**KÉPZÉSI PORTÁL**

**Csoportmunka feladat a**

**GKLB\_INTM005 – Projektmunka tantárgyhoz**

**Készítette:**

**Nemes Mátyás, DUBPK3, Mérnökinformatikus levelezős hallgató**

**Homlok Bence, JAFPIY, Gazdaságinformatikus levelezős hallgató**

**Rózsa László, A83LQ0, Gazdaságinformatikus levelezős hallgató**

**Tartalomjegyzék**

**Bevezetés 3**

**SWOT analízis 4**

**Követelmények 5**

Funkcionális 5

Nem funkcionális 5

**Programnyelvek kiválasztása 6**

**Use Case diagram 7**

**Az egyes funkciók bemutatása**

Bejelentkezés

Kezdőoldal

Képzések

Kompetenciáim

KPI értrékelések

Admin

Saját adatok karbantartása

Kijelentkezés

**Adatbázis bemutatása**

**GUI tervek**

**BEVEZETÉS**

A tantárgy keretében egy olyan web alapú képzési portált szerettünk volna elkészíteni, mely lehetőséget biztosít rá, hogy egy vállalat alkalmazottai a munkájukhoz szükséges oktatásokkal, képzésekkel, valamint a megszerzett tanúsítványaikkal és kompetenciáikkal kapcsolatos adminisztrációs feladatokat elintézzék. Az egyes felhasználók a portálon megtalálják a számukra elérhető képzéseket, jelentkezni tudnak azokra, illetve a már teljesített képzéseiket is áttekinthetik. Az alkalmazás használatával könnyen és egyszerűen szervezhetők céges oktatások, tanfolyamok, workshopok, továbbá nyomon követhetők az egyes alkalmazottak kompetenciái.

Ahogy az a való életben is lenni szokott, ezeket a feladatokat általában nem egy ember csinálja meg, hanem csapatmunka eredményeképp jön létre, mely során nagy hangsúlyt kap az együttműködés. A megfelelő koordináció érdekében mi is választottunk egy vezetőt Nemes Mátyás szerepében. Amellett, hogy aktívan részt vett a projekt gyakorlati megvalósításában, koordinálta a másik két személy munkáját.

**SWOT analízis**

A portál tervezésekor azt vettük alapul, hogy a (kis)vállalkozás, amely a weboldalt használni fogja, alapvetően nem rendelkezik ilyen célú informatikai rendszerrel. A képzések, valamint az egyes dolgozók kompetenciái Excel táblázatban kerülnek vezetésre, tárolásra. Könnyen belátható, hogy ez a fajta dokumentáció lassú, nehézkes és sok esetben átláthatatlan.

A SWOT analízis egy olyan elemzési technika, amivel egy termék, ötlet vagy projekt életképességét tudjuk megvizsgálni. Jól láthatóvá teszi az elemzés tárgyának az erősségeit, gyengeségeit, lehetőségeit és veszélyeit.

|  |  |
| --- | --- |
| **Erősségek:** | **Gyengeségek:** |
| * platformfüggetlen * felhasználóbarát * internet segítségével bármikor, bárhonnan elérhető * mobiltelefonról is elérhető | * korlátozott adatbázis méret |
| **Lehetőségek:** | **Veszélyek:** |
| * igény szerint fejleszthető * a moduláris felépítésnek köszönhetően könnyen bővíthető * az adminisztráción megtakarított idő hasznosabb felhasználása | * az informatikai rendszer leállása/meghibásodása esetén nem használható * internetkapcsolat nélkül nem használható |

**Követelmények**

A weboldal elkészítése előtt meg kellett határoznunk a funkcionális és nemfunkcionális követelményeket. A célok meghatározása azért fontos a tervezési szakaszban, mivel az ennek során felállított követelményekhez tudunk visszavisszatérni ellenőrizni, hogy a termék megfelel-e az előzetes elvárásoknak.

**Funkcionális követelmények:**

* bejelentkezés email cím és jelszó használatával;
* a regisztrált felhasználók adatait egy dinamikusan frissíthető adatbázisban tárolja;
* új felhasználó regisztrálására legyen alkalmas;
* az aktuálisan elérhető képzések jelenjenek meg a felhasználónál;
* az egyes képzések leírása legyen megtekinthető;
* új ismeretek felvitelét tegye lehetővé;
* az elért eredményeket tárolja és azt jelenítse is meg;
* saját adatok módosítására legyen lehetőség.

**Nem funkcionális követelmények:**

A nem funkcionális követelmények olyan elvárások a szoftverünk felé, ami a felhasználó kényelmét elégíti ki.

A programtól elvárjuk, hogy legyen:

* 24/7 legyen elérhető, hogy az egyes képzésekre bármikor lehessen jelentkezni, a megszerzett tanúsítványok bármikor megtekinthetőek legyenek;
* felhasználóbarát kezelőfelület (mobilon is használható legyen);
* megfelelő sebesség;
* adatvédelem a személyes adatok tárolása miatt különösen fontos, a GDPR-ral összhangban kell lennie.

**Technológiák kiválasztása**

**Fejlesztői eszközök**

A technológiák kiválasztása során azt tartottuk szem előtt, hogy modern, mégis minden nagyobb böngészővel (és azok régebbi verzióival) kompatibilis eszközöket használjunk.

A fejlesztés során mindhárman ugyanazt a fejlesztői környezetet, a Visual Studio Code-ot használtuk. Azért esett erre a szoftverre a választásunk, mert ez az egyik legnépszerűbb kódszerkesztő, ingyenes, nyílt forráskódú, platformfüggetlen, natívan támogatja a JavaScript-et, és beépített terminállal is rendelkezik.

Az alkalmazás architektúráját, és az egyes feladatok leírását Nemes Mátyás készítette el és priorizálta, majd felosztottuk azokat egymás között (Rózsa László: webdesign, CSS; Nemes Mátyás: frontend, architektúra, projektmenedzsment; Homlok Bence: backend, adatbázis, autentikáció). Az előrehaladás nyomon követéséhez egy Trello board-ot használtunk, ahol folyamatosan nyomon tudtuk követni, hogy kinek milyen feladatai vannak, éppen ki min dolgozik, és mely feladatok készültek el, tehát törekedtünk az agilis módszertanok követésére, és az egymás közötti munka koordinálására.

A csapat munkáját egy verziókezelő szoftverrel, a Git-tel hangoltuk össze. Az alkalmazás repository-ját feltöltöttük GitHub-ra is, hogy egy központi helyről mindig le tudjuk tölteni a legfrissebb verziót. A repository [ezen a linken](https://github.com/nemesmatyas/kepzesi-portal-projektmunka) található.

A fejlesztés során mindegyikünk egy külön branch-et használt. Az elkészült munkát commitoltuk, és pull request-et készítettünk. A kódellenőrzést és a tesztelést Nemes Mátyás végezte, és az eredménytől függően elfogadta, vagy elutasította a pull requestet. Elfogadás esetén a pull request merge-elésre került, azaz „beolvadt” az alkalmazás fő (main) ágába. A Git és GitHub lehetővé tette, hogy párhuzamosan dolgozzunk az alkalmazás különböző részein anélkül, hogy egymás kódját módosítanánk, vagy törölnénk. Emellett megkönnyítette a kódellenőrzést és a tesztelést is, hiszen a hozzáadott, módosított, vagy törölt kódrészleteket a GitHub színkódok segítségével kiemeli.

**Nyelvek, könyvtárak, keretrendszerek**

Mivel webalkalmazást fejlesztettünk, így a webfejlesztés három *de facto* szabványát, a HTML-t, a CSS-t, és a JavaScript-et nem kerülhettük meg, de a fejlesztés megkönnyítése, és az alkalmazás sebességének javítása érdekében két másik fejlesztői eszközt is felhasználtunk.

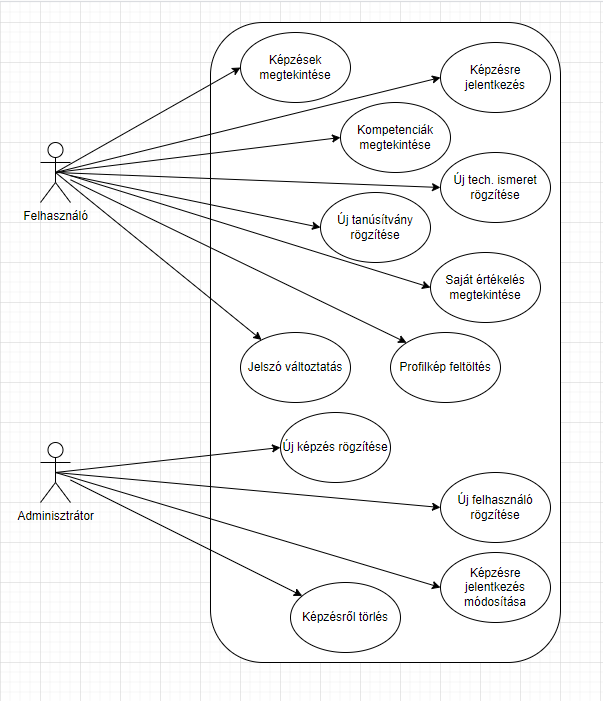
A frontenden az egyik legnépszerűbb JavaScript könyvtárat, a React-et használtuk. A React egy felhasználói felületek készítésére alkalmas könyvtár. Az interfészen elhelyezkedő egyes elemek (pl. navigációs sáv, képzéseket tartalmazó lista) komponensek formájában jelennek meg, amelyek kódszinten egyszerű függvények. A komponensek kaphatnak más komponensektől paramétereket *(props),* és tárolhatják az állapotukat *(state).* A függvényen belül bármilyen logikát megvalósíthatunk, sőt, további metódusokat is elkészíthetünk, hiszen a JavaScript első osztályú függvényeket használ, azaz objektumként kezeli azokat, így eltárolhatjuk őket változókban, és visszatérési értékként is szerepelhetnek.

A logika megvalósítása után visszatérési értékként megadhatunk egy HTML-szerű struktúrát, amiben definiáljuk, hogy mit jelenítsünk meg a UI-on. A JavaScript-be ágyazott HTML a React egyik sajátosságának, a JSX formátumnak köszönhető, futásidőben azonban ez is JavaScript-re fordul le.

A komponens-alapú felépítésnek köszönhetően átláthatóbbá vált a teljes kódbázis, és könnyebben tudtunk együtt dolgozni, hiszen külön fájlokban tároltuk a komponenseket, a CSS-t, és a backend-hez tartozó kódot. Emellett kihasználtuk a React egy másik előnyös tulajdonságát: SPA-t *(Single Page Application)* hoztunk létre, ami azt jelenti, hogy csak akkor küldünk HTTP kérést a szerverre, ha új adatra van szükségünk, minden egyéb esetben a már böngészőre letöltött JavaScript végzi a felhasználói felület módosítását, ezzel javítva az alkalmazás teljesítményét.

A másik eszköz, amit felhasználtunk, a Firebase. Minden modern webalkalmazástól elvárható, hogy rendelkezzen bejelentkezési funkcióval, és az adatokat ne lokálisan, hanem egy adatbázisban tárolja, valamint csak azokat az adatokat olvassuk ki, amelyekre szükségünk van és amelyekhez rendelkezünk jogosultsággal. Azért választottuk a Firebase-t, mert mindkét funkciót képes ellátni, ingyenes és robosztus infrastruktúra áll mögötte, hiszen a Google felügyeli a fejlesztését.

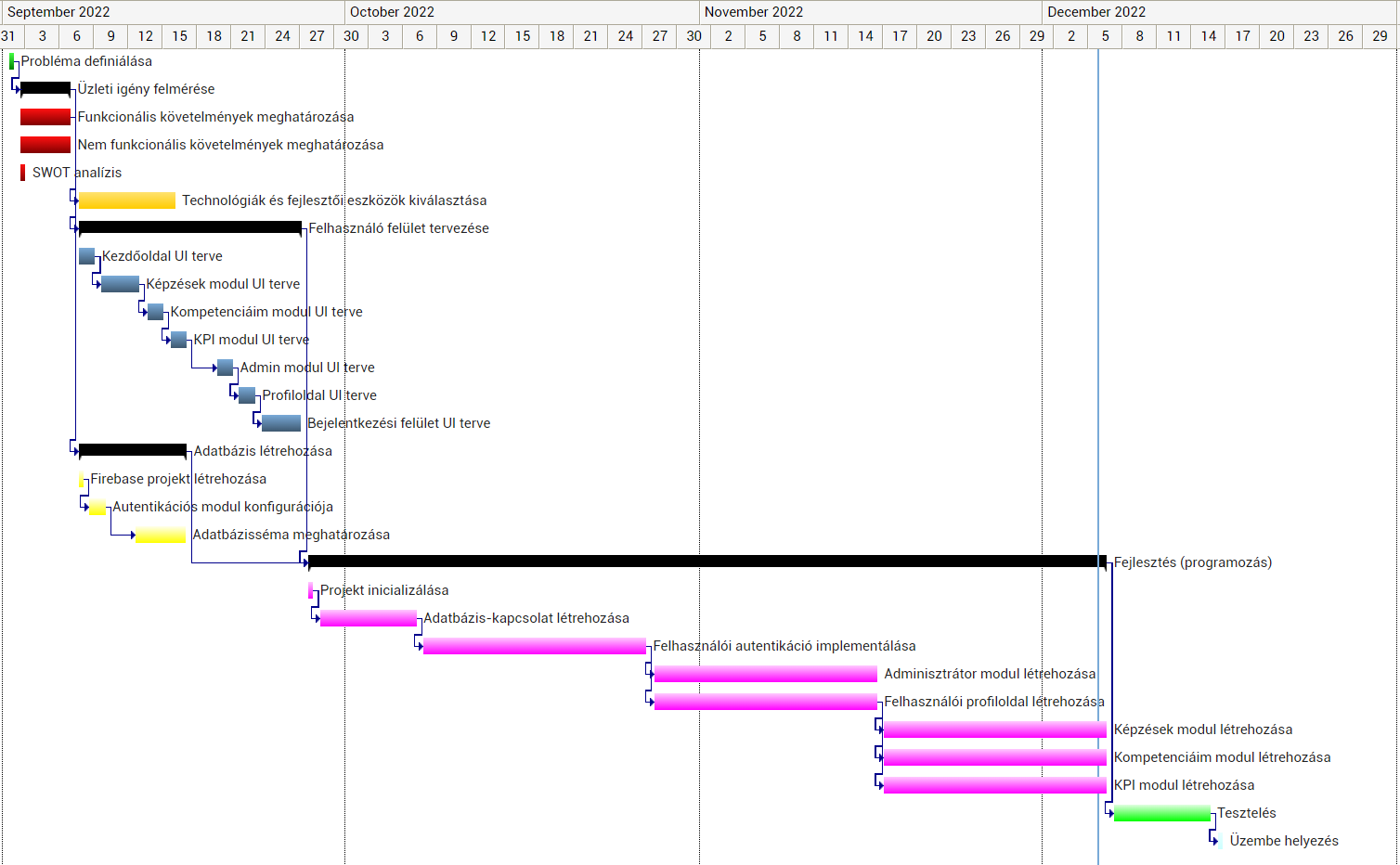
**Use Case diagram**



**Gantt diagram**

A képen asztal látható

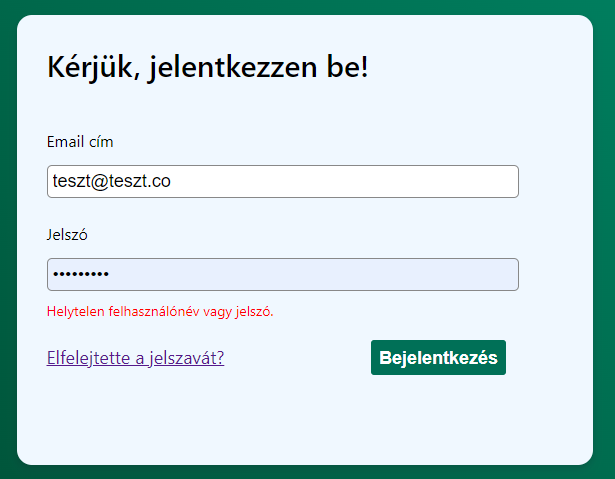
Automatikusan generált leírás

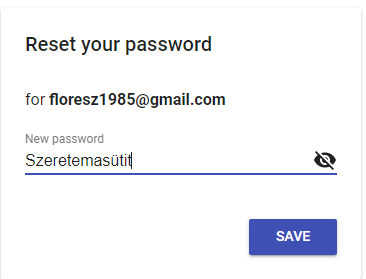


**Az egyes funkciók bemutatása**

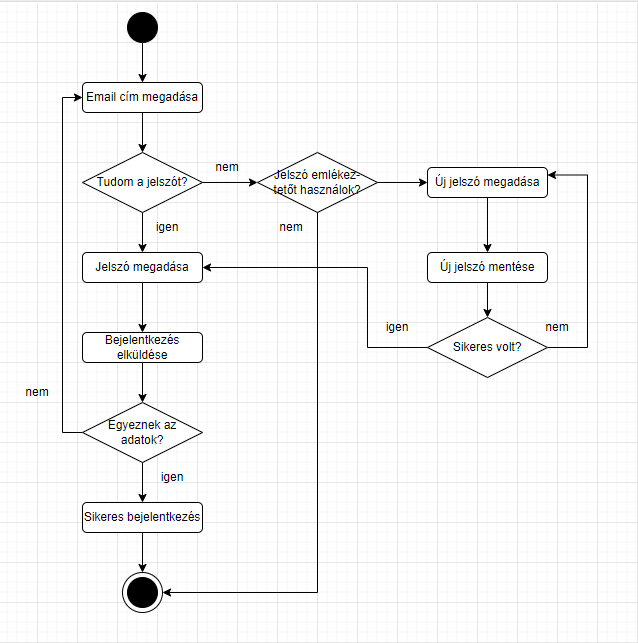
**Bejelentkezés:**

A portálra csak sikeres bejelentkezést követően lehet belépni. A felhasználók adatai egy adatbázisban kerülnek tárolásra, melyről később részletesebben is lesz szó. A bejelentkezéshez email cím és jelszó szükséges, mely két adat egyértelműen meghatároz egy konkrét felhasználót.

Sikertelen bejelentkezés esetén az oldal hibaüzenetet küld a felhasználó részére.

Amennyiben a felhasználó nem emlékezik a megadott jelszóra, lehetősége van jelszó-emlékeztetőt kérni. Ebben az esetben egy felugró ablak fog megjelenni, mely tájékoztatja a felhasználót, hogy jelszó-emlékeztető üzenet került a regisztrált email címére megküldésre. Az email – az alap tájékoztató szöveg mellett – egy linket tartalmaz, melyre kattintva egy új ablak jelenik meg a böngészőben. Itt van lehetőség az új jelszó megadására. Alapesetben – az adatok védelme érdekében – a jelszó rejtve jelenik meg. A felhasználónak a jobb oldalon található szem piktogram segítségével van lehetősége a begépelt szöveg helyességét ellenőrizni. Mivel ez a lehetőség adott, így jelszó ismétlést nem kér a rendszer. A „Save” gomra kattintva egy rendszerüzenet tájékoztatja a felhasználót arról, hogy a jelszó megváltozott.

A bejelentkezés folyamata az alábbi aktivitás diagrammal szemléltethető:



**Kezdőoldal:**

Sikeres bejelentkezés esetén a böngésző a kezdőoldalra navigál minket. A menüsor mellett az oldal közepén a képzésekhez, kompetenciákhoz kapcsolódó három legfontosabb kártya jelenik meg egymás mellett. Ha a kurzort rávisszük valamelyik kártyára, akkor vizuális effekt segítségével a kártya „kiemelkedik”. A navigálható felületek esetén a kurzor mutatója minden esetben megváltozik.



Az aktuális oldalról a menüsor, valamint a kártyák segítségével egyaránt el lehet navigálni. Az egyes oldalak esetében a bal felső sarokban megjelenő szöveg is az aktuális oldal megnevezésére fog változni. Emellett a menüsoron az éppen megjelenített oldal aláhúzással és eltérő színnel került megjelölésre. Mindkettő segíti a felhasználót a weboldalon történő könnyebb tájékozódásban.

**Képzések:**

Az oldalon az aktuálisan elérhető képzések listája jelenik meg táblázatos formában. A táblázat tartalmazza a képzés nevét, a képzés indulásának dátumát, a kategóriáját, valamint az aktuálisan jelentkezett személyek és maximum férőhelyek számát. Minden képzés rendelkezik egyedi azonosítóval, melyre kattintva a felhasználó – a fenti adatok mellett – elolvashatja a képzés részletes leírását is. Listanézetben a felhasználó a „Jelentkezem” gombra kattintva tud jelentkezni a képzésre. Amennyiben ez már megtörtént, akkor a jelentkezés helyett lejelentkezésre van lehetőség. Amennyiben a jelentkezők létszáma elérte a maximumot, akkor nem lehet jelentkezni a képzésre, melyet a rendszer egy nem kattintható „Betelt” gombbal jelez a felhasználó felé.

A táblázat utolsó oszlopai funkciógombok, melyek segítségével az adminisztrátor törölheti, vagy módosíthatja a képzést.

**Kompetenciáim:**

Az oldal részletes bemutatást ad a bejelentkezett felhasználó kompetenciáiról. Ez alapvetően négy kategóriába sorolható, az alábbiak szerint:

* Technikai ismeretek;
* Tanúsítványok;
* Sikeres képzések;
* Sikertelen képzések.

A megjelenítéshez – a könnyebb áttekinthetőség érdekében – itt is táblázatos formát használtunk.

Az első két kategória esetében lehetőség van új ismeret/tanúsítvány hozzáadására is, mely minden felhasználó részére elérhető. A funkciógombra kattintva a böngésző egy új oldalra navigál minket, ahol a szükséges adatok megadása után a kompetencia elmenthető. Lehetőség van a meglévő ismeretek/tanúsítványok a törlésére is, melyre az egyes kompetenciák végén található funkciógomb szolgál.

**KPI értékelések:**

Az oldalon legfelül az aktuális negyedévre vonatkozó adatait tekintheti meg a felhasználó. Ez alapján egyértelműen megállapítható, hogy melyik negyedévben vagyunk, mennyi az érintett dolgozó alap mozgóbére. A rendszer emellett azt is jelzi, hogy az érintett negyedév még aktív – folyamatban van –, valamint mennyi az eddig elért eredmény.

Ezt követi egy táblázat, melyben a negyedévre kitűzött célok felsorolása található. Ezek teljesítését – a jobb eredmény elérése érdekében – tehát még a felhasználónak végre kell hajtani.

Ezután van lehetőség a korábbi negyedéves értékelések áttekintésére is, mely a felhasználó részére visszajelzéssel szolgálhat a teljesítményéről az elmúlt időszakokban.

**Admin:**

Az egyes szerepkörök kialakításával ez a menüpont csak az adminisztrátorok számára lesz elérhető/kattintható. Az oldalra navigálva olyan további funkciók érhetők el, mint például az új felhasználó regisztrációja vagy új képzés létrehozása a rendszerbe. A szükséges adatok mindegyikének kitöltése szükséges ahhoz, hogy a „Létrehozás” gombra kattintva az érintett felhasználó/képzés bekerüljön a rendszerbe.

**Saját adatok karbantartása:**

Ennél a menüpontnál a rendszer a bejelentkezett felhasználó email címét fogja a menüsoron megjeleníteni. Az oldalra navigálva megjelennek a felhasználó adatai, mint név, törzsszám, beosztás és a munkaviszony kezdete. A felhasználónak lehetőség van profilkép beállítására is. Alapesetben egy „üres profilkép” kép jelenik meg. A profilkép megváltoztatása rész használatával az alapértelmezetten beállított kép lecserélhető, valamint a már feltöltött saját fotó is módosítható. Ezen kívül az oldalon belül is lehetőség van az aktuális jelszó módosítására. Ebben az esetben azonban az új jelszó mellett a régi jelszó megadása, valamint az új jelszó megismétlése is szükséges.

**Kijelentkezés:**

A menüsor legvégén egy funkciógomb került elhelyezésre, melyre kattintva a felhasználó kijelentkezhet az oldalról. Amennyiben a felhasználó rákattint, akkor a rendszer a bejelentkező felületre irányít vissza.

**Adatbázis szerkezet bemutatása**

Adatbázisként a Firebase saját adatbázisát, a Firestore-t használtuk. Ez egy NoSQL típusú adatbázis, tehát nem tartalmaz táblákat és kapcsolatokat, az adatok egyszerű JSON objektumokként vannak tárolva. A mi szempontunkból ez nagyon előnyös volt, mert nem kellett előre megtervezni az egész adatbázist, a frontendről olyan formátumú adatokat küldhettünk, amelyek éppen szükségesek voltak, és egyszerű volt bővíteni, módosítani. A JSON másik előnye, hogy kifejezetten a JavaScript-hez készült (JSON = JavaScript Object Notation), így beépített metódusokkal alakíthattuk a kapott adatokat JavaScript objektumokká, illetve küldés előtt JSON-ná.

A Firestore táblák helyett kollekciókat használ, amelyek dokumentumokat tartalmaznak, ez nagyjából megfeleltethető egy rekordnak SQL-ben. Egy dokumentumon belül definiálhatunk mezőket, amelyek bármilyen JS által használt adattípust tartalmazhat (szám, string, tömb, objektum).

Az adatbázisban külön kollekciókat hoztunk létre a felhasználóknak, a képzéseknek, a technikai ismereteknek, a tanúsítványoknak, és a KPI értékeléseknek.

A képen szöveg látható

Automatikusan generált leírás

**GUI tervek**

A design tervezésénél igyekeztünk egyszerű, letisztult, de mégis modern külsővel felruházni az oldalt. A tervek elkészítéséhez a Paint 3D programot használtuk. A bejelentkező felület mellett a kezdőoldal, valamint a profil oldal került megtervezésre. A képzések és kompetenciák esetében az adatok táblázatba szervezése mellett döntöttünk, így azokhoz nem láttuk szükségesnek külön látványterv készítését. Az új dolgok (pl.: felhasználók) rögzítése során pedig a bejelentkezési felület mintája volt az irányadó.





