**Chapitre 02 : généralité et environnement de développement**

1. **Introduction**

Dans ce chapitre, nous allons définir quelques généralités portant sur le développement web. Ensuite, nous allons présenter les langages, les environnements de développements et les SGBD ainsi que les approches de gestions de notre projet.

1. **Définitions et généralités.**
   1. **Plateforme web**
   2. **Logiciel**
2. **Les technologies et langages de développement**

Dans un esprit de défi et de vouloir mettre en œuvre les connaissances qu’on a acquises durant notre formation, nous avons choisi un ensemble de langage et des Framework qui sont très utilisés, notamment par un grand nombre de programmeurs professionnels ce qui en fait des langages de haut niveau.

* 1. **HTML**

HTML (Le HyperText Markup Language) est un langage de balisage, il nous permet de structurer et de mettre en forme le contenu des pages web, d’inclure des ressources multimédias exemple les formulaires de saisie, les images et les programmes informatiques. Il est souvent associé à des langages de programmation tels que le JavaScript, et des formats de présentation à savoir les feuilles de style en cascade.

* 1. **CSS**

Le CSS (Cascading Style Sheets) est un langage informatique coté client utilisé pour mettre en forme les fichiers HTML ou XML. Il permet de séparer la présentation d’une page html et sa structure

* 1. **SASS**

SASS est un langage de script préprocesseur compilé en CSS à l’aide d’une commande SASS. Il nous permet d’utiliser des variables, des règles imbriquées, des fonctions **etc.**

SASS il nous aide à garder les feuilles de style volumineuses bien organisées. Il est disponible en deux syntaxes :

* La syntaxe originale qui utilise extension **.sass**
* La nouvelle syntaxe, « SCSS », qui utilise l’extension **.scss**
  1. **Bootstrap**

Bootstrap est un Framework CSS populaire pour développer des platforms Web réactif. Il fournit des outils très puissants qui nous aide à développer très rapidement. On peut citer quelques composants comme les boutons, les formulaires, offcanvas. Etc.

* 1. **JavaScript**

JavaScript est un langage de script qui permet d’implémenter des programmes complexes sur les pages web, ces scripts peuvent être Écrits directement sur une page web HTML et exécutés automatiquement quand la page Se charge.

* 1. **Type Script**

Type script est un langage de programmation développé par Microsoft, il nous permet de spécifier les types de données transmises dans le code. Et de signaler des erreurs lorsque les types ne correspondent pas au moment de compilation de code, à la fin le code sera transpilé en JavaScript à l’aide d’un compilateur**.**

* 1. **ANGULAR et Angular matériel** 
     1. **Angular**

Angular est un Framework moderne open source basée sur le Type script développé par Google. Permet de crée des applications évolutives à page unique SPA (single page application). Il est basé sur l’architecture MVC qui permet de séparer les données.

* + 1. **Angular matériel**

Angular Matériel est une bibliothèque d’interface utilisateur (UI), qu’on peut utiliser dans les projets Angular pour accélérer le développement. Il nous offre des composants d’interface utilisateur magnifiques tels que des cartes, des entrées, des tableaux de données, des sélecteurs de date et bien plus encore.

* + 1. **Notre choix de Angular par rapport à REACT :**
* Angular est utilisé avec Type Script.
* Les services de langage Angular se complètent automatiquement à l’intérieur des fichiers HTML externes du composant.
* Documentation détaillée.
* Structure et architecture spécialement créées pour une meilleure scalabilité du projet.
* Étant le plus ancien Framework disponible, Angular bénéficie d’un large soutien de la communauté.
  1. **ELECTRONJS**

ElectronJs est un module Node.js qui permet de créer des applications desktop (logiciel) multi-platform (Win, Mac, Linux) en utilisant des technologies web telles Que HTML5, CSS et JavaScript. ElectronJs repose entièrement sur des normes Web connu par la majorité des développeurs Web. De plus, il fournit diverses fonctionnalités de base telles que la mise à jour automatique, le reporter de crash, le créateur du programme d’installation et des fonctionnalités spécifiques au système

* 1. **Framework Ionic**

Ionic est un Framework open-source créé en 2013 permet de créer un code multi-support en utilisant des outils Web comme HTML, CSS, JavaScript, afin de générer des applications iOS, Android, Chrome, Windows Phone et bien d’autres.

* 1. **Spring boot**

Spring Boot est un Framework Java, utilisé pour développer des applications Web. Il nous permet de créer des API REST avec des configurations minimales. Spring Boot fournit un code passe-partout avec toutes les configurations nécessaires pour commencer immédiatement le codage.

1. **Les technologies et langages de modélisation**

Nous avons choisi UML pour bien vouloir mettre en œuvre les connaissances qu’on a acquises durant notre formation. De plus, il est très utilisé, notamment par un grand nombre de concepteur par rapport ou autre langage de modélisation.

* 1. **UML**

L’UML (Unifined Modeling Language) est un langage de modélisation graphique et textuelle, destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes et concevoir des solutions.

1. **Les systèmes de gestion de bases de données**
   1. **Server web**
   2. **Mongo DB (…………)**
2. **Les environnements de développement**

Nous avons choisi ces logiciel grâce au service qu’ils fournissant

* 1. **XAMPP**
  2. **Visual Studio code**

Visual Studio Code est un éditeur de code open source léger et puissant, qui possède un riche écosystème d’extensions. Il est livré avec un support intégré pour JavaScript, Type Script et Node.js.

* 1. **Spring Tools**
  2. **Visual Paradigm**

Visual Paradigm est un logiciel qui permet aux programmeurs de mettre En place des diagrammes. Il possède plusieurs options permettant une large possibilité de modélisation en UML.

1. **Les normes et la sécurité**
2. **Démarche pattern de développement et approche de gestion de projets**
   1. **Approche de gestion de projet**
   2. **L’architecteur MVC**
3. **Conclusion**

Dans ce chapitre, nous avons réussi à définir les différents concepts informatiques qui nous ont semblé essentiels à la réalisation de notre projet.

Dans le chapitre qui se suit nous allons détailler la conception de notre travail.

Table des matières

[**1-** **Introduction** 1](#_Toc118459344)

[**2-** **Définitions et généralités.** 1](#_Toc118459345)

[**3-** **Les technologies et langages de développement** 1](#_Toc118459346)

[**3.1.** **HTML** 1](#_Toc118459347)

[**3.2.** **CSS** 1](#_Toc118459348)

[**3.3.** **SASS** 2](#_Toc118459349)

[**3.4.** **Bootstrap** 2](#_Toc118459350)

[**3.5.** **JavaScript** 2](#_Toc118459351)

[**3.6.** **Type Script** 2](#_Toc118459352)

[**3.7.** **ANGULAR et Angular matériel** 2](#_Toc118459353)

[**3.8.** **ELECTRONJS** 3](#_Toc118459354)

[**3.9.** **Framework Ionic** 3](#_Toc118459355)

[**3.10.** **Spring boot** 3](#_Toc118459356)

[**4-** **Les technologies et langages de modélisation** 3](#_Toc118459357)

[**4.1.** **UML** 3](#_Toc118459358)

[**5-** **Les systèmes de gestion de bases de données** 3](#_Toc118459359)

[**5.1.** **Server web** 3](#_Toc118459360)

[**5.2.** **Mongo DB (…………)** 3](#_Toc118459361)

[**6-** **Les environnements de développement** 3](#_Toc118459362)

[**6.1.** **XAMPP** 3](#_Toc118459363)

[**6.2.** **Visual Studio code** 3](#_Toc118459364)

[**6.3.** **Spring Tools** 3](#_Toc118459365)

[**6.4.** **Visual Paradigm** 3](#_Toc118459366)

[**7-** **Les normes et la sécurité** 3](#_Toc118459367)

[**8-** **Démarche pattern de développement et approche de gestion de projets** 4](#_Toc118459368)

[**8.1.** **Approche de gestion de projet** 4](#_Toc118459369)

[**8.2.** **L’architecteur MVC** 4](#_Toc118459370)

[**9-** **Conclusion** 4](#_Toc118459371)

MVC

SPA