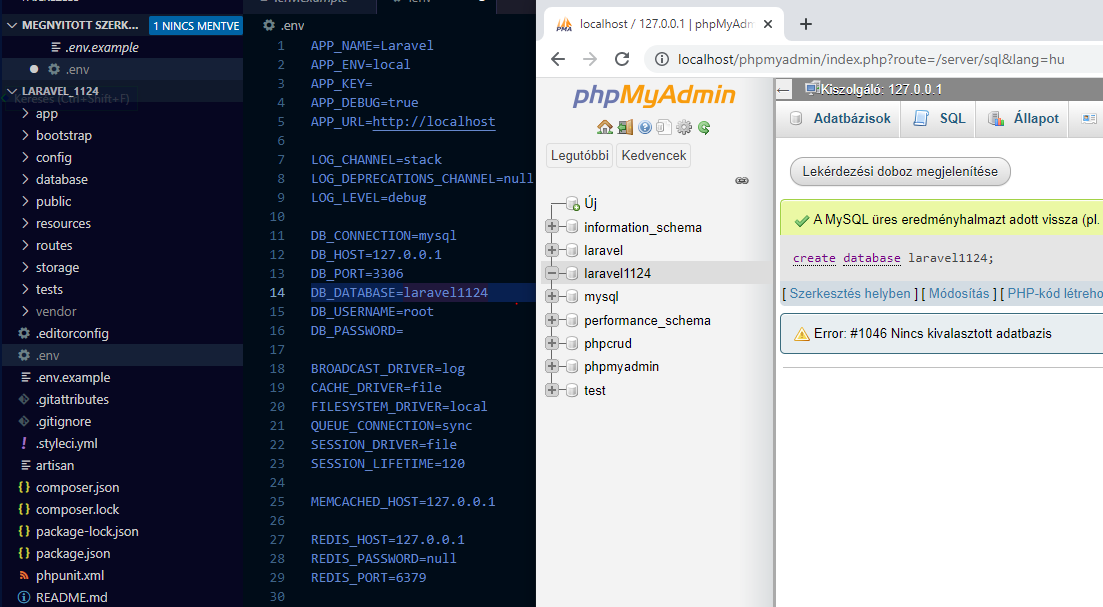
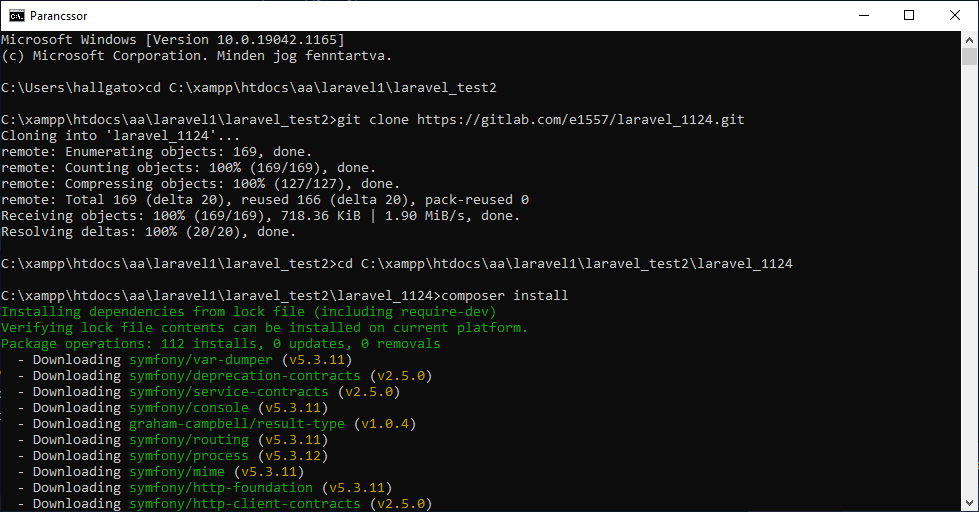
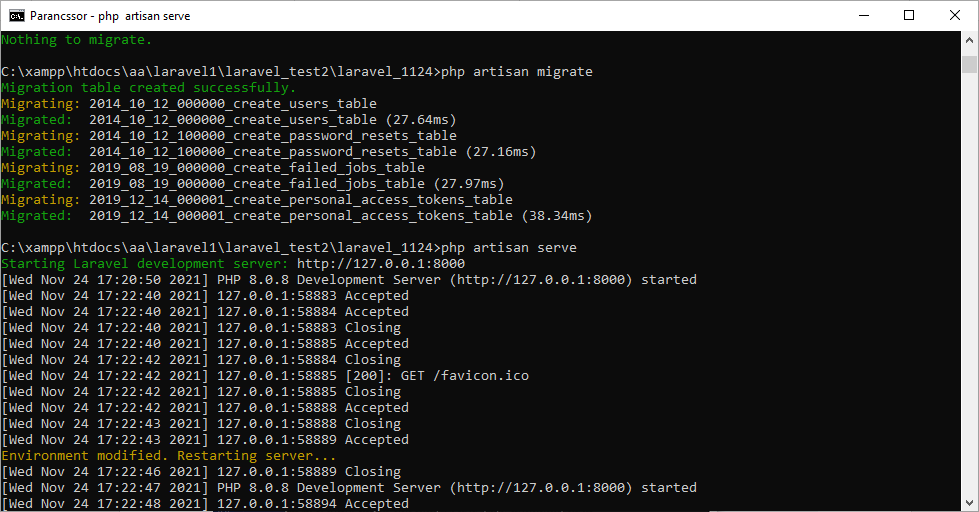


Futtatom az Apache-t és a MySQL-t a XAMPP-on keresztül. Létrehozok egy adatbázist phpMyAdmin-ban, ügyelve arra, hogy annak neve megegyezzen az .env file-ban megadott adatbázisnévvel:

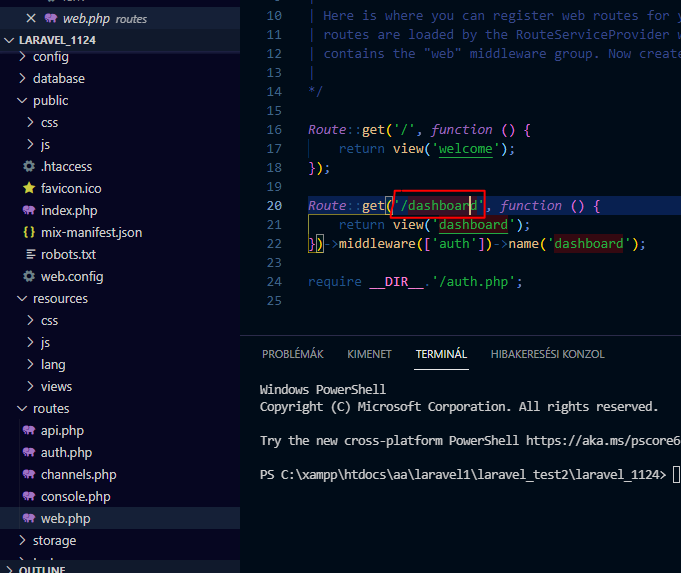


CMD 🡪 parancsok:

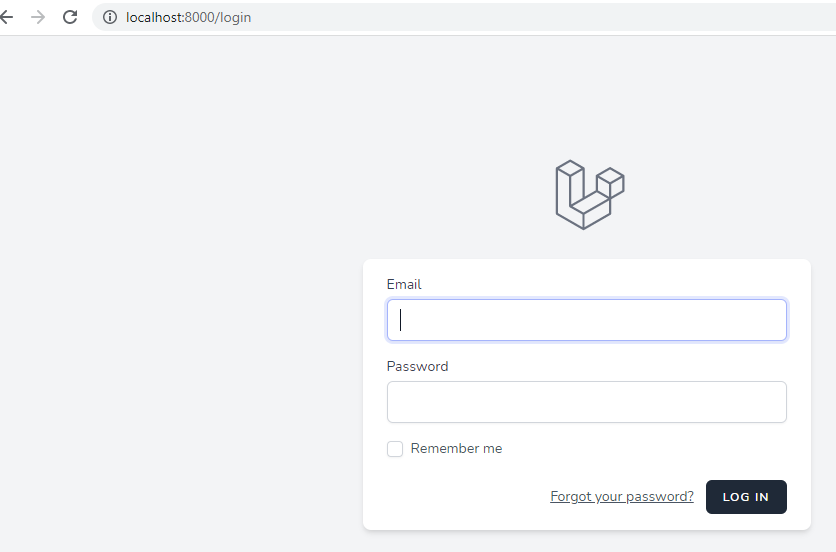




Megkeresem a tesztelendő útvonalat: /dashboard:







Regisztrálni kellett (ha jól emlékszem, ezen a linken: <http://localhost:8000/dashboard> )

**Feladat**: *hozzunk létre* ***taskokat****, amelyeket személyekhez rendelhetünk, szerkeszthetünk, törölhetünk, stb.*

Task:

* task\_id
* title
* description
* end\_date
* due\_date
* user\_id
* status (0,1/ closed/open)
* timestamp
* created\_by (🡪 ő is user!)

Task: JSON VIEW

GET – all /api/tasks /tasks

GET – 1 task /api/tasks/1 /task/edit/1

POST – /api/tasks /task/create (a kódban store) *form*

PUT - /api/tasks/1 *form*

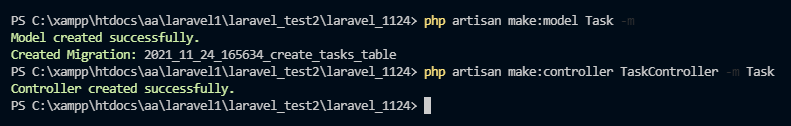
DELETE - /api/tasks/1 *form*

User:

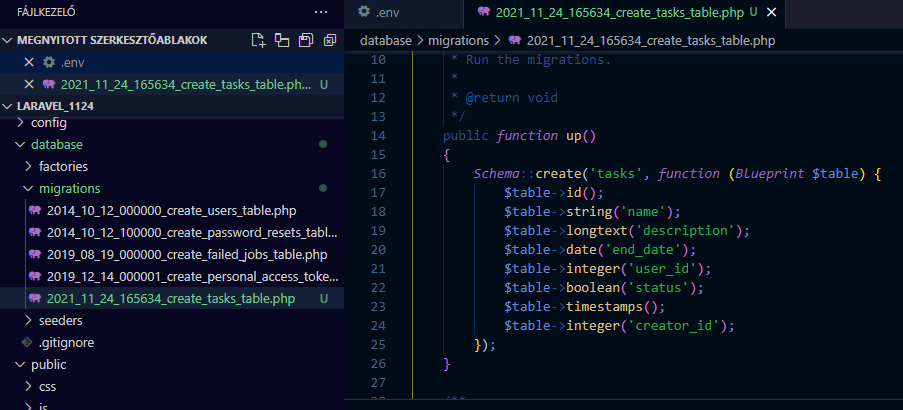
GET – all /api/users

Először létrehozzuk a *Task* modellt: php artisan make:model Task –m 🡪 a migrációt is léterhozza

Majd létrehozzuk a Task controllert: php artisan make:controller TaskController –m Task

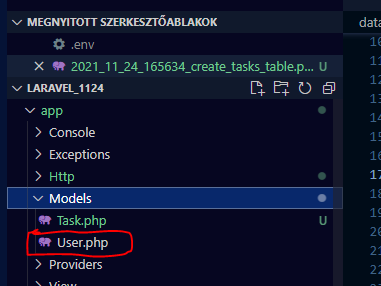


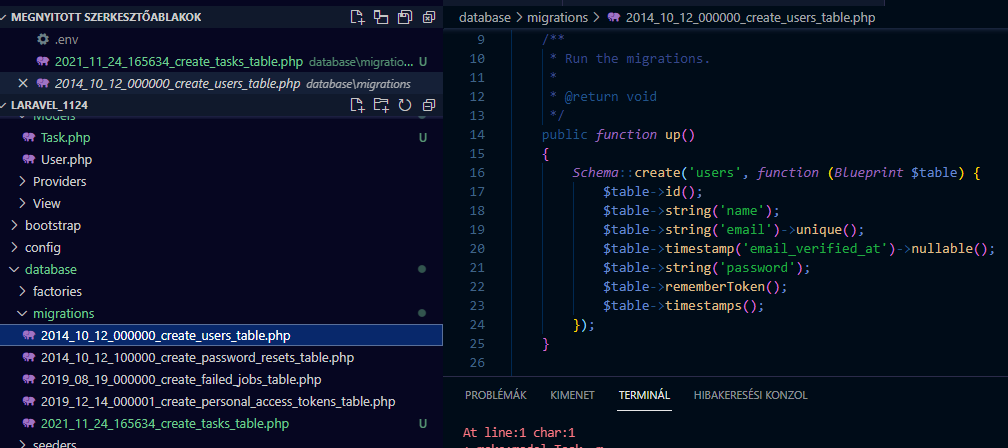
Létrehozzuk a tábla attribútumokat:



(Megjegyzés utólag: a boolean típusú változó – migráció után – tinyintként fog átszinkronizálódni MyPHPAdminba, amely odafigyelést igényel! )

Az User modell már létezik, azt nem kell újra létrehozni:





A Task tábla *user\_id* és *creator\_id* attribútumai külső kulcsok lesznek, és a User tábla *id* mezőjével lesznek összekapcsolva (párhuzamos kapcsolat).

Be kell állítani, hogy a Task táblában a user\_ID és a creator\_id külső kulcs legyen:

$table->foreignId('user\_id')->constrained('users');

Tehát a User táblából az id attribútumot nézi. Foreign key constraints:

<https://laravel.com/docs/7.x/migrations#foreign-key-constraints>

Idevágó példa:

schema::table('posts', function (Blueprint $table) {

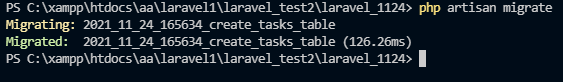
$table->unsignedBigInteger('user\_id');

$table->foreign('user\_id')->references('id')->on('users');

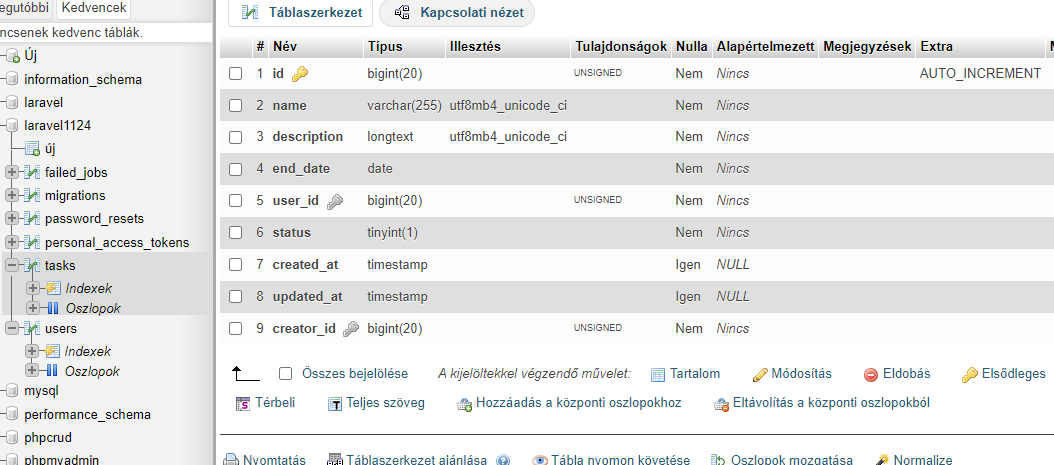
});

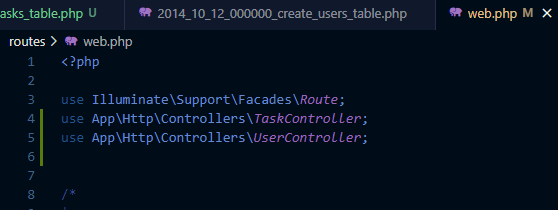


Újra migrálni kell:

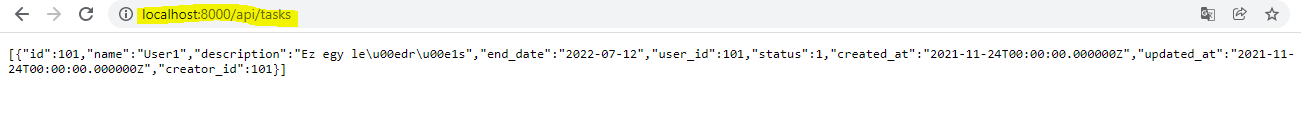


MySQL-ben is létrejöttek az attribútumok:









*Route*::get('/api/tasks',[*TaskController*::class,'index']);

*Route*::get('/api/task/{id}',[*TaskController*::class,show]);

*Route*::post('/api/task',[*TaskController*::class,store]);

*Route*::put('/api/task/{id}',[*TaskController*::class,update]);

*Route*::delete('/api/task/{id}',[*TaskController*::class,destroy]);

//views

*Route*::get('/task/new',[*TaskController*::class,newView]);

*Route*::get('/task/edit/{id}',[*TaskController*::class, editView]);

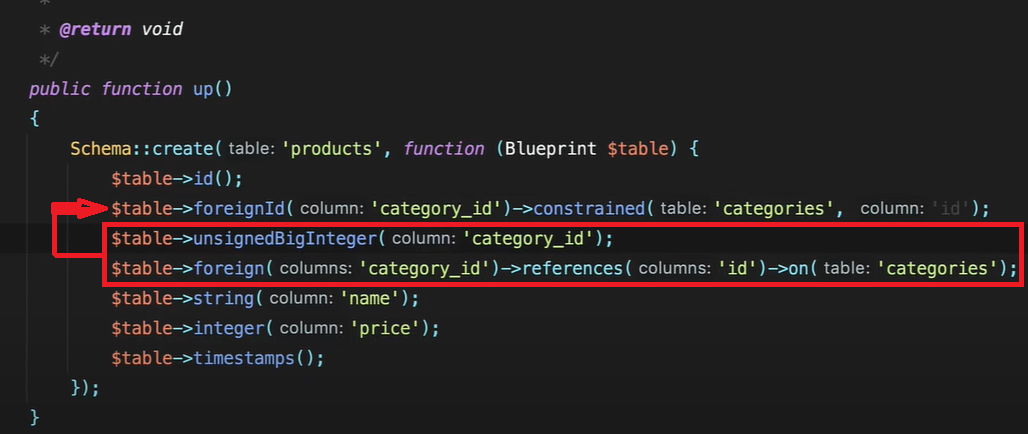
*Route*::get('/task/list',[*TaskController*::class, listView]);

**CSRF tokens**

A CSRF token is a unique, secret, unpredictable value that is generated by the server-side application and transmitted to the client in such a way that it is included in a subsequent HTTP request made by the client. When the later request is made, the server-side application validates that the request includes the expected token and rejects the request if the token is missing or invalid.

Mégegy példa a foreign key constraint-re:

<https://www.youtube.com/watch?v=QL0gKwGjJ-4>



A nyíllal jelzett sor a pirossal bekeretezett rész rövidebb verziója. Azaz, az aktuális, *products* tábla *category\_id* attribútuma a *categories* tábla *id* attribútumára/oszlopára mutató külső kulcs.