

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة عين تموشنت- بلحاج بوشعيب

Université Ain Témouchent- Belhadj Bouchaib

Faculté des Sciences et de la Technologie

Département de d'électronique et télécommunication



Projet de Fin d'études

Domaine : Sciences et de Technologie

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications Thème :

Conception et réalisation d'une application web pour la gestion des laboratoires de la Faculté des Sciences et Technologies de l'Université d'Ain Témouchent

Realisé par :

- ❖ MATTALLAH Amina Manar.
- ❖ SADDEK Wissame.

Devant le jury composé de :

Mm Chaib.A	MAA UAT.B.B (Ain Temouchent)	Président
Mm Hamlili.H	M C B UAT.B.B (Ain Temouchent)	Encadrant
Dr Yagoub.R	M C B UAT.B.B (Ain Temouchent)	Examinateur

Année universitaire : 2024/2025.

Remerciements

*N*ous tenons à remercier en premier lieu Allah le Tout puissant et le bienveillant, de nous avoir accordés le succès, les bénédictions, la santé et surtout le courage d'accomplir ce travail et que sans lui nous ne serions pas là.

Nous voulons exprimer notre profonde reconnaissance et nos vifs remerciements à notre encadreur Dr. HAMLILI HEYEM pour son aide, ses conseils précieux, ses sacrifices et sa disponibilité.

*U*n grand merci aux membres de jury qui ont accepté d'examiner ce travail :

DR. BENOSMANE.M qui nous a fait l'honneur de présider le jury de soutenance. On exprime nos vifs remerciements au Dr. YAGOUB.R qui nous ont fait l'honneur d'examiner notre travail

*C*e travail n'est pas seulement le fruit de nos efforts ; il doit beaucoup à ceux qui ont contribué à sa réalisation. Nous remercions chaleureusement nos enseignants, les responsables de la faculté des sciences et technologies d'AT.B.B, ainsi que le personnel des laboratoires pour leur aide et leurs encouragements. Enfin, une pensée reconnaissante à tous ceux qui, de près ou de loin, ont participé à notre réussite.

Dédicaces

À tous ceux qui sont chers, ceux à qui je dois mon succès :

À mon père qui n'a pas pu voir mon travail **À la mémoire de mon père**,
qui nous a quittés l'année dernière. Ton amour, tes sacrifices et tes encouragements m'ont
toujours guidé, même dans ton absence. Ce travail est le fruit de tout ce que tu m'as transmis :
la persévérance, l'honnêteté et la foi en mes rêves. Tu restes à jamais présent dans mon cœur
et dans chacun de mes accomplissements. Que ce mémoire soit un humble hommage à ta
mémoire.

À ma mère

Merci d'avoir été là, toujours, avec ton amour, ton courage et ton soutien sans faille.
Tu as porté mes peines, mes doutes, et tu m'as toujours incité à avancer. Je te dois tant.

À mon petit frère

Ta joie de vivre, ta gentillesse et ton regard admiratif m'ont donné la force de continuer.
Tu es une source de motivation, même sans le savoir.

À ma copine Selma

Merci d'avoir été mon repère dans les moments de fatigue et de découragement.
Ta patience, ton amour et tes mots réconfortants ont été précieux tout au long de ce parcours.

À toute ma famille

Merci pour votre amour, vos encouragements et votre présence, même discrète.

À mon binôme

Merci pour ta collaboration, ton sérieux et ta bonne humeur tout au long de ce travail.

Enfin, je dédie ce travail à moi-même, pour avoir tenu bon malgré les obstacles, pour ne pas
avoir abandonné, pour tous les efforts. Je suis fière de ce chemin parcouru, et je me le dois
aussi.

Mattallah Amina Manar

Dédicaces

À mes chers parents

Pour leur soutien et leurs sacrifices, d'avoir mis tout en œuvre pour que je fasse mes études dans les meilleures conditions, et d'avoir répondu mes besoins pendant mon parcours scolaire, secondaire et universitaire.

À mes frères

Merci pour votre soutien indéfectible, votre amour silencieux mais puissant, et votre présence qui m'a toujours donné la force d'avancer. Vous êtes ma fierté, mon ancrage, et ma motivation. Ce mémoire est aussi le vôtre ?

À ma belle-sœur Takia

Merci pour ta bienveillance, ton écoute et ton soutien tout au long de ce parcours. Ta présence chaleureuse et tes encouragements m'ont beaucoup apporté. Je te dédie ces pages avec gratitude et affection.

À mon binôme

Merci pour cette belle aventure partagée, pour ta rigueur, ton engagement et ton esprit d'équipe. Ton soutien constant, ta bonne humeur et ta persévérance ont grandement contribué à la réussite de ce travail. Ce mémoire est le fruit de notre collaboration et je suis fier de l'avoir réalisé à tes côtés.

À moi-même

Pour avoir cru, même quand le doute s'installait. Pour chaque nuit blanche, chaque effort silencieux, chaque pas vers l'inconnu. Pour ne pas avoir abandonné malgré les obstacles. Ce mémoire est aussi une preuve que je suis capable, que je peux aller au bout.

Saddek Wissame

Résumé

La numérisation a profondément transformé la façon dont nous vivons, travaillons et interagissons dans la société actuelle. Elle a révolutionné la manière dont nous accédons à l'information, communiquons et réalisons nos activités journalières. Ce mémoire présente le développement de l'application web "**Lab Organizer**", conçue pour faciliter la gestion des laboratoires à l'Université d'Aïn Témouchent. L'application permet de réserver les salles et les équipements des laboratoires de manière organisée et efficace, contribuant ainsi à améliorer la coordination entre les différents utilisateurs. Elle inclut également un système de gestion des plaintes, permettant de signaler toute anomalie ou dysfonctionnement des ressources disponibles. Lab Organizer se distingue par son interface utilisateur simple et intuitive, facilitant ainsi le processus de réservation et de suivi. Ce travail met en lumière les aspects techniques de l'application et son rôle dans l'amélioration de la gestion des ressources et l'optimisation des opérations au sein des laboratoires de l'université.

Mots clé: Application web, laboratoires, manager, UML, Html, CSS, Java Script, MySQL.

Abstract

Digitalization has profoundly transformed how we live, work, and interact in contemporary society. It has revolutionized the way we access information, communicate, and carry out daily tasks. This thesis presents the development of the web application "Lab Organizer", designed to **facilitate the management of laboratories at Aïn Témouchent University**. The application enables users to **reserve rooms and equipment** in an organized and efficient manner, improving coordination among different stakeholders. It also includes a complaint management system, allowing users to report any issues or malfunctions with available resources. **Lab Organizer** stands out with its simple and intuitive user interface, making the reservation and tracking process easier. This work highlights the technical aspects of the application and its role in enhancing resource management and optimizing operations within the university's laboratories.

Keywords : Web application, laboratory, manager, UML, Html, CSS, Java Script, MySQL.

المخلص

لقد غيرت الرقمنة بشكل عميق الطريقة التي نعيش ونعمل ونتفاعل بها في المجتمع الحالي. كما أحدثت ثورة في كيفية وصولنا إلى المعلومات، وتواصلنا، وإنجازنا لأنشطتنا اليومية. تقدم هذه الأطروحة تطوير تطبيق الويب "Lab Organizer" ، المصمم لتسهيل إدارة المختبرات في جامعة عين تموشنت. يسمح التطبيق بحجز غرف ومعدات المختبرات بطريقة منظمة وفعالة ، مما يساعد على تحسين التنسيق بين المستخدمين المختلفين. ويشمل أيضا نظاما لإدارة الشكاوى، مما يجعل من الممكن الإبلاغ عن أي شذوذ أو خلل في الموارد المتاحة. يتميز Lab Organizer بواجهة المستخدم البسيطة والبديهية ، مما يجعل عملية الحجز والتتبع سهلة. يسلط هذا العمل الضوء على الجوانب الفنية للتطبيق ودوره في تحسين إدارة الموارد وتحسين العمليات داخل مختبرات الجامعة.

الكلمات المفتاحية: تطبيق ويب، مخابر، إدارة، UML, Html, CSS, Java Script, MySQL.

Table des matières

INTRODUCTION GENERALE.....	1
----------------------------	---

CHAPITRE I : LES TECHNOLOGIES WEB

I.1 Introduction.....	4
I.2 Présentation du projet	4
I.2.1 L'objectif du projet	4
I.2.2 Étude et critique de l'existant	5
I.2.3 Solution proposée.....	5
I.3 Les technologies web.....	6
I.3.1 Les sites et pages web	7
Site web statique.....	7
Site web dynamique	8
I.3.2 Les applications web.....	9
a. Le fonctionnement d'une application web	10
I.3.3 L'architecture des applications web :.....	11
I.3.4 L'application client/serveur :	12
I.3.5 Les serveurs web.....	14
I.3.6 Les serveurs d'applications.....	14
I.3.7 L'hébergement d'une application web.....	15
I.4 Conclusion	15

CHAPITRE II : CHOIX DES TECHNOLOGIES

II. 1 Introduction	17
II.2 Choix des technologies.....	17
II.3 Présentation des langages utilisées	18
II.3.1 HTML.....	18
II.3.1.1 Définition du HTML	18
II.3.1.2 La structure d'une page HTML	18
II.3.1.3 Fonctionnement de HTML	20
II.3.1.4 Les formulaires HTML.....	20
II.3.2 CSS (Cascading Style Sheets)	21
II.3.2.1 Définition du CSS.....	21
II.3.2.2 La structure d'une page CSS	22
II.3.2.3 Fonctionnement de CSS	22
II.3.3 JavaScript.....	23

II.3.3.1 Définition du java script	23
II.3.3.2 La structure de java script.....	24
II.3.3.3 Le fonctionnement du java script	24
II.3.4 PHP	24
II.3.4.1 Définition du PHP	24
II.3.4.2 La structure d'une page PHP	25
II.3.4.3 Fonctionnement de PHP	25
II.4. Base de données SQL (Structured Query Language).....	26
II.4.1 Définition du SQL	26
II.4.2 Fonctionnement de base de données SQL.....	26
II.5. Outils de développement	27
II.5.1 Serveur Web Apache	27
II.5.2 Serveur MySQL.....	28
II.5.2.1 Fonctionnalités de MYSQL.....	28
II.5.3 Wamp.....	29
II.6 Conclusion	29

CHAPITRE III : CONCEPTION ET RESULTATS

III.1 Introduction	31
III.2 Le langage UML	31
III.3 Modélisation.....	32
III.3.1 Diagramme de cas d'utilisation (use case).....	32
III.3.2 Les acteurs et leurs rôles	33
III.3.3 Diagramme de classe.....	33
III.3.4 Diagramme de séquence.....	34
III.3.5 Schéma relationnelle	38
III.4 Les interfaces d'application web.....	39
III.4.1 Interface de d'inscription :	39
III.4.2 Interface de connexion	40
III.4.3 Interface de l'Espace utilisateur	41
III.4.4 Interface menu principale.....	41
III.4.5 Interface de réservation	42
III.4.6 Interface des plaintes.....	45
III.4.7 Interface de répondre aux plaintes.....	45
III.5 Conclusion.....	46

Conclusion générale	47
Bibliographie	48

LISTE DES FIGURES

CHAPITRE I :

Figure I.1 : Schéma application client/serveur.....	13
Figure I.2 : Le processus de communication entre le serveur et le navigateur web.....	13

CHAPITRE II :

Figure II.3 : Structure d'une page HTML.	19
Figure II.4 : Exemple d'un formulaire HTML.	21
Figure II.5 : Structure d'une page.	22
Figure II.6 : Structure d'une page JavaScript.	24
Figure II.7 : Structure d'une page PHP	25

CHAPITRE III:

Figure III.8 : Diagramme de cas d'utilisation.	32
Figure III.9 : Diagramme de class.	34
Figure III.10 : Diagramme de séquence (étudiant).....	35
Figure III.11 : Diagramme de séquence (enseignant).....	36
Figure III.12 : Diagramme de séquence (poser une plainte).	37
Figure III.13 : Diagramme de séquence (répondre aux plaintes).	38
Figure III.14 : Schéma relationnel.	39
Figure III.15 : Interface d'inscription.	40
Figure III.16 : Interface de connexion.	41
Figure III.17 : Interface des espaces.	41
Figure III.18 : Menu principale pour l'espace enseignant et étudiant.	42
Figure III.19 : Interface de réservation.	43
Figure III.20 : Interface de réservation des équipements.	43
Figure III.21 : Interface de réservation des salles.....	44
Figure III.22 : Interface d'accès refusée pour la réservation tant qu'étudiant.....	44
Figure III.23 : Interface de liste des réservations.	45
Figure III.24 : Interface des plaintes.....	45
Figure III.25 : Interface répondre aux plaintes (ingénieure).	45

LISTE DES TABLEAUX

Tableau I.1 : Les catégories des technologies web..... 6

Tableau II.2 : Les fonctionnalités de MYSQL 28

Acronymes

A

API Application Programming Interface

ASP Active Server Pages

C

CMS Content Management System

D

DCL Data Control Language

DDL Data Definitin Language

DML Data Manipulation Language

DQL Data Query Language

F

FTP File Transfer Protocol

H

HTTP Hypertext Transfer Protocol

I

IIS Internet Information Services

IU Interface utilisateur

IP Internet Protocol address

J

JSON JavaScript Object Notation

M

MYSQL	My Structured Query Language
N	
NAK	Negative Acknowledgment
P	
PSTC	Protocole simple de transfert de courrier
S	
SEO	Search Engine Optimization
SGBD	Système de gestion de base de données
SSL	Secure Socket Layer
T	
TCL	Transaction Control Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
U	
UML	Unified Modelling Language
X	
XML	EXtensible Markup Language

INTRODUCTION GENERALE

Avec l'essor des technologies numériques et l'évolution des besoins des établissements universitaires, la gestion des laboratoires nécessite une modernisation urgente. Ces espaces, essentiels à l'enseignement et à la recherche pratique, requièrent une organisation rigoureuse : réservation et suivi des équipements scientifiques, gestion des incidents techniques, etc. Pourtant, dans de nombreuses institutions, ces tâches s'effectuent encore manuellement ou à l'aide d'outils inadaptés, entraînant des problèmes de coordination, des conflits de planning et une perte de temps considérable.

Pour pallier ces limites, nous proposons dans ce travail LAB ORGANIZER, une application web dédiée exclusivement à l'université d'Ain Témouchent. Cette plateforme centralise toutes les activités liées aux laboratoires : réservations, consultation des équipements disponibles, suivi des incidents signalés et amélioration de la communication entre les utilisateurs (enseignants, étudiants, ingénieurs).

Ce travail s'articule autour de deux axes principaux une analyse des besoins et des fondements théoriques et la conception et la réalisation de l'application. Notre mémoire est structuré de la manière suivante :

Le premier chapitre analyse d'abord les besoins des laboratoires universitaires et les limites des solutions actuelles, puis présente notre application Lab Organizer comme réponse. Il introduit ensuite les bases théoriques du développement web nécessaires à sa réalisation, en couvrant l'architecture client-serveur et les technologies clés.

Le chapitre 2, Détaille les technologies retenues pour la réalisation de notre application, en expliquant les choix techniques retenus pour le développement, expliquant l'utilisation des langages front-end (HTML/CSS/JS), back-end (PHP) et base de données (MySQL) ainsi que les outils utilisés lors du développement.

Le troisième chapitre présente la phase concrète du projet à travers : la conception modélisée avec les diagrammes UML clés (cas d'utilisation, classes et séquences) et la réalisation effective illustrée par les interfaces fonctionnelles et les mécanismes techniques implémentés.

Grâce à ce projet, nous sommes impatients de proposer une solution spécifique et extensible pour améliorer la gestion des laboratoires universitaires et illustrer la contribution des technologies Web dans la transformation numérique des infrastructures éducatives.

CHAPITRE I : LES TECHNOLOGIES WEB

I.1 Introduction

Dans un monde où la technologie évolue à un rythme effréné, les applications web se sont imposées comme des outils incontournables pour les entreprises et les particuliers. Combinant flexibilité, performance et disponibilité, elles permettent d'accéder à des fonctionnalités innovantes directement depuis un navigateur. Qu'il s'agisse de gestion de projets, de communication, de divertissement ou d'e-commerce, ces solutions Cloud répondent à des besoins variés, accessibles sur tous les appareils connectés à Internet. Une véritable révolution, qui redéfinit notre rapport au numérique et rend les technologies web indispensables au quotidien.

(1)

Dans ce chapitre nous allons commencer par l'exposé de notre projet avant d'introduire les concepts clés du développement web.

I.2 Présentation du projet

Le laboratoire universitaire est une partie essentielle de la formation académique, car il offre aux étudiants un espace privilégié pour développer leurs compétences pratiques et expérimenter leurs connaissances théoriques, et il est donc nécessaire qu'il soit conforme à toutes les normes de base, telles que la gestion des équipements, des stocks et des réservations. Cependant, les méthodes classiques sont souvent peu efficaces. Elles peuvent causer des erreurs, faire perdre du temps et créer un désordre.

Face à ces défis, notre projet, l'application Lab Organizer, a été développé comme une solution numérique complète. Il vise à améliorer et simplifier la gestion des laboratoires universitaires grâce à une application web intuitive et efficace.

I.2.1 L'objectif du projet

L'objectif principal de ce projet est de concevoir et développer une application web (Lab Organizer) visant à optimiser la gestion des laboratoires universitaires. Cette solution cherche à :

- Centraliser les données (équipements, réservations, utilisateurs) sur une plateforme unique.
- Automatiser les processus manuels (gestion des stocks, planning et maintenances) pour réduire les erreurs et gagner du temps.

- Faciliter l'accès aux informations pour les chercheurs, les enseignants et les techniciens via une interface intuitive et responsive.

En fin Lab Organizer vise à transformer la gestion traditionnelle des laboratoires en un système numérique fiable et adapté aux besoins des établissements académiques.

I.2.2 Étude et critique de l'existant

De nos jours, la numérisation est intégrée dans de nombreux domaines et secteurs, y compris la gestion des laboratoires. Ils proposent plusieurs solutions avec des fonctionnalités diverses, comme OpenLIMS, Quartz et LabArchives. Cependant, le souci avec ces plateformes, c'est qu'elles ont des limites :

- Des interfaces compliquées et difficiles à utiliser.
- Des frais d'abonnement élevés par rapport certaines institutions académiques avec des budgets limités.
- Un manque de souplesse pour satisfaire les besoins particuliers des laboratoires universitaires qui nécessitent davantage de personnalisation.
- Intégration restreinte avec les systèmes universitaires en place, comme la gestion des étudiants, les bibliothèques et les plateformes d'apprentissage.

I.2.3 Solution proposée

Notre application, Lab Organizer, est une solution avec une approche plus accessible et adaptée aux universités avec :

- Une accessibilité sur plusieurs plateformes (ordinateurs, tablettes, smartphones) pour permettre aux utilisateurs de contrôler les laboratoires à distance.
- Une interface conviviale facilite l'expérience des utilisateurs.
- Un système amélioré de gestion des équipements et stocks, offrant un suivi précis de leur état et disponibilité.
- Une réservation efficace et facile des équipements, à l'aide d'un accès dédié aux enseignants.
- Un tableau de bord synthétique présentant les informations essentielles : état des équipements, ressources disponibles et réservations en cours.

I.3 Les technologies web

La technologie web englobe tous les outils, protocoles et langages de programmation nécessaires pour concevoir, déployer et interagir avec des sites et des applications web. Elles constituent le fondement des expériences numériques modernes, depuis la simple consultation de pages web jusqu'aux applications interactives les plus complexes. En 2025, ces technologies ont évolué pour intégrer des fonctionnalités avancées comme la communication instantanée, l'intelligence artificielle (IA) et les réseaux décentralisés. (2)

Le terme "technologies Web" englobe plusieurs éléments :

- Les langages de balisage comme HTML.
- Les langages de feuilles de style comme CSS.
- Langages de script côté client comme JavaScript.
- Langages de script côté serveur comme PHP, Ruby, Python, etc...
- Les technologies de base de données comme MySQL. (3)

Les technologies Web peuvent être regroupées en trois catégories selon leurs fonctions principales et leur position dans le développement Web. Comme indiqué dans le tableau I.1 :

Tableau I.1 : Les catégories des technologies web.

Technologies front-end	Elles s'occupent de la partie visible des applications web. Comprennent HTML, CSS, JavaScript et des Framework liés. Technologies de back-end.
Technologies back-end	S'occupent de la logique sur le serveur et du traitement des données. Comprennent les langages Python, PHP, Ruby, Node.js, les bases de données et les API du côté du serveur.
Technologies de base de données	S'occupent du développement à la fois du front-end et du back-end. Comprennent MySQL, PostgreSQL, MongoDB

I.3.1 Les sites et pages web

Une page web est un document numérique accessible via sur Internet, un fichier rédigé en HTML pouvant intégrer des images, des sons et/ou des vidéos. Une des caractéristiques des pages web est qu'elles n'ont pas de limite et peuvent contenir une quantité infinie d'informations. La seule contrainte est le poids de la page et le temps nécessaire pour qu'elle se charge. (4)

Chaque page web est accessible grâce à une adresse, appelée URL (positionnement unifié des ressources), en entrant cette URL dans la barre de navigation de votre navigateur, ce dernier envoie une requête au serveur d'hébergement afin de transmettre alors différents éléments constitutifs de la page. Votre navigateur utilise ensuite le code HTML, CSS et JavaScript pour interpréter et afficher la page de votre écran. (5)

Un site Web, souvent désigné comme site Internet, est un ensemble de pages Web qui sont connectées entre elles par des liens hypertextes, et peuvent être consultées via une même adresse URL.

Les sites Web peuvent remplir diverses fonctions : afficher des informations, vendre des articles, offrir des services, et bien plus encore. (6)

Les sites web sont divisés en deux types :

Site web statique

Un site statique est un site web créé uniquement avec les langages HTML et CSS une fois pour toutes. En effet l'administrateur du site compose avec un éditeur HTML des pages web stockées sur le serveur web, celui-ci renvoie ces pages à la demande au visiteur. Il fonctionne parfaitement, mais leur contenu ne peut pas être mis à jour automatiquement. L'administrateur, qui est le propriétaire du site, doit modifier le code source pour ajouter de nouvelles informations à l'aide d'un éditeur HTML. (4)

Les avantages des sites web statiques (7)

Rapidité et efficacité : les sites web statiques offrent une meilleure vitesse et réactivité, puisqu'ils ne requièrent ni traitement côté serveur ni appels de base de données pour présenter les pages web. Le serveur envoie directement les pages au navigateur de l'utilisateur, ce qui réduit le temps de chargement.

Stabilité : en minimisant le nombre de composants mobiles, les sites web statiques offrent une plus grande stabilité et sont moins exposés aux défaillances d'exécution souvent associées aux scripts côté serveur ou aux soucis de base de données.

Sécurité : les sites web statiques offrent un haut degré de sûreté, étant donné qu'ils ne s'appuient pas sur une base de données, où des vulnérabilités peuvent fréquemment être mises à profit. En outre, l'absence de scripts de traitement diminue la vulnérabilité potentielle.

Capacité d'hébergement et souplesse d'évolution : les sites internet statiques nécessitent peu de contraintes d'hébergement et peuvent être aisément installés sur n'importe quelle plateforme web. Ils s'ajustent aussi aisément à des charges lourdes, parce que la diffusion de contenu statique demande moins de moyen serveur.

Entretien : sans avoir besoin de bases de données ou de scripts côté serveur, la maintenance est grandement facilitée. Les ajustements se font généralement par l'édition des fichiers HTML suivie de leur réimportation sur le serveur.

Profitabilité : Les sites web statiques tendent à être moins onéreux à concevoir (Pas de back-end ni de base de données à configurer) et à loger (Options gratuites ou low-cost GitHub Pages, Netlify), étant donné qu'ils requièrent moins de ressources.

Site web dynamique

Un site Web dynamique est une collection de pages avec du contenu qui varient en fonction de profil du visiteur (géolocalisation, langue, historique), ses interactions (panier d'achat, connexion) et des données externes (météo, API, fuseau horaire). (8)

Un site Web dynamique est réalisé en utilisant un code complexe qu'un site Web statique (tel que PHP ou ASP) et a un niveau de fonctionnalité plus important.

En effet, de nombreux sites Web dynamiques peuvent être contrôlés par un système de gestion de contenu (CMS) permettent à des utilisateurs sans compétences techniques avancées de modifier facilement le contenu d'un site dynamique.

Chaque page d'un site Web dynamique est générée à partir d'informations stockées dans une base de données ou d'une source externe. Le système de gestion de contenu que vous pouvez utiliser pour maintenir votre site Web modifie directement ces informations stockées. (9)

Les avantages d'un site dynamique :

La plupart des services de conception Web personnalisés ont une conception Web dynamique pour leurs clients grâce aux avantages sans fin que nous citons :

Suivre le rythme de la concurrence du marché : Aucune entreprise ne peut se développer sans suivre la vitesse de la conversion technologique. S'ils échouent, après tout, ils risquent d'être interdits. De la même manière, un site Web est un avantage inestimable pour toute entreprise, car ce qu'il lui donne une forte présence en ligne. S'il ignore la tendance des sites Web dynamiques, les clients peuvent passer aux concurrents.

Améliorer le SEO : Les sites dynamiques constituent une carte stratégique pour le référencement grâce à leur capacité à produire un contenu abondant et parfaitement structuré. Leur architecture technique optimisée, avec des URLs propres et un temps de chargement minimal, répond idéalement aux exigences des moteurs de recherche comme Google. En générant automatiquement des pages hautement pertinentes et actualisées, ils élargissent considérablement leur visibilité organique, permettant ainsi d'attirer un trafic mieux ciblé et plus qualifié.

Convertir les visiteurs en clients potentiels : Les sites dynamiques proposent des recommandations personnalisées, des outils interactifs pratiques, et un processus d'achat simplifié. Ces fonctionnalités rendent l'expérience plus agréable et convaincante, ce qui augmente concrètement le nombre de ventes ou d'inscriptions. (9)

I.3.2 Les applications web

Les applications web sont des logiciels qui fonctionnent sur un serveur web et qu'on peut utiliser à travers un navigateur. Les applications web sont souvent créées avec des langages de programmation comme JavaScript, Symfony, Laravel ou Vue.js. (10)

À la différence d'un logiciel classique, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application en utilisant son navigateur préféré. La tendance actuelle consiste à fournir une expérience utilisateur et des fonctionnalités similaires à celles des logiciels installés directement sur les ordinateurs. Les outils utilisés pour concevoir des applications web sont identiques à ceux utilisés pour réaliser des sites internet. (11)

a. Le fonctionnement d'une application webClient/serveur :

Une application web opère selon le modèle d'architecture client-serveur. L'utilisateur, souvent un simple navigateur web, adresse des sollicitations au serveur afin d'exploiter les diverses fonctionnalités de l'application. Le serveur gère ces demandes et retourne les réponses appropriées au client.

Technologies web :

On a généralement recours aux technologies web comme HTML, CSS et JavaScript pour développer des applications web. HTML (HyperText Markup Language) est employé pour établir la structure et le contenu d'une page web. CSS (Cascading Style Sheets) est employé pour établir la mise en forme et l'apparence de la page. On utilise JavaScript pour intégrer des fonctionnalités interactives sur la page.

Avec l'évolution constante des technologies web, les applications web ont gagné en puissance et complexité. À l'heure actuelle, on peut concevoir des applications web qui se mesurent aux applications classiques en matière de fonctionnalités et de performance.

Architecture côté serveur :

Sur le plan du serveur, les applications web peuvent être construites sur diverses architectures. L'architecture monolithique et l'architecture fondée sur les micro services sont les deux structures les plus fréquemment utilisées.

Dans une structure monolithique, toutes les caractéristiques de l'application sont consolidées en un unique ensemble. Cela indique que l'ensemble des composants de l'application est mis en œuvre simultanément et utilise des ressources communes. Cette méthode est facile à appliquer, cependant elle peut engendrer des problèmes de maintenance et d'extension.

Dans une architecture fondée sur les micro services, l'application est segmentée en plusieurs services autonomes. Chaque service est chargé d'une fonctionnalité particulière de l'application. Ces services interagissent entre eux grâce à des API (Application Programming Interface) afin de fonctionner en synergie. Cette méthode offre une flexibilité et une scalabilité accrues, mais exige une gestion plus sophistiquée.

Stockage des données :

Il est également nécessaire que les applications web conservent et administrent les données des utilisateurs. On dispose de diverses alternatives pour conserver les données, comme les bases de données relationnelles, les bases de données NoSQL et le stockage en nuage.

Les bases de données relationnelles s'appuient sur le modèle de données relationnel, en exploitant des tables pour organiser les informations. Elles sont conçues pour les applications qui nécessitent des requêtes sophistiquées et des relations entre les données.

Les bases de données NoSQL adoptent un schéma de données non relationnel. Elles conviennent aux applications qui requièrent une grande capacité d'évolution et une flexibilité dans l'organisation des données.

Le cloud computing offre la possibilité de conserver les informations de l'application sur des serveurs éloignés administrés par des prestataires de services en nuage tels qu'Amazon Web Services, Google Cloud ou Microsoft Azure. Ceci procure une flexibilité et une évolutivité considérables, tout en diminuant les dépenses associées à la gestion des serveurs. (11)

I.3.3 L'architecture des applications web :

L'architecture d'une application web désigne la façon dont une application web est structurée et organisée. Elle comprend comment ces différents composants, modules et sous-systèmes sont organisés pour travailler ensemble et offrir les fonctions attendues de l'application, ainsi que leurs interactions entre eux et avec des systèmes externes. Cela assure un fonctionnement efficace, ainsi qu'une bonne évolutivité et une facilité de maintenance. À un niveau basique, l'architecture d'une application web se divise généralement en deux parties principales : le côté client et le côté serveur.

Le côté client :

Interface utilisateur (UI) : Cette partie comprend les éléments visuels que les utilisateurs manipulent, comme les pages web, les formulaires, les boutons et les menus. Ces composants sont enrichis par le scripting client, exécuté directement dans le navigateur via des langages comme JavaScript. Ce dernier permet de rendre l'interface dynamique et interactive par exemple en validant des formulaires en temps réel ou en actualisant partiellement une page.

Pour développer ces fonctionnalités complexes, des frameworks frontend comme React, Angular ou Vue.js offrent des structures et des bibliothèques qui aident à créer des applications complexes côté client plus facilement. (12)

Le côté serveur :

La logique d'application exécutée côté serveur, constitue le cœur métier du système. Elle gère les processus critiques : traitement des données, authentification des utilisateurs, et application des règles métier (ex : calcul de remises). Les langages et frameworks de développement côté serveur les plus utilisés incluent Node.js, Python (Django, Flask), Ruby on Rails et Java (Spring Boot) accélèrent son développement tout en intégrant des couches de sécurité.

Pour le stockage un système de gestion de base de données (SGBD) assure la persistance des données. Il est important que les applications web puissent stocker et retrouver des informations facilement, ce qui nécessite un stockage de données durables. Les bases de données relationnelles comme MySQL et PostgreSQL, ainsi que les bases de données NoSQL comme MongoDB et Redis, proposent plusieurs choix qui conviennent à des besoins particuliers. Elles peuvent être utilisées pour stocker des données qui sont bien organisées ou des données qui ne le sont pas.

Enfin, les Serveurs web les serveurs web agissent comme des intermédiaires entre les demandes des utilisateurs et la logique de l'application. Ils gèrent les demandes HTTP qui arrivent et fournissent les réponses nécessaires. Nginx, Apache et Microsoft Internet Information Services (IIS) sont parmi les serveurs web les plus couramment utilisés. (12)

Le protocole de communication :

Le protocole de transfert hypertexte (HTTP) et sa version sécurisée (HTTPS) forment la fondation de la communication entre les clients et les serveurs. HTTPS, qui utilise le cryptage SSL/TLS, sécurise l'envoi de données sensibles et augmente la confiance des utilisateurs. (12)

I.3.4 L'application client/serveur :

Le modèle client-serveur représente une architecture fondamentale où deux entités distinctes interagissent (**la figure I.1**) : le client, qui initie les requêtes, et le serveur, chargé de les traiter et d'y répondre. Dans ce système, les clients (comme les navigateurs web ou les applications mobiles) envoient des demandes au serveur pour accéder à des ressources ou services.

Le serveur, constamment en attente de ces sollicitations, reçoit chaque requête via un port réseau dédié, exécute les traitements nécessaires (accès aux bases de données, calculs, etc.), puis renvoie une réponse appropriée au client. Cette communication s'établit généralement via le protocole HTTP, suivant un principe sans état (stateless) où chaque requête est traitée indépendamment des précédentes. Ce modèle hiérarchisé permet une répartition claire des rôles et une grande efficacité dans les échanges réseau.

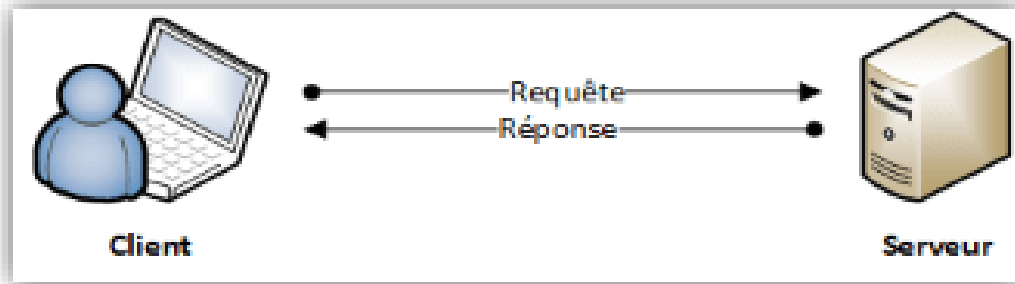


Figure I.1 : Schéma application client/serveur.

La figure I.2 illustre le processus d'accès à une page web à travers l'architecture client-serveur. Lorsqu'un utilisateur saisit une URL (par exemple `http://www.example.com/index.html`) dans son navigateur, ce dernier traduit cette adresse en une requête HTTP qu'il envoie au serveur web. Le serveur interprète alors la requête, récupère les fichiers correspondants (comme le fichier `index.html` et ses ressources associées), puis renvoie une réponse HTTP contenant un code de statut et les données de la page. Le navigateur reçoit cette réponse, adapte les données reçues pour construire la page web finale, et enfin affiche le résultat à l'utilisateur. Ce schéma met en évidence les interactions entre les trois composants principaux : l'utilisateur, le navigateur (client) et le serveur web.

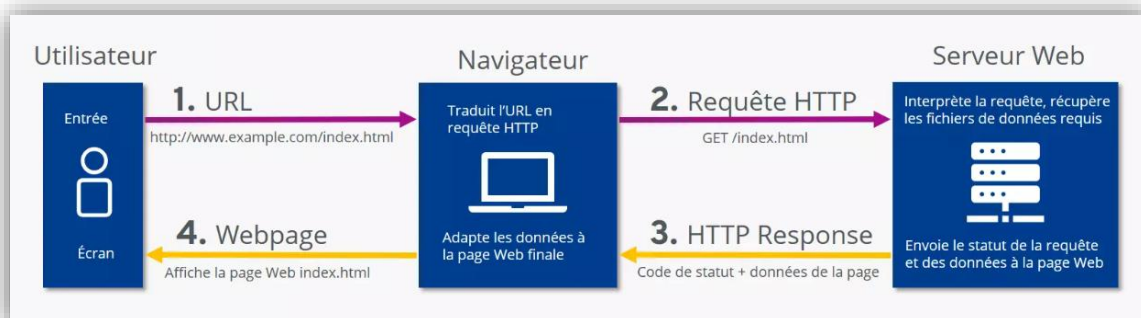


Figure I.2 : Le processus de communication entre le serveur et le navigateur web.

I.3.5 Les serveurs web

Un serveur Web, dans sa formulation la plus élémentaire, est un ordinateur spécifiquement élaboré pour conserver et diffuser des documents Web aux navigateurs des usagers. Ces fichiers contiennent les documents HTML, CSS, images, vidéos et d'autres ressources indispensables à la présentation d'une page Web. Deux composants majeurs constituent un serveur Web : (13)

Éléments matériels : ce sont les machines connectées à Internet permettant l'échange de divers types de fichiers (documents HTML, images, scripts JavaScript et feuilles de style CSS). Ces serveurs physiques hébergent simultanément le logiciel serveur nécessaire à leur fonctionnement. (14)

Éléments logiciels particulièrement le serveur HTTP, forment la couche intelligente du système : ils assurent la médiation entre les clients et le serveur en interprétant les requêtes HTTP des navigateurs, en traitant ces demandes, et en générant les réponses appropriées. (13)

Ce processus permet aux serveurs web de récupérer et de transmettre la page web sollicitée à l'application du navigateur, comme Google Chrome. Les serveurs web font aussi usage des protocoles SMTP (Protocole simple de transfert de courrier) et FTP (Protocole de transfert de fichiers) pour gérer les fichiers relatifs aux courriels ou au stockage. (14)

I.3.6 Les serveurs d'applications

Un serveur d'applications est un système logiciel qui crée un espace d'exécution et de gestion des applications. Il sert de lien entre les utilisateurs finaux (ou clients) et les systèmes back-end ou bases de données. Sa fonction principale est d'héberger, de contrôler et de fournir des applications aux utilisateurs via un réseau. Il agit comme un pont, permettant aux différentes parties d'une application, comme les serveurs web, les bases de données et la logique métier, de communiquer et de fonctionner ensemble. (15)

Le serveur d'applications s'occupe de toutes les actions entre l'interface utilisateur et le back-end, veillant à ce que le client n'interagisse pas directement avec la base de données. Cela rend l'application plus sûre, adaptable et capable de grandir, tout en offrant un accès contrôlé qui diminue le risque d'erreurs ou de problèmes de sécurité. (16)

I.3.7 L'hébergement d'une application web

L'hébergement d'applications signifie sauvegarder et exécuter des logiciels sur des infrastructures distantes (serveurs dédiés ou plateformes cloud). Ce qui permet d'y accéder en ligne. Héberger une application consiste à transférer des composants applicatifs - code source, bases de données, fichiers multimédias et ressources système - vers un environnement serveur géré par un fournisseur spécialisé. Ce fournisseur fait fonctionner votre application et la rend disponible pour les utilisateurs via différents canaux : navigateurs web, applications mobiles ou interfaces API. L'hébergement d'applications permet aux développeurs de rendre leurs applications disponibles à un plus grand nombre de personnes, car les utilisateurs peuvent y accéder de n'importe où avec une connexion internet. (17)

I.4 Conclusion :

Dans ce chapitre nous avons parlé des applications web qui représentent aujourd'hui un pilier incontournable du paysage numérique, alliant accessibilité, flexibilité et innovation. Leur capacité à s'adapter aux besoins des utilisateurs et des entreprises, tout en intégrant en permanence de nouvelles technologies, en fait un moteur essentiel de la transformation digitale.

L'avenir des applications web s'annonce aussi dynamique qu'indispensable. C'est précisément cette puissance d'innovation et cette accessibilité universelle qui nous ont conduits à numériser les laboratoires de l'université via une application web. Cette solution moderne offre une plateforme centralisée, accessible à tout moment et depuis n'importe quel appareil, tout en permettant des mises à jour continues pour s'adapter aux besoins évolutifs de la recherche et de l'enseignement. Ce choix stratégique illustre parfaitement comment les applications web transforment concrètement nos méthodes de travail et d'apprentissage.

CHAPITRE II : CHOIX DES TECHNOLOGIES

II. 1 Introduction

Dans ce chapitre, nous présentons les différentes technologies et outils mobilisés dans la réalisation de *Lab Organizer*, une application web dédiée à la gestion des laboratoires universitaires. Avant de détailler nos choix techniques, rappelons que le développement web s'articule autour de trois approches principales : le front-end (développement côté client), le back-end (développement côté serveur) et le full-stack (combinaison des deux).

Le front-end consiste en la création de l'interface utilisateur, via des langages tels que HTML, CSS et JavaScript. Le back-end, quant à lui, gère le logique métier et les données, en utilisant des technologies comme Python, PHP ou Node.js. Enfin, le développement full-stack permet à un seul développeur de prendre en charge l'ensemble des couches de l'application.

Nous détaillerons ici nos choix technologiques, notamment le framework, les langages de programmation et le système de base de données retenus, tout en expliquant leur rôle dans l'architecture globale du système, ainsi que les outils d'accompagnement utilisés lors du développement. (18)

II.2 Choix des technologies

a. WampServer

Nous avons choisi WampServer comme environnement de développement pour notre application, et ce, pour plusieurs raisons :

- **Facilité d'installation** : sa configuration est intuitive et rapide.
- **Compatibilité** : Il intègre nativement **PHP, Apache et MySQL**, des composants essentiels à notre projet.
- **Dynamisme et ressources** : Une large communauté et une abondante documentation en ligne facilitent la résolution de problèmes et l'apprentissage.

b. Langages utilisés

Le développement de notre application repose sur une **combinaison de langages complémentaires**, permettant une séparation claire entre :

- **La structure** (HTML).
- **Le design** (CSS).
- **Les interactions côté client** (JavaScript).
- **La logique serveur** (PHP).

Cette approche **modulaire** assure une maintenance simplifiée et une évolutivité optimale.

c. Base de données MySQL

MySQL s'est imposé comme une solution évidente pour notre application web, grâce à ses avantages clés :

- **Administration simplifiée** : Via des outils comme **phpMyAdmin**, la gestion des données est accessible même aux non-experts.
- **Performance adaptée** : Idéal pour des applications de **taille moyenne**, comme Lab Organizer.
- **Gestion des accès** : Son système de permissions sécurisé convient parfaitement à une application multi-utilisateur.
- **Fiabilité et compatibilité** : Son intégration native avec PHP garantit des échanges fluides entre le back-end et la base de données.

II.3 Présentation des langages utilisées

II.3.1 HTML

II.3.1.1 Définition du HTML

HTML, ou « Hyper Text Markup Language », est un langage de balisage utilisé pour structurer et organiser le contenu d'une page web via des balises (ou tags). Les balises servent à montrer comment le document doit être présenté et quels liens il crée avec d'autres documents. Le langage HTML permet de lire des documents sur Internet depuis différents appareils. Cela se fait grâce au protocole HTTP, qui facilite l'accès aux documents via le réseau en utilisant une adresse URL. L'HTML n'est pas vraiment un langage de programmation comme on l'entend habituellement. C'est plutôt un ensemble de règles qui montrent à un navigateur comment présenter une page web. Pour enrichir les fonctionnalités et la mise en forme, HTML est systématiquement combiné avec des langages de programmation comme JavaScript et des formats de présentation comme les feuilles de style en cascade. (19)

II.3.1.2 La structure d'une page HTML

On prend un code correspond à la base d'une page Web en HTML (Structure générale D'un document HTML **la figure II.3**) :

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
</head>
<body>
  <h1> This is a heading</h1>
  <p>This is a paragraph</p>
</body>
</html>
```

Figure II.3: Structure d'une page HTML.

Le `<!DOCTYPE html>` est une déclaration (et non une balise HTML) qui se situe au tout début d'un document HTML. Elle signale au navigateur que le document est en HTML.

La balise `<html>` :

C'est l'élément racine qui structure tout le document HTML. Elle encapsule l'intégralité du contenu de la page web. Comme le montre le code, sa balise fermante correspondante `</html>` apparaît à la toute fin du document, marquant la clôture de l'ensemble.

Un document HTML est structuré en deux parties : en-tête (Head) et corps (body). Chaque partie est repérée par des balises.

L'en-tête `<Head>`

Cette partie renferme les métadonnées et les détails techniques qui ne sont pas perceptibles directement sur la page, mais qui sont indispensables pour un fonctionnement optimal et une optimisation efficace. Comprend des éléments comme :

- Le titre de la page `<title>` affiché dans l'onglet du navigateur.
- L'encodage des caractères `<meta>` pour une correcte gestion des accents et symboles.
- Les liens vers les feuilles de style (CSS) ou scripts (JavaScript).

La balise `<body>` : le contenu visible de la page

Tous les éléments principaux de la page web sont inclus dans la balise `<body>`. À l'inverse de l'en-tête (`<head>`), ce qui est inscrit ici apparaît directement à l'écran pour les visiteurs. C'est dans cette partie que se situe la majorité du code HTML. Les éléments clés du `<body>`.

Les listes : HTML distingue les listes non ordonnées () des listes ordonnées(), en fonction du fait que l'ordre formel du contenu dans le code représente une information ou non. Il existe aussi des listes de définitions (<dl>), bien que leur domaine d'application ne soit pas précisément défini.

Les tableaux <table>

Cette fonctionnalité a été conçue pour la présentation de données sous forme de tableau, mais elle a rapidement été utilisée pour ses puissantes capacités de mise en page.

Cadres (<frames> / <iframe>) : Il s'agit d'une fonctionnalité qui autorise l'affichage de plusieurs documents HTML au sein d'une même fenêtre.

Formulaire (<form>) pour l'insertion interactive de données. Les champs du formulaire offrent aux utilisateurs la possibilité de saisir du texte et d'envoyer des fichiers.

Scripts (<script>) Permet d'associer des fragments de programmes aux actions effectuées par les utilisateurs sur le document. On utilise généralement JavaScript et VB Script comme langages de programmation. (20)

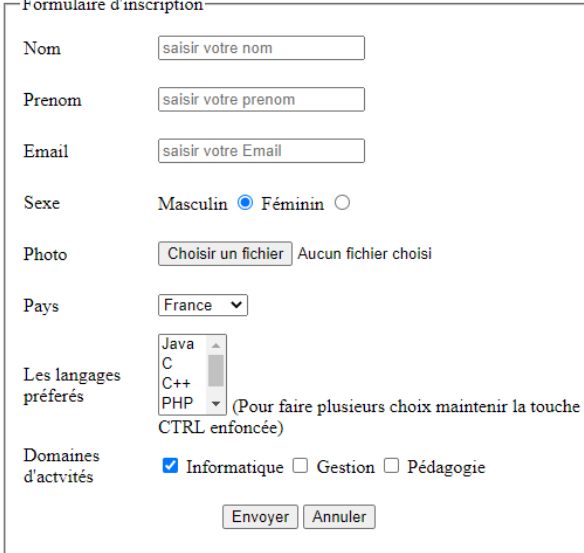
II.3.1.3 Fonctionnement de HTML

Le HTML est un langage de balisage que le navigateur web utilise pour rendre visible le contenu d'une page. Il organise l'information grâce à des balises qui délimitent et structurent les divers éléments tels que les titres, paragraphes, images et liens. Ces balises, identifiables grâce à leur syntaxe entre chevrons (< >), aident le navigateur à saisir la nature de chaque élément et à le présenter visuellement. Même si le HTML établit la structure fondamentale, c'est le CSS qui gère la présentation et l'aspect esthétique. La première étape pour maîtriser la création de pages web est d'apprendre les balises principales et leurs attributs, en sachant que leur utilisation suit des règles spécifiques mais assez simples à comprendre. (20)

II.3.1.4 Les formulaires HTML

Les formulaires HTML jouent un rôle clé dans l'interaction entre un utilisateur et un site web ou une application. Ils permettent à l'utilisateur d'envoyer des informations au site. Souvent, ces informations sont dirigées vers des serveurs web, mais la page peut aussi les prendre et les utiliser directement. Un formulaire HTML se compose d'un ou plusieurs éléments. Ceux-ci

peuvent inclure des zones de texte (que ce soit une ligne ou plusieurs), des listes déroulantes, des boutons, des cases à cocher ou des boutons radio. En général, ces éléments sont accompagnés d'une étiquette qui explique leur fonction. Des étiquettes bien conçues aident les utilisateurs, qu'ils soient voyants ou malvoyants, à comprendre ce qu'ils doivent remplir dans le formulaire. La principale distinction entre un formulaire HTML et un document HTML classique est que, en général, les données recueillies par le formulaire sont envoyées vers un serveur web (**La figure II.4**). Dans ce cas, nous devons avoir un serveur web pour recevoir et traiter ces données. Ce guide ne couvre pas la mise en place d'un tel serveur, mais si vous souhaitez en apprendre davantage, consultez la section « Envoi des données de formulaire » plus loin dans ce module. (21)



The image shows a screenshot of an HTML registration form titled "Formulaire d'inscription". The form contains the following fields and controls:

- Nom:** A text input field with the placeholder text "saisir votre nom".
- Prenom:** A text input field with the placeholder text "saisir votre prenom".
- Email:** A text input field with the placeholder text "saisir votre Email".
- Sexe:** Radio buttons for "Masculin" and "Féminin", with "Féminin" selected.
- Photo:** A "Choisir un fichier" button and the text "Aucun fichier choisi".
- Pays:** A dropdown menu currently showing "France".
- Les langages préférés:** A list box containing "Java", "C", "C++", and "PHP", with "PHP" selected. A note below says "(Pour faire plusieurs choix maintenir la touche CTRL enfoncée)".
- Domaines d'activités:** Checkboxes for "Informatique" (checked), "Gestion", and "Pédagogie".
- Buttons:** "Envoyer" and "Annuler" buttons at the bottom.

Figure II.4 : Exemple d'un formulaire HTML.

II.3.2 CSS (Cascading Style Sheets)

II.3.2.1 Définition du CSS

CSS, qui signifie « Cascading Style Sheets » ou « feuilles de style en cascades », est un langage utilisé pour définir l'apparence d'un document rédigé en HTML ou XML. Il explique les couleurs, les polices... Les feuilles de style ont été créées pour pallier les limitations du langage HTML en matière de mise en page et de présentation. Elles rendent un document plus plaisant à lire et plus soigné dans sa présentation. Son rôle est d'optimiser la présentation, de bien organiser la structure de son affichage et d'accroître l'accessibilité des pages web. Son

principe repose sur le regroupement, dans un même document, des éléments de mise en forme qui sont liés à des groupes d'éléments. Il suffit de créer un groupe de styles en lui donnant un nom, puis d'utiliser ce nom pour l'appliquer à un texte. (22)

II.3.2.2 La structure d'une page CSS

Pour appliquer des styles CSS à une page web, vous devez lier votre feuille de style à votre page HTML. Cela peut être fait en utilisant l'élément dans l'en-tête de votre page HTML.

Dès que votre feuille de style est associée à votre page HTML, il vous est possible d'employer des sélecteurs CSS afin d'appliquer des styles à des éléments spécifiques de cette dernière. Par exemple, il est possible de modifier la couleur d'arrière-plan de votre page via la propriété « background-color » et d'établir la taille du texte grâce à la propriété « font-size ». (La figure II.5)

En employant des styles CSS, vous avez la possibilité d'adapter le design de votre site internet afin d'offrir une expérience utilisateur plus plaisante et homogène. (23)

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
  <style>
    p { color: black;font-weight: bold;}
    h1{color: blue;font-size: medium;}
  </style>
</head>
<body>
  <h1> This is a heading</h1>
  <p>This is a paragraph</p>
</body>
</html>
```

Figure II.5 : Structure d'une page.

II.3.2.3 Fonctionnement de CSS

Les CSS opèrent en associant des règles de style aux éléments HTML. Ces styles déterminent des attributs visuels tels que la couleur, la taille de la police, l'interligne, le remplissage et ainsi de suite. Chaque règle stylistique se compose d'un sélecteur et d'une déclaration. Le sélecteur identifie l'élément HTML visé par la règle, tandis que la déclaration définit les propriétés esthétiques de cet élément. (24)

Plus techniquement, les balises HTML correspondent à divers éléments d'une page web, comme un paragraphe ou un titre. Par exemple, voici une balise HTML.

- `<p>Il s'agit d'un paragraphe</p>`.

N'oubliez pas que c'est le paragraphe dans son ensemble qui manque de style. Néanmoins, le code CSS ci-dessous offre la possibilité de donner du style à ce paragraphe :

- `p { color: purple; font-weight: bold; }`

Dans le code CSS précité, « p » sert de sélecteur, indiquant que le style est destiné à tous les paragraphes HTML. Le code entre parenthèses constitue la déclaration, comprenant les attributs (teinte et style de police) ainsi que leurs valeurs associées (violet et gras) dont le sélecteur tirera profit.

Ainsi, en ajustant les valeurs, on peut obtenir divers styles. Pour illustration, des attributs tels que « background-color », « background-position » et « text-align » peuvent présenter des valeurs correspondantes telles que « red », « top ». (25)

II.3.3 JavaScript

II.3.3.1 Définition du java script

JavaScript est un langage de script largement utilisé qui permet d'intégrer des fonctionnalités interactives et divers contenus dynamiques sur les pages web.

JavaScript est aussi assez intuitif et facile à apprendre. C'est une très bonne base pour ceux qui souhaitent approfondir leurs connaissances en matière de développement web.

JavaScript est la couche finale de fonctionnalité pour les sites web très interactifs. HTML constitue la structure fondamentale de la page. Le CSS constitue le composant de présentation de votre site web - il détermine l'apparence de votre site. JavaScript apporte ensuite une touche épicée. (26)

JavaScript propose une remarquable souplesse. Vous pouvez commencer modestement, avec des carrousels, des galeries d'images, des changements de disposition et des réponses aux pressions de boutons. Avec une meilleure expertise, vous pourrez créer des jeux, des animations en 2D et 3D ainsi que des applications complètes basées sur des bases de données. (21)

II.3.3.2 La structure de java script

L'incorporation de JavaScript dans une page Web se fait de manière très comparable à celle d'autres contenus HTML. (La figure II.6) Les balises utilisées pour l'insertion de JavaScript dans un document HTML sont `<script>` et `</script>`, le code qui se trouve entre les balises ces dernier est appeler script. (27)

```
<!DOCTYPE html>
<html>
<body>

<h2>Use JavaScript to Change Text</h2>
<p>This example writes "Hello JavaScript!" into an HTML element with id="demo":</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

</body>
</html>
```

Figure II.6 : Structure d'une page JavaScript.

II.3.3.3 Le fonctionnement du java script

JavaScript fonctionne avec HTML et CSS pour la conception d'applications ou de pages Web. La majorité des navigateurs web actuels, tels que Google Chrome, Firefox, Safari, Microsoft Edge, Opera et autres, supportent JavaScript. Désormais, la majorité des navigateurs mobiles sur Android et iPhone supportent également JavaScript.

JavaScript opère en exécutant des scripts qui sont écrits en JavaScript. Ces scripts sont mis en œuvre par un navigateur Web, ce qui leur donne la capacité d'interagir avec les utilisateurs. On peut utiliser des scripts pour concevoir des sites Web dynamiques et interactifs, en intégrant des fonctionnalités comme des animations, des formulaires, des menus ou encore des jeux. Il est également possible d'utiliser des scripts pour interagir avec des systèmes d'exploitation et des bases de données. (28)

II.3.4 PHP

II.3.4.1 Définition du PHP

PHP, qui signifie « PreHypertext Processor » ou « préprocesseur hypertexte », est un langage de programmation fait pour développer des sites web interactifs. À la différence

d'autres langages qui se limitent à montrer des informations fixes, PHP permet aux sites web de réagir aux actions des utilisateurs. Il offre la possibilité d'interagir avec le contenu de la page, de permettre à l'utilisateur de le modifier, de vérifier si la modification est pertinente et d'appliquer des règles de sécurité avant de l'enregistrer. PHP permet de créer des pages web dynamiques, ce qui signifie que le contenu peut évoluer selon les actions de l'utilisateur. Aujourd'hui, la plupart des sites web utilisent PHP pour rendre leur contenu interactif. (29)

II.3.4.2 La structure d'une page PHP

PHP est un langage de script qui s'intègre directement dans les pages Web. Cela indique que le code PHP est incorporé dans le code HTML. Les balises HTML encadrent le code PHP intégré dans votre fichier HTML, tout comme d'autres balises HTML le font. La création et la modification de pages Web utilisant PHP se déroulent de la même manière que pour les pages HTML traditionnelles. (La figure II.7) Les commandes du langage PHP sont encadrées par des balises PHP qui se présentent comme ceci : <?php ?> (30)

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello World!";
?>
</body>
</html>
```

Figure II.7 : Structure d'une page PHP

II.3.4.3 Fonctionnement de PHP

Exécution sur le serveur: Le langage de programmation PHP s'exécute sur un serveur web. Il n'est pas visible pour les utilisateurs car c'est un langage qui fonctionne à l'arrière-plan et non à l'avant. Lorsqu'un utilisateur envoie une requête HTTP (ex: accès à une page web), le serveur interprète le code PHP avant d'envoyer une réponse au client. Ce processus inclut :

- La connexion à une base de données (ex: MySQL).
- La génération dynamique de contenu (HTML, JSON, etc.).
- Le renvoi du résultat final au navigateur, sans jamais exposer le code PHP brut. (30)

Interaction entre le serveur et le client: À travers le navigateur, un utilisateur demande l'accès à une page (ex: formulaire de contact). L'utilisateur envoie une requête HTTP à travers le serveur client. Cette requête passe par un fichier avec l'extension .php qui se connectera au moteur PHP. Il exécute le script. Cette exécution permet de faire plusieurs choses :

- a. Requête HTTP: Le navigateur envoie une requête au serveur.
- b. Traitement PHP: Le serveur exécute le script PHP correspondant (fichier .php), qui peut :
 - Générer du HTML (construire la page dynamiquement).
 - Traiter des formulaires (récupérer \$_POST, valider des données).
 - Communiquer avec une BDD (SELECT, INSERT, etc.).
- c. Réponse au client : Le serveur renvoie le résultat (HTML/CSS/JS) affiché dans le navigateur. (30)

II.4. Base de données SQL (Structured Query Language)

II.4.1 Définition du SQL

Le SQL (Structured Query Language) est un langage de programmation utilisé pour manipuler des bases de données. Il autorise généralement la définition, la gestion et le contrôle de la sécurité des données. (31)

Ce langage offre non seulement la possibilité d'interroger une base de données via des requêtes, mais également de modifier sa structure ou ses informations. SQL permet l'ajout, la mise à jour ou la suppression de données, tout en offrant la possibilité de définir des autorisations d'accès pour assurer la protection des informations.

À l'inverse de certains outils tels que les tableurs, SQL est capable de gérer d'énormes quantités de données. On peut traiter des millions, voire des milliards de lignes sans que les performances ne soient affectées. De ce fait, SQL s'avère particulièrement bénéfique pour les entreprises qui gèrent des données à grande échelle. (32)

II.4.2 Fonctionnement de base de données SQL

SQL utilise des requêtes pour manipuler et interroger les données. Ses commandes sont classées en 5 catégories principales :

- **Data Definition Language (DDL)** : pour créer ou modifier des structures de bases (ex. CREAT, DROP).
- **Data Manipulation Language (DML)** : pour l'insertion, la mise à jour ou la suppression de données (ex. INSERT, UPDATE, DELETE).
- **Data Query Language (DQL)** : afin d'interroger les données (ex. SELECT).
- **Data Control Language (DCL)** : pour administrer les permissions d'accès (ex. GRANT, REVOKE).
- **Transaction Control Language (TCL)** : Pour superviser les transactions. (Ex. COMMIT, ROLLBACK). (33)

II.5. Outils de développement

II.5.1 Serveur Web Apache

Le serveur HTTP Apache, souvent désigné simplement sous le nom d'Apache, est un serveur web open source et gratuit facilitant la diffusion de contenu sur Internet. Apache facilite la communication entre les utilisateurs sur les réseaux, du client vers le serveur, en utilisant le protocole TCP/IP. Apache est compatible avec divers protocoles, cependant, le plus courant parmi eux est le HTTP/S. C'est l'un des protocoles majeurs du web et le plus reconnu d'Apache. HTTP/S définit la manière dont les messages sont structurés et transmis entre les serveurs web et les navigateurs. Il leur fournit des explications sur la manière de gérer les requêtes et les directives.

Le projet de serveur HTTP Apache permet de régler son fonctionnement à l'aide des fichiers de configuration Apache. Par défaut, Apache reçoit les demandes sur les adresses IP qui sont réglées dans ses fichiers de configuration. Avec la directive Listen, Apache peut recevoir et diriger un type particulier de trafic vers des ports et des domaines spécifiques, en fonction de requêtes qui combinent une adresse et un port.

Quand un message arrive à sa destination, Apache prévient l'expéditeur original pour lui confirmer que les données ont bien été reçues. Si l'hôte ou le client destinataire reçoit une erreur ou perd des paquets pendant le transport, il envoie un message « Not Acknowledged » (NAK) à l'expéditeur pour lui faire savoir que les données doivent être renvoyées. (34)

II.5.2 Serveur MySQL

MySQL Server est un système de gestion de bases de données relationnelles (SGBDR) open source, en même temps puissant et facile à utiliser, offrant une alternative gratuite aux solutions commerciales comme Microsoft SQL Server. En tant que serveur principal, il supervise l'établissement et la gestion des bases de données, coordonne des tâches comme la réalisation de requêtes et de transactions, tout en offrant des interfaces personnalisées pour les divers clients. Sa syntaxe SQL, claire et intuitive, le rend accessible tout en restant performant pour des besoins complexes. Conçu pour une intégration transparente, MySQL peut fonctionner en harmonie avec divers services, notamment les serveurs web, FTP ou de messagerie.

En outre, il offre des outils de configuration sophistiqués pour maximiser l'exploitation des ressources système, assurant par conséquent un équilibre entre performance et stabilité. À l'opposé de SQL Server, un produit sous licence, MySQL se caractérise par sa gratuité et sa souplesse tout en garantissant une grande fiabilité et performance. (35)

II.5.2.1 Fonctionnalités de MYSQL

Voici un tableau utile qui résume les fonctionnalités principales que nous avons abordées :
(36)

Tableau II.2 : Les fonctionnalités de MYSQL

Fonctionnalité	Description
Source ouverte	Libre à utiliser et à modifier
Multiplateforme	Opère sur différents systèmes d'exploitation.
Performance Élevée	Traitement et récupération des données de manière rapide.
Capacité d'évolution	Administre l'expansion des données et des utilisateurs
Protection des données	Caractéristiques de sécurité solides
Respect des principes ACID	Assurer l'intégrité des données
Procédures de stockage	Commandes de base de données préenregistrées
Éléments déclencheurs	Réponses automatiques aux événements de la base de données.
Recherche en texte intégral	Recherche efficace de contenu textuel

II.5.3 Wamp

WAMP (Windows, Apache, MySQL, PHP/Perl/Python) est une pile logicielle complète pour Windows, dédiée au développement et à l'hébergement d'applications web. Chaque composant joue un rôle essentiel : Windows fournit l'environnement système de base, Apache agit comme serveur web pour traiter et diffuser le contenu, MySQL gère les bases de données relationnelles, et PHP/Perl/Python permettent de développer des fonctionnalités dynamiques et d'interagir avec les données. Ensemble, ces technologies forment une plateforme puissante pour créer des sites web interactifs, des systèmes de gestion de contenu ou des applications e-commerce. (37)

II.6 Conclusion

Dans ce chapitre, nous avons particulièrement focalisé notre attention sur la compréhension des divers instruments employés dans l'établissement d'un site web. La sélection des technologies et des outils employés pour la conception de Lab Organizer a largement contribué à faire aboutir ce projet. L'utilisation de WampServer comme environnement local a permis une intégration harmonieuse des langages PHP, HTML, CSS et JavaScript, tout en étant très compatible avec le serveur de gestion de base de données MySQL. Chaque technologie a joué un rôle clé : HTML pour décrire la structure des pages, CSS pour leur mise en forme, JavaScript pour le côté interactif de la page et PHP pour traiter la logique de la page au sein du serveur. Cette combinaison des différentes technologies nous a permis d'assurer la réalisation d'un outil efficace, dynamique et extensible, capable de satisfaire les attentes d'un gestionnaire d'un laboratoire universitaire.

CHAPITRE III : CONCEPTION ET RESULTATS

III.1 Introduction

Le développement de logiciel implique une série d'activités qui, à partir d'une sollicitation pour numériser un processus (cette sollicitation pouvant varier d'une simple requête orale à un cahier des charges détaillé), permettent la conception et l'écriture. Et le développement d'un logiciel (et par conséquent de programmes informatiques) jusqu'à sa remise au requérant.

On mène conjointement une analyse des données et un examen des procédures à réaliser. C'est généralement durant cette étape que l'on met en œuvre les méthodes de modélisation. Cela conduit à l'élaboration d'une description des bases de données possibles à mettre en place, des programmes à rédiger et de la façon dont l'ensemble sera intégré.

Ce chapitre débutera par l'exposition de la méthodologie d'analyse, en expliquant pourquoi nous avons opté pour cette approche dans notre application. Par la suite, nous présenterons les intervenants de l'application et leurs fonctions respectives, ainsi que les différents diagrammes UML.

III.2 Le langage UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été conçu comme un langage de modélisation visuelle universel, offrant une grande richesse sémantique et syntaxique. Il est conçu pour l'architecture, la conception et l'implémentation de systèmes logiciels complexes, tant par leur structure que par leur comportement. L'UML trouve des utilisations dépassant le cadre du développement logiciel, y compris pour les flux de processus dans le secteur industriel.

Cela ressemble aux schémas utilisés dans d'autres disciplines et comprend divers types de diagrammes. Globalement, les diagrammes UML illustrent les frontières, la configuration et le fonctionnement du système ainsi que des objets qui y sont présents.

UML n'est pas un langage de codage, cependant, il existe des instruments permettant de produire du code en divers langages grâce aux diagrammes UML. L'UML est directement lié à l'analyse et la conception basées sur l'objet. (38)

III.3 Modélisation

III.3.1 Diagramme de cas d'utilisation (use case)

Les diagrammes de cas d'utilisation, qui font partie des diagrammes UML, sont employés pour fournir une vue d'ensemble du fonctionnement fonctionnel d'un système logiciel. Ils sont pratiques pour des présentations destinées à la direction ou aux intervenants d'un projet, cependant dans le cadre du développement, les scénarios d'usage sont plus appropriés. Un cas d'utilisation désigne une interaction distincte entre un utilisateur (que ce soit un humain ou une machine) et un système. C'est une unité de travail significative. Dans un schéma de cas d'utilisation (**La figure III.8**), on désigne les utilisateurs sous le terme d'acteurs (actors), car ils interagissent avec les scénarios d'utilisation (use cases).

Ce schéma vise à cerner les exigences du client afin de formuler le cahier des charges.

Principe : Établir les frontières du système. Caractérisation de l'environnement du système : les utilisateurs qui interagissent avec celui-ci.

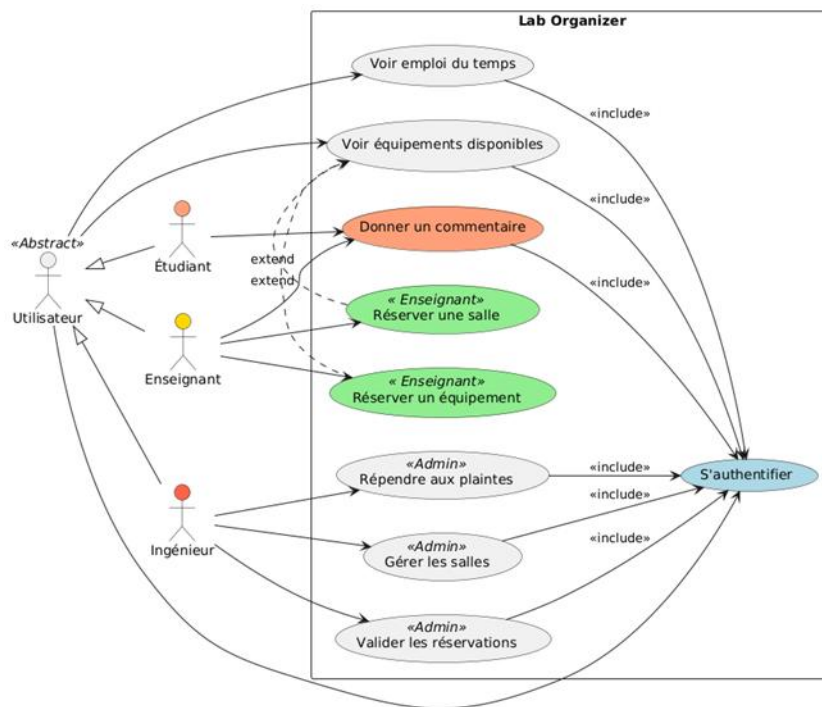


Figure III.8 : Diagramme de cas d'utilisation.

III.3.2 Les acteurs et leurs rôles

Ce sont des entités extérieures qui interagissent avec le système, comme une personne ou un robot. Une seule personne peut jouer plusieurs rôles dans un système. C'est pourquoi il est important de décrire les acteurs par leur rôle. Ce rôle montre ce dont l'acteur a besoin et ce qu'il peut faire. Un acteur influence le système. L'objectif de l'activité du système est de répondre aux besoins de l'utilisateur. Dans notre système, les acteurs et leurs rôles sont les suivants :

- Etudiant : s'authentifier, voir l'emploi de temps et les équipements disponibles, donner un commentaire (plaintes).
- Enseignant : s'authentifier, réserver une salle ou un équipement, voir l'emploi de temps et les équipements disponibles, donner un commentaire (plaintes).
- Ingénieur : avais l'accès de tous.

III.3.3 Diagramme de classe

Les diagrammes de classes jouent un rôle clé dans la modélisation des objets et illustrent la structure fixe d'un système. En fonction de la complexité d'un système, il est possible d'utiliser un seul diagramme de classes pour représenter l'ensemble du système, ou plusieurs diagrammes de classes pour illustrer ses différentes parties. Les diagrammes de classes représentent la structure principale de votre système ou sous-système. Ils permettent de représenter les objets qu'ils contiennent, de montrer les liens entre ces objets et de décrire leurs rôles ainsi que les services qu'ils offrent. La (**figure III.9**) illustre le diagramme de classe mettant en évidence les relations entre les objets d'un sous-système donné, ainsi que les services associés (39).

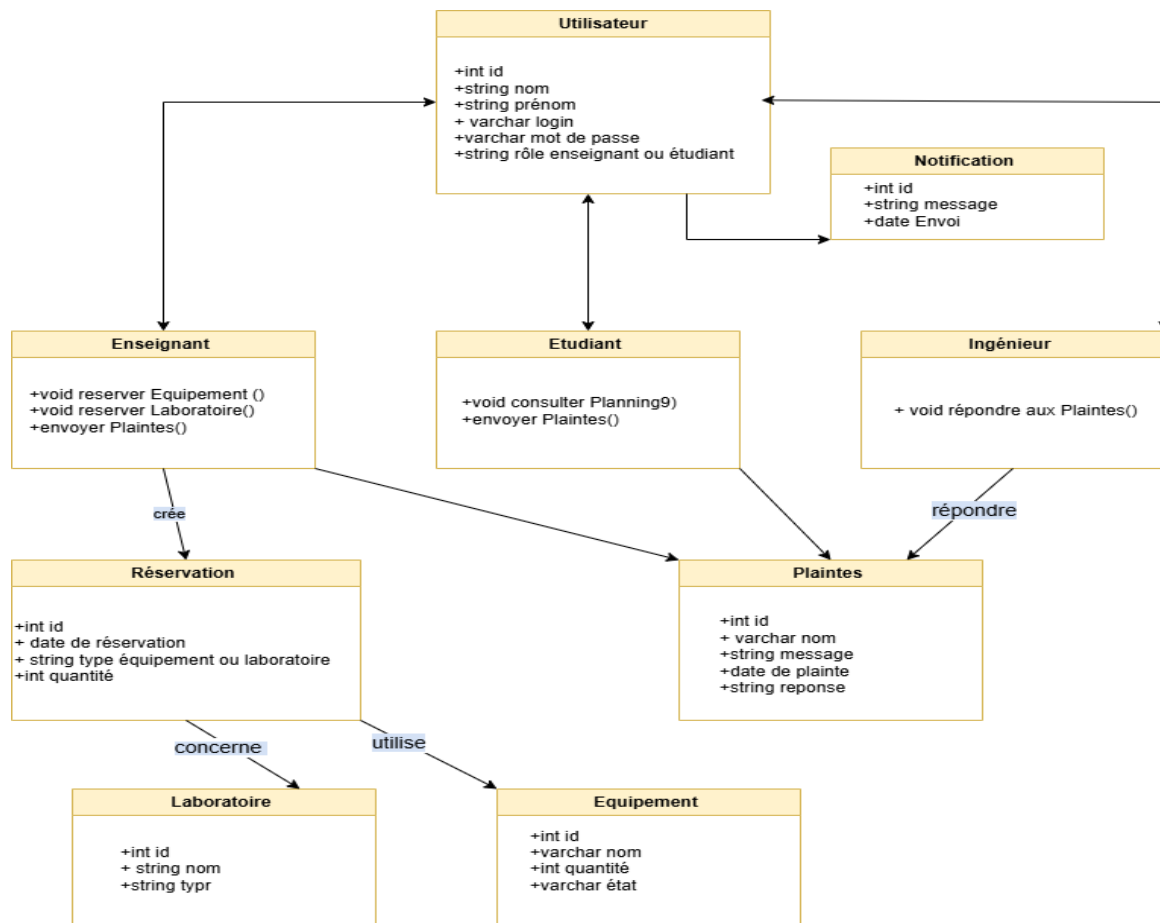


Figure III.9 : Diagramme de class.

III.3.4 Diagramme de séquence

Un diagramme de séquence est un élément essentiel du langage de modélisation unifié (UML) qui permet de visualiser les interactions entre objets dans un ordre séquentiel. Il se concentre sur la façon dont les objets interagissent au fil du temps, ce qui en fait un outil essentiel pour modéliser le comportement dynamique d'un système. Les diagrammes de séquence représentent les interactions entre objets, le flux de messages et l'ordre des opérations. Les (figure III 10, 11, 12,13) présentes les diagrammes de séquence de notre application mettant en évidence ces échanges successifs entre objets Ils sont donc utiles pour comprendre les cas d'utilisation, concevoir l'architecture système et documenter des processus complexes.

(40)

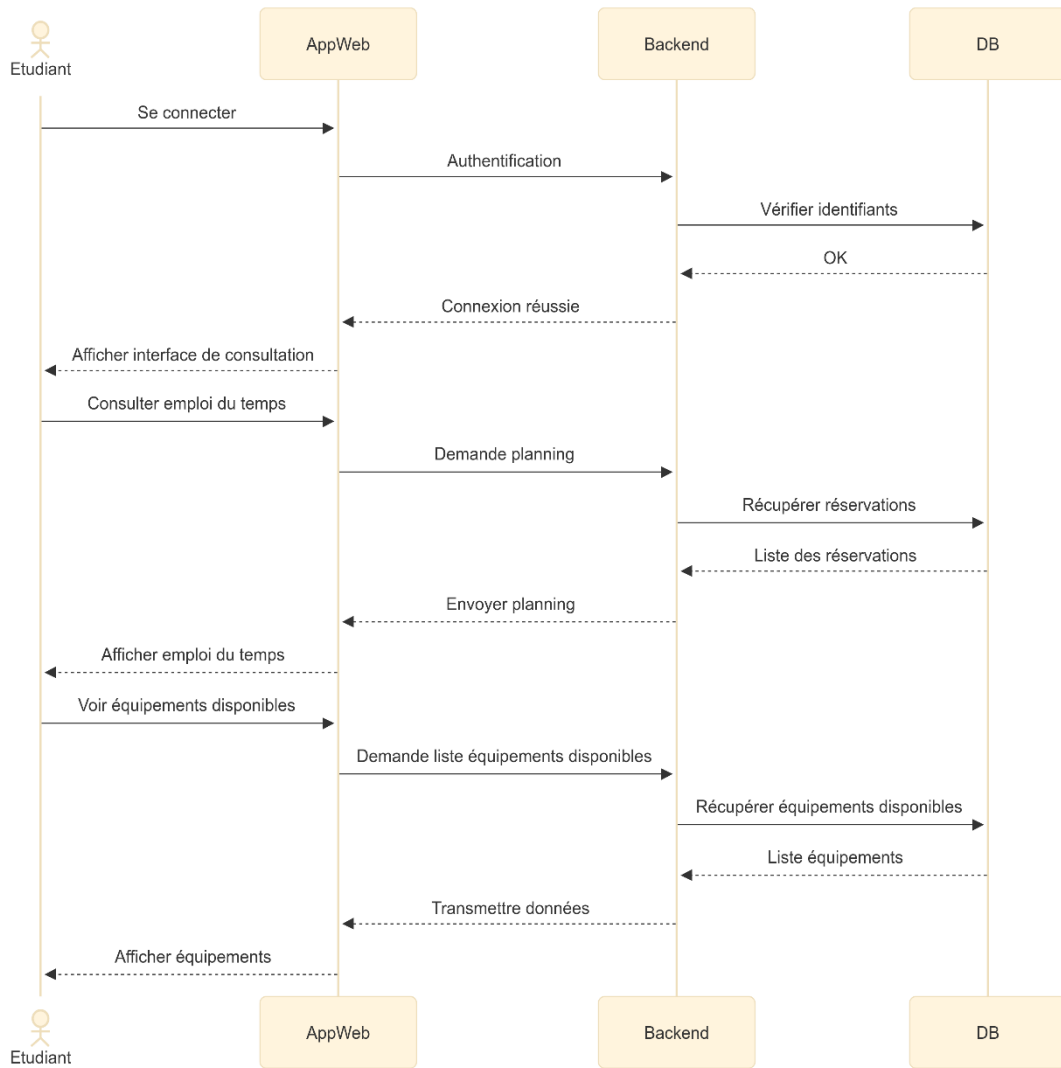


Figure III.10 : Diagramme de séquence (étudiant).

Ce diagramme de séquence (**Figure III.11**) décrit le processus de réservation d'un laboratoire et d'équipements par un enseignant via une application web, depuis l'authentification jusqu'à la confirmation de la réservation après vérification des disponibilités.

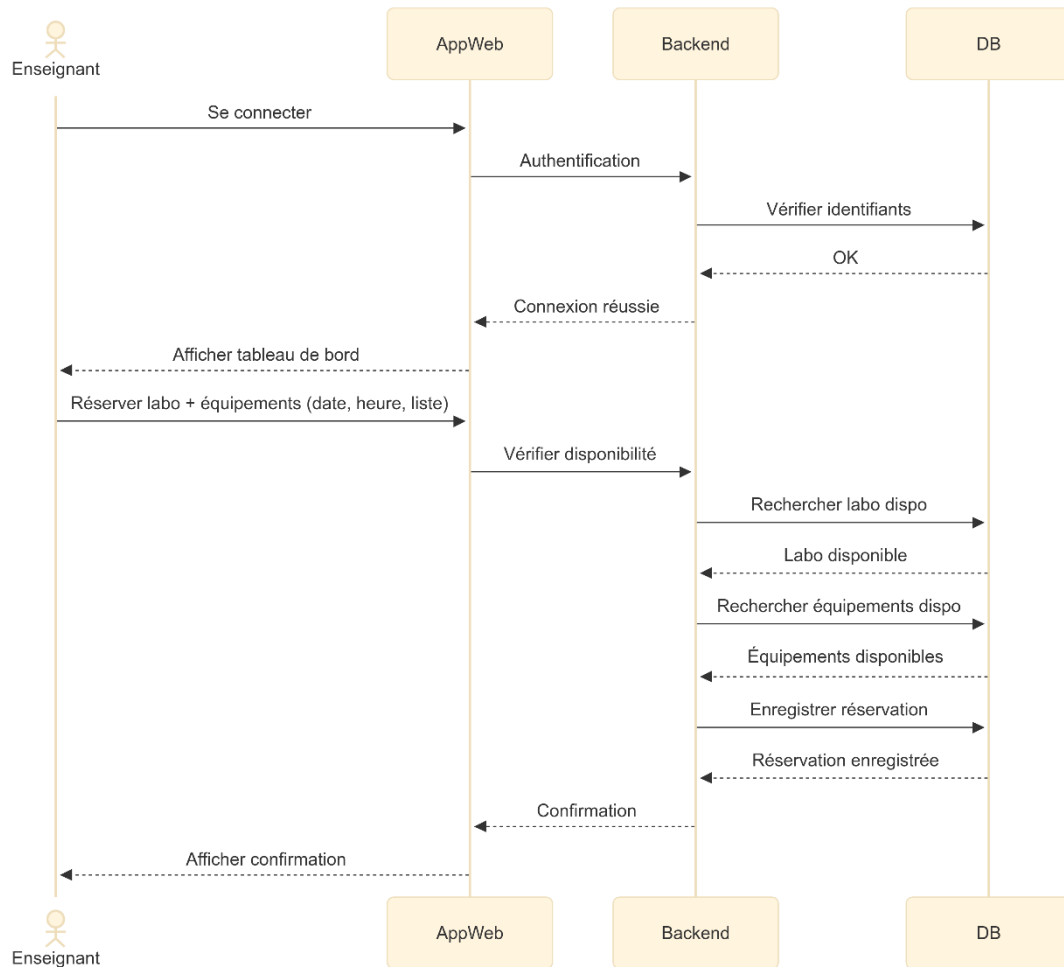


Figure III.11 : Diagramme de séquence (enseignant).

Ce diagramme de séquence (poser une plainte) (**figure III.5**) montre le processus par lequel un enseignant dépose une plainte via une application web. L'application interagit avec le backend et la base de données pour charger les informations nécessaires, enregistrer la plainte et afficher une confirmation à l'utilisateur.

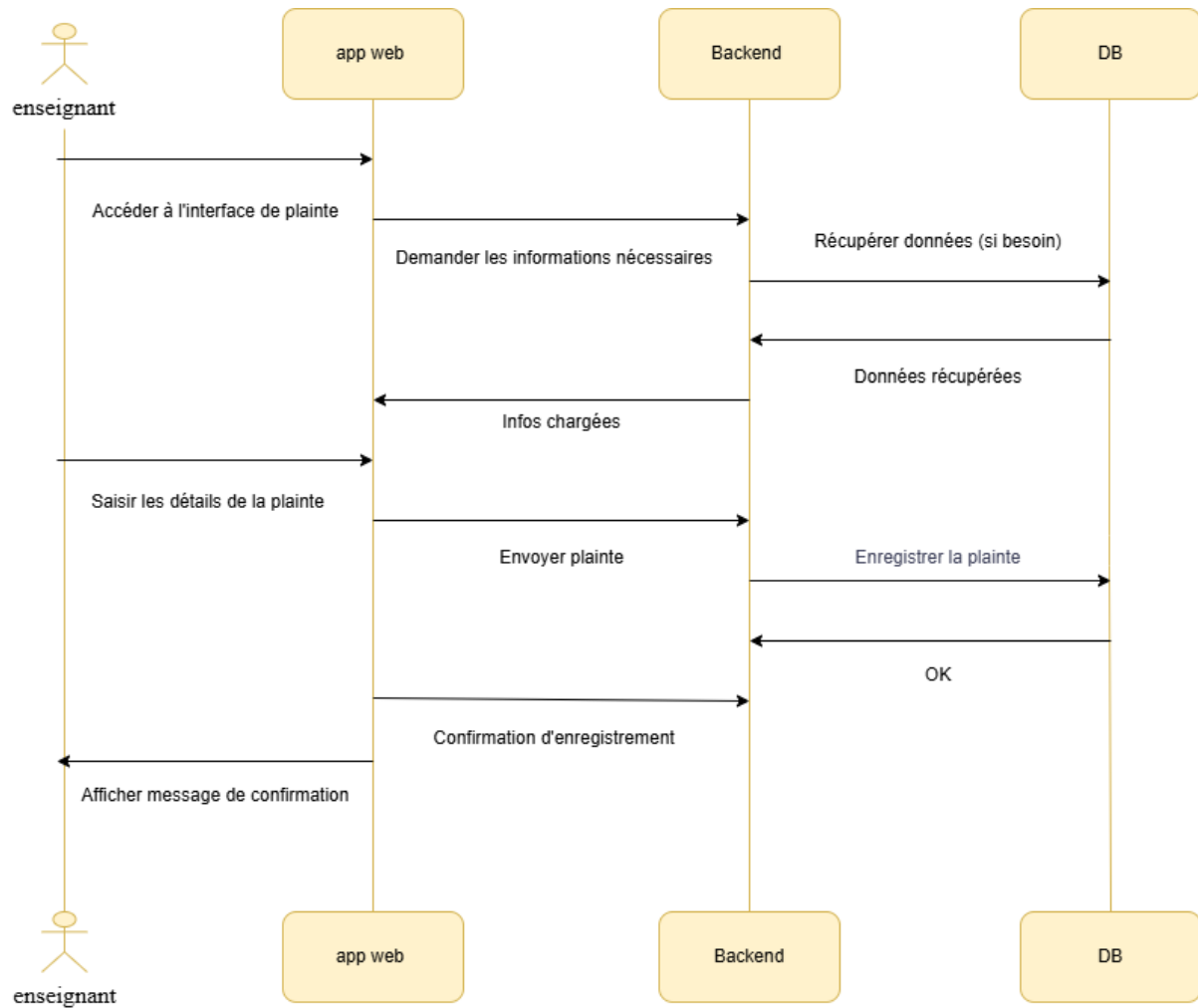


Figure III.12 : Diagramme de séquence (poser une plainte).

Ce diagramme répondre aux plaintes (**figure III.13**) illustre le processus de réponse à une plainte. L'utilisateur se connecte, sélectionne une plainte, saisit une réponse, puis celle-ci est enregistrée en base de données et la plainte est marquée comme traitée.

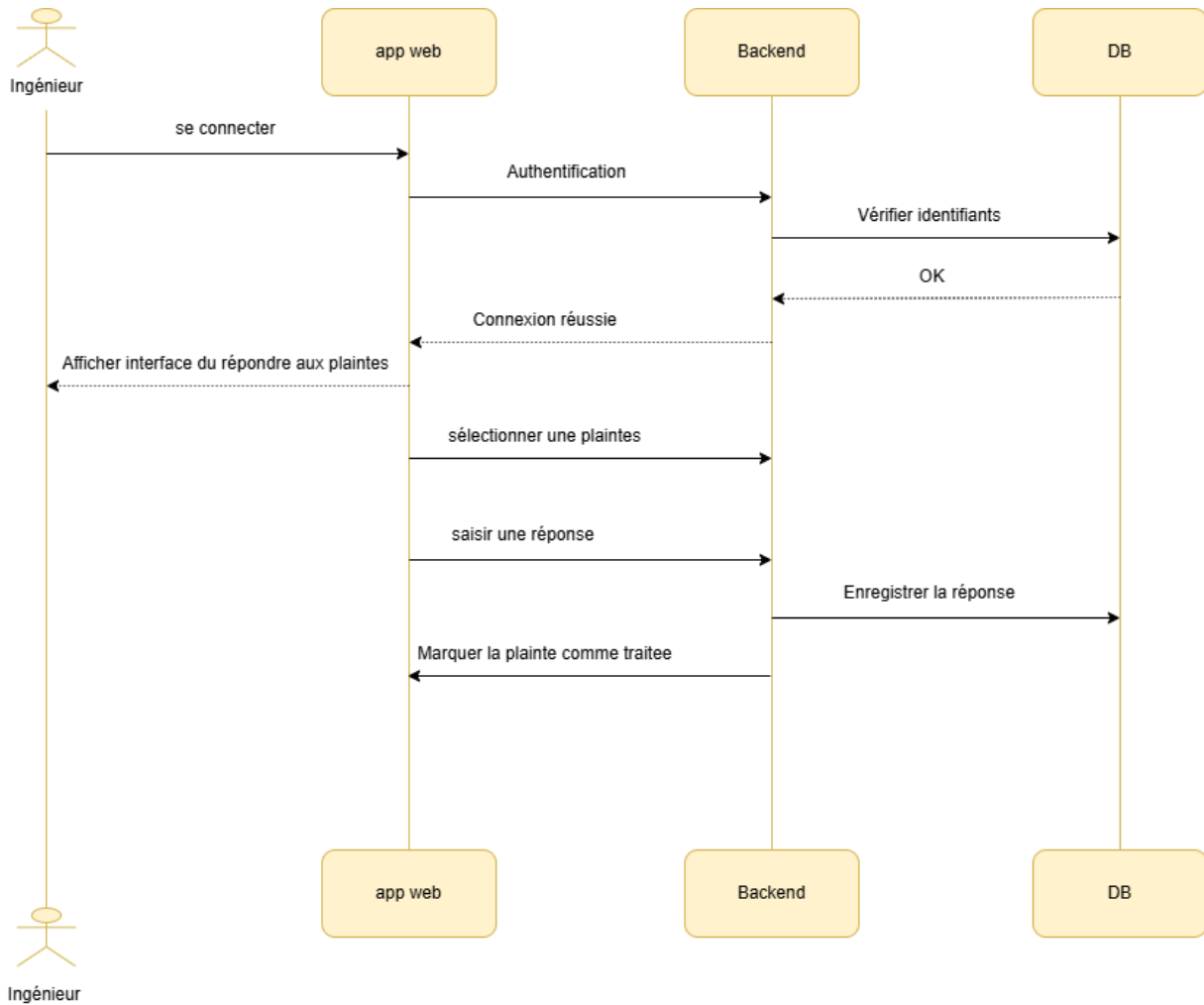


Figure III.13 : Diagramme de séquence (répondre aux plaintes).

III.3.5 Schéma relationnelle

Le schéma relationnel décrit la conception et la structure de la relation ou de la table dans la base de données. Il s'agit de la manière dont les états relationnels sont représentés, garantissant que chaque état de la base de données respecte les contraintes d'intégrité définies (telles que les contraintes de clé primaire, de clé étrangère, de non-null et d'unicité) au sein d'un schéma relationnel. Il se compose du nom de la relation, d'un ensemble d'attributs, de noms de champs et de noms de colonnes. Chaque attribut est lié à un domaine. La (Figure III.14) illustre notre schéma relationnel, mettant en évidence les attributs, les clés primaires et étrangères, ainsi que les relations entre différentes tables. (40)

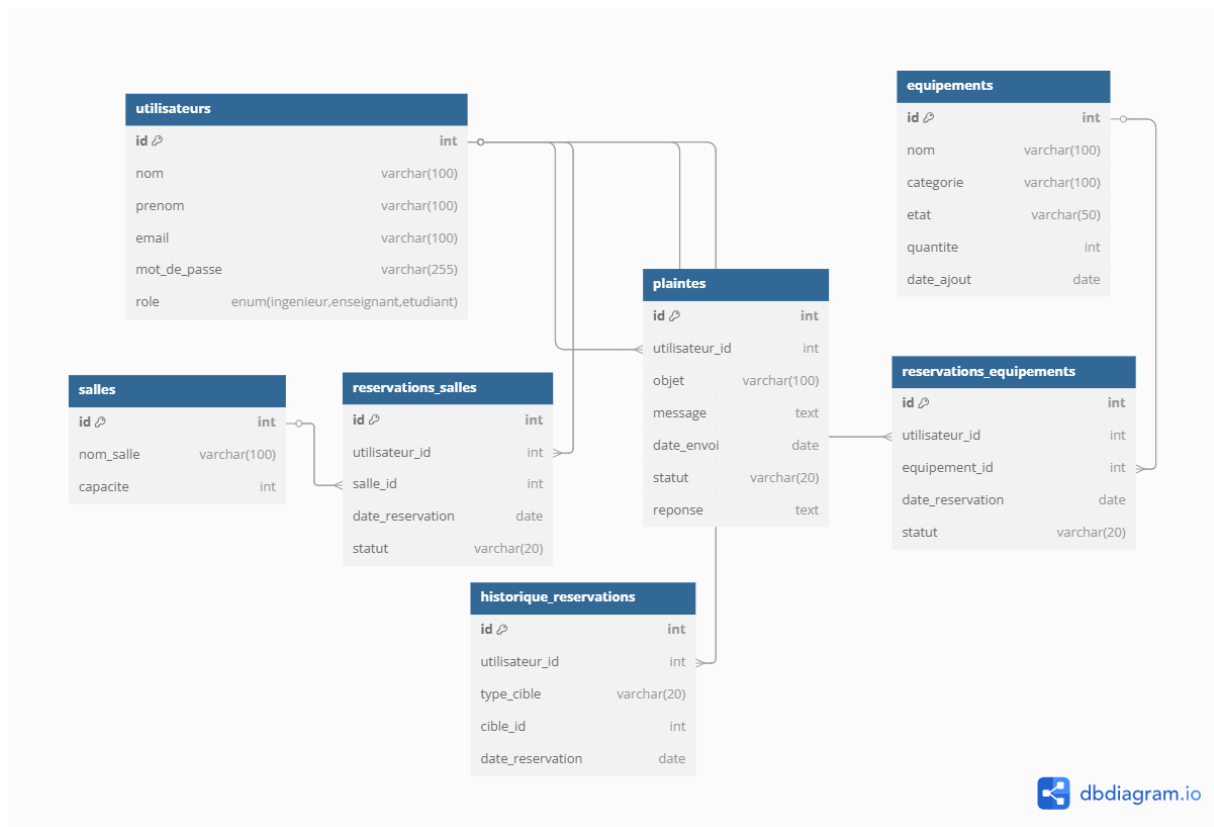


Figure III.14 : Schéma relationnel.

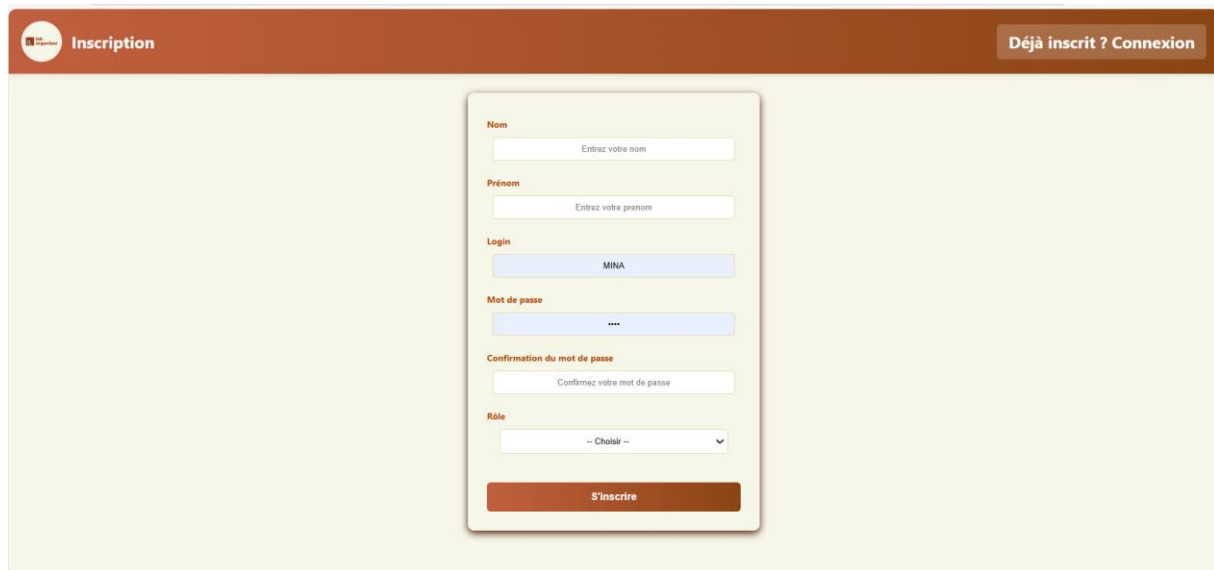
III.4 Les interfaces d'application web

III.4.1 Interface de d'inscription :

Cette interface autorise un nouvel utilisateur à s'inscrire sur Lab Organizer. Elle renferme un formulaire comportant les champs ci-après :

- Nom et Prénom : pour saisir le nom et prénom de l'utilisateur.
- Login : un nom d'utilisateur distinctif pour la connexion.
- Mot de passe : pour protéger l'accès.
- Confirmation du mot de passe : pour assurer que le mot de passe a été entré correctement.
- Fonction : une liste déroulante pour sélectionner Ingénieur, Enseignant ou Étudiant.
- Un bouton intitulé « S'inscrire » valide l'inscription.

Cette phase est cruciale pour l'accès à l'application et pour garantir que le système accorde les permissions appropriées à chaque utilisateur en fonction de son rôle.



The image shows a web interface for registration. At the top, there is a brown header bar with the word "Inscription" on the left and a button labeled "Déjà inscrit ? Connexion" on the right. Below the header, the main content area is light green. In the center, there is a registration form with the following fields and elements:

- Nom**: A text input field with the placeholder "Entrez votre nom".
- Prénom**: A text input field with the placeholder "Entrez votre prénom".
- Login**: A text input field containing the value "MINA".
- Mot de passe**: A text input field with a masked password "....".
- Confirmation du mot de passe**: A text input field with the placeholder "Confirmez votre mot de passe".
- Rôle**: A dropdown menu with the selected option "-- Choisir --".
- S'inscrire**: A brown button at the bottom of the form.

Figure III.15 : Interface d'inscription.

III.4.2 Interface de connexion

Cette interface offre la possibilité aux utilisateurs déjà enregistrés de se connecter à l'application Lab Organizer. Elle se présente comme un formulaire épuré et intelligible, comprenant :

- Un champ de connexion : pour saisir le nom de l'utilisateur.
- Un domaine Mot de passe : pour saisir le mot de passe.
- Un bouton Se connecter : afin de confirmer la connexion.
- Un lien « Cliquez ici pour vous inscrire » en bas, dirigeant vers la page d'inscription si l'utilisateur n'a pas encore de compte.

Dès l'ouverture de l'application, cette interface offre un accès à l'espace personnel en fonction du rôle choisi lors de l'inscription.

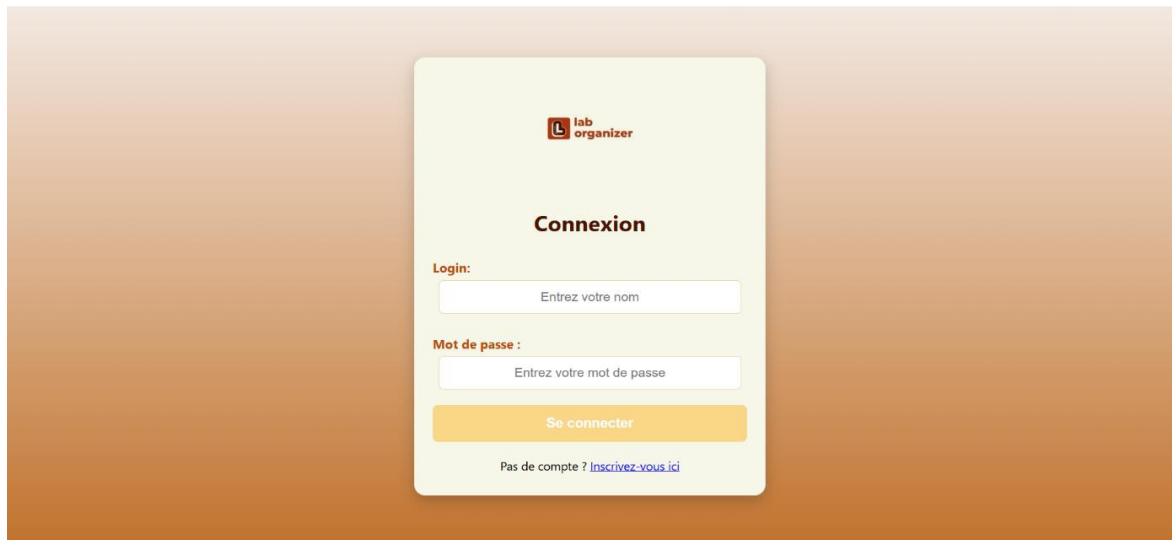


Figure III.16 : Interface de connexion.

III.4.3 Interface de l'Espace utilisateur

Après connexion, l'utilisateur est redirigé vers une **page d'accueil personnalisée**.

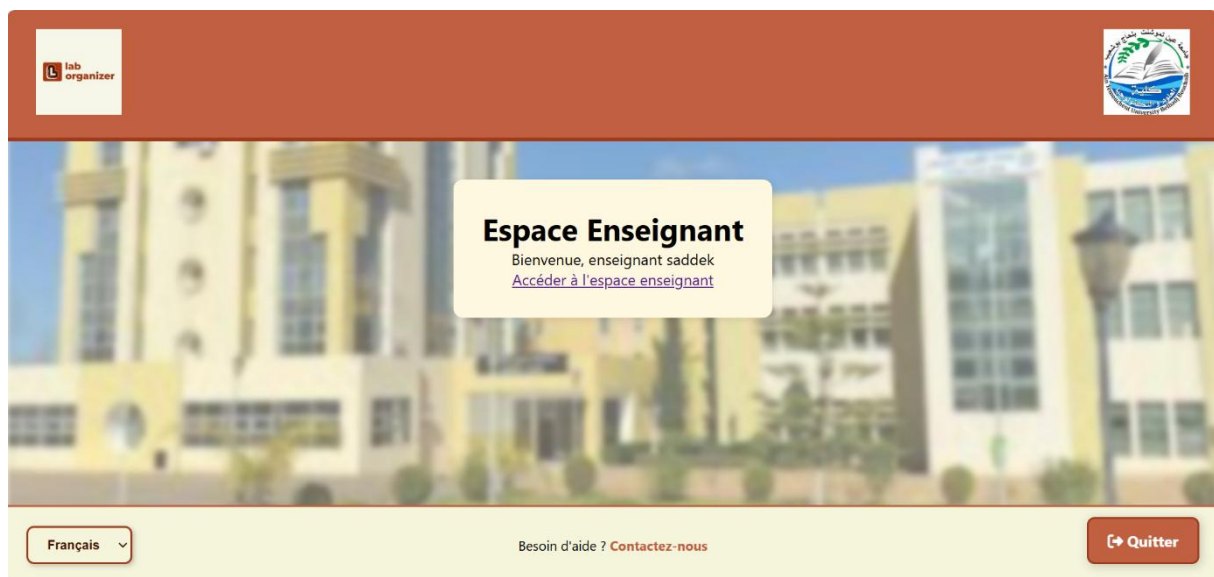


Figure III.17 : Interface des espaces.

III.4.4 Interface menu principale

Cette interface présente une page de contact de l'application. Elle est structurée de manière intuitive, présentant des éléments aisément identifiables :

- **Panneau de navigation (situé en haut)** : Elle comprend divers boutons (liens) pour naviguer vers les sections principales de l'appli :
 - Page d'accueil : Revenir à la page d'accueil.

- Matériel : Assurez-vous de la gestion ou de la consultation des équipements à disposition.
- Réservation : Faire une réservation pour une salle ou un équipement.
- Emploi de temps : Consulter ou administrer les horaires.
- plaintes: pour soumettre une réclamation.
- **Zone centrale (Zone de contact)** : contient Trois zones distinctes, contenant les coordonnées :
 - Email
 - Contacts téléphoniques
 - Localisation
- **Autres options (ci-dessous)** : la traduction de la page et Lien vers le support ou assistance.

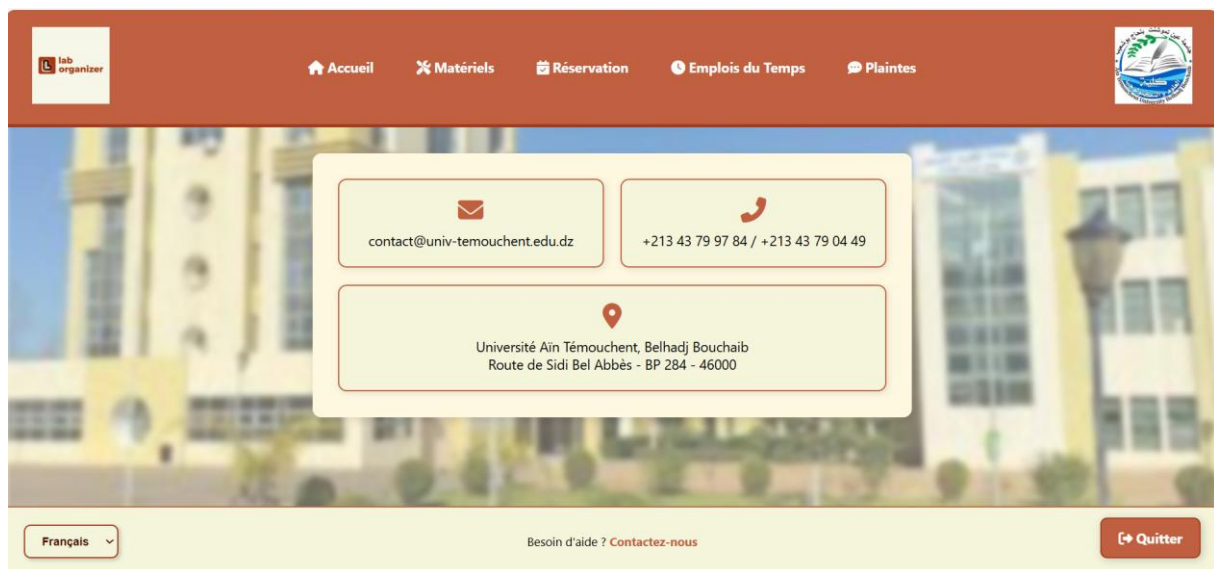


Figure III.18 : Menu principale pour l'espace enseignant et étudiant.

III.4.5 Interface de réservation

Cette interface figure est conçue pour être simple et intuitive. Elle constitue la première étape du processus de réservation, d'abord vous Choisissez un type dans la liste déroulante, Ensuite, l'utilisateur sera dirigé vers un formulaire plus détaillé selon le type choisi (équipement

Figure III.20 ou salle **Figure III.21**)



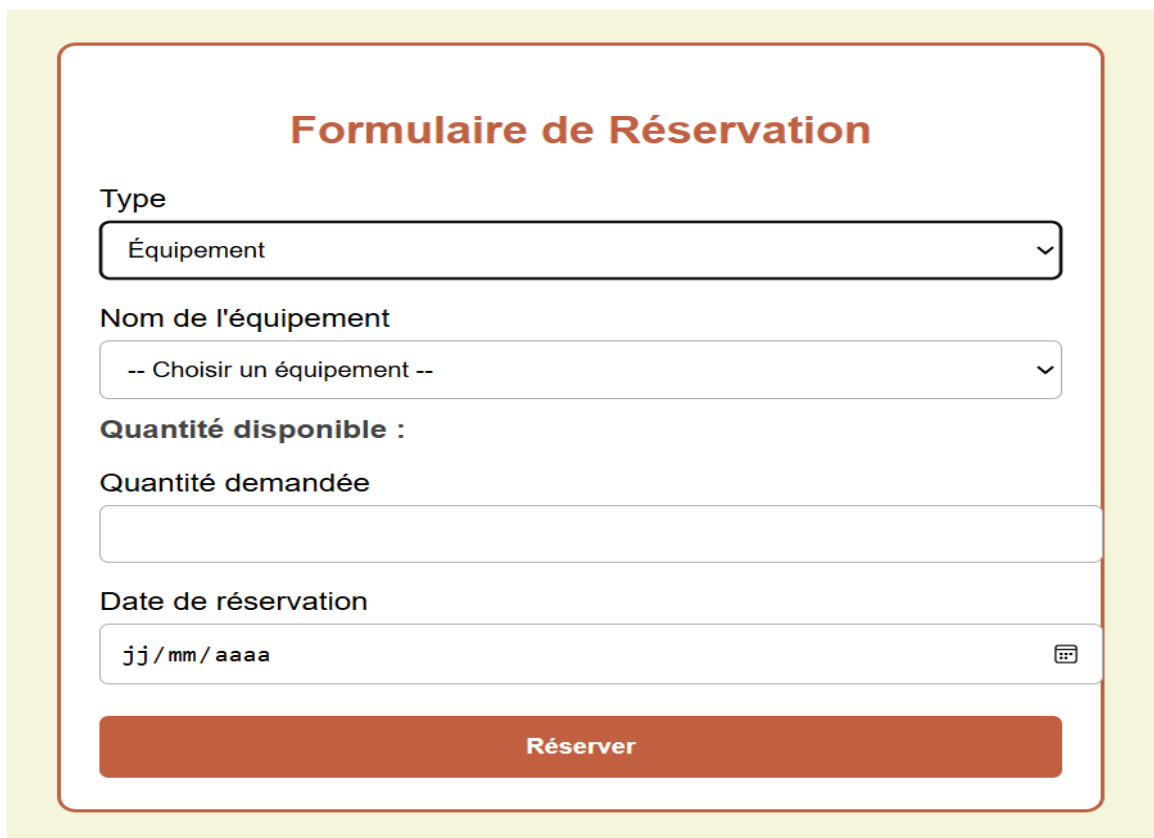
Formulaire de Réservation

Type

-- Choisir --

Réserver

Figure III.19 : Interface de réservation.



Formulaire de Réservation

Type

Équipement

Nom de l'équipement

-- Choisir un équipement --

Quantité disponible :

Quantité demandée

Date de réservation

jj/mm/aaaa

Réserver

Figure III.20 : Interface de réservation des équipements.



Formulaire de Réservation

Type

Salle

Nom de la salle

Ex: Salle A1

Date de réservation

jj/mm/aaaa

Réserver

Figure III.21 : Interface de réservation des salles.

Cette page (**Figure III.22**) s'affiche lorsque l'étudiant tente d'accéder à la réservation



Figure III.22 : Interface d'accès refusée pour la réservation tant qu'étudiant.

Enfin, interface **Figure III.23** de la liste des réservations montre toutes les réservations enregistrées



Liste des Réservations

#	Type	Objet	Date de Réservation	Quantité	Réservé par
1	Salle	A5	29/06/2025	1	wissem saddek
2	Équipement	voitmètre DC/AC:(3-30-100-300-600)V	28/06/2025	1	wissem saddek
3	Salle	A5	28/06/2025	1	Mina MAATALLAH

← Nouvelle Réservation

Figure III.23 : Interface de liste des réservations.

III.4.6 Interface des plaintes

Ce dispositif autorise les enseignants et étudiants à déposer une réclamation technique ou organisationnelle relative à un matériel, une salle de classe ou un service.

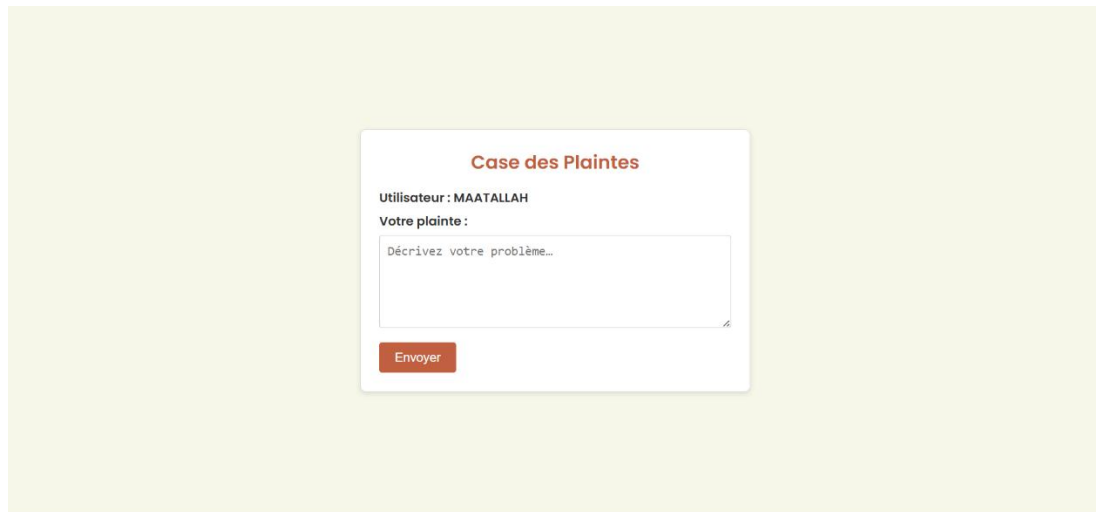


Figure III.24 : Interface des plaintes.

III.4.7 Interface de répondre aux plaintes

Cette interface est accessible uniquement aux ingénieurs ayant des rôles techniques et centralise toutes les plaintes obtenues pour une surveillance efficace et rapide.

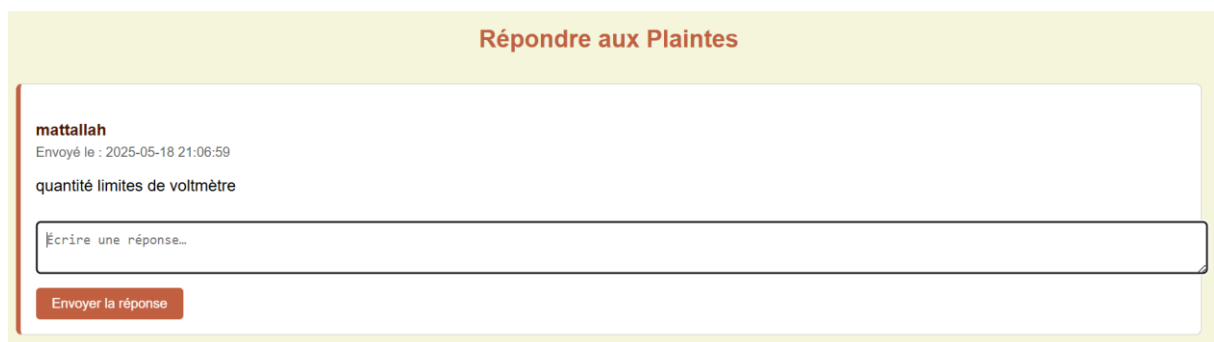


Figure III.25 : Interface répondre aux plaintes (ingénieure).

III.5 Conclusion

Dans ce chapitre, nous avons exposé les étapes clés de la phase de conception pour notre application web, en soulignant l'usage du langage UML (Unified Modeling Language) comme instrument de modélisation. Grâce à l'utilisation de divers diagrammes comme le diagramme de cas d'utilisation, le diagramme de classes et le diagramme de séquence, nous avons réussi à illustrer avec précision et organisation les exigences fonctionnelles, l'architecture logique et les interactions entre les composants du système.

Ces instruments de modélisation nous ont fourni une perspective intégrale et cohérente du système à concevoir, rendant ainsi l'organisation et la préparation des tâches futures plus aisées. Cette phase de conception représente un pont crucial entre l'évaluation des besoins et la phase de développement, assurant une mise en œuvre conforme aux spécifications initiales.

Conclusion générale

Notre projet s'inscrit dans la vision moderne des universités intelligentes. Le développement d'une application web de gestion de laboratoires de la faculté de technologies de l'université d'Ain Témouchent a concrétisé les compétences acquises durant notre formation, tout en répondant à un besoin urgent de numérisation.

L'application conçue facilite la gestion du matériel, des utilisateurs, des réservations et des activités, tout en garantissant une organisation améliorée et une traçabilité accrue. La phase de conception, basée sur l'UML, a donné lieu à une modélisation précise et organisée du système, tandis que l'étape de développement, utilisant des technologies web contemporaines (HTML, CSS, JavaScript, PHP, MySQL), a permis de concevoir une solution performante.

En dépit de certains défis techniques, ce projet a renforcé nos aptitudes en développement web, modélisation et gestion de projet, tout en apportant une contribution tangible à l'amélioration des services universitaires.

Enfin, ce projet ouvre des perspectives stratégiques : un déploiement étendu à l'échelle de l'université, une intégration avec des systèmes IoT pour le suivi temps réel du matériel, ou encore l'enrichissement par l'intelligence artificielle (notamment pour prédire les besoins en équipements). Par ailleurs, l'implémentation d'un tableau de bord analytique — affichant le taux d'utilisation du matériel ou optimisant les plannings de réservation — transformerait les données brutes en leviers décisionnels. Ces évolutions feraient de cette application un pilier central de la transformation numérique des universités, alliant efficacité opérationnelle et innovation.

Bibliographie

1. **DUHAUT, Patrick.** Les applications Web : définition, avantages et exemples concrets. *ONICIF - Agence Web No Code - Val d'Oise - 95*. [En ligne] Outsourcing Network Intelligence (ONI), avril 2025. <https://www.virtubox.io/blog/what-is-web-technology>.
2. **VirtuBox.** What Is Web Technology? Definition & Examples 2025. *VirtuBox*. [En ligne] VirtuBox, janvier 2025. <https://www.virtubox.io/blog/what-is-web-technology>.
3. **StudySmarter, l'équipe éditoriale de.** Technologies Web. *StudySmarter*. [En ligne] StudySmarter, 2025. [Citation : 28 mai 2025.] <https://www.studysmarter.fr/resumes/informatique/reseau-informatique/technologies-web/>.
4. **Desmont.** *Conception Web, Dominique Desmont*. 2008.
5. **Koredge.** définition et types de pages indispensables . *Koredge*. [En ligne] Koredge, 2 septembre 2024. [Citation : 29 mai 2025.] <https://www.koredge.fr/blog/creation-de-site-web/page-web-definition-et-types-indispensables/>.
6. **Delhaye, Eugénie.** Qu'est-ce qu'un site Internet ? *My Little Big Web*. [En ligne] My Little Big Web, 6 juillet 2023. [Citation : 25 mai 2025.] <https://mylittlebigweb.com/blogue/definition-site-web/#fhfdmq0c5>.
7. **Crystallize, l'équipe de rédaction ou de communication de.** Advantages and Disadvantages of Static Website. *Crystallize*. [En ligne] Crystallize, septembre 2023. [Citation : 29 mai 2025.] <https://crystallize.com/answers/tech-dev/advantages-and-disadvantages-of-static>.
8. **Ouarain, Ayoub.** Qu'est-ce Qu'un Site Web Dynamique ? *Hostinger*. [En ligne] Hostinger, 23 avril 2025. [Citation : 29 mai 2025.] <https://www.hostinger.fr/tutoriels/site-web-dynamique>.
9. **50A, l'équipe de rédaction ou de communication de l'agence.** Site dynamique. *50A*. [En ligne] 50A. [Citation : 29 mai 2025.] <https://www.50a.fr/nos-references>.
10. **Frissard., Loïc.** Application web : définition, utilité et avantages. *digitiz*. [En ligne] digitiz, 8 août 2022. [Citation : 25 mai 2025.] <https://digitiz.fr/definition-application-web/>.
11. **Idéematic.** Définition d'une application web. *Idéematic*. [En ligne] Idéematic. [Citation : 29 mai 2025.] <https://www.ideematic.com/dictionnaire-digital/application-web/>.
12. **Carmatec.** Architecture des applications Web : Guide complet 2025. *Carmatec*. [En ligne] Carmatec, 18 juin 2024. [Citation : 29 mai 2025.] https://www.carmatec.com/fr_fr/blog/guide-complet-de-larchitecture-des-applications-web/.
13. **Numérique, Assises du.** Tout ce qu'il faut savoir sur les serveurs : types, fonctionnement et applications. *Assises du Numérique*. [En ligne] Assises du Numérique, 13 janvier 2025. [Citation : 29 mai 2025.] <https://assisesdunumerique.fr/2025/01/13/tout-ce-quil-faut-savoir-sur-les-serveurs-types-fonctionnement-et-applications/#:~:text=%F0%9F%92%BE%20Les%20serveurs%20offrent%20un%20>

[moyen%20centralis%C3%A9%20de,en%20requ%C3%AAte%20pour%20programme s%20et%20trans.](#)

14. **Hostinger.** Qu'est-ce qu'un Serveur Web et Comment ça Marche ? *Hostinger*. [En ligne] Hostinger, 24 septembre 2024. [Citation : 29 mai 2025.] [https://chatgpt.com/c/6837abed-2694-8013-841e-9c718ab8802b.](https://chatgpt.com/c/6837abed-2694-8013-841e-9c718ab8802b)

15. **Innovations, eG.** Application Server. *eG Innovations*. [En ligne] eG Innovations. [Citation : 29 mai 2025.] [https://www.eginnovations.com/glossary/application-server.](https://www.eginnovations.com/glossary/application-server)

16. **Web, Liquid.** What is an Application Server? Definition and Examples. *Liquid Web*. [En ligne] Liquid Web. [Citation : 29 mai 2025.] [https://www.liquidweb.com/blog/application-server/#h-how-an-application-server-works.](https://www.liquidweb.com/blog/application-server/#h-how-an-application-server-works)

17. **wix.** Hébergement Web Gratuit. *wix*. [En ligne] wix. [Citation : 25 mai 2025.] [https://fr.wix.com/hebergement/gratuit/?utm_source=bing&utm_medium=cpc&utm_campaign=506129369^1275435133598958^search%20-%20dsa&experiment_id=https%3A%2F%2Ffr.wix.com%2Fhebergement%2Fgratuit%2F^b^^&msockid=285792f1c88a145c8522fac67e92d74a.](https://fr.wix.com/hebergement/gratuit/?utm_source=bing&utm_medium=cpc&utm_campaign=506129369^1275435133598958^search%20-%20dsa&experiment_id=https%3A%2F%2Ffr.wix.com%2Fhebergement%2Fgratuit%2F^b^^&msockid=285792f1c88a145c8522fac67e92d74a)

18. **papernest.** *Les technologies et outils utilisés dans le développement web*. [En ligne] 16 12 2002. [Citation : 29 mai 2025.] [https://www.papernest.com/blog/les-technologies-et-outils-utilises-dans-le-developpement-web/.](https://www.papernest.com/blog/les-technologies-et-outils-utilises-dans-le-developpement-web/)

19. **YAMINA, TALEB YAZID et ALILECHE.** *memoire Conception et réalisation d'une application web pour* . 2014/2015.

20. **Sophie.** tuto.com. *C'est quoi le HTML et à quel moment l'utiliser ?* 2020.

21. **docs, mdn web.** mdn web docs . *Mon premier formulaire HTML*. [En ligne] 1 1 2025. [Citation : 29 5 2025.] [https://developer.mozilla.org/fr/docs/Learn_web_development/Extensions/Forms/Your_first_form.](https://developer.mozilla.org/fr/docs/Learn_web_development/Extensions/Forms/Your_first_form)

22. **Oussama, Tigrine Youva et Tahir.** *memoire Conception et réalisation d'une application web pour la gestion des emplois du temps* . 2016/2017.

23. **Les bases du développement web: tout ce que vous devez savoir pour commencer.** *webpixelia*. [En ligne] mai 2023. [Citation : 29 mai 2025.] [https://webpixelia.com/les-bases-du-developpement-web-tout-ce-que-vous-devez-savoir-pour-commencer/#les-bases-du-d%C3%A9veloppement-web.](https://webpixelia.com/les-bases-du-developpement-web-tout-ce-que-vous-devez-savoir-pour-commencer/#les-bases-du-d%C3%A9veloppement-web)

24. **CSS : Signification, Fonctionnement et Rôle dans les Pages Web.** *commentouvrir it et technologie*. [En ligne] 2025. [Citation : 29 mai 2025.] [https://commentouvrir.com/tech/css-signification-fonctionnement-et-role-dans-les-pages-web/.](https://commentouvrir.com/tech/css-signification-fonctionnement-et-role-dans-les-pages-web/)

25. **Staf, Coursera.** Qu'est-ce que CSS ? *coursera*. [En ligne] 19 11 2024. [Citation : 29 mai 2025.] [https://www.coursera.org/frFR/articles/css?msockid=2d2a0bd5ba816d8a262b1f52bb6a6c77.](https://www.coursera.org/frFR/articles/css?msockid=2d2a0bd5ba816d8a262b1f52bb6a6c77)

26. Qu'est-ce que JavaScript ? Un aperçu du langage de script le plus populaire du web. *kinsta*. [En ligne] 3 12 2024. [Citation : 29 mai 2025.] <https://kinsta.com/fr/base-de-connaissances/qu-est-javascript/>.
27. JAVASCRIPT. *W3schools*. [En ligne] 2025. [Citation : 29 mai 2025.] https://www.w3schools.com/js/js_where.asp.
28. Comment fonctionne JavaScript ? 2023.
29. Qu'est-ce que PHP et comment ça fonctionne ? *systemin*. [En ligne] 13 9 2024. [Citation : 29 mai 2025.] <https://systemin.fr/actualites/quest-ce-que-php-et-comment-a-fonctionne>.
30. Langage PHP, Fonctionnement et utilisation. *aquilAPP*. [En ligne] 6 1 2023. [Citation : 29 mai 2025.] <https://www.aquilapp.fr/ressources/glossaire/langage-php-fonctionnement-et-utilisation>.
31. SQL (Structured Query Language) : définition, traduction et acteurs. *web&tech*. [En ligne] 2023. [Citation : 29 mai 2025.] <https://www.journaldunet.fr/web-tech/guide-du-big-data/1203603-sql-structured-query-language-definition-traduction-et-acteurs/>.
32. Introduction to SQL. *W3schools*. [En ligne] [Citation : 29 mai 2025.] https://www.w3schools.com/sql/sql_intro.asp.
33. Qu'est-ce que SQL ? Tout savoir sur le langage des bases de données. *lebigdata.fr*. [En ligne] 31 3 2025. [Citation : 29 mai 2025.] <https://www.lebigdata.fr/sql-tout-savoir-guide?nowprocket=1>.
34. How does Apache Web Server Work? *sumo logic*. [En ligne] [Citation : 29 mai 2025.] <https://www.sumologic.com/blog/apache-web-server-introduction>.
35. What is MySQL? *Greeks for Greeks*. [En ligne] 28 4 2025. [Citation : 29 mai 2025.] <https://www.geeksforgeeks.org/what-is-mysql/>.
36. MySQL - Fonctionnalités. *W3schools*. [En ligne] [Citation : 29 mai 2025.] <https://w3schools.tech/fr/tutorial/mysql/mysql-features>.
37. Qu'est-ce que WAMP ? *phoenixNAP*. [En ligne] 28 mars 2024. [Citation : 29 mai 2025.] <https://www.phoenixnap.fr/glossaire/qu%27est-ce-que-wamp#:~:text=WAMP%2C%20acronyme%20de%20Windows%2C%20Apache%2C%20MySQL%20et%20PHP%2FPerl%2FPython%2C,la%20le%20syst%C3%A8me%20d%27exploitation%20couche%20de%20la%20pile..>
38. Software, l'équipe éditoriale de Lucid. Types de diagrammes UML : Guide complet. *Lucidchart*. [En ligne] fin 2018 ou début 2019. <https://www.lucidchart.com/pages/fr/langage-uml>.
39. Corporation, IBM. Class diagrams in UML modeling. *IBM Documentation*. [En ligne] IBM Documentation, 2008. <https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>.
40. madhurihammad. Sequence Diagrams – Unified Modeling Language (UML). *GeeksforGeeks*. [En ligne] GeeksforGeeks, 3 janvier 2025. <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>.

