Réalisé par : **Guerfi Imen**

Encadrant : **Angoud Houssem**

Société IT Gate

Sousse

Rapport de stage

Du 02 juillet au 31 juillet

2018/2019

Remerciement

Je tiens à remercier toute ma reconnaissance aux personnels de centre de formation IT Gate Training Groupe, pour l’expérience enrichissante et pleine d’intérêt qu’elles m’ont fait vivre durant ce mois au sien de la société IT Gate Sousse. Cette expérience sera très importante pour ma carrière et les tâches auxquelles vous m’avez associé m’ont vraiment permis de consolider mes connaissances et d’en développer de nouvelles

Je remercie également Madame **Mariam Fatnassi**, mon encadrant, pour m’avoir encadré durant ce stage, pour sa confiance et sa disponibilité.

Je remercie plus particulièrement Monsieur **Angoud Houssem** de m’avoir guidé durant mon stage et m’aider avec une grande sincérité et gratitude.

Merci pour tous

***Présentation de la société***

IT Gate est une société d’ingénierie informatique, crée en 2015 et spécialisée en développement et services informatiques : L’expérience utilisateur, la conception, le développement et le webdesign, et avec les valeurs : Organisation, collaboration, confiance et responsabilité. IT Gate est une équipe jeune et dynamique réunie pour répondre aux besoins de ses clients et les accompagner tout au long de leurs projets.

**Sommaire**

[Introduction 1](#_Toc20258135)

[Chapitre I : Généralités sur le projet 2](#_Toc20258136)

[1 Le web 2](#_Toc20258137)

[1.1 Langages client (front end) 2](#_Toc20258138)

[1.2 Langages serveur (back end) 2](#_Toc20258139)

[1.3 HTTP : HyperText Transfert Protocol 3](#_Toc20258140)

[2 Les architectures web 3](#_Toc20258141)

[2.1 L’architecture client/serveur 3](#_Toc20258142)

[2.2 Architecture à 3 niveaux 3](#_Toc20258143)

[2.2.1 Model MVC 4](#_Toc20258144)

[2.2.2 Model MVVM 4](#_Toc20258145)

[3 Les différents web 5](#_Toc20258146)

[4 Description générale de Travail réalisé 5](#_Toc20258147)

[Chapitre2 Analyse et Conception 7](#_Toc20258148)

[1 Méthodologie de conception 7](#_Toc20258149)

[2 Concept et architecture 7](#_Toc20258150)

[3 Spécification des besoins 8](#_Toc20258151)

[3.1 Diagramme de cas d’utilisation 8](#_Toc20258152)

[3.1.1 Définition 8](#_Toc20258153)

[3.1.2 Diagramme de cas d’utilisation de site 8](#_Toc20258154)

[3.1.3 Les acteurs de notre projet 8](#_Toc20258155)

[3.2 Diagramme de classes 9](#_Toc20258156)

[3.2.1 Définition 9](#_Toc20258157)

[3.2.2 La composition d’un diagramme de classes 10](#_Toc20258158)

[3.2.3 Diagramme de classes de site 10](#_Toc20258159)

[Chapitre3 : Réalisation et implémentation de projet 12](#_Toc20258160)

[1 Environnement de travail 12](#_Toc20258161)

[1.1 Environnement Logiciel 12](#_Toc20258162)

[1.1.1 Langage de programmation 12](#_Toc20258163)

[1.1.2 Environnement de développement 12](#_Toc20258164)

[1.1.3 Serveur d’application 13](#_Toc20258165)

[1.1.4 Système de gestion de base de données 13](#_Toc20258166)

[1.1.5 Framework 13](#_Toc20258167)

[1.1.6 Logiciel de test 14](#_Toc20258168)

[2 Démonstrations des interfaces 14](#_Toc20258169)

[2.1 Partie Admin 15](#_Toc20258170)

[2.1.1 Fenêtre d’Authentification 15](#_Toc20258171)

[2.1.2 Liste d’administrateurs 16](#_Toc20258172)

[2.1.3 Liste des utilisateurs 18](#_Toc20258173)

[2.1.4 Liste des cours 18](#_Toc20258174)

[2.2 Partie Client 20](#_Toc20258175)

[2.2.1 Fenêtre d’accueil 20](#_Toc20258176)

[2.2.2 Fenêtre d’Authentification 21](#_Toc20258177)

[2.2.3 Liste des cours 22](#_Toc20258178)

[2.2.4 Page de contact 23](#_Toc20258179)

[Conclusion 24](#_Toc20258180)

**Liste des figures**

[**Figure 1 : Communication entre le front end et le back end** 9](#_Toc20256184)

[**Figure 2 : Modèle MVC** 10](#_Toc20256185)

[**Figure 3 : Modèle MVVM** 10](#_Toc20256186)

[**Figure 4 : diagramme de cas d’utilisation de site** 15](#_Toc20256187)

[**Figure 5: diagramme de classe de site** 16](#_Toc20256188)

[***Figure 6 : Interface Authentification*** 22](#_Toc20256189)

[**Figure 7 : Message d’erreur** 22](#_Toc20256190)

[**Figure 8 : Liste d’administrateurs** 23](#_Toc20256191)

[**Figure 9: Liste d’administrateurs** 23](#_Toc20256192)

[**Figure 10 : fenêtre des détails d’un administrateur** 24](#_Toc20256193)

[**Figure 11 : fenêtre d’email** 24](#_Toc20256194)

[**Figure 12: Liste d’utilisateurs** 25](#_Toc20256195)

[**Figure 13: Liste de cours** 26](#_Toc20256196)

[**Figure 14: Liste de cours** 26](#_Toc20256197)

[**Figure 15 : fenêtre d’ajout** 27](#_Toc20256198)

[**Figure 16: fenêtre de détails d’un cours** 27](#_Toc20256199)

[**Figure 17 : page d’accueil** 28](#_Toc20256200)

[**Figure 18 : Page Authentification & d’inscription** 28](#_Toc20256201)

[**Figure 19 : Liste de cours** 29](#_Toc20256202)

[**Figure 20 : Liste de cours** 30](file:///D:\FormationITGate\projet\Rapport\rapport.docx#_Toc20256203)

[**Figure 21 : Liste de cours** 30](file:///D:\FormationITGate\projet\Rapport\rapport.docx#_Toc20256204)

**Liste des abréviations**

**HTML :** Adénosine Triphosphate

**HTTP:** HyperText Transfert Protocol

**CSS:** Cascading Style Sheet

**JS**: JavaScript

**TS**: Type Script

**JEE :** Java Enterprise Edition

**SPA :** Single Page Application

**SGBD :** Système de Gestion de bases de données

**JDBC:** Java DataBase Connectivity

**REST:** REpresentational State Transfer

**API:** Application Protocol interface

**UML:** Unified Modeling Language

**IDE:** Integrated Development Environment

**JSON:** Java Script Object Notation

**MVC:** Model View Controller

**MVVM:** Model View ViewModel

Introduction

Pendant un mois, du 2 à 31 juillet 2019 j’ai effectué un stage au sein de l’entreprise « **IT GATE** **TRAINING GROUP SOUSSE »**. Au cours de ce stage j’ai pu m’intéresser à connaître le processus du travail du centre, à s’intégrer dans la vie professionnelle et surtout à améliorer mes connaissances théoriques avec l’aspect pratique dans le domaine de développement informatique

Le travail exposé dans ce rapport consiste à la conception et la réalisation d'un site web éducatif. Ce site offrira un ensemble de services à la communauté universitaire dont : Le téléchargement des documents (cours, séries d’exercices, travaux pratiques, des examens, des articles scientifiques voire des projets), faire des commandes sur le projet souhaité ou bien demander autres services (des tutoriels, des vidéos explicatives, création d’un projet bien déterminé, etc.)

Ce rapport est organisé autour de trois chapitres :

Le premier chapitre définit les généralités sur l'internet, le web et les sites web.

Le second chapitre est consacré à l'analyse et la conception de notre site.

Le troisième chapitre est dédié à la réalisation de notre application.

Chapitre I : Généralités sur le projet

Introduction

En quelques années, Internet et son utilisation s'est entièrement emparé de notre vie au quotidien. Ce réseau a permis la communication entre les ordinateurs et a offert un grand nombre de services dont le plus connu est le web. Ce dernier est considéré comme un outil de communication, de recherche d'information, et de création de nouveaux services.

Ce chapitre présente d'abord une introduction au web ainsi que son évolution au cours des années, puis définit quelques généralités sur notre projet.

1. Le web

Le web est un service de consultation de documents sur Internet, le plus connu et le plus utilisé. C'est le service d'Internet qui a contribué le plus à sa popularité. Pour construire un site web, on a recours à des langages qui servent en quelque sorte à établir les plans d'architecte dont on a besoin pour construire les sites web.

* 1. Langages client (front end)

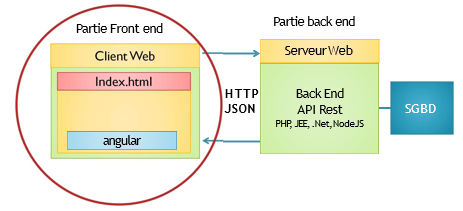
Les langages client ou encore les langages front-end sont indispensable dans la création des sites web, car ils sont lus par les machines des clients. Parmi ces langages, on peut citer le HTML, CSS, JS, etc. Ces langages permettent de créer les interfaces nécessaires des différentes pages Web. Dans notre projet on utilise l’Angular qui est un Framework pour créer la partie Front End et des applications web, [open source](https://fr.wikipedia.org/wiki/Open_source) basé sur Type Script.

* 1. Langages serveur (back end)

En plus des langages client il y a une quantité importante de langages serveurs. Ces langages sont gérés par les serveurs. Ils se diffèrent par les langages client par leur objectif. En effet, ils servent à décrire comment un site web doit se comporter. Parmi ces langages, on peut citer PHP, java, python, Ruby. Par ailleurs, de nos jours les langages serveurs ne sont plus suffisants c'est pour cela qu'on utilise des boites à outils pour nous aider à créer des sites web plus facilement. On parle donc de Framework : on dispose d'un ou plusieurs Framework en fonction du langage. Par exemple pour "PHP" on a "symphonie" ou "zend" Framework, pour "python" on a "Django", pour "java" on a le "javaEE". Aujourd'hui on utilise souvent les Framework car ils nous font gagner un temps très précieux. Dans notre cas on a utilisé le Spring Boot

* 1. HTTP : HyperText Transfert Protocol

Protocole qui permet au client de récupérer des documents du serveur. Ces documents peuvent être statiques (contenu qui ne change pas : HTML, PDF, Image, etc.) ou dynamiques (Contenu généré dynamiquement au moment de la requête : PHP, JSP, ASP, etc.). Ce protocole permet également de soumissionner les formulaires. La plupart des applications Front end communiquent avec les services back end via le protocole HTTP.



**Figure 1 : Communication entre le front end et le back end**

1. Les architectures web
   1. L’architecture client/serveur

L’architecture client/serveur désigne un mode de communications entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur ; chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique. Pour pouvoir communiquer le client et les serveurs doivent utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément

* 1. Architecture à 3 niveaux

Le développement des sites interactifs, qui permettent de conserver les données des visiteurs, repose aujourd'hui sur une architecture à 3 niveaux entre serveur de données, serveur d'application et client web. Ce type d'architecture est plus complexe et permet d'accéder aux bases de données stockées sur un serveur.

Les trois niveaux s'articulent de la manière suivante :

* Le premier niveau s'occupe de l'interface avec l'utilisateur depuis le navigateur.
* le second héberge le serveur web qui est complété par le serveur d'application qui exécutent les traitements demandés lors de l'appel HTTP d'une page,
* Le troisième niveau assure la gestion des données au sein d'un SGBD (Système de Gestion de bases de données) et répond aux requêtes du serveur HTTP.
  + 1. Model MVC

Le modèle MVC (modèle-vue-contrôleur), Ce modèle de conception (design pattern) a été imaginé à la fin des années 1970 pour le langage Smalltalk afin de bien séparer le code de l’interface graphique de la logique applicative. Il est utilisé dans de très nombreux langages : bibliothèques Swing et Model 2 (JSP) de Java, Framework PHP, ASP.NET MVC, etc.



**Figure 2 : Modèle MVC**

* + 1. Model MVVM

Le modèle-vue-vue modèle (en abrégé MVVM, de l'anglais Model View ViewModel) est une architecture et une méthode de conception utilisée dans le génie logiciel. Apparu en 2004, MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Presentation Foundation et Silverlight via l'outil MVVM Light par exemple.



**Figure 3 : Modèle MVVM**

1. Les différents web

On a passé dans les années 90 depuis le web 1.0 qui a produit unnombrelimité de sites Webvers le web 2.0 au milieu des années 2000 qui a permis à un grand nombre de personnes de devenir aussi des acteurs du Web. De nos jours, même si le terme fait encore débat, on parle du Web 3.0 qui mélange différents Web ayant chacun des caractéristiques particulières, Parmi ces web on distingue le web social qui englobe les réseaux sociaux, le web programmable offrant des API (Application Protocol interface) pour programmer, le web physique vous mettant en relation avec des objets physiques, le web temps réel et le web sémantique permettant d’enrichir les informations publiées sur le Web. L'internet des objets se trouve à l'interaction de tous ces web.

1. Description générale de Travail réalisé

Notre travail consiste à la réalisation d’un site web éducatif, Notre premier travail durant ce stage consiste à choisir et télécharger deux Templates Bootstrap (Template pour la partie admin et une autre pour la partie client) et faire leurs intégrations avec L’Angular. Puis on a fait l’analyse et la conception de notre site à travers la spécification de besoin et la préparation des diagrammes de cas d’utilisation et le diagramme de classes (les classes présentent nos tableaux dans la base de données). Après cette étape on a créé notre base de données avec le système de gestion de base de données (phpMyAdmin) puis on construit nos classes qu’on a déjà préparées à l’aide de Framework Spring boot qu’on a choisi pour travailler notre partie back-end. Dès qu’on a terminé la création de classes avec ses relations, l’injection des dépendances et la préparation des différents services on a passé à la dernière étape qu’est la consommation de ces services avec l’Angular.

Conclusion

Dans ce chapitre nous avons donné un aperçu de ce qu'est le web. Nous avons alors détaillé les concepts du web et des technologies associées, ainsi que la programmation du web et du projet de développement des sites web. Notre travail dans le cadre de ce rapport concerne la conception et l'implémentation d'un site web éducatif. Le chapitre suivant est dédié à l'analyse et à la conception de ce site.

Chapitre2 Analyse et Conception

Introduction

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité. C’est dans ce stade que nous devons clarifier en premier lieu la vue globale, en décrivant l’architecture générale que nous allons suivre dans la partie réalisation de notre projet. Puis, dans un deuxième lieu nous allons détailler notre choix conceptuel à travers deux types de diagrammes.

1. Méthodologie de conception

Pour faciliter notre tâche nous avons recours langage de Modélisation unifié (UML) c’est une notation qui permet de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et il est devenu une référence en terme de modélisation objet, à un tel point que sa connaissance devienne indispensable pour un développeur.

1. Concept et architecture

L’architecture qu’on a utilisé dans la partie back end c’est l’architecture micro services. En informatique, les micros services sont un style d'architecture logicielle à partir duquel un ensemble complexe d'applications est décomposé en plusieurs processus indépendants et faiblement couplés, souvent spécialisés dans une seule tâche. Les processus indépendants communiquent les uns avec les autres en utilisant des API indépendantes du langage de programmation.

Des API REST sont souvent employées pour relier chaque micro service aux autres. Un avantage avancé est que lors d'un besoin critique de mise à jour d'une ressource, seul le micro service contenant cette ressource sera mis à jour, l'ensemble de l'application restant compatible avec la modification, contrairement à la totalité de l'application dans une architecture classique, par exemple une architecture trois tiers. Cependant, le coût de mise en place, en raison des compétences requises, est parfois plus élevé.

1. Spécification des besoins

Les sites Web éducatifs peuvent comprendre des sites Web qui proposent des vidéos ou des ressources thématiques qui servent d’outils pour améliorer l’apprentissage et compléter l’enseignement en classe. Ces sites Web contribuent à rendre le processus d’apprentissage divertissant et attrayant pour l’étudiant, surtout à notre époque. C’est le premier but de notre travail, le deuxième but est de vendre des projets de la nouvelle technologie très intéressants et de faire des projets selon le besoin de client tout en donnant les explications nécessaires pour bien comprendre le processus de travail.

* 1. Diagramme de cas d’utilisation
     1. Définition

Les rôles des diagrammes de cas d’utilisation sont de recueillir, d’analyser et d’organiser les besoins, ainsi que de recenser les grandes fonctionnalités d’un système. Il s’agit donc de la première étape UML pour la conception d’un système.

Un diagramme de cas d’utilisation capture le comportement d’un système, d’un sous-système, d’une classe ou d’un composant tel qu’un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d’utilisation, ayant un sens pour les acteurs. Ainsi ces cas d’utilisation permettent d’exprimer le besoin des utilisateurs d’un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d’une vision informatique.

Il ne faut jamais négliger cette première étape pour produire un site web conforme aux attentes des utilisateurs ciblés. Pour élaborer les cas d’utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

* + 1. Diagramme de cas d’utilisation de site

La figure 4 représente le diagramme de cas d’utilisation de site. Nous avons principalement deux acteurs l’administrateur de site et l’utilisateur ou bien le client de site chaque acteur a son propre login et mot de passe pour bien assurer la sécurité de site.

* + 1. Les acteurs de notre projet

**Le visiteur :** c’est un individu qui est entrain de fouiller sur le net, cherchant une idée de projet ou bien des informations utiles. Jusqu’au ce stade c’est un utilisateur inconnu donc il n’est pas encore un client.

**Le Client :** cette acteur est un visiteur ayant déjà créer un compte sur notre site, il peut donc passer une commande sur un projet ou bien demander une services bien déterminée en toute sécurité sachant que notre système doit être l’unique responsable de la confidentialité des données personnelles de ses clients.

**L’administrateur :** pour les sites web on l’appelle généralement « le webmaster ». Le terme webmaster de site web désigne communément celui qui est chargé d'un site web. Il gère toute la mise en place technique et parfois la mission éditoriale, il doit gérer au jour le jour la technique et mettre à jour le contenu du site web. Il est le seul qui a l’accès à la base de données.



**Figure 4 : diagramme de cas d’utilisation de site**

* 1. Diagramme de classes
     1. Définition

Le diagramme de classes est un schéma utilisé en [génie logiciel](https://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel) pour présenter les [classes](https://fr.wikipedia.org/wiki/Classe_(informatique)) et les [interfaces](https://fr.wikipedia.org/wiki/Interface_(informatique)) des systèmes ainsi que les différentes relations entre celles-ci. Ce [diagramme](https://fr.wikipedia.org/wiki/Diagramme) fait partie de la partie [statique](https://fr.wikipedia.org/wiki/Statique) d'[UML](https://fr.wikipedia.org/wiki/Unified_modeling_language) car il fait abstraction des aspects temporels et [dynamiques](https://fr.wikipedia.org/wiki/Dynamique). Une [classe](https://fr.wikipedia.org/wiki/Classe_(informatique)) décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les [instances](https://fr.wikipedia.org/wiki/Instance_(programmation)) de la classe. Les classes sont utilisées dans la [programmation orientée objet](https://fr.wikipedia.org/wiki/Programmation_objet). Elles permettent de modéliser un [programme](https://fr.wikipedia.org/wiki/Programme_informatique) et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

* + 1. La composition d’un diagramme de classes

En général un diagramme de classe peut contenir les éléments suivants :

**Les classes:** une classe représente la description formelle d’un ensemble d’objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.

**Les associations :** une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances des classes associées.

**Les attributs :** les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l’attribut doit être unique dans la classe.

* + 1. Diagramme de classes de site

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. La figure 5 présente le diagramme de classes de notre site.



**Figure 5: diagramme de classe de site**

Conclusion

Nous venons de terminer cette partie de conception, qui consiste à déterminer aussi bien les méthodes de travail que les chartes graphiques de notre site web avec ses parties statiques et dynamiques.

Dans le chapitre suivant nous allons aborder la dernière partie qui représente la partie réalisation de notre site web, en se basant sur les mécanismes et les solutions déterminés dans la phase de conception.

Chapitre3 : Réalisation et implémentation de projet

Introduction

Ce chapitre est consacré à la réalisation et l’implémentation de notre système Pour ce faire, nous adaptons Plusieurs logiciels (de développement de programmes d'application et d’autre qui permettant de mettre en place facilement un serveur Web et d’autre application qui permet de créer des fantastiques interfaces pour notre projet). Cette phase est la phase de la construction logicielle et la partie finale du test et d’implémentation. Nous élaborons une présentation des différentes interfaces crées.

1. Environnement de travail

Au niveau de cette dernière partie, nous allons énumérer les outils que nous avons utilisés pour réaliser notre site ainsi que ses principales interfaces.

* 1. Environnement Logiciel
     1. Langage de programmation

Nous avons eu recours, pour le développement de notre application au langage de programmation J2EE. C’est la garantie de portabilité qui fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

D’autre part JAVA est sécurisée, il a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce cadre, la sécurité n’a pas été négligeable. C’est le langage le plus adopté par les développeurs grâce à sa fiabilité et sa performance élevé.

* + 1. Environnement de développement

**IntelliJ IDEA** : également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) de technologie Java destiné au développement de logiciels informatiques. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

**WebStorm:** est un IDE pour les langages Web (HTML, CSS et JavaScript), développé par l'entreprise JetBrains et basé sur la plateforme IntelliJ IDEA.

**JDK :** Java Développent Kit Java est l'environnement dans lequel le code Java est compilé pour être transformé en byte-code afin que la machine virtuelle JAVA (JVM) puisse l'interpréter.

**JEE :** Java Enterprise Edition, ou Java EE, c’est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d’entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d’extensions au Framework Java standard (JSE, Java Standard Edition) afin de faciliter la création d’applications réparties.

**JPA :** c’est une API Java Persistance qui repose sur des entités annotés et sur un gestionnaire de ces entités (EntityManager) qui propose des fonctionnalités pour les manipuler (ajout, modification suppression, recherche). Ce gestionnaire est responsable de la gestion de l'état des entités et de leur persistance dans la base de données.

* + 1. Serveur d’application

**XAMPP** : est un ensembe de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

* + 1. Système de gestion de base de données

**PhpMyAdmin :** C’est une interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme les créations de table de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier SQL et ainsi transférer facilement ses données.

* + 1. Framework

**Spring boot :** Spring Boot est un Framework qui permet de créer des applications basées sur des micros services. Atout de Spring Boot :

* Faciliter le développement d’applications complexes.
* Facilite à l’extrême l’injection des dépendances
* Réduire à l’extrême les fichiers de configurations
* Faciliter la gestion des dépendances Maven
* Auto Configuration : la plupart des beans sont créés si le ou les jar(s) adéquats sont dans le classpath.
* Fournir un conteneur de servelet embarqué (Tomacat, Jetty)
* Créer une application autonome (jar ou war)

**Angular :** Angular Permet de créer la partie front end des applications web de type SPA (Single Page Application réactive). Une SPA est une application qui contient une seule page HTML (index.html) récupérée du serveur. Pour naviguer entre les différentes parties de cette application, Java Script est utilisé pour envoyer des requêtes http (AJAX) vers le serveur pour récupérer du contenu dynamique généralement au format JSON.

Ce contenu JSON est ensuite affiché coté client au format HTML dans la même page.

Pour développer une application Angular il est recommandé d’utiliser Type Script qui sera compilé et traduit en Java Script.

**Node.js**: est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications Common JS.Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js. Concrètement, Node.js est un environnement bas niveau permettant l’exécution de JavaScript côté serveur.

* + 1. Logiciel de test

**Postman**: un outil pour tester les API RESTful. Une application moderne est construite sur des APIs. C'est sur cette phrase d'accroche que nous découvrons le site de Postman, un outil qui permet de construire et de tester rapidement des requêtes HTTP.

1. Démonstrations des interfaces

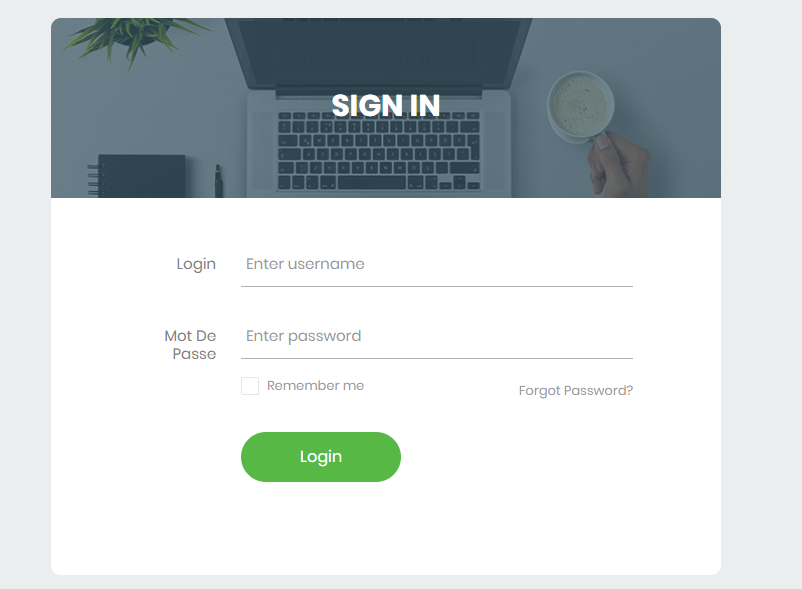
Cette partie dénombre la présentation des Scénarios applicatifs de l’application. Nous allons présenter dans ce qui suit, les imprimes-écran des principales interfaces réalisées dans notre site web.

Nous avons choisi de faire deux interfaces différents l’une pour l’admin et l’autre pour les utilisateurs/les clients.

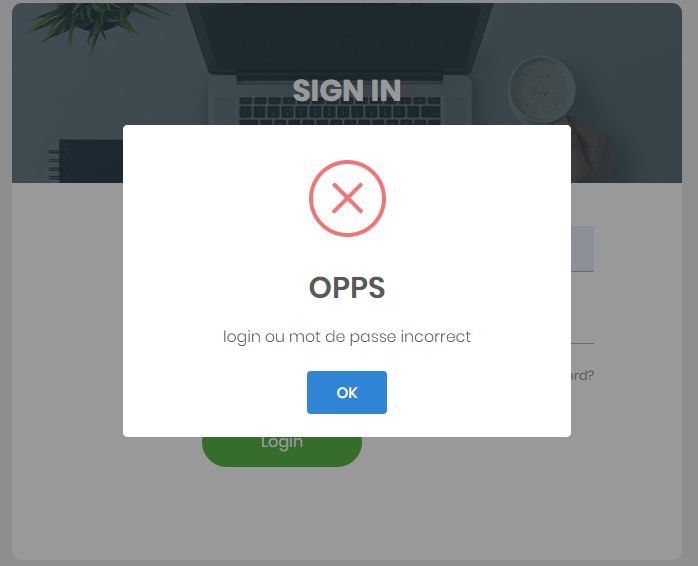
* 1. Partie Admin
     1. Fenêtre d’Authentification

Dans cette interface (figure 5) l’utilisateur doit taper son login et son mot de passe convenables qui sont stockés dans la base de données afin d'accéder à sa propre page. Pour bien assurer la sécurité de notre site nous avons fait un contrôleur de champ.

Si les mots clés sont incorrects, un message d’erreur va être affiché après le clic sur Le bouton **‘Login’** qui permet à l’utilisateur de confirmer son login et d’entrer au page d’accueil (figure 6)

****

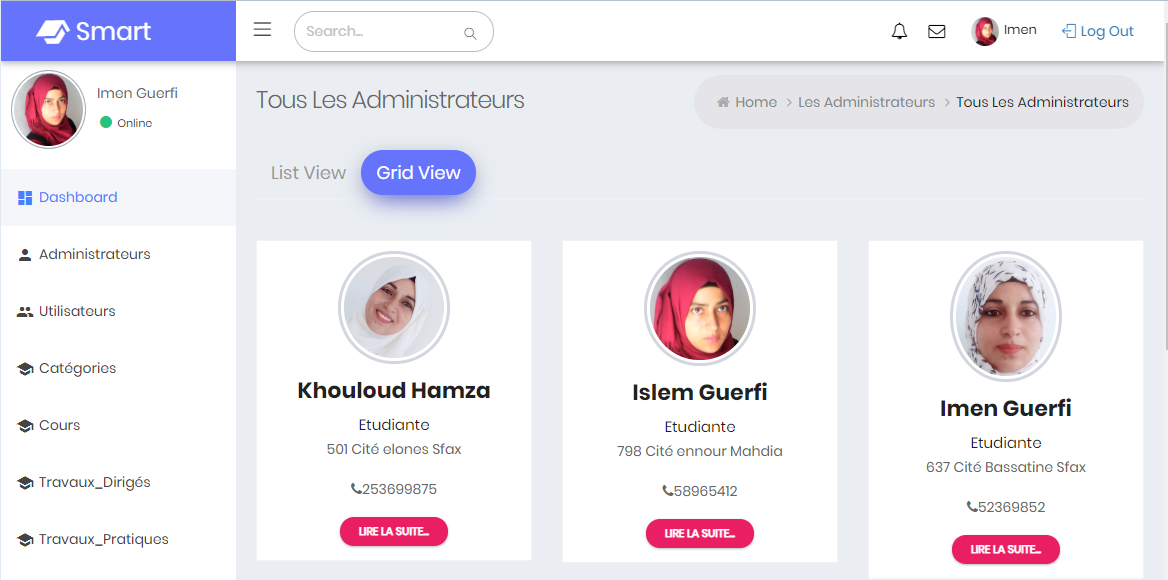
***Figure 6 : Interface Authentification***

****

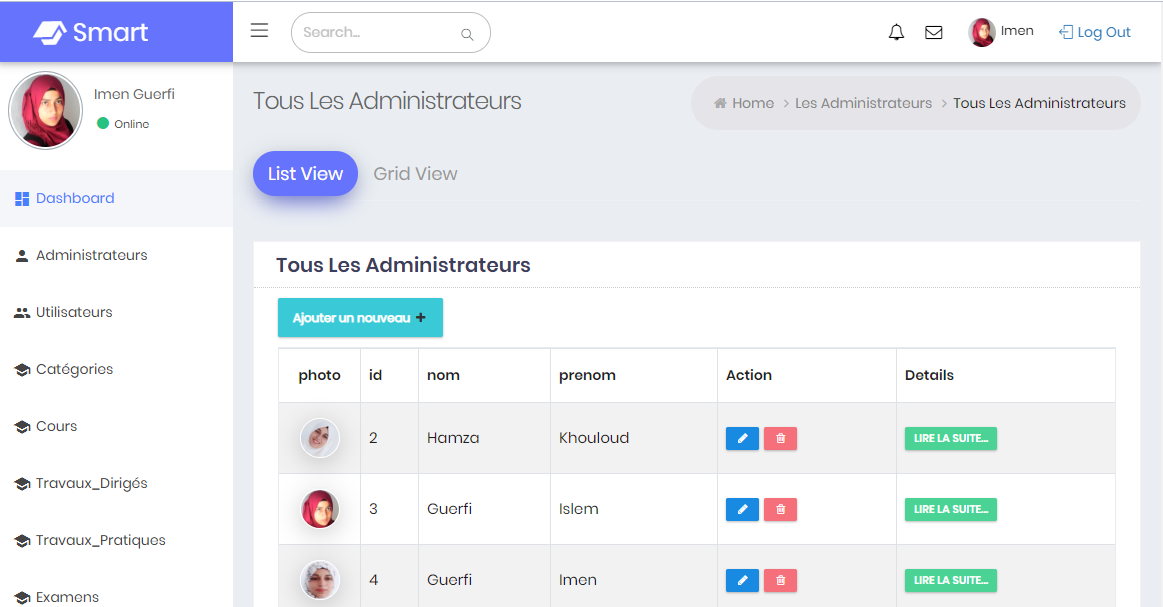
**Figure 7 : Message d’erreur**

* + 1. Liste d’administrateurs

Nous avons plusieurs administrateurs qui peuvent accéder à notre site, les figures 7 et 8 présentent la page des administrateurs, chaque administrateur peut connaitre les autres administrateurs de site, et a le droit d’ajouter, modifier ou bien supprimer.

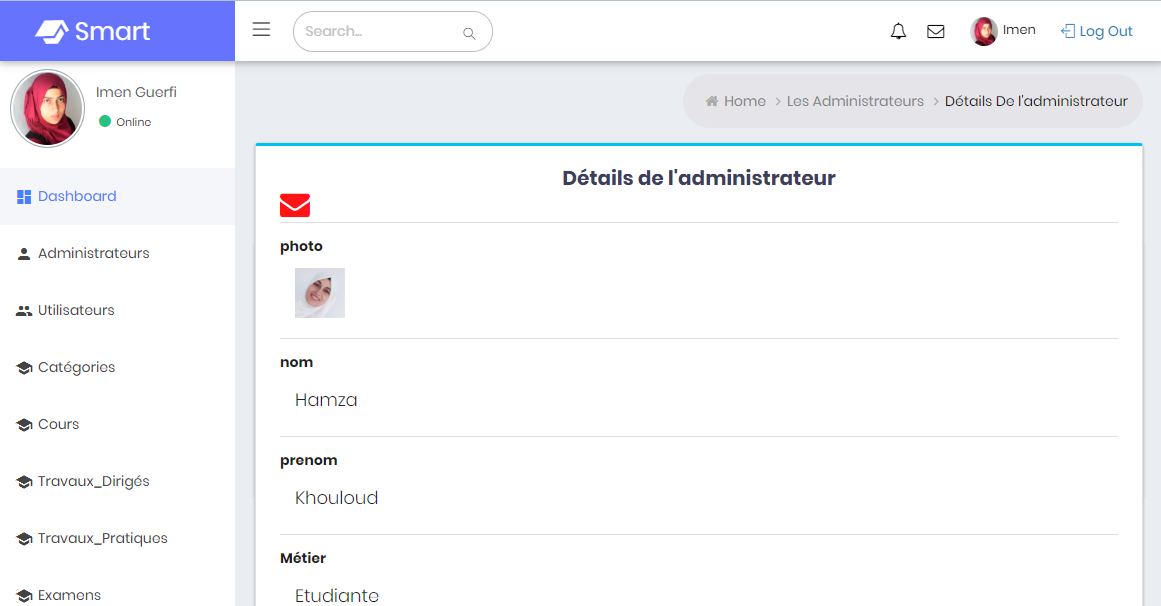
****

**Figure 8 : Liste d’administrateurs**

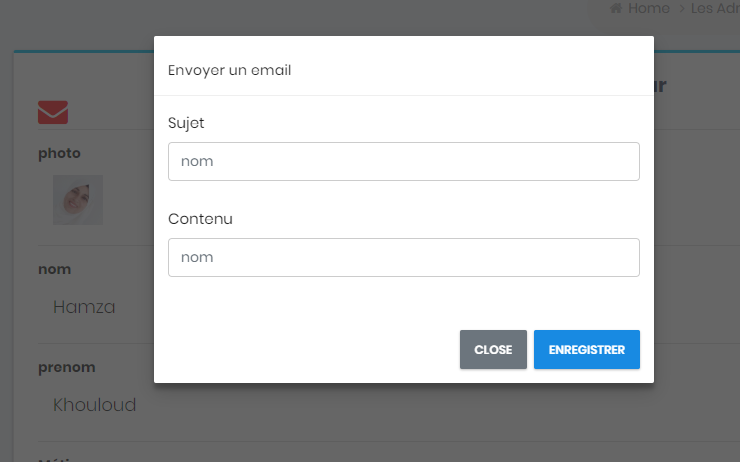
****

**Figure 9: Liste d’administrateurs**

La figure 9 présente toutes les informations personnelles de l’administrateurs, nous pouvons accéder aux informations de chaque admin en cliquant sur le bouton **‘lire la suite’** dans la figure 8, chaque admin peut envoyer un email à un autre admin ou bien à un client en cliquant sur l’envelopppe rouge une fenetre s’ouvre et nous pouvons envoyer l’ email (figure 10)

****

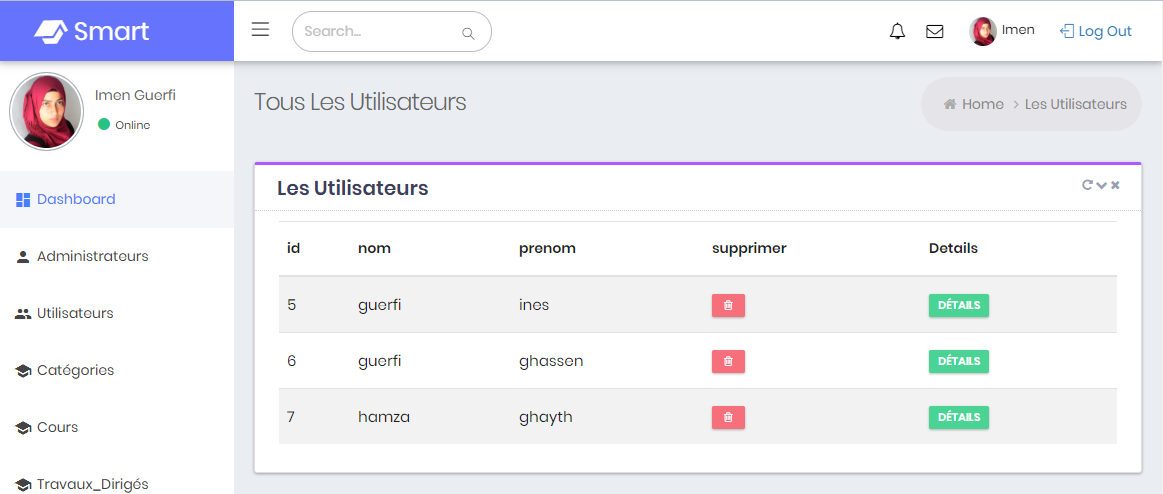
**Figure 10 : fenêtre des détails d’un administrateur**

****

**Figure 11 : fenêtre d’email**

* + 1. Liste des utilisateurs

Les administrateurs de site peuvent accéder à la liste des utilisateurs qui sont déjà inscrits et enregistrés dans la base de données, ils peuvent voir ses informations personnelles ou bien supprimer un utilisateur indésirable.

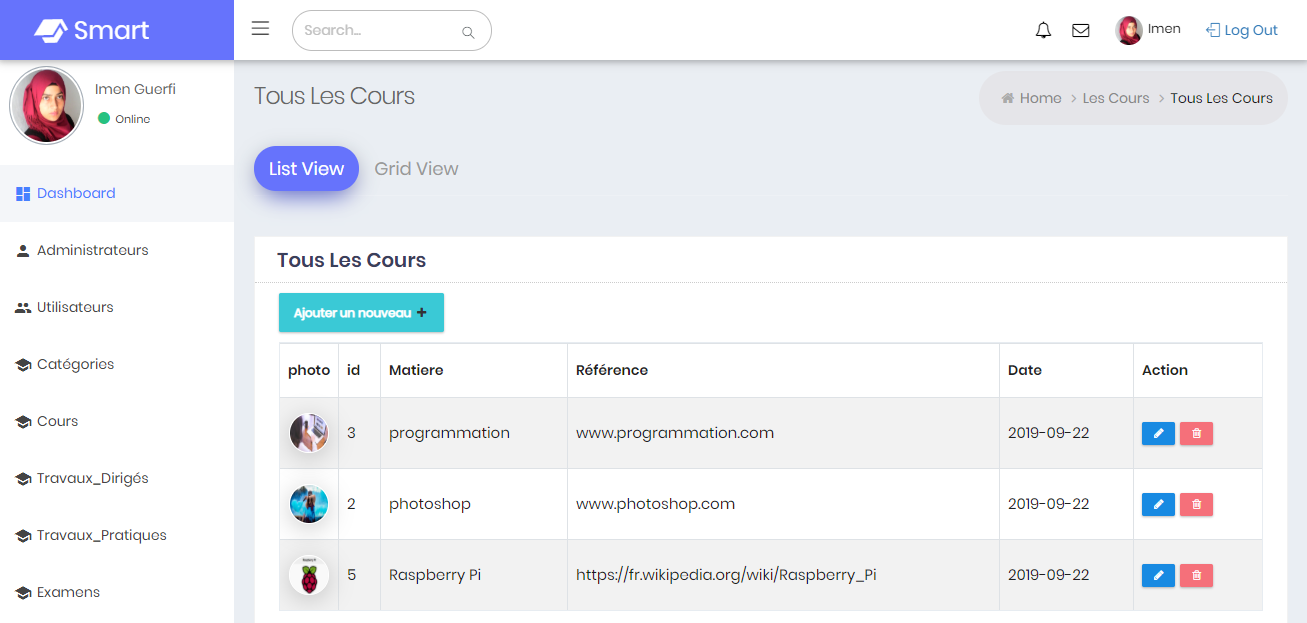


**Figure 12: Liste d’utilisateurs**

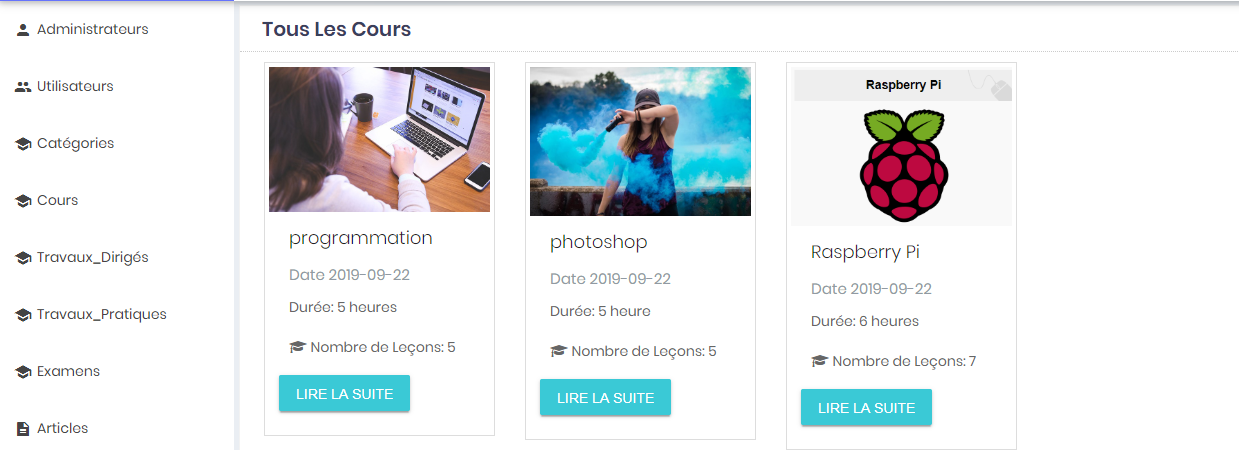
* + 1. Liste des cours

Cette figure représente la liste des Cours ajoutés par le webmaster, à partir de cette fenêtre l’admin peut faire l’action qu’il veut par exemple s’il veut ajouter un cours il suffit d’appuyer sur le bouton **‘Ajouter un nouveau’** (figure 12) et la fenêtre d’ajout s’ouvrira (figure 14), s’il veut voir les détails d’un cours il suffit de cliquer sur le bouton **‘lire la suite’** (figure 13) et un autre fenêtre contient tous les détails de cours va être afficher (figure 15).

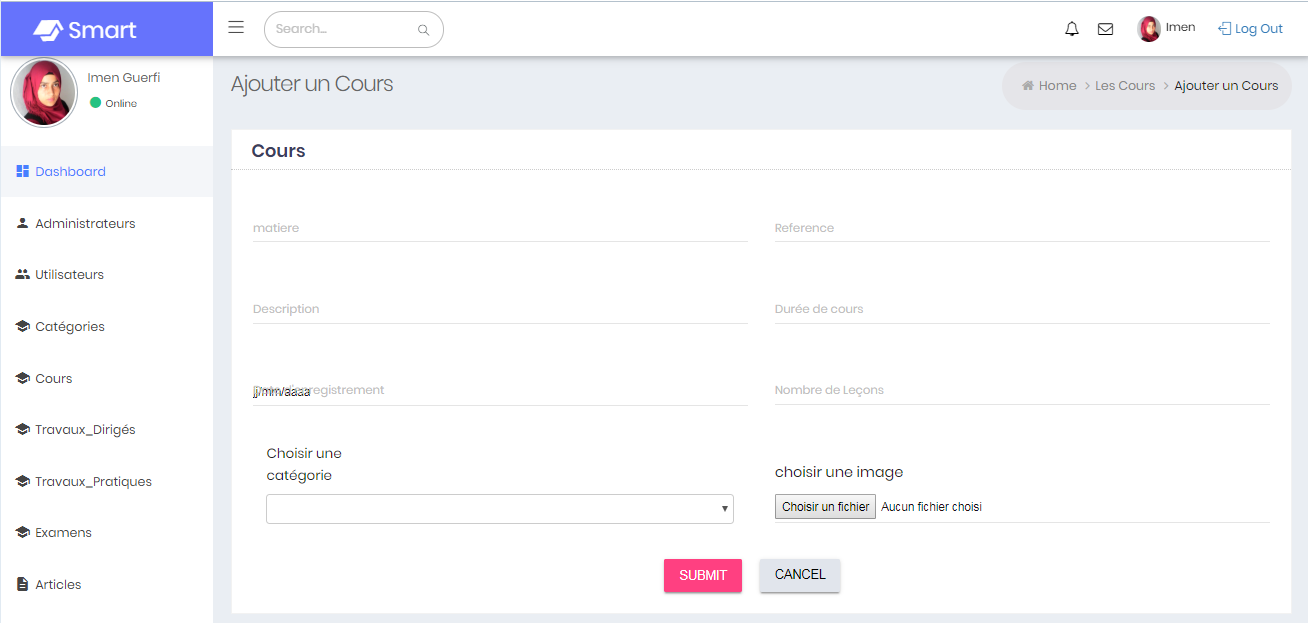
Nous voulons par cette figure donner un exemple de plusieurs listes qui s’affichent de la même manière que la présente comme par exemple (la liste des projets, la liste des articles, la listes des travaux dirigés, etc.).

****

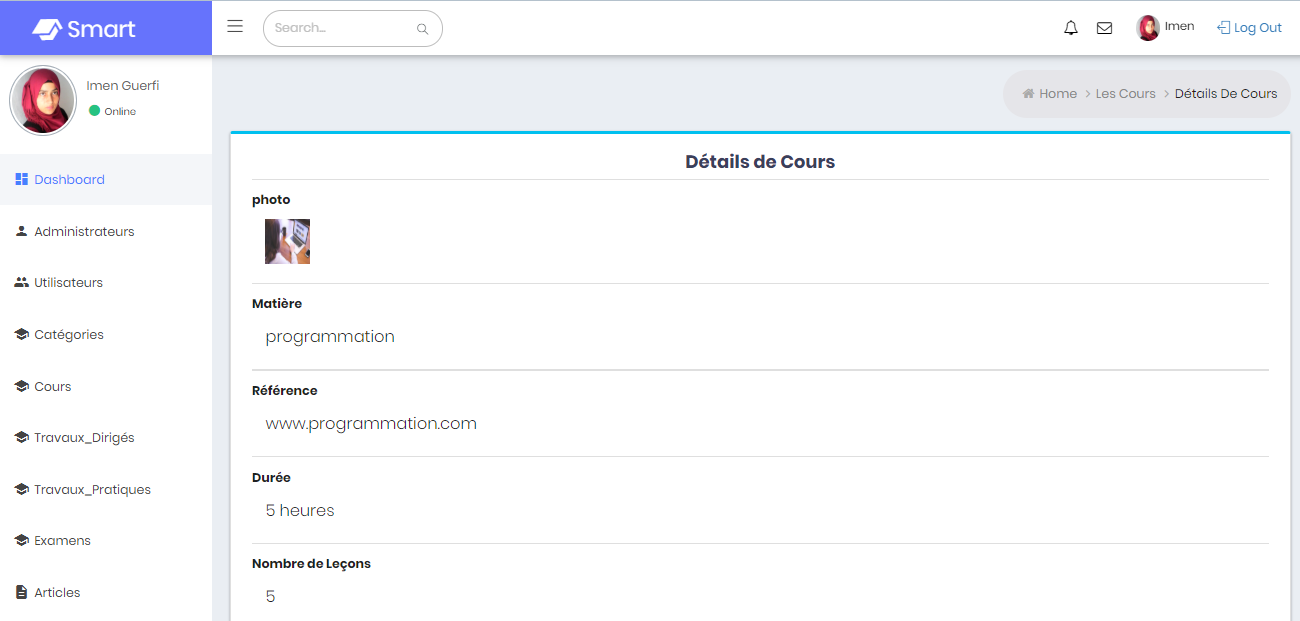
**Figure 13: Liste de cours**



**Figure 14: Liste de cours**

****

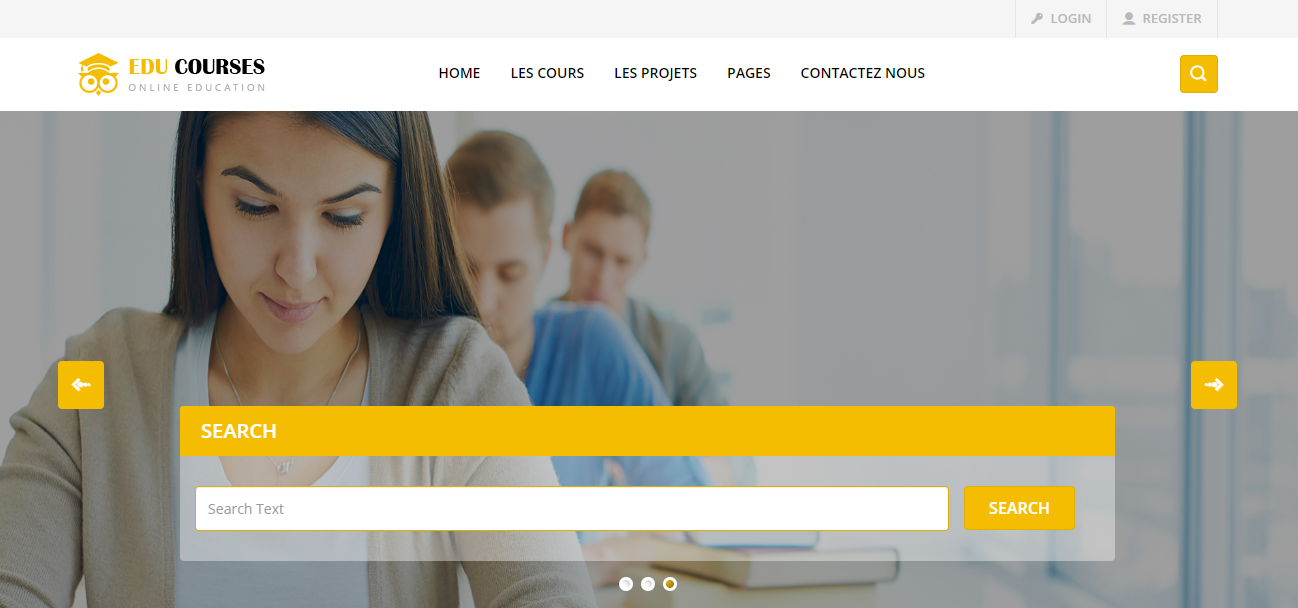
**Figure 15 : fenêtre d’ajout**

****

**Figure 16: fenêtre de détails d’un cours**

* 1. Partie Client
     1. Fenêtre d’accueil

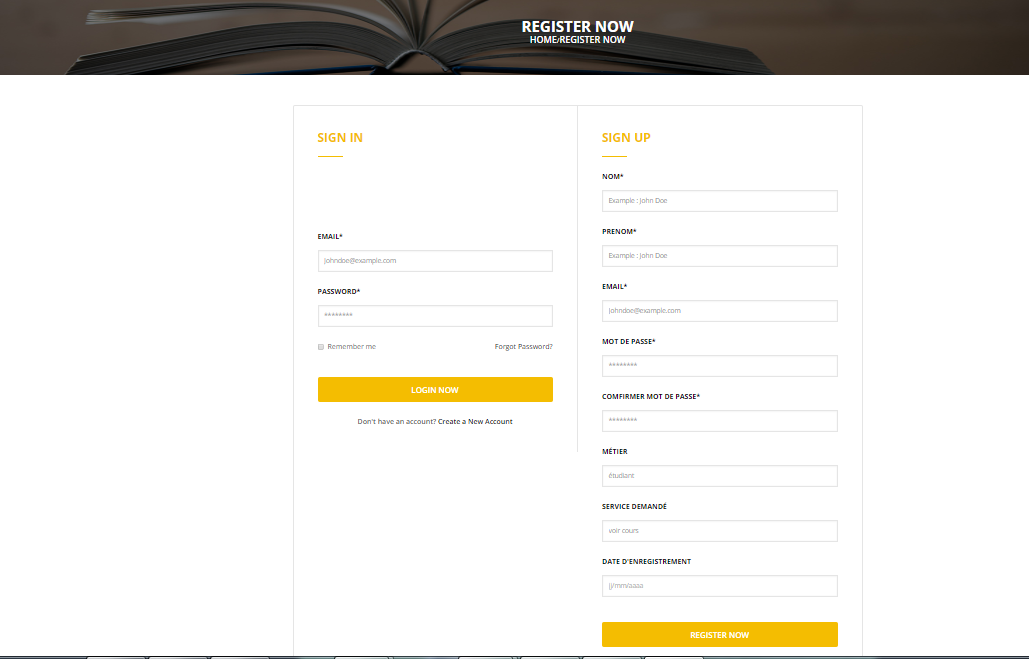
C'est la page d'accueil qui s'affiche dès l’accès à notre site web, à partir de cette page on peut accèder à n’importe quelle page de site



**Figure 17 : page d’accueil**

* + 1. Fenêtre d’Authentification

Comme dans tout site web commercial le visiteur ne peut devenir client qu’après la phase d’inscription, notre site web met à la disposition de ses visiteurs un formulaire d’inscription accessible à partir du menu ‘**register’** dans la barre des menus en haut de la page d’accueil.



**Figure 18 : Page Authentification & d’inscription**

Après la phase d’inscription présentée dans la figure 17 le client doit s’authentifier pour bien profiter des privilèges qu’un visiteur normal ne possède pas comme par exemple le passage des commandes.

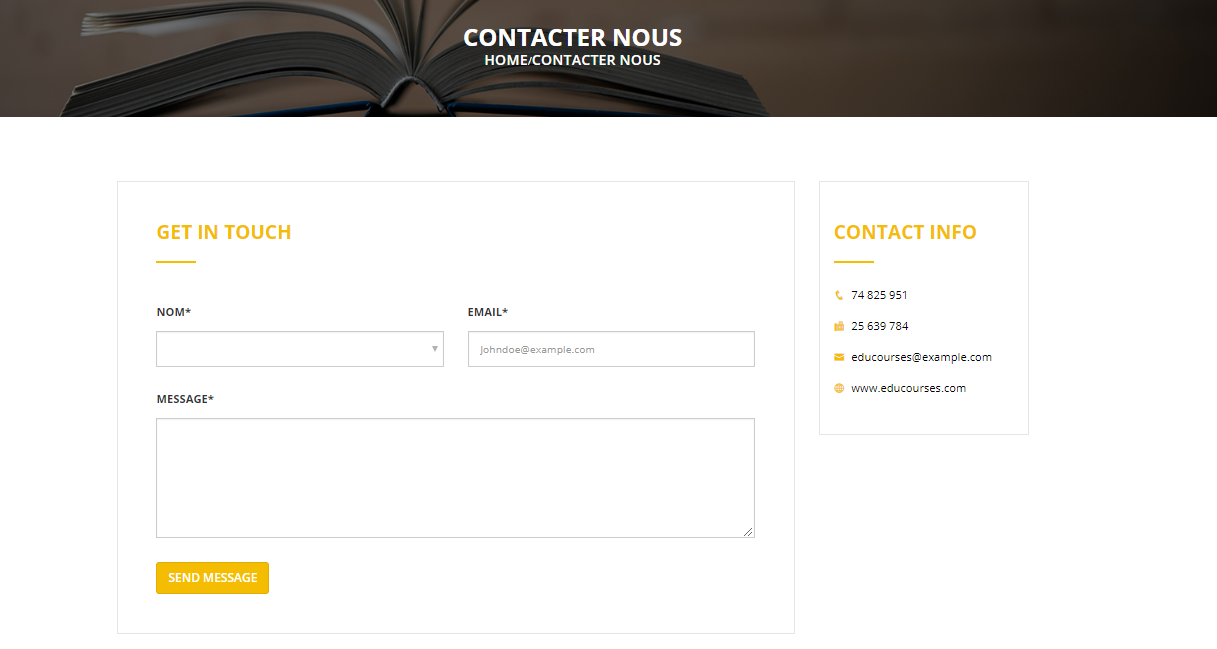
* + 1. Liste des cours

Cette figure représente la listes des cours ajoutés par le webmaster, nous voulons par cette figures donner un exemple de plusieurs listes qui s’affichent de la même manière que la présente.

**Figure 19 : Liste de cours**

**Figure 20 : Liste de cours**

* + 1. Page de contact

La figure 20 nous affiche les contacts de notre webmaster que tout client peut le contacter par téléphone ou par E-mail en cas de besoin.

**Figure 21 : Liste de cours**

Conclusion

Dans le chapitre réalisation nous avons appelé à présenter les interfaces réalisé dans notre site web pour clarifier les étapes d’utilisation de notre site avec ses deux parties admin et client.

Conclusion

Ce stage m'a apporté une nouvelle expérience professionnelle enrichissante. Grâce à ce mois passé au sein de la société It Gate, j'ai acquis de nouvelles connaissances autant sur le milieu de l'entreprise que sur les langages informatiques. Et cela permet d’augmenter mes capacités de compréhension.

Le stage dans un milieu professionnel est constructif. En effet, j'ai pu développer mes compétences professionnelles grâce à l'environnement dans lequel j'ai effectué mon stage. Travailler avec des Frameworks comme l’Angular et le Spring boot a été bénéfique puisque j'ai pu développer mes connaissances des langages HTML et CSS ainsi que J2EE.

Ce stage m'a aidé à développer mes connaissances personnelles et m'a fait découvrir de nouveaux outils. Ces quatre semaines m'ont permis de réaliser un site web et d’apprendre comment travailler avec les Frameworks. Ils s’agissent des outils que je ne connaissais que de nom et que j'ai appris à utiliser lors de ce stage.

J'ai aussi pu découvrir la vie au sein d'une entreprise. Le fait de se référer à un tuteur constitue une aide dont je n'aurais pu me passer et s'adresser à un supérieur hiérarchique en construisant une explication et une argumentation a été instructif.

Références

http://www.jcp.org/en/jsr/detail?id=220

http://www.eclipse.org/dali/

https://glassfish.dev.java.net/

http://hibernate.org/299.html

http://solarmetric.com/Software/Kodo/kodoejb.php

http://www.oracle.com/technology/products/ias/toplink/JPA/index.html

http://dev2dev.bea.com/persistence/

http://www.jroller.com/page/mkeith

http://blog.hibernate.org/

http://blogs.sun.com/theaquarium

http://apress.com/book/bookDisplay.html?bID=10093