Kivonat

Az internetes böngészésnek és problémák online megoldásának egyik velejárója, hogy felhasználói fiókokat hozunk létre az adatainknak a biztonságos tárolása végett, és hogy speciális funkciókhoz jussunk. Hogy ezeket az adatokat biztonságosan tároljuk, és más ne férhessen hozzá, különféle azonosítási módszereket dolgoztak ki a felhasználók védelme érdekében. Ha valaki megszerzi a belépési információnkat, akkor képes arra, hogy felhasználónk adatait, vagy akár anyagi javait birtokolja.

Diplomamunkám célja egy olyan másodlagos hitelesítő rendszer létrehozása webes alkalmazások számára, amely a felhasználónév-jelszó párossal való beléptetést még biztonságosabbá teszi. Ezt úgy éri el, hogy ellenőrzi a felhasználó által beírt karakterek leütési időit, és összehasonlítsa a korábbi mintákkal. Az adatokat a könnyű kezelhetőség érdekében Firebase valós idejű adatbázisban tároltam. Az alkalmazást Laravel PHP keretrendszerben készítettem el, mivel a célközönség főként a webes alkalmazások körében van.

Az eredmény egy gyűjtő felület, ahol a felhasználóktól adatokat gyűjthetünk, hogy saját adathalmazra tegyünk szert, valamint egy azonosító felület, ahol le tudjuk tesztelni, mennyire működik éles helyzetekben az alkalmazás. Emellett a kísérletek során kapott eredmények hasznosak lehetnek a jövőbeli kutatásokra nézve.

***Kulcsszavak:*** billentyűzés alapú felismerés, másodlagos hitelesítés, Laravel, webes biztoság

Tartalomjegyzék

[**1.** **Bevezető** 2](#_Toc517650667)

[**2.** **A projekt célja** 2](#_Toc517650668)

[**3.** **Bibliográfiai tanulmány** 2](#_Toc517650669)

[**4.** **Elméleti megalapozás** 3](#_Toc517650670)

[**4.1.** **A biometrián alapuló azonosítás áttekintése** 3](#_Toc517650671)

[**4.2.** **Billentyűzési biometria és a web** 3](#_Toc517650672)

[**4.2.1.** **Bemeneti mezők megválasztása** 3](#_Toc517650673)

[**4.2.2.** **Jellemzők kiemelése** 3](#_Toc517650674)

[**4.2.3.** **Felhasznált algoritmusok** 3](#_Toc517650675)

[**4.2.4.** **Az egyenlőségi hibaarány** 3](#_Toc517650676)

[**4.3.** **Felhasznált technológiák** 3](#_Toc517650677)

[**4.3.1.** **Adattárolás NoSQL-ben** 3](#_Toc517650678)

[**4.3.2.** **Más technológiák** 3](#_Toc517650679)

[**5.** **A rendszer specifikációi** 3](#_Toc517650680)

[**5.1.** **Követelmény specifikáció** 4](#_Toc517650681)

[**5.1.1.** **Kivonat** 4](#_Toc517650682)

[**5.1.2.** **Bevezető** 4](#_Toc517650683)

[**5.1.3.** **Áttekintés** 4](#_Toc517650684)

[**5.1.4.** **Rendszer követelmények** 4](#_Toc517650685)

[**5.1.5.** **Funkcionális követelmények** 4](#_Toc517650686)

[**5.1.6.** **Nem funkcionális követelmények** 4](#_Toc517650687)

[**5.1.7.** **Interfész követelmények** 4](#_Toc517650688)

[**6.** **Részletes tervezés** 4](#_Toc517650689)

[**6.1.** **Adatok tárolása Firebaseben** 4](#_Toc517650690)

[**6.2.** **Architektúra** 4](#_Toc517650691)

[**6.3.** **A rendszer tervezése és bemutatása** 4](#_Toc517650692)

[**6.3.1.** **A kontroller rész** 4](#_Toc517650693)

[**6.3.2.** **A modell rész** 4](#_Toc517650694)

[**6.3.3.** **A nézet (view) rész** 5](#_Toc517650695)

[**6.3.4.** **Segítő osztályok** 5](#_Toc517650696)

[**6.4.** **AJAX hívások bemutatása** 5](#_Toc517650697)

[**6.5.** **Adatok mozgása** 5](#_Toc517650698)

[**7.** **A rendszer felhasználása** 5](#_Toc517650699)

[**7.1.** **Adatgyűjtő felület** 5](#_Toc517650700)

[**7.2.** **Bejelentkezési felület** 5](#_Toc517650701)

[**8.** **Gyűjtő üzembe helyezése és kísérleti eredmények** 5](#_Toc517650702)

[**8.1.** **Gyűjtő üzembe helyezése** 5](#_Toc517650703)

[**8.2.** **Felmerült problémák és megoldásaik** 5](#_Toc517650704)

[**8.3.** **Kísérleti eredmények** 5](#_Toc517650705)

[**9.** **Következtetések** 5](#_Toc517650706)

[**9.1.** **Megvalósítások** 5](#_Toc517650707)

[**9.2.** **Összehasonlítás hasonló rendszerekkel** 5](#_Toc517650708)

[**9.3.** **További fejlesztési irányok** 6](#_Toc517650709)

[**10.** **Irodalomjegyzék** 6](#_Toc517650710)

[**11.** **Függelékek** 6](#_Toc517650711)

1. **Bevezető**

Az internet rohamos fejlődése által egyre több olyan tevékenységet vagyunk képesek lebonyolítani otthon, ami személyes jelenlétet igényelt volna évekkel ezelőtt. Manapság a legújabb okos telefonok megvásárlásától a banki átutalásokig bármit képesek vagyunk pár mozdulattal lebonyolítani, és ezzel arányosan hatalmasra nőtt a felhasználói fiókjainknak az értéke is. Bárki, aki az adatainknak, vagy éppen azonosító tárgyainknak (pl. bankkártya, beléptető tóken) a tulajdonában van, képes arra, hogy hozzáférjen a személyes adatainkhoz, vagy akár az anyagi értékeinkhez is.

Tegyük fel, hogy nem biztonságosan kezeltük egy online banki alkalmazásnál használt fiókunkat, és a belépési adatainkat megszerezte egy harmadik fél, akinek szándéka ezt a lehetőséget kihasználni. Ilyen esetben segít egy másodlagos hitelesítő rendszer, ami képes a detektálására az esetleges betolakodóknak. A rendszer észleli azt, hogy a belépett személynek egyes tulajdonságai nem egyeznek meg a regisztrált felhasználóéval, ezért jelzést küld, és további megerősítést kér. Amennyiben a megerősítés nem megy végbe, a rendszer elkönyvelheti, hogy feltörés ment végbe, és segítséget tud nyújtani a felhasználónak a további biztonsági lépések megtételéhez.

Egy ilyen rendszernél például ajánlatos viselkedési biometrián alapuló eszközökkel ellenőrizni, hogy valódi-e a felhasználó. Ezek az eszközök képesek arra, hogy a felhasználót a vele született, egyedi viselkedés mintái és mikro-mozzanatai alapján összehasonlítsák a korábbi belépéseihez, és megállapítsák, hogy mekkora hasonlóság mérhető nála.

Általában egy ilyen rendszernél nem csak egyfajta viselkedést ellenőriznek, ugyanis a viselkedési biometrián alapuló hitelesítők nem képesek száz százalékos eredményt garantálni, és megeshet, hogy a hamis felhasználót beengedik, vagy éppen a valós felhasználót kizárják. Ennek elkerülésére a rendszer több viselkedést is kell, hogy figyeljen, és az egyes tulajdonságok (billentyűzés, egérmozgás, szóhasználat, stb.) mind visszaadnak egy szavazatot, amit összegezve megkapjuk, hogy mi a teendő az adott felhasználóval.

A fiókunk biztonsága érdekében tehát érdemes minél jobban megerősíteni a védelmet, és erre egy módszer a billentyűzési ritmus figyelése. A felhasználónak tudnia nem kell róla, mert a háttérben működik. Emellett már egyre jobb és jobb eredményeket érnek el ilyen téren, tehát valószínű, hogy hamarosan egyre több oldalon lesz lehetőség ez a fajta másodlagos azonosítás is.

1. **A projekt célja**

Az első fejezetben említett billentyűzés alapú azonosítási modul megvalósításában nyilvánul meg a célom. Ez magába foglalja egy gyűjtő rendszer létrehozását, mely segítségével a további kísérletekhez tudunk tanító és tesztelő adathalmazokat biztosítani, valamint egy azonosító rendszer létrehozását, ami segítségével a felhasználót egy második biztonsági réteggel is meg tudjuk védeni.

Jelenleg számos kísérlet folyik jobb és jobb eredményekért a billentyűzési ritmus alapú felismeréssel kapcsolatosan. Már a telegráf korában felfigyeltek az emberek a tényre, hogy be tudják egymást azonosítani a beütési ritmusuk alapján. Ez többnyire annak is köszönhet, hogy hogy az emberi agynak ugyanazok a neuro-fiziológiai részei kerülnek használatba gépeléskor, mint amik az aláírások esetében is. Bár egyre több kísérlet folyik ilyen irányban, teljes pontosságú megoldást még nem találtak, és viszonylag kevés rendszernél alkalmazzák a módszert.

Az alkalmazás célközönsége az érzékeny adatokat kezelő alkalmazások köre, mivel ilyen esetekben kifejezetten fontos az, hogy az belépett felhasználó az legyen, aki a tulajdonosa a fióknak.

­Megvalósítás céljából a PHP egyik keretrendszerét, a Laravelt használtam. A választás azért esett a Laravel-re, mert számos ponton megkönnyíti a rendszer implementációját, és amennyiben más PHP alapú rendszerbe szeretnénk a modult integrálni, kevés átalakítással képesek vagyunk rá. Az alkalmazás egyaránt képes gyűjteni adatokat a valós felhasználótól, valamint külön oldal van létre hozva arra is, hogy pár, a kísérletbe beavatott személy próbáljon belépni a tesztelt felhasználó adataival. Ezeket a belépési kísérleteket szintén elmentjük. A teljes adatbázis JSON formátumban van eltárolva a Firebase valós idejű adatbázisban, ezért a kísérletekhez, amiket Matlabban folytattunk, létrehoztunk egy átalakítót is, ami a JSON formátumban megadott adatainkat csv formátumba konvertálja.

Egy másik célja az államvizsga munkámnak, hogy minél hatékonyabb algoritmust, illetve jellemző kiemelő módszert találjak kísérletek által. Emellett az alkalmazást működővé tenni mobil eszközökön illetve táblagépeken is.

1. **Bibliográfiai tanulmány**
2. **Elméleti megalapozás**
   1. **A biometrián alapuló azonosítás áttekintése**
   2. **Billentyűzési biometria és a web**
      1. **Bemeneti mezők megválasztása**
      2. **Jellemzők kiemelése**
      3. **Felhasznált algoritmusok**
      4. **Az egyenlőségi hibaarány**
   3. **Felhasznált technológiák**
      1. **Adattárolás NoSQL-ben**
      2. **Más technológiák**
3. **A rendszer specifikációi**
   1. **Követelmény specifikáció**
      1. **Kivonat**
      2. **Bevezető**
      3. **Áttekintés**
      4. **Rendszer követelmények**
      5. **Funkcionális követelmények**
      6. **Nem funkcionális követelmények**
      7. **Interfész követelmények**
4. **Részletes tervezés**
   1. **Adatok tárolása Firebaseben**
   2. **Architektúra**
   3. **A rendszer tervezése és bemutatása**
      1. **A kontroller rész**
      2. **A modell rész**
      3. **A nézet (view) rész**
      4. **Segítő osztályok**
   4. **AJAX hívások bemutatása**
   5. **Adatok mozgása**
5. **A rendszer felhasználása**
   1. **Adatgyűjtő felület**
   2. **Bejelentkezési felület**
6. **Gyűjtő üzembe helyezése és kísérleti eredmények**
   1. **Gyűjtő üzembe helyezése**
   2. **Felmerült problémák és megoldásaik**
   3. **Kísérleti eredmények**
7. **Következtetések**
   1. **Megvalósítások**
   2. **Összehasonlítás hasonló rendszerekkel**
   3. **További fejlesztési irányok**
8. **Irodalomjegyzék**
9. **Függelékek**