# DEDICACES

#### **DEDICACES**

À Dieu Tout-Puissant, source de tout savoir et de toute intelligence, je rends grâce pour la sagesse, la force et l'inspiration qu'Il m'a accordées tout au long de ce travail.

#### Je dédie ce modeste travail:

- À mes chers parents, **KPOMEGBE Kokou** et **FOLLY-BEBEY Dede edem**, pour votre soutien indéfectible, votre patience, vos sacrifices, votre affection et votre amour constants.
- À des personnes très chères, pour vos encouragements, votre affection et votre présence bienveillante, ainsi qu'à tous ceux qui ont contribué de près ou de loin à la poursuite de mes études.

Que Dieu vous garde et vous bénisse.

## **REMERCIEMENTS**

#### REMERCIEMENTS

Au terme de mes études à l'Institut de Formation Technique et Supérieur (IFTS), je rends grâce à Dieu le Tout-Puissant pour sa protection, sa grâce et son amour constant tout au long de ma formation.

Ce mémoire est le fruit des efforts conjoints de plusieurs personnes, à qui j'adresse mes sincères remerciements.

Je remercie profondément mes parents, pour leur amour, leur soutien indéfectible et leurs conseils avisés qui ont été essentiels à la réalisation de ce travail.

Je tiens sincèrement à adresser mes vifs et cordiaux remerciements :

- au Professeur *Koffi-Sa Eugène BEDJA*, Président du Conseil Scientifique et Pédagogique de l'IFTS, Ingénieur des Télécommunications, pour sa grande bienveillance, sa sympathie, ses conseils, ses encouragements précieux et enseignements de tous les jours ;
- à Mr *Kekely Moïse Kwoasi BIRAMAH*, Ingénieur Génie Électrique, Enseignant chercheur à l'IFTS, Directeur de ce mémoire, pour sa disponibilité, conseils et soutien lors de ce travail;
- aux enseignants de l'IFTS, pour la qualité de leur encadrement et la transmission de leurs savoirs ;
- au personnel de l'IFTS, pour leur accompagnement ;
- aux membres du jury, pour l'attention portée à mon travail;
- à mes camarades de promotion, pour les moments partagés ;
- à mon ami *Nathanaël edudzi SOMALI*, pour son aide précieuse pour ce projet ;
- et à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail. Que le Seigneur vous bénisse et vous récompense.



#### **RESUME**

Ce mémoire s'inscrit dans le contexte actuel de digitalisation des processus académiques et de modernisation des outils de gestion documentaire dans les établissements d'enseignement supérieur. Dans cette suite d'idée, notre projet consiste à concevoir et développer une plateforme web appelée MemorySpace, destinée à gérer efficacement les mémoires académiques de l'Institut de formation Technique et Supérieur (IFTS). Cette plateforme a pour objectif de centraliser les mémoires académiques de fin d'études, d'en faciliter leur consultation et de simplifier leur gestion. Pour mener à bien ce projet, une méthode structurée a été suivie, incluant le développement web avec le framework Laravel, le déploiement dans l'environnement Laragon, la modélisation de la base de données avec la méthode Merise, ainsi que l'intégration et les tests de la plateforme. Les résultats obtenus ont confirmé la fiabilité et la facilité d'utilisation du système, qui permet une consultation rapide des mémoires et une gestion centralisée efficace. À terme, *MemorySpace* pourra évoluer pour devenir une solution complète et moderne, répondant aux besoins des institutions académiques, valorisant les travaux des étudiants et améliorant l'accès aux ressources documentaires.

Mots-clés : Base de données ; framework Laravel ; Gestion documentaire ; Mémoires académique ; Plateforme web.



#### **ABSTRACT**

This thesis is part of the current context of digitalization of academic processes and the modernization of document management tools in higher education institutions. In line with this idea, our project involves designing and developing a web platform called MemorySpace, aimed at efficiently managing academic theses at the Institut de Formation Technique et Supérieur (IFTS). The purpose of this platform is to centralize final-year academic theses, facilitate their consultation, and simplify their management. To successfully carry out this project, a structured approach was followed, including web development using the Laravel framework, deployment in the Laragon environment, database modeling using the Merise method, as well as platform integration and testing. The results obtained confirmed the reliability and user-friendliness of the system, to theses efficient centralized which for quick access and management. Ultimately, Memory Space has the potential to evolve into a complete and modern solution that meets the needs of academic institutions, showcases student work, and improves access to documentary resources.

Keywords: Database; Laravel framework; Document management; Academic theses; Web platform.

## TABLE DES MATIERES

### TABLE DES MATIERES

## LISTE DES FIGURES

### LISTE DES FIGURES

CHAPITRE I:	
Figure 1.1:fonctionnement des application web	15
Figure 1.2:fonctionnement client/serveur	15
Figure 1.3:le rôle des serveurs d'application	16
CHAPITRE II :	
Figure 2.1: Entité	22
Figure 2.2: Association	23
Figure 2.3: icône du logiciel laragon	27
Figure 2.4: Modèle Entité –Association	30
Figure 2.5: Modèle Conceptuel de Données	31
Figure 2.6:Le Modèle Logique des Données (MLD)	32
Figure 2.7:Le diagramme des flux	33
Figure 2.8: Schéma de fonctionnement de la plateforme	33
CHAPITRE III :	
Figure 3.1: L'icône de MySQL	37
Figure 3.2: interface du PhpMyAdmin	37
Figure 3.3: Logiciel Visual Studio Code	39
Figure 3.4: Interface Visual Studio Code	40
Figure 3.5: Interface d'accueil	42
Figure 3.6: Interface d'inscription	43
Figure 3.7: Interface d'authentification	43
Figure 3.8: Interface du bibliothécaire	44
Figure 3.9: Interface d'authentification du bibliothécaire	45
Figure 3.10: Formulaire de saisie des mémoires	45
Figure 3.11: interface de la liste des documents déposer	46

## Liste des figures

Figure 3.12: Liste des étudiants inscrit	47
Figure 3.13: Interface de la Demandes de lecture	47
Figure 3.14: Interface d'importation	48
Figure 3.15: Interface d'étudiants	49
Figure 3.16: Interface documents	49
Figure 3.17: Interface de consultation des documents	50
Figure 3.18: Interface de lecture du document	50
Figure 3.19: Interface de consultation des documents de l'étudiant externe	51
Figure 3.20: Interface de lecture de description du document	52
Figure 3.21: Interface de Gestion de ses demandes de lecture	53

## LISTE DES TABLEAUX

## LISTE DES TABLEAUX

$\sim$ TT		_		<b>~</b> T	**	
<i>'</i> 'L	Λ	DΙ	' I ' L	JL		
CH.	$\boldsymbol{\vdash}$	ГΙ	11	<b>\</b> L:		

Tableau 2:1: passage du MCD au MLD	. 26
Tableau 2:2: passage du MCD au MPD	. 27
Tableau 2:3: Dictionnaire de données de notre application web	. 28

## LISTE DES ABREVIATIONS

#### LISTE DES ABREVIATIONS

IFTS : Institut de Formation Technique Supérieure

SGBD : Système de Gestion de Base de Données

URL: Uniform Resource Locator

HTML : Hypertext Markup Language

HTTP : Hypertext Transfer Protocol

CSS : Cascading Style Sheets

CMS : Content Management System

PHP : Hypertext Preprocessor

MVC : Modèle-Vue-Contrôleur

IIS : Internet Information Services

OVH : On Vous Héberge

AWS : Amazon Web Services

SQL : Structured Query Language

MCD : Modèle Conceptuel de Données

MLD : Modèle Logique de Données

MPD : Modèle Physique de Données

MCT : Modèle Conceptuel des Traitements

CIF : Contraintes d'Intégrité Fonctionnelle

DF : Dépendance Fonctionnelle

: Structured Query Language (Système de gestion de base

relationnelle)

VS Code : Visual Studio Code

BDD : Base de Données

MySQL

# INTRODUCTION GÉNÉRALE

Depuis l'invention de l'écriture, l'humanité a toujours cherché à préserver et transmettre le savoir à travers divers supports, notamment les livres. Ces derniers jouent un rôle essentiel dans la formation et l'acquisition des connaissances, qu'il s'agisse de romans, de manuels didactiques ou de mémoires académiques. Toutefois, avec l'accumulation croissante de ces documents, leur gestion et leur accessibilité représentent un défi majeur, nécessitant des systèmes modernes, efficaces et fiables.

Autrefois, la gestion des bibliothèques se faisait de manière entièrement manuelle. Bien que fonctionnelle, cette méthode présentait de nombreuses limites : risque élevé de perte ou de détérioration des documents, lenteur dans la recherche d'informations, difficulté de mise à jour, etc. Aujourd'hui, avec l'évolution des technologies de l'information et de la communication, des plateformes web, il est possible de numériser, centraliser et sécuriser ces documents, tout en facilitant leur consultation à distance via Internet.

C'est dans cette dynamique que s'inscrit notre projet de fin d'études. Après avoir constaté des difficultés dans la gestion des mémoires académiques à l'Institut de Formation Technique Supérieure (IFTS), nous avons entrepris « L'ETUDE ET CONCEPTION D'UN LOGICIEL DE GESTION DES DOCUMENTS DE SOUTENANCE DE L'IFTS », nommée MemorySpace, permettant le dépôt, la consultation, l'organisation et la traçabilité des documents de soutenance. Développée sous le framework Laravel, cette plateforme vise à moderniser le processus de gestion documentaire et à faciliter l'accès aux mémoires aussi bien pour les étudiants que pour les bibliothécaires, à travers une interface conviviale et sécurisée, accessible via un navigateur web.

Notre démarche comprend plusieurs étapes : l'étude des concepts fondamentaux des systèmes de gestion de bases de données (SGBD), l'analyse des besoins, la

conception du système, puis son implémentation et sa validation. Ce mémoire présentera également les choix techniques, les fonctionnalités de la plateforme, ainsi que les résultats obtenus à travers les tests réalisés.

À travers ce projet, nous souhaitons proposer une solution innovante, adaptée au contexte académique de l'IFTS, répondant aux exigences actuelles en matière de gestion numérique de l'information, tout en assurant l'accessibilité, la sécurité et la conservation des mémoires académiques

## **Chapitre I:**

# TECHNOLOGIES DE DÉVELOPPEMENT WEB

#### I.1.INTRODUCTION

Aujourd'hui, le web s'impose comme une source d'information incontournable et un moteur clé de la croissance rapide et continue d'Internet à travers le monde. Ce chapitre est organisé en deux sections principales : la première décrit en détail notre projet, et la seconde présente les technologies modernes utilisées pour le développement web.

#### I.2. PRESENTATION DU PROJET

Dans cette partie, nous allons présenter le projet que nous avons réalisé dans le cadre de notre mémoire. Il s'agit d'une plateforme web nommée MemorySpace, conçue pour améliorer la gestion et la consultation des mémoires académiques à l'IFTS. Avant de parler de ses objectifs et de son fonctionnement, il est important de comprendre le problème que ce projet cherche à résoudre.

#### I.2.1. Problématique

L'Institut de Formation Technique Supérieure (IFTS) est un établissement d'enseignement supérieur qui forme des étudiants dans divers domaines techniques, notamment le génie électrique et le génie civil. Sa mission principale est de préparer des techniciens supérieurs qualifiés, capables de répondre aux exigences du marché de l'emploi. À l'issue de leur formation, les étudiants doivent rédiger un mémoire de fin d'études, une étape cruciale pour évaluer leurs compétences et leurs acquis. Ces mémoires constituent également des ressources précieuses pour les promotions futures, servant à la fois de sources d'inspiration et de références académiques.

La bibliothèque actuelle de l'IFTS joue un rôle central en tant que centre de ressources documentaires pour les étudiants et les enseignants. Elle regroupe une

grande variété de documents académiques : mémoires de fin de formation, rapports de stage, ouvrages spécialisés et revues scientifiques.

Le processus de consultation de ces documents demeure également rudimentaire. Lorsqu'un étudiant souhaite consulter un document, il doit soit le lire sur place, soit inscrire son nom ainsi que le titre du document dans un cahier prévu à cet effet, le conserver temporairement, puis le restituer après consultation. Ce système, bien qu'il offre un minimum de suivi, reste très limité en termes de traçabilité et d'efficacité.

Cette gestion traditionnelle présente plusieurs limites majeures :

- ◆ Les documents sont stockés sur des supports variés et dispersés, sans véritable cohérence ni centralisation.
- ◆ La recherche d'un document s'avère souvent longue et nécessite l'intervention du personnel de la bibliothèque, ce qui engendre des délais importants, notamment en période de forte affluence.
- ◆ De plus, il n'existe pas de système numérique permettant une recherche rapide et efficace, rendant difficile, voire impossible, la localisation de certains mémoires, notamment ceux des années antérieures.
- ♦ Enfin, l'absence d'une traçabilité numérique nuit au suivi rigoureux de la consultation et de l'utilisation des documents.

Face à ces constats, la mise en place d'un système numérique de gestion documentaire s'impose comme une nécessité. Un tel système permettrait de centraliser les mémoires dans une base de données unique, d'en faciliter l'accès sécurisé via Internet, et d'assurer un suivi automatisé et rigoureux de toutes les opérations de consultation et de gestion, améliorant ainsi considérablement l'efficacité et la pérennité de la bibliothèque de l'IFTS.

#### I.2.2. Objectifs du projet

#### I.2.2.1. Objectif général

Dans le cadre de l'amélioration de la gestion académique et de l'accessibilité aux documents de fin de formation, il est essentiel de proposer une solution innovante, adaptée aux besoins de l'Institut. Cela conduit à la conception et au développement d'une plateforme web de gestion des documents de soutenance à l'IFTS, permettant un dépôt, une consultation et une administration numériques des documents, de manière structurée, efficace et sécurisée.

#### I.2.2.2. Objectifs spécifiques

Pour assurer une gestion optimale des documents de soutenance et répondre efficacement aux besoins des utilisateurs, il est nécessaire de définir des objectifs spécifiques qui guideront le développement de la plateforme. Ces objectifs visent à améliorer l'organisation, la sécurité et l'accessibilité des mémoires académiques, tout en garantissant une utilisation optimale et contrôlée du système.

- mettre en place un système de dépôt structuré des mémoires académiques;
- permettre une consultation rapide et filtrée grâce à des critères multiples (titre, filière, encadrant, année, etc.);
- gérer les différents types d'utilisateurs avec des droits d'accès spécifiques (étudiants internes, étudiants externes, bibliothécaires);
- garantir la sécurité des documents par un système d'authentification et de contrôle d'accès ;
- assurer la traçabilité et l'enregistrement des actions effectuées par les utilisateurs ;
- sauvegarder régulièrement les données pour éviter les pertes.

#### I.2.3. Cahier des charges

MemorySpace est une plateforme web développée sous Laravel, dédiée à la gestion et à la consultation des documents de soutenance. Son objectif est d'optimiser l'accès aux documents en offrant plusieurs options de recherche avancées. Actuellement, la recherche de documents à la bibliothèque de l'institut peut prendre une trentaine de minutes, voire une heure, car elle nécessite de connaître précisément certaines informations, ce qui peut être contraignant.

MemorySpace améliore ce processus en permettant de retrouver un document selon plusieurs critères :

- titre du mémoire
- filière concernée
- nom du directeur de mémoire
- année de soutenance

En plus d'optimiser l'efficacité de la recherche, la plateforme garantit également une meilleure sécurisation et accessibilité des documents.

L'accès à MemorySpace est réservé aux bibliothécaires, et aux étudiants. Chaque groupe d'utilisateurs dispose de fonctionnalités adaptées à ses besoins.

#### I.2.3.1. Bibliothécaire

Le bibliothécaire joue un rôle central dans la gestion des documents et des utilisateurs. Il doit d'abord se connecter via l'option *Connexion* de la plateforme pour accéder à son espace personnel, où il dispose des fonctionnalités suivantes :

- > gestion des documents :
  - dépôt de nouveaux documents ;

- consultation, téléchargement, modification et suppression des documents.
- > gestion des utilisateurs :
  - consultation et suppression des comptes étudiants ;
  - inscription des étudiants internes.
- gestion des demandes d'accès :

Le bibliothécaire est également chargé de gérer les autorisations d'accès aux mémoires pour les utilisateurs qui ne font pas partie de l'institut.

- traitement des requêtes de lecture soumises par les étudiants externes ou des personnes extérieures à l'établissement .
- > sécurisation des sessions :
  - option de déconnexion sécurisée.

#### I.2.3.2. Etudiant

L'accès aux documents varie selon le statut de l'étudiant :

- étudiants externes :
  - inscription requise pour consulter la description des documents ;
  - demande d'accès pour consulter le contenu intégral (soumise à l'approbation du bibliothécaire).
- > étudiants internes :
  - accès automatique et complet aux documents après inscription par le bibliothécaire ;
  - identifiants de connexion modifiables.

Comme pour le bibliothécaire, une option de déconnexion sécurisée est disponible.

#### I.3. TECHNOLOGIES DE DÉVELOPPEMENT WEB

Le développement web [1] est une discipline clé dans le domaine des technologies de l'information, impliquant la création et la maintenance de sites web et d'applications web. Depuis l'avènement d'Internet, le développement web a évolué pour devenir un pilier incontournable du monde moderne, influençant de nombreux aspects de notre vie quotidienne. Cet article explore les divers aspects du développement web, de ses fondements aux technologies les plus avancées, en passant par les compétences requises pour exceller dans ce domaine.

#### I.3.1. Les Fondations du Développement Web

Le développement web repose sur trois technologies de base : HTML, CSS et JavaScript.

#### I.3.1.1. HTML (HyperText Markup Language)

HTML est le langage standard utilisé pour structurer le contenu d'une page web. Il permet de créer des éléments tels que des paragraphes, des titres, des liens, des images, et bien plus encore. Chaque élément HTML est défini par des balises, qui structurent le contenu de manière hiérarchique.

#### **I.3.1.2.** CSS (Cascading Style Sheets)

CSS est utilisé pour décrire la présentation des pages web, y compris les styles, les couleurs et les mises en page. Grâce à CSS, les développeurs peuvent séparer la structure du contenu de son apparence visuelle, facilitant ainsi la maintenance et l'évolution des sites web.

#### I.3.1.3. JavaScript

JavaScript est un langage de programmation qui permet d'ajouter des interactions dynamiques et des comportements aux pages web. Il peut être utilisé pour créer des animations, valider des formulaires, et interagir avec les utilisateurs de

manière dynamique. JavaScript est également à la base de nombreux frameworks et bibliothèques populaires tels que React, Angular, et Vue.js.

#### I.3.2. Les Types de Développement Web

Le développement web peut être divisé en deux catégories principales : le développement front-end et le développement back-end.

#### I.3.2.1. Développement Front-End

Le développement front-end concerne la partie visible d'un site web avec laquelle les utilisateurs interagissent directement. Les développeurs front-end utilisent principalement HTML, CSS et JavaScript pour créer des interfaces utilisateur attrayantes et fonctionnelles. Ils doivent également s'assurer que les sites web sont réactifs, c'est-à-dire qu'ils s'adaptent à différentes tailles d'écran et appareils.

#### I.3.2.2. Développement Back-End

Le développement back-end se concentre sur la partie serveur d'une application web. Il comprend la gestion des bases de données, la logique côté serveur, l'authentification des utilisateurs, et la communication avec l'interface front-end. Les développeurs back-end utilisent divers langages de programmation, tels que PHP, Python, Ruby, Java, et Node.js, ainsi que des systèmes de gestion de bases de données comme MySQL, PostgreSQL, et MongoDB.

On peut toutefois aussi inclure la base de données parmi les types de développement web.

#### I.3.3. LES SITES WEB

Un site web est un ensemble de document visualisable dans un navigateur, nommé page web sous un seul nom de domaine. Il est hébergé sur un ordinateur(serveur) qui est connecté au réseau mondial internet, et possède un identificateur appelé

adresse URL (Uniform Resource Locator), qui permet de trouver le site sur le web et identifier le type de protocole d'accès qui lui est associé.

Une page web est ce qui fait fonctionner un site web. C'est un document unique qui peut être affiché par toutes les navigateurs web, elle est écrite en utilisant le langage HTML (Hyper Text Markup Langage) [2] et elle contient du texte, des images, de sons, de vidéos et des liens vers d'autres pages web.

Il existe deux types de site web:

#### I.3.3.1. Site web statique

Un site web statique [3] se compose de pages dont le contenu est fixe, c'est-à-dire qu'il ne change pas automatiquement en fonction des actions de l'utilisateur. Ces sites sont souvent utilisés pour des informations simples et ne nécessitent pas d'interaction complexe avec la base de données.

#### a) Les avantages du site statique

- ➤ simplicité de développement : la création de sites statiques est rapide et peu coûteuse, car elle nécessite seulement des connaissances en HTML, CSS et parfois JavaScript.
- performance : les sites statiques se chargent rapidement, car chaque page est déjà pré-générée et n'a pas besoin d'être modifiée à chaque requête.
- > sécurité : sans base de données ni logique serveur complexe, les sites statiques sont moins vulnérables aux attaques.

#### b) Les inconvénients du site statique

mise à jour difficile : Il faut modifier manuellement chaque page pour changer le contenu.

- manque d'interactivité : Il n'y a pas de fonctionnalités dynamiques comme des formulaires ou un contenu personnalisé.
- > scalabilité limitée : Il devient compliqué à gérer si le site se développe.
- > pas de base de données : Il ne peut pas stocker ou gérer des informations dynamiques.

#### I.3.3.2. Site web dynamique [4]

Un site web dynamique est un site dont le contenu peut changer en fonction des actions de l'utilisateur. Contrairement aux sites statiques, ils sont souvent alimentés par des bases de données et permettent une interactivité riche avec les utilisateurs.

#### a) Les avantages du site dynamique

- > interactivité : les utilisateurs peuvent interagir avec le site, remplir des formulaires, consulter des données personnalisées, etc.
- mise à jour facile : le contenu dynamique est souvent géré via un CMS (Content Management System) ou directement par des scripts, ce qui permet de modifier le contenu sans toucher aux pages du site.
- gestion des données : ces sites permettent une gestion et une consultation efficace des données stockées dans une base de données.

#### b) Les inconvénients du site dynamique

- > complexité de développement : Plus difficile à créer et à gérer qu'un site statique.
- ➤ coût de maintenance élevé : Nécessite des mises à jour régulières et des corrections.
- ➤ temps de chargement plus long : Le contenu est généré à chaque demande, ce qui peut ralentir le site.

- sécurité : Plus vulnérable aux attaques en raison de l'interaction avec les bases de données.
- ➤ hébergement plus cher : Nécessite plus de ressources serveur.

#### I.4. LES APPLICATIONS WEB

Les applications web sont des programmes accessibles via un navigateur internet. Elles sont souvent utilisées pour remplacer des logiciels installés localement. [5] Par exemple, des outils comme Google Docs ou des systèmes de gestion de contenu utilisent des applications web.

#### I.4.1. Fonctionnement d'une application web [6]

Une application web fonctionne généralement selon une architecture en trois niveau, comme illustré dans la figure 1.1 :

- ♦ le front-end (niveau1)
- ♦ le back-end (niveau2)
- ♦ la base de données (niveau3)

Dans le cadre de ce projet, Laravel a été utilisé comme framework PHP pour le développement de l'application web. Laravel facilite le développement rapide d'applications robustes grâce à son architecture MVC (Model-View-Controller), sa gestion des routes, ses migrations de base de données, et son moteur de template Blade.

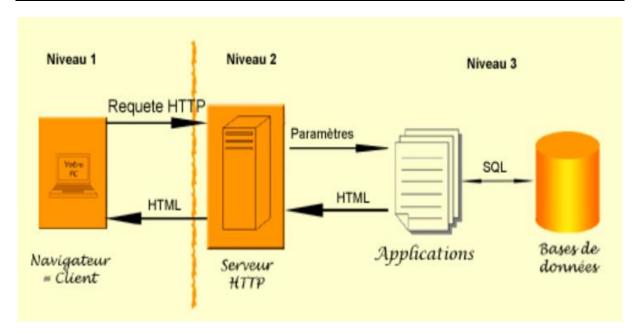


Figure 1.1: le fonctionnement des application web

#### I.4.2. L'APPLICATION CLIENT/SERVEUR

Le modèle client/serveur est au cœur des applications web. Dans ce modèle, le client (navigateur web) envoie des requêtes au serveur (qui héberge l'application web) et reçoit des réponses. Le serveur traite les requêtes, interagit avec la base de données et renvoie les résultats au client.

Ce modèle est utilisé dans la majorité des applications web modernes [9] et est illustre dans la figure 1.2.

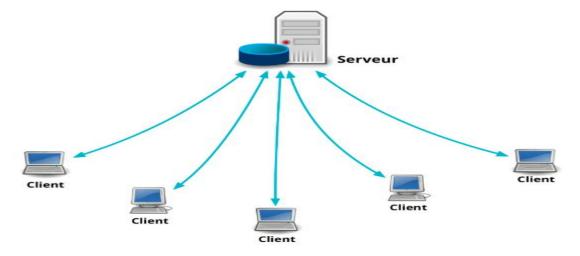


Figure 1.2: fonctionnement client/serveur

#### I.5. LES SERVEURS WEB

Un serveur web est un logiciel qui permet de gérer les requêtes HTTP (Hypertext Transfer Protocol) des utilisateurs. Il reçoit les demandes envoyées par les navigateurs et renvoie les pages web appropriées. Les serveurs web les plus couramment utilisés sont Apache, Nginx et IIS (Internet Information Services) [10].

Dans le cadre du développement local du projet, l'environnement Laragon a été utilisé. Laragon est une solution tout-en-un qui intègre un serveur Apache ou Nginx, MySQL, PHP, et d'autres outils pour créer un environnement de développement léger, rapide et facile à configurer.

#### I.5.1. Les serveurs d'application

Un serveur d'application est responsable du traitement des requêtes liées à la logique métier de l'application. Contrairement au serveur web, qui se charge principalement de la gestion des requêtes HTTP, le serveur d'application gère des processus complexes et assure la communication avec les bases de données.

Des serveurs comme Tomcat, WebLogic et JBoss sont des exemples de serveurs d'application utilisés dans le développement web [11]. Le rôle de ces serveurs est illustré dans la figure 1.3.

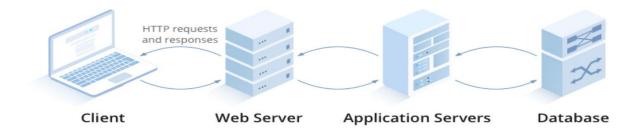


Figure 1.3: le rôle des serveurs d'application

#### I.6. L'HÉBERGEMENT D'UNE APPLICATION WEB

L'hébergement d'une application web fait référence au processus par lequel les fichiers et la base de données de l'application sont stockés sur un serveur distant, afin qu'ils soient accessibles via Internet. Les options d'hébergement incluent des services comme Heroku, AWS (Amazon Web Services), ou des hébergeurs traditionnels comme OVH (On Vous Héberge) [12]. Le choix de l'hébergement dépend de la taille de l'application, de son trafic et de ses besoins en sécurité.

Enfin, comme le projet repose sur une plateforme web, cela signifie que l'application est accessible via un navigateur à travers Internet. Les plateformes web offrent de nombreux avantages, notamment l'accessibilité, la compatibilité multi-appareils, et la facilité de déploiement sans nécessiter d'installation locale.

#### I.7. CONCLUSION

Ce chapitre a posé les bases de notre le projet MemorySpace, à travers ses objectifs et les technologies utilisées, telles que Laravel et Laragon. Ces bases serviront pour l'analyse et la conception du système dans le chapitre suivant, notamment la modélisation de la base de données et le traitement des informations.

# Chapitre II : ETUDE ET CONCEPTION DU SYSTEME

#### II.1. INTRODUCTION

La conception est une étape clé du développement web, facilitant l'organisation et la structuration des données. Ce chapitre présente les bases de données, les étapes de modélisation, ainsi que la méthode Merise utilisée pour concevoir la plateforme MemorySpace.

#### II.2. L'IMPORTANCE DE LA MODELISATION

La modélisation joue un rôle fondamental dans la conception d'une application web. Elle offre une représentation claire, structurée et abstraite du système à développer, avant même le début du codage. En définissant les entités, leurs relations et les interactions possibles, la modélisation permet de poser les bases solides sur lesquelles reposera l'architecture du projet. Elle facilite la compréhension globale du fonctionnement de l'application, aussi bien pour les développeurs que pour les parties prenantes, en assurant une vision commune du système. Cela permet d'anticiper les éventuelles erreurs de conception, d'identifier les incohérences ou redondances, et ainsi de réduire les coûts liés aux modifications ultérieures. En outre, la modélisation contribue à une meilleure planification des ressources humaines et techniques, en définissant clairement les tâches à accomplir et les priorités. En facilitant la communication et la validation des besoins fonctionnels et techniques, elle garantit une meilleure adéquation entre la solution développée et les attentes initiales, tout en renforçant la qualité et la fiabilité du produit final [13].

#### II.3. BASE DE DONNEE

Une base de données [14] est un système structuré permettant de stocker, d'organiser, de gérer et de manipuler des informations. Elle facilite un accès rapide aux données, leur mise à jour ainsi que leur sécurisation. Les bases de

données jouent un rôle fondamental dans toute application web moderne nécessitant une gestion dynamique et efficace de l'information.

Pour manipuler une base de données, on utilise généralement un Système de Gestion de Base de Données Relationnelles (SGBDR), qui constitue un ensemble de logiciels permettant l'interaction avec les données de manière structurée. Ce système repose sur le langage SQL (Structured Query Language), un langage standardisé permettant d'effectuer diverses opérations telles que la création de tables, l'insertion, la modification, la suppression ou la requête de données. Grâce à cette approche relationnelle, les données sont organisées sous forme de tables interconnectées, ce qui facilite leur cohérence, leur intégrité et leur traitement.

Ainsi, le SGBDR représente un outil incontournable pour les développeurs d'applications web, car il permet non seulement de gérer efficacement les volumes importants de données, mais aussi d'assurer la fiabilité et la sécurité des transactions effectuées sur ces données.

#### II.3.1. Les bases de données relationnelles

Il existe plusieurs types de bases de données, mais dans notre cas, le modèle relationnel est le plus adapté. En effet, notre application web nécessite la gestion de données dynamiques et liées entre elles. Le modèle relationnel permet d'organiser les données sous forme de tables, ce qui facilite leur manipulation et leur structuration. Ce type de base est également le plus répandu et utilisé actuellement dans le développement d'applications web.

#### II.3.1.1. Historique

Le modèle relationnel a été introduit en 1970 par Edgar F. Codd, chercheur chez IBM. Il propose une organisation des données sous forme de relations (tables à deux dimensions) manipulées via le langage SQL. Ce modèle est simple, normalisé et largement adopté [14].

#### II.3.1.2. Principe de fonctionnement

Les données sont stockées sous forme de tables, où chaque ligne représente un enregistrement, et chaque colonne, un attribut. Les clés primaires et clés étrangères assurent l'intégrité référentielle entre les tables.

#### SQL permet:

- ➤ la création (CREATE) des bases de données, des tables, des vues, des index, et des procédures stockées;
- ➤ la lecture (SELECT) des données, avec la possibilité d'effectuer des requêtes simples ou complexes;
- la mise à jour (UPDATE) des données existantes dans les tables ;
- > et la suppression (DELETE) des données dans les tables ;
- la modification de la structure des bases de données et des tables (ALTER, DROP);
- ➤ la gestion des droits d'accès et des utilisateurs (GRANT, REVOKE) ;
- ➤ le contrôle des transactions (BEGIN, COMMIT, ROLLBACK) pour garantir l'intégrité des données;
- ➤ l'optimisation des performances grâce à la création d'index et d'autres mécanismes de gestion.

# II.3.1.3. Avantages et inconvénients d'une BDD relationnelle

Les avantages des bases de données relationnelle sont les suivantes :

- organisation claire via les tables ;
- réduction des redondances grâce à la normalisation ;
- utilisation de langages déclaratifs (SQL);
- contrôle d'accès et sécurité des données ;
- intégrité des données assurée par les relations ;

Ils ont aussi des inconvénients :

- moins performant sur de très grands volumes en temps réel ;
- rigidité face aux modifications fréquentes de structure ;
- courbe d'apprentissage du SQL pour les optimisations complexes.

#### II.3.2 Le choix du modèle

Avant de concevoir une base de données, une analyse approfondie s'impose. Étant donné que notre base repose sur le modèle relationnel, nous avons choisi la méthode Merise, parfaitement adaptée à ce type de modélisation [15].

#### II.3.2.1 La méthode Merise

La méthode merise est une méthode de conception, d'analyse et de modélisation des systèmes d'information. Elle distingue les données des traitements pour une meilleure structuration du projet. Cette méthode permet de passer des besoins fonctionnels à la conception physique du système, en plusieurs étapes.

#### II.4. MODELISATION DE LA BASE DE DONNEES

Pour garantir une base de données cohérente, nous avons utilisé la modélisation en trois niveaux : MCD (Modèle Conceptuel), MLD (Modèle Logique), MPD (Modèle Physique) [16].

#### II.4.1. Le dictionnaire des données

Le dictionnaire de données recense tous les champs nécessaires à la base, avec leurs types, tailles, et remarques spécifiques.

Colonnes typiques:

- code mnémonique,
- désignation,

- type de données,
- taille,
- remarques (clé primaire, étrangère, etc.).

#### II.4.2. Le Modèle Conceptuel de Données (MCD)

Le MCD permet de représenter graphiquement les données d'un système d'information. Ces données sont regroupées en entités reliées par des relations appelées associations, dans le but de faciliter la compréhension du système d'information à concevoir.

Ce modèle est composé des entités, des relations, des cardinalités, des contraintes d'intégrité fonctionnelle ainsi que des dépendances fonctionnelles.

#### > Entités :

Une entité est un objet concret ou abstrait appartenant à l'univers du système d'information que l'on souhaite modéliser. Elle représente un ensemble d'objets possédant les mêmes caractéristiques. Chaque entité est unique et est définie par un identifiant (ou clé primaire) ainsi que par un ensemble de propriétés, appelées également attributs.

Dans le modèle conceptuel, une entité est généralement représentée graphiquement par un rectangle, comme illustré la figure 2.1.



Figure 2.1: Entité

#### > Association :

Une association représente une liaison entre une ou plusieurs entités. Aussi appelée lien, elle ne possède pas d'existence propre en dehors des entités qu'elle relie. Elle est généralement nommée par un verbe, ce qui permet d'exprimer clairement la nature de la relation entre les entités.

Graphiquement, une association est représentée par une ellipse, comme illustré la figure 2.2.

Nom de l'association liste des données portées

Figure 2.2: Association

#### Propriété :

Une propriété est un composant d'une entité, représentant une caractéristique ou une information la décrivant. On peut également retrouver des propriétés au niveau des associations, lorsque celles-ci portent des informations spécifiques.

#### > Cardinalités :

La cardinalité définit le nombre de fois qu'une entité peut participer à une relation. On distingue deux types de cardinalités :

- la cardinalité minimale (0 ou 1) :
  - o signifie que la participation de l'entité à la relation est facultative.
  - 1 signifie que la participation est obligatoire.
- la cardinalité maximale (1 ou N) :
  - o 1 signifie que l'entité ne participe qu'une seule fois à la relation.

 N signifie qu'elle peut y participer plusieurs fois (valeur indéfinie mais supérieure à 1).

#### > Identifiant d'une relation :

Il s'agit d'un identifiant unique, obtenu par la combinaison des identifiants des entités participant à la relation. Cet identifiant permet de distinguer de manière unique chaque occurrence de la relation.

#### Contraintes d'intégrité fonctionnelle (CIF) :

Les CIF s'appliquent entre deux entités liées par une relation où l'une d'elles possède une cardinalité maximale de 1 (par exemple : 0,1 ou 1,1). Ces contraintes sont généralement notées sous la forme 1,1-XX, et indiquent qu'il existe une dépendance fonctionnelle entre les entités. Lors du passage du modèle logique (MLD) ou du modèle physique (MPD), la CIF se traduit par une clé étrangère ajoutée à la table correspondant à l'entité du côté "XX", c'est-à-dire celle qui dépend de l'autre.

# > Les dépendances fonctionnelles :

Une dépendance fonctionnelle entre deux attributs A et B signifie que B dépend fonctionnellement de A (notée  $A \rightarrow B$ ), c'est-à-dire qu'à chaque valeur de A correspond au plus une seule valeur de B.

#### a) Dépendance fonctionnelle élémentaire :

Soit deux attributs A et B, la dépendance  $A \to B$  est dite élémentaire s'il n'existe pas de sous-ensemble de A (par exemple C) tel que  $C \to B$ . Cela signifie que A est le plus petit ensemble possible déterminant B.

#### b) Dépendance fonctionnelle directe :

La dépendance  $A \to B$  est dite directe s'il n'existe pas d'attribut C tel qu' $A \to C$  et  $C \to B$ . Autrement dit, il n'existe pas de dépendance intermédiaire entre A et B.

#### II.4.3. Le Modèle Logique de Données (MLD)

Le Modèle Logique de Données (MLD) constitue une étape de transformation du modèle conceptuel vers une structure exploitable par un système de gestion de base de données relationnelle. Il s'agit de formaliser les entités et les relations identifiées précédemment sous forme de relations logiques cohérentes, en respectant les règles du modèle relationnel.

Chaque entité du système est traduite en une relation (ou table), et chacun de ses attributs devient un champ dans cette table. À ce stade, les dépendances fonctionnelles, les cardinalités, et les contraintes sont définies de manière plus rigoureuse afin de garantir la cohérence et la fiabilité des données à stocker.

#### II.4.3.1. Passage du MCD → MLD

La transition du MCD au MLD repose sur l'application de règles de transformation normalisées. Ces règles permettent de convertir les entités et les associations en structures tabulaires prêtes à être implémentées dans un environnement de base de données.

# II.4.3.2. Règles principales de transformation

- les entités deviennent des tables : chaque entité identifiée dans le MCD correspond directement à une table contenant ses attributs comme colonnes.
- les associations sont converties selon leur cardinalité :

- une association de type (1,n) avec (1,1) est représentée par une clé étrangère insérée dans la table du côté (1,n).
- une association de type (n,n) implique la création d'une table intermédiaire contenant les clés étrangères des deux entités concernées.
- une association (1,1)-(1,1) peut être intégrée à une seule table, en fonction des attributs associés.
- une association (1,n)-(1,n) nécessite une table relationnelle dédiée comportant les deux clés étrangères pour assurer l'intégrité des liens.

Tableau 2.1: passage du MCD au MLD

MCD	MLD
Association / Entité	Relation
Propriété de l'entité	Attribut de la relation
Identifiant de l'entité	Clés de la relation
Relation plusieurs (0,n ou 1,n) à plusieurs (0,n ou 1,n)	Relation
Relation n binaire (0,1 ou 1,1) à	Exportation clé étrangère et attributs
plusieurs (0,n ou 1,n)	portes

#### II.4.3.3. Formes normales

Les formes normales vérifient la cohérence du MLD :

➤ 1NF : présence d'une clé et DF élémentaires.

➤ 2NF : toutes les DF sont élémentaires.

> 3NF: toutes les DF sont directes

# II.4.4. Le Modèle Physique de Données (MPD)

Le **MPD** est la dernière étape, traduisant chaque entité/table avec les **types de données précis** (VARCHAR, INT, DATE, etc.), en vue de la création effective de la base dans MySQL.

### II.4.4.1. Passage MCD $\rightarrow$ MPD

Chaque entité devient une table avec :

- déclaration des clés primaires,
- ➤ attribution de types SQL aux champs.

**Tableau 2.2:** passage du MCD au MPD.

MCD	MPD		
Entité	Table		
Propriété de l'entité	Champs		
Identifiant de l'entité	Clés de la table		

# II.5. MODELISATION DE LA BASE DE DONNEES DE NOTRE APPLICATION WEB

Dans notre projet, nous avons utilisé Laragon, un environnement de développement qui intègre MySQL comme moteur de base de données. L'icône du logiciel Laragon en figure 2.3.



Figure 2.3: icône du logiciel laragon

# II.5.1. Le Modèle Conceptuel de Données (MCD)

#### Dictionnaire de données

Dans ce tableau, nous avons recensé l'ensemble des identifiants et des entités, ainsi que leurs désignations, types et tailles respectifs, afin de faciliter la modélisation de la base de données.

**Tableau 2.3 :** Dictionnaire de données de notre application web.

Nom du champ	Type	Taille	Entité	Description
id	INT	-	Utilisateur	Identifiant unique auto- incrémenté
Nom	VARCHAR	100	Utilisateur	Nom de l'utilisateur
Prénom	VARCHAR	100	Utilisateur	Prénom de l'utilisateur
Email	VARCHAR	150	Utilisateur	Adresse email utilisée comme identifiant de connexion
Mot_de_passe	VARCHAR	255	Utilisateur	Mot de passe crypté
Thème	VARCHAR	255	Document	Titre du document de mémoire
Auteur	VARCHAR	200	Document	Nom de l'auteur du document de mémoire
directeur	VARCHAR	200	Document	Nom du directeur de mémoire
Filière	VARCHAR	100	Document	Filière concernée

Tableau 2.3 : (suite)

Annee_Soutenance	YEAR	-	Document	Année de soutenance
Résume	TEXT	-	Document	Résumé du mémoire
Chemin fichier	VARCHAR	255	Document	Lien vers le fichier PDF du mémoire stocké
Visible_par_externe	BOOLEAN	-	Document	Définit si le document peut être consulté par un étudiant externe
id_demande	INT	-	Demande Acces	Identifiant unique de la demande
id_etudiant	INT	-	Demande Acces	Référence à l'étudiant ayant fait la demande
id_document	INT	-	Demande Acces	Référence au document demandé
Statut_demande	ENUM	-	Demande Acces	"en_attente", "acceptee", "refusee"
id_bibliothecaire	INT	-	Utilisateur	Bibliothécaire ayant traité la demande

# Légende

VARCHAR : Variable Character = chaîne de caractères de longueur variable

INT : Integer = nombre entier

YEAR : Year = année sur 4 chiffres

TEXT : Text = texte long (pour des contenus de grande taille)

BOOLEAN : Boolean = valeur booléenne (vrai ou faux)

ENUM : Enumeration = liste de choix prédéfinis (valeurs limitées, ex : "interne", "externe")

#### II.5.2. Modèle Entité –Association (MEA)

Ce modèle représente la schématisation de la structure d'accès théorique en faisant intervenir les notions d'entité, d'association et de cardinalité.

La figure 2.4 montre le formalisme du modèle entité association.

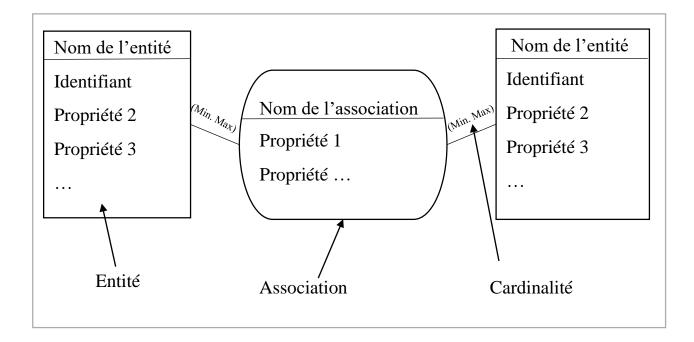


Figure 2.4: Modèle Entité – Association

Pour mieux comprendre l'organisation logique des informations au sein du système, il est essentiel de représenter graphiquement les entités et leurs relations. Cette représentation permet de visualiser la structure conceptuelle des données de manière claire et structurée.

La figure 2.5 montre la schématisation du modèle conceptuel des données.

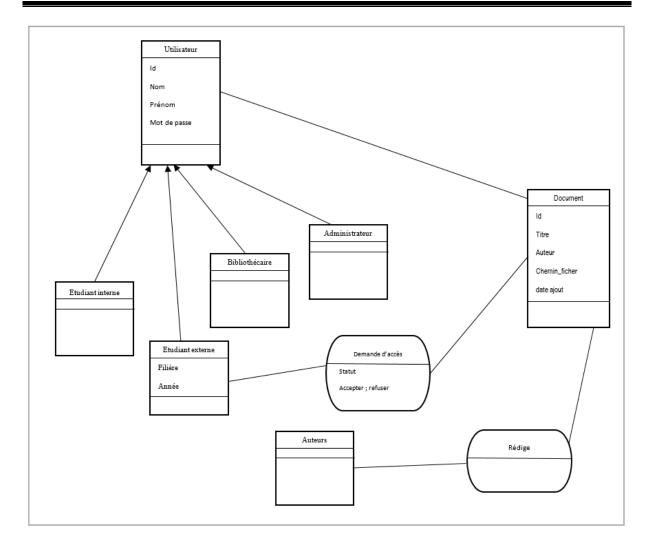


Figure 2.5: Modèle Conceptuel de Données

# II.5.3. Le Modèle Logique des Données (MLD)

Le modèle logique des données consiste à décrire la structure des données utilisées sans faire référence à un langage de programmation.

La figure 2.6 montre la schématisation du modèle logique des données.

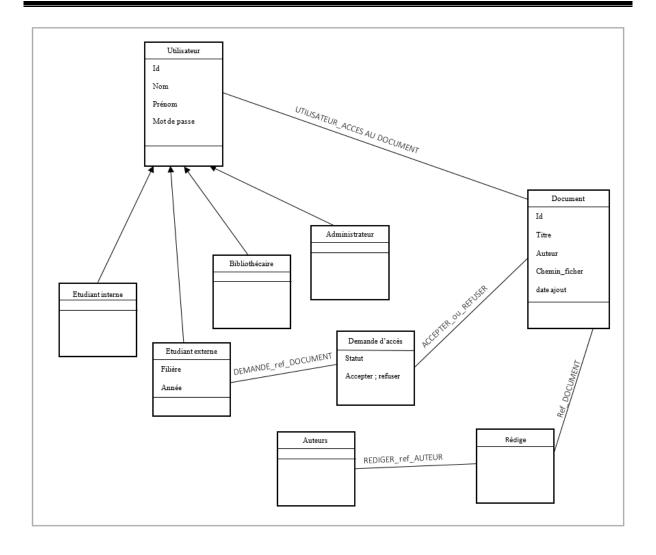


Figure 2.6: Le Modèle Logique des Données (MLD)

# II.5.4. Le Modèle Conceptuel des Traitements (MCT)

Le Modèle Conceptuel des Traitements (MCT) permet de représenter la dynamique du système d'information, c'est-à-dire l'ensemble des opérations réalisées en réponse à des événements. Ce modèle offre une vue schématique de l'activité du système d'information, sans faire référence aux choix organisationnels ni aux moyens techniques d'exécution.

#### II.5.4.1. Le diagramme des flux

La figure 2.7 montre les flux entre la bibliothèque de IFTS et les étudiants (étudiants de l'IFTS ; étudiants externe).

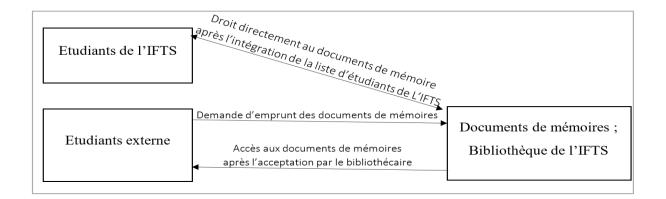


Figure 2.7: Le diagramme des flux

La figure 2.8 montre Schéma de fonctionnement de la plateforme.

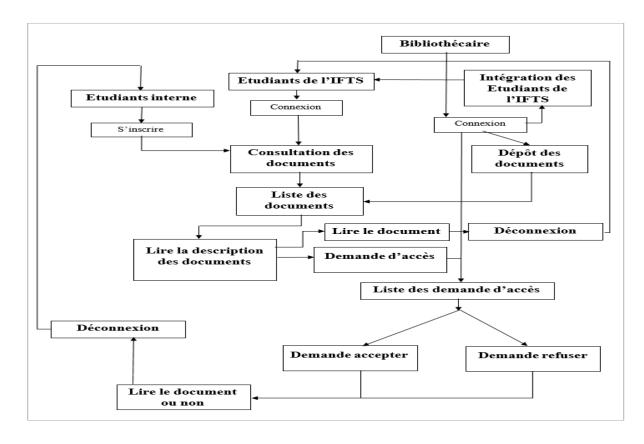


Figure 2.8: Schéma de fonctionnement de la plateforme

#### II.8. CONCLUSION

Ce chapitre a présenté l'étude et la conception du système. À travers la méthode Merise, nous avons défini la structure des données et leurs relations. Les modèles MCD, MLD et MPD ont permis une modélisation précise. Cette étape prépare efficacement la phase de développement de l'application.

# Chapitre III: REALISATION DE L'APPLICATION WEB

#### III.1. INTRODUCTION

Dans ce chapitre, nous présentons les différents outils informatiques utilisés pour le développement de l'application web. Nous décrivons également les principales interfaces de l'application, en mettant en lumière leur utilité ainsi que leur rôle dans le fonctionnement global du système.

# III.2. TECHNOLOGIES EMPLOYÉES

Pour réaliser notre application web, nous avons opté pour les outils informatiques suivants : Laragon, un environnement de développement local qui intègre plusieurs services essentiels tels que MySQL, PHP, PhpMyAdmin et Apache ; Visual Studio Code, que nous utilisons pour travailler avec Laravel ; et Laravel, un framework PHP moderne facilitant le développement d'applications web robustes.

#### III.2.1. Laragon

Laragon est un environnement de développement local qui permet de créer des applications web en toute simplicité. Il est particulièrement apprécié des développeurs en raison de sa configuration facile et de sa légèreté, offrant une alternative rapide et efficace à des solutions plus complexes comme XAMPP ou WAMP. Laragon intègre un certain nombre d'outils essentiels pour le développement d'applications web, tels que :

MySQL: un système de gestion de base de données relationnelle (SGBDR) utilisé pour stocker et organiser les données de l'application. L'icône de MySQL en figure 3.1.



Figure 3.1: L'icône de MySQL

**PHP**: un langage de programmation côté serveur, utilisé pour le traitement des données et la génération dynamique de pages web.

**PhpMyAdmin :** est une interface graphique permettant de gérer les bases de données MySQL de manière conviviale et intuitive, comme illustré à la Figure 3.2.

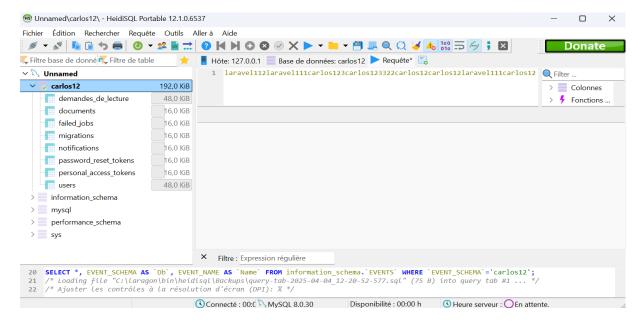


Figure 3.2: interface du PhpMyAdmin

**Apache**: un serveur web permettant de servir les fichiers et applications web aux utilisateurs.

Laragon se distingue par sa simplicité d'installation et sa capacité à offrir un environnement de développement complet dès le départ, avec des configurations

par défaut adaptées aux besoins des développeurs web. De plus, il permet d'utiliser plusieurs versions de PHP, ce qui est particulièrement utile pour tester la compatibilité de l'application avec différentes configurations.

L'une des fonctionnalités les plus appréciées de Laragon est sa capacité à gérer des environnements virtuels, ce qui permet de configurer facilement des applications distinctes dans des conteneurs séparés. Cela facilite la gestion de projets multiples sans risque de conflit entre les versions des outils ou des dépendances.

Dans le cadre du développement de notre application web, Laragon a constitué la base de notre environnement de développement local, permettant de travailler efficacement avec Laravel et d'assurer une gestion fluide des bases de données et des services associés.

#### III.2.2. Visual Studio Code

Visual Studio Code, est un éditeur de code source moderne et léger développé par Microsoft. Il est largement utilisé par les développeurs pour son efficacité, sa rapidité et sa flexibilité. VS Code prend en charge de nombreux langages de programmation, dont PHP, ce qui le rend parfaitement adapté pour le développement de notre application web avec le framework Laravel. L'icône de VS Code en figure 3.3.



Figure 3.3: Logiciel Visual Studio Code

Parmi les fonctionnalités clés qui font de Visual Studio Code un outil apprécié des développeurs, on retrouve :

- la coloration syntaxique, qui améliore la lisibilité du code.
- ➤ l'auto complétion intelligente (IntelliSense), qui propose des suggestions de code adaptées au contexte.
- le débogage intégré, facilitant l'identification et la correction des erreurs.
- ➤ le support des extensions, permettant d'ajouter des fonctionnalités, notamment pour Laravel, PHP, MySQL et Git.
- ➤ le terminal intégré, qui permet d'exécuter des commandes sans quitter l'éditeur.

Léger et rapide, Visual Studio Code offre pourtant des fonctionnalités proches des environnements de développement plus lourds. Dans le cadre de notre projet, il a été utilisé pour écrire et organiser le code source de l'application. Ses extensions dédiées à Laravel et PHP ont facilité un développement efficace, et l'intégration avec Git a simplifié la gestion des versions du code. La figure 3.4 illustre l'interface de Visual Studio Code utilisée durant le développement.

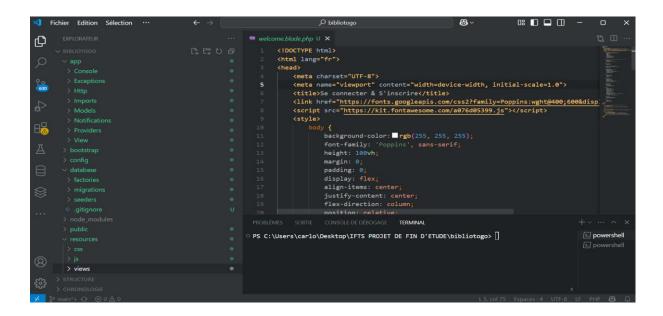


Figure 3.4: Interface Visual Studio Code

#### III.2.3. Laravel

Laravel est un framework PHP open source, utilisé pour le développement des applications web dynamiques. Il est composé d'un ensemble de fichiers PHP et utilise une base de données relationnelle comme MySQL. Il est destiné aux développeurs ayant des connaissances en programmation, notamment en PHP et en architecture MVC (Modèle-Vue-Contrôleur). Laravel permet de créer des applications web robustes et performantes grâce à des outils modernes et efficaces. Il est parmi les frameworks PHP les plus populaires et il est en constante amélioration.

Ce framework se compose de deux parties principales :

- La partie back-end (administration) : c'est la partie qui permet de gérer la logique métier, les bases de données, les utilisateurs, et les différentes fonctionnalités internes de l'application.
- La partie front-end (interface utilisateur) : c'est la partie visible par les utilisateurs, développée à l'aide du moteur de template Blade intégré dans Laravel.

# Les avantages de Laravel :

- > structure MVC bien organisée : permet de séparer les données, l'affichage et la logique pour un code plus propre.
- ➤ ORM Eloquent pour gérer facilement les bases de données : manipule les tables comme des objets sans écrire du SQL compliqué.
- sécurité renforcée : protège contre les attaques comme CSRF, XSS et injections SQL.
- routage simple et puissant : facilite la gestion des liens et des URL dans l'application.

- > moteur de template Blade rapide : simplifie la création de pages HTML dynamiques avec des instructions faciles.
- gestion facile des migrations de bases de données : permet de créer et modifier les tables sans accéder directement à la base.
- > système d'authentification intégré : offre des fonctionnalités prêtes pour l'inscription, la connexion et la protection des pages.

#### III.3. PRESENTATION DE L'APPLICATION

Dans cette section, nous allons présenter l'application web Memoryspace développée dans le cadre de ce projet. Cette présentation portera sur l'interface utilisateur, les principales fonctionnalités offertes ainsi que l'organisation générale du système. Elle permettra de mieux comprendre comment les différentes parties de l'application interagissent pour répondre aux besoins identifiés lors de la phase d'analyse.

#### III.3.1. Interface d'accueil

Au lancement de l'application **MemorySpace**, l'utilisateur est accueilli par une interface d'accueil claire et intuitive. Celle-ci propose un bouton À **propos**, qui montre une page explicative détaillant le fonctionnement de la plateforme. Deux autres options sont également disponibles : **S'inscrire** et **Connexion**, qui permettent respectivement à l'utilisateur de créer un compte ou de se connecter à son espace personnel afin d'accéder aux différentes fonctionnalités offertes par **MemorySpace**.

La **figure 3.5** illustre l'interface d'accueil de l'application.

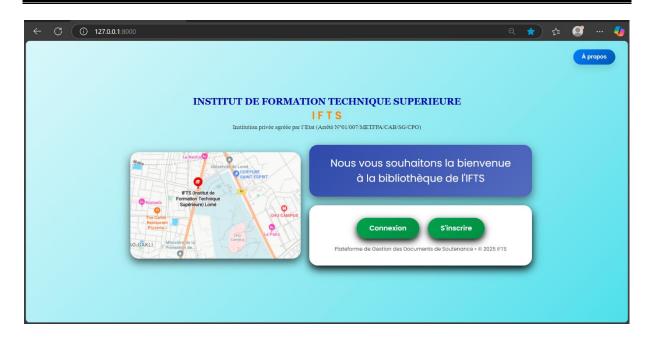


Figure 3.5: Interface d'accueil

#### S'inscrire

L'option S'inscrire permet aux nouveaux utilisateurs de créer un compte personnel afin d'accéder à la plateforme et à ses différentes fonctionnalités. La figure 3.6 présente l'interface d'inscription proposée par l'application.

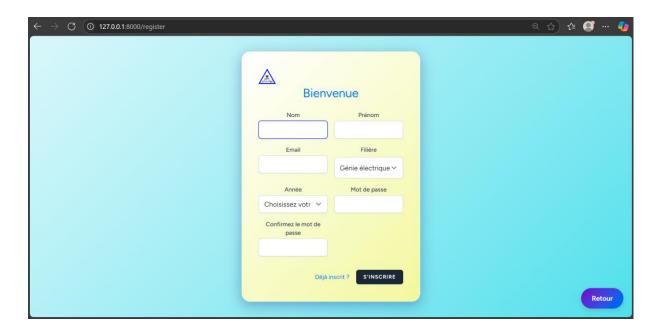


Figure 3.6: Interface d'inscription

#### **Connexion**

L'option Connexion permet aux utilisateurs déjà inscrits d'accéder à leur espace personnel. La figure 3.7 ci-dessous présente Interface d'authentification proposée par l'application.

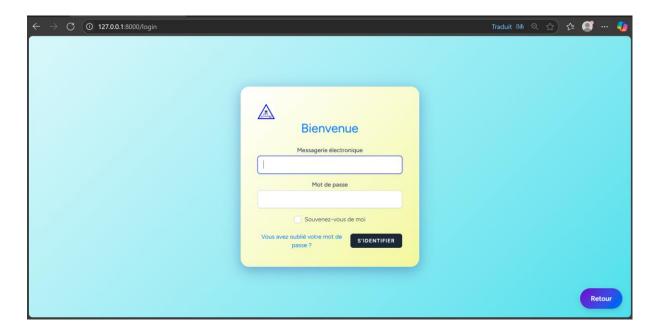


Figure 3.7: Interface d'authentification

#### III.3.2. Interface de travaille du bibliothécaire

Les fonctionnalités principales de la plateforme sont exclusivement accessibles depuis l'espace de travail du bibliothécaire, comme illustré par la figure 3.8.

- dépôt des documents
- liste des documents déposer
- liste des étudiants inscrit
- demandes de lecture (demandes d'accès)
- > importation des étudiants interne sur la plateforme

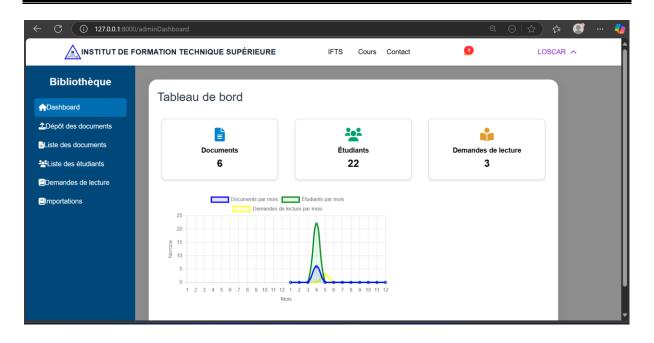


Figure 3.8: Interface du bibliothécaire

Pour accéder à cet espace de travail, le bibliothécaire utilise ses identifiants personnels enregistrés dans la base de données de l'application. Ces informations peuvent être modifiées une fois connecté. La figure 3.9 présente Interface d'authentification du bibliothécaire



Figure 3.9: Interface d'authentification du bibliothécaire

#### III.3.2.1. Dépôt des documents

Cette interface permet l'enregistrement des mémoires des étudiants ayant soutenu.

La figure 3.10 nous montre le formulaire à remplir pour enregistrer les documents de mémoires.

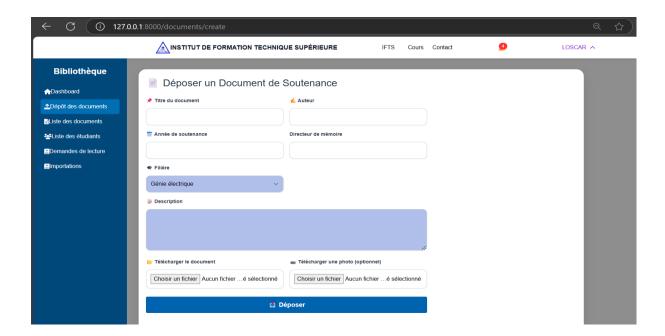


Figure 3.10: Formulaire d'enregistrement des mémoires

# III.3.2.2. Liste des documents déposer

L'interface de la liste des documents déposés affiche les documents ajoutés sur la plateforme par le bibliothécaire. Elle propose plusieurs fonctionnalités, notamment la suppression, la modification des formulaires d'enregistrement des documents, ainsi que le téléchargement des documents déposés. De plus, elle intègre une option de recherche permettant de filtrer les documents par nom, par filière ou par nom du directeur de mémoire.

La figure 3.11 nous montre l'interface de la liste des documents déposer

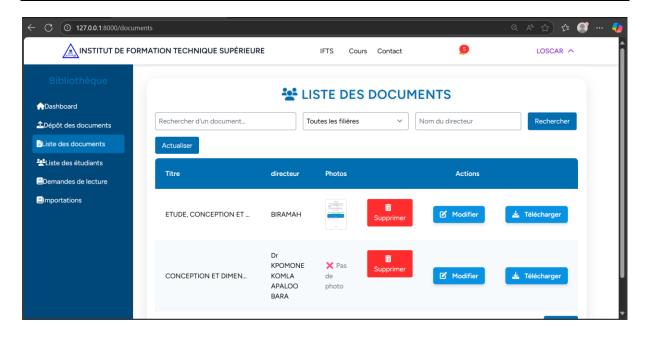


Figure 3.11: interface de la liste des documents déposer

#### III.3.2.3. Liste des étudiants inscrit

Sur cette interface, on dispose de la liste des étudiants utilisant la plateforme, avec la possibilité de supprimer un compte utilisateur. Elle offre également une option de recherche permettant de filtrer les étudiants par nom et par filière.

La figure 3.12 nous montre l'interface de la Liste des étudiants inscrit

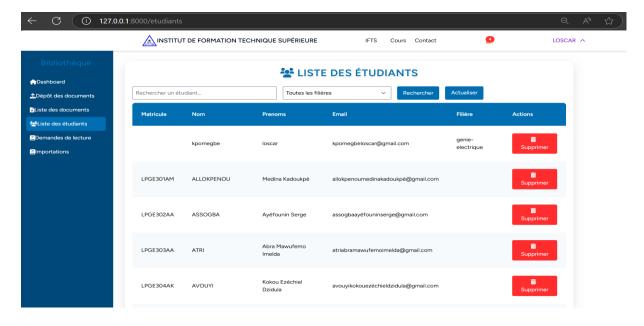


Figure 3.12: Liste des étudiants inscrit

#### III.3.2.4. Demandes de lecture (demandes d'accès)

Pour cette partie, le bibliothécaire a la possibilité d'accepter ou de refuser les demandes de lecture des étudiants externes, avec également une option de recherche permettant d'identifier la personne ayant effectué la demande.

La figure 3.13 nous montre l'interface de la Demandes de lecture

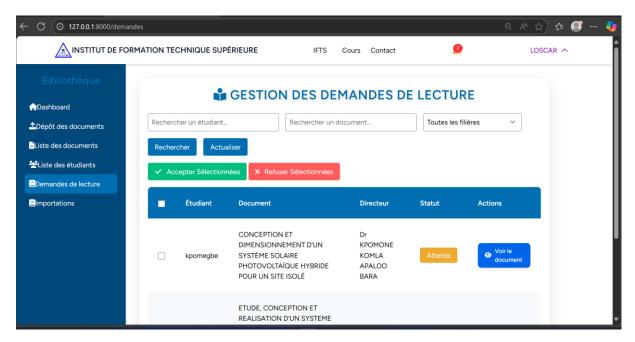


Figure 3.13: Interface de la Demandes de lecture

# III.3.2.5. Importation des étudiants interne sur la plateforme

Cette option permet d'inséré le fichier Excel des étudiants de l'IFTS sur la plateforme comme illustré par la figure 3.14:

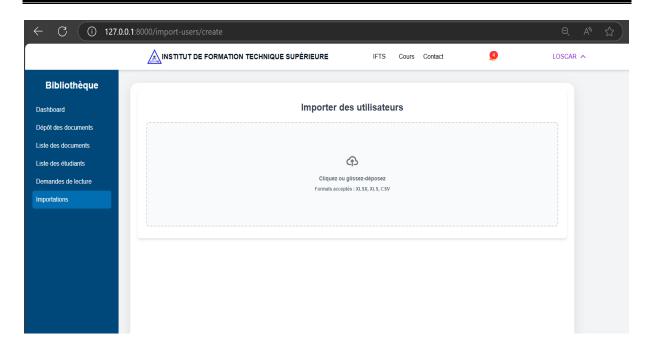


Figure 3.14: Interface d'importation

Les étudiants bénéficient également d'un espace personnel sur la plateforme, leur permettant d'accéder aux fonctionnalités qui leur sont dédiées.

#### III.3.3. Interface d'étudiants

Dès sa connexion, l'étudiant accède à son espace personnel sur la plateforme, où il dispose de deux options : la consultation des documents et la gestion de ses demandes de lecture.

La figure 3.15 nous montre l'interface d'étudiants

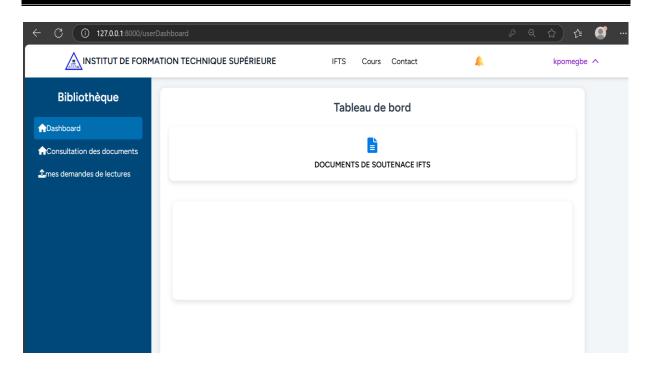


Figure 3.15: Interface d'étudiants

#### III.3.3.1. Consultation des documents

La figure 3.16 montre l'interface de disponibilité des documents.

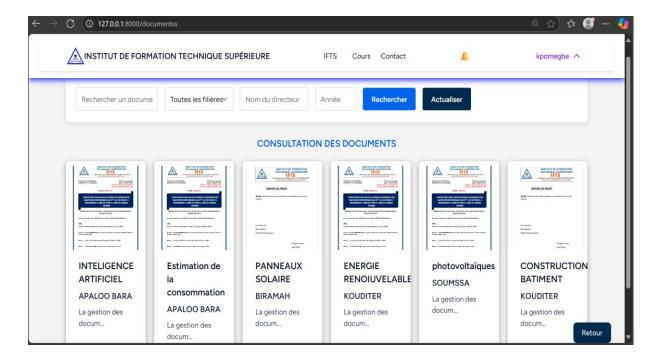


Figure 3.16: Interface documents

#### **Etudiants interne:**

L'étudiant interne a la possibilité de consulter les documents disponibles sur la plateforme sans faire demande de lecture.

La figure 3.17 montre l'interface de consultation des documents et la figure 3.18 montre l'interface de lecture du document.

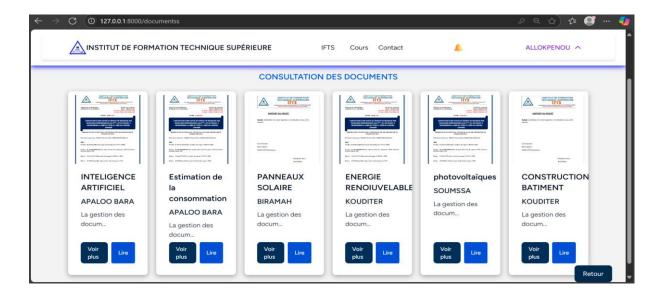


Figure 3.17: Interface de consultation des documents

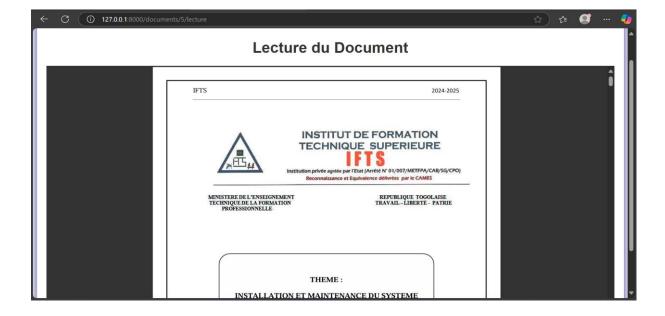


Figure 3.18: Interface de lecture du document