



Webalkalmazás fejlesztése az MNIST adatbázis humán validációjához

Készítette

Vámos Márton István

Programtervező informatikus BSc

Témavezető

Dr. Tajti Tibor Gábor

egyetemi docens

EGER, 2024

Tartalomjegyzék

Rövidítések	4
Bevezetés	5
1. A kérdőívekről	7
1.1. Kérdőív vagy felmérés?	7
1.2. A kérdőívek csoportosítása	8
1.2.1. Kvantitatív kérdőívek	8
1.2.2. Kvalitatív kérdőívek	10
1.2.3. Online kérdőívek	11
1.3. Egyik fő felhasználási területe, a Big Data	13
1.4. Validáció és megbízhatóság	13
2. Felhasználói dokumentáció	14
2.1. Üdvözljük a weboldalon!	14
2.1.1. Az MNIST-ről	15
2.2. Adatvédelmi irányelvek	16
2.3. Általános Szerződési Feltételek	17
2.4. Személyes adatok beállítása	18
2.4.1. Miért fontos ez nekünk?	19
2.5. A kérdőív kitöltése	19
2.5.1. CAPTCHA	20
2.5.2. Számok azonosítása	20
2.6. Visszajelzés küldése	21
2.7. A képgenerálás módjai	23
2.7.1. Véletlenszerű	23
2.7.2. Kiegyensúlyozott képválasztás	24
2.7.3. A leggyakrabban félreazonosított képek	24
2.7.4. További szempontok	24
3. Fejlesztői dokumentáció	26
3.1. Felhasznált technológiák	26

3.1.1.	Tervezéshez használt eszközök	26
3.1.2.	Webfejlesztéshez használt környezetek és eszközök	26
3.1.3.	Tesztelési környezet	27
3.1.4.	Verzió- és feladatkövetés	27
3.2.	Tervezés	28
3.2.1.	Tervezési szempontok	28
3.2.2.	Használati eset diagram	28
3.2.3.	Adatbázis	30
3.2.4.	Képernyőképek („mockup”)	31
3.3.	Fejlesztés	32
3.3.1.	A keretrendszer kiválasztása	32
3.3.2.	Az MNIST adatbázis integrációja	33
3.3.3.	Az oldalak működése – HTTP kérések kezelése	36
3.3.4.	További fontos funkciók	40
4.	Telepítés	44
4.1.	Egyszerű telepítés zip fájlból	44
4.2.	Teljes telepítés GitHub-ról	45
5.	Tesztelés	47
5.1.	API teszt	47
5.2.	Cypress – Automatizált teszt	48
6.	Továbbfejlesztési lehetőségek	50
6.1.	Magyar fordítás	50
6.2.	Eredmények megjelenítése	50
6.3.	Kitöltés folytatása	50
6.4.	Optimalizálás	51
6.5.	Az alkalmazás általánosítása más adatbázisokra	51
	Összegzés	52
	Irodalomjegyzék	53

Rövidítések

- **ÁSZF** – Általános Szerződési Feltételek
- **AI** – Artificial Intelligence
- **API** – Application Programming Interface
- **CAPTCHA** – Completely Automated Public Turing test to tell Computers and Humans Apart
- **CRUD** – Create, Read, Update and Delete
- **CSRF** – Cross-Site Request Forgery
- **GDPR** – General Data Protection Regulation
- **HTML** – Hypertext Markup Language
- **HTTP** – Hypertext Transfer Protocol
- **JSON** – JavaScript Object Notation
- **MNIST** – Modified National Institute of Standards and Technology
- **MVC** – Model-View-Controller
- **PDF** – Portable Document Format
- **PHP** – Personal Home Page
- **SPA** – Single Page Application
- **URL** – Uniform Resource Locator
- **UUID** – Unique User Identity

Bevezetés

Napjaink egyik legmeghatározóbb és leggyakrabban emlegetett területe az informatika. Ez egyáltalán nem meglepő, hiszen az utóbbi 50 évben, ezelőtt még sosem látott gyorsasággal fejlődött és költözött be a mindennapokba. Semmi sem bizonyítja ezt jobban, minthogy egyes kutatások[1] alapján a 2023-as év szava az *AI* (*Artificial Intelligence*, magyarul MI – Mesterséges Intelligencia) rövidítés lett.

Sokan használjuk tehát ezeket a szavakat és bár nehezen körülírható kifejezés, nagyjából tisztában vagyunk a jelentésével. Ebből eredően sokan feltétel nélkül hiszünk az ilyen mesterséges intelligencia által vezérelt „robotoknak” és nem vesszük figyelembe, hogy „ők” is képesek hibázni és téves információkat adni. Ezt nevezzük ilyenkor úgy, hogy az AI „hallucinál”. E tulajdonsága pedig sokszor félreértéshez és problémához vezethet.

Kevesen tudják, de a mesterséges intelligencia alapjai az úgynevezett *Machine Learning* (gépi tanulás), a *Deep Learning* (mély tanulás), illetve ezeknek még alacsonyabb szintje a neurális hálózatok.[2] Ehhez kapcsolódik szakdolgozati témám; *Webalkalmazás fejlesztése az MNIST adatbázis humán validációjához*, amely többek között az előbb említett hallucinálás elkerülésére is adhat választ és nyújthat megoldást.

Egy webalkalmazás, amely segítségével a különböző neurális hálónak vagy kis túlzással a mesterséges intelligenciának háttérébe való betekintést nyerhetünk az által, hogy megfigyeljük saját magunk, továbbá általánosságba véve az emberi agy működését. A webalkalmazásom során ugyanis az embereknek ugyanazokat a képeket kell azonosítaniuk, illetve felismerniük, mint amit egy neurális hálónak vagy ha jobban tetszik egy még „csecsemő szinten” lévő AI-nak.

Röviden tehát, vesszük az MNIST¹ adatbázis tartalmát, melyek esetemben nem mások, mint 0-9-ig tartó, kézzel írott számjegyek az emberi agy és szem számára is befogadható állapotban. Ezek valójában egyszerű fekete-fehér képek és egy teszt formájában kell eldöntenünk, hogy pontosan melyik számot ábrázolják. Ennek eredménye nagyon fontos visszajelzésekkel szolgálhat. Gondoljunk csak bele, hogy egyes képeket (számokat) sokszor teljesen másnak lát két hasonló gondolkodású ember. Mi a helyzet akkor, a már korábban is említett neurális hálóval? Hogyan tudnánk betanítani „öt”,

¹ A gépi tanulás területén használatos adatbázis, főként különböző algoritmusok betanításhoz használják.[3]

ha még mi magunk sem tudjuk eldönteni, hogy mit látunk igazán?

Célom, hogy az általam készített webalkalmazás segítségével egy olyan felületet biztosítsak, amely mindenki számára lehetőséget kínál egy kérdőív kitöltésére, mely képes felmérni az ember egyéni látásmódját. Továbbá a felmérésen résztvevők ezen tevékenységükkel rávilágíthatnak lényeges tanulmányokra és megfigyelésekre. Nem utolsósorban pedig, bízom benne hogy a gépi tanulás és az emberi agy összehasonlítása előrébb viheti majd tudásunkat a neurális hálók betanításához, segítségével tudatosabban állunk a mesterséges intelligenciához és összességében mind hozzájárulunk egy kicsit a jövő fejlődéséhez.

1. fejezet

A kérdőívekről

Ebben a fejezetben általánosságban írok a kérdőívek szerepéről, fajtáiról, felhasználási területeiről, illetve arról, hogy milyen kapcsolatban állnak a szakdolgozati témámmal. Igaz, a szakdolgozatom témája egy webalkalmazás, de szorosan kapcsolódik hozzá ez a fogalom. Ahhoz, hogy be tudjam mutatni és át tudjam adni a választott témám jelentőségét, fontos megértenünk, hogy a kérdőívekre miért és mikor van szükség. Célom, hogy a fejezet elolvasása után kapjunk egy tisztán érthető magyarázatot a kérdőívekről, illetve megfelelő háttérrel biztosítson, hogy később párhuzamot tudjunk vonni az alkalmazásommal kapcsolatban.

1.1. Kérdőív vagy felmérés?

A kérdőív szó nem tűnik túl bonyolultnak. A magyar nyelvtanban mégis rengeteg szinonimáját[4] ismerjük. Legtöbbször talán a *felmérés* vagy *teszt* szóval hasonlítanak össze, de itt vannak még az *úrlap*, *adatlap*, *blanketta*, *kartoték*, *feladatlap* és *kérdés-sor* szavak is, amelyekkel többé-kevésbé, mind körül lehetne írni. Igaz, a magyar nyelv sokrétű és változatos, ezért talán érdemes lehet megvizsgálnunk az angol nyelvű fordítását is. Angolul a kérdőívet úgy mondjuk, hogy *questionnaire*. Ez azonban itt sem az egyetlen megfelelője. Leggyakoribb rokon értelmű szavai[5] pl. az *application*, *census*, *sampling*, *survey* és a *poll*.

Sajnos ez sem könnyítette meg a dolgunkat, hiszen látjuk, hogy nincsen rá egyértelmű szó és emiatt nem tudjuk, hogy pontosan melyik szónak keressük a fogalmát. Próbáljuk meg tehát más módon megközelíteni a problémát. Saját magamon azt figyeltem meg, hogy legtöbbször a kérdőív és felmérés szavakat használok a fent felsoroltak közül. E két rokon értelmű szó különbségére az alábbi magyarázatot találtam:

„the main difference between a questionnaire and a survey is that a questionnaire is often used to get information from an individual, while a survey is a method of data collection targeted at a specific group.”[6]

A cikk írója azt mondja, hogy a kérdőív és a felmérés között az a számottevő különbség, hogy a kérdőíveknél főként az egyéni személyek által megszerzett adatok (válaszok) vannak a középpontban, míg a felmérés egy jó adatszerzési lehetőség arra, ha a célközönségünk egy bizonyos csoport.

A fenti megfogalmazást azért tartom fontosnak, mert kiemeli webalkalmazásom funkcióját és mondhatni „tekintélyét” azzal, hogy elmondhatom róla, mindkét szónak egyszerre tesz eleget. Azaz nyugodtan lehet kérdőív vagy felmérő oldalként is nevezni, egyik kifejezéssel sem fogunk tévedni.

Miért állítom ezt? Azért, mert a kérdőívet kitöltők mind egyéni személyek, akik saját döntéseik alapján válaszolnak az egyes képekre. Másrészt pedig, célcsoportunk közé tartozik minden olyan ember, aki kíváncsi az agyának azon működésére, ami meghatározza látásmódját. Tehát az egyéni válaszgyűjtés és egy bizonyos célcsoportból való adatgyűjtés is egyformán jelentős. Az általam készített weboldal pedig elsősorban arra adhat lehetőséget, hogy az emberek azonosítsák azokat a számokat, amelyeket a számítógépekre is bízunk. Így kapcsolódik össze témám a felmérésekkel és *„Az MNIST adatbázis humán validációjával”*.

1.2. A kérdőívek csoportosítása

Az alábbi alszakaszban különböző – a szakdolgozati témám szempontjából releváns – kérdőívek csoportjait mutatom be a teljesség igénye nélkül. Kitérek az egyes módszerek előnyeire és hátrányaira, valamint megpróbálom elhelyezni benne a saját szoftveremet is.

1.2.1. Kvantitatív kérdőívek

A kvantitatív (mennyiségi) kérdőívek vagy felmérések a számszerűsítést célozzák meg. Pontosabban azoknak az adatoknak a számszerűsített változatát, melyek az emberi tulajdonságokból és tevékenységekből származnak. Ez a fajta megközelítés jobban elősegíti a kutatási hipotézisek és feltevések könnyebb feltárását és megerősítését. Továbbá a kapott adatokat meghatározott szabályszerűségek szerint lehet csoportosítani, így kényelmesebb feldolgozást is biztosítanak.

Jellemzően a következő kérdésekre adnak választ az ilyen kérdőívek:

- mennyi(t)?
- hányan?
- hány százalékban?

Ebből következik, hogy mennyiségi mutatókkal, vagyis kvantitatív eredményekkel dolgozunk. Ezek az eredmények pedig, nem véletlenül a kérdőíves kutatások során szerezhetők meg.

A kvantitatív felmérések előnyei:

- biztosítja a kutatás eredményeinek strukturált felépítését
- a kutató rövid idő alatt jelentős mennyiségű információhoz juthat
- a kvantitatív eredmények alapvető következtetések levonására alkalmasak
- az eredmények könnyen átláthatóak és gyorsan feldolgozhatóak

A kvantitatív felmérések hátrányai:

- a kutató általában nem találkozik közvetlenül a kutatás alanyaival
- a módszer csak a strukturált válaszokat biztosítja, így a bonyolultabb összefüggések kevésbé nyilvánvalóak
- a kvantitatív eredmények gyakran csak „feketék vagy fehérek”, azaz a kérdéskör árnyalására kevésbé alkalmasak

[7]

Mindezeket egybevéve kijelenthetném, hogy webalkalmazásom kérdőív fajtáját tekintve kvantitatív. Látszatra talán egyértelműnek is tűnhet, mert azt gondoljuk, hogy csak számok azonosításáról vonunk le következtetéseket. Valójában, jobban belegondolva rengeteg számadatot gyűjtünk egy-egy válaszról úgy, hogy az azt kitöltő nem is sejtí. Félreértés elkerülése végett nem a *GDPR*-al szembemenő adatokat gyűjtjük, – erről külön a **Személyes adatok beállítása** szakaszban lesz szó – hanem a válasz azon tulajdonságait, melyek mindössze *metaadatoknak* minősülnek.

Metaadat

A metaadatok olyan információk vagy leírások, amelyek más adatokkal vagy tartalmakkal kapcsolatosak, segítve ezáltal azok megtalálását, értelmezését és rendezését. Ezek az adatok nem közvetlenül az eredeti tartalom részei, hanem azokat kiegészítik vagy magyarázzák.

A metaadatok lehetnek címek, szerzők nevei, dátumok, műfajok, formátumok vagy más technikai információk, attól függően, hogy milyen típusú tartalomról van szó (pl. képek, zenék, videók, dokumentumok stb.).

A metaadatok rendkívül fontosak lehetnek a tartalom hatékony kezelésében, keresésében és szervezésében, valamint segíthetnek a tartalomhoz kapcsolódó információk strukturált és könnyen áttekinthető módon történő tárolásában.[8]

A könnyebb megértés érdekében bemutatok egy példát a strukturált metaadatokra:

Dokumentum

- **cím:** „Webalkalmazás fejlesztése az MNIST adatbázis humán validációjához”
- **szerző:** Vámos Márton István
- **kiadás dátuma:** 2024. április 14.
- **témakör:** Szakdolgozat
- **nyelv:** Magyar
- **méret:** 1,59 MB
- **formátum:** PDF

Ennek tudatában úgy gondolom, hogy mielőtt megfogalmaznám webalkalmazásom kérdőív típusát, előbb vizsgáljuk meg a kvalitatív kérdőívek tulajdonságait is.

1.2.2. Kvalitatív kérdőívek

A kvalitatív megközelítés abban különbözik a kvantitatív módszertől, hogy nem az általános érvényű megállapítások feltárására, hanem inkább a specifikus, „szürke mező” elemzésére összpontosít. Ennek lényege, hogy a bonyolultabb, mélyebb összefüggéseket tanulmányozza és az egyedi helyzeteket, jelenségeket részletesebben vizsgálja meg. A kvalitatív kutatások gyakran kevesebb, de alaposabb adatokat használnak fel, valamint a kutatók közelebb kerülnek a felmérés résztvevőihöz.

A kvalitatív felmérések előnyei:

- a kutatók és a résztvevők jobban megérthetik a kutatás céljait és motivációit
- a módszer lehetővé teszi a jelenségek mélyebb megismerését, például a célcsoport vagy egyéni személyek gondolkodását és viselkedését
- a kutatók közelebbi kapcsolatot alakíthatnak ki a résztvevőkkel, ami elősegítheti a kutatás sikerét és az eredmények relevanciáját

A kvalitatív felmérések hátrányai:

- bár a módszer hitelességet és mélységet ad a kutatásnak, a felmérés szubjektivitást is hordozhat, ami befolyásolhatja az eredményeket
- a kvalitatív kutatások időigényesek lehetnek, mivel a részletes adatok gyűjtése és elemzése több időt vesz igénybe
- a módszer általában kevesebb adatot biztosít, ami csökkentheti a kutatás reprezentativitását

[7]

Így, hogy tudjuk mik tartoznak a kvalitatív kérdőívek közé, már egészen biztosan megtudom fogalmazni, hogy webalkalmazásom melyikhez tartozik igazán.

Véleményem szerint a kvantitás és a kvalitás nem zárja ki egymást. Esetemben úgy gondolom, hogy kvantitatív azért, mert rövid időn belül hatalmas információ mennyiséget vagyunk képesek felhalmozni. Másrészt kvalitatív azért, mert bár nem úgy tűnik, de mögöttes információkat („szürke mezős” adatokat) is nyerünk ki a válaszokból. Itt többek között például arra gondolok, hogy a kitöltőknek lehetőségük van névtelenül megnevezni egyetemi szakjukat, illetve azt, hogy jobb- vagy balkezes az illető (lásd 18. oldalon lévő **Személyes adatok beállítása** szakaszban). Az ilyen információkból és leadott válaszaikból pedig, mélyebb szintű következtetéseket és eredményeket kaphatunk.

A végső válasz tehát abban rejlik, hogy a későbbiekben mire fogjuk felhasználni a felmérésből származott adathalmazt.

1.2.3. Online kérdőívek

A szakdolgozati témámat illetően még mindenképpen ki szeretnék térni erre a pontra is. Esetemben egyértelműen meghatározható, hogy egy online kérdőívről van szó. Ennek ellenére, témáját és jellegét tekintve ez nem egy hétköznapi mondható online kérdőív. Összességében azonban egyáltalán nem áll tőle messze, csak megvalósításában más egy kicsit. Mire gondolok a hétköznapi online kérdőívek alatt?

Az utóbbi években, egyre népszerűbbé váltak az online, más néven internetes kérdőívek használata. Ez a módszer adataink gyűjtésének egyik meghatározó eszköze. Jellemzően egy tudatos mintával tesznek fel kérdéseket, a kitöltők pedig az interneten keresztül tudnak válaszolni. A kitöltés általában e-mailen, weboldalakon, közösségi média platformokon és más online csatornákon keresztül történik.

Manapság a szervezetek, kis- és nagyvállalatok különböző eszközöket alkalmaznak, hogy felméréseken keresztül visszajelzéseket és véleményeket gyűjtsenek különféle té-

mákról, termékeikről, szolgáltatásaikról. Ezen visszacsatolásokat pedig marketingstratégiák változtatásához és meglévő funkciók fejlesztéséhez használják fel.

Az online kérdőívek előnyei:

- a célcsoport gyors elérése
- lehetőséget kínál valós idejű elemzésre
- költséghatékony
- kisebb hibahatár
- a válaszadók számára könnyen értelmezhető
- időmegtakarítás a kutatóknak
- őszintébb válaszok

Az online kérdőívek hátrányai:

- félreinformálás, fals válaszadás
- kérdéses adatvédelem és megbízhatóság
- korlátozott hozzáférés

[9]

A szempontokat megnézve a legnagyobb kérdést számomra a válaszadás módja vetette fel. Mi garantálja azt, hogy nem-e szándékosan adnak téves válaszokat a képekre, befolyásolva ezzel a mérési eredményeket? Nehéz kérdés ez, nem véletlenül írtam mindkét csoportosításba. A legtöbb online kérdőívnek nem igazán van megoldása arra, hogy az ilyen hamis információkat megfelelően szűrje.

Webalkalmazásomban, a válaszadásnál minden egyes képen külön a „gondolkodásra” szánt időt is eltárolom, így többek között, azt a funkciót is szolgálhatja, hogy segítségével kiszűrhetjük a gyanúsán beérkezett válaszokat.

Összességében, előnyeiből és hátrányaiból egyértelműen látjuk, hogy az online kérdőív mire ad megoldást, illetve mire kell figyelniünk. Mindent összevetve azonban, úgy gondolom, hogy a kitöltők és az információkat feldolgozó kutatók számára is kényelmesebb az online módszer.

1.3. Egyik fő felhasználási területe, a Big Data

Szakdolgozati témámat tekintve az oldalam egyik legjelentősebb felhasználási területe a *Big Data*. A Big Data a megszokotthoz képest sokkal nagyobb adatmennyiségeket jelent, amelyeket hagyományos módszerekkel nehéz vagy lehetetlen lenne feldolgozni. Ez a jelenség új korszakot nyitott többek között a design, az orvostudomány, a szoftverfejlesztés és a marketing területén is.

A Big Data megjelenése forradalmasította a kutatási módszereket, ideértve a kvantitatív kutatásokat is. A kvantitatív és online kutatásokban ez azt jelenti, hogy nagyobb adatmennyiségeket gyűjthetünk be és dolgozhatunk fel, mint valaha. Az olyan forrásokból, mint a különböző online tevékenységek (például webes keresések és tranzakciók) hatalmas adathalmazokat hoznak létre, amelyek segítségével részletesen feltérképezhetjük az emberi preferenciákat és viselkedéseket, illetve taníthatunk be neurális hálózatokat.

A nagy adatmennyiség feldolgozásához azonban komoly számítási kapacitásra van szükség. A Big Data gyakran egy folyamatosan termelődő adatsor, nem pedig egy rögzített adatbázis, ezért folyamatos megfigyelést és elemzést igényel. Ehhez elengedhetetlen az emberi felügyelet és kézben tartás, nem elég pusztán csak algoritmikus megközelítés. Gondoljunk csak bele, hogy mi történik akkor, ha valaki szándékosan rossz válaszokat ad le a kérdőív többszöri kitöltésével. Ezek befolyásolhatják a többi adat hitelességét. Az emberi beavatkozás könnyen és gyorsan kiszűrheti ezt, például ha a hibás válaszok között milliszekundum eltérés van, erős gyanút kelthet, hogy valami nincs rendben.

Végül, de nem utolsó sorban a Big Data és gépi tanulás kombinációja számos felhasználási lehetőséget kínál hibaszűrésre és prediktív elemzésre az ipar és a kutatás területén. Azonban itt fontos megjegyezni, hogy a Big Data és az AI együttesen hatékony eszközök, de attól még továbbra is szükség lehet az emberi közreműködésre.[10]

1.4. Validáció és megbízhatóság

A kérdőív megbízhatósága számos dologtól függ. A webalkalmazásom már önmagában is az emberi validációt állítja középpontba, hiszen emberek sokasága dönti el, hogy melyik számot látja a kérdőíven megjelenő képeken. *Hogyan tudjuk biztosan, hogy az emberek válaszai megbízható forrásból származnak?* Ez egy nagyon nehéz kérdés, hiszen minden ember gondolkodásmódja más és más, ezért nincsen erre bevált protokoll.

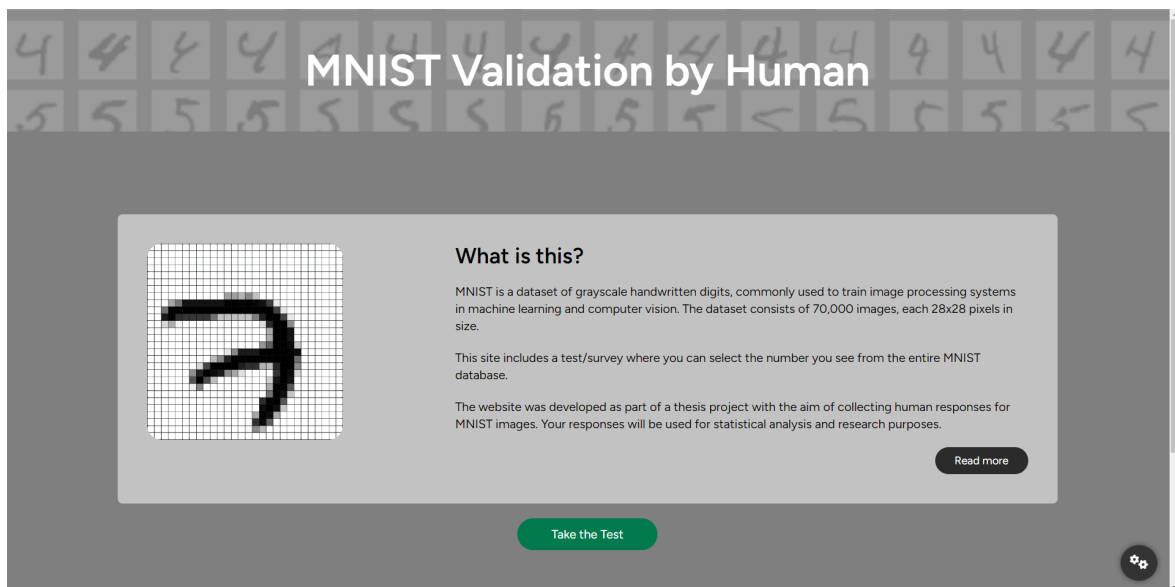
Azonban, ha a *hogyan*-ra fektetjük a hangsúlyt lehet, hogy közelebb jutunk a válaszhoz. Véleményem szerint azzal, hogy egyértelmű utasításokat és kérdéseket teszünk fel a teszt kitöltőinek, minimalizáljuk az ugyanarra a képekre érkező válaszok közötti eltéréseket és értelmezési nehézségeket. Ilyen szempontból, az alkalmazásom témáját tekintve rendkívül egyszerű, biztosítva ezzel a megfelelő validációt és megbízhatóságot.

2. fejezet

Felhasználói dokumentáció

Ebben a fejezetben felhasználói oldalról mutatom be webalkalmazásomat. Kitérek minden olyan részletre, ami fontos lehet az oldal látogatóinak, illetve teszt kitöltőinek. A fejezetben írok többek között az MNIST adatbázisról, az **Adatvédelmi irányelvek** és **Általános Szerződési Feltételek** jelentőségéről, magáról a kérdőív használatáról és működéséről, valamint a visszajelzés küldéséről és a személyes beállításokról. Az utolsó részben pedig egy kicsit más nézőpontból közelítem meg az oldalt.

2.1. Üdvözljük a weboldalon!



2.1. ábra. A webalkalmazás kezdőoldala

Az URL betöltésekor a kezdőoldalt pillantjuk meg először. Itt röviden összefoglalom az oldal funkcióját és létrejöttének célját. Írok az MNIST adatbázisról, illetve arról, hogy milyen összefüggésben van a gépi tanulással. Továbbá ismertetem a látogatókkal,

hogyan az oldal egy kérdőívet tartalmaz, amely során számokról fog látni képeket, amelyeket be kell azonosítani. Végül tájékoztatom a válaszok feldolgozásának módjáról és céljáról, valamint hogy a projekt egy szakdolgozati munka keretében jött létre.

2.1.1. Az MNIST-ről

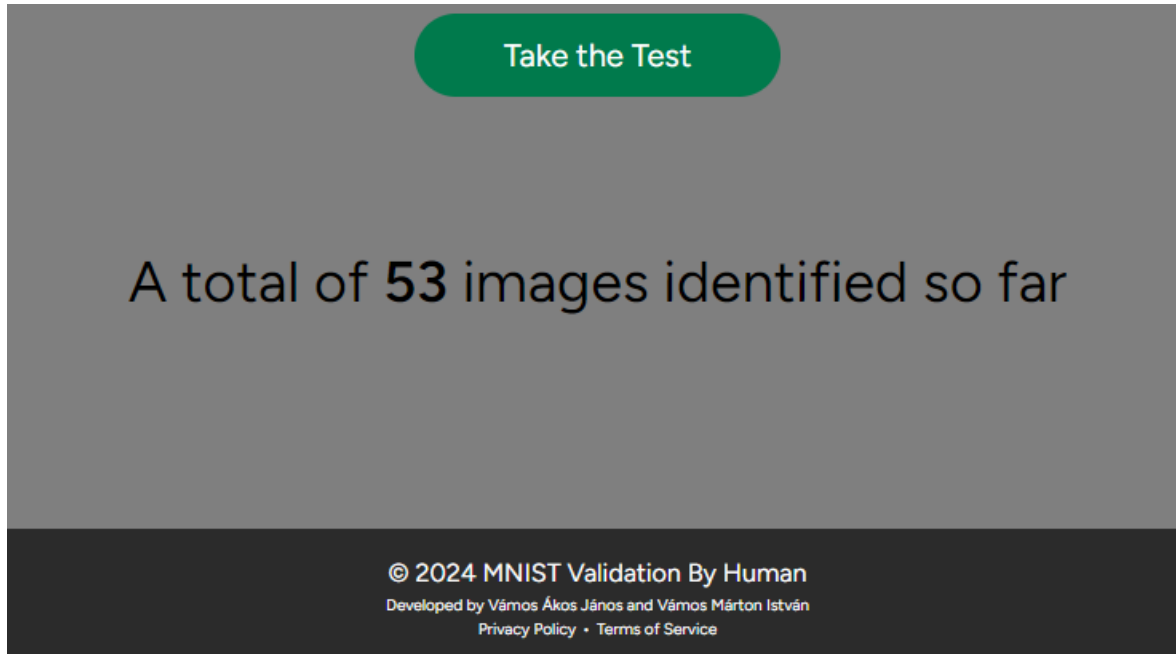
Az MNIST egy kézzel írott számjegyeket tartalmazó óriási adathalmaz. Főként a gépi tanulás területén alkalmazzák, tehát az adatok egy részével be tanítják, a másikkal pedig tesztelik, hogy lássák a tanítás eredményességét. Ez a folyamat mára már bárki számára elérhető és megvalósítható.

A *Python* programozási nyelvben számos **könyvtár importálására** van lehetőség. A *Tensorflow* és a *Keras* könyvtárak egyszerűen, mindössze két sor megadásával már elérhetővé is válnak. Segítségükkel a gépi tanulás és a neurális hálók modelljeit könnyedén elérhetjük. Pár sorral és egy kevés utánajárással már saját környezetünkben is tudjuk futtatni a fent leírt tanítást és tesztelést.

Az egész folyamat gyors és hatékony. Ennek következtében pedig már nem csak az én témám által feldolgozott MNIST adatbázis létezik. Rengeteg hasonló, előre elkészített adatbázist találhatunk az interneten. A témámhoz közelálló adatkészletek:

- **Extended MNIST (EMNIST)**: egy olyan adatkészlet, amely kiterjeszti az eredeti MNIST képeket és tartalmazza az angol ábécé kis- és nagybetűs változatát, valamint az arab számjegyeket.
- **Fashion-MNIST**: egy olyan adatkészlet, amely ruhadarabok képeit tartalmazza, amelyeket 10 kategóriába sorolnak (pl. pulóverek, cipők, ruhák stb.). Célja hasonló az MNIST adatkészlethez, de változatosabb képekkel.
- **medMNIST**: az orvosi képfeldolgozásra szakosodott adatkészlet, amely különböző orvosi képeket tartalmaz (pl. retinográfiai képeket).
- **3D MNIST**: a 3D MNIST adatkészlet olyan képek válogatását tartalmazza, amelyek 3 dimenziós (térbeli) információval rendelkeznek.
- **notMNIST**: a notMNIST adatkészlet a latin ábécé képeit tartalmazza. Célja a betűfelismerés mély tanulás segítségével.
- **Simpsons MNIST**: egy adatkészlet, amely az ismert televíziós sorozat, *A Simpsons család* karaktereinek képeit tartalmazza. Célja a szereplők felismerése és osztályozása.
- **AudioMNIST**: az AudioMNIST adatkészlet hangfelvételeket tartalmaz, amelyeket számok szövegének kiejtése alapján rögzítettek. Célja a beszédfelismerés mély tanuláson alapuló megközelítésének demonstrálása.[11]

A kezdőoldalon lejjebb görgetve láthatjuk, hogy eddig mennyi képzazonosítás, azaz válaszadás történt. Továbbá, az oldal alján a *footer* részben olvashatunk az oldal fejlesztőiről, valamint átléphetünk az **Adatvédelmi irányelvekre** és az **Általános Szerződési Feltételekre**.



2.2. ábra. A képzazonosítások száma és a lábléc

A webalkalmazást egy szakdolgozati társammal készítettem el. A kérdőív biztosítása, valamint az adatok gyűjtése tartozott az én feladataim közé. A kérdőívben megkapott válaszok feldolgozásáról, statisztikai eredmények kimutatásáról, illetve az ehhez tartozó felület biztosításáról pedig az Ő szakdolgozatában olvashatunk.

2.2. Adatvédelmi irányelvek

Az adatvédelem fontos fogalom az online környezetben. Az adatvédelmi gyakorlatok célja, hogy megvédjék a felhasználók személyes adatait a jogosulatlan hozzáférésektől vagy visszaélésektől.

Az adatvédelmi nyilatkozat egy olyan dokumentum, amely részletesen tájékoztatja a weboldal látogatóit az adatgyűjtés, kezelés és védelem gyakorlatáról. A nyilatkozatnak célja, hogy a felhasználók számára biztosítsa személyes adataik biztonságos és felelősségteljes kezelését.[12]

Az általam alkalmazott adatvédelmi irányelvek részletesen bemutatják, hogyan gyűjtjük, használjuk és védelmezzük a felhasználók (látogatók) adatait. A dokumentum, a *cookie*-k (sütik) használatától kezdve az adatkezelési célokig minden fontos információt tartalmaz.

1. Gyűjtött adatok

Weboldalunk látogatásakor három fő típusú *cookie*-t használunk:

- **XSRF-TOKEN:** ez a *cookie* segíti oldalunkat a *Cross-Site Request Forgery* (CSRF) támadások elleni védelemben.
- **laravel_session:** ez a *cookie* tárolja az információkat a jelenlegi látogatás során gyűjtött adatokról. Érvényessége egy év és nélkülözhetetlen a weboldal biztonsága és funkcionalitása szempontjából.
- **X-Client Token & UUIId:** létrehozunk egy ideiglenes, egyedi azonosítót minden látogató számára a felmérések kitöltésekor. Ez biztosítja azt, hogy a vendégek ne kaphassák meg ugyanazt a képet többször egy bizonyos időszakban. Ezeket az azonosítókat bármikor törölheti a böngészőbeállításokban.

2. Adatkezelés célja

A gyűjtött adatokat kizárólag weboldalunk biztonságának és funkcionalitásának javítására használjuk. Nem osztunk meg vagy értékesítünk személyes adatokat harmadik felekkel.

3. Cookie kezelése

Általában a *cookie*-kat a böngésző beállításaiban kezelheti. Fontos azonban, hogy a *cookie*-k letiltása vagy törlése befolyásolhatja a weboldal funkcionalitását és biztonságát.

4. Elfogadás

Weboldalunk használatával elfogadja az adatkezelési nyilatkozatot.

2.3. Általános Szerződési Feltételek

Az Általános Szerződési Feltételek (ÁSZF) fontos részét képezik a szerződéses jogrendszernek és gyakran találkozunk velük a mindennapi életben (pl. online vásárlás során vagy szolgáltatások igénybevételének alkalmával). Az ÁSZF fogalma alatt olyan előre meghatározott feltételeket értünk, amelyeket az egyik fél egyoldalúan határoz meg. Továbbá a szerződéskötés során ezek automatikusan alkalmazásra kerülnek, anélkül hogy a másik féllel egyedileg tárgyalnánk meg azokat.[13]

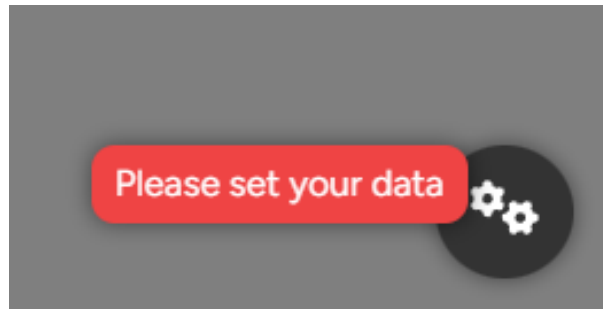
Weboldalunk egy kérdőív alapú platform.

- a kapott eredményeket statisztikai elemzésre és kutatási célokra használjuk fel
- a látogatók csak az általunk meghatározott módon használhatják a weboldalt
- kereskedelmi vagy etikátlan tevékenységekre nem alkalmazható

- fenntartjuk a jogot, hogy bármikor módosítsuk vagy frissítsük ezeket a feltételeket

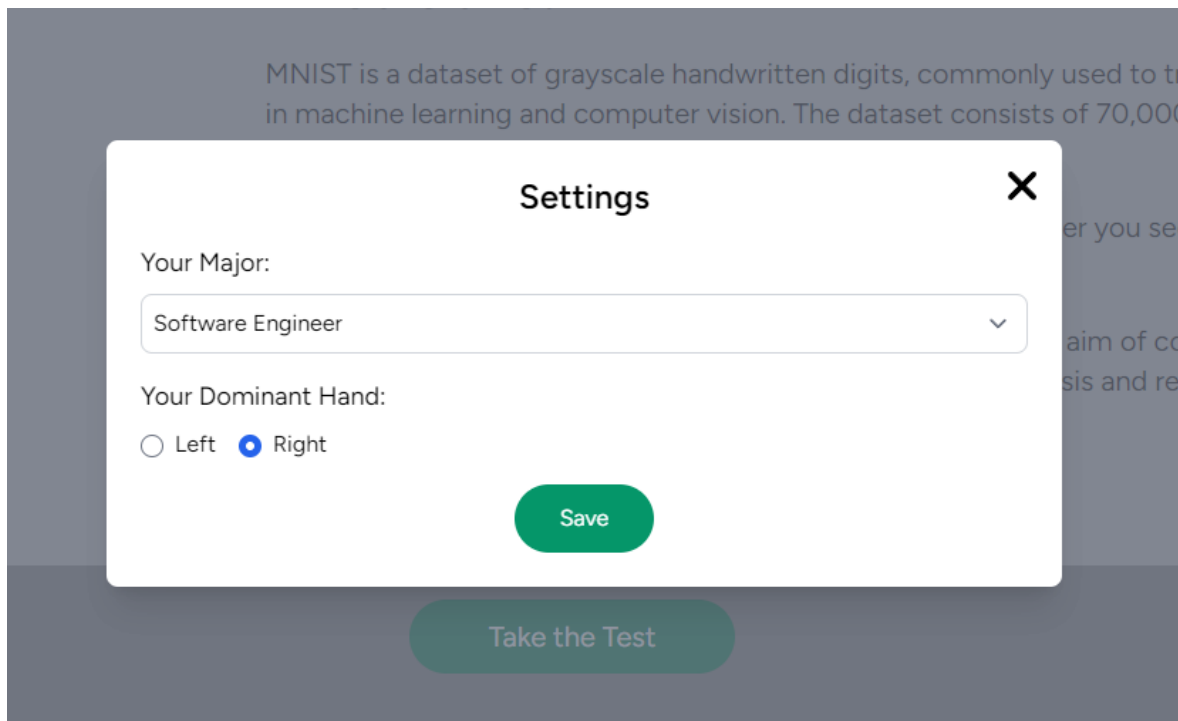
2.4. Személyes adatok beállítása

A korábban már említett személyes adatok beállítására is kínál lehetőséget a webalkalmazás. Az első látogatás kezdetétől „felugrik” egy beállítások ikon az oldal jobb alsó sarkán. Az ikonon lévő piros színű „*Please set your data*” („Kérem állítsa be az adatait”) felirat figyelmeztet arra, ha még nem konfiguráltuk volna azt.



2.3. ábra. Beállítások ikonon[14] lévő figyelmeztető felirat

Fontos, hogy a figyelmeztető jelzés mindaddig jelen lesz az ikonon, amíg be nem állítjuk azt. Azonban ettől függetlenül az alkalmazás teljes körűen használható.

A white modal window titled "Settings" with a close button (X) in the top right corner. Inside the modal, there is a form with two sections. The first section is "Your Major:" followed by a dropdown menu showing "Software Engineer". The second section is "Your Dominant Hand:" followed by two radio buttons: "Left" and "Right", with "Right" being selected. Below these sections is a green "Save" button. In the background, a dark grey button labeled "Take the Test" is visible.

2.4. ábra. A „Beállítások” menü előugró ablaka

Amint a fenti képen is látható, a „Beállítások” menü, az ikonra kattintva egy *pop up*-ban (előugró ablakban) jelenik meg. Ezen a felületen választható ki a látogató egyetemi

szakja, továbbá megadható, hogy bal vagy jobb kezes. A „Save” (Mentés) gombra kattintva fogadjuk el a beállításokat.

A sikeres beállítás után a piros figyelmeztető jelzés eltűnik. Az ikonra visszakattintva természetesen ismét módosíthatjuk adatainkat.

2.4.1. Miért fontos ez nekünk?

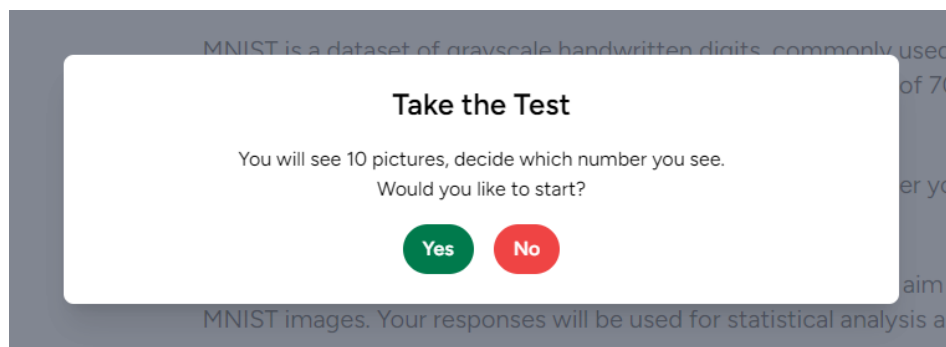
Ahogy már az **Adatvédelmi irányelvek** szakaszban leírtam, az oldal látogatóinak, valamint a teszt kitöltőinek egyedi *session*-t (munkamenetet) biztosítunk, így nincs szükség regisztrációra és bejelentkezésre. Ezzel az egyedi azonosítóval érzük el azt, hogy névtelenül, de valamennyire mégis megkülönböztethessük a látogatókat. Így amikor kitöltik a kérdőívet, teljesen anonim módon történik az adattárolás, de számunkra egy-egy plusz információt (metaadatot) tartalmaz minden válasz.

Biztosan sokan hallottuk már azt, hogy egy balkezesnek jobban megy valami, mint egy jobbkezesnek vagy éppen fordítva. Erről rengetegféle kutatás és cikk íródott. Egyesek szerint a balkezesek kreatívabbak, jobb sportolók, könnyebben mennek át a vezetési vizsgán, több pénzt kereshetnek, gyorsabb gépírók, könnyebben felépülnek a betegségekből, illetve nagyobb az önkontrolljuk.[15] Míg mások szerint ezek a jobbkezesekre jellemzőek.

Egy viszont biztos. Különbözőek vagyunk és akár a „kezességtől” függően is gondolkodhatunk másképp. Ezek a szempontok pedig új nézőpontot kínálnak számunkra és előrébb vihetek kutatásainkat, tanulmányainkat.

2.5. A kérdőív kitöltése

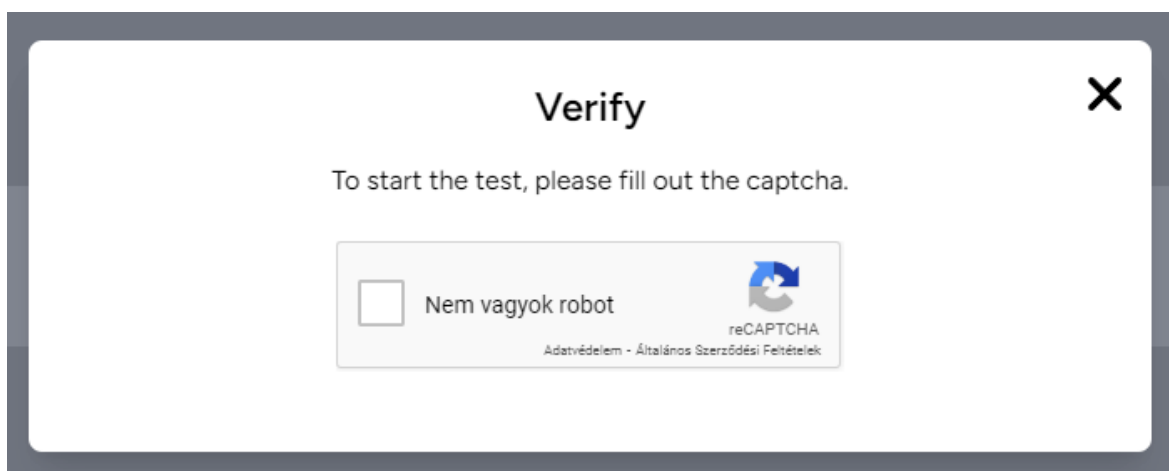
A kérdőív kitöltését a kezdőoldalon lévő „*Take the Test*” (Teszt indítása) gombbal kezdetjük el. A gombra kattintva egy előugró ablak tájékoztat minket, hogy 10 kép fog megjelenni, amelyekről el kell döntenünk, hogy melyik számot látjuk. A „Yes” gombot megnyomva már csak a biztonsági okokból történő CAPTCHA tesztet kell megoldanunk és kezdetjük is a kitöltést.



2.5. ábra. A „Take the Test” gomb kattintására előugró ablak

2.5.1. CAPTCHA

A CAPTCHA a webbiztonság egyik fontos eszköze. Ez a modul segít megkülönböztetni az emberi felhasználókat a gépi, számítógépes programoktól. Egy olyan technológia, amely a felhasználónak felad egy feladatot vagy kérdést, melyet csak az emberi intelligencia képes megoldani. Ez lehet például szöveg vagy képfelismerés, betűk vagy számok beírása, illetve egy képen lévő megfelelő elem(ek) kiválasztása. Használata megakadályozza az **automatizált számítógépes programokat** (botokat) abban, hogy a webhelyeket rosszindulatú vagy félrevezető tevékenységekre használják. Az én webalkalmazásom is az előbb felsorolt indokok miatt tartalmaz CAPTCHA-t.



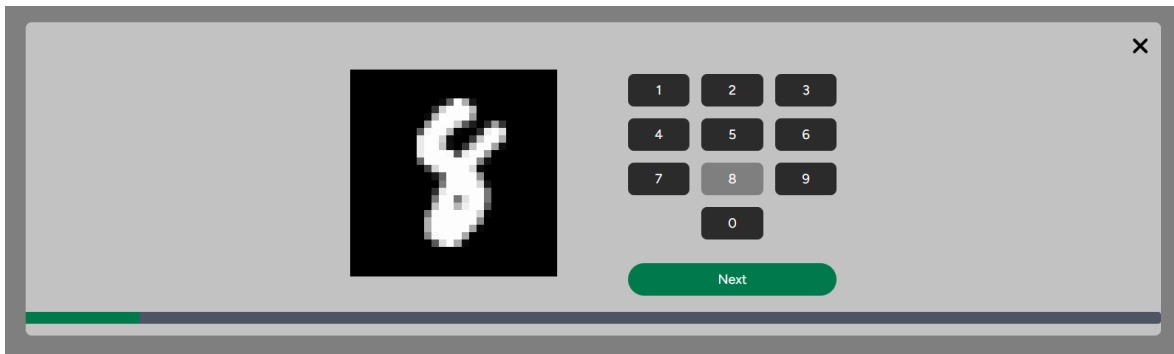
2.6. ábra. A kérdőív megkezdése előtt kitöltendő CAPTCHA

Valójában viszont ez nem egészen az a CAPTCHA amire gondolunk. Csak az egyszerűség kedvéért hivatkoztam arra, mert igazából amit a webalkalmazásomban használok az egy úgynevezett reCAPTCHA. Ez egy fejlettebb változata a CAPTCHA-nak, amelyet a Google fejlesztett ki. Nemcsak az emberi felhasználókat képes megkülönböztetni a gépi programoktól, hanem közben segíti a digitális archívumok és könyvek valós időben történő feldolgozását is. A reCAPTCHA gyakran olyan egyszerű feladatokat kínál fel, mint az utcaképeken található gépjárművek vagy épületek felismerése. Emellett folyamatosan tanul az emberi és a gépi viselkedési mintákból, hogy egyre hatékonyabban különböztesse meg a felhasználókat.[16]

Bővebben a reCAPTCHA-ról, a Fejlesztői dokumentáció **40.** oldalán írok. Addig is – az egyszerűség kedvéért – a későbbiekben csak CAPTCHA-ként fogok rá hivatkozni.

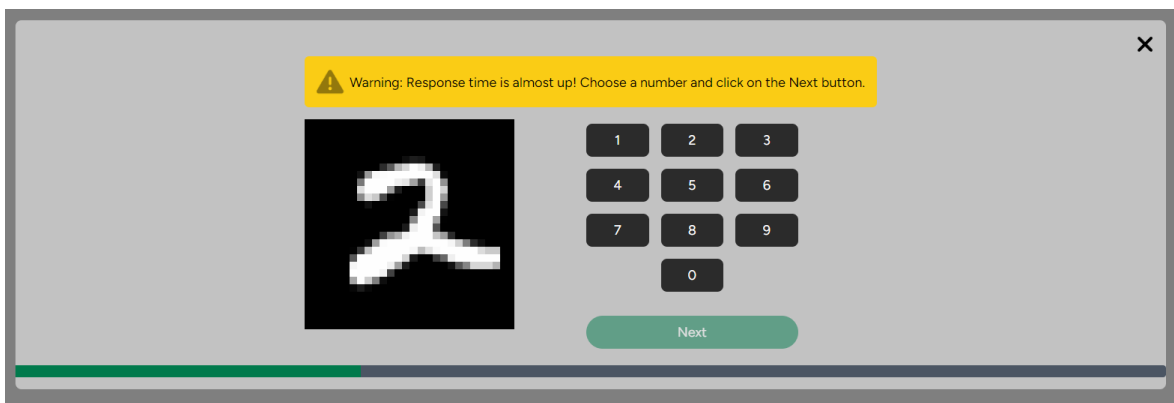
2.5.2. Számok azonosítása

Amint sikeresen teljesítettük a CAPTCHA-t, elindul a kérdőív teszt része. Itt egyesével 10 kép fog megjelenni, amelyekről el kell döntenünk, hogy melyik számot látjuk. A megfelelő szám kiválasztása után a „Next” gombra kattintva töltjük be a következő képet. A képek alatti „progress bar”-on (előrehaladási sávon) láthatjuk a kérdőív folyamatát.



2.7. ábra. A kitöltés képernyőképe

A számok azonosítására képenként 30 másodpercünk van. Az első 25 másodperc leteltével egy „warning” (figyelmeztető) jelzés jelenik meg a képernyőn, ami arról tájékoztat, hogy nem sok időnk maradt, így válasszunk egy számot. Amennyiben letelt az idő, úgy automatikusan a következő képre lép anélkül, hogy a válaszunkat elmentette volna.



2.8. ábra. „Warning” üzenet 25 másodperc elteltével

A kérdőív megszakítására bármikor van lehetőség. A teszt jobb felső sarkában lévő „X” gombra kattintva megnyílik egy párbeszéd ablak, amely során eldönthetjük, hogy folytatni szeretnénk-e a kérdőív kitöltését vagy ténylegesen befejezni. Az addigi válaszaink automatikusan feltöltődnek, így az adatok nem vesznek el.

2.6. Visszajelzés küldése

A webalkalmazásomban fontosnak tartottam a visszajelzés, angolul „feedback” funkciót is. A 10. kép azonosításakor – miután a kérdőívet befejeztük – lehetőségünk van visszajelzést küldeni a fejlesztők és a kutatók számára. Véleményem szerint a visszajelzés biztosítása emeli az alkalmazás színvonalát. Számtalan előnye lehet annak, ha a tesztet kitöltők és az oldal üzemeltetői „közvetlenül” elérik egymást. A visszajelzés funkció fő előnyeinek a következőket gondolom;

– **elégedettség növelése:**

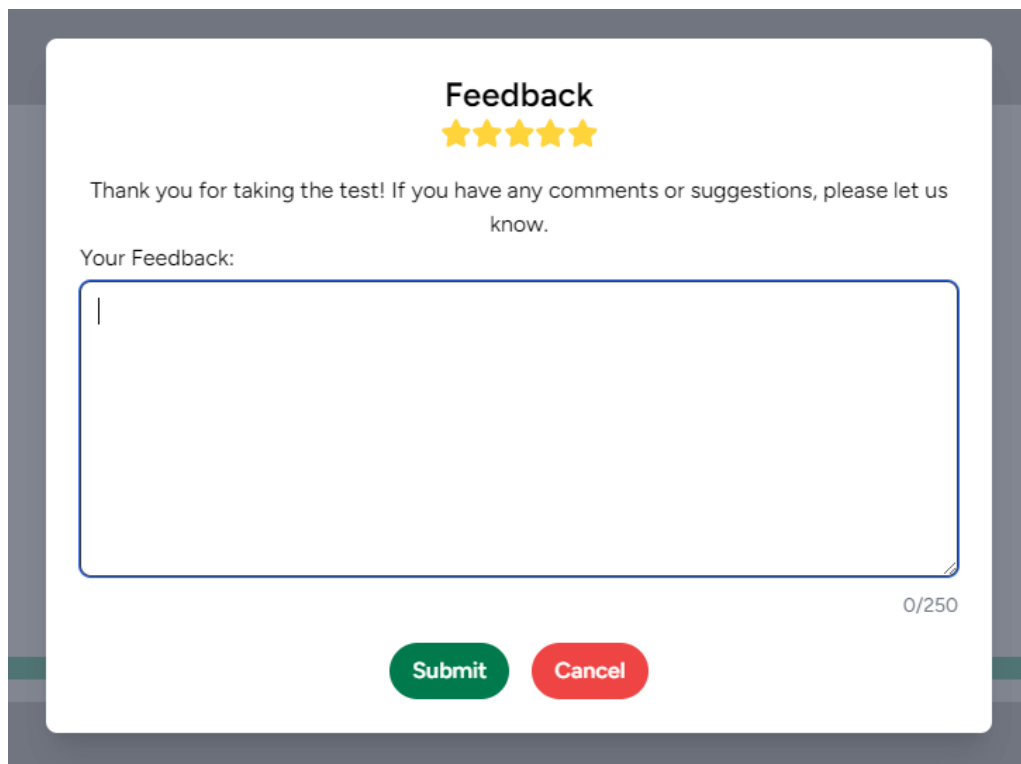
amikor a felhasználóknak – esetemben teszt kitöltőinek – lehetőségük van visszajelzést küldeni a webalkalmazásról, az megmutatja számukra, hogy fontosak és figyelembe veszik a véleményüket. Ezáltal javul az elégedettségük és hajlandóbbak lesznek tartósabb ideig használni az alkalmazást.

– **hibák és problémák azonosítása:**

a visszajelzések révén könnyebben azonosíthatók a felhasználói élményt negatívan befolyásoló hibák és problémák. Ez lehetővé teszi a fejlesztők számára, hogy gyorsan reagáljanak ezekre és javítsák az alkalmazás minőségét.

– **funkciók és tartalom fejlesztése:**

segíthetnek olyan új funkciók vagy tartalmi elemek létrejöttéhez, amelyeket a felhasználók szeretnének látni vagy használni az alkalmazásban. Ez lehetővé teszi a fejlesztők számára, hogy az oldalt használók igényei szerint alakítsák az alkalmazást.



The image shows a feedback form interface. At the top, the word "Feedback" is centered in bold, with five yellow stars below it. Below the stars, a message reads: "Thank you for taking the test! If you have any comments or suggestions, please let us know." Underneath this is the label "Your Feedback:" followed by a large, empty text input field with a blue border. In the bottom right corner of the input field, the text "0/250" indicates the character count. At the bottom of the form, there are two buttons: a green "Submit" button and a red "Cancel" button.

2.9. ábra. A visszajelzés küldésének képernyőképe

Az itt beküldött visszajelzéseket legfeljebb 250 karakterben, szigorú szűrésekkel fogadjuk el. A szöveg nem tartalmazhat linkeket és különböző „szkripteket”, amelyek az oldal biztonságát befolyásolhatják. Ezt valós időben érzékeljük és figyelmeztetjük a felhasználót. Ennek működéséről bővebben a Fejlesztői dokumentáció **41.** oldalán olvashat.

2.7. A képgenerálás módjai

A felhasználói dokumentáció ezen szakaszában más szerepkörből nyerünk betekintést a webalkalmazásba. Ez a nézet – illetve funkció – az oldal „karbantartójához”, adminisztrátorához tartozik.

Az adminisztrátori oldallal a szakdolgozati társam dolgozott. Azonban ez a rész az én általam bemutatott kérdőív működésének alapja, ezért ennek működését én valósítottam meg.

Az oldalon tehát van egy bejelentkezési felület, amely segítségével elérhetővé válik az adminisztrátori nézet. A sikeres bejelentkezés után a navigációs sávon az „Image Generation” fül megnyitásával, a következő beállítási lehetőség tárul elénk:

Random Balancing Most Often Misidentified

Selects completely randomly. Image weighting is ignored. Speed: ●●●

Selects randomly from the least generated images based on the weights of previously generated images. Users won't receive the same image twice within an hour. Speed: ●●●

Selects randomly from the images that were most often misidentified based on their weights. Users won't receive the same image twice within an hour. Speed: ●●●●

☒ From Train dataset ☒ From Test dataset

Save

More ● = Faster

2.10. ábra. A képgenerálás módjának beállítása az adminisztrátori nézeten

Ez az irányítópult a kérdőívben szereplő képek megjelenéséért felelős. A képek ugyanis mindig egy bizonyos feltétel szerint választódnak ki az adatbázisból. Minden képhez tartoznak súlyok, melyeket generálásukkor veszünk figyelembe. Két fő szempontot (súlyt) különböztetünk meg:

- az eddigi generálások számát
- a félreazonosítások számát

Ez azt jelenti, hogy több módszerünk is van, melyek alapján a képeket választjuk és küldjük a kérdőívet kitöltők felé. Fontos azonban megjegyezni, hogy mindhárom véletlenszerűen választ. Az utolsó két módszernél viszont a súlyokat előtérbe helyezzük.

2.7.1. Véletlenszerű

Véletlenszerűen választ. A képek súlyozása figyelmen kívül maradnak.

Ez azt jelenti, hogy akár egy bizonyos kérdőívben belül is előfordulhat, hogy ugyanazt a képet látja majd a tesztkitöltő. Erre azonban elég kevés az esély, hiszen 60.000 „train” (tanító) és 10.000 „test” (tesztelő) képet tartalmaz a teljes adatbázis.

Meglátásom szerint ez az opció az alkalmazás életciklusának elején a leghatékonyabb. Olyankor, amikor még nincsen sok válaszuk, így kevés képhez tartozik még súly és nem biztos, hogy fontos számunkra a megkülönböztetés.

2.7.2. Kiegyensúlyozott képválasztás

Véletlenszerűen választ a legkevesebbet generált képek súlyai alapján. A kitöltők egy órán belül nem kaphatják meg kétszer ugyanazt a képet.

A második módszer kiegyensúlyozza a már sokszor generált képek és a ritkán vagy egyáltalán nem generált képek közötti eltérést. Ez segíthet abban, hogy egy bizonyos válaszmennyiség után minden képre hasonló számú referenciánk legyen és az ezt követő kutatások megbízható forrásból származzanak.

Az ismétlődő képekre vonatkozó időkorlát bevezetése azért volt fontos, mert ha az alkalmazás azon szakaszában használjuk ezt a módszert, amikor még nincsen sok adatunk, a tesztkitöltők gyakran ugyanazokat a képeket látnák.

2.7.3. A leggyakrabban félreazonosított képek

Véletlenszerűen választ a leggyakrabban félreazonosított képek súlyai alapján. A kitöltők egy órán belül nem kaphatják meg kétszer ugyanazt a képet.

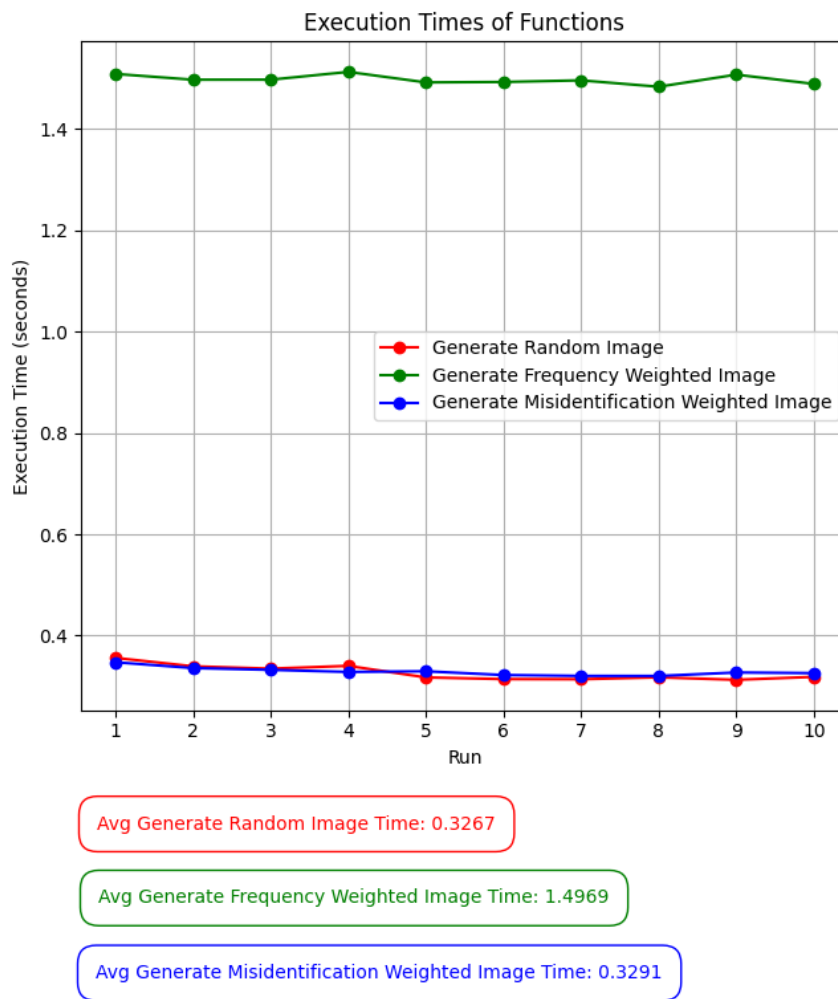
A harmadik és egyben utolsó módszer, a leggyakrabban félreazonosított képeket helyezi előtérbe. Ez azt jelenti, hogy a kitöltők azokat a képeket kapják meg nagy valószínűséggel, amelyeket már sokan félreazonosítottak. Ennek segítségével ténylegesen kideríthetjük, hogy melyik képek vagy számok azok, amelyek felismerése még az ember számára is nehézséget okoznak. Az időkorlát bevezetése itt is ugyanazt a célt szolgálja.

2.7.4. További szempontok

Az oldal üzemeltetőjének nemcsak a „súlyozott random” szempont lehet az egyetlen. Fontos kérdés még a generálási idő is. Ezeket jelzem az egyes módszerek alatt. A generálási idő módszerenként eltérő. Nagyjából 0.3 és 1.5 másodperc között van képenként. Az, hogy melyik módszer mi alapján gyorsabb, nagyon sok mindentől függ. Többek között ilyen az, hogy az alkalmazás melyik életciklusában használjuk. Ilyenkor vizsgálnunk kell, hogy a hatékonyságot mérlegelve melyik számunkra a leghasznosabb. A generálási időt (negatívan vagy pozitívan) az alábbi dolgok befolyásolják:

- a képekre beérkező azonosítások mennyisége
- a félreazonosított képek mennyisége

Végül pedig minden módszernél kiválasztható, hogy a teljes adathalmazból vagy csak egy részéből szeretnénk generálni a képeket. Azaz, ha a tanító és tesztelő adathalmazt választjuk ki, úgy mind a 70.000 kép szóba jöhet a generáláskor. Ezt azonban lecsökkenthetjük úgy, ha csak az egyiket jelöljük ki. Ezzel is azt a célt szolgáljuk, hogy az üzemeltető vagy éppen a kutatást végző, saját szempontja alapján szabhassa személyre igényeit.



2.11. ábra. A három képgeneráló függvény futási ideje [17]

A fent látható vonaldiagramon az látszik, ahogyan a teljes adathalmazból 10 képet generáltam mindhárom módszerrel. A véletlenszerű és a leggyakrabban félreazonosított képek generálási ideje nagyjából megegyezik. A kiegyensúlyozott képgenerálás azonban ötször több időt vett igénybe. Ez nem meglepő, hiszen az alkalmazás azon szakaszában teszteltem, amikor mindössze pár kép volt azonosítva. Ez azt jelenti, hogy azokat a képeket is ki kellett listázni, amelyek még egyáltalán nem fordultak elő. Esetünkben ez közel 70.000 kép volt. Így talán már jobban érthető, hogy hogyan függ össze a képgenerálás ideje azzal, hogy milyen életciklusában van a webalkalmazás.

3. fejezet

Fejlesztői dokumentáció

Ebben a fejezetben csoportosítom, illetve ismertetem a webalkalmazásomhoz használt különböző technológiákat. Részletesen bemutatom a tervezés, valamint a fejlesztés folyamatát. Az egyes lépéseken keresztül, megpróbálom úgy átadni a megvalósított funkciókat, hogy először megfogalmazom az igényt/problémát, majd mérlegelek a lehetőségeim között, végül pedig bemutatom a megoldásom.

3.1. Felhasznált technológiák

3.1.1. Tervezéshez használt eszközök

- PlantUML
- dbdiagram.io
- uizard

3.1.2. Webfejlesztéshez használt környezetek és eszközök

Szerveroldali környezet (backend)

- Composer
- Laravel (PHP)
- XAMPP (Apache, MySQL)
- Python
- Node.js

Kliensoldali keretrendszerek és könyvtárak (frontend)

- React
- Inertia.js

További felhasznált könyvtárak, csomagok és modulok

- react-google-recaptcha
- dompurify
- uuid
- matplotlib (Python)
- python-mnist
- keras
- tensorflow
- mysql-connector-python

3.1.3. Tesztelési környezet

- Postman
- Cypress

3.1.4. Verzió- és feladatkövetés

- GitHub
- GitHub Desktop

3.2. Tervezés

3.2.1. Tervezési szempontok

A tervezés lehet a siker kulcsa, különösen a bonyolultabb projekteknél és feladatoknál. Egy jó tervezés segít megérteni az elvárásokat, meghatározni a célokat és strukturálni a feladatokat. Ezáltal csökkenti a kockázatokat, optimalizálja az erőforrásokat és elősegíti a hatékony munkát. Röviden tehát, a tervezés a biztos alap, amelyre építhetjük a sikeres projektünket és megvalósításainkat.

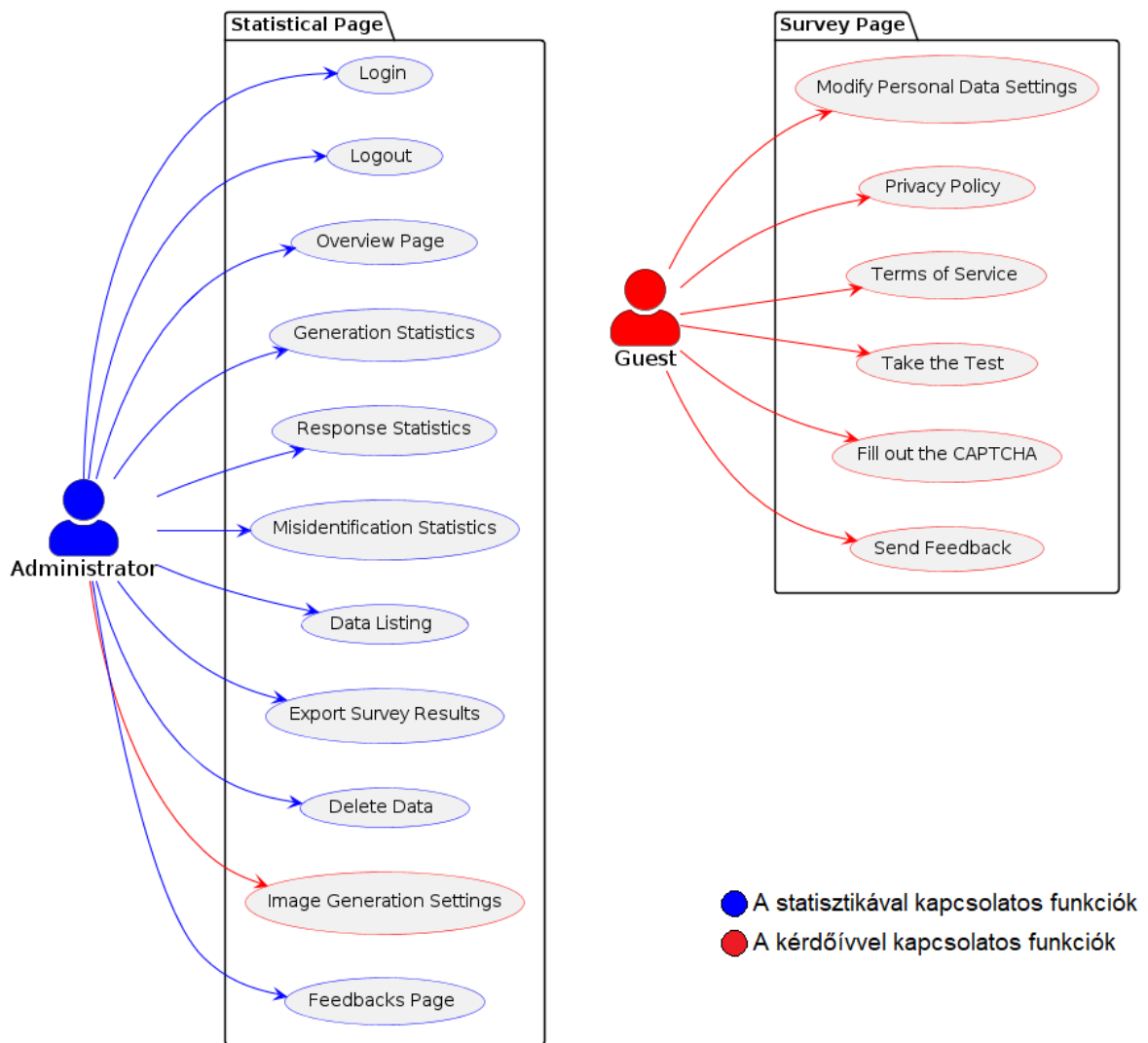
Az alkalmazásom elsőre talán egyszerűnek tűnik. Egy kérdőív, ahol az emberek eldönthetik, hogy melyik számot látják. Azonban, ha részletesebben belegondolunk az egyes feladatokba, ennek kivitelezése mégsem annyira egyértelmű. Rengeteg szempont volt, amire a tervezéskor nem is gondoltam. Éppen ezért megpróbáltam úgy megtervezni és felépíteni, hogy az alkalmazás mindig bővíthető legyen.

Ahogy már korábban is írtam, az alkalmazás két részre osztható. A kérdőíven és látogatói oldalon keresztül történő adatgyűjtés, illetve az adatok feldolgozásához és a statisztikák/eredmények kimutatásához szolgáló adminisztrátori nézet. Az adatfeldolgozással kapcsolatos teendőket a szakdolgozati társam végezte el. Az egyszerűbb megértés érdekében, a tervezéshez készült diagramokon mindkét rész megjelenik. A feladatok elkülönítéséhez piros és kék színt használunk. A piros az én, a kék pedig az Ő munkáját jelzi.

3.2.2. Használati eset diagram

A használati eset alapú tervezés megközelítése az informatika területén a felhasználók és rendszerek interakcióinak leírására szolgál. Az idők során számos tervezési módszer létezett, amelyek középpontjában különböző elemek (pl. adatfolyam, adatbázis-kezelés vagy felhasználói felület) álltak. Azonban a használati esetekkel történő tervezés elterjedésével elfogadottá vált az, hogy a középpontba, a felhasználók és rendszerek viszonyát helyezzük.

A használati esetek leírása egyszerű és könnyen érthető módszer arra, hogy megmutassuk, mely szereplők milyen funkciókat használhatnak. Ez segít elkerülni a félreértéseket és pontosan megérteni a megrendelő igényeit. A használati esetek diagramjai könnyen értelmezhetőek mind a megrendelők, mind a tervezők számára. Továbbá segítenek az összes fontos funkció és szereplő azonosításában. Ezáltal a használati esetek egyszerűsége és érthetősége lehetővé teszi, hogy hatékonyan kommunikáljunk a projektről és biztosítsuk a megfelelő megértést mind a megrendelő, mind pedig a tervező részéről.[18]



3.1. ábra. Az alkalmazás tervezéséhez készített „Használati eset diagram” [19]

Az ábrán tehát jól látszik, hogy hogyan különülnek el az egyes szerepek. Ahogy már azt írtam, az én általam készített alkalmazás a „*Guest*” (vendég) szempontjából valósul meg. Ide tartozik a;

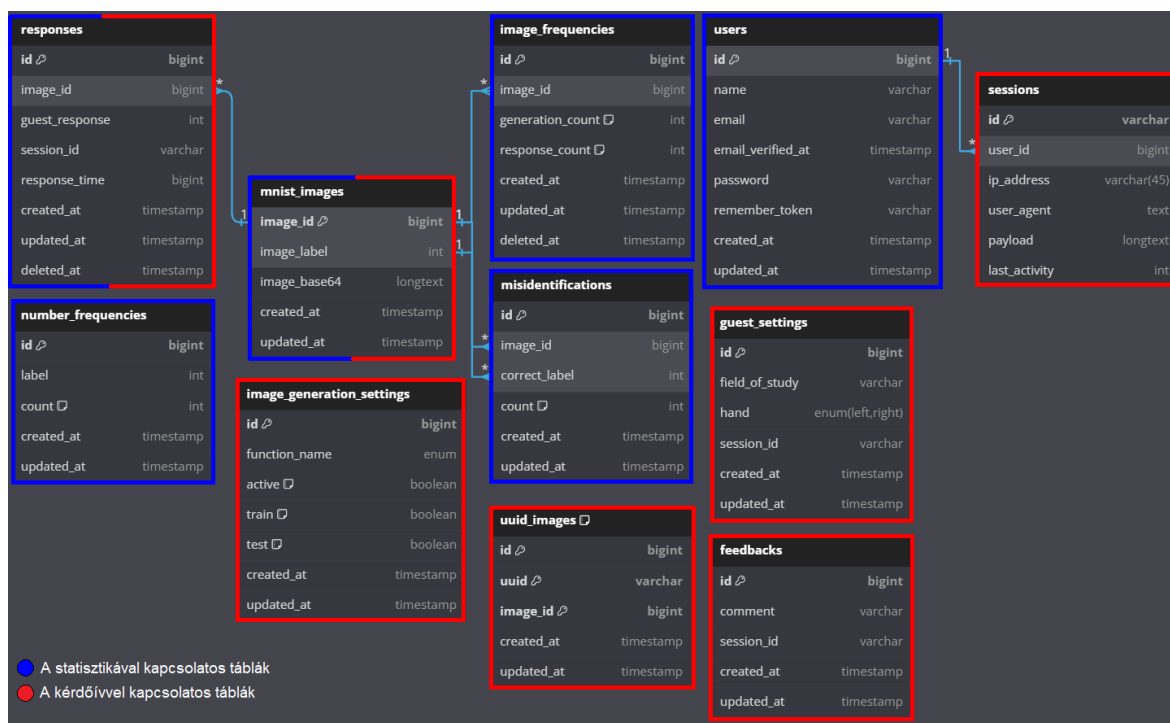
- személyes adatok beállítása
- az adatvédelmi irányelvek oldala
- az általános szerződési feltételekért felelős oldala
- a CAPTCHA integrálása
- a kérdőív kitöltése
- a visszajelzés küldése

Adminisztrátori oldalról pedig a képgenerálás módjának beállítása.

3.2.3. Adatbázis

Az adatbázis sémák készítése, a tervezésének egy kulcsfontosságú lépése, amelynek célja az adatok struktúrájának és kapcsolatainak meghatározása. Olyan tervrajzok, amelyek leírják az adatok tárolásának és szervezésének módját. Segítenek meghatározni, hogy milyen adatokat tárolunk, milyen formában és hogyan vannak ezek kapcsolatban egymással. Ez magában foglalja az adatok típusát, attribútumait, kapcsolatait és egyéb fontos jellemzőiket.

Összességében az adatbázis sémák a tervezés során segítenek abban, hogy strukturált és jól szervezett adatbázist hozzunk létre, amely megfelel az alkalmazás igényeinek és biztosítja az adatok integritását² és konzisztenciáját. Ezáltal elősegítik a hatékony adatkezelést és hozzájárulnak az alkalmazás sikeres működéséhez.



3.2. ábra. Az alkalmazás tervezéséhez készített adatbázis séma [20]

Az adatbázis sémában szereplő táblák és mezők strukturált formában reprezentálják az alkalmazás által kezelt adatokat. A projekt során azonban számos tervezési döntést kellett meghozni annak érdekében, hogy az alkalmazás megfelelően működjön. Ezek közül a legszembetűnőbbet szeretném bemutatni.

² „Az adatintegritás biztosítja, hogy az adatok helyesek, érvényesek és összhangban vannak a meghatározott szabályokkal és korlátozásokkal.”[22]

A munkamenetek kapcsolata

Az adatbázis sémából egyértelműen látható, hogy a *users* és *sessions* táblák közötti kapcsolat a *users.id* és *sessions.user_id* mezők között van. Ez egy „1-N” típusú kapcsolatot jelent, ahol egy felhasználóhoz több munkamenet is tartozhat, például egyidejűleg több eszközről történő bejelentkezés esetén. Azonban egy munkamenethez csak egy felhasználó tartozhat. Fontos megjegyezni, hogy ez kizárólag az adminisztrátori jogosultsággal rendelkező felhasználókra vonatkozik és a biztonság érdekében az ő munkamenetük minden bejelentkezéskor és kijelentkezéskor változik.

Mi a helyzet a vendégekkel? Az adatbázis terven látható, hogy három olyan táblát is létrehoztunk, amelyekben kapcsolat nélküli munkamenet-azonosítókat tárolunk: a *responses*, a *feedbacks* és a *guest_settings* táblák. Ennek oka, hogy ezek a tevékenységek nem igénylik a felhasználói fiókot, így a válaszok, visszajelzések és személyes beállítások nem kapcsolódnak közvetlenül a felhasználóhoz. Ennek ellenére mégis szükség van azonosítókra ezekben a táblákban is. Az itt lévő kapcsolatok hiánya, hogy a látogatók munkamenet-azonosítói is változnak (1 év), viszont ebben az esetben a kapcsolat fenntartása sértené az adatintegritást és megnehezítené az adatbázis-kezelést. Ezért egyszerűen csak az adott művelethez mentjük el a látogató munkamenet-azonosítóját és nincs kifejezett kapcsolat a *sessions* táblával.

3.2.4. Képernyőképek („mockup”)

A képernyőképek tervezése, más néven „*mockup*” készítése az alkalmazás vagy weboldal vizuális tervezésének fontos lépése. A „*mockup*”-ok olyan statikus képek vagy vázlatok, amelyek megjelenítik az alkalmazás vagy weboldal tervezett felhasználói felületét és funkcióit. Ezek a tervek általánosságban a következőket segítik elő;

- **vizualitás:**

segít a megjelenítés és az elrendezés tervezésében. Megmutatják, hogy hogyan néz majd ki a végső termék, és segítenek eldönteni, hogy milyen színek, betűtípusok és egyéb vizuális elemek legyenek jelen az alkalmazásban.

- **felhasználói élmény:**

lehetővé teszik az interaktív elemek, funkciók és felhasználói utak tervezését. Ezáltal segítenek kialakítani egy intuitív és hatékony felhasználói élményt a weboldal számára.

A uizard[21] segítségével könnyedén és gyorsan készíthettem interaktív és testre szabott felhasználói felületet a webalkalmazásom számára. A „redundancia” elkerülése érdekében az oldalon elkészített terveket nem illeszttem be ide, mert nagyon hasonló

lettek a végleges megvalósításhoz. Az oldal képernyőképei a **Felhasználói dokumentáció** 14. oldalán tekinthetők meg.

3.3. Fejlesztés

Ebben a szakaszban ismertetem a különböző funkciók megvalósítását. Ahogy azt már korábban is írtam, az egyes funkcióknál először körülírom a megrendelő igényét (megoldandó problémát), majd felsorolom a lehetőségeket, végül pedig bemutatom a megoldásomat.

3.3.1. A keretrendszer kiválasztása

Laravel

Az egyértelmű, hogy a szakdolgozati alkalmazásomhoz valamilyen webkeretrendszer szükséges. Az egyetemi tanulmányaim alatt több ilyenrel is megismerkedtem. Azonban a legmeghatározóbb számomra a Laravel volt. Ez volt az első, amelyet kipróbáltam és hosszú távon használtam. De miért pont a Laravel?

Egy új technológia megtanulása hosszú és fáradtságos folyamat. Azonban a Laravel széleskörű közössége és logikusan felépített dokumentációja nagyban megkönnyítette ezt. Mindig volt hová nyúlnom, ha elakadtam. Véleményem szerint ez az a tulajdonsága, ami a legtöbb embert többek között engem is megfogott.

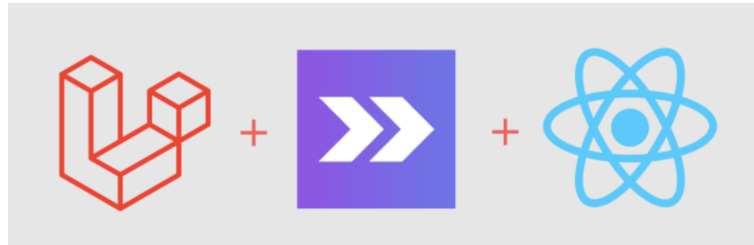
Természetesen a tanulási fázis folyamán voltak olyan nagyon hasznos tulajdonságai, amelyekre csak utólag figyeltem fel. Ez is azt támasztotta alá bennem, hogy jól döntöttem. Többek között gondolok itt a számos beépített funkciók kínálatáról, mint pl. az autentikáció, az adatbázis-műveletek hatékony kezelése és az útvonalak definíciója. Végül, de nem utolsó sorban amely a legmeghatározóbb volt számomra, az MVC architektúra. Ez biztosítja a kód elkülönítését és a fejlesztés strukturáltabbá tételét. Ezáltal könnyen karbantartható, bővíthető és tesztelhető alkalmazást hozhatunk létre.

React + Inertia.js

A webalkalmazás felhasználói felületének kialakításához és dinamikus működtetéséhez két ismert technológia, a React és az Inertia.js kombinációját választottam.

A React egy rendkívül hatékony JavaScript könyvtár, amely lehetővé teszi a dinamikus és interaktív felhasználói felületek egyszerű létrehozását. A komponensalapú megközelítés támogatja az egyes részek újrafelhasználhatóságát és a könnyű karbantarthatóságot. Emellett az egyik legnépszerűbb megoldás a frontend fejlesztésben, így támogatottsága és dokumentációja megkönnyíti a tanulást és a használatát.

Az Inertia.js egy olyan könyvtár, amely lehetővé teszi a Laravel és a React együttműködését úgy, hogy közben a szerver és a kliens közötti kommunikációt minimalizálja. Ennek eredményeként az alkalmazás SPA jellegűvé válik, ami gyorsabb betöltési időket és simább felhasználói élményt eredményez. Az Inertia.js használata lehetővé teszi a Laravel alkalmazásunkba való könnyű integrációját, miközben megőrzi a React erejét és rugalmasságát.



3.3. ábra. Laravel + Inertia.js + React [23]

A Laravel, React és az Inertia.js kombinációja lehetővé teszi a felhasználók számára, hogy gyorsan és hatékonyan navigáljanak az alkalmazásunkban, miközben egyszerűsíti a fejlesztési folyamatot és javítja az alkalmazás teljesítményét.[23]

3.3.2. Az MNIST adatbázis integrációja

Az adatbázis megtervezve, a keretrendszer kiválasztva és inicializálva. Következő lépés az MNIST adatbázis képeinek integrációja. Erről már írtam a **15.** oldalon, de most kicsit részletesebben mutatom be.

Az egyszerűbb adatfeldolgozás érdekében a képeket előre tároljuk az adatbázisban. Ahogy már korábban említettem az adatkészlet 70.000 képet tartalmaz (60.000 tanításra és 10.000 tesztelésre). Minden kép egyedi azonosítóval és címkével rendelkezik. A címke adja meg, hogy pontosan melyik szám van a képen. Ez az adathalmaz egy python szkript segítségével, valamint a *Tensorflow* és *Keras* könyvtárak használatával könnyedén generálható. Azonban ezt az eljárást a gépi tanulás folyamatához találták ki. Ez azt jelenti, hogy a képek mindössze 28x28 pixel méretűek, nem személyre szabottak és a létrejött adatokat külön még sehol nem tároljuk.

Mindent egybevéve tehát, úgy kell generálnom a képeket, hogy azokat az alkalmazásomban fel tudjam használni. A képeket emberi szemmel láthatóra, de a minőségét megőrizve kell elmentenem az adatbázisba.

Python script

A Laravel keretrendszer lehetőséget biztosít a python szkriptek használatára. A `./storage` könyvtárban létrehoztam egy külön, a python szkriptek számára fenntartott „*scripts*” mappát. Innen indítom az `mnist_images_init.py` szkriptet, ami a generálást

és az adatbázisba való mentést végzi. Ezt a folyamatot az alkalmazásunk **telepítésének** elején kell csak elvégezni. Ez biztosítja azt, hogy a kérdőív megfelelően tudjon működni.

3.1. kódrészlet. Importok a szkript működéséhez

```
1 import mysql.connector
2 import base64
3 import matplotlib.pyplot as plt
4 from keras.datasets import mnist
5 from io import BytesIO
6 from datetime import datetime
```

- **mysql.connector:**
adatbázis kapcsolódásra használt modul.
- **base64:**
bináris adatok (képek) Base64 kódolására használt modul.
- **matplotlib.pyplot:**
képek megjelenítésére, formázására és grafikonok készítésére használt modul.
- **keras.datasets.mnist:**
MNIST adatbázis betöltésére használt modul a Keras könyvtárból.
- **io.BytesIO:**
bináris adatok tárolására és kezelésére használt modul.
- **datetime.datetime:**
időbélyeg létrehozására használt modul a Python alapkönyvtárból.

3.2. kódrészlet. Adatbázis kapcsolat és az adathalmaz betöltése

```
1 mydb = mysql.connector.connect(
2     host="localhost",
3     user="root",
4     password="",
5     database="mnist_validate_by_human"
6 )
7 mycursor = mydb.cursor()
8
9 # Adatkészlet betöltése a kerasból
10 (train_X, train_y), (test_X, test_y) = mnist.load_data()
11
12 # Plot inicializálása fehér háttér nélkül, 280x280-as méretben
13 fig, ax = plt.subplots(figsize=(2.8, 2.8), dpi=130)
```

3.3. kódrészlet. Az első 60.000 kép feldolgozása és mentése az adatbázisba

```
1  # Az első 60000 kép mentése az adatbázisba (tanító képek)
2  for i in range(60000):
3      image = train_X[i]
4      label = int(train_y[i])
5
6      # Az előző ax felszabadítása
7      ax.clear()
8
9      # Plot átlátszó háttérrel
10     plt.imshow(image, cmap='gray')
11     plt.axis('off')
12
13     # A plot mentése pufferbe PNG-ként
14     buffer = BytesIO()
15     plt.savefig(buffer, format='png', transparent=True,
16                 ↪ bbox_inches='tight', pad_inches=0)
17     buffer.seek(0)
18
19     # A kép konvertálása base64 formátumba
20     image_base64 = base64.b64encode(buffer.getvalue()).decode(
21         ↪ 'utf-8')
22
23     now = datetime.now().strftime('%Y-%m-%d_%H:%M:%S')
24     sql = "INSERT INTO mnist_images (image_id, image_label,
25         ↪ image_base64, created_at, updated_at) VALUES (%s, %s,
26         ↪ s, %s, %s, %s)"
27
28     val = (i, label, image_base64, now, now) # A tesztelő
29     ↪ képeknél az i-t majd eltoljuk 60.000-el
30     mycursor.execute(sql, val)
31     mydb.commit()
32
33     print("Inserted train image with index:", i)
34
35     ...
36
37 mycursor.close()
38 mydb.close()
```

Amint látható, először a tanításhoz használt képeket dolgozom fel és töltöm be az adatbázisba, majd a teszteléshez használt képeket. Az utóbbi kódrészletét már nem illesztettem be, mert nagyon hasonlít az első módszerre. Az eltérés mindössze annyi, hogy ott az indexet el kell tolni 60.000-el, mert alapvetően a Kerasból 0-tól 9999-ig tartalmazza a 10.000 tesztelő rekordot. Összesen tehát a 70.000 kép eloszlása a következő módon néz ki:

- Train (tanító) képek azonosítói: 0 - 59999
- Test (tesztelő) képek azonosítói: 60000 - 69999

A python szkript teljes lefutási ideje az én számítógémemen nagyjából 12-15 percet vesz igénybe. Az egyik legjelentősebb része a plot inicializálása (**plt.subplots()**) és a memória felszabadítása (**ax.clear()**). Ezzel ugyanis azt érjük el, hogy nem kell minden alkalommal új plot objektumot létrehozni és felszabadítjuk a memóriát a korábbi plot objektumoktól. Így a kód mérete csökken és gyorsabb futási időt eredményez.

3.3.3. Az oldalak működése – HTTP kérések kezelése

A különböző nézeteknél – a React egyik sajátosságából adódóan – lehetőségünk van a *JavaScript* és HTML kódokat „egybeolvasztva” használni. Az így keletkezett fájlok a *.jsx* kiterjesztést (JavaScript XML) nevet kapták. Ez a fajta használat megkönnyíti a kód strukturálását és olvashatóságát. Az alkalmazásomban ilyen fájlokban dolgoztam.

Kezdőoldal

A kezdőoldal betöltésekor – a tartalom megjelenítésén kívül – két háttérfolyamatot kellett megvalósítanom:

- betölteni az eddigi képazonosítások számát
- megvizsgálni az aktuális látogatóhoz tartozó munkamenet-azonosítót

Az oldal betöltését a React által kínált *useEffect()* „Hook” érzékeli. Itt hívom meg azokat függvényeket és állapotváltozókat, amelyekre szükségem van a betöltéskor.

3.4. kódrészlet. useEffect() Hook használata

```
1  useEffect(() => {
2      // Azonosítások számának lekérdezése
3      fetchIdentificationsCount();
4
5      // Ellenőrzi a munkamenetet, ha a beállítások felugró
        ↪ ablaka zárva van
6      if (!settingsOpen) {
7          checkSession();
8      }
9  }, [settingsOpen]);
```

Mindkét művelet egy adatbázis táblából való lekérdezés eredményét használja fel. Ezeknek a megvalósítására két különböző, ismert módszert alkalmaztam. A képzazonosítások számát a „backendről” lekérve a *Fetch API* segítségével oldottam meg.

3.5. kódrészlet. Fetch API használata

```
1  const fetchIdentificationsCount = async () => {
2    try {
3      const response = await fetch('/api/identifications/
      ↪ count');
4      const data = await response.json();
5      setIdentificationsCount(data.count);
6    } catch (error) {
7      console.error('Error fetching identifications count
      ↪ :', error);
8    }
9  };
```

A másik feladat, a munkamenet-azonosító vizsgálata azt a célt szolgálja, hogy amennyiben új látogatóról van szó, úgy a **Személyes adatok beállítása** ikonon megjelenítjük a figyelmeztető jelzést. Ennek vizsgálatához az Axios könyvtár által kínált *axios.get()* metódust használtam.

3.6. kódrészlet. Axios használata

```
1  const checkSession = async () => {
2    try {
3      const response = await axios.get('/api/check-
      ↪ session-in-guest-settings');
4      setShowSettingsWarning(!response.data.exists);
5
6      if (response.data.exists) {
7        setExistingRecord(response.data.record);
8      }
9    } catch (error) {
10     console.error('Error checking session in guest
      ↪ settings:', error);
11   }
12 };
```

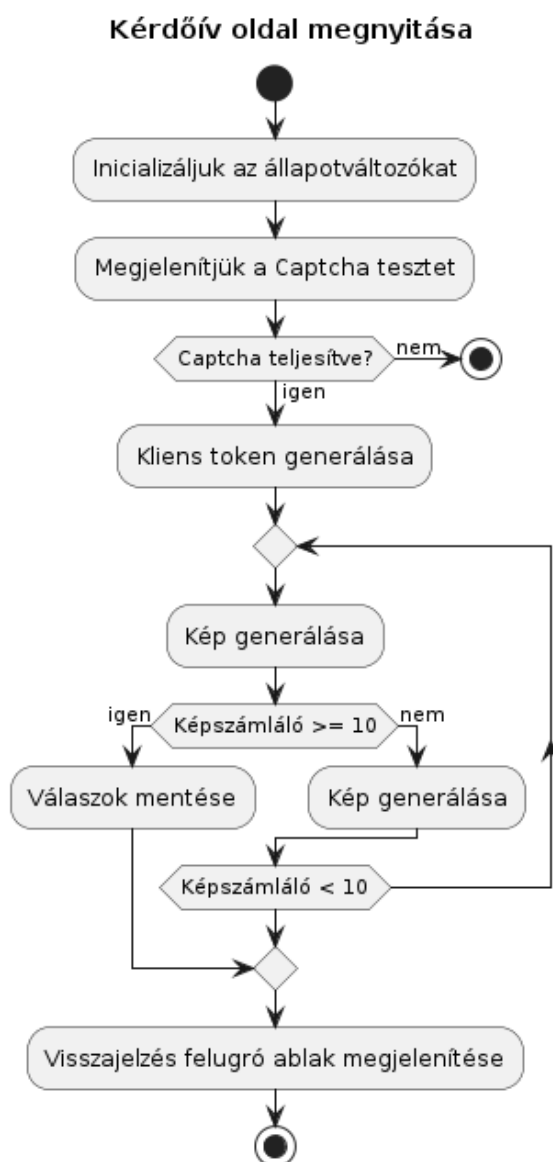
Jól lehet látni, hogy mindkét *get* kérés esetén az adatok lekérdezése majdnem megegyezik. Ilyen és ezekhez hasonló kérések és küldések háttérfolyamataiból épülnek fel a különböző oldalak és funkciók. Így, hogy láthatjuk a HTTP kérések feldolgozását, a későbbiekben ezekre a módszerekre fogok hivatkozni. A kétféle technika közül, – az Axios könyvtár rugalmasságának köszönhetően – az alkalmazásom többi részében, az axios modul által kínált megvalósításokat választottam.

Adatvédelmi irányelvek és Általános Szerződési Feltételek

Az **Adatvédelmi irányelvek** és az **Általános Szerződési Feltételek** oldala statikus módon készültek el. Ezek tartalma már előre elkészítettek és nem változnak dinamikusán a felhasználói interakciók vagy a szerveroldali adatok hatására.

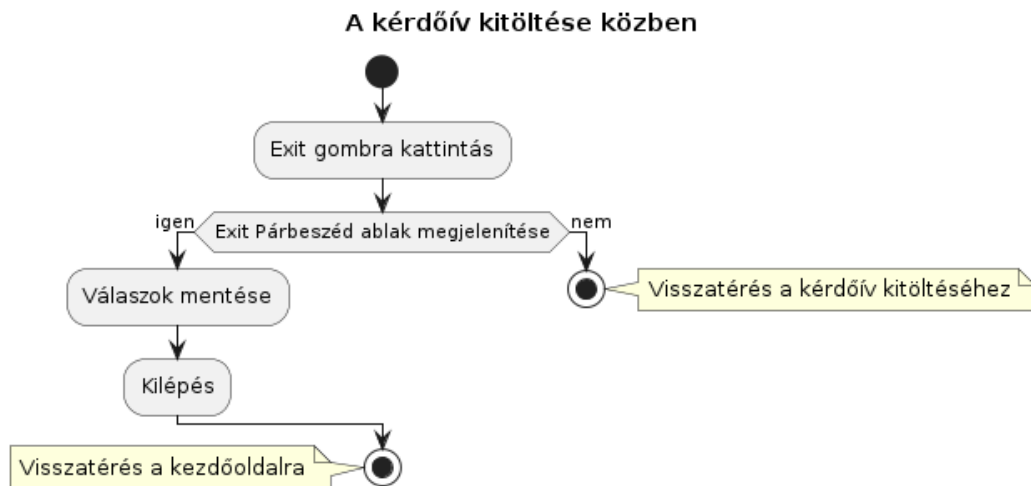
A Kérdőív oldala

A kérdőív oldal felépítése a háttérfolyamatoknak köszönhetően, szerintem az egyik legbonyolultabb nézet. A könnyebb megértés érdekében, egyszerűsített és szétválasztott folyamatábrákkal írom le az oldal működését.



3.4. ábra. A kérdőív egyszerűsített működése [19]

A folyamatábrán jól látszik, hogy normál esetben a képek generálása addig megy, amíg a kitöltő el nem éri a 10. és egyben utolsó képet. Ez után történik meg egyszerre a válaszok feltöltése az adatbázisban. Ez a folyamat az adatbázisba való mentés optimalizálást hivatott megvalósítani. Ugyanis ahelyett, hogy minden azonosítás után töltenénk fel a válaszokat, inkább csak az utolsó válasz után, egyszerre mennek fel az adatok. Így kevésbé terheljük a szerveroldali folyamatokat.



3.5. ábra. A kérdőív kitöltése közben történő kilépés [19]

A fent látott folyamatábra a **kérdőív megszakítása** közbeni válaszok rögzítését mutatja be. Ezzel a funkcióval azt értjük el, hogy hiába nem közvetlen az azonosítások után mentjük a válaszokat, a kérdőív megszakítása előtt még elmentjük az addigi képekre érkező összes választ.



3.6. ábra. A képgenerálások közötti időmérés és válasz rögzítés [19]

Az utolsó folyamatábrán a figyelmeztetés működését mutatom be. Minden kép megjelenésekor, – amennyiben 25 másodperc eltelt úgy, hogy még nem érkezett válasz – 5 másodperc erejéig egy **figyelmeztető jelzés** jelenik meg a képernyő tetején. Miután az 5 másodperc lejárt és még továbbra sem érkezett semmi, úgy a következő kép generálása jön anélkül, hogy bármilyen válasz rögzítésre került volna.

A Képgenerálás beállításának oldala

Ezen az oldalon tudjuk kiválasztani a **képek generálásának módját**. A három különböző módszer kiválasztása az ***image_generation_settings*** adatbázis tábla mezőinek értékeiből dőlnek el. Tehát az oldal betöltésekor lekérjük a mezők értékét és azok alapján állítjuk be az egyes gombokat. Az itt történő adatlekérdezések és módosítások pedig a **37.** oldalon bemutatott *axios.get()* és *axios.post()* kérésekkel valósulnak meg.

A kérdőívben történő tényleges képgeneráláskor, ugyancsak az előbb említett tábla mezőit olvassuk ki. Az itteni értékek alapján hívjuk meg az aktuális függvényt a három **„súlyozott random”** közül, a megfelelő paraméterekkel. A súlyozás a *misidentifications* és *image_frequencies* táblák *count* mezőjéből számítva, illetve egy bizonyos *threshold* (határérték) alatti érték összehasonlításával történik.

3.3.4. További fontos funkciók

Előugró ablakok

Az alkalmazásom során 5 darab előugró ablakot használok:

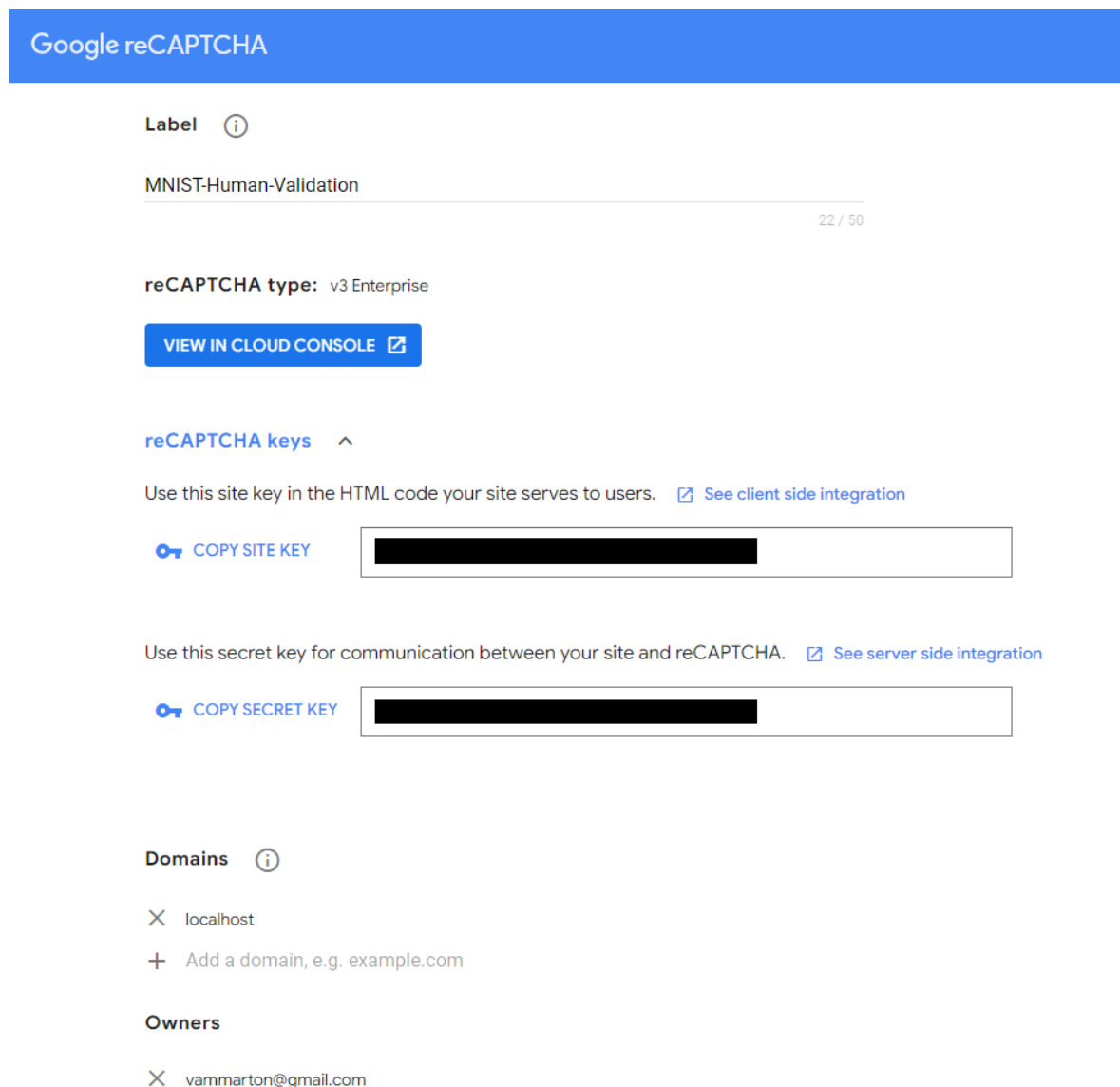
- a személyes adatok beállítására
- a teszt indítására
- a CAPTCHA teszt megjelenítésére
- a kérdőív felfüggesztésére
- a visszajelzés küldésére

Az ilyen párbeszéd ablakok vagy „pop-up”-ok javítják a felhasználói élményt. A React könyvtár előre megírt komponenseket tartalmaz. A „Modal” segítségével kódismétlés nélkül tudtam személyre szabni az egyes nézetekre szánt előugró ablakokat.

CAPTCHA integráció

Ahogy már **korábban említettem**, a kérdőív indítása előtt mindenki számára kötelező egy CAPTCHA teszt kitöltése. A modult egy előugró ablakon jelenítem meg. Az ***'npm install react-google-recaptcha'*** parancs segítségével válik elérhetővé a könyvtár. A reCAPTCHA v2 fejlesztőknek szánt hivatalos oldalán[24] olvashatunk részletesen az API használatáról.

Ahhoz, hogy saját alkalmazásunkba integrálni tudjuk, regisztrálni kell google fiókunkkal a CAPTCHA Admin Console-on. Ezen keresztül kapjuk meg a fejlesztéshez szükséges kulcsokat.



3.7. ábra. Google CAPTCHA Admin Console [25]

Visszajelzés küldése

A visszajelzés küldése során biztosítunk lehetőséget arra, hogy a látogatók „saját” *inputot* küldjenek fel a szerverre. Ezt biztonsági okokból előre kezelnünk kell.

– CSRF védelem:

a CSRF támadások elleni védelem érdekében – axios segítségével – lekérjük a Laravel által generált CSRF tokenet. Ezek a tokenek segítenek ellenőrizni, hogy a kérések valóban az alkalmazáson belülről származnak-e és nem külső forrásból kerültek be.

– **tisztítás:**

az inputok tisztításához és ellenőrzéséhez, illetve az esetleges veszélyes tartalmak vagy injekciók elkerülése érdekében a DOMPurify segítségével (***'npm install dompurify'***) eltávolításra kerülnek a veszélyes HTML elemek (**'<'**) az inputból. Továbbá szűrjük a különböző linkek beküldését.

– **input hossz korlátozás:**

az input mezője – *comment* mező a **feedbacks** adatbázis táblában – *varchar(255)* típusú. Ennek érdekében hosszuk maximum 250 karakterre van korlátozva.

Munkamenet-kezelés

A Laravelben a *./app/Http/config/session.php* útvonalon találhatjuk meg a munkamenet beállításához szükséges fájlt. Itt számomra a *session* driver kiválasztása, élettartama és törlése volt fontos. A bemutatott kódrészleteken látható, hogy alapértelmezetten az *env()* függvénnyel, az alkalmazás környezeti változóiból olvassuk be az adatokat. A *.env* fájl azonban mindenkinél eltérő lehet, így azokat külön is be kell állítani. Erre szolgál a *session.php* konfigurációs fájl.

3.7. kódrészlet. A munkamenet meghajtó kiválasztása

```
1 'driver' => env('SESSION_DRIVER', 'database'),
```

A driver határozza meg a munkamenetadatok elhelyezését. Az én esetemben az adatbázis tárolórendszerben van. Ezeket külön a **sessions** táblában tárolom.

3.8. kódrészlet. A munkamenet élettartamának megadása

```
1 'lifetime' => env('SESSION_LIFETIME', 1 * (60 * 24 * 365))  
    ↪ ,  
2  
3 'expire_on_close' => false,
```

Ebben a kódrészletben a munkamenet élettartamát állítjuk be. Ez határozza meg, hogy mennyi idő után járnak le a munkamenetek. Ha a korábban említett *.env* fájlban nincs konfigurálva, akkor az érték 1 év lesz.[27] A harmadik sorban pedig azt állítom be, hogy a munkamenetek nem fognak lejárni az oldal bezárásakor. Azonban a Laravel biztonsági okokból bejelentkezésakor, illetve kijelentkezésakor mindenképp új sessiont hoz létre. Így az adminisztrátorok munkamenet-azonosítója folyamatosan változik.

3.9. kódrészlet. A munkamenet-söprő sorsolás beállítása

```
1 'lottery' => [2, 100],
```

Ez a sor meghatározza, hogy milyen valószínűséggel történjen munkamenet-söprés egy adott kérés során. Az előre megadott [2, 100] tartomány alapján a Laravel véletlenszerűen választ egy számot és ha ez a szám az adott tartományba esik, akkor

aktiválódik a munkamenet-söprő mechanizmus. Ez a funkció segít megőrizni az adatbázis tisztaságát és optimalizálni a teljesítményt, ugyanis ezzel időnként eltávolítja az inaktív munkameneteket.

Kliens token használat

A kliens tokenek használata szorosan kapcsolódnak a munkamenetekhez. Számomra azonban a különböző kérések elérésénél van szerepük. Ugyanis, ha ismerjük a képek generálásáért felelős *endpointot*, akkor a CAPTCHA és kérdőív oldal megkerülésével halomszámra gyárthatjuk a téves adatokat. Ennek elkerülése érdekében a kérdőív nézetben, olyan kliens tokenet hozunk létre, amely elengedhetetlen lesz a képgeneráláshoz.

3.10. kódrészlet. A kliens token átadása a headerben

```
1 headers: {  
2   'X-Client-Token': clientToken  
3 }
```

A korábban bemutatott (37. oldalon) axios kérés paraméterében küldünk a szerveroldalra egy kliens tokenet minden képgeneráláskor.

3.11. kódrészlet. Kliens token megvizsgálása a képgenerálás előtt

```
1 public function generateMisidentificationWeightedImage(  
2     ↪ Request $request)  
3 {  
4     // Get the unique ID from the request header  
5     $uniqueId = $request->header('X-Client-Token');  
6  
7     if (!$uniqueId) {  
8         return response()->json(['error' => 'Unauthorized'  
9             ↪ ], 401);  
10    }  
11  
12    ...  
13 }
```

A fenti kód, a három **képgenerálási módszer** közül a „**Leggyakrabban félreazonosított képek**” függvény elejét tartalmazza. Jól látható, hogy a függvény tartalmaz egy *\$request* paramétert, amelyet rögtön megvizsgál. Ha a *\$uniqueId* üres, akkor az azt jelenti, hogy a kérést indító személy nem kapta meg a kérdőív oldalról érkező kliens tokenet. Azaz nem az oldalról, hanem csak az URL-ből szeretné generálni a képet. Ilyenkor 401-es számú hibával „*Unauthorized*” (Engedély nélküli) üzenettel tér vissza az oldal. Erről tesztelést is készítettem, amely **itt** tekinthető meg.

4. fejezet

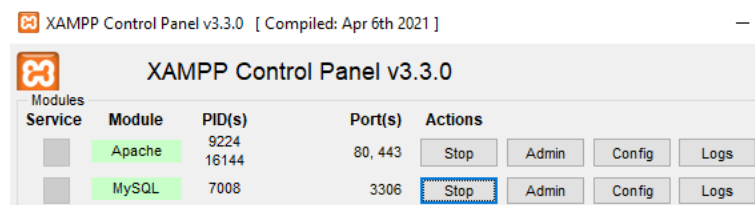
Telepítés

4.1. Egyszerű telepítés zip fájlból

A mellékelt zip fájl tartalma:

- szakdolgozati dokumentáció (F6L4X9_2024.pdf)
- webalkalmazás (mnist-validate-by-human)

A webalkalmazás telepítése előtt rendelkezniünk kell XAMPP vezérlőpulttal. Ennek letöltését **itt** találja. A sikeres letöltést követően az Apache és MySQL modulokat kell elindítani.



4.1. ábra. XAMPP vezérlőpult [26]

A MySQL Admin gombjára kattintva nyissa meg a grafikus interfészt és készítsen egy új adatbázist *mnist_validate_by_human* néven.

Csomagolja ki a webalkalmazást, majd nyissa meg valamilyen fejlesztői környezetben. A fejlesztői környezetben nyisson egy új terminált ahol, az *mnist-validate-by-human* könyvtárban adja ki a következő parancsokat:

- **php artisan migrate** (az adatbázisba való migrációk elvégzése)
- **php artisan db:seed** (az adminisztrátori fiók létrehozása, valamint az alapértelmezett adatok feltöltése)
- **php artisan serve** (az alkalmazás lokális szerveroldali futtatása)

Ezek után, az előzővel párhuzamosan nyissunk új terminált és futtassuk a frontendért felelős parancsot is:

- **npm run dev**

A parancsok sikeres lefutása után keresse meg a `./storage/scripts` mappában található `mnist_images_init.py` python szkriptet. Futtassa le a szkriptet. Ennek segítségével az MNIST adatbázis összes rekordját feltöltjük az adatbázisunkba. A szkript lefutása hosszabb ideig is eltarthat. Működéséhez az alábbi csomagokkal kell rendelkezniünk:

- **pip install matplotlib**
- **pip install python-mnist**
- **pip install mysql-connector-python**
- **pip install keras**
- **pip install tensorflow**

Amennyiben nem sikerült a szkript futtatása, úgy a teljes `mnist_images` tábla tartalmát `itt[30]` tudja letölteni.

Az alkalmazást a **php artisan serve** parancs kiadása utáni http elérésű útvonalon érjük el (általában `http://127.0.0.1:8000`). Ha az alkalmazás teljes funkcióját ki szeretnénk használni, akkor érdemes egy harmadik terminált is nyitni. Itt pedig a **php artisan schedule:work** parancs segítségével a laravel ütemezőjét elindítva automatikusan történik majd a `uuid_images` tábla karbantartása.

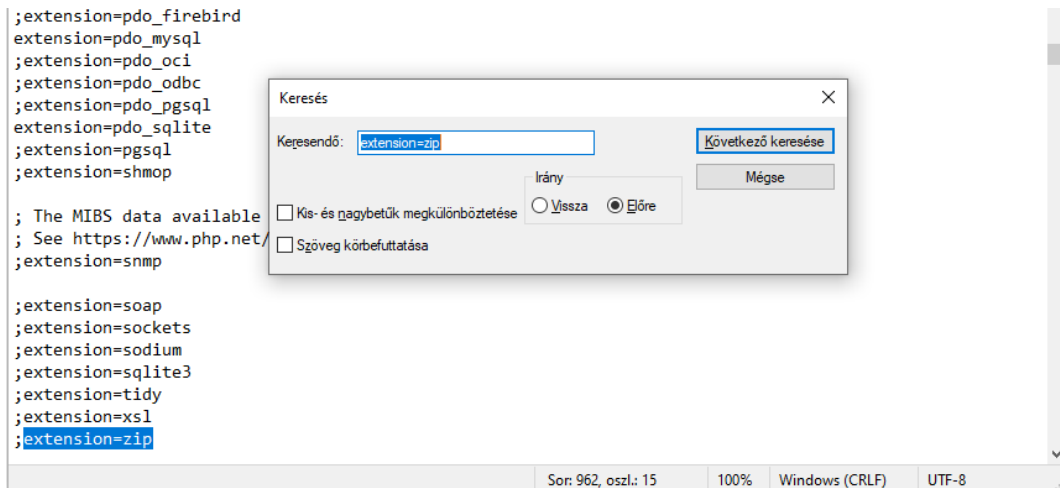
A sikeres műveletek után kitölthetjük a kérdőívet, illetve a `/login` elérési útvonalon bejelentkezhetünk az adminisztrátori fiókunkba, ahol beállíthatjuk a képgenerálás szempontjait.

- **Email:** admin@validatebyhuman.com
- **Password:** password

4.2. Teljes telepítés GitHub-ról

A GitHub-ról való telepítés hasonló a zip-elt verzióhoz. Itt azonban több lépésre van szükségünk. Az alkalmazást az **alábbi linken** éri el. A repository-ból történő klónozás előtt azonban rendelkeznie kell a már fent említett XAMPP vezérlővel, illetve **Node.js** és **Composer** telepítésével.

A 44. oldalon lévő ábrán, az Apache Config gombjára kattintva a `php.ini` konfigurációs fájlban végezzük el az alábbi módosítást:



4.2. ábra. A php.ini konfigurációs fájl *'extension=zip'* sor keresése

Amennyiben az *'extension=zip'* sor előtt *';* (pontosvesszőt) lát, törölje azt ki. Ez azért fontos, mert ha meghagynánk a sor előtt lévő pontosvesszőt, úgy a Composer (a webalkalmazás telepítéskor) az összes csomagot le fogja húzni, ami több gigabyte-os fájlt fog okozni, a mindössze pár száz MB helyett.

Amint végzett az előzetes beállításokkal, nyissa meg a klónozott repository mappáját egy fejlesztői környezetben. Keresse meg a *.env.example* nevű fájlt. Készítsen belőle másolatot és nevezze át *.env*-re. Ezek után keresse meg benne az alábbi sorokat és módosítsa azokat:

4.1. kódrészlet. A *.env* fájl 3 mezőjének módosítása

```
1 DB_DATABASE=mnist_validate_by_human
2 SESSION_DRIVER=database
3 SESSION_LIFETIME=525600
```

Hozzon létre új adatbázist a MySQL grafikus felületén, *mnist_validate_by_human* néven. Majd nyisson három új terminált az *mnist-validate-by-human* könyvtárban és adja ki sorrendben a következő parancsokat:

Szerveroldali terminál:

```
composer install | php artisan migrate | php artisan db:seed | php artisan key:generate
| php artisan serve
```

Kliensoldali terminál:

```
npm install | npm install uuid | npm install react-chartjs-2 chart.js | npm install dom-
purify | npm run dev
```

Az ütemezőért felelős terminál:

```
php artisan schedule:work
```

Az MNIST adatbázis betöltéséhez kövesse az **előző oldalon** leírt python szkript indításáról szóló részt.

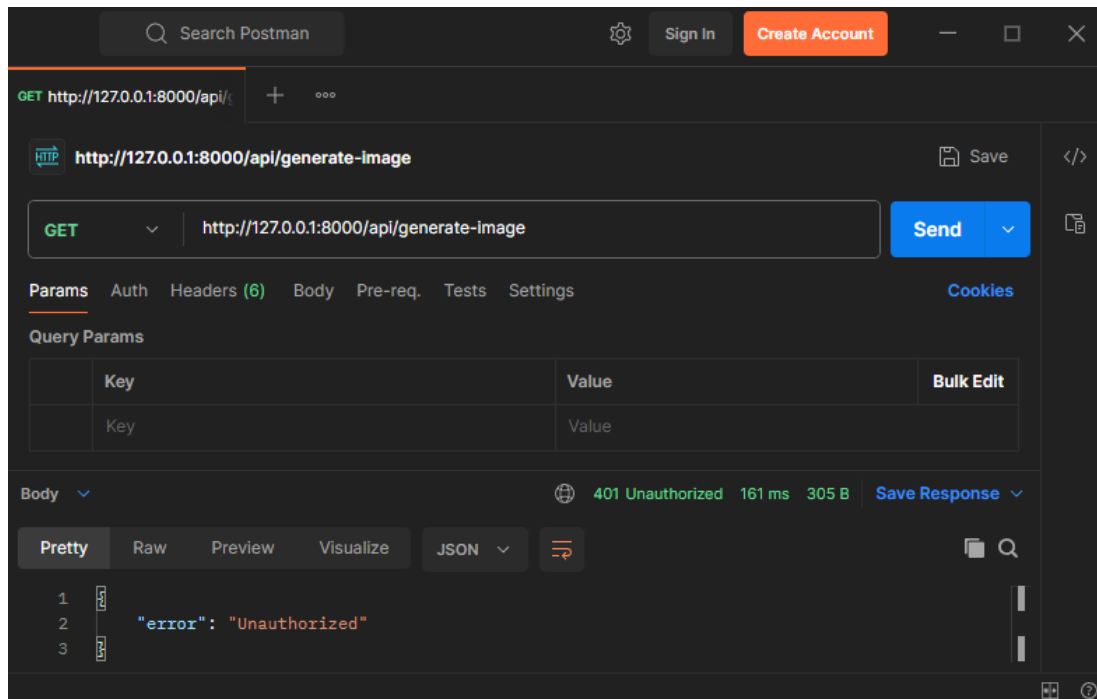
5. fejezet

Tesztelés

Ebben a fejezetben a 43. oldalon bemutatott kliens token, valamint a kérdőív elérésének tesztelését mutatom be. Véleményem szerint ez a webalkalmazás egyik legkritikusabb része, hiszen itt dől el igazán, hogy a kérdőív által szerzett adatokat meglehet-e hamisítani vagy sem.

5.1. API teszt

Elsőként a képgenerálásért felelős *endpoint* segítségével próbálok elérni egy képet az adatbázisból. Postman-en keresztül elküldök egy GET kérést a `http://127.0.0.1:8000/api/generate-image` útvonalra.



5.1. ábra. A `/generate-image` endpoint tesztelése Postman-ből [28]

Mivel nem a kliensoldalon keresztül indítom a kérést, ezért *'Unauthorized'* hibát kapok, ami jelen esetünkben a sikeres kimenetelt jelenti.

5.2. Cypress – Automatizált teszt

A Cypress segítségével automatizált teszteket futtatok az oldalamon, így ellenőrizhetem, hogy a rendszer hogyan kezeli a különböző felhasználói eseteket és interakciókat.

5.1. kódrészlet. Endpoint tesztelése automatizált Cypress szkripttel

```
1 describe('/generate-image_API_teszt', () => {
2   it('ránavigál_a_/generate-image_API_endpoint-ra', () => {
3     cy.request({
4       method: 'GET',
5       url: 'http://127.0.0.1:8000/api/generate-image',
6       failOnStatusCode: false
7     }).then((response) => {
8
9       expect(response.status).to.eq(401)
10
11       expect(response.body).to.include({ error: 'Unauthorized'
12         ↪ })
13     })
14   })
15 })
```

A fenti kódban azt láthatjuk ahogyan megadom az elérni kívánt útvonalat és erre válaszul két dologra számítok. Az első, hogy 401-es hibát dobjon az oldal. A második pedig, az API tesztből kiderülő *JSON objektum* válasza.

```
1 request ○ GET 401
  http://127.0.0.1:8000/api/generate-image
2 - assert expected 401 to equal 401
3 - assert expected { error: Unauthorized } to
  have property error
4 - assert expected { error: Unauthorized } to
  have property error** of **Unauthorized
```

5.2. ábra. Sikeres lefutási teszt: az endpoint nem elérhető [29]

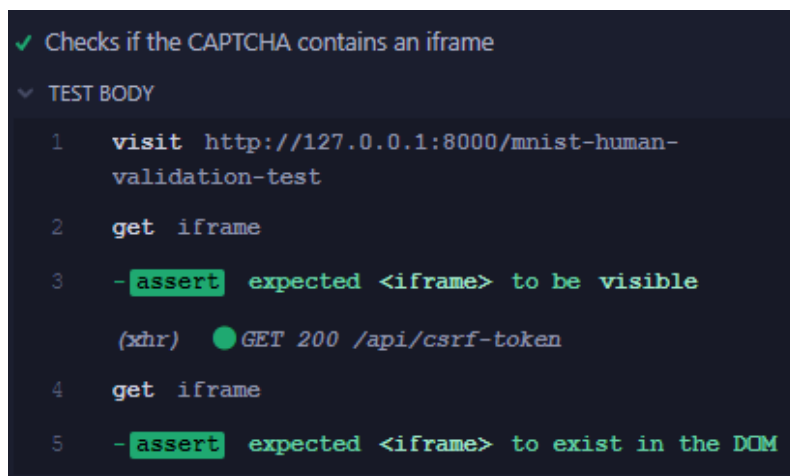
A képről jól kivehető, hogy minden elvárásom teljesült, így egy sikeres teszthez hűen zölden futott le.

Az utolsó teszt a kérdőív oldal eléréséről szól. Itt azt vizsgálom, hogy a kérdőív kitöltése előtti CAPTCHA teszten át tudok-e jutni. Ezt automatizáltan tesztelni nagyon egyszerű volt. Mindössze annyi, hogy rá navigálok a kérdőív oldalának elérési útvonalára (*/mnist-human-validation-test*) és egy *iframe* elemet keresek.

5.2. kódrészlet. CAPTCHA azonosítása

```
1 describe('CAPTCHA és iframe elem teszt', () => {
2   it('ellenőrzi, hogy a CAPTCHA iframe elembe van-e', () => {
3     cy.visit('http://127.0.0.1:8000/mnist-human-validation-
4       ↪ test')
5
6     cy.get('iframe').should('be.visible')
7
8     cy.get('iframe').should('exist')
9   })
10 })
```

Miért pont *iframe* elemet keresek? Azért, mert az általam integrált CAPTCHA egy *iframe*-ben van beágyazva. Ezek pedig közvetlen nem manipulálhatók, így nem is kattinthatók és nem is léphetünk velük interakcióba. Többek között ez is egyfajta biztonsági korlátozás, amely megakadályozza a robot-vezérelt kitöltéseket.



5.3. ábra. Sikeres lefutási teszt: a CAPTCHA nem elérhető [29]

A zöld színű lefutások ismét azt jelzik, hogy az *iframe* elemet sikeresen beazonosította. Így már nem tud továbblépni a CAPTCHA teszten, ezáltal pedig nem jut el a kérdőív oldalán lévő kitöltéshez.

Összességében tehát kijelenthetem, hogy a tesztelések bebizonyították, hogy az oldalt sem automatizált botokkal, sem pedig a generálásért felelős endpoint-ok kihasználásával, nem lehet befolyásolni.

6. fejezet

Továbbfejlesztési lehetőségek

Az utolsó fejezetben azokról a továbbfejlesztési lehetőségekről írok, amelyek a szakdolgozat írása közben jutottak eszembe. Úgy gondolom, hogy ezek mindenképpen növelnék az alkalmazás színvonalát. Nem kizárt tehát, hogy a felsorolt pontokból némelyik implementálásra kerül a közeljövőben.

6.1. Magyar fordítás

A weboldalon teljes egészében angol nyelvet használok. Ez tudatosan van így. Azonban, ha már magyar nyelvű dokumentáció készült hozzá, érdemes lehet az oldalon a magyar fordítást megvalósítani.

6.2. Eredmények megjelenítése

A webalkalmazás elsősorban a kutatásokat segítő adatgyűjtés céljából készült. Azonban biztosan sok kitöltőt érdekelhet a teszt eredménye. Itt arra gondolok, hogy a kérdőív kitöltése után lehetőséget nyújtunk, hogy megtekintsék az általuk megválaszolt számok helyességét.

6.3. Kitöltés folytatása

A kérdőív kitöltése mindössze 1-2 percet vesz igénybe. Mi van akkor, ha kitöltő még szívesen folytatná anélkül, hogy mindig újraindítaná a tesztet? Tíz szám azonosítása nem túl megterhelő, így lehetne egy olyan funkció, amely felajánlja a felmérés kitöltésének folytatását.

6.4. Optimalizálás

Az optimalizálásáról már volt szó a **39.** oldalon, amikor a válaszokat csak a kérdőív megszakításával vagy befejezésével töltöttük fel egyszerre. Az új ötletem is a szerveroldali optimalizálást célozta meg. Jelenleg minden „Next” gomb nyomása után generáljuk a képeket. Ez egy teljes kérdőívnel 10 különböző kérést jelent. Ezt azonban könnyen lecsökkenthetjük egyre, ha csak a kérdőív elindításakor generáljuk le előre a 10 képet. Ezek után pedig, egyesével jelenítjük meg a kitöltőknek. Ha valaki idő előtt befejezné, akkor csak az addig megválaszolt adatokat mentjük el az adatbázisba. Ezzel nem csak a kéréseket optimalizálnám, hanem a töltési időt is. Hiszen ahelyett, hogy minden gombnyomás után várnunk kellene, csak egyszer – a kérdőív indításakor – várakozunk hosszabb ideig. Utána már gördülékenyen jelennének meg a képek.

6.5. Az alkalmazás általánosítása más adatbázisokra

Az alkalmazás jelenlegi formájában csak az MNIST adatbázis képeit dolgozza fel. Ezt azonban könnyen kibővíthetnénk más, pl. a **15.** oldalon felsorolt adatbázisok bármelyikével. Ezáltal akár általánosítva is működhet a webalkalmazás. Még az **adatbázisunkat** sem kellene nagymértékben bővíteni, – hiszen ahogy a tanító és tesztelő képeknél – csak az azonosítók (*image_id*) alapján is megtudjuk mondani, hogy melyik fajta képekről van szó. Ehhez tehát mindössze egy új kérdőív felület kell, amelyen csak az input mezők/gombok változnak.

Összegzés

Az alkalmazás fejlesztése és a szakdolgozat írása közben rengeteget tanultam. Az egyik célom többek között ez is volt. Úgy gondolom, hogy nem fektettem volna bele ennyi időt és energiát, illetve a projekt sem készült volna el, ha magában a témában nem lettem volna ennyire érdekelt. Örülök, hogy ezen dolgozhattam és örülök az eredménynek is. Sokat köszönhetek a konzulensemnek, az egyetemi tanárainknak, a statisztikai oldalon dolgozó szakdolgozati társamnak és a családtagjaimnak. Nélkülük ez ilyen formában nem valósult volna meg.

A későbbiekben bízom benne, hogy a webalkalmazás éles környezetbe is kikerülhet és helyt tud majd állni. A végső célom ugyanis az volt, hogy egy olyan környezetet biztosítsak, amely valamilyen szempontból mindenki számára hasznos és kényelmes. Azt gondolom, hogy az alkalmazás számos területen előrébb viheti majd a tudásunkat. Tisztában vagyok vele, hogy egy egyszerű adatgyűjtő platformról van szó, de az adatok, valamint az információ az egyik legnagyobb érték a mai világban. Ezekből tanulhatunk és taníthatunk. Ebből kiindulva pedig talán nem is annyira elrugaszkodott gondolat, hogy ezen adatok segítségével akár hatékonyabb neurális hálózatokat fejleszthetünk, sőt még a mesterséges intelligenciát is formálhatjuk egy bizonyos szinten.

Összességében tehát elégedett vagyok a munkámmal és szívesen gondolok majd vissza a teljes folyamatra.

Irodalomjegyzék

- [1] COLLINS DICTIONARY WORLD OF THE YEAR 2023
URL: <https://www.collinsdictionary.com/woty>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [2] XENONSTACK – DEEP LEARNING VS MACHINE LEARNING VS NEURAL NETWORK – WHAT’S THE DIFFERENCE?
URL: <https://www.xenonstack.com/blog/deep-learning-vs-ml-vs-neural-network>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [3] LEXIQ – MNIST ADATBÁZIS
URL: <https://lexiq.hu/mnist-adatbazis>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [4] SZINONIMASZOTAR.HU – KÉRDŐÍV
URL: <https://szinonimaszotar.hu/keres/kérdőív>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [5] THESAURUS.COM – QUESTIONNAIRE
URL: <https://www.thesaurus.com/browse/questionnaire>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [6] QUALTRICS – SURVEY VS QUESTIONNAIRE: THE DIFFERENCES FOR MARKET RESEARCH
URL: <https://www.qualtrics.com/experience-management/research/survey-vs-questionnaire/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [7] MERSZ.HU – KUTATÁSI MÓDSZEREK
URL: <https://mersz.hu/blog/kutatasi-modszerek/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [8] LEXIKON.UNI-NKE.HU – METAADAT
URL: <https://lexikon.uni-nke.hu/szocikk/metaadat-informatika/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.

- [9] QUESTIONPRO – ONLINE SURVEY ADVANTAGES
URL: <https://www.questionpro.com/blog/what-are-online-surveys/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [10] LEXUNIT.HU – MI AZ A BIG DATA ÉS MIRE HASZNÁLJUK?
URL: <https://lexunit.hu/blog/mi-az-a-big-data-es-mire-hasznaljuk/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [11] SIMONWENKEL – LIST OF MNIST-LIKE DATASETS
URL: <https://www.simonwenkel.com/lists/datasets/list-of-mnist-like-datasets.html>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [12] EUROPA.EU – AZ ÁLTALÁNOS ADATVÉDELMI RENDELET SZERINTI ADATVÉDELMI SZABÁLYOK
URL: https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index_hu.htm
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [13] KOCSIS ÉS SZABÓ ÜGYVÉDI IRODA – MIT KELL TUDNI AZ ÁLTALÁNOS SZERZŐDÉSI FELTÉTELEKRŐL?
URL: <https://kocsisszabougyved.hu/mit-kell-tudni-az-altalanos-szerzodesi-feltetelekrol/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [14] FONTAWESOME.COM – A WEBALKALMAZÁSHOZ HASZNÁLT IKONOK
URL: <https://fontawesome.com/search?o=r&m=free>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [15] HEALTHWIRE.PK – 11 SURPRISING ADVANTAGES OF BEING A LEFT-HANDED PERSON
URL: <https://healthwire.pk/healthcare/advantages-of-a-left-handed-person/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [16] DATADOME.CO – CAPTCHA VS. reCAPTCHA: WHAT'S THE DIFFERENCE?
URL: <https://datadome.co/bot-management-protection/captcha-vs-recaptcha-whats-the-difference/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [17] MATPLOTLIB.ORG – KÉPGENERÁLÁS MÉRÉSÉRE HASZNÁLT DIAGRAM
URL: <https://matplotlib.org/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.12.

- [18] DR. KUSPER GÁBOR – INFORMATIKAI RENDSZEREK TERVEZÉSE
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14. (VERZIÓ: 0.9.1.3)
- [19] PLANTUML.COM – HASZNÁLATI ESET DIAGRAM ÉS FOLYAMAT ÁBRÁK
URL: <https://plantuml.com/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.12.
- [20] DBDRAIGRAM.IO – ADATBÁZISTERV ELKÉSZÍTÉSE
URL: <https://dbdiagram.io/docs/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.12.
- [21] UIZARD.IO – KÉPERNYŐ TERVEK ELKÉSZÍTÉSE
URL: <https://uizard.io/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [22] MERNOKKAPU.HU – MIT JELENT AZ ADATINTEGRITÁS AZ ADATBÁZIS-KEZELÉSBEN?
URL: <https://mernokkapu.hu/mit-jelent-az-adatintegritas-az-adatbazis-kezelesben/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [23] HOYO TECH – THE POWER COUPLE: EXPLORING THE ADVANTAGES OF INERTIA.JS WITH LARAVEL AND REACT
URL: <https://hoyo.tech/article/the-power-couple-exploring-the-advantages-of-inertiajs-with-laravel-and-react>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [24] RECAPTCHA – RECAPTCHA v2 FEJLESZTŐKNEK
URL: <https://developers.google.com/recaptcha/docs/display>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [25] RECAPTCHA – ADMIN CONSOLE
URL: <https://www.google.com/recaptcha/about/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [26] XAMPP – VEZÉRLŐPANEL
URL: <https://www.apachefriends.org/hu/index.html>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [27] MAZER.DEV – HOW TO INCREASE LARAVEL SESSION LIFETIME
URL: <https://mazer.dev/en/laravel/posts/how-to-increase-laravel-session-lifetime/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.

- [28] POSTMAN – TESZTELÉSI KÖRNYEZET I.
URL: <https://www.postman.com/>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [29] CYPRESS.IO - TESZTELÉSI KÖRNYEZET II.
URL: <https://docs.cypress.io/guides/overview/why-cypress>
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.
- [30] VÁMOS MÁRTON ISTVÁN – AZ TELJES MNIST_IMAGES TÁBLA TARTALMA
URL: https://drive.google.com/file/d/1dWj4gTjfn2vhWxcfQn0BbfdjDvnErAAD/view?usp=drive_link
UTOLSÓ MEGTEKINTÉS DÁTUMA: 2024.04.14.

NYILATKOZAT

Alulírott *Vámos Márton István*, büntetőjogi felelősségem tudatában kijelentem, hogy az általam benyújtott, *Webalkalmazás fejlesztése az MNIST adatbázis humán validációjához* című szakdolgozat (diplomamunka) önálló szellemi termékem. Amennyiben mások munkáját felhasználtam, azokra megfelelően hivatkozom, beleértve a nyomtatott és az internetes forrásokat is.

Tudomásul veszem, hogy a szakdolgozat elektronikus példánya a védelem után az Eszterházy Károly Katolikus Egyetem könyvtárába kerül elhelyezésre, ahol a könyvtár olvasói hozzájuthatnak.

Kelt: Eger, 2024. április 14.

Vámos Márton István

aláírás