



Université Mohamed V
École Nationale Supérieure d'Informatique
Et d'Analyse des Systèmes
Rabat



Filière : Ingénierie en Data Science and IoT (IDSIT)

RAPPORT DE PROJET DE FIN D'ANNÉE

Conception et Développement d'une application web de gestion de vente de livres d'occasion

Présenté par :

ELMARRHOUB Chaimae
RACHED Jihane

Membres de Jury :

Mme ABNANE Ibtissam
M. RACHAD Taoufik

Encadré par :

M. RACHAD Taoufik
M. ZELLOU Ahmed
Mme MOUMANE Karima

Année Universitaire : 2022-2023

Dédicace

Nous souhaitons commencer par dédier ce travail à toutes les personnes qui nous ont apporté leur soutien, que ce soit de manière concrète ou morale. Votre présence et votre encouragement ont été d'une valeur inestimable tout au long de ce projet.

Nous exprimons également notre profonde reconnaissance à nos familles et à nos amis qui ont été des piliers inconditionnels tout au long de ce projet. Vos encouragements constants, votre soutien indéfectible et votre confiance en nos capacités ont été des moteurs essentiels pour surmonter les défis et persévérer.

Remerciements

Nous souhaitons remercier en premier lieu nos encadrants M. RACHAD Taoufik, M. ZELLOU Ahmed et Mme MOUMANE Karima pour leurs soutiens indéfectibles tout au long de notre projet. Leurs orientations pertinentes et leurs conseils avisés nous ont permis d'acquérir une expérience précieuse et enrichissante qui ont guidé au bon déroulement de notre projet.

Nous remercions aussi toute l'équipe pédagogique de l'ENSIAS pour leur engagement et leur dévouement envers notre réussite académique. Leur expertise et qualité d'enseignement de haut niveau nous a permis non seulement d'approfondir nos connaissances mais aussi à développer notre aspect académique tout comme celui du personnel.

Nos remerciements vont également aux membres du jury Mme ABNANE Ibtissam et M. RACHAD Taoufik qui ont accepté de juger ce travail et de l'enrichir avec leurs remarques pertinentes.

Nous adressons finalement notre sincère gratitude envers toutes les personnes précédemment mentionnées, ainsi que toutes celles qui ont contribué de près ou de le loin à la réalisation de ce projet.

Résumé

Ce projet est témoin du travail réalisé dans le cadre de notre projet de fin d'année. Intitulé : application web de gestion de vente des livres d'occasion, notre projet a pour objectif de développer une application web qui permet de vendre des livres d'occasion avec un bon rapport qualité-prix. Notre plateforme présente une architecture à deux facettes distinctes : une interface administrateur et une interface utilisateur, favorisant ainsi une expérience fluide pour toutes les parties impliquées en facilitant le processus d'achat.

Pour la conception de notre application, nous avons opté pour la méthodologie Merise, qui nous a permis de modéliser efficacement les différentes entités et les relations au sein de notre système. Cette approche a facilité la compréhension globale de notre application et a servi de base solide pour le développement.

Concernant le développement de l'application, nous avons utilisé le framework Laravel qui nous a offert une structure claire et organisée, simplifiant ainsi la mise en œuvre des fonctionnalités de notre application.

Au cours de ce rapport, nous décrirons en détail les étapes de conception et de développement que nous avons suivies, en mettant l'accent sur l'utilisation de la méthodologie Merise pour concevoir notre application en détaillant les entités, les relations et les contraintes et du framework Laravel pour développer l'application, profitant de sa robustesse et de son architecture MVC.

Mots clés : développement, MERISE, framework Laravel, MVC.

Abstract

This project showcases the work done as part of our end-of-year project. Titled "Web Application for Second-hand Book Sales Management," our project aims to develop a web application that facilitates the sale of used books with good value for money. Our platform consists of two distinct facets : an administrator interface and a user interface, ensuring a seamless experience for all parties involved by simplifying the purchasing process.

For the design of our application, we opted for the Merise methodology, which allowed us to efficiently model the different entities and relationships within our system. This approach facilitated a comprehensive understanding of our application and served as a solid foundation for development.

Regarding the development of the application, we utilized the Laravel framework, which provided us with a clear and organized structure, simplifying the implementation of our application's functionalities.

In this report, we will describe in detail the design and development steps we followed, with a focus on the use of the Merise methodology for designing the entities, relationships, and constraints, as well as the utilization of the Laravel framework for application development, benefiting from its robustness and MVC architecture.

Keywords : developpment, MERISE, Laravel Framework, MVC.

Liste des abréviations

MERISE	<i>Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise</i>
SGBD	<i>Système de Gestion de Base de Données</i>
CSS	<i>Cascading Style Sheet</i>
PHP	<i>Hypertext Preprocessor</i>
HTML	<i>Hypertext Markup Language</i>
SQL	<i>Structured Query Language</i>

Table des figures

1.1	Schéma de modèle en V.	5
1.2	Diagramme de Gantt de base.	6
2.1	Le dictionnaire de données.	11
2.2	Le modèle conceptuel de données.	13
2.3	Le modèle conceptuel des traitements.	14
3.1	Le schéma Modèle-Vue-Contrôleur.	18
3.2	Logo « Visual Studio Code »	18
3.3	Logo « PHP »	19
3.4	Logo « HTML5 »	19
3.5	Logo « Css3 »	20
3.6	Logo «JS»	20
3.7	Logo «Bootstrap»	20
3.8	Logo «Xampp»	21
3.9	Logo «MySQL»	21
3.10	Logo «PhpMyAdmin»	22
3.11	Logo framework « Laravel »	22
3.12	Home page	23
3.13	login	24
3.14	registration form	24
3.15	books on sale	25
3.16	user's cart	25
3.17	user's cart	26
3.18	online payment	26
3.19	order details	27
3.20	invoice of the order	27
3.21	all books	28
3.22	add book	28
3.23	all orders	29

Table des matières

Dédicace	I
Remerciements	II
Résumé	III
Abstract	V
Introduction générale	1
Chapitre 1	2
1 Contexte général du projet	2
1.1 Contexte et définition du problème	3
1.2 Cahier de charges	3
1.2.1 Introduction	3
1.2.2 Les besoins fonctionnels	3
1.2.3 Les besoins non fonctionnels	4
1.3 Cycle en V	5
1.4 Diagramme de Gantt	5
1.5 Conclusion	6
Chapitre 3	7
2 Analyse et Conception	7
2.1 Analyse	8
2.1.1 Introduction	8
2.1.2 Etude de l'existant	8
2.1.3 Solution envisagée	8
2.2 Conception : Merise	9
2.2.1 Introduction et définition de base	9
2.2.2 Etapes de réalisation de MERISE	9
2.2.3 Modèle conceptuel de données (MCD)	10

2.2.4	Modèle logique de données (MLD)	13
2.2.5	Modèle conceptuel de traitement (MCT)	14
2.3	Conclusion	15
Chapitre 3		16
3	Réalisation	16
3.1	Architecture MVC	17
3.1.1	Visual Studio code	18
3.1.2	PHP	18
3.1.3	HTML	19
3.1.4	CSS	19
3.1.5	JavaScript	20
3.1.6	Bootstrap	20
3.1.7	Xampp	21
3.1.8	MySQL	21
3.1.9	phpMyAdmin	21
3.1.10	Laravel	22
3.1.11	Interfaces Graphiques	22
3.2	Conclusion	29
Conclusion et perspectives		30
Bibliographie		32

Introduction générale

L'évolution rapide de la technologie a considérablement modifié nos habitudes de consommation et de commerce. De nos jours, de plus en plus de personnes se tournent vers les achats en ligne pour leur praticité, leur diversité et leurs offres attractives. Dans le cadre de notre projet de fin d'année, nous avons entrepris le développement d'une application web dédiée à la vente en ligne de livres d'occasion.

Le marché des livres d'occasion offre de nombreuses opportunités, tant pour les acheteurs à la recherche de bonnes affaires que pour les vendeurs souhaitant donner une deuxième vie à leurs livres, offrant une alternative abordable pour accéder à une vaste sélection d'ouvrages. Notre application a pour but d'une part d'améliorer l'expérience d'achat pour les amateurs des livres et d'une autre part de faciliter le processus de vente, en offrant une plateforme qui favorise les échanges entre les deux parties.

Le rapport se divise en trois chapitres distincts, chacun abordant une étape clé de notre projet. Dans le premier chapitre, nous procéderons à une analyse approfondie des besoins fonctionnels et non fonctionnels de notre application. Le deuxième chapitre mettra l'accent sur l'analyse et la conception de notre application, en tenant compte des résultats de notre analyse des besoins. Enfin, le troisième chapitre se concentrera sur la réalisation concrète de notre projet où nous détaillerons les outils, les technologies et les langages de programmation que nous avons utilisés pour développer notre application.

Chapitre 1

Contexte général du projet

Ce chapitre se concentre sur l'identification des besoins fonctionnels et non fonctionnels de notre application web, afin de définir les attentes et les exigences du projet. Cette étape est essentielle pour orienter le développement de notre application et garantir sa pertinence et sa performance conformément aux attentes des utilisateurs.

1.1 Contexte et définition du problème

Le marché des livres d'occasion connaît une popularité grandissante, car il permet aux lecteurs de trouver des ouvrages à des prix abordables et de donner une seconde vie aux livres déjà lus. Cependant, malgré cette demande croissante, il existe encore des obstacles et des inefficacités dans le processus de vente des livres d'occasion. Notre application web est développée pour répondre à un besoin croissant de faciliter l'achat de livres d'occasion avec un bon rapport qualité-prix.

1.2 Cahier de charges

1.2.1 Introduction

Le cahier de charges joue un rôle essentiel en définissant de manière claire les besoins et attentes du projet, fournissant ainsi une vision précise des objectifs à atteindre.

1.2.2 Les besoins fonctionnels

Les besoins fonctionnels englobent les actions et les fonctionnalités particulières que le système ou le logiciel doit mettre en œuvre. Ils décrivent les tâches et les opérations que le système doit exécuter afin de satisfaire les besoins et les attentes des utilisateurs. Notre projet comporte deux types de fonctionnalités :

Type administrateur :

- Gestion des livres : L'administrateur peut ajouter, modifier ou supprimer des livres d'occasion de la base de données, y compris les détails tels que le titre, l'auteur, l'état et le prix de vente.
- Gestion des commandes : L'administrateur peut accéder aux détails des commandes passées par les utilisateurs, notamment les informations de livraison, les articles commandés et les montants facturés.

Type utilisateur :

- Gestion des utilisateurs : La plateforme devra permettre la création de comptes utilisateurs pour la gestion de leurs informations personnelles et la gestion de leurs commandes.
- Recherche de livres : La plateforme devra permettre aux utilisateurs de rechercher des livres.
- Ajout au panier : Les utilisateurs peuvent ajouter des livres sélectionnés à leur panier pour les acheter ultérieurement.
- Passer commande : Les utilisateurs peuvent passer commande en sélectionnant les livres souhaités dans leur panier.
- Paiement : Les utilisateurs peuvent effectuer le paiement en ligne en utilisant des méthodes de paiement sécurisées, ou choisir l'option de paiement à la livraison.
- Téléchargement de facture : Les utilisateurs peuvent télécharger une facture au format PDF pour chaque commande passée, afin d'avoir une preuve d'achat.

1.2.3 Les besoins non fonctionnels

Les besoins non fonctionnels précisent le comportement attendu du système ou les contraintes auxquelles il doit se conformer. Ils sont essentiels pour garantir que le système ou le logiciel répond aux exigences de qualité et aux attentes des utilisateurs :

- Simplicité de la plateforme : La plateforme doit être facile à utiliser, offrant une interface intuitive et compréhensible pour les utilisateurs, afin de faciliter leur navigation et leurs interactions avec le système.
- La disponibilité : La plateforme doit être accessible à tous les utilisateurs, leur permettant ainsi de l'utiliser sans aucune restriction.
- Confidentialité des données : limitation de l'accès aux informations personnelles des utilisateurs.

- Gestion de panier : La plateforme doit assurer la stabilité et la fiabilité du processus de gestion de panier, en évitant les erreurs ou les pertes de données lors de l'ajout, la suppression ou la modification des articles dans le panier.

1.3 Cycle en V

Pour la gestion du cycle de développement de notre logiciel, nous avons adopté la méthode en V. Ce cycle représente visuellement la séquence des étapes de développement et de test, mettant l'accent sur la validation et la vérification des différentes phases du projet.

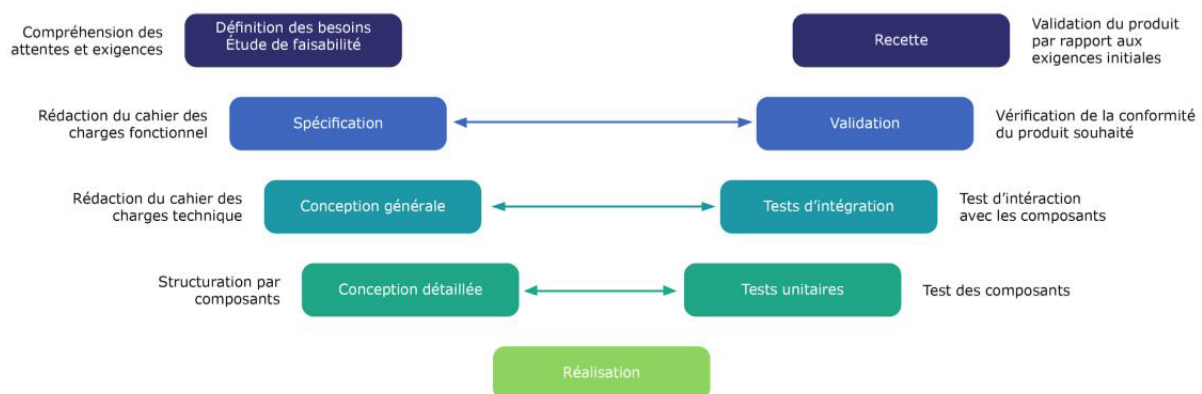


FIG. 1.1 : Schéma de modèle en V.

1.4 Diagramme de Gantt

Ce diagramme permet la planification et la coordination des différentes étapes du projet. C'est un outil de gestion de projet utilisé pour visualiser les différentes tâches et activités d'un projet sur une ligne de temps. Le diagramme présenté dans la page qui suit représente les différentes étapes que nous avons suivies dans notre projet :

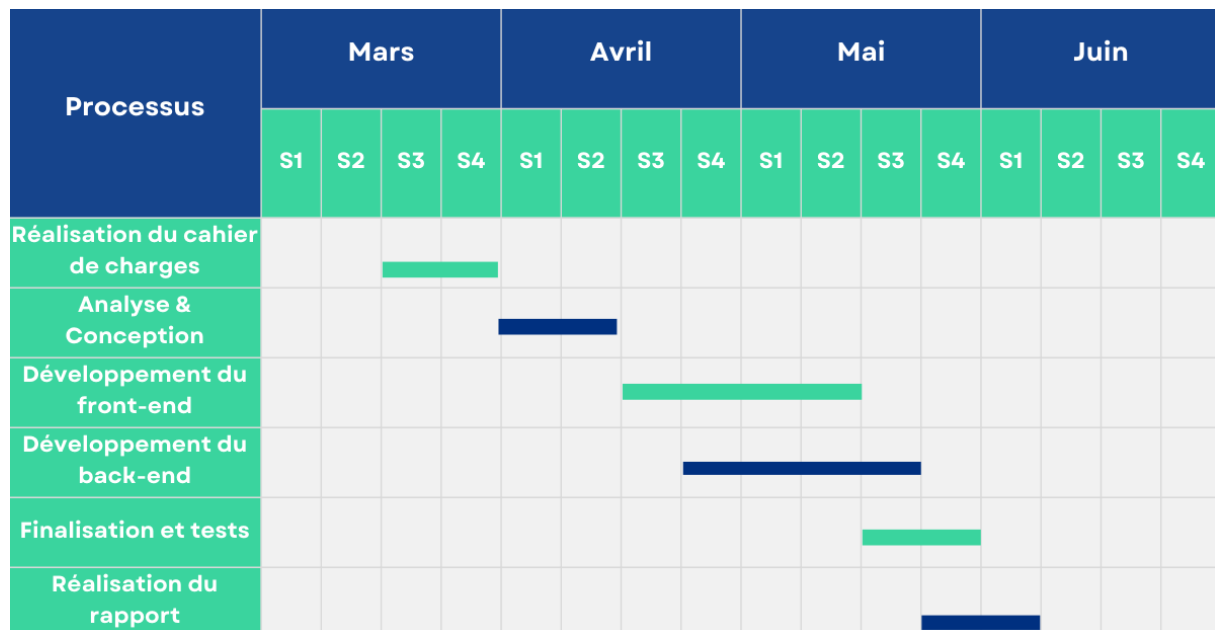


FIG. 1.2 : Diagramme de Gantt de base.

1.5 Conclusion

Ce chapitre a permis d'établir le contexte général de notre projet en identifiant le besoin principal qui a conduit au développement de notre application web. Nous avons présenté en détail le cahier des charges, en mettant en évidence les besoins fonctionnels et non fonctionnels de notre application. De plus, nous avons exposé la méthodologie en V que nous avons adoptée pour gérer le cycle de développement du logiciel, ainsi que le diagramme de Gantt qui a permis de planifier et de suivre les différentes étapes du projet. Ces approches ont été essentielles pour garantir la cohérence, la qualité et la gestion efficace de notre projet.

Chapitre 2

Analyse et Conception

Ce chapitre constitue une étape clé dans notre projet, où nous nous penchons spécifiquement sur l'analyse et la conception des systèmes d'information en définissant une architecture solide et cohérente pour notre système.

2.1 Analyse

2.1.1 Introduction

En considérant le contexte général du projet, marqué par une demande croissante de vente de livres d'occasion et les inefficacités actuelles du processus, il est essentiel de mener une analyse approfondie pour identifier les fonctionnalités, les flux de données, les interactions et les contraintes qui façonneront la conception et le développement de notre application.

2.1.2 Etude de l'existant

Dans notre étude de l'existant, nous avons constaté un manque de plateformes marocaines dédiées à la vente de livres d'occasion offrant un bon rapport qualité-prix, ainsi qu'une possibilité de livraison pour assurer le confort du client. Les plateformes existantes ne répondent pas pleinement à ces besoins spécifiques du marché marocain.

En effet, la plupart des plateformes disponibles se concentrent principalement sur la vente de livres neufs, laissant un vide dans le domaine de la vente de livres d'occasion. Cela crée une opportunité pour notre projet d'application web de combler cette lacune en offrant une plateforme dédiée à la vente de livres d'occasion au Maroc.

2.1.3 Solution envisagée

Après une étude approfondie du marché, nous avons identifié deux solutions principales pour répondre à la problématique :

Création d'une plateforme dédiée : par la conception d'une application web spécifiquement dédiée à la vente de livres d'occasion. Cette plateforme permettra aux utilisateurs de parcourir facilement les livres disponibles et de passer des commandes selon leurs préférences.

Gestion optimisée des ventes : par le développement d'un tableau de bord pour l'espace vendeur, afin de permettre le suivi des ventes et la gestion efficace des activités. Ce tableau de bord fournira au vendeur des informations sur les livres disponibles en stock, ainsi que les détails des commandes passées par les utilisateurs. Cela facilitera l'organisation des

ventes afin d'optimiser les performances des vendeurs

2.2 Conception : Merise

2.2.1 Introduction et définition de base

MERISE est une méthodologie de conception et de modélisation des systèmes d'information. Son objectif est de fournir un cadre structuré et méthodique pour analyser, concevoir et mettre en œuvre des systèmes d'information efficaces et cohérents. En d'autres termes, MERISE permet de modéliser les différentes composantes d'un système, telles que les données, les processus métier, les acteurs et les interactions.

2.2.2 Etapes de réalisation de MERISE

La méthodologie MERISE se décompose généralement en trois niveaux de modélisation : le niveau conceptuel, le niveau logique et le niveau physique. Chaque niveau se concentre sur des aspects spécifiques de la conception et de l'implémentation d'un système d'information.

- **Le niveau conceptuel** : Le niveau conceptuel est axé sur la compréhension des besoins des utilisateurs et la représentation des concepts clés du système. À ce stade, on identifie les entités principales du système, c'est-à-dire les objets concrets ou abstraits qui interagissent avec le système. On établit également les relations entre ces entités pour capturer les liens et les dépendances entre elles. Le résultat de cette étape est généralement représenté par un modèle conceptuel des données (MCD), qui illustre les entités, les attributs et les associations.
- **Le niveau logique** : Une fois le niveau conceptuel établi, on passe au niveau logique qui se concentre sur la conception de la structure des données et des processus métier. On définit les tables de base de données, les contraintes d'intégrité et les règles de gestion des données. On analyse également les processus métier pour identifier les tâches, les actions et les règles de traitement. Cette étape conduit généralement

à la création d'un modèle logique des données (MLD) qui spécifie les tables, les relations, les clés primaires et étrangères, ainsi que les règles de normalisation.

- Le niveau physique : Une fois le niveau logique défini, on passe au niveau physique qui concerne la mise en œuvre technique du système. À ce stade, on se penche sur les choix de technologies, les architectures matérielles et logicielles, et les contraintes de déploiement. On détermine les détails de l'implémentation, tels que les types de bases de données, les langages de programmation, les frameworks et les outils nécessaires. Le résultat de cette étape est généralement un modèle physique des données (MPD), qui décrit la structure et la configuration techniques de la base de données.

2.2.3 Modèle conceptuel de données (MCD)

Ce modèle représente les entités, les attributs et les relations entre les données d'un système d'information. Il s'agit d'une représentation abstraite et indépendante des détails techniques de la structure des données. La réalisation du modèle conceptuel des données (MCD) comporte plusieurs étapes que nous détaillerons dans les points suivants.

- Dictionnaire de données C'est une description centrale de la structure des données stockées dans la base de données. Il est utilisé à titre informationnel par les modules du SGBD pour savoir où et comment sont stockées les informations.

Nom	Signification	Type	Longueur	Nature
book_id	L'identifiant du livre	Entier	20	E
titre_livre	Titre du livre	Chaine de caractères	100	Concaténée
nom_auteur	Nom de l'auteur du livre	Chaine de caractères	100	E
prenom_auteur	Prénom de l'auteur du livre	Chaine de caractères	100	E
image_livre	Image du livre	Chaine de caractères	255	E
prix_livre	Prix du livre	Réel	50	E
qte_stock_livre	Quantité disponible dans le stock	Entier	6	E
etat_livre	Etat sur lequel se trouve le livre	Enum : ('comme neuf', 'bon', 'acceptable')	-	Concaténée
Commande_id	L'identifiant de la commande	Entier	20	E
com_passe	Détermine si la commande est passée	Entier : 0 ou 1	1	E
que	Quantité des livres commandés	Entier	11	E
user_id	L'identifiant de l'utilisateur	Entier	20	E
prenom	Prénom de l'utilisateur	Chaine de caractères	255	E
nom	Nom de l'utilisateur	Chaine de caractères	255	E
email	Email de l'utilisateur	Chaine de caractères	255	E
isAdmin	Détermine si le compte est celui d'un admin ou d'un utilisateur	Entier : 0 ou 1	1	E
adresse	Adresse de l'utilisateur	Chaine de caractères	255	Concaténée
telephone	Numéro de téléphone de l'utilisateur	Chaine de caractères	50	E

FIG. 2.1 : Le dictionnaire de données.

- Dépendances fonctionnelles :

Les dépendances fonctionnelles sont une représentation des relations et des contraintes entre les attributs d'une table dans une base de données. Elles décrivent comment la valeur d'un attribut ou d'un groupe d'attributs détermine de manière unique la valeur d'un autre attribut ou d'un groupe d'attributs dans la même table.

Pour notre système les dépendances fonctionnelles sont comme suit :

- L'identifiant de la table books permet de déterminer le titre du livre, nom et prénom de l'auteur, le prix et l'état du livre.

- L'identifiant de la table commande permet de déterminer l'identifiant et la quantité commandée pour chaque livre.

- L'identifiant de la table users permet de déterminer le nom, prénom, email, adresse et le numéro de téléphone de l'utilisateur.

- Règles de gestion :

Les règles de gestion du MCD précisent les contraintes d'intégrité qui doivent être respectées par le modèle.

Pour notre système, les règles de gestion ont été définies de la manière suivante :

-Chaque client est enregistré dans le système avec des informations telles que son nom, son adresse, son numéro de téléphone, son email etc.

-Chaque client peut passer une ou plusieurs commandes.

-Chaque panier est associé à un et un seul client.

-Chaque utilisateur a son propre compte (Email+Mot de passe).

-Chaque commande peut contenir un ou plusieurs livres.

-La quantité de stock varie au fur et à mesure des commandes.

- **Elaboration du Modèle conceptuel de données (MCD)**

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information à l'aide d'entités et d'associations.

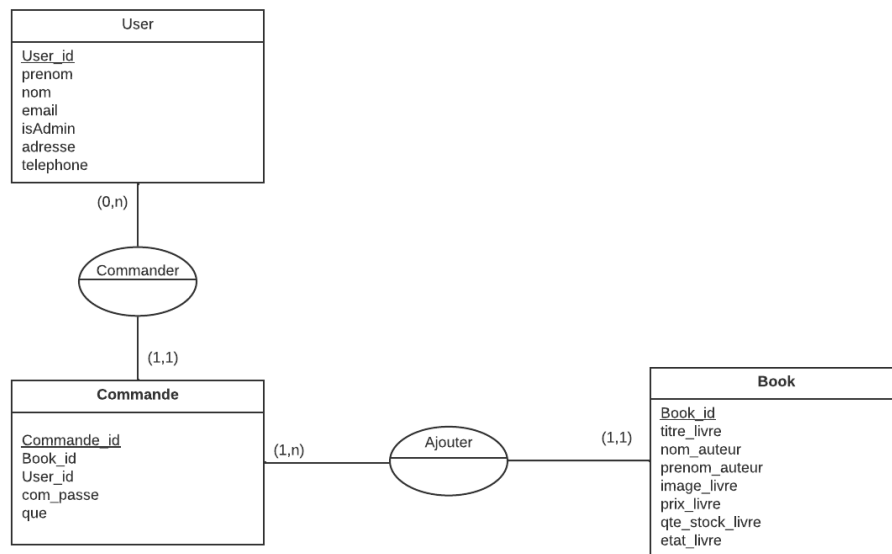


FIG. 2.2 : Le modèle conceptuel de données.

2.2.4 Modèle logique de données (MLD)

Le modèle logique de données permet de modéliser la structure selon laquelle les données seront stockées dans la future base de données. Il est adapté à une famille de SGBD et permet ainsi d'implémenter la base de données dans un SGBD donné.

- **User**(User_id, prenom, nom, email, isAdmin, adresse, telephone)
- **Book**(Book_id, titre_livre, nom_auteur, prenom_auteur, image_livre, prix_livre, qte_stock_livre, etat_livre)
- **Commande**(Commande_id, #Book_id, #User_id, com_passe, que)

2.2.5 Modèle conceptuel de traitement (MCT)

MCT (Modèle Conceptuel des Traitements) est un composant de la méthodologie MERISE. Il se concentre sur l'analyse et la conception des processus et des activités du système. Le MCT représente les aspects de traitement du système d'information et décrit la circulation des données entre les différents processus. Cette représentation visuelle facilite la compréhension des différentes étapes et transformations des données au sein du système. En utilisant le MCT, on peut identifier et spécifier les fonctionnalités et les opérations qui doivent être effectuées dans un système d'information.

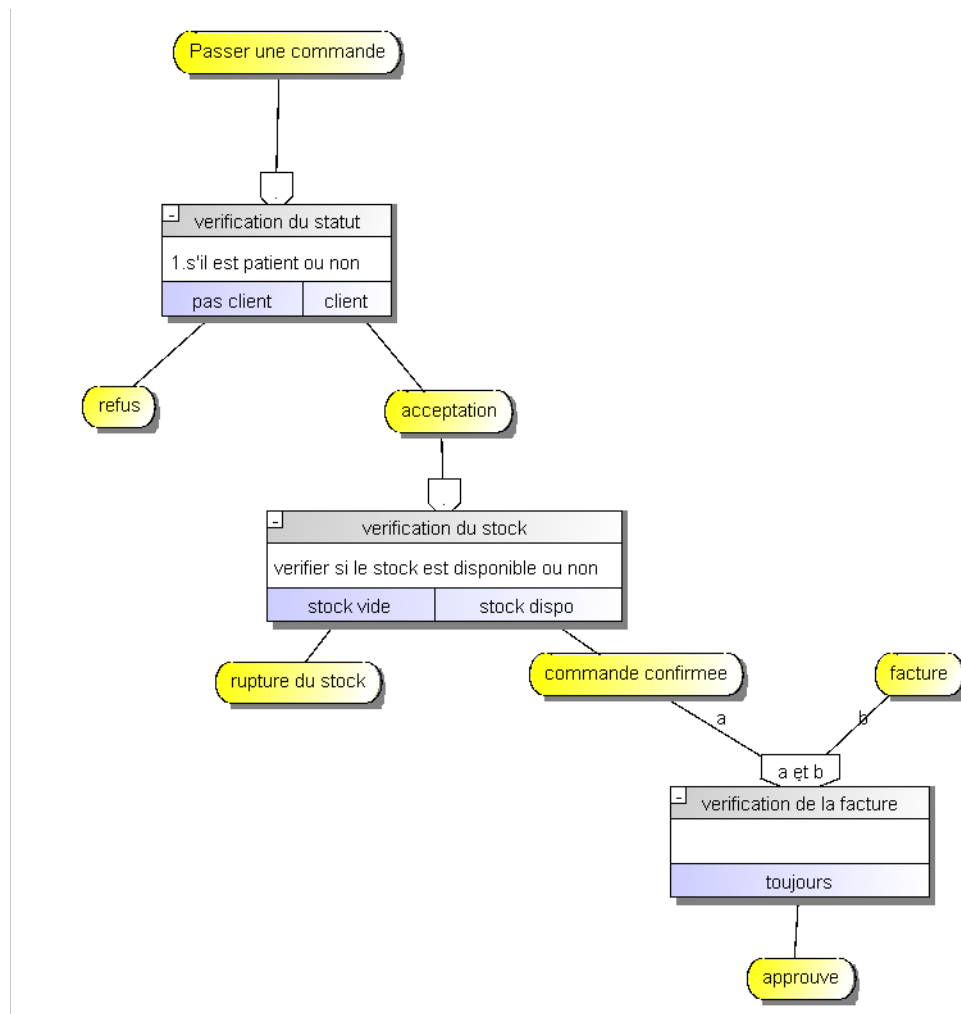


FIG. 2.3 : Le modèle conceptuel des traitements.

2.3 Conclusion

Dans ce chapitre nous avons abordé la modélisation des données et des traitements en utilisant la méthode Merise (modèle conceptuel des traitements, modèle conceptuel de données, modèle logique des données) qui permettent de donner une vision bien détaillée sur le fonctionnement du système. Maintenant nous allons entamer la phase de réalisation de notre application.

Chapitre 3

Réalisation

Ce chapitre est dédié à l’environnement logiciel du projet. Nous commencerons par présenter les différents composants logiciels utilisés dans le cadre de notre projet. Ensuite, nous nous concentrerons sur la description des interfaces du système, en mettant en évidence quelques scénarios d’utilisation spécifiques.

3.1 Architecture MVC

Afin de clarifier l'architecture de notre application et de simplifier la tâche du développement nous avons choisi le patron MVC (Modèle-Vue-Contrôleur) qui est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Ce patron regroupe les fonctions nécessaires en trois catégories :

- **Modèle :**

Un modèle contient les données utilisées par un programme. Il peut s'agir d'une base de données, d'un fichier ou d'un simple objet. Par exemple, un objet Client récupérera les informations de la base de données, les manipulera et mettra à jour ses données dans la base de données.

- **Vue :**

C'est avec quoi l'utilisateur interagit et se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, le fait de cocher une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

- **Contrôleur :**

Les contrôleurs agissent comme une interface entre le modèle et la vue, pour traiter toute la logique métier et les requêtes entrantes, manipuler les données à l'aide du composant Modèle et interagir avec les Vues pour rendre le résultat final. Par exemple, le contrôleur « Client » va traiter toutes les interactions et les entrées de la Vue « Client » et mettre à jour la base de données en utilisant le Modèle « Client ». Le même contrôleur sera utilisé pour visualiser les données du client.

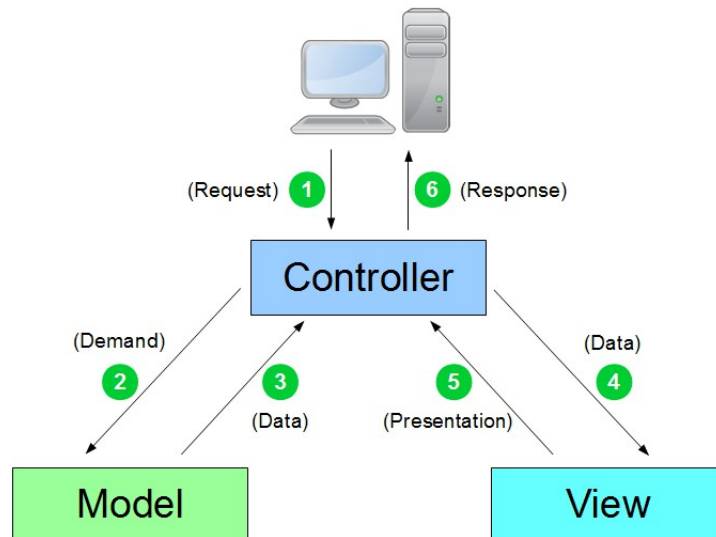


FIG. 3.1 : Le schéma Modèle-Vue-Contrôleur.

3.1.1 Visual Studio code

Visual Studio Code (VS code) est un éditeur de code source léger et extensible, développé par Microsoft. Dans notre projet, VS Code a été notre éditeur pour écrire, exécuter et gérer notre code source. Nous avons également utilisé les extensions de VS Code pour ajouter des fonctionnalités spécifiques à nos besoins.

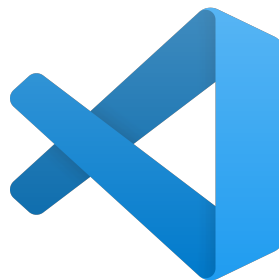


FIG. 3.2 : Logo « Visual Studio Code »

3.1.2 PHP

Nous avons utilisé PHP (HyperText Préprocesseur) en raison de la structure solide qu'il offre. PHP, combiné avec le framework Laravel, nous a permis de suivre une architecture MVC claire et organisée pour notre

application. Nous avons pu séparer la logique métier (contrôleurs), la présentation des données (vues) et la gestion des données (modèles) de manière efficace. Cette approche nous a permis d'avoir un code structuré, maintenable et évolutif tout au long du développement de notre application.



FIG. 3.3 : Logo « PHP »

3.1.3 HTML

HyperText Markup Language est un langage de balisage utilisé pour structurer le contenu et la présentation visuelle des éléments d'une page Web.

Dans le cadre de notre projet , HTML a été utilisé pour structurer le contenu de l'application.



FIG. 3.4 : Logo « HTML5 »

3.1.4 CSS

Cascading Style Sheets est un langage informatique permettant de mettre en forme des pages web, Il permet de contrôler l'apparence visuelle des éléments d'une page Web.

CSS a été utilisé pour le styliser.



FIG. 3.5 : Logo « Css3 »

3.1.5 JavaScript

JavaScript est un langage de programmation de haut niveau. Il utilise des scripts pour créer un contenu dynamique permettant de manipuler, modifier et interagir avec les éléments HTML d'une page web.

JavaScript a été utilisé pour manipuler les données côté client et améliorer l'expérience utilisateur



FIG. 3.6 : Logo « JS »

3.1.6 Bootstrap

Nous avons également utilisé Bootstrap, un framework CSS, pour développer l'interface utilisateur et admin de l'application. En exploitant les composants prêts à l'emploi et le système de grille de Bootstrap, nous avons pu créer une interface responsive et captivante.



FIG. 3.7 : Logo « Bootstrap »

3.1.7 Xampp

Pour exécuter l'application localement, nous avons utilisé XAMPP, qui est un ensemble d'outils de développement web comprenant un serveur web Apache, un système de gestion de bases de données MySQL et un interpréteur PHP. XAMPP nous a permis de créer un environnement de développement complet sur notre machine.



FIG. 3.8 : Logo « Xampp »

3.1.8 MySQL

Pour la gestion de la base de données, nous avons utilisé MySQL, un système de gestion de bases de données relationnelles. Nous avons créé des tables pour stocker les informations des utilisateurs, les livres mis en vente et les commandes passées. MySQL nous a permis d'interagir avec la base de données, d'effectuer des requêtes et de stocker les données de manière sécurisée.



FIG. 3.9 : Logo « MySQL »

3.1.9 phpMyAdmin

phpMyAdmin est une interface web open source qui permet de gérer et d'administrer des bases de données MySQL de manière conviviale. Il simplifie les tâches d'administration, offre une interface visuelle conviviale et permet d'effectuer rapidement des opérations courantes sur les bases de données.



FIG. 3.10 : Logo « PhpMyAdmin »

3.1.10 Laravel

Nous avons choisi d'utiliser le framework Laravel pour développer l'application en raison de sa structure MVC qui facilite l'organisation du code et la séparation des préoccupations. Grâce à Laravel, nous avons pu mettre en place rapidement les fonctionnalités essentielles de l'application telles que la gestion du stock des livres et des commandes.



FIG. 3.11 : Logo framework « Laravel »

3.1.11 Interfaces Graphiques

3.1.11.1 Interfaces de l'espace Client

La page d'accueil de l'application est comme suit :

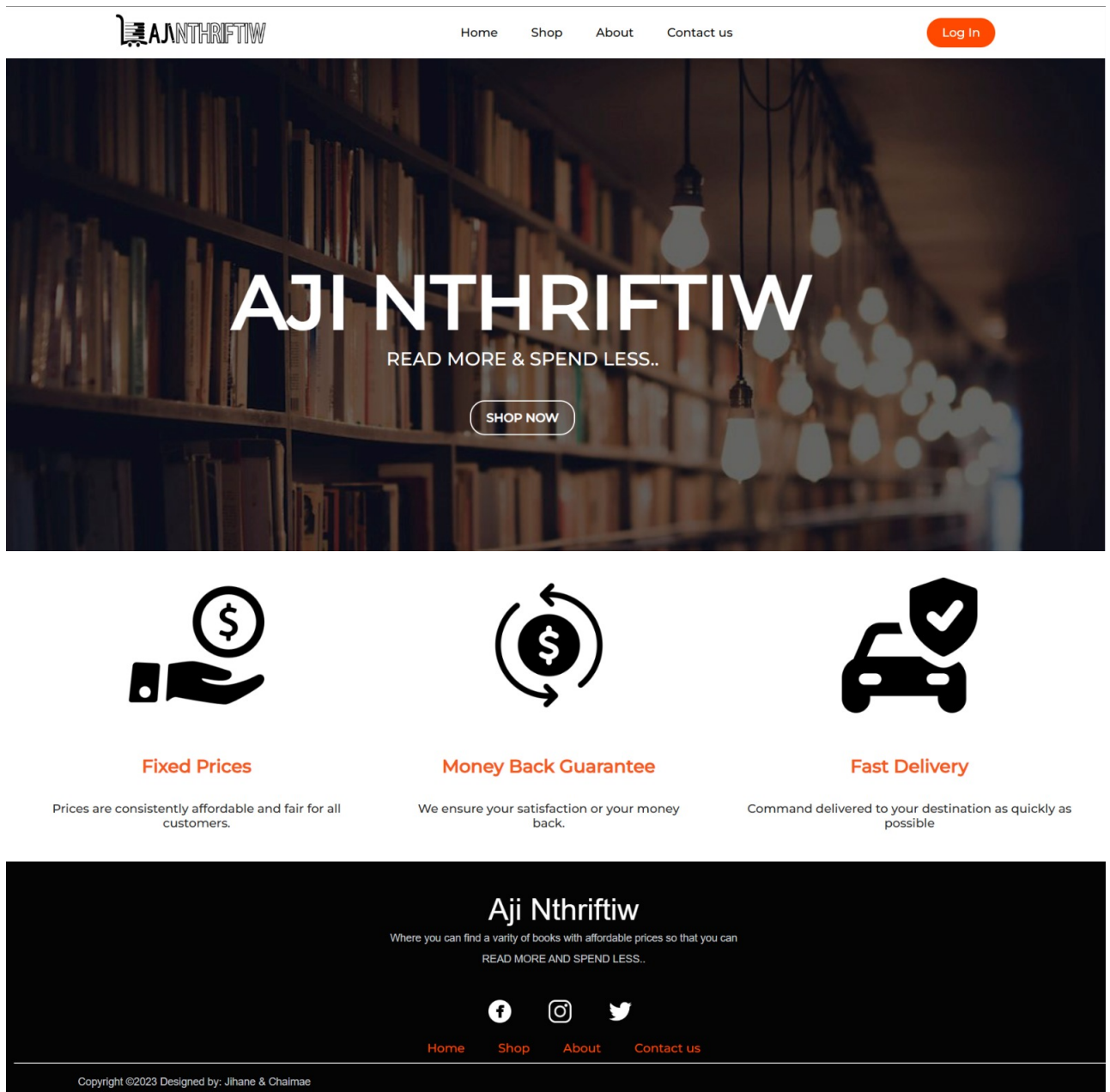
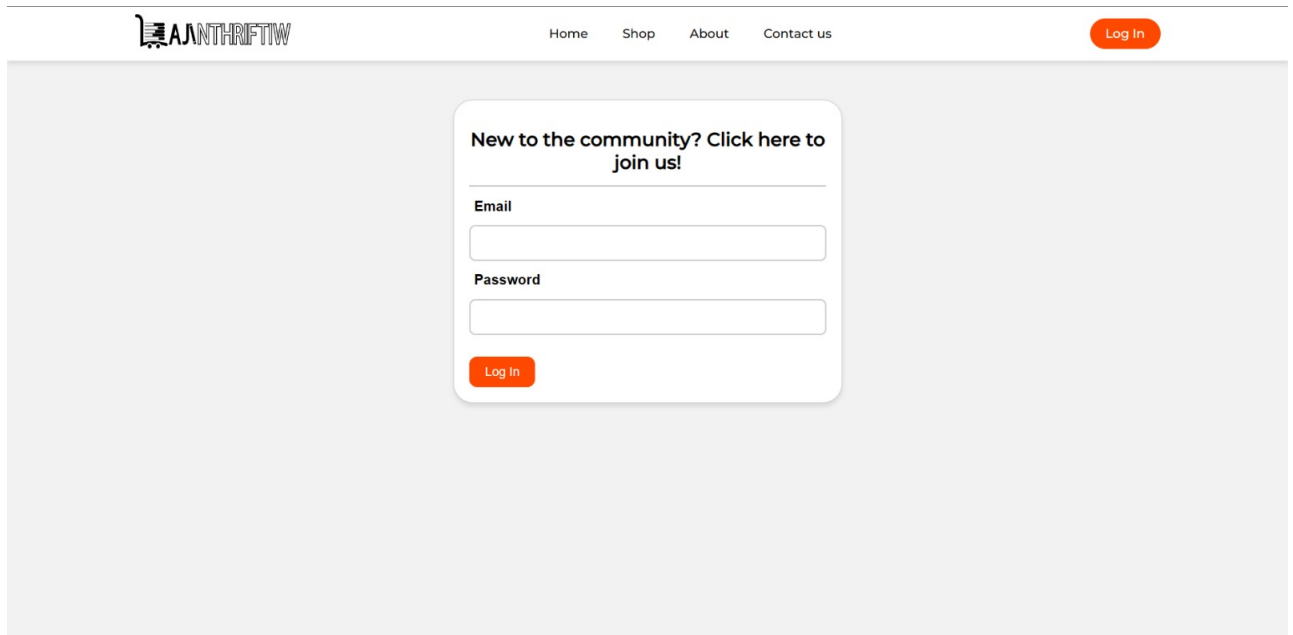


FIG. 3.12 : Home page

Page d'authentification

Pour se connecter, l'utilisateur fait entrer ses informations :

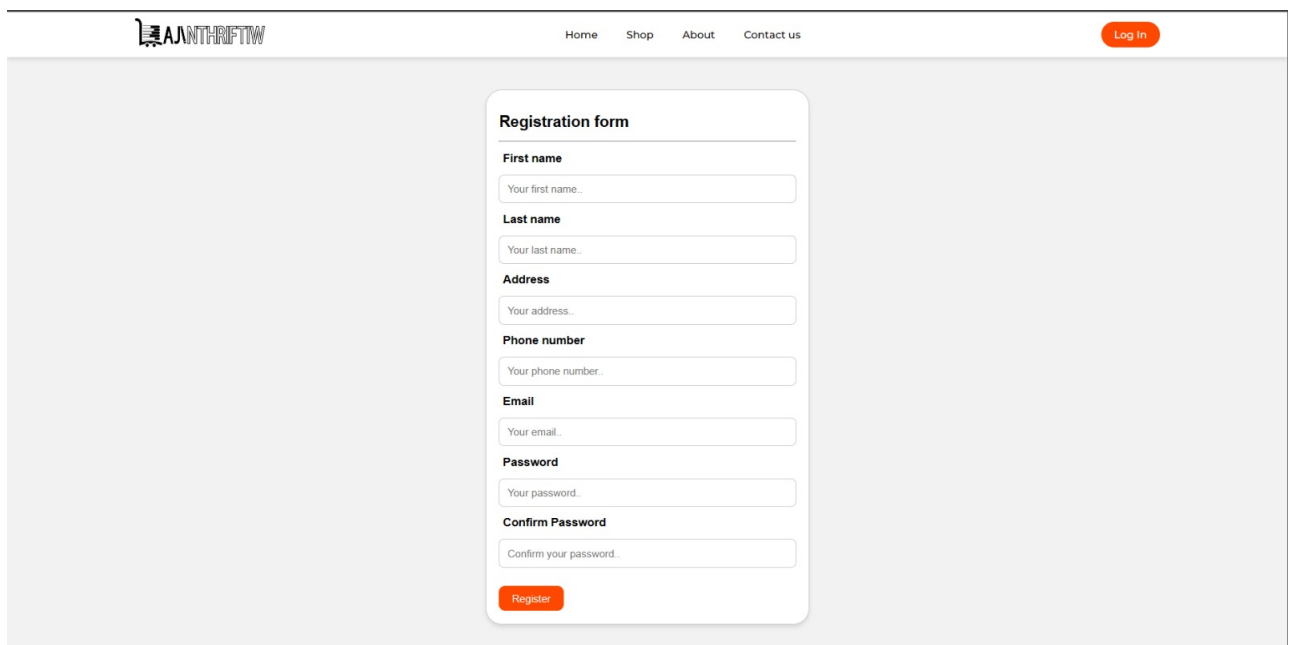


The screenshot shows a web page for 'ANTHRIFTW'. The header includes a logo, navigation links (Home, Shop, About, Contact us), and a 'Log In' button. The main content area features a login form with the text 'New to the community? Click here to join us!'. The form contains two input fields labeled 'Email' and 'Password', and a 'Log In' button.

FIG. 3.13 : login

Formulaire d'inscription

L'utilisateur peut créer son propre compte à travers ce formulaire :



The screenshot shows a web page for 'ANTHRIFTW'. The header includes a logo, navigation links (Home, Shop, About, Contact us), and a 'Log In' button. The main content area features a registration form titled 'Registration form'. The form contains several input fields: 'First name', 'Last name', 'Address', 'Phone number', 'Email', 'Password', and 'Confirm Password'. A 'Register' button is located at the bottom of the form.

FIG. 3.14 : registration form

Page de présentation des livres

L'utilisateur peut trouver les livres mis en vente dans cette page et ajouter

ensuite les articles souhaités dans son panier :

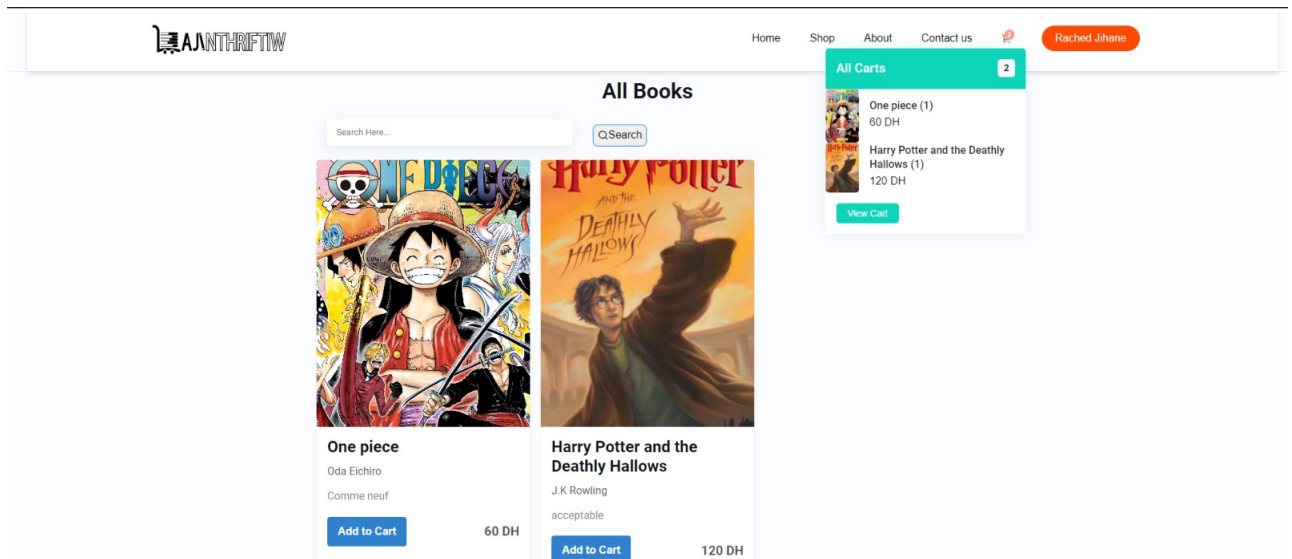


FIG. 3.15 : books on sale

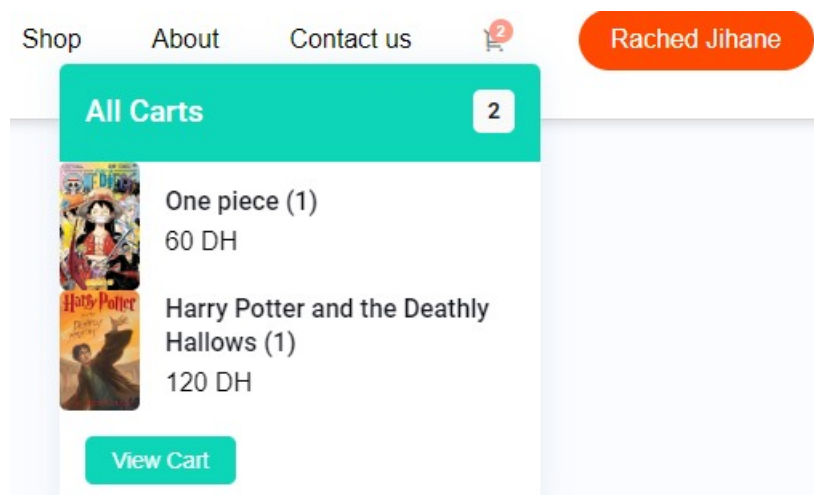


FIG. 3.16 : user's cart

En cliquant sur view cart, le détail du panier s'affiche.

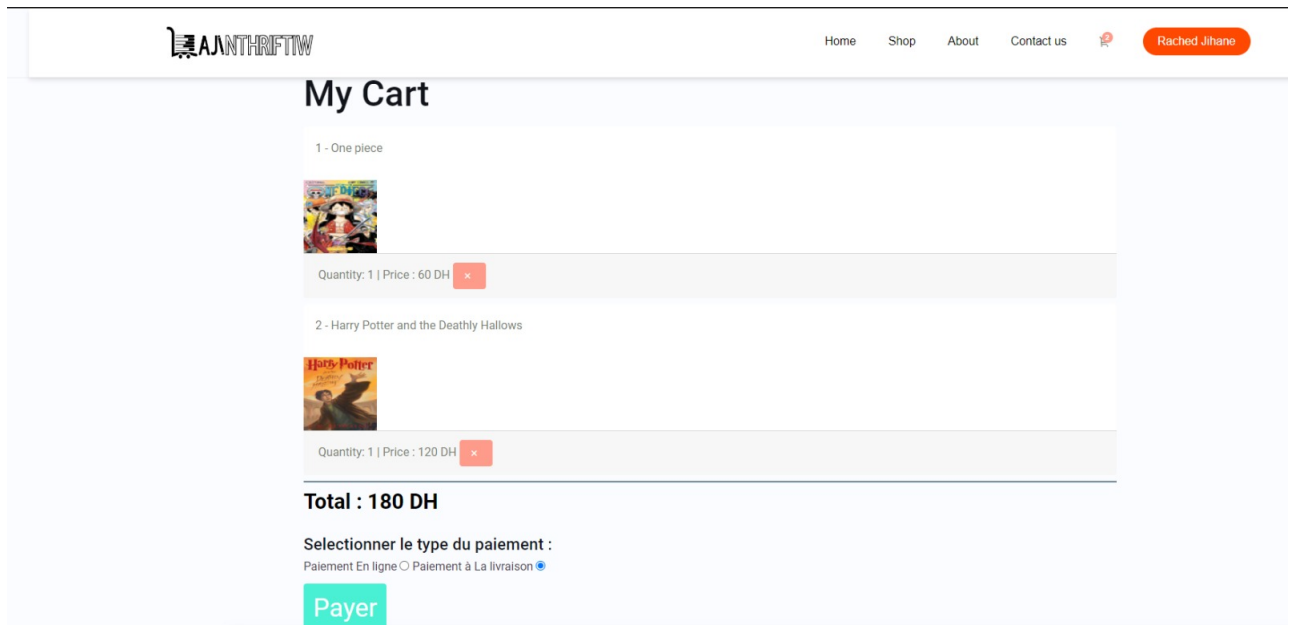


FIG. 3.17 : user's cart

L'utilisateur a le choix entre les deux options de paiements :
paiement en ligne :

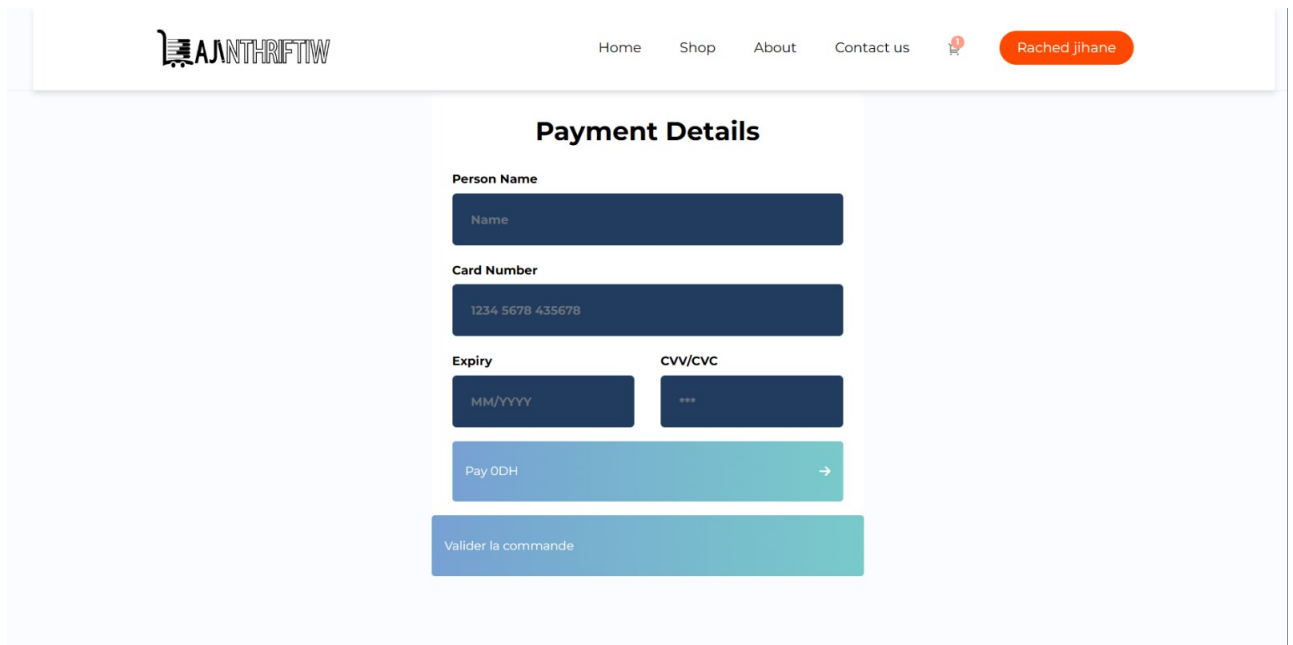


FIG. 3.18 : online payment

paiement à la livraison :
En cliquant sur cette option, l'utilisateur se redirige vers la page suivante :

My Books
One piece : 1 X 60 DH
Harry Potter and the Deathly Hallows : 1 X 120 DH
TOTAL : 180 DH
Payment a la livraison
Nom :
Rached Jihane
Telephone :
0687888546
Adresse du paiement :
Adresse 11, Num 5, Rue 32 , Casablanca

Valider la commande

FIG. 3.19 : order details

Affichage du reçu de la commande en format PDF

Après validation de la commande, un reçu de la commande passée en format PDF s'affiche comme suit :

Jihane Rached, email:
Jihane.Rached@gmail.com, Adresse
11, Num 5, Rue 32 , Casablanca,
0687888546
This is the receipt for a payment of **180 DH** you made to books.

Payment No.
#88305
Payment Date
2023-06-06 00:59:29

DESCRIPTION	AMOUNT
One piece	60 DH
Harry Potter and the Deathly Hallows	120 DH

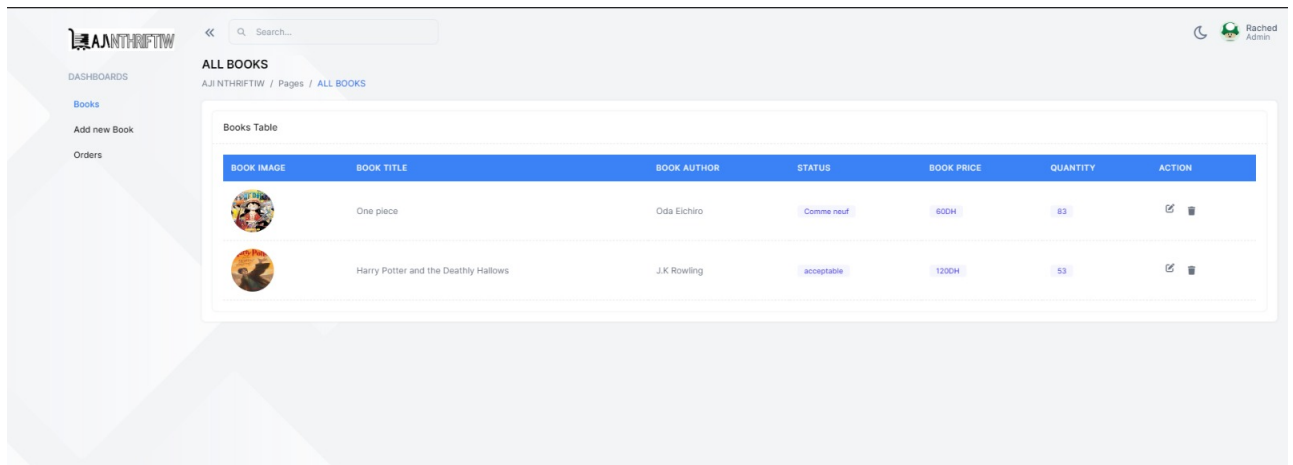
total:
180 DH

FIG. 3.20 : invoice of the order

3.1.11.2 Interfaces de l'espace Admin

Page d'affichage des livres

L'admin peut consulter la liste des livres et leurs disponibilités dans le stock. Il peut également modifier ou supprimer un livre à travers les boutons dans la colonne "action" :



ALL BOOKS

AJI NTHRIFTW / Pages / ALL BOOKS

Books Table



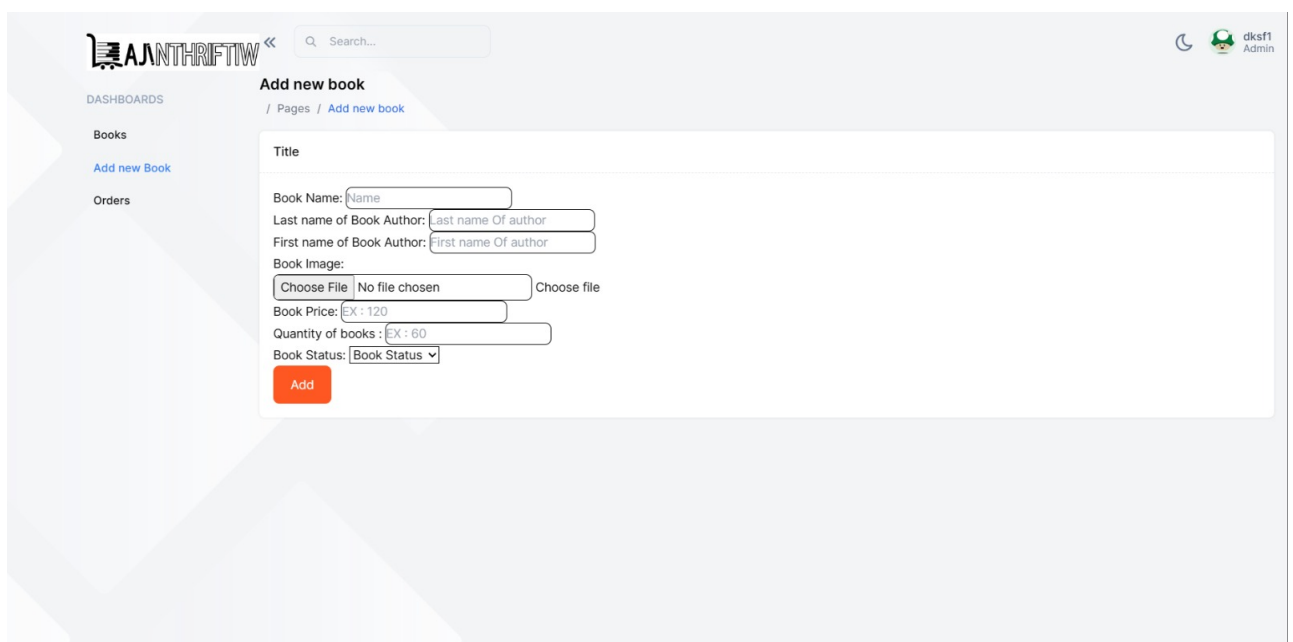
BOOK IMAGE	BOOK TITLE	BOOK AUTHOR	STATUS	BOOK PRICE	QUANTITY	ACTION
	One piece	Oda Eiichiro	Comme neuf	600DH	83	Edit Delete
	Harry Potter and the Deathly Hallows	J.K Rowling	acceptable	1200DH	53	Edit Delete

FIG. 3.21 : all books

Page d'ajout des livres

L'admin peut ajouter des livres à travers ce formulaire en cliquant sur le bouton Add new Book du sidebar :



ADD NEW BOOK

AJI NTHRIFTW / Pages / Add new book

Title

Book Name:

Last name of Book Author:

First name of Book Author:

Book Image: No file chosen

Book Price:

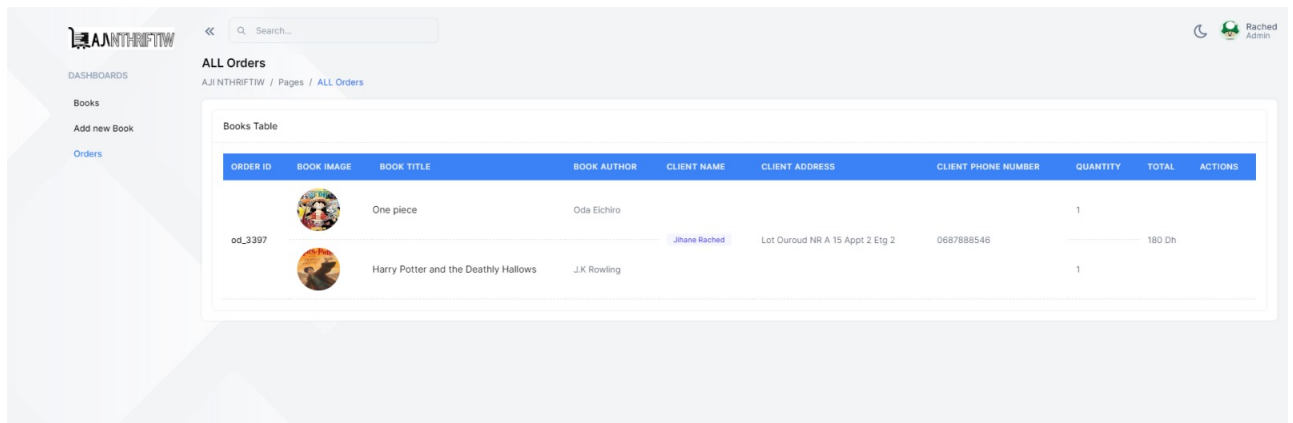
Quantity of books:

Book Status:

FIG. 3.22 : add book

Page d'affichage des commandes

L'admin peut consulter la liste des commandes passées par les utilisateurs en cliquant sur le bouton Orders du sidebar :



ALL Orders

AJITHRIFTW / Pages / ALL Orders

Books Table



ORDER ID	BOOK IMAGE	BOOK TITLE	BOOK AUTHOR	CLIENT NAME	CLIENT ADDRESS	CLIENT PHONE NUMBER	QUANTITY	TOTAL	ACTIONS
od_3397		One piece	Oda Eichiro	Jihane Rached	Lot Ouroud NR A 15 Appt 2 Etg 2	0687888546	1	180 Dh	
		Harry Potter and the Deathly Hallows	J.K. Rowling	Jihane Rached	Lot Ouroud NR A 15 Appt 2 Etg 2	0687888546	1		

FIG. 3.23 : all orders

3.2 Conclusion

Dans ce chapitre nous avons montré et détaillé l'architecture et les technologies utilisées pour la réalisation de notre projet, en présentant ainsi les principales interfaces qui permettent de répondre aux besoins de l'application.

Conclusion et perspectives

Conclusion générale

Tout au long de ce rapport, nous avons détaillé la démarche que nous avons suivie pour développer cette application, dans le cadre de notre projet de fin d'année à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS). Notre objectif était de créer une application web de gestion de vente de livres d'occasion.

Nous avons commencé par présenter le sujet et le cahier des charges, afin de définir les attentes et les spécifications de l'application. Ensuite, nous avons entamé la phase d'analyse et conception, où nous avons examiné les besoins des utilisateurs, les contraintes techniques et les solutions envisagées afin d'élaborer ensuite les différentes étapes et fonctionnalités de l'application.

Enfin, nous avons procédé à la réalisation finale de l'application. Cela a impliqué le développement des fonctionnalités, la mise en place de l'interface utilisateur et l'interface admin en parallèle avec les tests pour garantir un bon fonctionnement. Tout au long de ce processus, nous avons mis à profit nos compétences en développement web, en gestion et en analyse.

Ce projet nous a permis non seulement de consolider nos acquis informatiques, mais aussi d'élargir nos compétences et de comprendre les attentes du marché pour un ingénieur en développement web. De plus, en travaillant en binôme et en favorisant l'entraide entre les étudiants de la filière IDSIT, nous avons également pu amélioré nos soft skills.

En somme, cette expérience a été enrichissante tant sur le plan technique que sur le plan personnel, et nous sommes fiers d'avoir pu concrétiser ce projet en créant une application web fonctionnelle pour la gestion de vente de livres d'occasion.

Bibliographie

- [1] Cours Mme Naoual Chaouni Benabdellah - MODÉLISATION ET SI.
- [2] Cours M. Ahmed ZELLOU - Développement des Applications Web Framework Laravel.
- [3] [https ://laravel.com/docs/10.x](https://laravel.com/docs/10.x)
- [4] [https ://openclassrooms.com/fr/](https://openclassrooms.com/fr/)
- [5] [https ://www.overleaf.com/](https://www.overleaf.com/)
- [6] [https ://getbootstrap.com/](https://getbootstrap.com/)