**Backend fejlesztői dokumentáció**

Visual Studio Code: Laravel apik, kontrollerek megírása és tesztelése

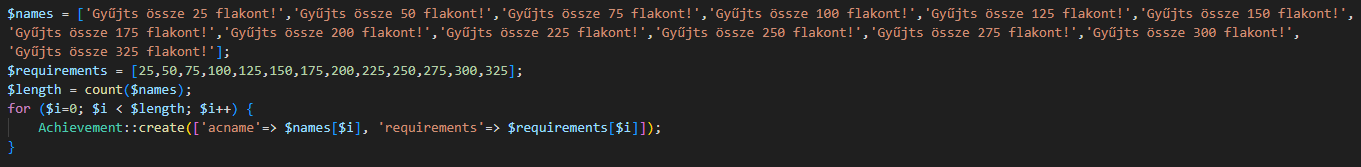
XAMPP: Adatbázis lokális futtatása és kezelése

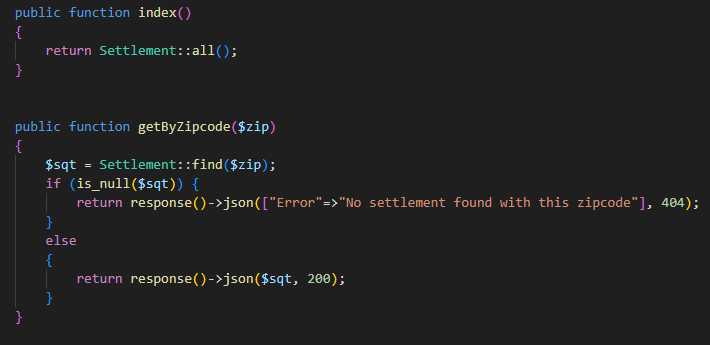
**Backend dokumentáció**

A screen shot of a computer program

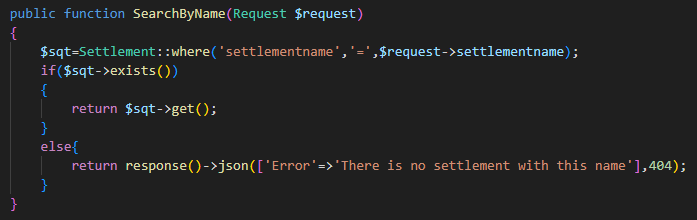
Description automatically generatedA backend feladata az adatbázis és a frontend közötti zökkenőmentes kapcsolat létesítése és fenntartása. A lehető legtöbb feladat ellátására lettek apik létrehozva, hogy a későbbiekben már ne kelljen nagymértékű változtatásokat eszközölni a kódban. A cél az volt, hogy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

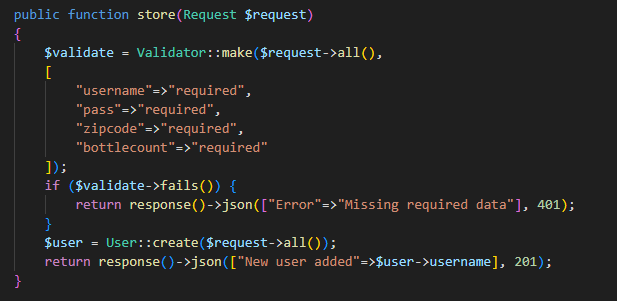
Ezen a képen fellelhető az api.php, amiben benne vannak a megírt apikhoz vezető elérési útvonalak. A frontend kódja erre hivatkozik és a felhasználó vagy adminisztrátor által elrendelt műveletet végrehajtja.

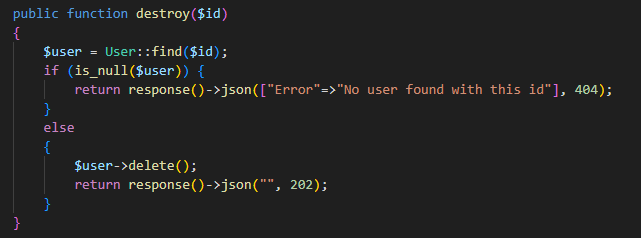


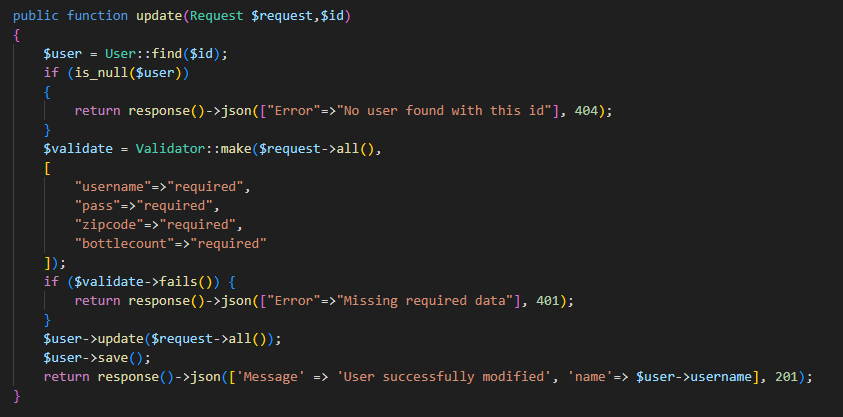
Itt látható a databaseseeder, ennek a feladata alapvetően a felmigrált adattáblák feltöltése. Ez esetben az achievementek adattáblát tölti fel előre meghatározott adatokkal. A többi adattáblát külső sql fájból importáljuk, az adatok nagy mennyisége miatt.

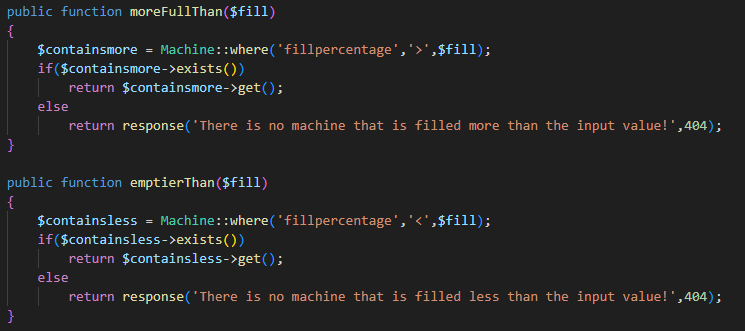
Ebben a képernyőfotóban leledzik az index és a getById metódus (itt getByZipcode).E jelenlegi specifikus szituációban a településekhez tartozó Settlement Controller kódrészlete tárul szemünk elé. Ennek segítségével hívható le az összes adat a szóban forgó adattáblából, vagy ha csak egy adott elemre van 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.