



Projet de Programmation Web :

Rapport

du projet « MyFashion »

Contributeurs et participations :

NOMS	MATRICULE	PARTICIPATION
KOUASSI DE YOBO GILBERT BRYAN (CHEF)	21P082	100%
MAFFO FONKOU NATACHA	21P272	100%
MBOCK JEAN DANIEL	21P269	100%
ATABONG STEPHANE	23P781	85%
BENGONO AMVELA NATHAN	21P091	85%
WOTCHOKO YOHAN	21P228	90%
NGOM CHRISTINE	21P336	75%
NGHOGUE TAPTUE FRANCK	21P279	80%
YIMBOU JUDE	21P417	65%
NOUNDJEU NOUBISSIE FRANCK	21P318	75%

SUPERVISEUR : Mme NDONGSONG ESTELLE

INTRODUCTION	1
PRÉSENTATION GÉNÉRALE	1
Contexte du projet.....	2
Objectif du projet.....	2
Description du projet.....	2
Périmètre du projet	3
FONCTIONNALITÉS DE L'APPLICATION	4
CONTRAINTES TECHNIQUES.....	5
Technologies	5
Compatibilité	6
SÉCURITÉ	7
Fonctionnement	7
Avantages du JWT.....	7
Inconvénients.....	7
CALENDRIER DU PROJET	8
Fonctionnalités Développées.....	8
Planification hebdomadaire du travail	8
RESSOURCES.....	12
Ressources Physiques	12
Ressources Logistiques : Technologies et Gestion	12
Déploiement	13
CONCLUSION	14

INTRODUCTION

Lors de la création d'une application web mono-page (single-page application ou SPA), il est bon d'utiliser un framework qui peut faire une partie du travail pour nous pour que nous puissions nous concentrer sur les fonctionnalités. AngularJS répond parfaitement à ce besoin, notamment grâce à l'injection de dépendances et au data binding. Parfois nous avons également besoin d'un serveur. Vu que notre choix est porté sur PHP alors Laravel est l'une des meilleures options, il est facile d'utilisation et très

puissant. Notre travail consiste à utiliser ces framework pour créer une application web permettant d'une part pour les stylistes, d'offrir une plateforme pour exposer leurs créations, présenter les prix et détails des tenues, et faciliter la gestion des commandes; d'autre part, pour les utilisateurs, de permettre de parcourir les créations, prendre leurs mesures via l'application, vérifier la disponibilité des stylistes, et effectuer des paiements en ligne. Tout au long de ce rapport nous allons faire une présentation générale de l'application, puis de ses fonctionnalités, ensuite nous allons rappeler les contraintes techniques, présenter les ressources et enfin nous allons présenter les livrables.

PRÉSENTATION GÉNÉRALE

Contexte du projet

Dans le cadre de notre formation en développement web, nous avons été amenés à réaliser un projet full stack permettant de mettre en pratique nos connaissances en Angular et Laravel. Le choix s'est porté sur une application de mise en relation entre stylistes et clients, qui présente des défis techniques intéressants pour explorer les différentes fonctionnalités de ces frameworks. Ce projet nous permet d'expérimenter le développement d'une application web complète, du frontend au backend, en utilisant des technologies modernes très demandées sur le marché du travail.

Objectif du projet

Ce projet s'inscrit dans une démarche pédagogique approfondie, visant à consolider nos compétences en développement web moderne. Notre objectif principal est de maîtriser deux frameworks essentiels du développement web : Angular pour la partie frontend et Laravel pour le backend. À travers la réalisation de cette application de couture sur mesure, nous cherchons à acquérir une compréhension complète de l'architecture d'une application web contemporaine. Cette expérience pratique nous permet d'explorer en profondeur les concepts du développement full stack, en mettant particulièrement l'accent sur la communication entre le frontend et le backend via des API RESTful. La gestion d'une base de données relationnelle MySQL constitue également un aspect crucial de notre apprentissage, nous permettant de comprendre les enjeux du stockage et de la manipulation des données dans un contexte professionnel. Ce projet représente ainsi une opportunité concrète de mettre en application les connaissances théoriques acquises durant notre formation, tout en développant des compétences pratiques essentielles pour notre future carrière dans le développement web.

Description du projet

Notre solution prend la forme d'une application web moderne, développée avec les technologies Angular et Laravel, offrant une expérience utilisateur riche et intuitive. La plateforme permet aux stylistes de créer leur vitrine virtuelle personnalisée, où ils peuvent mettre en valeur leurs créations à travers des photos haute qualité et des descriptions détaillées. L'innovation majeure réside dans notre système de prise de

mesures guidé par vidéo, qui simplifie considérablement cette étape crucielle du processus de commande. Les stylistes bénéficient d'un calendrier en temps réel pour gérer leurs disponibilités, tandis que les clients peuvent suivre l'avancement de leurs commandes de manière transparente. L'intégration des solutions de paiement mobile money locales garantit des transactions sécurisées et adaptées aux habitudes de paiement locales. Un espace communautaire vient enrichir l'expérience en permettant aux clients de partager leurs retours d'expérience et leurs photos.

Périmètre du projet

Le projet embrasse l'ensemble des aspects nécessaires à une expérience de couture sur mesure digitalisée. Du côté des clients, nous avons conçu une interface intuitive qui facilite la découverte des créations et la gestion des commandes. Chaque client dispose d'un espace personnel sécurisé où il peut sauvegarder ses mesures et suivre l'évolution de ses commandes en temps réel. Pour les stylistes, nous avons développé un espace professionnel complet intégrant la gestion du catalogue, le suivi des commandes, et un système de notification performant. Un tableau de bord analytique leur permet de piloter efficacement leur activité.

Sur le plan technique, notre application repose sur une architecture moderne et robuste. Le frontend, développé avec Angular, offre une expérience utilisateur fluide et responsive. Le backend, construit avec Laravel, assure une gestion efficace des données et une sécurité optimale. La base de données MySQL stocke les informations de manière structurée, tandis que notre API RESTful facilite la communication entre les différentes parties de l'application. L'intégration des solutions de paiement mobile garantit des transactions sécurisées.

Le déploiement de la plateforme suivra une approche progressive, débutant par une phase de test auprès d'un groupe sélectionné de stylistes et clients. Cette approche nous permettra d'affiner l'expérience utilisateur et d'optimiser les fonctionnalités avant un lancement à plus grande échelle.

FONCTIONNALITÉS DE L'APPLICATION

L'application myFashion possède plusieurs fonctionnalités à savoir :

- Consultation des tenues des stylistes par les clients ;
- Commande de tenues par les clients;
- Consultation de la disponibilité d'un styliste;
- Réservation d'une ou plusieurs journées de travail avec un styliste;
- Notation et commentaire des tenues par les clients ;
- Upload ses mensurations personnelles ou pour une commande ;
- Présentation des créations d'un styliste ;
- Dashboard avec détails sur les commandes et les revues des tenues par les clients.

CONTRAINTES TECHNIQUES

Technologies

Frontend : AngularJS

- **Description** : AngularJS est un framework JavaScript développé par Google. Il permet de créer des interfaces utilisateur dynamiques et des applications single-page (SPA).
- **Avantages pour ce projet** :
 - Facilite la gestion des vues dynamiques.
 - Structure claire pour manipuler les données grâce à son modèle MVVM (Model-View-ViewModel).
 - Support pour les API REST, ce qui est idéal pour interagir avec un backend comme Laravel.

Backend : Laravel

- **Description** : Laravel est un framework PHP qui simplifie le développement backend grâce à son architecture MVC, ses outils intégrés, et son support pour les bases de données relationnelles.
- **Avantages pour ce projet** :
 - Gestion facile des utilisateurs (authentification, rôles, permissions).
 - Intégration avec les bases de données pour stocker les créations des couturiers et les interactions des clients.
 - API REST pour connecter le frontend AngularJS.

Base de données

- **Description** : MySQL est un système de gestion de base de données relationnel très populaire. Il est performant, fiable, et bien adapté pour les applications web.
- **Avantages pour ce projet** :

Compatible nativement avec Laravel grâce à Eloquent ORM.

Permet de gérer efficacement les relations entre les utilisateurs (couturiers et clients) et leurs créations (avec des tables pour les utilisateurs, les créations, les commentaires, etc.).

Facile à déployer et maintenir sur des environnements de production.

Compatibilité

Communication via API REST :

- AngularJS envoie des requêtes HTTP (GET, POST, PUT, DELETE) au backend Laravel.
- Laravel expose des endpoints API pour permettre à AngularJS de récupérer ou d'envoyer des données (par exemple, les créations des couturiers, les commentaires des clients).

JSON comme format d'échange :

- AngularJS consomme des données en JSON, qui est un format nativement supporté par Laravel pour les API REST.
- Cela rend la communication rapide et standardisée.

Sécurité et authentification :

- Laravel offre des fonctionnalités intégrées comme Passport ou Sanctum pour gérer les jetons d'authentification. AngularJS peut utiliser ces jetons pour accéder aux données de manière sécurisée.

Simplicité de déploiement :

- Laravel peut être hébergé sur des serveurs traditionnels (comme Apache ou Nginx).
- AngularJS est une application frontend qui peut être déployée sur un CDN ou sur le même serveur via Laravel.

SÉCURITÉ

Le JWT (JSON Web Token) est un standard ouvert (RFC 7519) qui permet d'échanger des informations de manière sécurisée entre des parties sous forme d'objet JSON

Fonctionnement

1. **Création du Token** : Lorsqu'un utilisateur s'authentifie, le serveur génère un JWT en utilisant les informations de l'utilisateur et le secret.
2. **Transmission du Token** : Le JWT est renvoyé au client, souvent dans un en-tête HTTP (par exemple, [Authorization](#)).
3. **Vérification du Token** : À chaque requête suivante, le client envoie le JWT. Le serveur vérifie la signature pour s'assurer que le token est valide et n'a pas été modifié. Si le token est valide, le serveur peut extraire les informations de l'utilisateur du payload.

Avantages du JWT

- **Stateless** : Les JWT sont auto-contenus, ce qui signifie que le serveur n'a pas besoin de garder une trace des sessions utilisateur.
- **Sécurité** : Les JWT peuvent être signés pour garantir leur intégrité et, dans certains cas, chiffrés pour protéger les données sensibles.
- **Flexibilité** : Les JWT peuvent être utilisés pour l'authentification et l'autorisation dans diverses applications (web, mobile, etc.).

Inconvénients

- **Taille** : Les JWT peuvent être plus volumineux que d'autres méthodes d'authentification.
- **Expiration** : Si un token est compromis, il peut être difficile de révoquer le token avant son expiration.

CALENDRIER DU PROJET

Fonctionnalités Développées

1. Fonctionnalités Utilisateurs

- Authentification et Gestion de Profil
- Exploration et Recherche de Tenues
- Prise de Mesures
- Réservation et Disponibilité
- Panier et Paiement
- Système d'Avis et Commentaires (Notation des stylistes, Commentaires sur les tenues, Système de modération)

2. Fonctionnalités Stylistes

- Gestion de Profil
- Gestion des Produits
- Gestion des Commandes

3. Fonctionnalités Commune

- Notifications

Planification hebdomadaire du travail

Semaine du 16/12/2024 : Phase de conception

Activités Effectuées

- Planification des interfaces Front-End et des modules Back-End.
- Identification des tables nécessaires (Measure, Order, Photo, Product, Review, Stylist... etc).
- Conception initiale des diagrammes UML avec DrawIO.

Semaine du 23/12/2024 : Phase d'implémentation

Au cours de cette semaine, nous avons initialisé et structuré l'arborescence de nos projets Laravel et Angular. Voici les activités principales effectuées et les détails associés :

Angular :

- **Initialisation des Composants :**

- Nous avons créé les composants principaux pour la gestion des pages utilisateur, notamment, nous pouvons :
 - **HomeComponent** : pour la page d'accueil avec les tenues mises en vedette.
 - **ProfileComponent** : pour la gestion des profils utilisateurs et stylistes.
 - **CartComponent** : pour gérer le panier et afficher les articles ajoutés.
- Chaque composant a été conçu en respectant une structure modulaire, facilitant ainsi la maintenabilité du projet.

- **Création des Services :**

- Nous avons établi des services pour centraliser les appels API et la logique applicative :
 - **AuthService** : gère l'authentification, l'inscription et la gestion des tokens utilisateur.
 - **ProductService** : permet de récupérer les produits disponibles avec leurs détails.
 - **NotificationService** : gère la réservation des créneaux et la disponibilité des stylistes.

Laravel :

- **Structure de Projet :**

- Nous avons configuré Laravel en tant que backend principal, avec une structure claire incluant les dossiers **Controllers**, **Models**, et **Services**. Cela garantit une séparation des responsabilités et une meilleure maintenabilité.

- **Création des Modèles :**

- Les modèles suivants ont été conçus pour interagir avec la base de données :
 - **User** : pour la gestion des utilisateurs (clients et stylistes).
 - **Product** : pour les informations sur les tenues.
 - **Order** : pour suivre les commandes passées par les utilisateurs.
 - **Measurement** : pour sauvegarder les mesures des clients.

- **API RESTful :**

- Nous avons développé des endpoints RESTful pour permettre une communication efficace entre le backend et le frontend :
 - `/api/login` : authentification utilisateur.
 - `/api/products` : récupération de la liste des tenues.
 - `/api/bookings` : gestion des réservations.
 - `/api/dashboard` : création et suivi des commandes.

Semaine du 30/12/2024 : Integration des APIs

Au cours de cette semaine, nous nous sommes concentrés sur l'intégration des APIs pour les mesures, les emails et les paiements (Plus de détails dans le code).

- **Backend :**

- Création et configuration des endpoints dans Laravel pour chaque API, en s'assurant qu'ils respectent les normes de performance et de sécurité.
- Intégration de bibliothèques tierces pour gérer des tâches spécifiques comme l'envoi d'emails ou les transactions de paiement.
- Mise en place de validations robustes pour garantir l'intégrité des données.

- **Frontend :**

- Développement de services Angular pour communiquer avec les APIs.
- Création d'interfaces utilisateur intuitives permettant une interaction fluide avec les fonctionnalités de mesures, notifications par email, et paiements.
- Tests pour garantir une bonne synchronisation entre le frontend et le backend.

Semaine du 6/12/2024 : Phase Finale et Tests d'Intégration

Au cours de cette dernière semaine, nous avons effectué un test d'intégration pour évaluer comment les différents modules communiquent et interagissent entre eux. Cette étape était cruciale pour garantir que l'ensemble des fonctionnalités de l'application fonctionne de manière ensemble de façon cohérente.

Travaux Réalisés

- **Vérification des Interfaces :**

- Test des interactions entre le backend (Laravel) et le frontend (Angular).

- Validation de la transmission des données entre les modules API, comme les mesures, les emails, et les paiements.
- **Scénarios de Test :**
 - Simulation de scénarios utilisateur réels, tels que la prise de mesures, la personnalisation d'articles, et le passage de commandes.
 - Test des notifications par email et des mises à jour en temps réel.
 - Vérification des transactions de paiement pour assurer leur bon fonctionnement.

RESSOURCES

Les caractéristiques minimales niveau physique et logistique en fonction des technologies utilisées. Et parles du déploiement sur Vercel

Pour la création de ce site e-commerce, plusieurs ressources ont été mobilisées, tant sur le plan physique que logistique. Dans cette section, nous présentons les différentes ressources nécessaires pour mener à bien ce projet.

Ressources Physiques

Parmi les ressources physiques, l'élément le plus important est l'ordinateur utilisé pour construire et déployer l'application. Nous recommandons les spécifications minimales suivantes :

- **Mémoire RAM** : Au moins 8 Go pour garantir une performance fluide lors de l'exécution des environnements de développement.
- **Processeur** : Multicœur, préférablement avec une architecture 64 bits pour supporter les outils modernes.
- **Stockage** : 256 Go SSD ou plus pour un accès rapide aux fichiers du projet et aux dépendances.

Ressources Logistiques : Technologies et Gestion

Frameworks et Outils Techniques

1. **Angular** : Utilisé pour le frontend, ce framework permet de créer des interfaces interactives et réactives grâce à ses composants modulaires et sa gestion efficace de l'état.
2. **Laravel** : Framework PHP pour le backend, offrant un puissant ORM (Eloquent), un système de migration pour la gestion des bases de données, et des outils pour le développement d'API RESTful.
3. **MySQL** : Base de données relationnelle utilisée pour stocker les informations critiques du site, telles que les utilisateurs, les produits et les commandes.

Outils Supplémentaires

- **Composer** : Gestionnaire de dépendances pour Laravel, essentiel pour installer et maintenir les bibliothèques PHP.
- **Node.js et npm** : Utilisés avec Angular pour gérer les dépendances et

exécuter les scripts de construction du projet frontend.

- **Git** : Outil de contrôle de version pour suivre les modifications du code et permettre la collaboration entre les membres de l'équipe.
- **Draw.io** : Logiciel utilisé pour créer des diagrammes UML. Cela inclut les diagrammes de cas d'utilisation, de classes, et de séquence, facilitant ainsi la planification et la communication autour de la conception du projet.

Déploiement

Vercel : Cette plateforme a été choisie pour déployer le site web. Elle offre un déploiement rapide et efficace des applications web modernes grâce à :

- L'intégration continue (CI/CD).
 - Un CDN intégré pour une distribution rapide des contenus.
 - Le support des domaines personnalisés avec certificats SSL gratuits.
-

LIMITES ET PERSPECTIVE

- Le rafraîchissement des pages a cause des composants et service mals indenté
- Le déploiement de l'API sur hébergeur
- La possibilité de supprimer tous les objets liés à un objet après le destroy
- Avoir un bulle succès ou échec pour chaque action
- Le profil de l'utilisateur
- La redirection automatique sur n'importe page en cas de problème
- Visualiser son propre Calendrier sur son dashboard
- Visualiser les tendances sur son dashboard
- Centraliser les service de mail et paiement par le même corps
- Avoir des profils administrateurs
- Système de recommandation en fonction des prix
- Catégoriser le produit par d'autres tables
- Persistance au niveau des mails
- Persistance au niveau des paiements
- Historique du comportement
- Opportunité d'annuler la destruction de son compte après quelques jours
- Envoyer les mesures via réservation

CONCLUSION

En conclusion, ce projet a permis de prendre en main les technologies Angular et PHP en développant une plateforme web pour les stylistes. Malgré les difficultés rencontrées, nous avons trouvé des solutions existantes et efficaces pour gérer l'authentification, les paiements, le rechargement automatique des pages et le stockage des images. Ce projet a démontré notre capacité à intégrer ces technologies dans un projet concret et nous a permis d'acquérir des compétences précieuses pour nos futurs projets de développement web.