

Cadriciel Web - 582-41B-MA

TP 2 : Laravel - Login

par: François Hébert (e2296133)

DOCUMENTATION

RÉFÉRENCES:

- LIEN VERS LE GITHUB: [GIT DU PROJET](#)
- LIEN VERS LE PROJET (WEBDEV):
 - A NOTER QUE LA VERSION NE FONCTIONNE PAS CORRECTEMENT due a php7, vous m'avez corrigé le travail directement.
 - <https://e2296133.webdev.cmaisonneuve.qc.ca/cadricieltp2v2COPY/>

USER:

- User: francois@francois.com
- Password: francois

REQUIS DÉMANDÉS

Félicitations, votre client a apprécié votre premier projet et aimerait vous offrir un deuxième mandat pour améliorer le réseau social du collège. Vous souhaitez impliquer toutes les nouvelles connaissances acquises dans le cours Cadriciel Web pour la poursuite de ce projet.

Le client souhaite maintenant

- ☒ ajouter une page de connexion pour que chaque élève puisse se connecter à son propre compte,
- ☒ la table étudiante doit donc être connectée à la table utilisateur (users).
- ☒ Pour maintenir la sécurité du système, le mot de passe doit être crypté (2 pts)

Étant donné que les étudiants du collégial sont polyglottes, le client vous a demandé de créer un système multilingue, français en anglais.

```
php artisan lang:publish
```

- ☒ Tout le contenu du système doit être dans les deux langues (2 points)

Pour compléter le système et le mettre en production, le client a demandé 2 autres modifications majeures.

Le Forum

Le système doit avoir

- ☒ un forum dans lequel les étudiants peuvent écrire des articles.
- ☒ Les articles doivent être visibles par tous les étudiants connectés.
- ☒ Seul l'étudiant qui a écrit l'article peut le modifier et/ou le supprimer. (2 pts)

Pour créer le forum

- ☒ il faut ajouter un tableau dans la base de données, il est important d'enregistrer

```
php artisan make:model Ville
php artisan make:model Etudiant
php artisan make:model BlogPost
INFO Model [app/Models/BlogPost.php] created successfully.
(edit model file)

php artisan make:migration create_villes_table
(updater le schema dans le php)
php artisan make:migration create_etudiants_table
(updater le schema dans le php)
php artisan make:migration create_blogPosts_table
INFO Migration
[database/migrations/2023_09_28_205026_create_blog_posts_table.php]
created successfully.
(edit migration file)
php artisan make:migration create_documents_table
(updater le schema dans le php)

php artisan make:factory VilleFactory
(updater le schema dans le VilleFactory.php)
php artisan make:factory EtudiantFactory
(updater le schema dans le EtudiantFactory.php)
hp artisan make:factory DocumentFactory
(updater le schema dans le EtudiantFactory.php)
php artisan make:factory BlogPostFactory
INFO Factory [database/factories/BlogPostsFactory.php] created
successfully.
(updater le schema dans le BlogPost Factory.php)

php artisan migrate
(Table created)

php artisan tinker

> \App\Models\Ville::factory()->times(15)->create();
> \App\Models\Etudiant::factory()->times(100)->create();
> \App\Models\BlogPost::factory()->times(15)->create();
```

- ☒ le titre de l'article, le contenu et la date,
- ☒ les articles peuvent être rédigés en français et en anglais,

- ☒ et le système doit gérer la langue de choix lors de la publication. (2 pts)

Le répertoire de documents

- ☒ Pour compléter, le système doit disposer d'un répertoire de fichiers, dans lequel les étudiants peuvent partager des documents au format pdf, zip et doc.

```
php artisan make:controller DocumentController --m Document
```

- ☒ Ce répertoire doit être accessible par tous les étudiants connectés.
- ☒ Seul l'élève qui a partagé le document peut le modifier et/ou le supprimer. (2 pts)
- ☒ Pour créer le répertoire, vous devez ajouter un tableau dans la base de données
- ☒ il est important d'enregistrer le titre du document et la date, les titres peuvent être rédigés en français et en anglais,
- ☒ et le système doit gérer la langue de choix lors de la publication. (2pts)
- ☒ Le répertoire des fichiers doit être affiché dans un tableau avec la technique de pagination, le titre et le nom de l'utilisateur qui a partagé le fichier doivent être visibles sous ce tableau. (2 pt)
- ☒ Le système doit avoir une interface sécurisée, conviviale avec du CSS, Bootstrap et un menu de navigation. (3pts)
- ☒ Tous les formulaires doivent être validés, y compris les formulaires développés dans le TP1 (2pts).
- ☒ Publier votre projet dans GitHub (doit être publique)
- ☒ et envoyer le lien dans la documentation. (1 pt)

```
https://github.com/fhmaisonneuve/s4CadricielWeb\_TP2\_Laravel-Public
```

Déposez le fichier ZIP, sur Léa - Omnivox.

Vous devez m'envoyer sur Omnivox :

- ☒ Enregistrez le projet avec une extension ZIP et
- ☒ ajoutez la documentation (lisez-moi) à la racine du projet et la déposer dans omnivox
- ☒ Ajoutez dans la documentation le lien pour y accéder le projet sur WebDev ou un autre serveur public,

```
voir readme
```

- ☒ aussi comme le nom d'utilisateur et le mot de passe d'un étudiant enregistré dans le système.

```
User: francois@francois.com  
Password: francois
```

- ☒ Ajouter dans le document le lien GitHub.
-