

Cadriciel Web

TP1

Gr 21647

Par

Nicolas Fontaine

22 janvier 2023

1. En utilisant les lignes de commande, créer un nouveau projet Laravel nommée Maisonneuve{votre matricule} (1 pt)

```
nikon@DESKTOP-LM3U9GN MINGW64 ~/OneDrive/Desktop/session-4/cadri/projetsCadriciel
$ composer create-project --prefer-dist laravel/laravel Maisonneuve2295422 "8.*"
Creating a "laravel/laravel" project at "./Maisonneuve2295422"
```

2. En utilisant les lignes de commande, créer les modèles (2 pts)

```
ve2295422
$ php artisan make:model Etudiant
Model created successfully.
```

```
ve2295422
$ php artisan make:model Ville
Model created successfully.
```

3. En utilisant les lignes de commande, créer les tables (2 pts)

```
ve2295422
$ php artisan make:migration create_etudiants_table
Created Migration: 2023_01_23_000356_create_etudiants_table
```

```
ve2295422
$ php artisan make:migration create_villes_table
Created Migration: 2023_01_23_000443_create_villes_table
```

Après avoir remplis les table dans le dossier migrations :

```
ve2295422
$ php artisan migrate
Migration table created su
```

4. En utilisant les lignes de commande, saisir 15 nouvelles villes (1 pts)

```
ve2295422
$ php artisan make:factory VilleFactory -m Ville
Factory created successfully.
```

```
ve2295422
$ php artisan tinker
Psy Shell v0.11.10 (PHP 8.1
```

```
> App\Models\Ville::factory()->times(15)->create();
= Illuminate\Database\Eloquent\Collection {#3596
  all: [
```

5. En utilisant les lignes de commande, saisir 100 nouveaux étudiant (1 pts)

```
ve2295422
$ php artisan make:factory EtudiantFactory -m Etudiant
Factory created successfully.
```

```
ve2295422
$ php artisan tinker
Psy Shell v0.11.10 (PHP 8.1
```

```
> App\Models\Etudiant::factory()->times(100)->create();[]
```

Pour les questions 5 et 6, effectuez une recherche des propriétés de "Factory" pour remplir des valeurs telles que des noms, des adresses, des téléphones, etc. (pas de phrases ou de texte aléatoires).

6. En utilisant les lignes de commande, créer les contrôleurs (2 pts)

```
ve2295422
$ php artisan make:controller EtudiantController -m Etudiant
Controller created successfully.
```

```
ve2295422
$ php artisan make:controller VilleController -m Ville
Controller created successfully.
```

7. Créez votre layout.blade avec vous CSS, vous devez importer bootstrap et le concevoir selon vos préférences. (2 pts)

Fait

8. Travailler avec bootstrap pour respecter les concepts d'ergonomie, soyez créatif (2pts).

Fait

9. Créer un contrôleur "index" et une vue, pour afficher tous les étudiants, avec un lien pour sélectionner l'étudiant et le mettre à jour. (2 pts)

Fait

10. Créer un contrôleur "create" et une vue, pour saisir un nouvel étudiant. Le formulaire doit avoir un champ "select" avec toutes les villes qui viennent de la base de données. (2 pts)

Fait

11. Créer un contrôleur "show" et une vue, pour afficher un étudiant sélectionné. (2 pts)

Fait

12. Créer un contrôleur "edit" et une vue, pour afficher un étudiant sélectionné dans un formulaire et le mettre à jour. (2 pts)

Fait

13. Créer un contrôleur "destroy" pour supprimer un étudiant sélectionné. (1 pt)

Fait

14. Publier votre projet dans GitHub (publique) et envoyer le lien dans la documentation. (2 pts)

Lien GitHub : <https://github.com/Naljorpa/cadriciel-tp1>

15. Enregistrez le projet avec une extension ZIP et ajouter la documentation dans la racine (1pt)

Fait

Lien webdev : <https://e2295422.webdev.cmaisonneuve.qc.ca/tp1-cadri/>