**Adatbázis automatikus frissítése:**

A megoldásához egy Linux szerveren is futó automatikus parancsindító programot kellett keresnünk. Hosszas kutatás és utánajárás következményeként A Cron nevű feladatidőzítőt találtuk a legoptimálisabbnak. Szerencsénkre a Synology NAS adattároló (amin az adatbázis és a webszerver is fut) önmagába foglal egy beépített feladatidőzítőt, aminek grafikus felülete miatt egyszerűen lehet kezelni és konfigurálni. Ez a beépített Task Scheduler egy Cron alapú szoftver, szóval tökéletesnek bizonyult számunkra.

Mindezek után utánajártunk a Laravel beépített függvényeinek, hogy találunk-e egy célnak megfelelőt, hogy ne kelljen a 0-ról megírni. Rá is bukkantunk a Laravel beépített Task Scheduling funkciójára. Ehhez mindössze az app/Console/Kernel.php fájlban kellett egy függvényt létrehoznunk:

$schedule->call(function () {

// Az adatbázis frissítő kód

})->everyMinute();

A függvény feladata csupán, hogy a megadott értékek és metódusok alapján minden percben futtassa le a kódot, ami egy update parancs az adatbázisnak.

A függvényt hosszasan teszteltük és ellenőriztük és beépítettük a Laravel projektbe.

Ezt a: php artisan schedule:run paranccsal tudjuk megtenni.

Ezt a parancsot kell a Synology NAS beépített feladatidőzítőjében beállítani percenkénti futtatásra.

A kód: /volume4/@appstore/PHP8.0/usr/local/bin/php80

/volume4/web/HouseOfSwords/artisan schedule:run

Így a parancs az általunk is használt php 8.0-ás verzióval fog futni és a későbbiekben elkerülve a kompatibilitási problémákat, ha esetleg automatikusan frissülne a php verzió.

Ezután egy problémába ütköztünk. Egy „access denied for user 'root'@'localhost'” hibakódú oldalt kaptunk válaszul. Ezt hosszas tesztelgetés után meg tudtuk oldani. A változtatások miatt a Linux szerveren az alapértelmezett http-felhasználónak újra kellett osztani az egész mappára és almappáira az írás és olvasás jogot, ezek után hiba nélkül futott a weboldal is és az adatbázis frissítő is sikeresen implementálva lett.