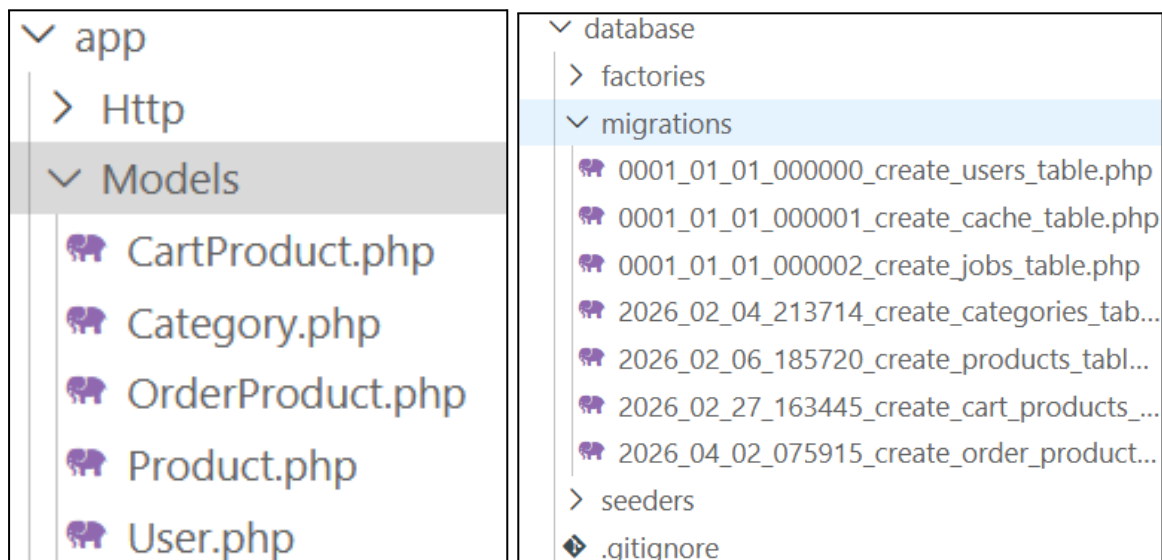


↳ Gestion des modèles, des migrations, et de leurs clés étrangères (Avec *Eloquent ORM*):

• Les modèles sont des objets chargés de définir les données et leur tables respectives de base de données au sein de notre application/site. Les migrations sont l'ensemble des méthodes par lesquelles *Laravel* instancie les changements et les ajouts dans la base de données. Les clés étrangères sont des colonnes spécifiques qui, en accord avec la théorie des cardinalités, sont implantées à partir d'une table différente dans une autre, elles sont définies à la fois dans les classes modèles et dans les scripts de migrations.

• Il est à noter qu'à aucun moment le langage SQL ne sera utilisé dans un environnement *Laravel* par défaut (par ailleurs le script SQL précédemment cité n'est réalisé que dans un objectif de test). L'ORM *Eloquent* spécifique à *Laravel*, et sa syntaxe est la manière prévue d'interagir avec les données.



**Nous constatons que les dossiers modèle et migrations, tous deux portant sur la gestion de la base de données, ne sont pas dans le même répertoire, le premier étant groupé avec les contrôleurs et le deuxième dans un répertoire 'database'.*

• Dans les faits, les fichiers modèles de *Fullfort* ne contiennent que très peu de contenu, hormis ceux se rapportant aux tables contenant des relations étrangères (Un à Un/Un à Plusieurs/Plusieurs à Plusieurs); elles contiennent alors les clés étrangères que contiendra la table, et les clés primaires qu'elles référencent.

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class OrderProduct extends Model
9  {
10     use HasFactory;
11
12     public function user()
13     {
14         return $this->hasOne('App\Models\User', 'id', 'user_id');
15     }
16
17     public function product()
18     {
19         return $this->hasOne('App\Models\Product', 'id', 'product_id');
20     }
21 }
22

```

**Modèle de la table des produits commandés, cette tables inclut donc en tant que clés étrangères les clés primaires 'id' de la table 'user' et 'id' de la table 'product'*

- Les migrations contiennent les requêtes de type “action” servant à créer et/ou modifier des données, voire des tables. La toute première migration sera celle qui créera la base de données ‘fullfort_db’. Chaque migration successive (qui peut d’ailleurs être `rollback` (annulées et restaurées) si le besoin survient) se fait à l’aide de la commande: `php artisan migrate`. C’est finalement dans ces fichiers migration que se font toutes les requêtes d’insertion de données, alors que les requêtes de lecture se font dans les vues.

```

7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('products', function (Blueprint $table) {
15             $table->id();
16             $table->string('name')->nullable();
17             $table->longText('description')->nullable();
18             $table->integer('quantity')->nullable();
19             $table->integer('price')->nullable();
20             $table->string('image')->nullable();
21             $table->string('category')->nullable();
22             $table->timestamps();
23         });
24     }

```

**Fichier de migration de la table 'users', contient toutes les colonnes de la table à insérer tout comme le ferait un script SQL.*

- Une fois déclarés, ces modèles sont systématiquement importés dans le contrôleur ayant besoin des données des tables qu'ils référencent pour qu'ils puissent opérer.

```

5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Auth;
7  use App\Models\User;
8  use App\Models\Product;
9  use App\Models\CartProduct;
10 use App\Models\OrderProduct;

```

**Importation de toutes les classes modèles dans UserController, sans modèles, aucune opération sur les données ne pourrait être effectuée.*