## Exercici Crear migració a la taula tasks . Ha de tenir els camps (id, name, description i completed)

Crear factory
Crear seed
Crear inserts a partir del seed
Crear controlador de tasks
Crear model de tasks
Crear ruta a routes/web.php i afegir la nova ruta al navegador al fitxer resources/views/components/layout.blade.php
Crear la vista tasks.blade.php

Amb la comanda php artisan make creare el següent que serà necessari per completar la activitat.

```
alumne@harvey:~/laravelApp$ php artisan make:migration create_tasks_table

INFO Migration [database/migrations/2024_01_08_160709_create_tasks_table.php] created successfully.

alumne@harvey:~/laravelApp$ php artisan make:model Task

INFO Model [app/Models/Task.php] created successfully.

alumne@harvey:~/laravelApp$ php artisan make:controller TaskController

INFO Controller [app/Http/Controllers/TaskController.php] created successfully.

alumne@harvey:~/laravelApp$ php artisan make:factory TaskFactory

INFO Factory [database/factories/TaskFactory.php] created successfully.

alumne@harvey:~/laravelApp$ php artisan make:view tasks

INFO View [resources/views/tasks.blade.php] created successfully.
```

Després de executar les comandes prèvies, m'ha creat el seguent [.php] que editare per crear la taula tasks amb els seus camps id, name (string), description (string), completed (boolean) i timestamps (per defecte).

També editare el fitxer de [Task.php] per afegir els camps a protected \$visible i els altres camps a protected \$fillable (camps no sensibles) visibles i els que es autocompletaran per el sistema automaticament.

Dins de [TaskController.php] creare una funció anomenada index, que em retorna una vista de totes les tasques.

A continuació dintre del fitxer [TaskFactory.php] definire un return dels camps que vull completar amb informació. Cada camp amb el seu tipus.

```
Inamespace Database\Factories;

anamespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

use App\Models\Task;

/**

* @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Task>

*/

nousages

class TaskFactory extends Factory

{

/**

* Define the model's default state.

* * @return array<string, mixed>

* /*

nousages

public function definition(): array

{

return [

'name' => \this->faker->name(),

'description' => \this->faker->boolean(),

];

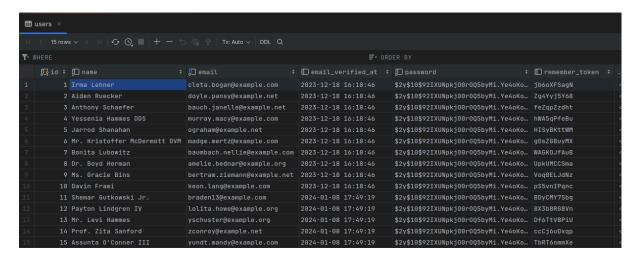
}

if this indicates the sector of the sector o
```

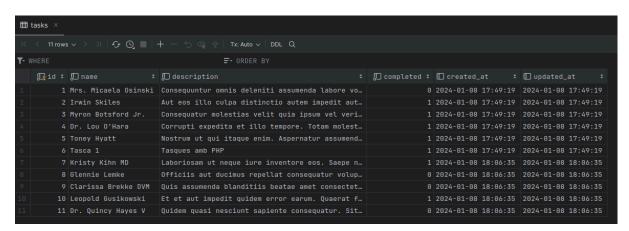
Per crear registres automàtics indicare el factory amb el número de registres que vull crear, en aquest cas 5 i modifico el de usuaris per a que sigue 5 també.

Amb la comanda [php artisan migrate –seed] afegire els registres de forma automàtica i creara la taula de tasks.

## TAULA USUARIS:



## TAULA TASKS:



Copiare el contingut de users\_blade.php i ho pegare a tasks\_blade.php. Ademés afegint un bucle de la taula tasks que mostrara totes les tasques de la taula de la base de dades que s'han creat un detras de l'altre.

```
div class="py-10">

cdiv class="py-10">

cdiv class="py-10">

cdiv class="py-10">

cdiv class="py-10">

cdiv class="text-ski font-bold tracking-light text-white">USUARIS</hi>
chi class="text-ski font-bold text-ski font-bold text-ski font-ski fort-ski font-ski fort-ski fort-ski font-ski fort-ski font-ski font-ski
```

Al fitxer que ja tinc creat de [web.php] afegire una línia al final que fa referència a l'índex de tasks indicant la ruta.

```
Route::get( uri: '/',[UserController::class,'index']);
Route::get( uri: '/users',[UserController::class,'index']);
Route::get( uri: '/contact',[PagesController::class,'contact']);
Route::get( uri: '/about',[PagesController::class,'about']);
Route::get( uri: '/tasks',[\App\Http\Controllers\TaskController::class,'index']);
```

Finalment al layout afegeixo una línia al nav per a mostrar una nova opció a la barra de navegació.

```
| Solution | Solution
```

Obrire la pàgina i com a resultat final veurem que em mostra les tasques com hauria de fer.

