**CHAPITRE 3 : REALISATION DU PROJET**

* 1. **Introduction**

La phase de réalisation est la partie essentielle durant le développement de notre projet informatique. De ce fait, nous allons présenter les outils et les langages de programmation utilisés pour le développement de ce projet. Ainsi qu’une présentation de la plateforme via ces interfaces graphiques.

* 1. **Langages et Framework de développements**
     1. **Les langages utilisés**
        1. Langage de balise HyperText Markup Language (HTML)

L’HTML signifie « HyperText Markup Language » qu’on peut traduire par « Langage de balises pour l’hypertexte ». Il est utilisé afin de créer et de représenter le contenu d’une page web et sa structure. D’autres technologies sont utilisées avec HTML pour décrire la présentation d’une page et/ou ses fonctionnalités interactives. [3.01]

L’« hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d’un même site web ou entre différents sites web. Les liens sont un aspect fondamental du web. Ce eux qui forment cette « toile », mot traduit par web en anglais. En téléchargeant du contenu sur l’Internet et en les reliant à des pages créées par d’autres personnes. [3.01]

Le HTML a été inventer par Tim Berners-Lee en 1990 alors qu’il travaillait au CERN de Genève. Son objectif principal était de relise des pages entre elles via des liens hypertexte et de rendre les documents lisibles sur tous les ordinateurs indépendamment de leur système d’exploitation. Depuis le HTML a évolué à travers plusieurs versions : HTML en 1991, HTML+, HTML 2, HTML3.2, HTML 4.01 en 1999 et XHTMLen 2000.

En 2004, le Web Hypertexte Application Technologie working Group (WebWG) a été créé pour relancer le développement du HTML, conduisant à la spécification de HTML 5 en 2007. HTML 5 a été officialiser en aout 2012 et continue de s’améliorer. Cette version met l’accent sur la compatibilité avec les contenus existants et l’adaptation aux besoins des utilisateurs grâce a une approche pragmatique des évolutions technologiques [3.02]

* + - 1. Cascading Style Sheet (CSS)

Le Cascading Style Sheets ou le CSS, est un langage de feuille de style utilise pour décrire la présentation d’un document écrit en HTML ou XML. Il permet de séparer le contenu HTML de la présentation, en contrôlant l’apparence des éléments, la mise en page, les polices, les couleurs, etc., sur différents supports comme les écrans, le papier ou la voix

CSS est l’un des langages principaux du Web ouvert et a été standardisé par le World Wide Web Consortium (W3C). Auparavant, le développement des différentes parties de la spécification CSS était réaliser de façon synchrone, permettant d’avoir une version pour l’ensemble de la recommandation. Nous avons donc entendu parler de CSS1, CSS2.1, voire CSS3. Toutefois, il n’y aura pas de version CSS4 ou d’autres version globale numérotée de CSS. [3.03]

* + - 1. Langage de programmation JavaScript (JS)

Le JavaScript souvent abrégé en « JS » est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.JS, Apache CouchDB, voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée. C’est un langage à objets utilisant le concept de prototype, disposant d’un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orienté objet. [3.04]

Le JavaScript est un langage de programmation informatique open source créée en 1995 par Brendan Eich, un programmeur travaillant pour Netscape Communication Corporation. Crées-en seulement 10 jours, le JavaScript est appelé à l’origine « Mocha », mais a ensuite été renommé « LiveScripté » avant de se fixer finalement sur JavaScript, dans une démarche stratégique visant à surfer sur la popularité de Java à l’époque. [3.05]

Avec HTML et CSS, le JavaScript représente en fait l’un des trois principaux langages de programmation utilisé dans le développement web. [3.05]

1. JavaScript moderne (à partir de 2015)

Le JavaScript continue d’évoluer grâce à des mise à jour régulières de la spécifications ECAMScript. Les fonctionnalités introduites dans les version suivantes (ES2016, ES2017, etc…) comprennent async/await, de nouvelles méthodes et une syntaxe améliorée. Ces dernières années, avec le développement de cadres et de bibliothèques puissants pour les développements frontaux et dorsaux, l’écosystème JavaScript s’est considérablement développé. [3.05]

1. L’utilisation de JavaScript aujourd’hui

De nos jours, JavaScript est un langage de programmation d’une impressionnante polyvalence qui peut être déployé dans divers aspects du développement web. Sa flexibilité et son large éventail de cadres le rendent idéal pour les types de développement web suivants [3.05] :

* Développement de l’interface utilisateur (UI) : dans le cadre du développement web coté client, JavaScript est essentiel pour créer des interfaces utilisateur dynamiques et interactives. Il est utilisé pour manipuler le Document Object Model ou DOM et répondre aux actions de l’utilisateur, afin d’offrir une expérience transparente et attrayante.
* Développement web coté serveur : avec l’introductions de Node JS, JavaScript peut être utiliser cote serveur, ce qui permet aux développeurs de créer des applications serveur évolutives et performantes. Cela permet d’utiliser un langage unifié pour le développement coté client et cote serveur.
* Développement complet : JavaScript est un composant clé des piles MEAN qui signifie MongoDB, Express.JS, Angular.JS, Node.JS et MERN, MongoDB, Express.JS, React.JS, Node.JS, qui sont des choix populaires pour le développement à pile complète. Ces piles permettent aux développeurs d’utiliser JavaScript dans l’ensemble de l’application, de la base de données au frontend.
* Développement d’application mobile : dans des cadres comme React Native et NativeScript, JavaScript est utiliser pour créer des applications mobiles multiplateformes. Cela permet aux développeurs d’écrire du code une seule fois et de le déployer sur les plateformes IOS et Android.
* Applications à page unique (SPA) : JavaScript est un langage de base pour la construction de SPA, ou une seule page HTML est dynamiquement mise à jour lorsque l’utilisateur interagit avec l’application. Des framework comme Angular.JS, React.JS et Vue.JS simplifient le développement des SPA en fournissant des architectures structurées.
* API Web et intégration : couramment utilisé pour interagir avec des API externes, JavaScript permet l’intégration de services et de données tiers dans les applications web. Cela est essentiel pour créer des applications riches en fonctionnalité et axées sur les données.
* Développement de jeux sur le web : pour créer des jeux sur le web, JavaScript peut être déployé en combinaison avec HTML 5 et CSS3. Des bibliothèques comme Phaser et Three.js facilitent la création de jeux interactifs et visuellement attrayantes directement dans le navigateur.
  + 1. **Les Framework utilisés**
       1. Vue JS

La framework Vue.JS est un framework évolutif pour construire des interfaces utilisateur. A la différence des autre framework monolithiques, Vue a été conçu et pense pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque se concentre uniquement sur la partie vue, et il est vraiment simple de l’intégrer avec d’autres bibliothèque ou projets existants. D’un autre côté, Vue est tout à fait capable de faire tourner des applications web mono-pages quand il est couplé avec des outils modernes et des bibliothèques complémentaires. [3.06]

1. Historique

Vue.JS a été créé à l’origine par une seule personne, Evann You, ancien ingénieur de Google ayant notamment travaillé avec Angular.JS puis sur le framework Meteor. Après avoir expérimenté une mécanique de réactivité qu’il jugeait plus intéressante que celle d’Angular, il décide de publier ses premiers résultats en juillet 2013. Cinq ans plus tard, Vue.JS est le 3ème projet sur GitHub en nombre de stars et Evan enchaine les conférences à travers le monde.[3.07].

1. L’équipe de développement

Vue dispose aujourd’hui d’une équipe internationale d’une trentaine de personnes, constituée de contributeurs bénévoles qui se sont formé avec les années. La décentralisation complétée de l’équipe est à la fois une contrainte et une force qui lui a permis de diffuser le framework beaucoup plus rapidement à plusieurs endroits à la fois. [3.07].

1. Comparaison avec les autres framework

Tâchons de comparer Vue avec les deux autres framework JS les plus populaires, React et Angular. [3.07].

Premièrement, les points communs entre ces 3 framework :

* Très populaires, utilisés par de grosses entreprises
* Mature, stables, support à long terme financé
* Codebase orienté composants
* Paradigme principalement déclaratif et non impératif
* Adapté aux stacks modernes (ES6+ / TypeScript)
* Large écosystème de composants et d’outillage

Deuxièmement, le positionnement : React, Vue et Angular se positionnent dans des catégories différentes et cela s’observe dès les premières lignes d’introduction. React se définit comme une bibliothèque indépendante du reste de la stack technique, Angular se définit comme le framework unique répondant à tous les besoins, desktop comme mobile. Tandis que Vue se positionne entre deux comme un framework progressif, polyvalent et que l’on peut adopter par étapes. Un juste milieu entre une bibliothèque et un framework tout équipé.

Enfin une comparaison de langage, style et stack technique

|  |  |  |  |
| --- | --- | --- | --- |
|  | React | Vue.js | Angular |
| Langage | JSX, (TypeScript) | TypeScript, JSX | TypeScript |
| Gestion d’état | State centraliser non mutable (flux) | State centraliser en option (Vuex), sinon interne et mutable | State interne et mutable, pas de state centraliser officiel |
| Gestion de la réactivité | Manuelle (setState + VDOM diffing) | Automatique (observers/Proxies) | Automatique (Zones/dirty-checking) |
| Stack | Ne se suffit pas à lui-même, à intégrer dans une stack web avec d’autres outils en externe. Ecosystème riche | Quelques outils officiels maintenus par la team Vue, d’autres à chercher en externe. Promeut des solutions tierces si elles sont appropriées | Framework full stack et auto suffisant, avec options de sécurité incluses nativement. Complet mais plus fermé. |

State : objet de données représentant l’état de l’application ;

Interne : propre à chaque composant

* + - 1. Express JS

Express.JS, parfois aussi appelé « Express », est un framework backend Node.JS minimaliste, rapide et de type Sinatra qui offre des fonctionnalités et des outils robustes pour développer des applications backend évolutives. Il vous offre le système de routage et des fonctionnalités simplifiées pour étendre le framework en développant des composants et des parties plus puissants en fonctions des cas d’utilisations de notre application. [3.08]

C’est de ce fait le framework standard pour le développement de serveur en Node.JS. L’auteur original, TJ Holowaychuck, le décrit comme étant un serveur inspiré de Sinatra dans le sens qu’il est relativement minimaliste tout en permettant d’étendre ses fonctionnalités via des plugins.

Express fournit un ensemble d’outils pour les applications web, les requêtes et les réponses http, le routage et les intergiciels permettant de créer et de déployer des applications à grande échelle, prêtes pour l’entreprise. Il fournit également un outil d’interface de ligne de commande appelé Node Package Manager ou NPM, où l’on peut s’approvisionner en paquets développés. [3.08]

Notamment utilisé dans un large éventail de choses dans l’écosystème JavaScript/Node.JS, nous pouvons également développer des applications, des points de terminaison API, des systèmes de routage et des frameworks avec lui.

Voici quelques-uns des types d’applications créer avec Express.JS :

* Applications à page unique
* Outils de collaboration en temps réel
* Applications de streaming
* Applications Fintech
  + - 1. Socket.IO

Le Socket.IO est une bibliothèque JavaScript open source qui fournit une solution pour la communication en temps réel entre un client et un serveur dans les applications web modernes. Créée par Guillermo Rauch en 2010, elle est développée en continu par une communauté de contributeurs.

Couramment utilisée dans les applications web qui nécessitent une communication en temps réel, telles que les applications de messagerie instantanée, les jeux en ligne et les tableaux de bord en temps réel. Socket.IO fonctionne avec des technologies web telles que Node.JS et les WebSockets pour offrir une communication bidirectionnelle efficace entre un client et un serveur.

Contrairement au WebSockets, Socket.IO n’est pas une spécification officielle, elle a donc sa propre API. Toutefois, elle prend en charge les WebSockets et d’autres protocoles de communication en temps réel tels que Server-Sent Event ou SSE et Long Polling, selon la disponibilité du navigateur et du serveur. [3.09]

Le Socket.IO offre plusieurs avantages par rapport à WebSockets :

* Socket.IO diffuse un message à tous les clients connectés.
* Les proxys et les répartiteurs de charge rendent la mise en œuvre et l’échelle des WebSockets difficiles. Socket.IO prend en charge ces technologies en standard.
* Comme mentionné précédemment, Socket.IO peut passer à d’autres technologies que WebSockets lorsque le client ne le prend pas en charge.
* Lors des interruptions de connexion, WebSockets ne se reconnectera pas automatiquement tandis que Socket.IO le fait grâce à un mécanisme de ping/pong, qui vérifie périodiquement l’état de la connexion.
  + - 1. Tailwind CSS

Tailwind CSS est un framework utility-First CSS avec des classes prédéfinis que nous pouvons utiliser pour construire et concevoir des pages web directement dans votre balisage. Il nous permet d’écrire du CSS dans notre HTML sous la forme de classes prédéfinies. [3.10]

Créer en 2017 par Adam Wathan, un homme passionné par le développement web et est à l’origine de nombreux projet tels que Laravel Valet, Jigsaw et bien sûr Tailwind CSS. C’est en travaillant sur un projet nommé KiteTail que lui vient l’idée de créer le framework Tailwind CSS. Avec ce framework, il est possible de créer un design d’interface au sein même du fichier HTML. Cette façon de programmer n’interfère pas avec les pratiques recommandées par le W3C comme celle de séparer le HTML des feuilles de styles CSS. Pour utiliser Tailwind, il faut prendre les classes CSS prédéfinies par le framework en les appelant dans un fichier HTML comme ceci : [3.11]

* 1. **Logiciels utilisés durant le développement**
     1. **Editeur de code Visual Studio Code**

Visual Studio Code est un éditeur de code source léger, mais puissant, qui s’exécute sur notre bureau et est disponible pour Windows, MacOs et Linux. Il est livré avec la prise en charge intégrée pour JavaScript, TypeScript et Node.JS et dispose d’un riche écosystème d’extensions pour d’autres langages et environnements d’exécution tels que C++, C#, Java, Python, PHP, Go, .Net [3.12].

Il offre de nombreuses fonctionnalités à l’utilisateur comme la coloration syntaxique, l’auto-complétions, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l’intégration avec Git. Grâce à l’aide d’une grande variété d’extensions développées par la communauté, l’éditeur permet aux développeurs de personnaliser ce dernier selon leurs besoins.

* + 1. **Visual Paradigm**

Visual Paradigm est un outil complet qui intègre de nombreuses fonctionnalités et modules essentiels pour la modélisation et le développement de logiciels. Voici quelques-unes des principales caractéristiques [3.13] :

* Modélisation UML : Visual paradigm prend en charge tous les diagrammes UML standard, y compris les diagrammes de cas d’utilisation, de classes, de séquence, d’activité et d’état.
* Modélisation de processus métier : Grace à la notation BPMN, nous pouvons modéliser, documenter et optimiser des processus métier
* Conception de base de données : Créer des modèles de donnée conceptuels logiques et physiques à l’aide de diagrammes entité-relation ou ERD.
* Architecture d’entreprise : utiliser ArchiMate pour modéliser et documenter l’architecture d’entreprise, y compris les aspects métier, applicatifs et technologiques
* Gestion des exigences : capturer, organiser et gérer les exigences de nos projets de manière centralisée.
* Génération de code : générer automatiquement du code source à partir de nos modèles UML, économisant ainsi du temps et des efforts.
* Développement agile : panifier et suivit de nos sprints, créez des backlog de produits et de sprints, et gérer les taches de manière efficace.

Le Visual paradigm se distingue par sa capacité à s’intégrer a d’autres outils et plateformes populaires, tels que Git, Jira, Confluence et Microsoft Office. Cette interopérabilité facilite la collaboration au sein des équipes et améliore l’efficacité globale du processus de développement [3.13].

* + 1. **GitHub**

GitHub est un plateforme web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu’à suivre et contrôler les modifications qui lui sont apportée.

Créer en 2008 par Tom Preston-Werner, Chris Wanstrath et PJ Hyett. Preston-Werner a expliqué que l’idée de créer GitHub était venue de l’utilisation de Git. A l’époque, il n’y avait pas de plateforme en ligne permettant de stocker et de partager du code avec d’autre développeurs. [3.15]

GitHub se base sur deux principes : [3.14]

* Contrôle de version : le contrôle de version aide les développeurs à suivre et à gérer les modifications apportées au code d’un projet logiciel. Au fur et à mesure qu’un projet prend de l’ampleur, le contrôle de version devient essentiel.
* Git : c’est un système de contrôle de version open-source spécifique crée par Linus Torvalds en 2005. Concrètement, git est un système de contrôle de version distribué, ce qui signifie que l’ensemble de la base du code et de l’historique est disponible sur l’ordinateur de chaque développeur. Ce qui permet des branchements et une fusion facile.
  + 1. **Logiciel Node.JS**

Node JS est une plateforme de développement JavaScript. Ce n’est pas un serveur, ce n’est pas un framework, c’est juste le langage JavaScript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier. [3.16]

On peut souvent le confondre avec un serveur car c’est son origine : Node.JS a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel ou le serveur est capable de pousser de l’information au client. C’est dans ce but qu’il utilise la bibliothèque libuv pour réaliser son modèle d’entrer sortie non bloquante. [3.16]

Node.JS présente de nombreux intérêts :

* Logiciel libre sous la licence MIT
* Performance du moteur V8
* Modèle non bloquant
* Communauté très active
* Système de paquet intégrer : NPM

Grâce a une vaste liste de fonctionnalités, Node JS a connu une croissance rapide au cours de ces dernière année : [3.17]

* Facile : Easy-Node.JS est assez facile à prendre en main. C’est un choix incontournable pour les débutants en développement web.
* Evolutif : il offre une grande évolutivité aux applications. Comme Node.JS est un single-thread, il est capable de gérer un grand nombre de connexions simultanées avec un débit élevé.
* Vitesse : l’exécution non bloquante des threads rend Node.JS encore plus rapide et plus efficace
* Paquets : un vaste ensemble de paquets Node.JS open source est disponible et peut simplifier notre travail
* Backend solide : Node.JS est écrit en C et C++, ce qui le rend rapide et ajoute des fonctionnalités comme le support réseau.
* Multi-plateforme : la prise en charge multi-plateforme nous permet de créer des sites web SaaS, des applications de bureau et même des applications mobiles, le tout en utilisant Node.JS
* Maintenable : Node.JS est un choix facile pour les développeurs, car le frontend et le backend peuvent être gères avec JavaScript comme un seul langage.
  + 1. **Logiciel Xampp X(cross) Apache MariaDB Perl PHP**

Le logiciel Xampp est un logiciel de développement web permettant de mettre en place un serveur web local, un serveur FTP et un serveur de messagerie électronique qui fonctionne sous tous types de systèmes d’exploitation comme Windows, Linux et Mac. Il contient du MySQL que nous allons utiliser comme base de données pour notre

* 1. **Système de gestion de base de données : MySQL**

Pour bien réaliser notre plateforme, nous avons besoin d’une base de données que nous utiliserons en tandem avec les autre frameworks et logiciels précédemment mentionnés. Cette base nous servira aussi à stocker les informations qui seront communiquer aux utilisateurs de notre plateforme. La base de données que nous utiliserons sera le MySQL.

MySQL est un système de gestion de base de données relationnelles SQL ou Structured Query Language open source développé et supporté par Oracle. [3.18]

MySQL est un SGBDR open source qui utilise SQL pour créer et gérer des bases de données. En tant que base de données relationnelle, MySQL stocke les données dans des tables de lignes et de colonnes organisées en schémas. Un schéma définit l’organisation et le stockage des données et décrit la relation entre les différentes tables. Avec ce format, les développeurs peuvent facilement stocker, extraire et analyser de nombreux types de données, y compris du texte simple, des chiffres, des dates, des heures et plus récemment, des vecteurs et JSON [3.19]

Nous avons choisi d’utiliser MySQL pour diverses raisons :

* La rapidité : le serveur MySQL est très rapide
* La facilité d’utilisation : MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux ;
* API diverse : on peut effectuer diverse opération sur une base MySQL en utilisant des interfaces écrites en C, Perl, C++, Java, Python, PHP ;
* Les connexion et sécurité : MySQL dispose d’un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.
  1. **Interface graphique**

Dans cette partie du chapitre, on va présenter des aperçus de la plateforme.

* + 1. **Page d’accueil d’UniSphere**
    2. **Page de création de compte**
    3. **Page d’authentification**
    4. **Dashboard étudiant**
    5. **Liste des matières**
    6. **Page d’une matière**
    7. **Dashboard enseignant**
    8. **Liste de classe**
    9. **Page d’une classe**
    10. **Dashboard administrateur**
    11. **Page de gestion des annonces**
    12. **Page de gestion d’étudiant et enseignant**
    13. **Page d’annonces officiels**
  1. **Conclusion**

Pour conclure, comme présenter dans ce chapitre, nous avons pu expliquer les langages de programmation, outils et logiciel que nous utilisons durant le développement de notre plateforme. Et aussi des illustrations des aperçus de la plateforme UniSphere qui est notre projet. Ainsi dans le prochain chapitre nous allons effectuer à évaluation de notre projet et ces perspectives d’avenir.