A képen clipart, macska, illusztráció, rajzfilm látható

Automatikusan generált leírás

**Széchenyi István Katolikus Technikum és Gimnázium**

**Szoftverfejlesztő és -tesztelő projektfeladat**

**„REpont vizsgafeladat”**

Készítették:

Kriston-Széles Flórián,

Krizsai Tamás,

Kanta Áron

**Ózd, 2025**

Tartalomjegyzék:

1. Bevezetés
2. Felhasználói dokumentáció
   1. Rendszerkövetelmény
      1. Hardver
      2. Szoftver
   2. Webalkalmazás indítása
   3. Webalkalmazás használata
      1. Általános információk a webalkalmazásról
      2. Főoldal leírása + képernyőképek
      3. Aloldal1 leírása + képernyőkép
      4. Aloldal2…
3. Fejlesztői dokumentáció
   1. Alkalmazott fejlesztői és csoportmunka eszközök
   2. Adatbázis
   3. Frontend
   4. Backend
4. Adatbázis
   1. Modell leírása (adatmezők – típusok - leírásuk)
   2. Táblák, kapcsolatok
   3. E-K diagram
5. Frontend
   1. Alkalmazás frontend részének részletes leírása + képernyőfotók
6. Backend
   1. Alkalmazás backend részének részletes leírása + képernyőfotók
7. Tesztelés
8. Továbbfejlesztési lehetőségek
9. Irodalomjegyzék
10. **Bevezetés**
11. **Felhasználói dokumentáció**
12. **Fejlesztői dokumentáció**
13. **Alkalmazott fejlesztői és csoportmunka eszközök**

**Git/GitHub:** Project tárolása és verziómenedzselés

**Discord:** Kommunikáció

1. **Adatbázis**

**Visual Studio Code:** SQL kód megírása

**XAMPP:** SQL kód tesztelése, Apache webszerver futtatása

**MySQL:** Itt van tárolva maga az adatbázis

1. **Frontend**
2. **Backend**

**Visual Studio Code**: Laravel api, kontrollerek megírása és tesztelése

**XAMPP**: Adatbázis lokális futtatása és kezelése

**Laravel**: Keretrendszer az apihoz

1. **Adatbázis**
2. Az adatbázis azzal a céllal lett létrehozva, hogy a REpont gépeknek a nyomon követésében, és a végfelhasználóknak ezeknek a gépeknek a hely alapú mutatásában segítsen. Könnyen meg lehet tudni egy gép jelenlegi adatait, például melyik településen van, a pontos címét, illetve milyen szinten van megtelítve. Azon kívül, van még egy másik tábla, melyben különböző teljesítményekért ún. ’achievement’-eket lehet elérni, amelyek x mennyiségű üveg beváltásáért ajándékokkal jutalmazzák meg a felhasználót, pl. kuponok különböző boltoknál. Van egy tábla, melyben Magyarország települései és azoknak az irányítószámai vannak eltárolva, ez a többi adattábla könnyebb összeköttetésére szolgál. Véglegesen pedig a felhasználók is el vannak tárolva az adatbázisban, felhasználónév, jelszó, a felhasználó címe, hogy tudjuk melyik gépek vannak hozzá a legközelebb stb.
3. **Táblák Bemutatása:**
4. **Települések**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mező | Rövid Leírás | Adattípus | Kulcs típusa | Egyéb specialitások |
| zipcode | Itt a települések irányítószáma van tárolva | int | PRIMARY KEY | NOT NULL, UNIQUE |
| settlementname | A települések neve | varchar  (75) |  | NOT NULL |

1. **Gépek**

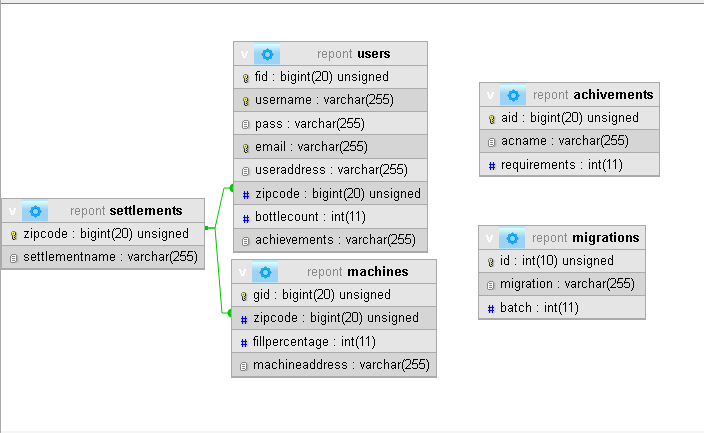
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mező | Rövid Leírás | Adattípus | Kulcs típusa | Egyéb specialitások |
| gid | Itt a gép azonosító száma van tárolva | int | PRIMARY KEY | NOT NULL, AUTO\_INCREMENT |
| zipcode | A gépnek a tartozási helyének az irányítószáma | int | FOREIGN KEY | NOT NULL |
| fillpercentage | A gépnek a töltöttségi százaléka | int |  |  |
| machineaddress | A gépnek a pontos címe | varchar (100) |  | NOT NULL |

1. **Felhasználók**

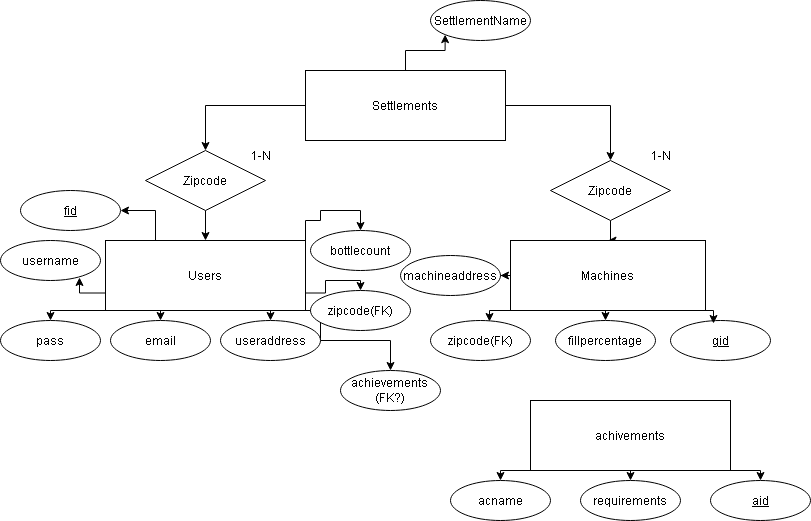
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mező | Rövid Leírás | Adattípus | Kulcs típusa | Egyéb specialitások |
| fid | **Itt a felhasználó azonosító száma van tárolva** | **int** | **PRIMARY KEY** | **NOT NULL, AUTO\_INCREMENT** |
| username | **A végfelhasználóknak a felhasználóneve** | **varchar**  **(30)** |  | **NOT NULL, UNIQUE** |
| pass | **A felhasználónak a jelszava van itt eltárolva** | **varchar**  **(125)** |  | **NOT NULL** |
| useraddress | A felhasználó a pontos címe | varchar  (100) |  |  |
| zipcode | A felhasználó irányítószáma | int | FOREIGN KEY | NOT NULL |
| bottlecount | Az újrahasznosított üvegek száma | int |  | NOT NULL |
| achivements | Az elért achivementek listája | varchar (250) |  |  |
| email | A felhasználó e-mail címe | varchar (65) |  | NOT NULL UNIQUE |

1. **Achievementek**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mező | Rövid Leírás | Adattípus | Kulcs típusa | Egyéb specialitások |
| aid | Az achievement azonosítója van itt | int | PRIMARY KEY | NOT NULL, AUTO\_INCREMENT |
| acname | Az achievement neve | varchar  (75) |  |  |
| requirements | A kötelező üvegszám az eléréshez | int |  | NOT NULL |

**Kapcsolat ábra:**

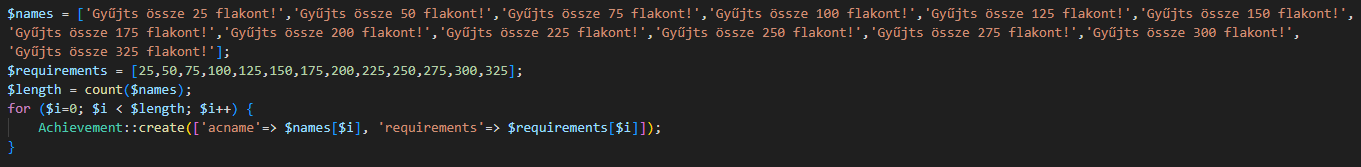
1. **ER diagram:**

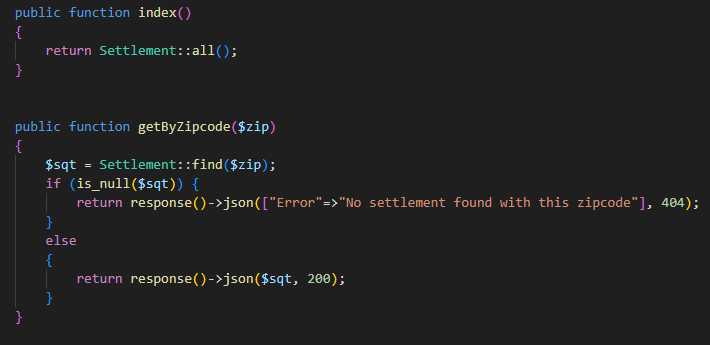
****

1. **Frontend**
2. **Backend**
   1. A screen shot of a computer program

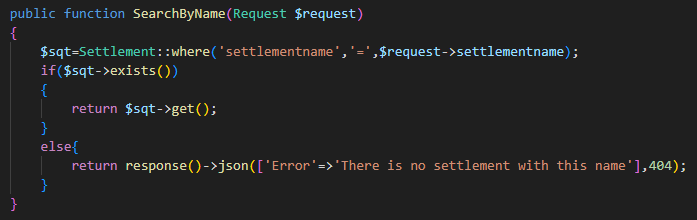
      Description automatically generatedA backend feladata az adatbázis és a frontend közötti zökkenőmentes kapcsolat létesítése és fenntartása. A lehető legtöbb feladat ellátására lettek apik létrehozva, hogy a későbbiekben már ne kelljen nagymértékű változtatásokat eszközölni a kódban. A cél az volt, hogy a felhasználó a frontenden keresztül minél gyorsabban tudjon ténykedni.

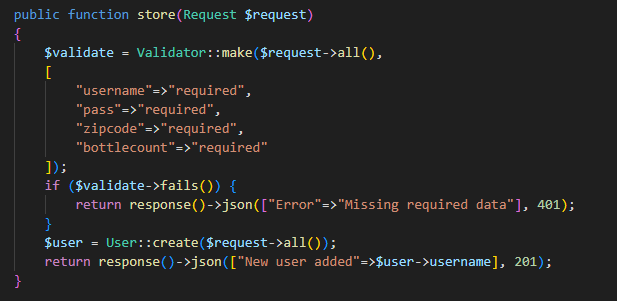
Ezen a képen fellelhető az api.php, amiben benne vannak a megírt végpontokhoz vezető elérési útvonalak. A frontend kódja erre hivatkozik és a felhasználó vagy adminisztrátor által elrendelt műveletet végrehajtja.

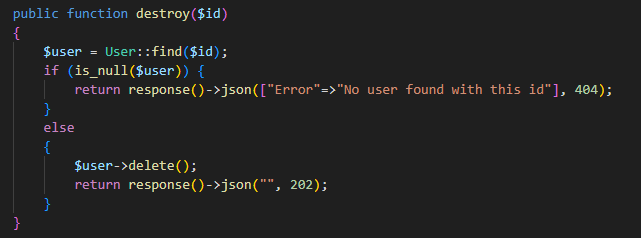


Itt látható a databaseseeder, ennek a feladata alapvetően a felmigrált adattáblák feltöltése. Ez esetben az achievementek adattáblát tölti fel előre meghatározott adatokkal. A többi adattáblát külső sql fájból importáljuk, az adatok nagy mennyisége miatt.

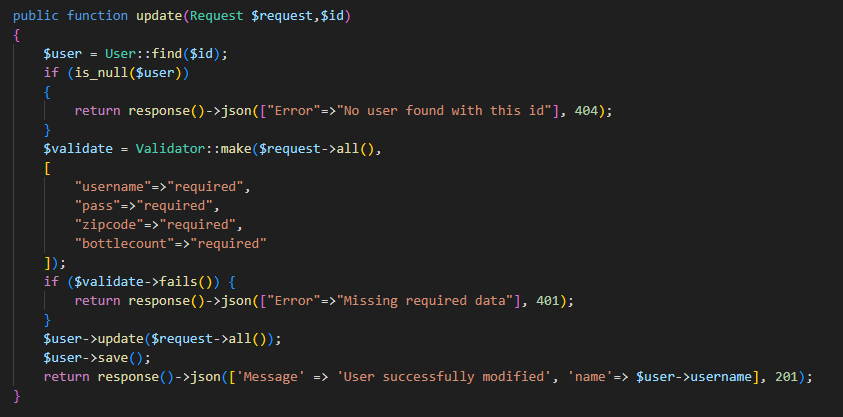
Ebben a képernyőfotóban leledzik az index és a getById metódus (itt getByZipcode) E jelenlegi specifikus szituációban a településekhez tartozó Settlement Controller kódrészlete tárul szemünk elé. Ennek segítségével hívható le az összes adat a szóban forgó adattáblából, vagy ha csak egy adott elemre van szükség, kivitelezhető az arra való keresés, táblától függ, hogy milyen adat alapján, ez lehet irányítószám, azonosító, cím stb.

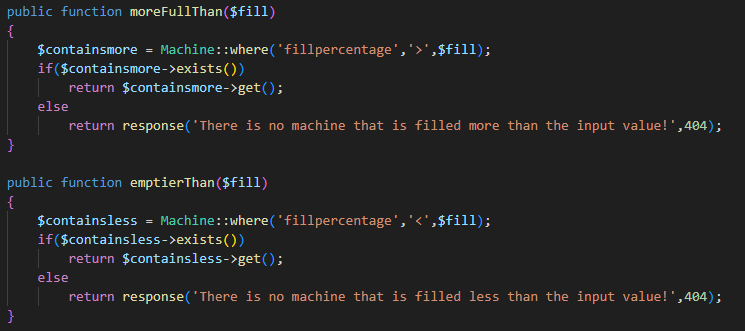
****Lent látható a search metódus. Célja ugyanaz, mint a getById eljárásnak, a különbség, hogy más alapján keres és enyhén másként kell használni.

A fent lévő, szemünk által perceptuálható vizuális manifesztáció lényegi tartalma a store metódus. Ez létfontosságú eleme a backendnek, különösen az User Controllerben. Enélkül a teljes rendszer működésképtelenné válna, mivel kivitelezhetetlen lenne a felhasználók adatainak elraktározása az adatbázisban.



Az alábbi elektronikus felületmomentum-extrakció szubsztanciájának részeként konstituálja a destroy metódust. Eme procedúra funkcionális determinációja az adott elem törlése az adatbázisból. A jelenlegi projekt esetében ezt alkalmazni kizárólag a felhasználóknál volt értelme, máshol inszignifikáns incidenciával fordul elő, hogy az adatok megsemmisítése operatívan indokolt legyen.

Ez a virtuális képkivágat inkludálja az update metódust. Ez a műveleti egység az adatállomány frissítéséért és szerkesztéséért felelős, ha a felhasználó vagy adminisztrátor egy már meglévő, de potencionálisan revízióra szoruló, vagy csak szimplán megváltoztatandó adatot kíván módosítani.

A fent látható digitális pillanatfelvételen láthatók a moreFullThan és az emptierThan metódusok. Ezek alapvető célja, hogy a kapott paramétert valamivel hasonlítsák, ez minden adattáblában más lehet, jelen esetben a REpont gépek töltöttségi szintjét vizsgálják, és ami abba a tartományba tartozik azt visszaadják további felhasználásra.

1. **Tesztelés**
2. **Továbbfejlesztési lehetőségek**
3. **Irodalomjegyzék**