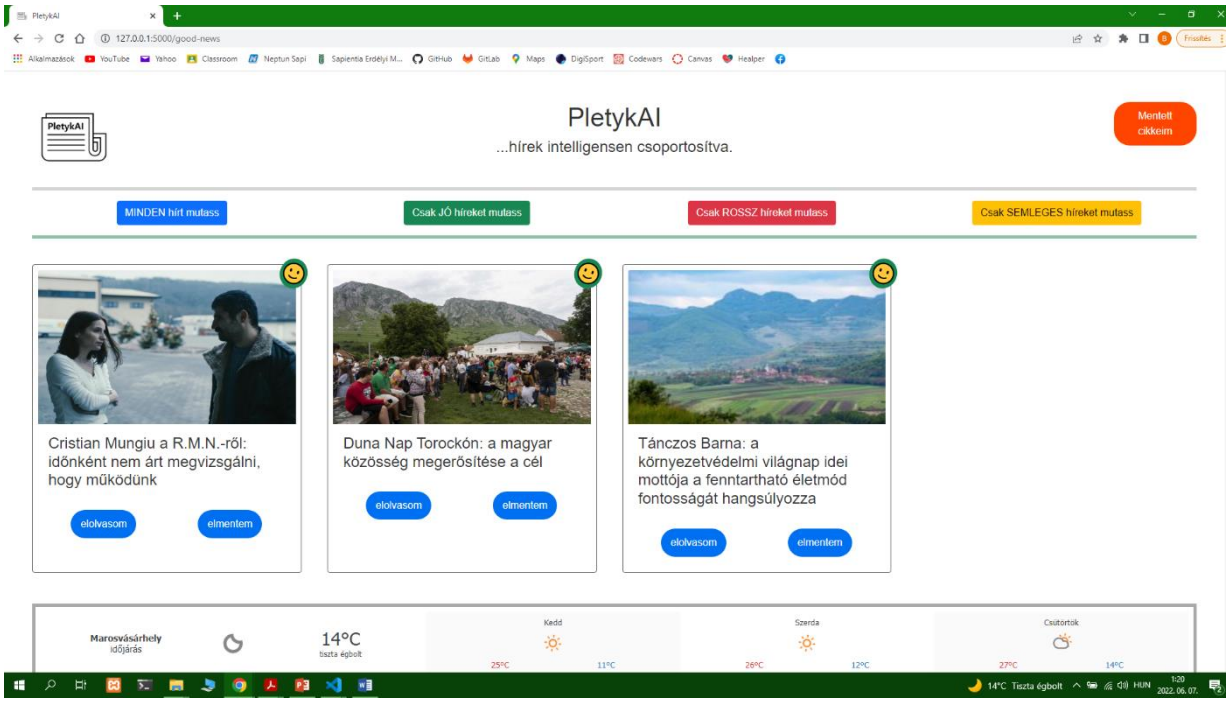
Használati esetek

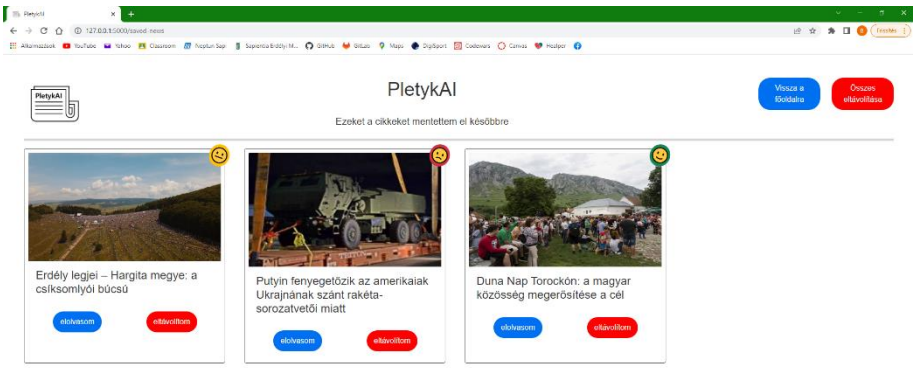



Use case 2
Clkk mentése
MongoDB
nyers HTML
BeautifulSoup
<articles>
googletrans
lefordított címek
VADER
elemzett szöveg
app.py
/save-article
index.html
localhost:5000
/saved-articles
saved_news.html
felcímkézett cikkek

Képernyőképek







Források

https://analyticsindiamag.com/sentiment-analysis-made-easy-using-vader/

https://github.com/cjhutto/vaderSentiment

https://jinja.palletsprojects.com/en/3.1.x/

https://flask.palletsprojects.com/en/2.1.x/

https://en.wikipedia.org/wiki/Sentiment_analysis

https://getbootstrap.com/docs/5.2/getting-started/introduction/

https://pypi.org/project/googletrans/

https://pypi.org/project/beautifulsoup4/