

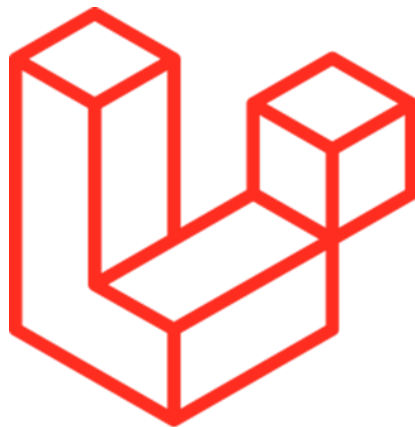


المدرسة العليا للتكنولوجيا الناصور
École Supérieure de Technologie de Nador
ⵜⴰⵎⴰⵔⵜ ⵜⴰⵎⴰⵏⴰⵢⵜ ⵜⴰⵏⴰⵔⵉⵜ ⵜⴰⵙⴰⵎⴰⵏⴰⵢⵜ ⵜⴰⵖⴰⵏⴰⵏⵜ

Développement web avancé

Filière : Ingénierie Logicielle et Cybersécurité (ILCS)

Projet Fin de module



Réalisé par : IBTIHAL LETRACH

IHSSANE OUKHSSANE

Première année, Semestre S2

Professeur : RABHI Ouzayr

Année universitaire : 2024/2025

Résumé :

Le projet **DevProfile** est une application web développée avec le framework **Laravel**, visant à créer une plateforme de gestion de profils pour les développeurs. L'objectif principal est de permettre aux utilisateurs (développeurs) de créer, modifier et partager leur profil professionnel de manière structurée, tout en facilitant la consultation de ces profils par des recruteurs ou d'autres utilisateurs.

Ce projet a été réalisé dans le cadre d'un apprentissage approfondi du développement web moderne avec Laravel. Il répond à un besoin concret : centraliser les informations clés d'un développeur dans une interface claire et accessible. Le site offre des fonctionnalités essentielles telles que l'**authentification**, la **gestion des profils**, le **téléversement de fichiers (CV, photo, projets)**, un **tableau de bord administrateur**, et un **design responsive** assurant une compatibilité optimale sur tous les appareils.

D'un point de vue technique, **DevProfile** repose sur une architecture **MVC (Model-View-Controller)**, utilise **Blade** pour le rendu des vues, **MySQL** pour la base de données, et intègre **Bootstrap ou Tailwind CSS** pour l'interface utilisateur. Le projet est versionné avec **Git** et hébergé sur **GitHub**, garantissant une bonne traçabilité et un travail collaboratif possible.

Au cours du développement, plusieurs défis techniques ont été relevés, notamment la sécurisation de l'accès, la gestion des fichiers, et l'organisation logique des différentes routes et vues. Ce projet a permis de mettre en pratique l'ensemble des compétences acquises en développement web back-end et front-end.

En conclusion, **DevProfile** constitue une expérience complète de gestion de projet Laravel, depuis l'analyse des besoins jusqu'au déploiement final. Il représente un aboutissement pédagogique significatif et ouvre la voie à des projets web plus complexes et collaboratifs.

Tableau de matières :

Résumé-----	2
Introduction-----	4
Objectifs du projet-----	4
Méthodologie-----	5
Cahier des charges-----	5
Technologie et outils utilisés-----	7
Réalisation de l'application-----	9
Architecture de l'application-----	11
Fonctionnalités développées-----	13
Conclusion-----	15
Bibliographie-----	16

Introduction :

Dans un monde numérique en constante évolution, le besoin de se démarquer professionnellement est devenu une priorité, surtout pour les développeurs. Que ce soit pour intégrer une entreprise, collaborer sur un projet open-source ou simplement exposer ses compétences, disposer d'un profil professionnel bien structuré et facilement accessible est aujourd'hui essentiel.

C'est dans ce contexte que s'inscrit DevProfile, une application web visant à simplifier la gestion et la diffusion de profils pour les développeurs. Ce projet a été conçu dans le cadre d'un apprentissage approfondi du framework Laravel, et répond à une problématique concrète : comment offrir aux développeurs une solution simple, intuitive et sécurisée pour présenter leurs compétences, expériences et réalisations ?

Objectifs du projet

Le principal objectif de DevProfile est de développer une **application web professionnelle**, stable et ergonomique, permettant :

- aux utilisateurs de **créer, modifier et supprimer leur profil** développeur ;
- de **gérer les informations clés** : compétences, projets, formations, expériences, liens GitHub ou LinkedIn ;
- d'offrir un **espace d'administration** pour gérer les comptes et les profils ;
- d'assurer un **design moderne et responsive** accessible sur tous types de supports ;
- de mettre en œuvre les **principes fondamentaux de Laravel** : architecture MVC, migrations, routage, middleware, système d'authentification, etc.

Méthodologie

Le projet a été réalisé en suivant une démarche incrémentale :

1. Analyse des besoins fonctionnels et techniques
2. Conception de l'architecture de l'application
3. Développement des différentes fonctionnalités (authentification, gestion des profils...)
4. Tests, débogage et améliorations
5. Documentation et préparation à la mise en production

Cahier des charges

1. Présentation générale

L'objectif principal de l'application **DevProfile** est de proposer une plateforme web destinée aux développeurs pour présenter leur profil professionnel de manière claire, organisée et attractive. Elle doit permettre à tout développeur inscrit de renseigner ses compétences, projets, expériences, formations, et liens vers ses portfolios externes.

L'application sera développée en **Laravel**, et reposera sur une architecture MVC (Model-View-Controller), avec une base de données **MySQL** pour la gestion des données utilisateurs et profils.

2. Besoins fonctionnels

L'application devra permettre les fonctionnalités suivantes :

Pour les utilisateurs authentifiés :

- Créer un compte développeur
- S'authentifier (connexion/déconnexion)
- Accéder à un tableau de bord personnalisé
- Ajouter, modifier ou supprimer :
 - Informations personnelles (nom, prénom, email...)
 - Compétences techniques

- Projets réalisés (avec lien GitHub éventuellement)
- Expériences professionnelles
- Formations et certifications
- Réseaux sociaux professionnels (LinkedIn, GitHub...)

Pour l'administrateur :

- Voir la liste de tous les profils
- Supprimer ou désactiver un profil
- Gérer les utilisateurs inscrits

3. Contraintes techniques

- Utilisation du framework **Laravel** (version 9 ou supérieure)
- Base de données **MySQL**
- Système d'authentification Laravel (Sanctum ou Breeze)
- Interface développée avec **Blade** (template engine)
- Déploiement local (XAMPP et Laravel Serve)
- Gestion de versions avec **Git & GitHub**

Technologies et outils utilisés

1. Technologies principales

Laravel : Framework PHP utilisé pour structurer l'application, faciliter le routage, l'authentification et la gestion des bases de données via Eloquent ORM.

PHP : Langage de programmation backend principal.

MySQL : Système de gestion de base de données relationnelle.

Blade : Moteur de templates intégré à Laravel pour générer les vues HTML dynamiquement.

HTML5/CSS3 : Technologies de base pour le rendu visuel et la mise en forme des pages.

JavaScript : Utilisé pour des interactions dynamiques dans certaines vues (par ex. validations front).

Laravel Breeze : un kit de démarrage officiel et léger pour Laravel, conçu pour implémenter rapidement un système d'authentification simple et fonctionnel.

2. Outils et environnements

Visual Studio Code : Environnement de développement (éditeur de code).

Git : Système de gestion de versions pour suivre les modifications du code.

GitHub : Hébergement du code source du projet et collaboration.

**XAMPP / Laravel
Serve :** Serveur local pour tester l'application.

Composer : Gestionnaire de dépendances PHP, utilisé pour installer Laravel et ses packages.

phpMyAdmin : Interface web pour gérer la base de données MySQL.

3. Pourquoi Laravel ?

- Intégration native de fonctionnalités de sécurité (auth, middleware, CSRF).
- Organisation MVC claire pour la maintenabilité.
- Artisan CLI pour générer des modèles, contrôleurs, migrations, etc.
- Moteur Blade pour générer des vues dynamiques facilement.
- Facilité à implémenter des APIs REST ou interfaces web classiques.

Réalisation de l'application

1. Environnement de développement

- **Langage principal** : PHP (Laravel)
- **Framework** : Laravel 10
- **Base de données** : MySQL
- **Serveur local** : Laravel Serve / XAMPP
- **IDE utilisé** : Visual Studio Code
- **Contrôle de version** : Git, avec dépôt distant sur GitHub

2. Structure du projet Laravel

L'application suit l'architecture MVC (Modèle-Vue-Contrôleur) :

- **Models** : représentent les entités comme User, Skill, Project, etc.
- **Controllers** : contiennent la logique métier (UserController, DashboardController, etc.)

- **Views (Blade)** : gèrent l'interface utilisateur
- **Routes** : définies dans web.php (interfaces publiques et privées)
- **Migrations** : pour la structure des tables
- **Seeders** : pour insérer des données de test

3. Fonctionnalités implémentées

Authentification

- Inscription, connexion, déconnexion
- Vérification des emails (optionnelle)
- Gestion des sessions sécurisées

Gestion de profil

- Modification des informations personnelles
- Ajout de compétences (skills)
- Ajout de projets (liens, descriptions, stack technique)
- Ajout d'expériences professionnelles
- Ajout de formations

Tableau de bord

- Affichage résumé des données du profil
- Statistiques sur les projets/skills
- Interface responsive

Interface d'administration (optionnelle)

- Consultation des utilisateurs
- Suppression/désactivation de comptes

Architecture de l'application

L'architecture de l'application **DevProfile** repose sur le modèle **MVC (Model-View-Controller)**, une structure classique dans le développement web qui permet une séparation claire des responsabilités :

Models

Les modèles représentent les entités principales de la base de données et contiennent la logique métier associée.

- User : représente les utilisateurs enregistrés. Gère les informations personnelles, les identifiants, et les relations avec les projets et compétences.
- Project : représente les projets liés à un utilisateur.
- Skill : représente les compétences liées à un utilisateur.

Les modèles utilisent **Eloquent**, l'ORM (Object Relational Mapper) de Laravel, pour gérer les relations one-to-many ou many-to-many.

Controllers

Les contrôleurs gèrent la logique applicative et orchestrent les échanges entre la vue et les modèles.

- `DashboardController` : affiche les informations de l'utilisateur dans le tableau de bord.
- `ProjectController` : gère les opérations CRUD sur les projets.
- `SkillController` : gère les compétences de l'utilisateur.
- `ProfileController` : permet la modification des informations personnelles.
- `PdfController` : génère le CV au format PDF via DomPDF.

Tous les contrôleurs sont protégés par le middleware auth, sauf ceux pour l'accès public au profil.

Views

Les interfaces utilisateur sont générées avec **Blade**, le moteur de templates de Laravel.

- `dashboard.blade.php` : tableau de bord de l'utilisateur.
- `profile/show.blade.php` : affichage du profil public.
- `projects/index.blade.php` : liste des projets.
- `skills/index.blade.php` : liste des compétences.
- `pdf/cv.blade.php` : vue dédiée à l'exportation PDF.

Le layout général est géré via `layouts/app.blade.php`.

Routes

Les routes sont définies dans `routes/web.php`. Elles sont organisées avec des **groupes de middleware** (auth, verified, etc.) et des **routes nommées** (`name()`), facilitant leur utilisation dans les vues et redirections.

Autres composants

- **Migrations** : définissent la structure des tables de la base de données.
- **Factory & Seeder** : permettent de générer des données factices pour les tests.
- **Middleware** : gèrent les accès, notamment l'authentification.

- **Service Providers** : configurent les composants de l'application.

Fonctionnalités Développées

L'application **DevProfile** a été pensée pour offrir aux développeurs une plateforme simple, efficace et personnalisée de gestion de leur profil professionnel. Voici les principales fonctionnalités mises en œuvre :

1 .Authentification

L'authentification a été gérée avec **Laravel Breeze**, un starter kit léger et rapide. Fonctionnalités incluses :

- Création de compte (inscription)
- Connexion sécurisée
- Déconnexion
- Validation de l'email
- Protection des routes avec middleware auth

Cette couche de sécurité permet de limiter l'accès à certaines pages à des utilisateurs authentifiés.

2. Profil Utilisateur

Chaque utilisateur peut :

- Modifier ses **informations personnelles** (nom, email, mot de passe)
- Télécharger une **photo de profil**
- Générer une **page publique** accessible via une URL unique contenant :
 - Son nom
 - Ses projets
 - Ses compétences

Cette page agit comme une **carte de visite numérique professionnelle**.

3 Gestion des Projets (CRUD)

Les utilisateurs peuvent gérer leurs projets via une interface intuitive :

- **Ajouter** un projet (titre, description, lien GitHub ou site, date, etc.)
- **Consulter** la liste de leurs projets
- **Modifier** les projets existants
- **Supprimer** un projet

Chaque projet est lié à un utilisateur et peut être affiché sur sa page publique.

4. Gestion des Compétences

Les utilisateurs peuvent :

- Ajouter leurs **compétences techniques** (ex. Laravel, JavaScript, MySQL)
- Supprimer des compétences obsolètes

Les compétences sont affichées dans leur profil public, sous forme de tags ou de liste.

5. Génération de CV en PDF

La plateforme propose une génération automatique d'un **CV en PDF**, exporté à partir de la page publique du profil.

Technologie utilisée : **laravel-dompdf**.

- Le design du CV respecte les standards professionnels.
- Les informations du CV incluent les données du profil, les projets et les compétences.
- Le CV est téléchargeable en un clic depuis le dashboard.

6. Responsive Design & Expérience Utilisateur

Grâce à **Tailwind CSS**, l'interface est :

- Responsive (compatible mobile, tablette, desktop)
- Minimaliste et professionnelle
- Facile à naviguer

Conclusion

Le développement de l'application **DevProfile** a été une expérience formatrice et enrichissante à plusieurs niveaux. Ce projet nous a permis de mettre en œuvre concrètement des compétences essentielles en développement web moderne avec le framework **Laravel**, en suivant les bonnes pratiques du modèle **MVC**.

Grâce à cette application, nous avons appris à :

- Concevoir et structurer une base de données relationnelle efficace,
- Mettre en œuvre une **authentification sécurisée** avec Laravel Breeze,
- Gérer des fonctionnalités **CRUD** complètes pour les projets et les compétences,
- Exploiter pleinement les **Blade templates** pour une interface utilisateur cohérente,
- Générer dynamiquement un **CV en PDF** grâce à laravel-dompdf,
- Gérer les versions du projet via **Git** et **GitHub**,
- Déployer une architecture modulaire et réutilisable.
- Au-delà des aspects purement techniques, **DevProfile** incarne aussi une réflexion sur l'importance de l'expérience utilisateur, de la **personnalisation**, de l'esthétique, et du **professionnalisme** dans le développement d'applications destinées à des utilisateurs finaux.

En somme, **ce projet marque un jalon important dans notre parcours de développeurs**. Il représente non seulement une réussite fonctionnelle, mais aussi une démonstration de notre capacité à concevoir des solutions numériques complètes, robustes et orientées utilisateur. Il constitue une base solide pour des projets futurs encore plus ambitieux dans le domaine du développement web.

Bibliographie :

Source	Description
Laravel Documentation	Documentation officielle du framework Laravel, utilisée comme référence principale.
Laravel Breeze	Kit de démarrage utilisé pour mettre en œuvre l'authentification.
Laravel DomPDF	Bibliothèque utilisée pour générer dynamiquement les CV en PDF.
Tailwind CSS	Framework CSS utilisé pour le design responsive et moderne.
MySQL Documentation	Référence pour la création et la gestion des bases de données relationnelles.
Git Documentation	Outil de gestion de version utilisé tout au long du projet.
GitHub	Plateforme d'hébergement du code source.