

ESPACE UNIVERSITAIRE REGIONALE DE L’OCEAN INDIEN

*(Excellence – Professionnalisme – Dépassement de soi)*

MEMOIRE DE FIN D’ETUDES POUR L’OBTENTION DU DIPLOME D’INGENIORAT

**Plateforme de communication, de gestion et de partage de documents dans un établissement scolaire**

**Présenté par** **:** RAKOTOARINTSIFA Loïc Judicaël Harrison.

**Encadreur :**

**Filière :** Electronique – Télécommunication – Informatique.

**Niveau :** Master II.

**Année :** 2023.

**REMERCIEMENTS**

Ce travail est le fruit de la combinaison d’efforts de plusieurs personnes. Je remercie tout d’abord le Tout puissant qui, par sa grâce m’a permis d’arriver au bout de mes efforts en me donnant la santé, la force, le courage et en me faisant entourer des merveilleuses personnes dont je tiens à remercier. Je remercie :

* **Madame Henriette Marie RASOANOMENJANAHARY**, Directeur Général de l’Espace Universitaire Régional de l’Océan Indien, de m’avoir permis de poursuivre mes études supérieures dans son honorable établissement.
* **Monsieur** , ce travail ne serait pas aussi riche et n’aurait pas pu voir le jour sans l’aide et la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant la préparation de ce mémoire.
* Je remercie également **tout le comité éducatif et les professeurs de l’EUROI**, pour leur générosité et la qualité de l’enseignement qu’ils m’ont prodigué au cours de ces trois années passées dans cet établissement.

Un grand merci qui s’adresse à toutes ma famille et particulièrement à mes chers parents qui m’ont soutenu moralement, physiquement, spirituellement et surtout financièrement durant ma formation entière jusqu’à la réalisation de ce mémoire de fin d’études.

De même, un grand merci à toutes les personnes qui ont contribuées, de loin ou de près, à l’achèvement de ce travail ; pour leurs conseils et leurs directives. Les discussions et les critiques ont été largement appréciés.

**AVANT – PROPOS**

L’Espace Universitaire Régional de l’Océan Indien connu sous le nom de l’EUROI, est une institution qui forme des jeunes bacheliers dans les domaines suivants : Télécommunication, Informatique et Electronique. Elle a pour devise « Excellence – Professionnalisme – Dépassement de soi ».

Tout étudiant au niveau M2 de formation au sein de l’établissement doit obligatoirement présenter un rapport du projet réalisé à la fin de la formation, qui lui permettra d’acquérir un diplôme d’ingéniorat.

La réalisation qui suit est une plateforme d’application web pour faciliter la communication des utilisateurs dans un établissement scolaire. Elle utilise le langage de programmation JavaScript ainsi que quelque une de ses Framework et librairie.

Ainsi, ce travail dévoilera les acquis théoriques et pratiques que nous avons reçus durant ces années d’études et de formation dans cet établissement.

**SOMMAIRE**

**REMERCIEMENTS**

**AVANT-PROPOS**

**SOMMAIRE**

**LISTE DES ACRONYMES**

**LISTE DES FIGURES**

**LISTE DES TABLEAUX**

**INTRODUCTION**

**PARTIE I : CONCEPTION THEORIQUE**

**CHAPITRE I :** PRESENTATION DU THEME

**CHAPITRE II :** ETUDE THEORIQUE DU THEME

**CHAPITRE III :** ETAPE DE CREATION D’UN SITE WEB

**PARTIE II : CONCEPTION PRATIQUE**

**CHAPITRE IV :** ETUDE PRATIQUE DU PROJET

**CHAPITRE V :** DIFFERENT SERVICES UTILISES

**PARTIE III : ESSAIS ET RESULTATS**

**CHAPITRE VI :** ESSAIS ET RESULTATS

* SIMULATION
* RESULTATS DES SIMULATIONS
* BILAN ET RECOMMANDATION

**CONCLUSION**

**ANNEXES**

**REFERENCES BIBLIOGRAPHIQUE ET WEBOGRAPHIQUE**

**TABLE DES MATIERES**

**LISTE DES ACRONYMES**

**LISTES DES FIGURES**

**LISTE DES TABLEAUX**

**INTRODUCTION**

Les NTIC (Nouvelles Technologies de l'Information et de la Communication) font référence à l'ensemble des technologies utilisées pour collecter, stocker, traiter et diffuser des informations numériques. Cela inclut les ordinateurs, les réseaux informatiques, les logiciels, les applications mobiles, les médias sociaux, les plateformes de commerce électronique, les systèmes de gestion de données, les systèmes de sécurité informatique, etc.

Les NTIC ont radicalement transformé notre façon de communiquer, de travailler, d'apprendre, de divertir et de consommer. Elles ont permis la création de nouveaux métiers, de nouveaux marchés et de nouvelles formes de collaboration et d'innovation.

Cependant, les NTIC ont également soulevé des préoccupations en matière de protection de la vie privée, de sécurité des données, de dépendance technologique, de désinformation et de polarisation, entre autres. Il est donc important de comprendre les enjeux et les défis liés à l'utilisation des NTIC et de développer une culture numérique responsable et critique.

C’est vers cette direction que notre projet se focalise avec comme thème : « Plateforme de communication, de gestion et de partage de documents dans un établissement scolaire »

Nous allons vers la suite nous focaliser sur la description d’une plateforme web, voir ces avantages et inconvénients ainsi que comment la créer et quelles sont les dispositions à prendre pour la création.

Partie 1 : Partie Théorique

# **CHAPITRE I : PRESENTATION DU THEME**

Ce mémoire de fin d’étude est le fruit de tout ce qu’on nous a enseigné pendant ces cinq années d’études au sein de l’EUROI. Suivi d’une réalisation qui a pour but d’exprimer la maîtrise des enseignements acquis ainsi que les diverses expériences professionnelles obtenues au sein du monde professionnelles.

* 1. **Choix de l’objet de la recherche**

De nos jours, l’homme utilise beaucoup les réseaux sociaux, notamment le Facebook. Cette dernière a permis à l’homme de se communiquer plus facilement, de partager des informations, etc. Or l’utilisation de Facebook n’est pas professionnelle dans certain domaine surtout dans le milieu professionnel. D’où la conception de cette plateforme de communication, de gestion et de partage de documents au sein d’un établissement scolaire.

* 1. **Objectif**

Le projet a pour but de faire communiquer les utilisateurs entre eux, de partager les informations ou documents dans une seule plateforme accessible à tous ses utilisateurs. Il nous permet aussi d’envoyer des mails sans sortir de la plateforme.

* 1. **Intérêts de l’objet de la recherche**

D’après l’analyse du projet, nous avons identifié les intérêts suivant :

* Un établissement possédant sa propre plateforme de communication et de partage de données à ses utilisateurs rend une meilleure image.
* Il suffit de quelques clics pour trouver des informations concernant les utilisateurs de la plateforme (administrateurs, professeurs, étudiants).
* Un tableau de communiqués pour les nouvelles à partager au sein de l’établissement visible par tous les utilisateurs n’importe quand et n’importe où.
* Possibilité d’envoi de mail dans la plateforme.
* Facilité de partage des documents/support de cours entre professeurs et étudiants.
* Facilité de discussion entre utilisateurs.
* Facilité de téléchargement de documents utiles comme les notes ou les différents certificats.
* Différentes accessibilités en fonction du rôle de chaque utilisateur.

## **1 – 4 – Architecture de la plateforme**

L'architecture d'une plateforme web se réfère à la structure, à la conception et à l'organisation globale d'un système ou d'une application web. Elle définit comment les différents composants d'une plateforme web interagissent entre eux pour offrir les fonctionnalités et les services attendus. Voici les principales composantes d'une architecture de plateforme web :

**Front-end :** C'est la partie visible de la plateforme web, qui est accessible par les utilisateurs via leur navigateur. Elle est généralement développée en utilisant des technologies de développement web comme HTML, CSS et JavaScript, et elle est responsable de l'interface utilisateur (UI) et de l'expérience utilisateur (UX) de la plateforme.

**Back-end :** C'est la partie invisible de la plateforme web, qui gère les fonctionnalités et les données en arrière-plan. Elle est généralement développée en utilisant des langages de programmation comme Python, PHP, Ruby, Java, ou Node.js, et elle est responsable de la logique métier, de l'authentification, de l'autorisation, de la gestion des bases de données, de l'intégration de services externes, et de la gestion des requêtes et des réponses entre le front-end et les autres composants.

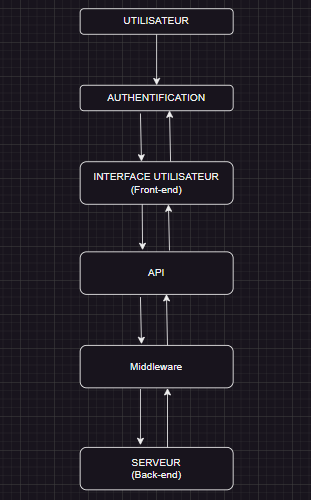
**Base de données :** C'est le composant qui stocke les données de la plateforme web, comme les informations des utilisateurs, les données des produits, les paramètres de configuration, etc. Elle peut être basée sur différents systèmes de gestion de base de données (SGBD) comme MySQL, PostgreSQL, MongoDB, ou d'autres, en fonction des besoins de la plateforme.

**Services externes** : Ce sont des services tiers utilisés par la plateforme web pour étendre ses fonctionnalités, comme les API de paiement, les services de messagerie, les services de géolocalisation, les services de stockage en ligne, etc. Ils sont intégrés à la plateforme web via des API (interfaces de programmation d'application) pour permettre l'échange de données et la communication avec ces services externes.

**Sécurité :** C'est un aspect essentiel de l'architecture d'une plateforme web pour garantir la protection des données sensibles des utilisateurs et la sécurisation des transactions. Cela peut inclure la mise en œuvre de protocoles de sécurité tels que HTTPS, la gestion des authentifications et des autorisations, la protection contre les attaques de sécurité, la validation des données d'entrée, la gestion des erreurs, etc.

**Scalabilité :** C'est la capacité d'une plateforme web à gérer efficacement une augmentation du nombre d'utilisateurs et de la charge de travail. Une architecture scalable doit permettre l'ajout de ressources (comme des serveurs) pour répondre à la demande croissante sans compromettre les performances et la disponibilité de la plateforme.

Concernant le projet, voici l’architecture correspondante :



***Figure 1 :*** *Architecture de la plateforme*

## **1 – 5 – Explication de l’architecture**

* **Authentification**

L'authentification est le processus de vérification de l'identité ou de l'authenticité d'une personne, d'un appareil ou d'une entité avant d'accorder l'accès ou les autorisations à certaines ressources, systèmes ou services. Il s'agit d'une mesure de sécurité utilisée pour s'assurer que seules les personnes ou entités autorisées sont autorisées à accéder à des informations protégées, des comptes ou des systèmes, tout en empêchant les accès non autorisés ou les violations de données.

L'authentification implique généralement la présentation de justificatifs tels que des noms d'utilisateur et des mots de passe, des jetons de sécurité, des données biométriques ou d'autres preuves d'identité. Ces justificatifs sont vérifiés par rapport à une source de confiance, telle qu'une base de données ou un serveur d'authentification, pour déterminer si l'entité qui tente d'accéder au système ou à la ressource est effectivement le propriétaire légitime des justificatifs.

L'authentification est un élément crucial de la sécurité de l'information et est utilisée dans divers contextes, notamment l'authentification des utilisateurs pour les comptes en ligne, le contrôle d'accès aux installations physiques ou aux réseaux informatiques, et l'authentification des appareils dans l'écosystème de l'Internet des objets (IoT). Différentes méthodes et techniques d'authentification peuvent être utilisées en fonction du niveau de sécurité requis et du cas d'utilisation spécifique.

Bien évidemment, la plateforme dispose d’une interface d’authentification pour permettre seulement aux utilisateurs vérifiés d’accéder à ses ressources et pour garantir la sécurité des données.

* **Interface Utilisateur (UI)**

Une interface utilisateur (IU) est un terme utilisé dans le domaine de l'informatique pour désigner le moyen par lequel un utilisateur interagit avec un système informatique, qu'il s'agisse d'un logiciel, d'un site web, d'une application ou d'un autre type de système. Une IU est conçue pour permettre à un utilisateur d'interagir avec un système informatique de manière conviviale et efficace.

Une IU peut inclure divers éléments, tels que des éléments visuels (comme des boutons, des menus déroulants, des icônes), des éléments interactifs (comme des formulaires, des boîtes de dialogue, des curseurs) et des éléments de navigation (comme des menus, des onglets, des liens). L'objectif d'une IU est de fournir à l'utilisateur une expérience de navigation fluide et intuitive, lui permettant d'accomplir ses tâches de manière efficace et agréable.

Les IU peuvent varier considérablement en fonction du type de système informatique et du public cible. Les principes de conception d'une IU peuvent inclure la facilité d'utilisation, la cohérence, la clarté, l'efficacité et l'accessibilité, entre autres.

En somme, une interface utilisateur est la partie du système informatique avec laquelle les utilisateurs interagissent pour accomplir leurs tâches et obtenir les résultats souhaités. Une IU bien conçue peut améliorer l'expérience de l'utilisateur, rendre l'utilisation d'un système plus facile et plus efficace, et contribuer à la satisfaction de l'utilisateur.

* **API (Application Programming Interface)**

Une API (Application Programming Interface) est un ensemble de règles et de conventions qui permettent à différents logiciels de communiquer et d'interagir les uns avec les autres. En d'autres termes, c'est une interface qui définit comment les différents composants logiciels peuvent interagir et échanger des informations.

Une API agit comme un pont entre différentes applications, permettant à celles-ci de partager des données, d'accéder à des fonctionnalités ou de communiquer entre elles de manière structurée. Elle définit les méthodes de communication, les formats de données, les conventions d'appel et les autorisations nécessaires pour interagir avec une application spécifique.

Les APIs sont largement utilisées dans le développement de logiciels et sont essentielles pour la création d'applications interconnectées et pour l'intégration de services tiers. Par exemple, de nombreuses plateformes de médias sociaux, services de paiement en ligne, services de cartographie, services de messagerie et autres fournisseurs de services proposent des APIs pour permettre aux développeurs d'accéder à leurs fonctionnalités et d'intégrer leurs services dans leurs propres applications.

Les APIs peuvent être de différents types, notamment les APIs RESTful (Representational State Transfer), les APIs SOAP (Simple Object Access Protocol), les APIs GraphQL, les APIs de bases de données, les APIs de messagerie, les APIs de médias sociaux, et bien d'autres. Elles sont utilisées dans une large gamme d'applications, du développement d'applications web et mobiles à l'Internet des objets (IoT), en passant par les services cloud, les applications d'entreprise, les systèmes intégrés et plus encore.

Autrement dit, une API est une interface qui permet à différentes applications de communiquer et d'interagir entre elles de manière structurée, en définissant les règles et les conventions de communication. Elle est un élément clé de l'intégration et de l'interopérabilité des applications logicielles.

* **Middleware**

Un middleware est un concept informatique qui fait référence à un logiciel ou à un composant logiciel qui agit comme une couche intermédiaire entre différentes applications, systèmes ou composants dans un environnement informatique. Il peut être utilisé dans différents contextes, tels que le développement de logiciels, la gestion de réseaux, les systèmes d'exploitation, les serveurs web, les applications serveur et les applications client-serveur.

En général, le middleware sert d'intermédiaire pour faciliter la communication, la coordination et l'intégration entre les différentes parties d'un système informatique. Il peut gérer les interactions entre les composants, les protocoles de communication, la sécurité, la gestion des erreurs, la transformation des données, la gestion des transactions, la mise en cache et d'autres fonctionnalités nécessaires pour la mise en œuvre d'une architecture logicielle complexe.

Un exemple courant de middleware est le middleware de serveur web, qui intervient entre le serveur web et les applications web pour gérer des tâches telles que la gestion des sessions utilisateur, la gestion des requêtes HTTP, la gestion des cookies, la mise en cache et d'autres fonctionnalités liées à la communication entre le client et le serveur. Un autre exemple est le middleware de gestion des transactions utilisé dans les systèmes de bases de données pour garantir l'intégrité des transactions et la cohérence des données.

Le middleware peut être développé en interne, acheté auprès de fournisseurs tiers ou utilisé en tant que service dans le cloud. Il peut être essentiel pour permettre l'intégration et l'interopérabilité entre les différents composants d'un système informatique complexe et faciliter le développement et la gestion des applications.

* **Serveur (back- end)**

Un serveur back-end est une composante d'un système informatique qui gère la logique de traitement des données et la communication avec les clients ou les utilisateurs d'une application ou d'un site web. Il est responsable de la gestion des données, du traitement des requêtes, de l'authentification et de l'autorisation des utilisateurs, et du logique métier de l'application.

Le serveur back-end traite les requêtes envoyées par les clients ou les utilisateurs, effectue les opérations nécessaires sur les données, et renvoie les réponses appropriées. Il peut communiquer avec d'autres composantes du système, tels que les bases de données, les services externes, ou les autres serveurs, pour récupérer ou stocker des données nécessaires au fonctionnement de l'application.

Le serveur back-end est souvent développé en utilisant des langages de programmation tels que Java, C#, Python, Ruby, ou PHP, et peut s'appuyer sur des Framework ou des bibliothèques pour faciliter le développement et la gestion des tâches courantes, tels que la gestion des requêtes HTTP, la gestion des bases de données, l'authentification et l'autorisation des utilisateurs, etc. Dans notre cas, le serveur back-end est développé par le langage de programmation JavaScript avec le Framework Node.js.

Le serveur back-end est essentiel pour assurer le bon fonctionnement d'une application ou d'un site web et pour gérer les opérations complexes liées au logique métier de l'application, tandis que le client ou l'interface utilisateur front-end est responsable de la présentation des données et de l'interaction avec les utilisateurs. Les deux côtés (front-end et back-end) travaillent ensemble pour fournir une expérience utilisateur complète et fonctionnelle.

## **1 – 6 – Gestion des utilisateurs**

La gestion des utilisateurs dans une plateforme fait référence à la manière dont les comptes d'utilisateurs sont créés, gérés, modifiés et supprimés dans le cadre d'une application ou d'un système en ligne. Cela peut inclure la gestion des informations d'identification, des autorisations d'accès, des rôles et des responsabilités, ainsi que la sécurité des données des utilisateurs.

**Création de comptes d'utilisateurs :** La plateforme permet la création de comptes d'utilisateurs de manière simple et sécurisée. Cela peut inclure la collecte des informations d'identification nécessaires, telles que les noms d'utilisateur, les mots de passe et les adresses e-mail valides, ainsi que la vérification de ces informations pour éviter les comptes frauduleux.

**Authentification et autorisation :** La plateforme met en place un système d'authentification pour vérifier l'identité des utilisateurs et garantir qu'ils ont l'autorisation d'accéder aux fonctionnalités et aux données appropriées. Cela peut inclure l'utilisation de méthodes d'authentification à plusieurs facteurs pour renforcer la sécurité.

**Gestion des autorisations :** La plateforme permet la gestion fine des autorisations d'accès des utilisateurs en fonction de leurs rôles et de leurs responsabilités. Cela inclut la définition de différents niveaux d'autorisation pour différents types d'utilisateurs, ainsi que la gestion des autorisations spécifiques pour les fonctionnalités et les données sensibles.

**Gestion des profils d'utilisateurs :** La plateforme permet aux utilisateurs de gérer leurs profils, y compris les informations personnelles, les préférences de communication, les préférences de confidentialité, etc. Les utilisateurs peuvent également modifier et mettre à jour leurs informations de profil en toute sécurité.

**Gestion des comptes inactifs ou supprimés :** La plateforme possède des politiques en place pour gérer les comptes inactifs ou supprimés, y compris la désactivation ou la suppression sécurisée des comptes inutilisés ou indésirables, ainsi que la gestion des données associées à ces comptes.

La gestion des utilisateurs dans une plateforme est un aspect essentiel de la sécurité et de la convivialité d'une application ou d'un système en ligne. Dans notre cas, seul les administrateurs peuvent ajouter des nouveaux utilisateurs.

## **1 – 7 – Fonctionnalités dans l’application**

Tout d’abord, il y a la fonctionnalité d’authentification mentionné ci-dessus, qui est primordial pour garantir la sécurité des données.

Ensuite, l’application possède les fonctionnalités suivantes :

* **Gestion des utilisateurs**

L’application permet la gestion des utilisateurs tels que la création, les mises à jour, lecture et la suppression des utilisateurs. Cela dit, seuls les administrateurs en ont le droit et l’accès.

Un rôle spécifique est attribué à chaque utilisateur et leurs accessibilités dans la plateforme dépendent de ce même rôle. Ces rôles sont les suivantes : Administrateur, Professeur, Etudiant.

La plateforme possède des interfaces/formulaire pour la création, la lecture, la mise à jour et la suppression des données d’utilisateur.

A chaque utilisateur créer, la plateforme envoi un mail à cette dernière. Cet e-mail est une forme de salutation de bienvenue dans la plateforme mais aussi il contient un mot de passe par défaut pour permettre au destinataire de se connecter à la plateforme.

* **Gestion des communiqués**

L’application enregistre une fonction d’ajout et affichage des communiqués. Il s’agit d’un tableau virtuel où seront exposé tous les communiqué/nouvelles dans l’établissement scolaire.

Son fonctionnement est la suivante : un administrateur ajoute une information à partager, que ce soit un fichier ou une image ou simplement du texte, ensuite il le publie. Après cette action, chaque utilisateur reçoit une alerte de nouveau communiqué et ils peuvent directement les consulter dans l’onglet dédié spécialement pour ces informations.

Il est à noter que seuls les administrateurs ont le droit et privilège d’ajouter ces nouvelles/communiqués.

* **Envoi de mail à partir de la plateforme**

Autre que les mails envoyés à chaque nouvel utilisateur, l’application offre une possibilité d’envoyer des mails à partir de la même plateforme. Cela est possible par la librairie du Framework Node.js qui est nodemailer.

L’application possède 500 mails journaliers gratuits à disposition. Il suffit de se diriger vers l’onglet spécifique pour l’envoi des mails.

* **Communication et partage de documents entre utilisateurs**

La plateforme permet de se communiquer entre utilisateurs. Les utilisateurs peuvent s’envoyer des messages entre eux, les étudiants entre étudiants, ou étudiants entre professeurs, ou professeurs entre professeur ou encore avec des administrateurs.

Autre que les messages de type textes, ils peuvent aussi envoyer des fichiers ou des images.

* **Téléchargement de document**

L’application permet la fonction de téléchargement de document. Autre que les fichiers partagés qui sont téléchargeables, on peut aussi télécharger des documents internes. Par exemple, les lettres d’introduction, les disciplines internes de l’établissement scolaire, les certificats de scolarités ainsi que les notes et les résultats des tests.

# **CHAPITRE II : ETUDE THEORIQUE DU THEME**

## **2 – 1 - Application Web**

Une application web est une application informatique qui est exécutée dans un navigateur web, tel que Google Chrome, Mozilla Firefox ou Microsoft Edge. Contrairement aux applications traditionnelles qui doivent être installées sur un ordinateur ou un appareil mobile, une application web ne nécessite qu'un accès à internet et un navigateur web pour être utilisée.

Les applications web peuvent avoir différentes fonctionnalités et peuvent être utilisées pour diverses tâches, telles que la gestion de projet, la communication, la comptabilité, la planification et bien plus encore. Les applications web peuvent être développées pour être utilisées sur des ordinateurs de bureau, des ordinateurs portables, des tablettes et des smartphones, et peuvent être accessibles à partir de différents systèmes d'exploitation.

## **2 – 2 - Avantages d’une application web**

Les avantages des applications web comprennent leur facilité d'utilisation, leur accessibilité multiplateforme, leur mise à jour automatique et leur faible coût de développement et de maintenance par rapport aux applications traditionnelles. Les applications web peuvent également être utilisées pour fournir des services en ligne, tels que le commerce électronique, la banque en ligne, les médias sociaux et les services de streaming de contenu.

**2 – 3 – Une plateforme web**

Une plateforme web est un ensemble de technologies, de logiciels et d'infrastructures qui permettent à des utilisateurs de créer, de partager, de stocker et d'accéder à des informations et des services en ligne.

Les plateformes web sont souvent constituées d'un ensemble d'applications et de services qui sont intégrés de manière transparente pour permettre aux utilisateurs d'interagir facilement avec elles. Elles peuvent être utilisées pour de nombreuses applications différentes, allant des réseaux sociaux aux systèmes de gestion de contenu, en passant par les plateformes de commerce électronique et les applications de productivité.

Les plateformes web sont des environnements en ligne qui permettent aux utilisateurs de créer, de partager, de stocker et d'accéder à du contenu et des services. Ces plateformes peuvent prendre différentes formes, allant des réseaux sociaux aux plateformes de commerce électronique, en passant par les blogs, les forums de discussion et les applications de productivité.

Les plateformes web peuvent être utilisées à des fins personnelles ou professionnelles, et elles offrent de nombreux avantages, tels que la facilité d'utilisation, l'accessibilité à partir de n'importe quel appareil connecté à Internet, la possibilité de collaborer avec d'autres utilisateurs en temps réel et la capacité à stocker des données et des fichiers en toute sécurité dans le cloud.

Les plateformes web ont également transformé de nombreux aspects de la vie quotidienne et des affaires, en offrant des moyens innovants de communication, de marketing, de vente, de gestion des projets et de collaboration à distance. Cependant, elles soulèvent également des préoccupations en matière de confidentialité des données, de sécurité et de dépendance envers les fournisseurs de services en ligne.

En fin de compte, les plateformes web ont transformé notre façon de travailler, de communiquer, de partager des informations et de consommer des produits et des services en ligne, et elles continuent d'évoluer pour répondre aux besoins changeants des utilisateurs et des entreprises.

## **2 – 4 - Différence entre une application web et une plateforme web**

Une plateforme web est un environnement en ligne qui offre une variété de services et de fonctionnalités, généralement sous la forme de logiciels en tant que service (SaaS). Les plateformes web peuvent inclure des outils pour la gestion de projets, la communication en ligne, la création de sites web, le stockage en ligne et la gestion des relations clients. Les utilisateurs peuvent accéder à ces services via un navigateur web sans avoir à télécharger ou installer de logiciel sur leur ordinateur.

Une application web, quant à elle, est un programme informatique qui s'exécute sur un navigateur web. Les applications web peuvent être des sites web interactifs, des applications de commerce électronique, des applications de réseaux sociaux, des jeux en ligne, des applications de productivité et plus encore. Contrairement à une plateforme web, une application web est généralement conçue pour répondre à un besoin spécifique et offre souvent des fonctionnalités avancées qui ne sont pas disponibles sur un site web standard.

En résumé, une plateforme web est une collection de services et de fonctionnalités en ligne, tandis qu'une application web est un programme informatique qui s'exécute sur un navigateur web et offre une fonctionnalité spécifique.

## **2 – 5 - Un navigateur web**

Un navigateur web est un logiciel qui permet aux utilisateurs d'accéder à des pages web sur Internet. Il interprète le code HTML, CSS et JavaScript utilisé pour créer les pages web et affiche le contenu dans une fenêtre de navigateur.

Les navigateurs web permettent également aux utilisateurs de naviguer sur Internet en suivant des liens hypertextes, de saisir des requêtes de recherche, de remplir des formulaires en ligne, de visionner des vidéos et d'interagir avec des applications web. Les navigateurs les plus populaires incluent Google Chrome, Mozilla Firefox, Safari, Microsoft Edge et Opera.

En résumé, un navigateur web est une application logicielle qui permet aux utilisateurs de naviguer sur Internet et d'accéder aux pages web.

## **2 – 6 - Un serveur web**

En termes simples, un serveur web est un ordinateur qui stocke, traite et fournis des fichiers de sites internet aux navigateurs web.

Les serveurs web se composent de matériel et de logiciels qui utilisent le protocole HTTP (HyperText Transfer Protocol). Il s’agit ici de répondre aux requêtes des utilisateurs web effectuées via le World Wide Web (www).

Grâce à ce processus, les serveurs web chargent et délivrent la page web demandée au navigateur de l’utilisateur.

Les serveurs web emploient également le protocole SMTP (Simple Mail Transfer Protocol) et le protocole FTP (File Transfer Protocol) pour traiter les fichiers pour les courriers électroniques ou le stockage.

## **2 – 7 - Composant matériels et logiciels d’un serveur web**

Côté matériel, un serveur web se connecte à Internet. Cela lui permet d’échanger des données ou des fichiers avec d’autres appareils également connectés. Ces données peuvent se présenter sous différentes formes, telles que des fichiers HTML, des images, des fichiers JavaScript ou des feuilles de style CSS. Le matériel du serveur web stocke en même temps le logiciel de serveur web.

Le logiciel de serveur web contrôle la manière dont les utilisateurs web accèdent aux fichiers hébergés. Il contient plusieurs composants, hébergeant au moins un serveur HTTP. Un serveur HTTP est un logiciel capable de comprendre les requêtes HTTP et les URLs.

## **2 – 8 - Fonctionnement d’un serveur web**



***Figure 2 :*** *Fonctionnement du HTTP (source : HOSTINGER)*

Les serveurs web suivent un modèle client-serveur. Dans cette structure, un programme, également appelé client, demande une ressource ou un service à un autre programme, le serveur.

Pour traiter les requêtes des clients web, les serveurs web suivent quelques étapes nécessaires :

* Lorsqu’un internaute souhaite charger le contenu d’un site web, son navigateur web demande l’accès via Internet. C’est ce qu’on appelle une requête HTTP. Le navigateur web recherche l’adresse IP du site internet demandé en traduisant l’URL des pages web via le Système de Noms de Domaines (DNS) ou en cherchant dans son cache. Ce processus localise le serveur web sur lequel les fichiers du site internet sont hébergés.
* Le serveur web reçoit la requête HTTP et la traite via son serveur HTTP. Une fois que son serveur HTTP accepte la requête, il effectue une recherche dans les fichiers du serveur afin d’obtenir les données requises du site internet.
* Après cela, le serveur web renvoie les fichiers du site au navigateur web qui a envoyé la demande. Ensuite, l’internaute voit le contenu des pages web.

Cependant, si le serveur HTTP ne parvient pas à trouver ou à traiter les fichiers demandés sur le serveur, il répond au navigateur web par un message d’erreur.

## **2 – 9 - Serveur web statique ou serveur web dynamique**

Les serveurs web peuvent transmettre du contenu statique ou dynamique. Un serveur web statique se compose d’un ordinateur et d’un logiciel HTTP. Les serveurs web statiques renvoient les fichiers d’un site web à un navigateur web sans aucune modification.

Un serveur web dynamique comprend un serveur web statique et des logiciels supplémentaires. Ces logiciels se composent le plus souvent d’un serveur d’application et de base de données.

Les serveurs web dynamiques mettent essentiellement à jour les fichiers hébergés avant de les diffuser via un serveur HTTP. Cela lui permet de générer et d’envoyer de contenu dynamique à un navigateur web.

## **2 – 10 - Fonctionnalités du serveur web**

Les serveurs web ont trois fonctions principales :

1. Héberger plusieurs sites web ou applications web.
2. Traiter les demandes de Protocole de transfert de fichiers (FTP).
3. Envoyer et recevoir des e-mails.

Outre la prise en charge des protocoles HTTP pour traiter les demandes et les réponses entrantes, la plupart des serveurs web offrent les fonctionnalités standards suivantes :

* **Journalisation des fichiers** : Les fichiers journaux documentent tous les événements ou activités que les serveurs web effectuent, tels que les requêtes, la sécurité et les erreurs. Chaque fois qu’un serveur web reçoit une nouvelle requête, une ligne de texte est ajoutée au journal.
* **Authentification** : De nombreux serveurs offrent cette fonctionnalité avant d’autoriser un accès partiel ou complet aux ressources d’un site web. Les fonctions d’authentification impliquent souvent des demandes d’autorisation. Cela se produit lorsqu’un nom d’utilisateur et un mot de passe sont requis.
* **Limitation de la bande passante** : La bande passante d’un serveur web est la quantité de données qu’il peut transférer ou traiter à un moment donné. La limitation de la bande passante contrôle la vitesse des réponses pour s’assurer qu’un réseau n’est pas saturé et peut livrer les fichiers sans problèmes.
* **Espace de stockage** : Il fait référence à la quantité d’espace disque disponible pour stocker des fichiers. Il détermine par conséquent si un serveur web peut héberger un site internet avec des caractéristiques spécifiques.
* **Langage de programmation** : Le langage de programmation d’un serveur web est le type de code utilisé pour développer des programmes exécutés par un serveur. Il est également connu sous le nom de langages de script côté serveur.
* **Disponibilité** : La disponibilité du serveur suit la durée pendant laquelle un serveur web fonctionnel et peut traiter les demandes ou fournir des fichiers.

# **CHAPITRE III : ETAPE DE CREATION D’UN SITE WEB**

La création d'une application web nécessite plusieurs étapes. Voici un aperçu des étapes générales pour créer une application web:

1. **Définir les objectifs et les fonctionnalités de l'application web :** Avant de commencer à construire une application web, vous devez définir clairement les objectifs et les fonctionnalités de l'application. Cela vous aidera à déterminer les technologies nécessaires pour construire l'application.
2. **Concevoir l'interface utilisateur :** La conception de l'interface utilisateur est une étape importante de la création d'une application web. Elle doit être facile à utiliser et intuitive pour les utilisateurs.
3. **Choisir une technologie de développement web :** Il existe de nombreuses technologies de développement web disponibles pour créer une application web, comme les langages de programmation (par exemple, JavaScript, Python, Ruby), les Framework (par exemple, React, Angular, Vue) et les bases de données.
4. **Développer l'application web :** Cette étape consiste à écrire du code pour créer l'application web en utilisant la technologie choisie. Vous devrez également créer une base de données pour stocker les informations de l'application.
5. **Tester l'application web :** Il est important de tester l'application web pour s'assurer qu'elle fonctionne correctement et qu'elle est exempte d'erreurs. Vous pouvez effectuer des tests manuels et automatisés pour identifier les bogues et les problèmes de performance.
6. **Déployer l'application web :** Lorsque vous êtes satisfait de l'application, vous pouvez la déployer sur un serveur pour qu'elle soit accessible aux utilisateurs. Il existe de nombreux services d'hébergement web disponibles pour déployer une application web.

## **3 – 1 –** **Langage de programmation**

### **3 - 1 - 1 – HTML**

L'HTML, abréviation de HyperText Markup Language, est un langage de balisage utilisé pour créer et structurer des pages web. Il s'agit d'un langage de base pour le développement web, permettant de décrire la structure et le contenu d'une page web en utilisant des balises et des attributs.

L'HTML est utilisé pour créer des pages web statiques et interactives, en définissant la structure du contenu, tels que les titres, les paragraphes, les listes, les images, les liens, etc.

Les balises contiennent des éléments, éventuellement enrichis avec des attributs. La page web est construite depuis un fichier HTML, élément par élément.

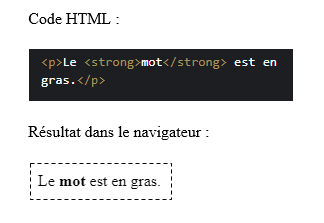
Pour écrire en HTML, il faut insérer des éléments et des attributs dans des balises, en utilisant la syntaxe imposée. Un paragraphe de texte, par exemple, constitue un élément à écrire entre les balises < >

Les éléments HTML, « tags » en anglais, désignent les différents types de contenus qui structurent la page web. Un paragraphe de texte, une image et une liste numérotée, par exemple, sont trois types de contenus distincts, chacun constitue donc un élément HTML spécifique. Pour indiquer au navigateur le type de contenu à afficher, il faut utiliser le nom de l'élément correspondant.

En langage HTML, la plupart des éléments s'écrivent, dans l'ordre :

* Une balise ouvrante : elle indique que l'élément commence. Elle s'écrit avec un chevron ouvrant et un chevron fermant, à l'intérieur duquel renseigner le nom de l'élément : < nom de l'élément >. La balise ouvrante n'est pas visible à l'écran.
* Le contenu : le contenu de l'élément s'affiche à l'écran.
* Une balise fermante : elle clôture l'élément. Elle s'écrit comme la balise ouvrante, en ajoutant une barre oblique après le chevron ouvrant : < /nom de l'élément >.

Pour insérer un élément paragraphe, par exemple, il faut écrire : <p> texte du paragraphe< /p>. Seul le contenu, c'est-à-dire la partie de l'élément « texte du paragraphe », est affiché par le navigateur, le reste du code est invisible. À noter que les éléments HTML ne sont pas sensibles à la casse, c'est-à-dire que l'emploi de majuscules et de minuscules est indifférent. L'élément paragraphe s'écrit indistinctement < p > et < P >. Pour des raisons de lisibilité et de cohérence, le développeur néanmoins a intérêt à utiliser systématiquement les minuscules.



***Figure 3 :*** *Exemple de code HTML*

Le langage HTML qualifie certains éléments :

* Les éléments HTML dits « vides » n'utilisent pas la syntaxe balise d'ouverture, contenu, balise de fermeture : ils se composent d'une balise unique. C'est notamment le cas des éléments vides image « img » et saut de ligne « br ». Pour sauter une ligne à la fin d'une phrase, par exemple, il suffit d'insérer l'élément < br > sans contenu ni balise de fermeture.
* Le langage HTML distingue les éléments de bloc des éléments en ligne. Les éléments de bloc structurent la page : chacun représente une portion de contenu, séparée des autres. Les éléments en ligne donnent des informations sur une partie d'un contenu. Illustration : l'élément de bloc paragraphe < p > inclut l'élément en ligne lien hypertexte < a >, pour insérer une partie du texte cliquable à l'intérieur du paragraphe.

Il est possible d'imbriquer des éléments les uns dans les autres : un élément est inséré dans un autre élément, pour augmenter le niveau d'information à fournir au navigateur à propos d'un contenu. Exemple : pour mettre un mot en gras dans un paragraphe, il faut imbriquer l'élément en ligne < strong > dans l'élément de bloc < p >. Le langage HTML est le suivant : < p > Le < strong >mot< /strong > est en gras. < /p >.

Le langage HTML suffit à coder une page web. En pratique néanmoins, d'autres langages informatiques lui sont souvent associés : les feuilles de style en cascade CSS permettent de personnaliser de manière rapide et avancée la mise en forme de la page web, JavaScript permet d'ajouter du contenu dynamique.

**Ecriture des attributs HTML**

Comme les éléments, les attributs HTML donnent de l'information au navigateur pour afficher le contenu. Mais à la différence de l'élément :

L'attribut n'est pas structurant : il ne détermine pas un type de contenu à afficher, il décrit le contenu.

L'attribut ne se suffit pas à lui-même : il est toujours l'attribut d'un élément. À noter que l'attribut est majoritairement facultatif. Certains éléments néanmoins ne fonctionnent qu'accompagnés d'un attribut. C'est le cas notamment de l'élément lien hypertexte < a > qui ne fonctionne qu'à condition de renseigner l'attribut href.

L'attribut s'écrit différemment : il a un nom et une valeur. Le nom de l'attribut est renseigné après le nom de l'élément, et suivi du signe égal ; la valeur de l'attribut est renseignée entre guillemets : < élément attribut="" > contenu < /élément >. Un élément peut inclure plusieurs attributs.

À noter que le langage HTML utilise des caractères spéciaux : ils sont part intégrante de la syntaxe du code. Le chevron « < », par exemple, est interprété comme du code et ne paraît donc pas à l'écran. Pour utiliser des caractères spéciaux dans du texte à afficher à l'écran, il faut les remplacer par des références spéciales, entourées des caractères « & » et « ; ». Pour insérer « < » afin de symboliser « inférieur à », par exemple, il faut écrire « < ».

**Forme d’une page HTML**

Le fichier HTML doit se présenter sous la forme suivante :

* La 1ère ligne du fichier mentionne le type de document comme suit : < !DOCTYPE html >. Cette information n'est pas visible à l'écran.
* La 2ème ligne indique le langage de balisage utilisé, avec l'élément racine : < html >. Cet élément englobe tout le code, il doit être fermé à la fin du fichier, avec la balise < /html >. Ces indications n'apparaissent pas à l'écran.
* La 3ème ligne insère l'élément < head > : toutes les informations contenues à l'intérieur des balises < head > informations < /head > sont invisibles par l'internaute. Il s'agit d'informations à destination du navigateur : meta description, title et CSS interne, notamment.
* L'élément < body > est inséré en suivant : c'est cet élément qui englobe le contenu à afficher sur la page web. Texte, images, tableau HTML, vidéos, formulaires ou encore boutons : tous les éléments sont insérés entre les balises < body > contenu à afficher < /body >.



***Figure 4 :*** *Squelette d’une page web*

### **3 – 1 – 2 – CSS**

CSS, ou "Cascading Style Sheets" (Feuilles de style en cascade en français), est un langage de programmation utilisé pour décrire la présentation et le style d'un document HTML. Il est largement utilisé dans le développement web pour contrôler l'apparence des pages web et permettre la séparation des préoccupations entre la structure (HTML) et la présentation (CSS) d'un site web.

Avec CSS, les développeurs web peuvent définir des règles de style qui déterminent la façon dont les éléments HTML sont affichés sur une page web. Cela inclut des propriétés comme la couleur, la police, la taille, la mise en page, les marges, les bordures et bien d'autres. CSS offre également la possibilité de créer des animations, des transitions et d'autres effets visuels pour améliorer l'expérience utilisateur sur un site web.

Une des principales caractéristiques de CSS est sa capacité à appliquer des styles en cascade, ce qui signifie que les règles de style peuvent être définies à différents niveaux (tel que le style inline, le style interne ou le style externe) et se combiner pour déterminer le style final d'un élément. Cela permet une gestion efficace et modulaire du style d'un site web.

En somme, CSS est un langage de programmation utilisé pour définir la présentation et le style des pages web, offrant ainsi un contrôle précis sur l'apparence des éléments HTML et permettant de créer des designs attractifs et modernes pour les sites web.

Comme pour HTML, CSS a commencé petit et grandit au fil des versions. La première version a vu le jour vers 1996. Il s'agissait de CSS1, suivie de CSS2 qui était le plus populaires jusqu'à l’apparition de HTML5 qui a intégré de nouvelles fonctionnalités qui ont emmené à développer la version 3 de CSS (dite CSS3) qui est le standard le plus utilisé actuellement. On utilise même le terme HTML5/CSS3 pour désigner les deux technologies qui vont alors ensemble.

CSS est un langage coté client, c'est à dire que sa syntaxe est comprise par le navigateur qui l’exécute avec le HTML. Ceci peut conduire à des problèmes de compatibilité, puisque par fois, les navigateurs ne comprennent pas toujours certains codes CSS de la même manière. Heureusement, ce sont des cas rares qui disparaissent petit à petit.

* **Framework CSS**

Un Framework CSS (Cascading Style Sheets) est une bibliothèque préparée de code CSS qui offre une collection de styles et d'outils normalisés pour simplifier le développement web. Les Framework CSS sont conçus pour simplifier le processus de création de pages web cohérentes, réactives et esthétiquement attrayantes en fournissant des classes CSS prêtes à l'emploi et des composants pouvant être facilement utilisés dans différents projets web.

En d'autres termes, un Framework CSS est un ensemble de fichiers CSS pré-écrits et organisés qui fournissent un ensemble de styles, de mises en page, de composants et d'utilitaires pour faciliter la conception et la mise en page d'un site web. Les développeurs web peuvent utiliser ces Framework pour accélérer leur processus de développement en réutilisant du code CSS existant, en adoptant des conventions de codage cohérentes et en bénéficiant de fonctionnalités prêtes à l'emploi pour créer des sites web rapidement et efficacement. Les Framework CSS sont généralement conçus pour être flexibles, réactifs et personnalisables, ce qui permet aux développeurs de les adapter aux besoins spécifiques de leurs projets.

Voici quelque exemple de Framework CSS :

* Bootstrap
* Foundation
* Materialize
* Skeleton
* Tailwind
* UIkit
* Bulma
* Semantic UI
* Animate css
* Material UI

### **3 – 1 – 3 – Javascript**

* **Introduction**

Jusqu'ici nous avons vu les langages HTML et CSS. Le premier est un langage de description de pages Web et se base sur les balises et le deuxième est un langage de style qui repose sur des déclarations simples sous la forme de directives.

Javascript quant à lui, est bel et bien un langage de programmation. Pour être plus précis, il s'agit d'un langage de script mais repose sur les mêmes principes qu'un langage de programmation évolué. D'ailleurs il est inspiré des fameux langages C et Java (et d'autres langages peu connus).

Les fonctions JavaScript peuvent permettre d'améliorer l'expérience utilisateur d'un site web, de la mise à jour des flux de médias sociaux à l'affichage d'animations et de cartes interactives. En tant que langage de script côté client, c'est l'une des principales technologies du web. Lors de la navigation sur Internet, à tout moment vous pouvez par exemple voir un carrousel d'images, un menu déroulant « Cliquer pour afficher » ou le changement dynamique de la couleur des éléments d'une page web. Tout cela est possible grâce à JavaScript.

* **Utilités du JavaScript**

Auparavant, les pages web étaient statiques, similaires aux pages d'un livre. Une page statique affichait principalement des informations dans une mise en page fixe et ne présentait pas toutes les fonctionnalités d'un site web moderne. JavaScript est apparu comme une technologie côté navigateur conçue pour dynamiser les applications web. Grâce à JavaScript, les navigateurs pouvaient répondre aux interactions des utilisateurs et modifier la mise en page du contenu sur la page web.

À mesure que le langage a évolué, les développeurs JavaScript ont établi des bibliothèques, des Framework et des pratiques de programmation et ont commencé à utiliser JavaScript en dehors des navigateurs web. Aujourd'hui, vous pouvez utiliser JavaScript pour le développement côté client et côté serveur.

* **Fonctionnement du JavaScript**

JavaScript est globalement catégorisé comme un langage de script ou langage interprété. Le code JavaScript est interprété, c'est-à-dire qu'il est directement transposé en un code de langage machine sous-jacent par un moteur JavaScript. Pour les autres langages de programmation, un compilateur compile l'intégralité du code en un code machine lors d'une étape distincte. Ainsi, tous les langages de script sont des langages de programmation, mais tous les langages de programmation ne sont pas des langages de script.

* **Moteur JavaScript**

Un moteur JavaScript est un programme informatique qui exécute du code JavaScript. Les premiers moteurs JavaScript n'étaient que de simples interprètes, alors que tous les moteurs modernes utilisent la compilation à la volée ou à l'exécution pour améliorer leurs performances.

* **JavaScript côté client**

JavaScript côté client fait référence à la manière dont JavaScript fonctionne sur votre navigateur. Dans le cas présent, le moteur JavaScript se trouve dans le code du navigateur. Les principaux navigateurs web sont équipés de leur propre moteur JavaScript intégré.

Les développeurs d'applications web rédigent le code JavaScript avec différentes fonctions associées à différents événements, comme le clic de souris ou le survol de souris. Ces fonctions apportent des modifications au HTML et au CSS.

Voici un aperçu du fonctionnement de JavaScript côté client :

1. Le navigateur charge une page web lorsque vous la consultez.

2. Lors du chargement, le navigateur transforme la page et tous ses éléments, comme les boutons, les étiquettes et les listes déroulantes, en une structure de données appelée Document Object Model (DOM, modèle d'objets de document).

3. Le moteur JavaScript du navigateur convertit le code JavaScript en bytecode. Ce code joue le rôle d'intermédiaire entre la syntaxe JavaScript et la machine.

4. Plusieurs éléments, comme le clic d'une souris sur un bouton, déclenchent l'exécution du bloc de code JavaScript associé. Le moteur interprète ensuite le bytecode et apporte les modifications au DOM.

5. Le navigateur affiche le nouveau DOM.

* **JavaScript côté serveur**

JavaScript côté serveur fait référence à l'utilisation du langage de codage dans une logique de serveur dorsal. Dans le cas présent, le moteur JavaScript se trouve directement sur le serveur. Une fonction JavaScript côté serveur peut accéder à la base de données, effectuer des opérations logiques différentes et réagir à plusieurs événements déclenchés par le système d'exploitation du serveur. L'avantage principal de la rédaction de scripts côté serveur est que vous pouvez grandement personnaliser la réponse du site web selon vos exigences, vos droits d'accès et les demandes d'informations provenant du site web.

* **Framework JavaScript**

Un Framework JavaScript est une bibliothèque ou un ensemble d'outils pré-écrits qui fournissent une structure et des directives pour construire des applications web en utilisant le langage de programmation JavaScript. Les Framework JavaScript simplifient le processus de développement en fournissant des fonctions, des classes et des API prêtes à l'emploi qui peuvent être utilisées pour gérer des tâches courantes, telles que la manipulation du DOM, la liaison de données, le routage et les requêtes AJAX, entre autres.

Il existe de nombreux Framework JavaScript disponibles, chacun avec ses propres fonctionnalités uniques et ses cas d'utilisation. Voici quelques exemples de Framework JavaScript populaires :

**React** : Développé par Facebook, React est une bibliothèque d'interface utilisateur largement utilisée pour construire des interfaces utilisateur dans les applications web. Il utilise une architecture basée sur les composants et fournit un DOM virtuel pour un rendu efficace des éléments d'interface utilisateur.

**Angular** : Développé par Google, Angular est un Framework complet pour les applications web qui suit une approche déclarative pour construire des applications web dynamiques. Il utilise un ensemble de composants, de directives, de services et de modules pour créer des applications web complexes.

**Vue.js** : Un Framework progressif pour la construction d'interfaces utilisateur, Vue.js permet aux développeurs d'adopter progressivement ses fonctionnalités au besoin. Il offre une architecture flexible et évolutive pour la construction d'applications web modernes.

**Ember.js** : Un Framework avec des opinions pour la construction d'applications web ambitieuses, Ember.js suit l'approche "convention plutôt que configuration". Il fournit un ensemble de conventions et de meilleures pratiques pour structurer des applications web complexes.

**Backbone.js** : Un Framework léger pour la construction d'applications monopages, Backbone.js fournit un ensemble minimaliste d'outils pour gérer les événements, les modèles, les vues et les collections. Il est souvent utilisé en combinaison avec d'autres bibliothèques pour construire des applications web complexes.

Ce ne sont que quelques exemples parmi les nombreux Framework JavaScript disponibles. Le choix d'un Framework dépend des besoins spécifiques de votre application web, de votre familiarité avec le Framework et des préférences de votre équipe de développement. Il est important de bien se renseigner et de comprendre les fonctionnalités, les avantages et les limitations des différents Framework avant d'en choisir un pour votre projet.

Partie 2 : Partie Pratique

# **CHAPITRE IV : ETUDE PRATIQUE DU PROJET**

## **4 – 1 – Outils utilisés**

Pour créer une application web, nous aurons besoin de plusieurs outils pour différents aspects du développement. A savoir, un environnement de développement intégré (IDE), un langage de programmation, un système de gestion de version et des Framework.

### **4 – 1 – 1 – Environnement de développement intégré (IDE)**

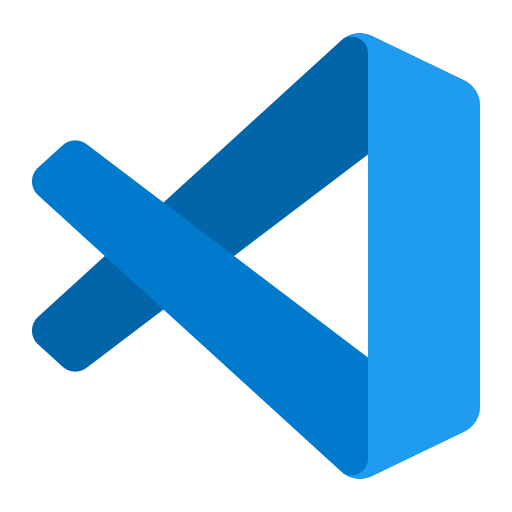
Un IDE (Environnement de Développement Intégré) est un logiciel qui offre un ensemble d'outils pour faciliter le développement de logiciels, y compris les applications web. Un IDE regroupe généralement plusieurs fonctionnalités essentielles pour le développement, telles que l'édition de code, la gestion de fichiers, le débogage, la compilation ou l'exécution de code, la gestion de version, la complétion automatique de code, et bien plus encore. Les IDE sont conçus pour aider les développeurs à être plus productifs et à créer des applications de manière plus efficace.

Voici quelques exemples d'IDE populaires utilisés pour le développement d'applications web :

* **Visual Studio Code** : Il s'agit d'un IDE gratuit et open source développé par Microsoft, qui offre une interface utilisateur conviviale, une grande variété d'extensions et une intégration étroite avec de nombreux langages de programmation couramment utilisés pour le développement web.
* **Sublime Text** : C'est un éditeur de texte avancé qui offre de nombreuses fonctionnalités pour le développement web, comme la coloration syntaxique, la complétion automatique de code et la gestion de projet.
* **JetBrains WebStorm** : C'est un IDE payant spécifiquement conçu pour le développement web, offrant une gamme complète d'outils pour le développement front-end et back-end, ainsi que pour le débogage et le test d'applications web.
* **Eclipse** : C'est un IDE populaire utilisé principalement pour le développement Java, mais il offre également des fonctionnalités pour d'autres langages de programmation couramment utilisés dans le développement web.
* **Atom** : C'est un éditeur de texte open source conçu pour les développeurs web, offrant une interface personnalisable et de nombreuses extensions pour faciliter le développement web.

Il existe de nombreux autres IDE disponibles, et le choix d'un IDE dépend généralement des préférences personnelles du développeur, du langage de programmation utilisé et des besoins spécifiques du projet.

Dans notre cas, on s’intéresse beaucoup à Visual Studio Code pour rédiger et compiler notre code.



***Figure 4 :*** *Icone de Visual Studio Code*

* **Téléchargement et installation de Visual Studio Code**

1. **Téléchargement**

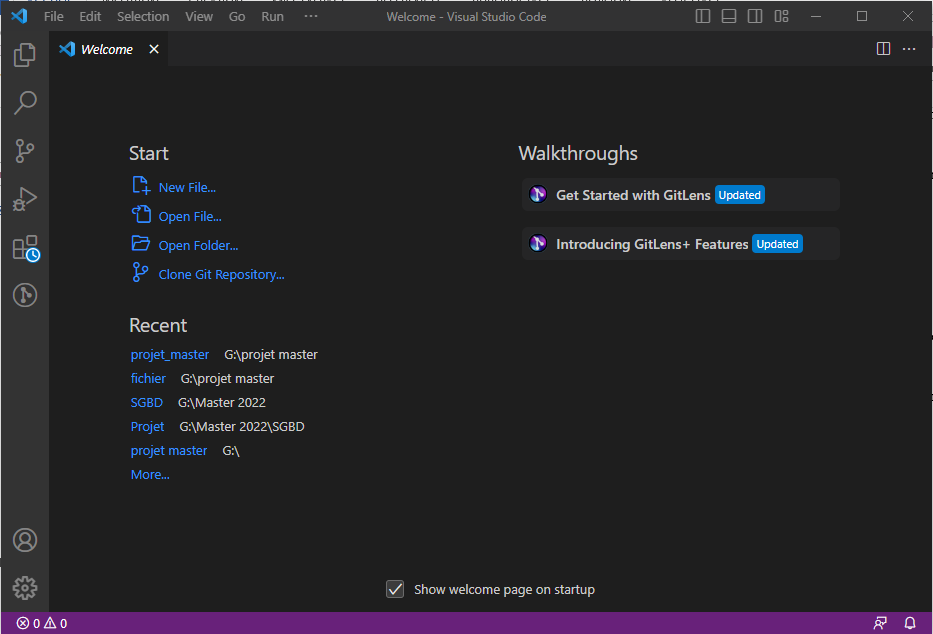
Pour télécharger Visual studio code, il suffit de se rendre sur cette page : <https://code.visualstudio.com/download>

Puis téléchargez la version pour Windows User Installer en fonction de votre processeur : 64-bit, 32-bit ou ARM.

1. **Installation**

Une fois le téléchargement fini, suivez les instructions suivante :

* Exécutez l'installateur.
* Acceptez la licence.
* Choisissez le répertoire d'installation.
* Choisissez les fonctionnalités supplémentaires à ajouter.
* Cliquez sur « install »pour finaliser l'installation.



***Figure 5****: IDE Visual Studio Code*

### **4 – 1 – 2 – Langage de programmation**

La plateforme est une application MERN, c’est-à-dire que cette dernière est entièrement basée sur le langage de programmation JavaScript. Nous allons alors utiliser ces différents Framework, à savoir, React pour le côté client (interface utilisateur) et Node.js pour le côté serveur (back-end).

# **4 – 2 – Conception d’une application MERN**

MERN est un acronyme qui fait référence à un ensemble de technologies utilisées pour développer des applications web modernes. Chaque lettre représente un outil ou un Framework spécifique :

M : MongoDB : une base de données NoSQL (Not Only SQL) orientée documents. Elle est basée sur un modèle de données flexible et est utilisée pour stocker les données dans des documents au format JSON.

E : Express : un Framework web pour Node.js, qui permet de créer des applications web côté serveur. Express facilite la gestion des requêtes HTTP, des routes, des cookies et des sessions, ainsi que la création d'APIs RESTful.

R : React : une bibliothèque JavaScript pour la création d'interfaces utilisateur interactives. React permet de créer des composants réutilisables pour construire des interfaces utilisateur modernes et réactives.

N : Node.js : un environnement d'exécution JavaScript côté serveur basé sur le moteur V8 de Google Chrome. Node.js permet d'exécuter du code JavaScript sur le serveur, ce qui permet de créer des applications web côté serveur et de gérer les requêtes et les réponses.

En combinant ces quatre technologies, MERN permet de développer des applications web à architecture complète (full-stack) en JavaScript. MongoDB est utilisé comme base de données, Express comme framework web côté serveur, React comme bibliothèque pour l'interface utilisateur côté client, et Node.js pour exécuter le code JavaScript côté serveur. MERN est donc une pile de technologies complète pour le développement d'applications web modernes et performantes.

### **4 – 2 – 1 – React**

React.js, communément appelé simplement React, est une bibliothèque JavaScript utilisée pour construire des interfaces utilisateur. Chaque application web React est composée de composants réutilisables qui constituent des parties de l’interface utilisateur, nous pouvons avoir un composant distinct pour notre barre de navigation, un pour le pied de page, un autre pour le contenu principal, et ainsi de suite.

Le fait de disposer de ces composants réutilisables facilite le développement car nous n’avons pas à répéter le code récurrent. Il nous suffit de créer sa logique et d’importer le composant dans n’importe quelle partie du code où il est nécessaire.

React est également une application à page unique. Ainsi, au lieu d’envoyer une requête au serveur à chaque fois qu’une nouvelle page doit être rendue, le contenu de la page est chargé directement à partir des composants React. Cela conduit à un rendu plus rapide sans rechargement de la page.

Dans la plupart des cas, la syntaxe utilisée pour construire des applications React est appelée JSX (JavaScript XML), qui est une extension syntaxique de JavaScript. Cela nous permet de combiner la logique JavaScript et la logique de l’interface utilisateur d’une manière unique. Avec JSX, nous éliminons le besoin d’interagir avec le DOM en utilisant des méthodes comme document.getElementById, querySelector, et d’autres méthodes de manipulation du DOM.

Bien que l’utilisation de JSX ne soit pas obligatoire, elle facilite le développement des applications React.

1. **Pourquoi choisir React**

De nombreux développeurs et organisations ont choisi React plutôt que d’autres bibliothèques ou Framework, voici la raison :

**Facile à apprendre :** React est facile à apprendre et à comprendre tant que vous avez une bonne maîtrise des prérequis. React dispose d’une solide documentation et de nombreuses ressources gratuites créées par d’autres développeurs en ligne via la communauté très active de React.

**Composants réutilisables :** Chaque composant de React possède sa propre logique qui peut être réutilisée partout dans l’application. Cela réduit le besoin de réécrire le même morceau de code plusieurs fois.

**Opportunités d’emploi :** Un plus grand pourcentage d’opportunités de développement web front-end en ce moment ont React comme l’une des compétences requises. Le fait de comprendre le fonctionnement de React et de pouvoir travailler avec lui augmente donc vos chances de décrocher un emploi.

**Performances améliorées :** Grâce au DOM virtuel de React, le rendu des pages peut se faire plus rapidement. En utilisant une bibliothèque de routage comme React Router, vous aurez différentes pages rendues sans aucun rechargement de page.

**Largement extensible :** React est une bibliothèque qui assure uniquement le rendu de l’interface utilisateur de notre application. C’est au développeur de choisir les outils avec lesquels il souhaite travailler, comme les bibliothèques de rendu de différentes pages, les bibliothèques de conception, etc.

1. **Qui utilise React**

React a été utilisé par de nombreux ingénieurs front-end dans des start-ups et des entreprises établies comme Facebook, Netflix, Instagram, Yahoo, Uber, The New York Times, et plus encore.

Bien que toutes les entreprises citées ci-dessus n’aient pas construit l’ensemble de leur produit à l’aide de React, certaines de leurs pages ont été construites avec React. Cela s’explique par les hautes performances, la facilité d’utilisation et l’évolutivité de React.

1. **Caractéristiques de React**

React possède une pléthore de fonctionnalités étonnantes qui continuent à en faire une option populaire pour les développeurs.

Voici quelques-unes des principales fonctionnalités de React :

* JSX : Il s’agit d’une extension syntaxique JavaScript qui étend les fonctionnalités de l’ES6 (ECMAScript 2015). Cela nous permet de combiner la logique et le balisage JavaScript dans un composant.
* DOM virtuel : Il s’agit d’une copie de l’objet DOM qui met d’abord à jour et rend à nouveau nos pages lorsque des modifications sont apportées ; il compare ensuite l’état actuel avec le DOM original pour le maintenir en phase avec les modifications. Cela permet un rendu plus rapide des pages.
* Composants : Les applications React sont composées de différents composants réutilisables qui ont leur propre logique et interface utilisateur respective. Cela le rend efficace pour la mise à l’échelle des applications et le maintien de performances élevées, car vous ne dupliquez pas le code aussi souvent que dans d’autres Framework.

1. **Avantages de l’utilisation de React**

Voici les avantages de l’utilisation de React :

* React est facile à apprendre et à comprendre.
* React dispose d’une communauté très active où vous pouvez contribuer et obtenir de l’aide si nécessaire.
* Il existe de nombreuses opportunités d’emploi pour les développeurs React.
* React s’accompagne d’une augmentation des performances des applications.

1. **Inconvénients de l’utilisation de React**

Voici les inconvénients de l’utilisation de React :

* Les débutants qui n’ont pas une solide compréhension de JavaScript (en particulier ES6) peuvent avoir du mal à comprendre React.
* React est dépourvu de certaines fonctionnalités courantes comme la gestion d’un état unique et le routage ; vous devrez installer et apprendre à utiliser des bibliothèques externes pour les obtenir.

1. **Installation de React**

Avant d’installer React, il nous faut s’assurer que Node.js est installé sur notre ordinateur. Pour vérifier il suffit d’ouvrir un terminal et exécuter le code suivant :



***Figure 6****: Commande d’affichage de version de Node.js*

Cette commande nous affichera la version de Node.js s’il est déjà installé.

Ensuite, nous allons exécuter une autre commande pour commencer à installer React.

Tout d’abord, créez un dossier ou naviguez jusqu’à l’emplacement où vous voulez que l’application React soit installée, puis exécutez la commande ci-dessous dans votre terminal :



***Figure 7 :*** *Commande pour créer une application React*

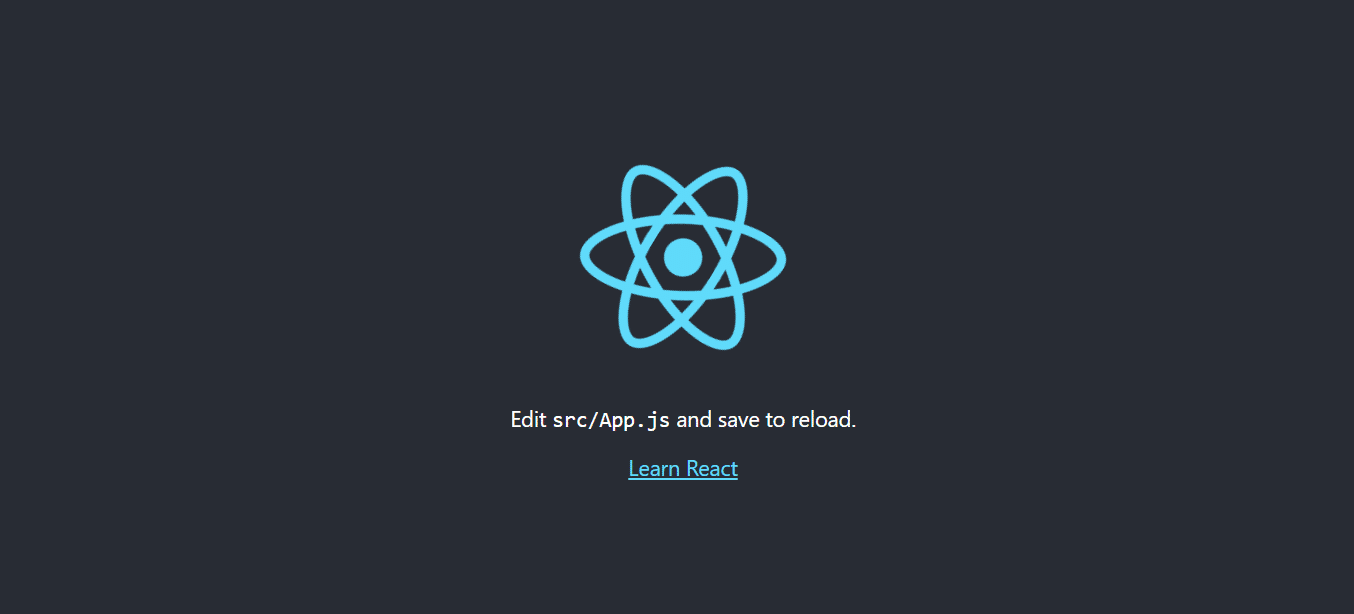
La commande ci-dessus va installer React dans un dossier appelé mon-app.

Une fois l’installation terminée, ouvrez votre dossier React nouvellement installé dans l’éditeur de code de votre choix. Nous utiliserons Visual Studio Code dans ce projet. Visual Studio Code est livré avec un terminal intégré. Si vous comptez utiliser un autre terminal comme Git Bash ou CMD, il vous suffit de naviguer dans le répertoire de votre application React et d’exécuter la commande ci-dessous :



***Figure 8 :*** *Commande pour lancer l’application React*

Cette commande met en route votre serveur de développement. Après un petit moment, vous devriez avoir cette page ci-dessous rendue sur localhost:3000 dans votre navigateur :



***Figure 9 :*** *Première page de démarrage d’une nouvelle application React*

**4 – 2 – 2 – Node.js**

**TABLE DES MATIERES**

[**CHAPITRE I : PRESENTATION DU THEME** 1](#_Toc132262574)

[**1-1-** **Choix de l’objet de la recherche** 1](#_Toc132262575)

[**1-2-** **Objectif** 1](#_Toc132262576)

[**1-3-** **Intérêts de l’objet de la recherche** 1](#_Toc132262577)

[**1 – 4 – Architecture de la plateforme** 2](#_Toc132262578)

[**1 – 5 – Explication de l’architecture** 3](#_Toc132262579)

[ **Authentification** 3](#_Toc132262580)

[ **Interface Utilisateur (UI)** 4](#_Toc132262581)

[ **API (Application Programming Interface)** 5](#_Toc132262582)

[ **Middleware** 5](#_Toc132262583)

[ **Serveur (back- end)** 6](#_Toc132262584)

[**1 – 6 – Gestion des utilisateurs** 7](#_Toc132262585)

[**1 – 7 – Fonctionnalités dans l’application** 7](#_Toc132262586)

[ **Gestion des utilisateurs** 8](#_Toc132262587)

[ **Gestion des communiqués** 8](#_Toc132262588)

[ **Envoi de mail à partir de la plateforme** 8](#_Toc132262589)

[ **Communication et partage de documents entre utilisateurs** 9](#_Toc132262590)

[ **Téléchargement de document** 9](#_Toc132262591)

[**CHAPITRE II : ETUDE THEORIQUE DU THEME** 10](#_Toc132262592)

[**2 – 1 -**  **Application Web** 10](#_Toc132262593)

[**2 – 2 -** **Avantages d’une application web** 10](#_Toc132262594)

[**2 – 3 – Une plateforme web** 10](#_Toc132262595)

[**2 – 4 -** **Différence entre une application web et une plateforme web** 11](#_Toc132262596)

[**2 – 5 -** **Un navigateur web** 12](#_Toc132262597)

[**2 – 6 -** **Un serveur web** 12](#_Toc132262598)

[**2 – 7 -**  **Composant matériels et logiciels d’un serveur web** 12](#_Toc132262599)

[**2 – 8 -**  **Fonctionnement d’un serveur web** 13](#_Toc132262600)

[**2 – 9 -**  **Serveur web statique ou serveur web dynamique** 14](#_Toc132262601)

[**2 – 10 - Fonctionnalités du serveur web** 14](#_Toc132262602)

[**CHAPITRE III : ETAPE DE CREATION D’UN SITE WEB** 16](#_Toc132262603)

[**3 – 1 –** **Langage de programmation** 16](#_Toc132262604)

[**3 - 1 - 1 – HTML** 16](#_Toc132262605)

[**3 – 1 – 2 – CSS** 20](#_Toc132262606)

[**3 – 1 – 3 – Javascript** 21](#_Toc132262607)

[**CHAPITRE IV : ETUDE PRATIQUE DU PROJET** 25](#_Toc132262608)

[**4 – 1 – Outils utilisés** 25](#_Toc132262609)

[**4 – 1 – 1 – Environnement de développement intégré (IDE)** 25](#_Toc132262610)

[ **Téléchargement et installation de Visual Studio Code** 26](#_Toc132262611)

[**4 – 1 – 2 – Langage de programmation** 27](#_Toc132262612)

[**4 – 2 – Conception d’une application MERN** 27](#_Toc132262613)

[**4 – 2 – 1 – React** 28](#_Toc132262614)

[**a)** **Pourquoi choisir React** 29](#_Toc132262615)

[**b)** **Qui utilise React** 29](#_Toc132262616)

[**c)** **Caractéristiques de React** 29](#_Toc132262617)

[**d)** **Avantages de l’utilisation de React** 30](#_Toc132262618)

[**e)** **Inconvénients de l’utilisation de React** 30](#_Toc132262619)

[**f)** **Installation de React** 31](#_Toc132262620)