

Exercici

Creem una migració de taula i creem la taula tasks amb els camps que li pertocuen:

```
→ laravelApp git:(main) × php artisan make:migration create_tasks_table

[INFO] Migration [database/migrations/2024_01_08_175635_create_tasks_table.php] created successfully.

→ laravelApp git:(main) ×
```

```
1  <?php
2
3  > use ...
4
5
6
7  > return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create(table: 'tasks', function (Blueprint $table) {
15             $table->id();
16             $table->string(column: 'name');
17             $table->string(column: 'description');
18             $table->string(column: 'completed')->default(value: false);
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      */
26     public function down(): void
27     {
28         Schema::dropIfExists(table: 'tasks');
29     }
30 };
31
```

Un cop creat i emplenat el fitxer de migrate, executem al terminal la comanda **php artisan migrate**

```
→ laravelApp git:(main) × php artisan migrate

[INFO] Running migrations.

2024_01_08_175635_create_tasks_table ..... 7ms DONE

→ laravelApp git:(main) ×
```

Creem un nou model anomenat Task amb la comanda ***php artisan make:model Task***

```
→ laravelApp git:(main) ×  
→ laravelApp git:(main) × php artisan make:model Task  
  
INFO Model [app/Models/Task.php] created successfully.  
→ laravelApp git:(main) ×
```

Dins al model hauriem de veure-ho així i afegir els camps de la taula baix a protected

```
1 <?php  
2  
3 namespace App\Models;  
4  
5 use Illuminate\Database\Eloquent\Factories\HasFactory;  
6 use Illuminate\Database\Eloquent\Model;  
7  
8 no usages  
9 class Task extends Model  
10 {  
11     use HasFactory;  
12  
13     no usages  
14     protected $visible = [  
15         'id'  
16     ];  
17  
18     no usages  
19     protected $fillable = [  
20         'name',  
21         'description',  
22         'completed',  
23     ];  
24 }  
25
```

Ara crearem el controls amb ***php artisan make:controller TaskController***

```
→ laravelApp git:(main) × php artisan make:controller TaskController  
  
Problems Alt+6 Controller [app/Http/Controllers/TaskController.php] created successfully.  
→ laravelApp git:(main) ×
```

Dins al controller, hem de posar la vista que volem que retorni i completar les rutes que falten per que no done error:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Task;
6  use Illuminate\Http\Request;
7
8  class TaskController extends Controller
9  {
10     public function index()
11     {
12         return view('tasks', [
13             'tasks' => Task::all()
14         ]);
15     }
16 }
17
```

I creem també la factory amb la comanda ***php artisan make:factory TaskFactory***

```
→ laravelApp git:(main) ×
→ laravelApp git:(main) × php artisan make:factory TaskFactory

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

→ laravelApp git:(main) ×
```

Dins la factory, també hem de posar els camps de la taula per vincular-los amb el model

```
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6
7  /**
8   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Task>
9   */
10 no usages
11 class TaskFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array<string, mixed>
17      */
18 no usages
19 public function definition(): array
20 {
21     return [
22         'name' => $this->faker->name(),
23         'description' => $this->faker->paragraph(),
24         'completed' => $this->faker->boolean(),
25     ];
26 }
```

I afegirem el nombre de inserts que volem que insereixi a la taula:

```
2     */
3     pmonfort13 *
4     public function run(): void
5     {
6         \App\Models\User::factory(count: 5)->create();
7
8         \App\Models\Task::factory(count: 7)->create();
9     }
```

Ara ja podem executar la comanda ***php artisan migrate --seed*** per crear els inserts de forma automàtica a la taula:

```
→ laravelApp git:(main) × php artisan migrate --seed

INFO Nothing to migrate.

INFO Seeding database.

→ laravelApp git:(main) ×
```

Podem veure com la taula queda plena amb els inserts:

	id	name	description	completed	created_at	updated_at
1	1	Prof. Maximus Ratke Jr.	Optio aperiam odio consequatur do...	1	2024-01-08 18:19:20	2024-01-08 18:19:20
2	2	Priscilla Hahn	Numquam incidunt eveniet animi ea...	1	2024-01-08 18:19:20	2024-01-08 18:19:20
3	3	Dr. Ray Cummings	Omnis ut nobis consequatur repudi...	0	2024-01-08 18:19:20	2024-01-08 18:19:20
4	4	Skyla Koss	Quas enim quod dolores laudantium...	1	2024-01-08 18:19:20	2024-01-08 18:19:20
5	5	Nadia Green	Voluptas minima dolor facilis ali...	1	2024-01-08 18:19:20	2024-01-08 18:19:20
6	6	Alexa Dach DVM	Dolore et repellat deserunt et at...	1	2024-01-08 18:19:20	2024-01-08 18:19:20
7	7	Raphael Gutkowski I	Molestiae libero et voluptatum an...	1	2024-01-08 18:19:20	2024-01-08 18:19:20

Creem la vista Tasks que serà el que veurem al navegador:

```
→ laravelApp git:(main) × php artisan make:view tasks

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

→ laravelApp git:(main) ×
```

Ara hem de crear la vista Tasks i afegir els camps que volem que es vegin a la taula a la web:

```
27 </thead>
28 <tbody class="bg-white">
29 @foreach($tasks as $task)
30 <tr class="even:bg-gray-50">
31 <td class="whitespace-nowrap py-4 pl-4 pr-3 text-sm font-medium text-gray-900 sm:pl-3">{{ $task->id }}</td>
32 <td class="whitespace-nowrap px-3 py-4 text-sm text-gray-500">{{ $task->name }}</td>
33 <td class="whitespace-nowrap px-3 py-4 text-sm text-gray-500">{{ $task->description }}</td>
34 <td class="whitespace-nowrap px-3 py-4 text-sm text-gray-500">{{ $task->completed }}</td>
35 </tr>
36 @endforeach
37 <!-- More people... -->
38 </tbody>
39 </table>
40
41
42
```

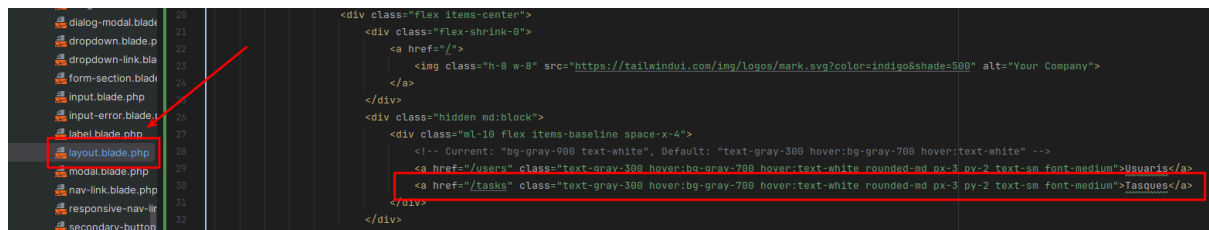
Creem la ruta a la pàgina Tasks al fitxer de **routes**

```

30 //});
31 Route::get( uri: '/', [UserController::class, 'index']);
32 Route::get( uri: '/users', [UserController::class, 'index']);
33 Route::get( uri: '/tasks', [UserController::class, 'index']);
34 Route::get( uri: '/contact', [PagesController::class, 'contact']);
35 Route::get( uri: '/about', [PagesController::class, 'about']);
36

```

I per últim afegirem al menú de navegació de **layout.blade.php** el botó per veure la vista de Tasks

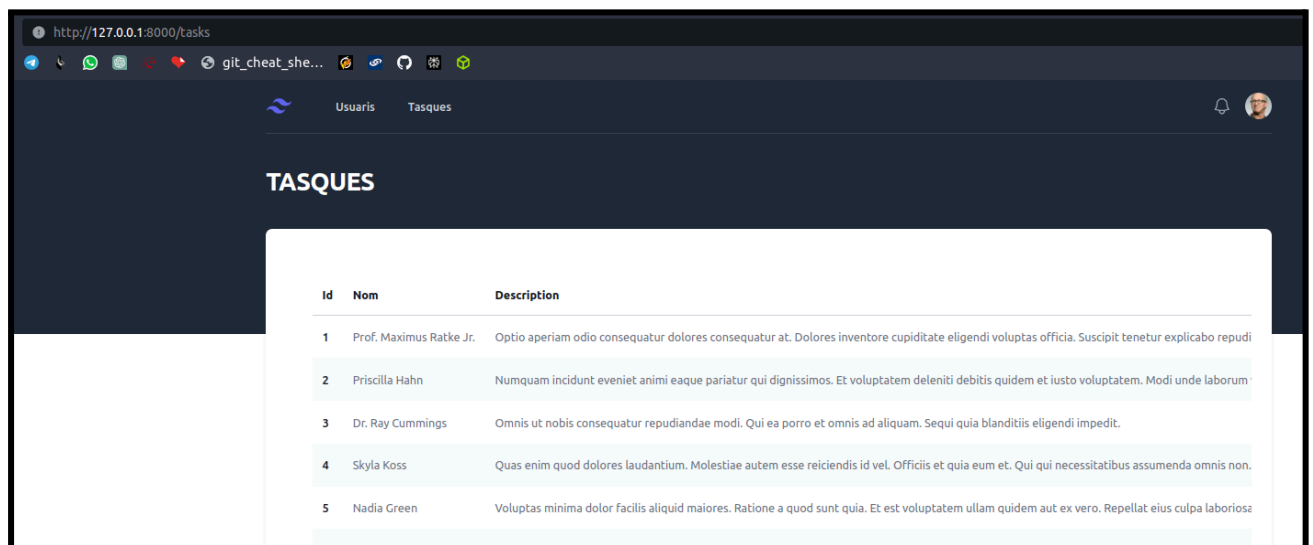


```

20 <div class="flex items-center">
21   <div class="flex-shrink-0">
22     <a href="/">
23       
24     </a>
25   </div>
26   <div class="hidden md:block">
27     <div class="ml-10 flex items-baseline space-x-4">
28       <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->
29       <a href="/users" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">Users</a>
30       <a href="/tasks" class="text-gray-300 hover:bg-gray-700 hover:text-white rounded-md px-3 py-2 text-sm font-medium">Tasks</a>
31     </div>
32   </div>

```

Ara ja podem entrar a la Web i si anem a l'apartat de tasques ens surt la taula de tasques que hem creat:



Id	Nom	Description
1	Prof. Maximus Ratke Jr.	Optio aperiam odio consequatur dolores consequatur at. Dolores inventore cupiditate eligendi voluptas officia. Suscipit tenetur explicabo repudi
2	Priscilla Hahn	Numquam incidunt eveniet animi eaque pariatur qui dignissimos. Et voluptatem deleniti debitis quidem et iusto voluptatem. Modi unde laborum
3	Dr. Ray Cummings	Omnis ut nobis consequatur repudiandae modi. Qui ea porro et omnis ad aliquam. Sequi quia blanditiis eligendi impedit.
4	Skyla Koss	Quas enim quod dolores laudantium. Molestiae autem esse reiciendis id vel. Officiis et quia eum et. Qui qui necessitatibus assumenda omnis non.
5	Nadia Green	Voluptas minima dolor facilis aliquid maiores. Ratione a quod sunt quia. Et est voluptatem ullam quidem aut ex vero. Repellat eius culpa laborios

e voila!