Felhasználói dokumentáció

Az alkalmazás használatához a felhasználóknak rendelkezniük kell gmail-es email címmel, vagy Google+ fiókkal, mivel a felhasználói autentikáció a Google API konzolon keresztül történik. Sikeres belépés után az oldal átirányít a felhasználó saját profiljára, ahol megjelennek az adatai. Ennek a fajta regisztrációnak / belépésnek nagy előnye, hogy egy biztonságos és megbízható szolgáltatást tudunk igénybe venni, így nekünk például nem kell a jelszó titkosításáról gondoskodni, ezt elvégzi a Google API. Mivel minden adat szinkronizálódik a Google profilban beállítottak alapján, így nincs szükség külön profilképet feltölteni, vagy egyéb fontos adatokat megadni, ezek automatikusan jönnek.

A szerver feladata fogadni az adatokat a frontend felől HTTP protokollon keresztül, valamint az adatbázissal való kapcsolatok és lekérdezések kezelése és végrehajtása. A felhasználók sikeres bejelentkezés után módosíthatják a saját pozíciójukat, csapatokat hozhatnak létre és törölhetnek. Ezekhez csatlakozhatnak, valamint kiléphetnek belőlük. Közösen ötletelést végezhetnek a többi fejlesztővel. Saját és csapatszintű metrikákat generálhatnak, üzeneteket küldhetnek egymásnak a csapaton belül. Visszatekintést tarthatnak egy adott fejlesztési ciklusra, és három szempont alapján véleményeket nyilváníthatnak, észrevételeket tehetnek. Ezeket a funkcionalitásokat http protokollon és API hívásokon keresztül kommunikálja le a frontend a szervernek, ami végrehajtja a változtatásokat az SQL adatbázisban, és ezeket a változásokat közvetítheti a frontend felé, folyamatosan szinkronban tartva az adatbázis réteget a megjelenítési réteggel.

Rendszerkövetelmények:

Szerver oldali követelmények

* Operációs rendszer
  + Microsoft Windows 10
* Hardware
  + Processzor: >= 2.6 GHz
  + Memória: >= 8GB
  + Háttértár: >= 30 GB
* Adatbázis
  + MySQL server: >= 8.0
* IDE
  + Visual Studio Code >= 1.38
  + MySQLWorkBench >= 8.0.16 (64 bit)
* SDK
  + Node.js + NPM (Node Package Manager)
* NPM csomagok
  + Express
  + cors
  + passport
  + passport-google-oauth20
  + chalk
  + body-parser
  + sequelize
  + concurrently
  + mysql2
  + http-proxy-middleware

Kliensoldali követelmények:

A frontend alkalmazás tesztelése és fejlesztése React.js és Material-ui környezetben, keretrendszerben valósult meg. Ezek lehetővé teszik dinamikus felhasználói felületek létrehozását. Jelen kliens bármelyik modern webböngészőn futtatható és elérhető. A frontend rendszer teszteléséhez és tervezéséhez javasolt követelmények:

* Operációs rendszer
  + Microsoft Windows 10
* Hardware
  + Processzor: >= 2.6 GHz
  + Memória: >= 8GB
  + Háttértár: >= 30 GB
* IDE
  + Visual Studio Code >= 1.38
  + Google Chrome
  + React Develper Tools (Google Chrome kiegészítő)
* SDK
  + Node.js + NPM (Node Package Manager)
* NPM csomagok
  + react
  + create-react-app
  + @material-ui/core
  + react-chartjs-2
  + react-dom
  + react-router-dom
  + react-scripts
  + react-swipeable-views

Telepítés:

Adatbázis telepítése és konfigurálása

Frontend és Backend alkalmazás telepítése és konfigurálása

A frontend telepítése pár egyszerű lépésből áll.

1. Másoljuk át a DVD-n található teljes projekt mappát.
2. CMD használatával navigáljunk el a projekt gyökér könyvtárába, ahol a *package.json* fájl található.
3. Ha megbizonyosodtunk róla, hogy van feltelepített Node.js modul a számítógépünkön, Node Package Managerrel, akkor a következő *„npm install”* parancs kiadása után az NPM felinstallálja a szükséges függőségeket. Ez akár több percig is eltarthat.

Az alkalmazás felépítése és használata

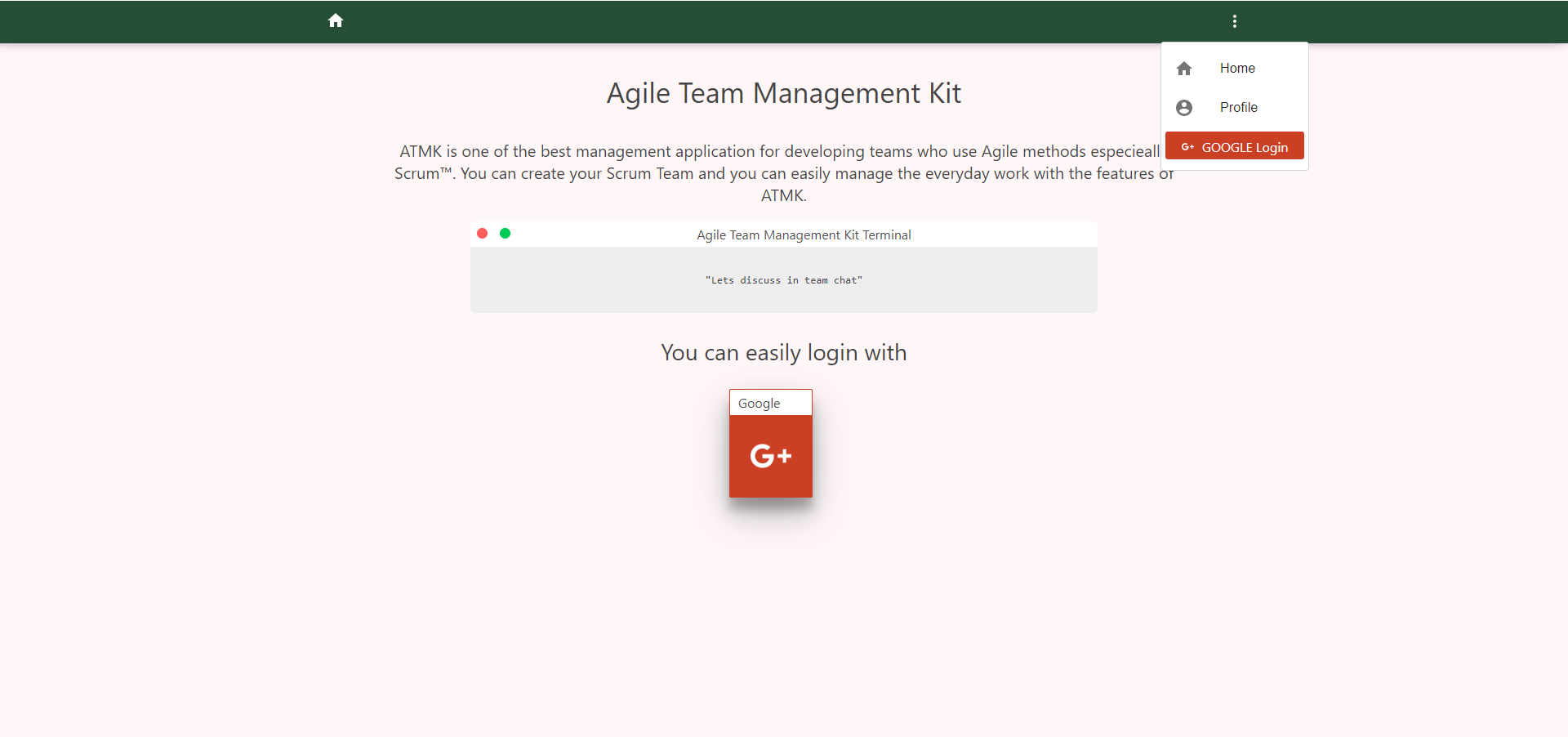
Az alkalmazás elindítását követően bejelentkezés és regisztráció nélkül csak a főoldal érhető el. A többi oldalnak előfeltétele, hogy a felhasználó be legyen jelentkezve.

Az alkalmazás a következő oldalakból áll:

* Főoldal / Home
* Profil oldal / Profile
* Csapatok oldal / Teams
* Ötletek oldal / Ideas
* Visszatekintés / Retrospective
* Metrikák és kimutatások / Metrics
* Áttekintés / Dashboard
* Üzenő fal / Message Board

**Főoldal / Home**

A főoldal mind a regisztrált és nem regisztrált látogatók számára elérhető. A főoldal nem rendelkezik navigációs listával, mivel azok bejelentkezéshez kötött funkciók. Ami itt elérhető és látható, az egy folyamatosan változó css-ben formázott konzolablak, ami előre táplált üzeneteket közvetít eltérő időközönként, valamint az alkalmazás leírása található itt. A főoldal rendelkezik egy Google Login logóval, amire kattintva a rendszer átirányít a Google API konzolra.



1. ábra Agile Team Management Nyitóoldala

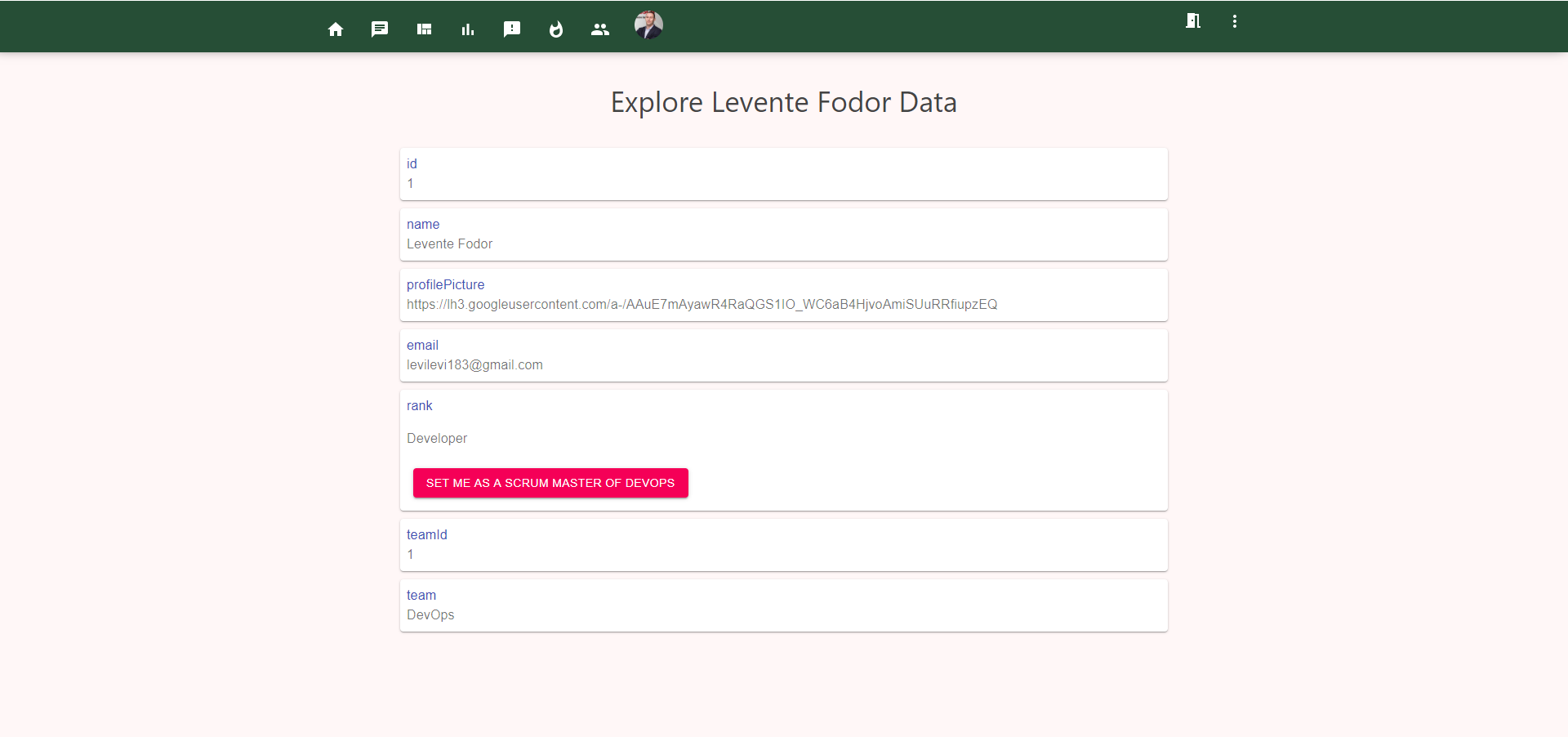
**Navigációs sáv**

C:\Users\Levi\Pictures\Screenpresso\2019-12-04_00h55_14.png

2. ábra Navigációs sáv

**Profil Oldal / Profile**

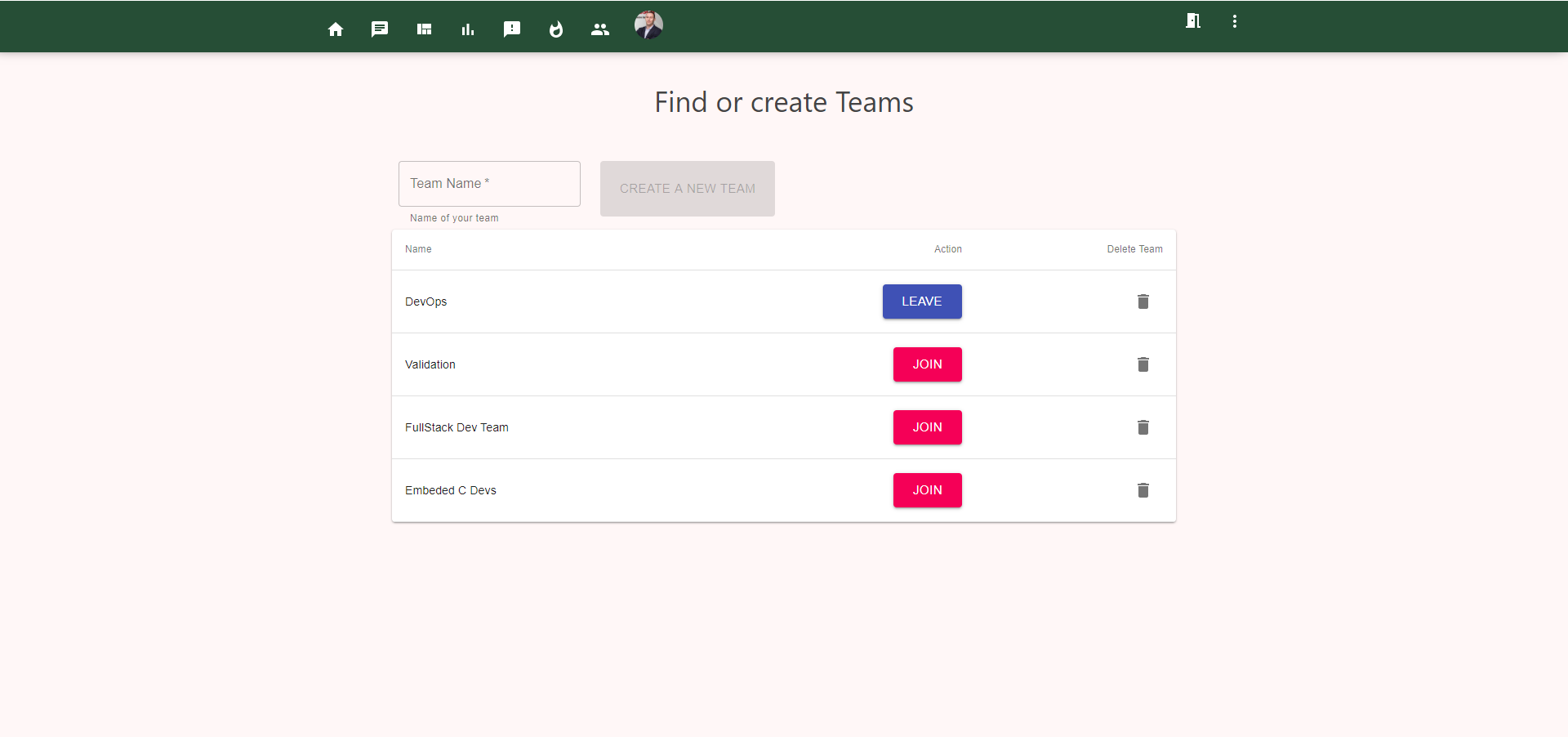
A felhasználó a belépést és sikeres regisztrációt követően ezt az oldalt látja. Itt találhatóak a felhasználó releváns adatok táblázatos formában vizualizálva, ezek az ID, név, profilkép url címe, email cím, rang, csapatId és csapatnév. Két rangot kezel az alkalmazás. Developert és Scrum Mastert. Kezdeti értékként minden felhasználó Developer rangot kap. Ha a felhasználó csatlakozott egy már létező csapathoz és a rangja Developer, ez esetben megjelenik egy „*Set me as a Scrum Master of <kiválasztott csapat neve>”* gomb, amire rákattintva az adott felhasználó rangja Developerről Scrum Masterré változik. Erre később azért van szükség, mert bizonyos funkciók csak Scrum Mastereknek érhetőek el.



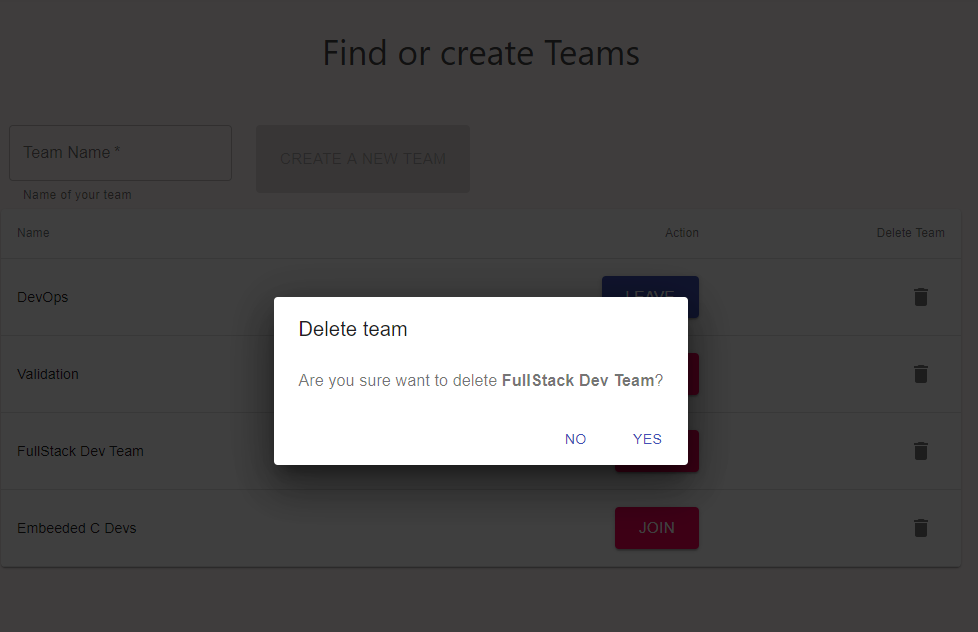
3. ábra Profil oldal

**Csapatok Oldal / Teams**

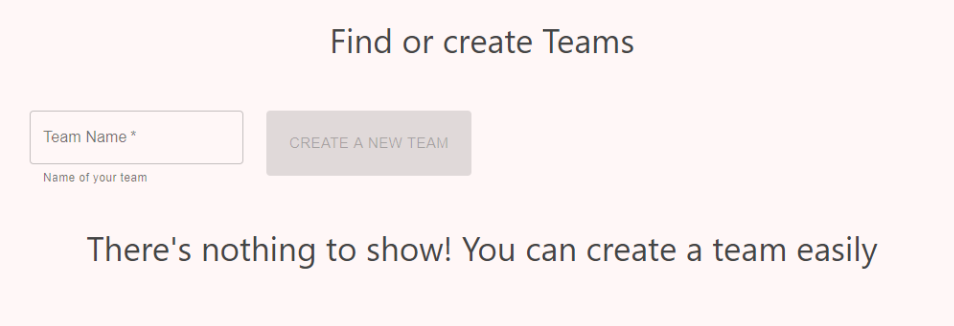
A csapatok oldalra navigálva, két dolgot láthatunk a képernyőn. Ha még egyetlen egy csapat sem lett regisztrálva a rendszerbe, akkor egy üzenet jelenik meg, hogy még nincsenek regisztrált csapatok, de bárki könnyedén létre tud hozni csapatot. Miután legalább egy csapat létrejött Ez a szöveg eltűnik, és helyette táblázatos formában láthatjuk egymás alatt a csapatokat. A táblázat egy sora tartalmazza a csapat nevét, egy akció gombot, valamint a törlés gombot. A csapat neve egyedi, eszerint nem lehetséges két ugyanolyan névvel rendelkező csapatot létrehozni. Sikeres létrehozás után a rendszer egy felugró üzenet ablakkal jelzi, a létrehozás sikerességét, valamint a dinamikus komponensek és az adatbázis szinkronizáció miatt azonnal láthatjuk az új csapatot a táblázatban. Egy felhasználó egyszerre egy csapatnak lehet tagja. A második mezőben lévő akciógomb segítségével csatlakozni lehet egy csapatba, vagy éppen elhagyni azt. A harmadik mezőben lévő törlés ikonra kattintva a csapat törlésre kerül az adatbázisból. Ezt egy felugró ablak előzi meg, ami megerősítést kér a felhasználótól törlés előtt. Fontos, hogy bizonyos elemek vizuális elkülönüljenek egymástól, így ha már tagjai vagyunk egy csapatnak, akkor a csapat elhagyása gomb a csatlakozás gombhoz képest eltérően néz ki, ezzel is elkülönülve a többi csapattól, jól láthatóvá téve melyik csapatnak vagyunk a részesei.



4. ábra Csapatok oldal



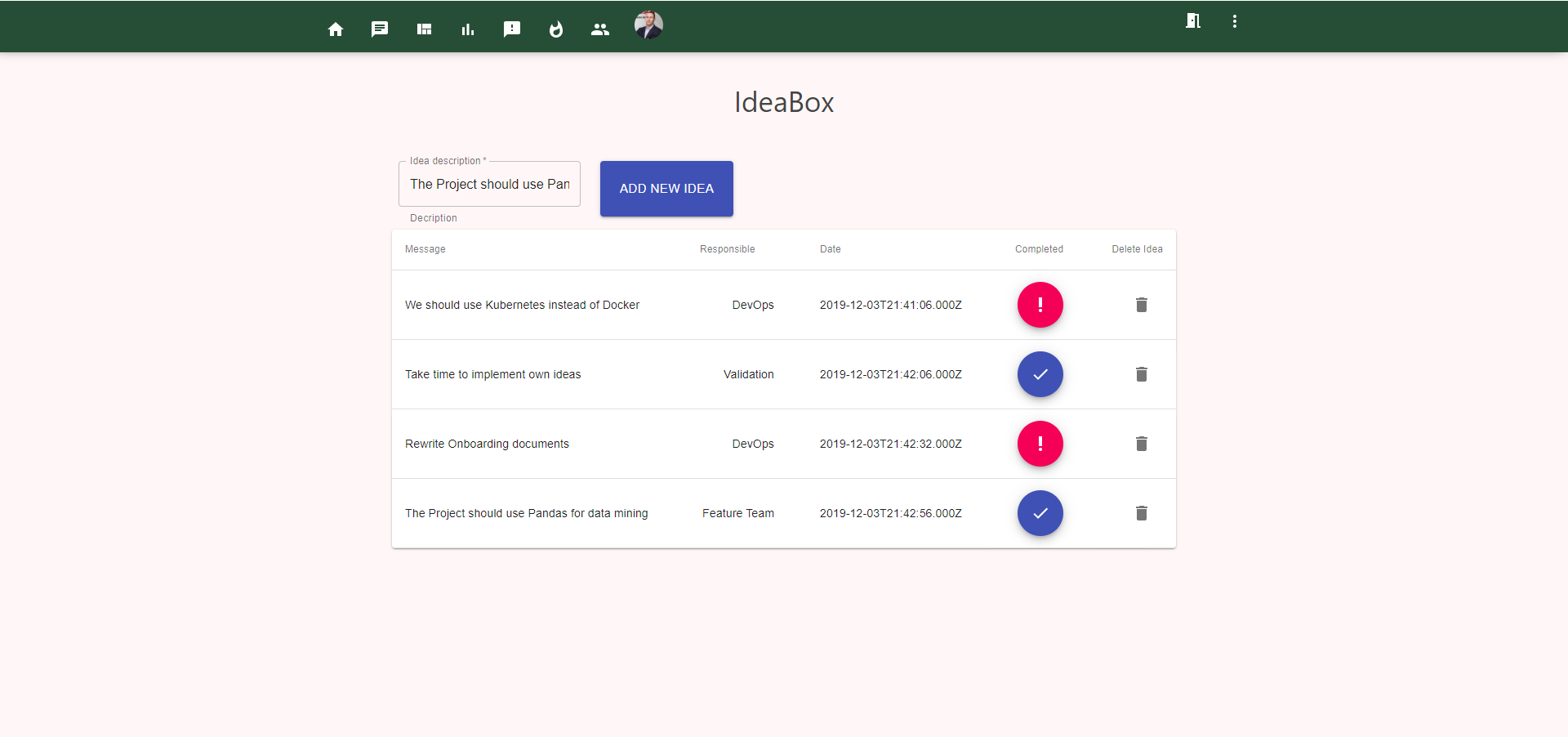
5. ábra Csapat törlése előtti megerősítő ablak



6. ábra Csapat oldal tartalma, ha még nincs csapat létrehozva

**Ötletek Oldal / Ideas**

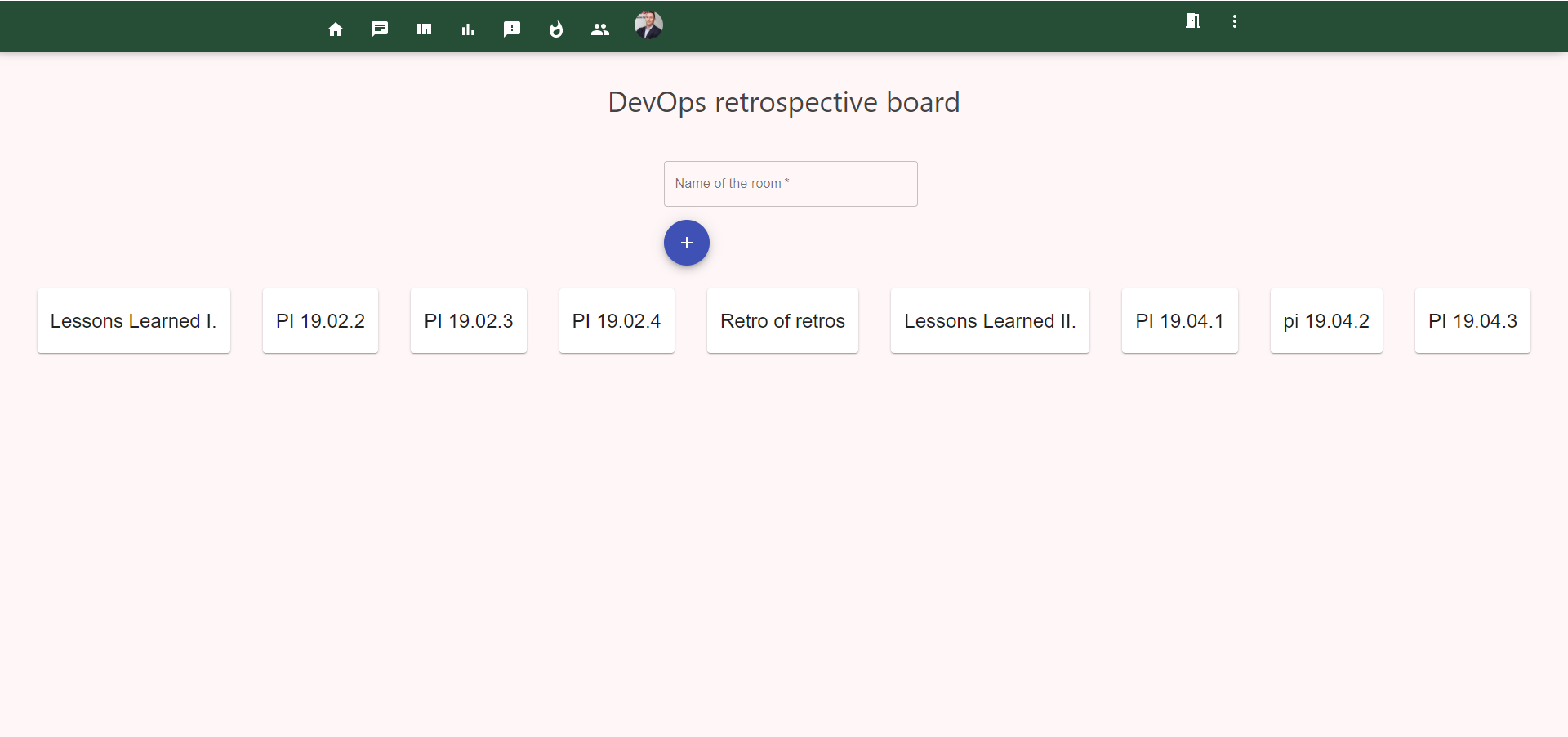
Az ötletek oldal megnyitása után szintén két felület kerülhet megjelenítésre. Ha a rendszerben még egyetlen egy ötlet sincs definiálva, akkor egy tájékoztató szöveg íródik ki a képernyőre. Ellenkező esetben megjelennek az ötletek egymás alatt táblázatos formában. Egy sor öt darab mezővel rendelkezik. Az első a név, ami az ötlet leírása, ezt követi a csapatnév, ami az ötletet megalkotó felhasználó csapatának a neve. Utána jön az ötlet megalkotásának dátuma, ezt követi két funkciógomb, ezek közül az első vizuálisan eléggé eltér a környezetétől. Ez jelzi, hogy egy ötlet elkészült e már, vagy még nem. Ha még nincs megvalósítva, akkor egy piros felkiáltó jel található ebben a mezőben. Ha az ötlet megvalósításra került, akkor rákattintva az előbbi felkiáltó jel ikonra az megváltozik kék pipa ikonná, ezzel jelezve, hogy be lett fejezve. Az utolsó cellában található törlés ikon segítségével lehet ötletet törölni.



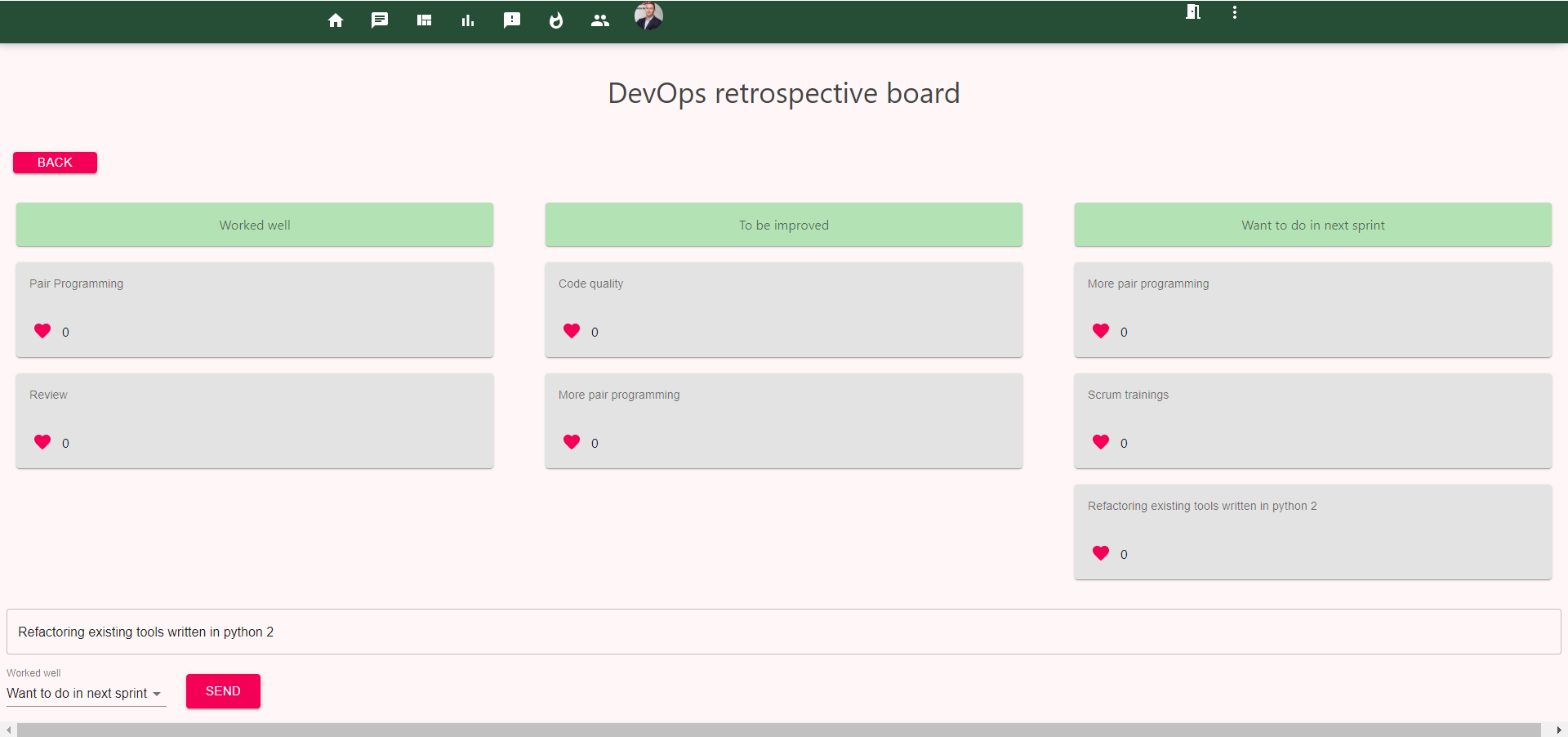
7. ábra Ötletek oldal, a már létező ötletek listájával

**Visszatekintés Oldal / Retrospective**

A visszatekintés, vagyis más néven retrospective oldalra navigálva láthatjuk a saját csapatunknak a retrospective tábláját. Ezek a kártyák különböző már létrehozott szobák nevei. Lehetőség van új szoba létrehozására az oldal tetején található mező és gomb segítségével. Rákattintva az egyik szoba kártyára, egy új felületen találjuk magunkat. Ez maga az adott retrospective szoba, ami három csoportra van osztva. Ezek a *„worked well” „To be impoved”*, és a *„want to do in the next sprint”*. A lentebb található szövegmezőt használva, valamint az az alatt elhelyezkedő legördülő menü alapján kiválasztva az adott csoporthoz megjegyzéseket tudunk hozzáfűzni. Ezek adatbázisba mentődnek, a későbbi visszakereshetőség érdekében. A *back* gomb segítségével vissza tudunk navigálni a retrospective főoldalra.



8. ábra Visszatekintés / Retrospective oldal, és a rajta található szobák nevei

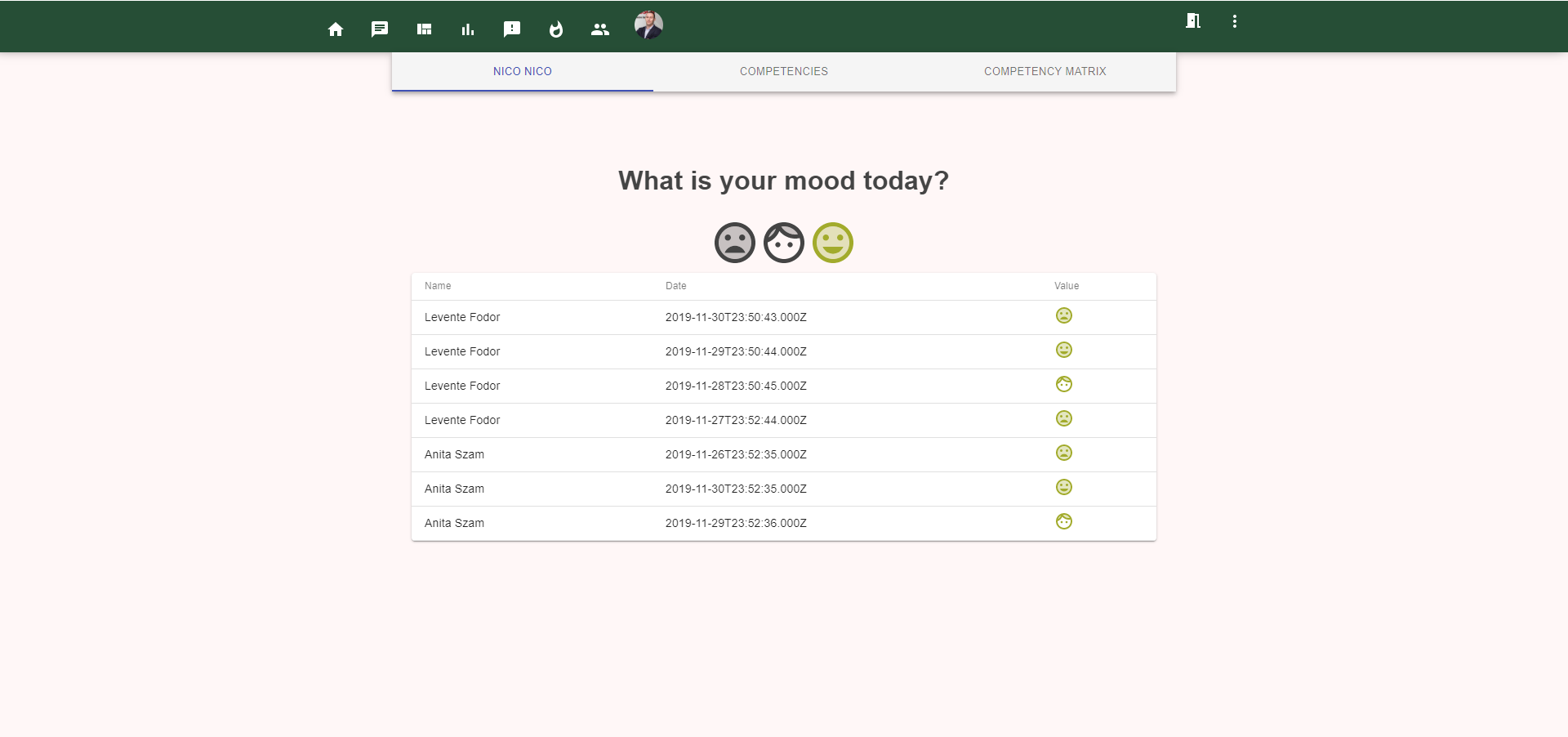


9. ábra Kiválasztott retrospective szoba, a három tulajdonság alatt, megjegyzésekkel feltöltve

**Metrikák és kimutatások Oldal / Metrics**

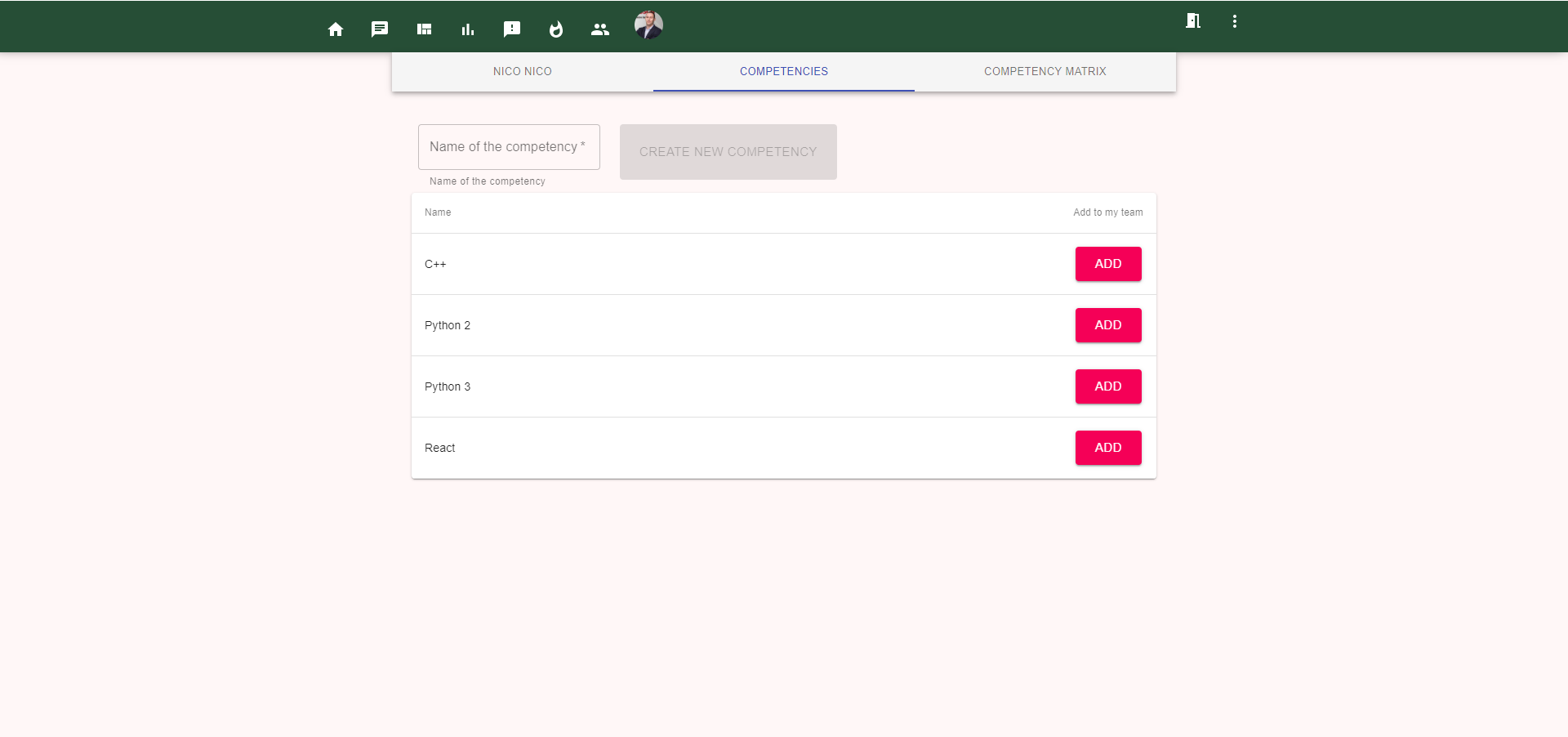
Ez az egyik talán legérdekesebb oldal, mivel a többitől eltérő navigációval rendelkezik. Amit a felhasználó lát, az a saját csapatára releváns összes kimutatás és metrika. Az első oldala Nico Nico táblázat, ahol a felhasználók három darab smiley formájában kattintással tudják a rendszerben vizualizálni az aznapi hangulatukat, és motiváltságukat. Itt táblázatos formában egymás alatt jelennek meg a csapattagok a saját értékeikkel.

A Nico Nico táblázat első oszlopában a felhasználók nevei, a másodikban az aznapi dátum, a harmadikban pedig az aznapi hangulatuk jelenik meg vizuális formában. Új rekordot létrehozni a táblázat felett található három smiley egyikére való kattintással lehet. Ezek jelentése egyértelmű, szomorú, neutrális, és boldog.



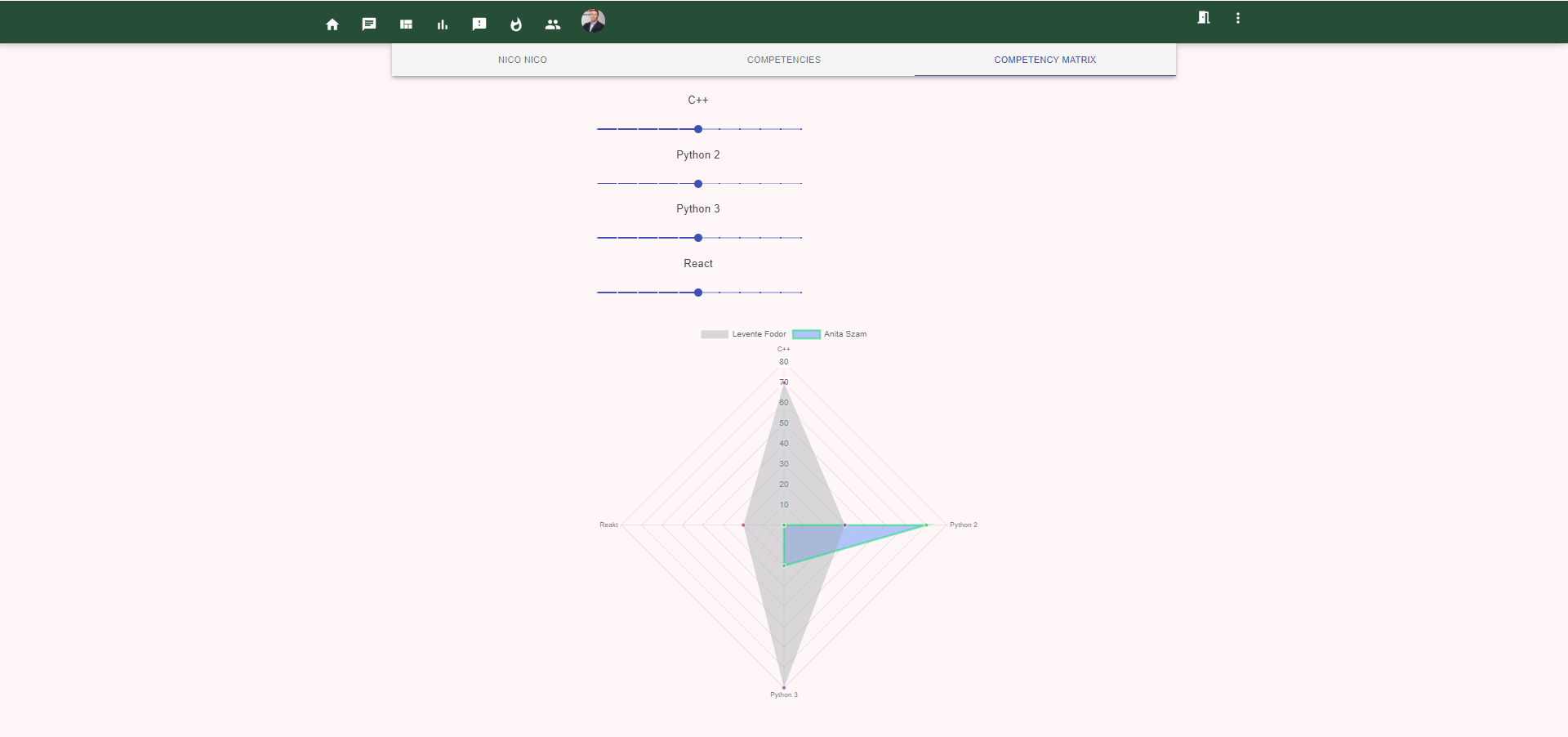
10. ábra Metrics oldal első navigációs pontja, a NicoNico tálázat.

A második kompetenciák menüpontra kattintva a végkifejlet kétféle lehet. Itt már fontos a rang, hiszen ezen a felületen lehet kompetenciákat létrehozni és az adott csapathoz rendelni. Ez a funkcionalitás csak Scrum Mastereknek érhető el. Ha egy Developer rangban lévő felhasználó navigál erre az oldalra, a rendszer egy üzenettel jelzi, hogy számára ez a funkcionalitás nem érhető el. Scrum Master felhasználó esetében egy szövegmező és a hozzá tartozó gomb, valamint egy a már létező kompetenciák vizualizálására szolgáló táblázat jelenik meg. Ezen táblázat első eleme a kompetencia neve, a második egy akciógomb, ami segítségével az adott csapathoz hozzá lehet rendelni a kompetenciát.



11. ábra Kompetenciák rész a Metrics oldalon belül

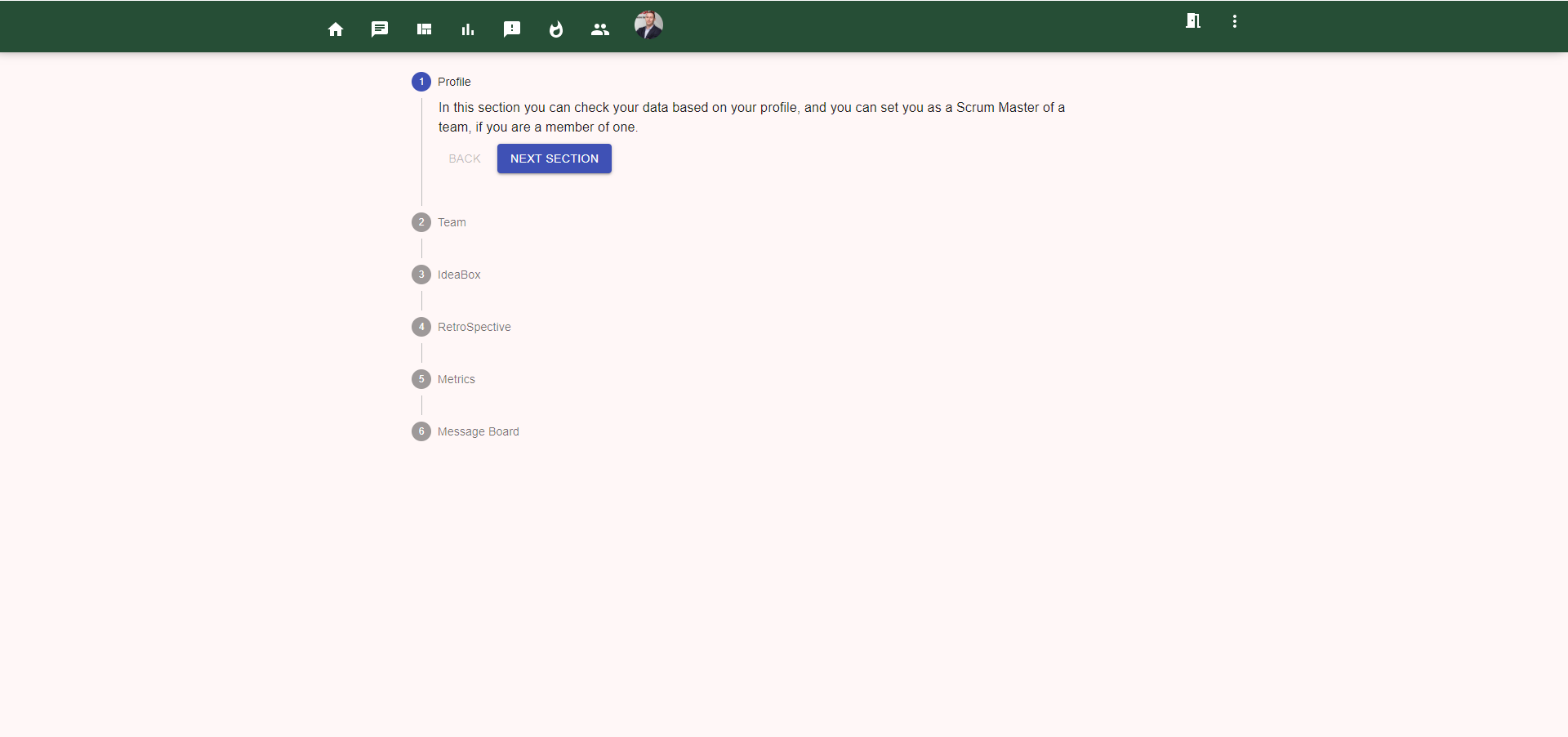
A harmadik oldalon található a kompetencia mátrix, ami egy komplex sugárdiagram a csapathoz rendelt kompetenciák és a csapattagok szerint. A diagram felett található intervallumos jelölő formájában a csapathoz rendelt kompetenciák. Ennek a csúszkának a segítségével tudjuk az adott kompetenciát egy 0 és 100 közötti intervallumon a profilunkhoz rendelni. Beállítás után az adataink és a többi csapattag adatai megjelennek a diagramon, egy az egész csapat kompetenciáira vonatkozó általános mégis reprezentatív kimutatás formájában. Szűrésre a csapattagok nevére való kattintással van lehetőség, ekkor a kiválasztott csapattagot le lehet venni a táblázatról, valamint újra vissza lehet rakni.



12. ábra Kompetencia mátrix a csapattagok és a definiált kompetenciák alapján

**Áttekintés / Dashboard**

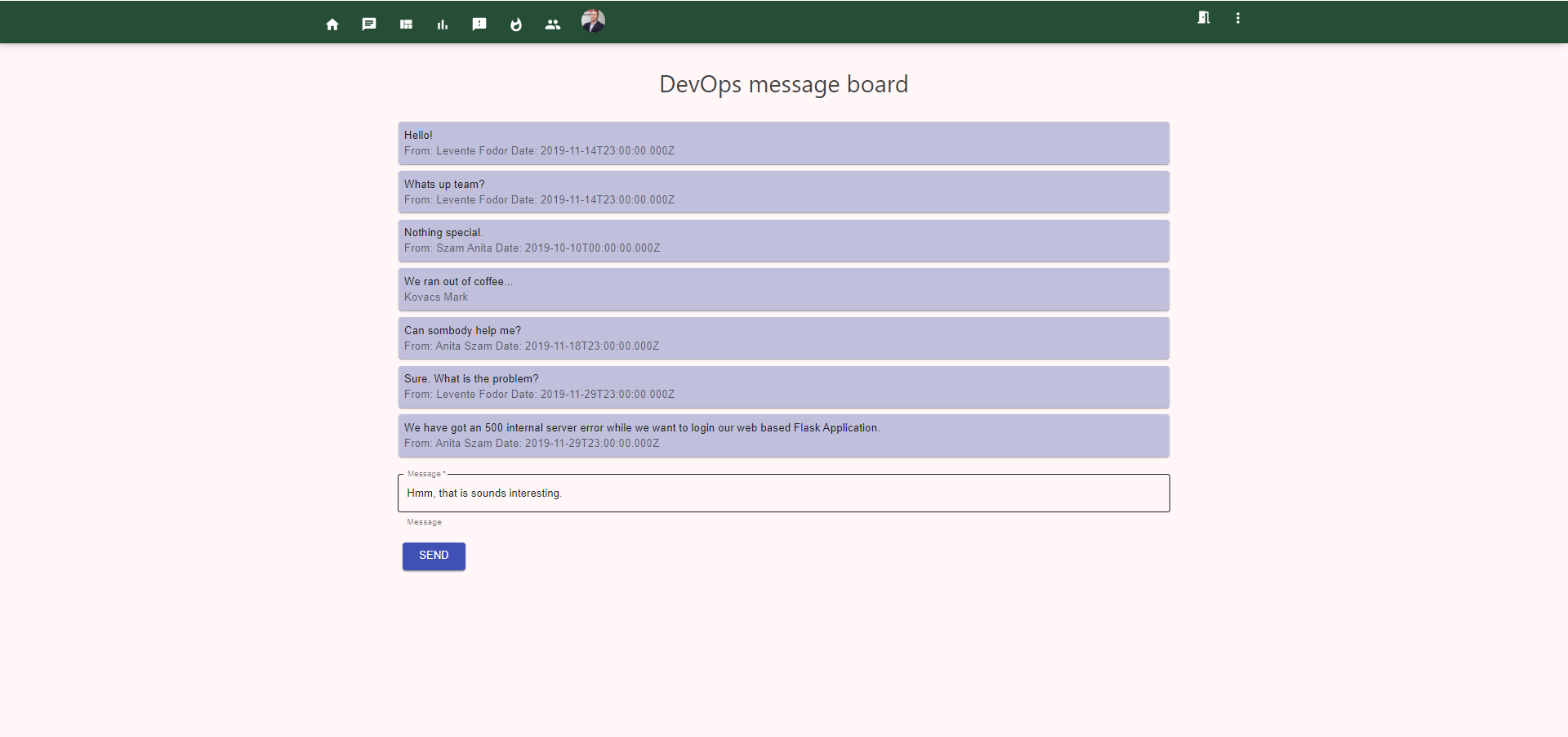
Az áttekintés oldalra navigálva egy lépésről lépésre típusú végigvezetés tárul elénk, ami segítségével képet kaphatunk az alkalmazás funkcionalitásait illetően.



**13**. ábra Áttekintés oldal

**Üzenőfal / Message Board**

Az üzenőfal oldal információ átadás elősegítésére szolgál a csapattagok között. Az üzeneteket a csapat összes tagja látja, valamint a csapat összes tagja tud üzenetet küldeni. Az üzenet megjelenítésének formája tartalmazza magát az üzenetet, valamint a feladó nevét, és a küldés időpontját. Az oldal alján található szövegmező és gomb segítségével könnyedén adhatunk hozzá új üzenetet.



14. ábra Üzenőfal oldal, és a benne tárolt üzenetek

**Frontend oldali követelmények**

dass