



**ÉTABLISSEMENT D'ENSEIGNEMENT
POUR ADULTES ET DE FORMATION CONTINUE**
EAFC SUD-LUXEMBOURG



**WALLONIE-BRUXELLES
ENSEIGNEMENT**

Projet de développement Web

Hanut Audric

2023-2024

Table des matières

1	Déploiement	2
2	Documentation	3
2.1	Base de données	3
2.1.1	Schéma de la database	3
2.1.2	Implémentation	4
2.2	Utilisation du pattern de conception MVC	5
2.2.1	Routes	5
2.2.2	Modèles	5
2.2.3	Vues	6
2.2.4	Contrôleurs	8
2.3	Autres outils Laravel utilisés	8
2.3.1	Requests	8
2.3.2	Middlewares	9
2.3.3	Policies	9
3	Mode d'emploi utilisateur	10
3.1	Première utilisation	10
3.2	Utilisation	11
3.3	Deconnexion	14

Déploiement

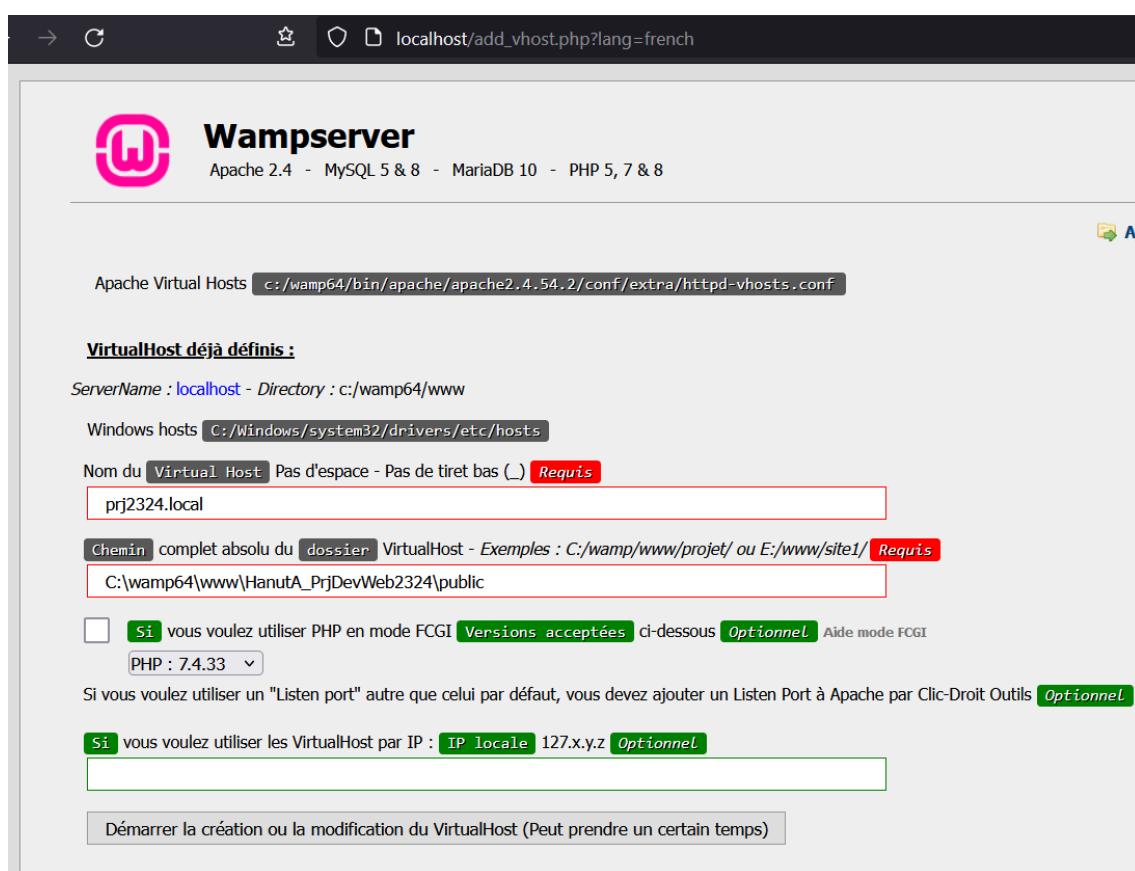
Le projet est pré-déployé sur la machine virtuelle (VM) fournie. Voici les étapes réalisées :

1. Placer le code source du site web dans le répertoire suivant :

- `C:\wamp64\www\NomDuProjet`

NomDuProjet est dans notre exemple : `HanutA_PrjDevWeb2324`

2. Créer le virtual host sur Wamp64 : dans ce cas, le lien vers le projet sera : `http://prj2324.local`



The screenshot shows the Wampserver web interface for adding a new VirtualHost. The browser address bar shows `localhost/add_vhost.php?lang=french`. The Wampserver logo and version information (Apache 2.4, MySQL 5 & 8, MariaDB 10, PHP 5, 7 & 8) are at the top. The 'Apache Virtual Hosts' section shows the configuration file path: `c:/wamp64/bin/apache/apache2.4.54.2/conf/extra/httpd-vhosts.conf`. Below, the 'VirtualHost déjà définis :' section shows the current configuration for `localhost` with directory `c:/wamp64/www` and hosts file `C:/Windows/system32/drivers/etc/hosts`. The form for adding a new VirtualHost includes: a 'Nom du Virtual Host' field with the value `prj2324.local` and a red 'Requis' label; a 'Chemin' field with the value `C:\wamp64\www\HanutA_PrjDevWeb2324\public` and a red 'Requis' label; a checkbox for 'Si vous voulez utiliser PHP en mode FCGI' with a dropdown set to 'PHP : 7.4.33'; a checkbox for 'Si vous voulez utiliser les VirtualHost par IP' with a dropdown set to 'IP locale' and a value `127.x.y.z`; and a 'Démarrer la création ou la modification du VirtualHost' button at the bottom.

FIGURE 1.1 – Interface de création d'un VirtualHost via Wamp64

Documentation

2.1 Base de données

2.1.1 Schéma de la database

Database :

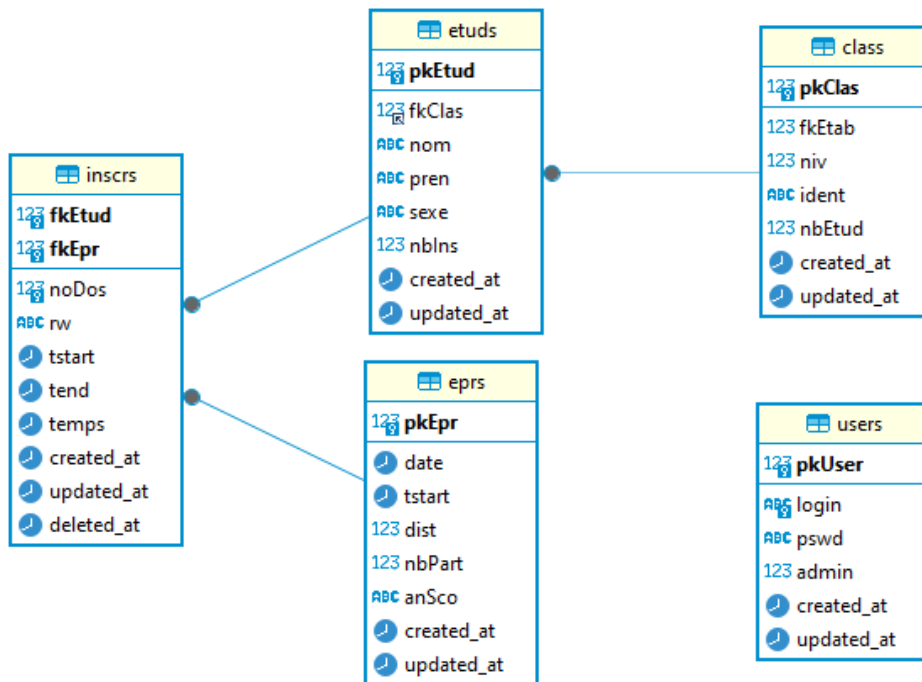


FIGURE 2.1 – Schéma de la database du projet

- Par rapport à l'ébauche de schéma relationnel de la DB de départ, quelques modifications ont été apportées :
 - Ajout des colonnes `created_at` et `updated_at` ; attributs générés par défaut par laravel que j'ai choisi de laisser, si jamais nous décidons des triers des informations sur leur date de création et/ou de mise à jour
 - Suppression de la clé primaire `PkInscr` de la table `Inscrs` : J'ai choisi d'utiliser une clé primaire composite de `[fkEtud ;fkEpr]` afin de mieux contrôler la duplication de données dans cette table

2.1.2 Implémentation

J'ai utilisé l'outil de migration de Laravel pour créer la base de données. Voici les fichiers de migration utilisés :

Table	Fichier source
Users	database/migrations/2014_10_12_000000_create_users_table.php
Class	database/migrations/2014_10_12_000000_create_class_table.php
Etuds	database/migrations/2014_10_12_000000_create_etuds_table.php
Eprs	database/migrations/2014_10_12_000000_create_eprs_table.php
Inscrs	database/migrations/2014_10_12_000000_create_inscrs_table.php

TABLE 2.1 – Listes des fichiers migrations utilisés

Pour la configuration et la connexion à la base de données dans le projet, j'ai utilisé les fichiers suivants :

- `.env`
 - Configuration de l'accès à la base de données.
- `app/Providers/AppServiceProvider.php`
 - Définition de la longueur par défaut des chaînes de caractères dans la base de données.
- `config/database.php`
 - Définition du moteur de stockage InnoDB.

2.2 Utilisation du pattern de conception MVC

J'ai utilisé le patron de conception Modèle-Vue-Contrôleur (MVC) dans le projet. Une documentation complète est présente dans le code source.

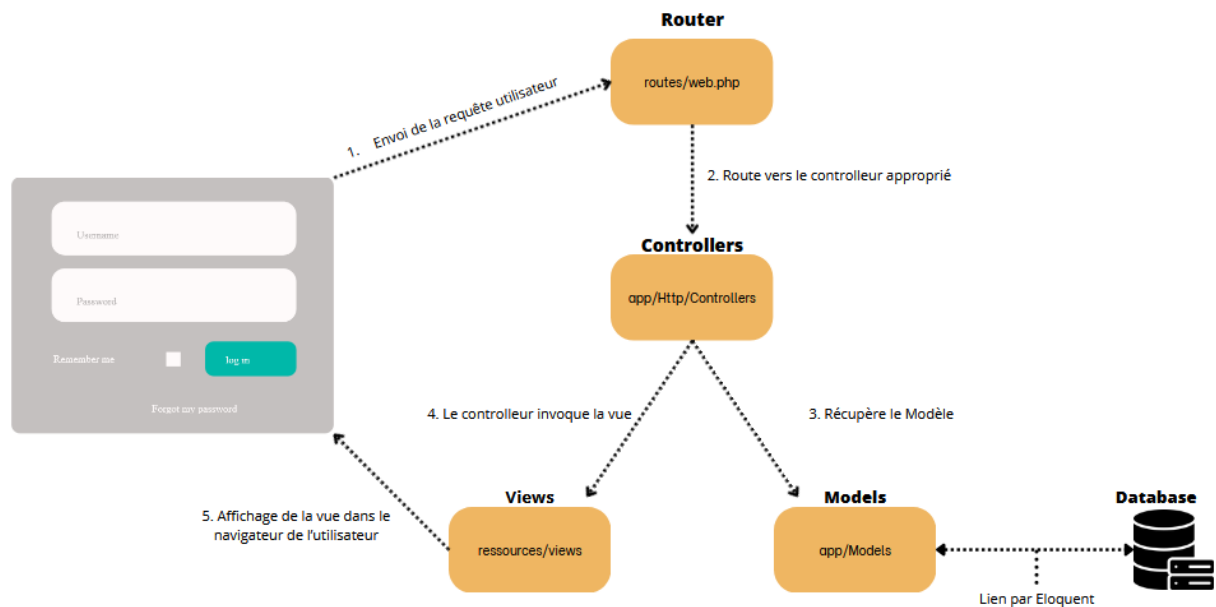


FIGURE 2.2 – Schéma présentant l'architecture d'un projet Laravel - MVC

2.2.1 Routes

Les différentes routes sont implémentées dans le fichier suivant :

- routes/web.php

2.2.2 Modèles

Création des modèles suivants pour représenter les différentes entités de l'application :

Model	Description	Fichier source
Users	Représentation d'un utilisateur de notre Site Web.	app/Models/User.php
Class	Ce modèle représente une classe dans l'application.	app/Models/Clas.php
Etuds	Ce modèle représente un étudiant dans l'application.	app/Models/Etud.php
Eprs	Représente une épreuve.	app/Models/Epr.php
Inscrs	Représente le modèle d'une inscription à une épreuve pour un étudiant.	app/Models/Inscr.php

TABLE 2.2 – Liste des Models utilisés

2.2.3 Vues

Layouts

Layout de base de ma page HTML

- `resources/views/Layouts/default.blade.php`

Un autre layout qui implémente le default est utilisé, celui-ci inclus la navbar et sera donc utilisé sur les pages où l'utilisateur est authentifié.

- `resources/views/Layouts/default_with_navbar.blade.php`

Vues

Vue	Description	Fichier source
Admin.create	Vue du formulaire de création de l'administrateur	<code>resources/views/Admin/create.blade.php</code>
Session.create	Vue du formulaire de connexion	<code>resources/views/Session/create.blade.php</code>
Session.delete	Vue de confirmation de déconnexion	<code>resources/views/Session/delete.blade.php</code>
Welcome	Vue de la page d'accueil, affichant les statistiques	<code>resources/views/welcome.blade.php</code>
Clas.create	Vue du formulaire de création d'une classe	<code>resources/views/Clas/create.blade.php</code>
Clas.delete	Vue du formulaire de suppression d'une classe	<code>resources/views/Clas/delete.blade.php</code>
Clas.edit	Vue du formulaire d'édition d'une classe	<code>resources/views/Clas/edit.blade.php</code>
Clas.index	Vue de l'index reprenant les infos de toutes les classes encodées	<code>resources/views/Clas/index.blade.php</code>
Etud.create	Vue du formulaire de création d'un étudiant	<code>resources/views/Etud/create.blade.php</code>
Etud.delete	Vue du formulaire de suppression d'un étudiant	<code>resources/views/Etud/delete.blade.php</code>
Etud.edit	Vue du formulaire d'édition d'un étudiant	<code>resources/views/Etud/edit.blade.php</code>
Etud.index	Vue de l'index reprenant les infos de toutes les étudiants encodées	<code>resources/views/Etud/index.blade.php</code>
Epr.create	Vue du formulaire de création d'une épreuve	<code>resources/views/Epr/create.blade.php</code>
Epr.delete	Vue du formulaire de suppression d'une épreuve	<code>resources/views/Epr/delete.blade.php</code>
Epr.edit	Vue du formulaire d'édition d'une épreuve	<code>resources/views/Epr/edit.blade.php</code>
Epr.index	Vue de l'index reprenant les infos de toutes les épreuves encodées	<code>resources/views/Epr/index.blade.php</code>
Inscr.show	Vue des détails d'une inscription à une épreuve.	<code>resources/views/Inscr/show.blade.php</code>
Inscr.edit	Vue du formulaire pour l'inscription à une épreuve spécifique.	<code>resources/views/Inscr/edit.blade.php</code>
Inscr.index	Vue de l'index permettant de choisir l'épreuve des inscriptions à afficher	<code>resources/views/Inscr/index.blade.php</code>
Arriv.edit	Vue du formulaire permettant d'encoder les arrivées	<code>resources/views/Arriv/edit.blade.php</code>
Arriv.index	Vue de l'index permettant de choisir l'épreuve pour laquelle encodées les arrivées	<code>resources/views/Arriv/index.blade.php</code>

TABLE 2.3 – Liste des vues utilisées

Components

Utilisation de composants blade afin d'utiliser la réutilisabilité de ceux-ci dans plusieurs vues

Component	Description	Fichier source
FlashMessage	Composant utilisé pour afficher un message éphémère à l'utilisateur. Il peut être utilisé pour afficher des messages d'alerte, de succès, etc.	<ul style="list-style-type: none"> — <code>resources/views/components/flash-message.blade.php</code> — <code>app/View/Components/FlashMessage.php</code>
Breadcrumb	Composant utilisé pour afficher une barre de navigation en utilisant des éléments de navigation spécifiés.	<ul style="list-style-type: none"> — <code>resources/views/components/breadcrumb.blade.php</code> — <code>app/View/Components/Breadcrumb.php</code>
ClasTable	Composant Blade pour afficher une table de classes.	<ul style="list-style-type: none"> — <code>resources/views/components/clas-table.blade.php</code> — <code>app/View/Components/ClasTable.php</code>
Search	Composant Blade pour une barre de recherche.	<ul style="list-style-type: none"> — <code>resources/views/components/search.blade.php</code> — <code>app/View/Components/Search.php</code>
ClasForm	Composant Blade pour afficher un formulaire de classe.	<ul style="list-style-type: none"> — <code>resources/views/components/clas-form.blade.php</code> — <code>app/View/Components/ClasForm.php</code>
DeleteForm	Composant Blade pour afficher un formulaire de suppression.	<ul style="list-style-type: none"> — <code>resources/views/components/delete-form.blade.php</code> — <code>app/View/Components/DeleteForm.php</code>
EtudTable	Composant Blade pour afficher un tableau d'étudiants.	<ul style="list-style-type: none"> — <code>resources/views/components/etud-table.blade.php</code> — <code>app/View/Components/EtudTable.php</code>
EtudForm	Composant Blade qui représente un formulaire pour la création ou la modification d'un étudiant.	<ul style="list-style-type: none"> — <code>resources/views/components/etud-form.blade.php</code> — <code>app/View/Components/EtudForm.php</code>
EprTable	Composant Blade pour représenter un tableau d'affichage des épreuves.	<ul style="list-style-type: none"> — <code>resources/views/components/epr-table.blade.php</code> — <code>app/View/Components/EprTable.php</code>
EprForm	Composant Blade pour représenter un formulaire pour les épreuves.	<ul style="list-style-type: none"> — <code>resources/views/components/epr-form.blade.php</code> — <code>app/View/Components/EprForm.php</code>
ArrivForm	Composant Blade pour afficher le formulaire d'arrivée des étudiants à une épreuve.	<ul style="list-style-type: none"> — <code>resources/views/components/arriv-form.blade.php</code> — <code>app/View/Components/ArrivForm.php</code>
EditInscrForm	Composant Blade pour afficher le formulaire de modification des inscriptions à une épreuve.	<ul style="list-style-type: none"> — <code>resources/views/components/edit-inscr-form.blade.php</code> — <code>app/View/Components/EditInscrForm.php</code>
ShowInscrForm	Composant pour afficher le formulaire de détails d'une inscription à une épreuve.	<ul style="list-style-type: none"> — <code>resources/views/components/show-inscr-form.blade.php</code> — <code>app/View/Components/ShowInscrForm.php</code>
ArrivTable	Composant Blade pour afficher le tableau de détails d'arrivée des étudiants à une course.	<ul style="list-style-type: none"> — <code>resources/views/components/arriv-table.blade.php</code> — <code>app/View/Components/ArrivTable.php</code>

TABLE 2.4 – Liste des composants utilisés

2.2.4 Contrôleurs

Controller	Description	Fichier source
AdminController	Ce contrôleur gère la création et le stockage de nouveaux administrateurs.	app/Http/Controllers/AdminController.php
SessionController	Ce contrôleur gère les opérations de gestion de session utilisateur, telles que la connexion, la déconnexion, etc.	app/Http/Controllers/SessionController.php
ClasController	Ce contrôleur gère les opérations liées aux classes dans l'application.	app/Http/Controllers/ClasController.php
EtudController	Contrôleur pour la gestion des étudiants.	app/Http/Controllers/EtudController.php
EprController	Gère les opérations liées aux épreuves.	app/Http/Controllers/EprController.php
InscrController	Contrôleur pour la gestion des inscriptions aux épreuves.	app/Http/Controllers/InscrController.php
ArrivController	Contrôleur gérant les arrivées des étudiants aux épreuves.	app/Http/Controllers/ArrivController.php
StatsController	Contrôleur générant les statistiques.	app/Http/Controllers/StatsController.php

TABLE 2.5 – Liste des controllers utilisés

2.3 Autres outils Laravel utilisés

2.3.1 Requests

Les requests dans Laravel servent à valider et à traiter les données provenant des requêtes HTTP entrantes. Ils permettent de valider les données en fonction de règles définies, de simplifier le processus de récupération des données et de centraliser la logique de traitement des données dans des classes dédiées

Requests	Description	Fichier source
AdminRequest	représente une requête pour un administrateur	app/Http/Requests/AdminRequest.php
ClasRequest	Requête de validation pour la création ou la mise à jour d'une classe.	app/Http/Requests/ClasRequest.php
EprRequest	Valide les données de la requête pour la création ou la mise à jour d'une épreuve.	app/Http/Requests/EprRequest.php
EtudRequest	Valide les requêtes de création ou de mise à jour d'un étudiant.	app/Http/Requests/EtudRequest.php
InscrRequest	Valide les requêtes de création ou de mise à jour d'une inscription à une épreuve.	app/Http/Requests/InscrRequest.php
SessionRequest	Cette classe représente une requête de session utilisateur.	app/Http/Requests/SessionRequest.php

TABLE 2.6 – Liste des Requests utilisés

2.3.2 Middlewares

Les middlewares dans Laravel servent à filtrer les requêtes HTTP entrantes de l'application avant qu'elles n'atteignent les routes ou les contrôleurs. Ils permettent d'ajouter des couches de logique intermédiaire pour effectuer des tâches telles que l'authentification, l'autorisation, la manipulation des données de requête, la gestion des sessions,...

Middleware	Description	Fichier source
FirstTimeSetup	Ce middleware vérifie si c'est la première fois que le setup est effectué dans l'application. S'il n'y a aucun utilisateur enregistré dans la base de données, il redirige l'utilisateur vers la page d'inscription de l'administrateur.	<ul style="list-style-type: none"> — <code>app/Http/Middleware/FirstTimeSetup.php</code> — <code>app/Http/Kernel.php</code>
AdminCreated	Ce middleware vérifie si un administrateur a déjà été créé dans l'application. Si un utilisateur est enregistré dans la base de données, il redirige l'utilisateur vers la page d'accueil.	<ul style="list-style-type: none"> — <code>app/Http/Middleware/AdminCreated.php</code> — <code>app/Http/Kernel.php</code>

TABLE 2.7 – Liste des middlewares utilisés

2.3.3 Policies

Les politiques (policies) dans Laravel servent à définir des règles d'autorisation pour les différentes actions sur des modèles de données. Elles permettent de centraliser la logique d'autorisation dans des classes dédiées, en spécifiant quelles actions un utilisateur peut effectuer sur un modèle donné. Les politiques sont souvent utilisées pour déterminer si un utilisateur peut créer, afficher, mettre à jour ou supprimer une instance particulière d'un modèle.

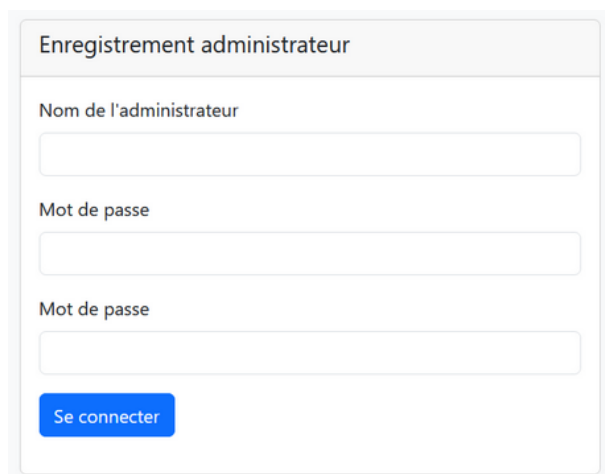
Policy	Description	Fichier source
ClasPolicy	Politique d'accès pour les classes.	<code>app/Policies/ClasPolicy.php</code>
EtudPolicy	Politique d'accès pour les étudiants.	<code>app/Policies/EtudPolicy.php</code>
EprPolicy	Politique d'accès pour les épreuves.	<code>app/Policies/EprPolicy.php</code>
InscrPolicy	Politique d'accès pour les inscriptions.	<code>app/Policies/InscrPolicy.php</code>

TABLE 2.8 – Liste des policy utilisés

Mode d'emploi utilisateur

3.1 Première utilisation

Lors de la première utilisation, il est impossible d'accéder au site web sans enregistrer un administrateur :



Enregistrement administrateur

Nom de l'administrateur

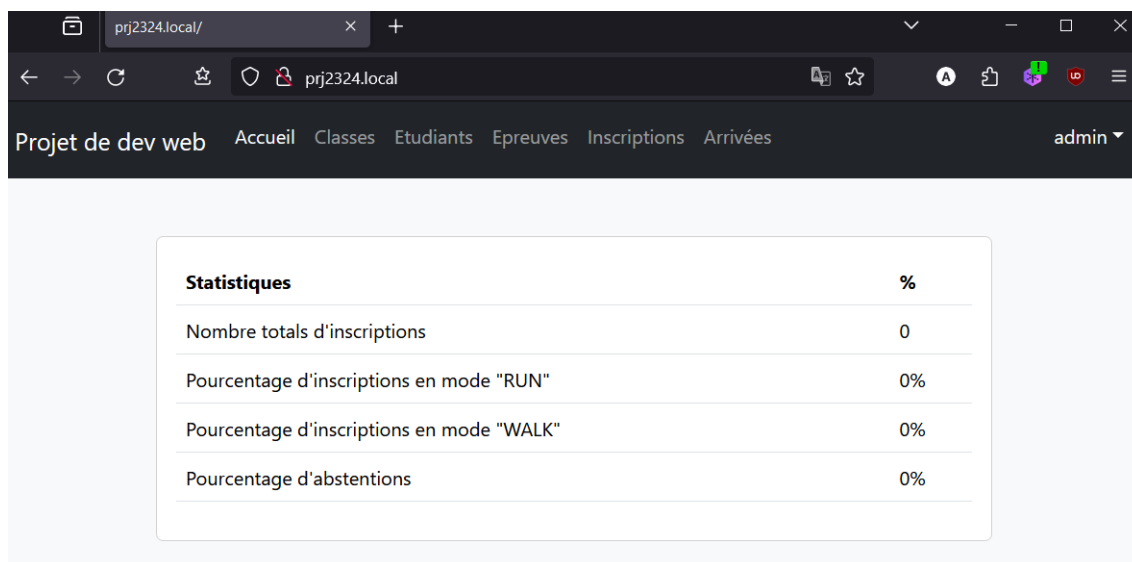
Mot de passe

Mot de passe

Se connecter

FIGURE 3.1 – Formulaire de création du compte Administrateur

Une fois l'administrateur créé, celui-ci est automatiquement connecté et arrive sur la page d'accueil, reprenant les différentes statistiques.



prj2324.local/

prj2324.local

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées admin

Statistiques	%
Nombre totaux d'inscriptions	0
Pourcentage d'inscriptions en mode "RUN"	0%
Pourcentage d'inscriptions en mode "WALK"	0%
Pourcentage d'abstentions	0%

FIGURE 3.2 – Page d'accueil

3.2 Utilisation

Le menu permet d'accéder aux différentes possibilités du site web.

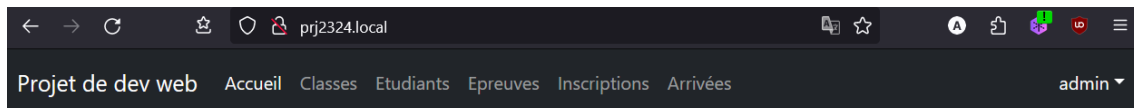


FIGURE 3.3 – Navbar du site web

La structure pour les Classes, les Etudiants et les Epreuves est identiques. La page principal permet de visualiser la liste des éléments, de faire une recherche sur ceux-ci, d'accéder aux formulaire de création de l'élément et d'afficher les détails de ceux-ci :

1. Permet d'effectuer une recherche d'un élément
2. Permet d'accéder au formulaire de création de l'élément
3. Permet d'afficher un élément, ce qui permet de voir / éditer / supprimer celui-ci
4. Permet de naviguer parmit les éléments quand ceux-ci sont supérieur à 10.

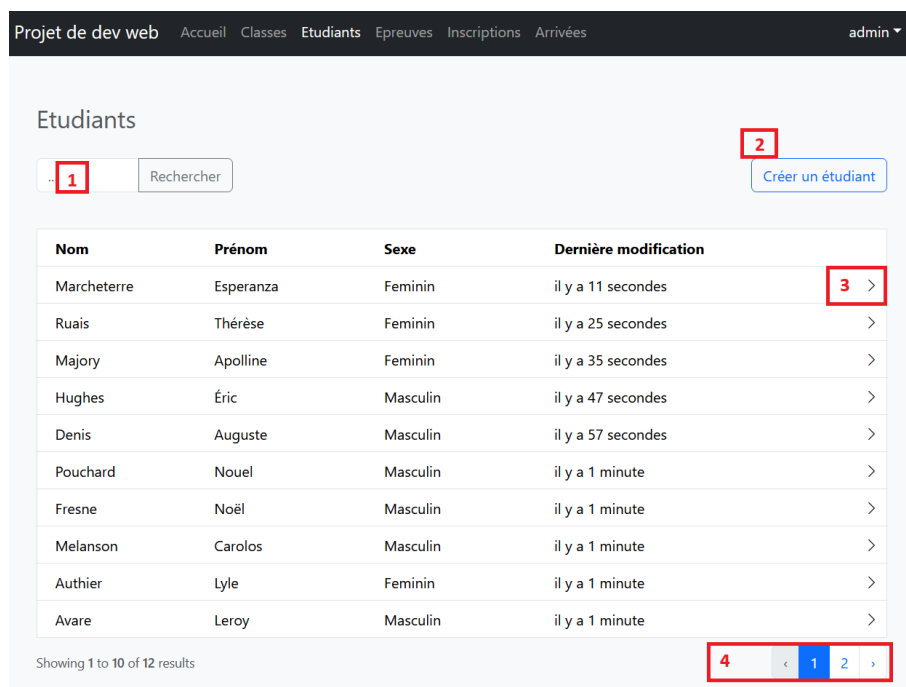


FIGURE 3.4 – Page CRUD Etudiant

Affichage de l'élément

1. Zone d'affichage / d'édition de l'élément
2. Accéder a la suppression de l'élément (avec confirmation)
3. Enregistre les modifications effectuée dans la zone d'édition

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées admin

Etudiants / Denis

1

Nom
Denis

Prénom
Auguste

☒ Masculin
☐ Féminin

Classe
Bac Info

Supprimer 2

3 Enregistrer

Liste des inscriptions de l'étudiant

... Rechercher

Créer une épreuve

Date	Heure de départ	Distance	Nb parts	Année scolaire	Dernière modif.
------	-----------------	----------	----------	----------------	-----------------

FIGURE 3.5 – Formulaire création Etudiant

Pour les inscriptions, il nous faut sélectionner l'épreuve qui nous interesse

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées

Inscriptions

Selectionnez l'épreuve

Entrer

Selectionnez l'épreuve

2024-02-28

FIGURE 3.6 – Formulaire de sélection d'une épreuve

Nous sommes alors rediriger vers la page d'inscription, pour l'inscription il faut une heure (si pas encodé dans l'épreuve) et un type de coureur (run/walk) : Ensuite appuyé sur le + Vert afin de valider l'inscription.

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées admin

[Inscriptions](#) / 2024-02-28

... Rechercher

Etudiants non inscrit

Nom	Prénom		
Dupont	John	17:30	Walk +
Tachel	Normand	00:00	+ -
Avare	Leroy	00:00	+ -

Etudiants inscrit

Nom	Prénom	N° dossard
Aucun élément trouvé		

FIGURE 3.7 – Formulaire d'inscription

L'étudiant devrait alors apparaître dans la liste des étudiants inscrits

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées admin

Etudiant inscrit

[Inscriptions](#) / 2024-02-28

... Rechercher

Etudiants non inscrit

Nom	Prénom		
Tachel	Normand	00:00	+ -
Avare	Leroy	00:00	+ -

Etudiants inscrit

Nom	Prénom	N° dossard
Dupont	John	1001

FIGURE 3.8 – Formulaire d'inscription

- Le '- Rouge' permet alors de désinscrire l'étudiant (uniquement si il n'a pas fini l'épreuve).
 - Le 'i bleu' permet d'afficher et d'éditer (si l'épreuve n'est pas terminé) l'inscription.
- Pour les arrivées, il nous faut sélectionner l'épreuve qui nous interesse

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées

Arrivées

Selectionnez l'épreuve

2024-02-28

Entrer

FIGURE 3.9 – Formulaire de sélection d'une épreuve

Il faut encodé le n° de dossard (trouvable dans les tableaux d'inscriptions aux épreuves) et enregistré :

Projet de dev web Accueil Classes Etudiants Epreuves Inscriptions Arrivées admin

Arrivées / 2024-02-28

Encodage des arrivées

Nombre d'étudiants arrivés : 1/1

N° Dossard 1001 Enregistrer

WALK				RUN			
#	Nom	Prénom	Temps	#	Nom	Prénom	Temps
Aucun élément trouvé				1	Dupont	John	00:55:47

FIGURE 3.10 – Formulaire d'encodage des arrivées

3.3 Deconnexion

Pour se déconnecter, il suffit de cliquer en haut à droite de l'application, sur son nom d'utilisateur et de sélectionner "Se déconnecter"

admin

Se déconnecter

FIGURE 3.11 – Formulaire de déconnexion