**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

**ET DE LA RECHERCHE SCIENTIFIQUE**

**ECOLE SUPERIEURE PRIVEE D’INGENIERIE**

**ET TECHNOLOGIE APPLIQUEE - SOUSSE**

****

**RAPPORT**

**De**

**PROJET FIN D’ANNEE**

Présenté Par

**CHADOU MOHAMED ALI**

Spécialité : GENIE INFORMATIQUE

Conception et développement d’une site web/mobile éducatif avec générateur de code

Présenté devant le jury d’examen :

|  |  |
| --- | --- |
| Mr. Mehrez Issa  Mr. Badis Souissi | **Jury**  **Encadrant** |

**Année Universitaire 2020 – 2021**

Table des matières

**Introduction générale3**

**Présentation du projet5**

Introduction 5

Présentation du projet 5

Problématique5

Etude de l’existant5

Objectifs du projet 6

Solution proposée 6

Conclusion 6

**Architecture et conception7**

Introduction 7

Besoins fonctionnels et non fonctionnels 7

Traduction des besoins en diagramme de cas d’utilisation 8

Description des acteurs8

Diagramme de cas d’utilisation globale8

Conception générale9

Diagramme de classe11

Conclusion 12

**Réalisation13**

Introduction 13

Réalisation13

Environnement logiciel13

Environnement technique14

Travail réalisée15

Introduction 15

Interface web15

Conclusion20

**Conclusion et perspectives21**

Listes des figures

[Figure 1 4](#_Toc72483309)

[Figure 2 8](#_Toc72483310)

[Figure 3 11](#_Toc72483311)

[Figure 4 15](#_Toc72483312)

[Figure 5 16](#_Toc72483313)

[Figure 6 16](#_Toc72483314)

[Figure 7 17](#_Toc72483315)

[Figure 8 17](#_Toc72483316)

[Figure 9 18](#_Toc72483317)

[Figure 10 18](#_Toc72483318)

[Figure 11 19](#_Toc72483319)

[Figure 12 19](#_Toc72483320)

[Figure 13 20](#_Toc72483321)

***Introduction générale***

La programmation informatique est l'écriture de programmes servant à créer un logiciel. Un programme contient l'ensemble des informations nécessaires pour que le logiciel remplisse sa tâche.

La programmation exige une rigueur de la méthode de travail et de l'esprit qui peut par la suite être mise à profit dans tous les domaines.

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine.

Les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par l'appareil informatique, et d'autre part d'indiquer comment sont effectuées les manipulations, selon quels algorithmes. Ils servent de moyens de communication par lesquels le programmeur communique avec l'ordinateur, mais aussi avec d'autres programmeurs ; les programmes étant d'ordinaire écrits, lus, compris et modifiés par une équipe de programmeurs.

La programmation dans le cycle de développement :

Le développement est un processus qui est souvent considéré comme un cycle. Il existe plusieurs méthodologies de développement qui déterminent de manières différentes le nombre et l’ordre (e.g. séquentiel ou itérative) des étapes, mais en général on peut regrouper les étapes en trois grandes parties :

Les besoins, ou plus généralement tout ce qui précède, motive et/ou justifie le développement ;

La construction, c'est à dire toutes les activités nécessaires à l'implémentation concrète de l'application ;

Le test ou, plus en général, tout ce qui se passe une fois que l'application/logiciel est disponible.

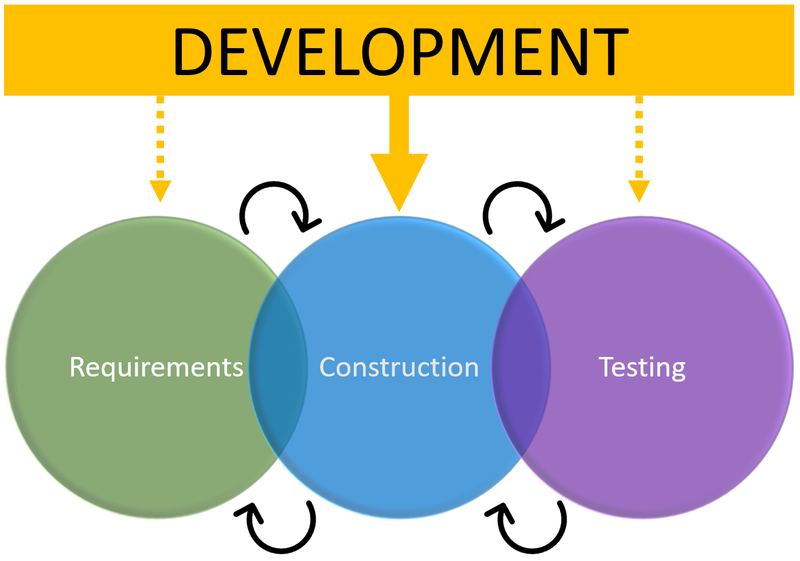


Figure 1

Le concept de l'éducation traditionnelle a radicalement changé au cours des deux dernières années. Être physiquement présent dans une salle de classe n'est plus la seule option d'apprentissage - du moins pas avec l'essor d'Internet et des nouvelles technologies. Aujourd'hui, vous avez accès à une éducation de qualité quand et où vous le souhaitez, à condition d'avoir accès à un ordinateur. Nous entrons maintenant dans une nouvelle ère - la révolution de l'éducation en ligne.

Le présent rapport synthétise le travail que nous avons effectué. Il est organisé en chapitres comme suit :

– Le premier chapitre donne une présentation générale du projet.

– La conception de notre solution sera présentée dans le chapitre deux.

– Le troisième chapitre décrit les tâches accomplies et les tests réalisés.

– Et nous allons finir par une conclusion et des perspectives.

***Chapitre I : Présentation du projet***

**I.1 Introduction**

Ce chapitre a pour objectif de situer le projet dans son contexte général, à savoir la problématique qui a inspiré la création de notre site/app, la description du projet et les objectifs à atteindre.

**I.2 Présentation générale du projet**

**I.2.1 Problématique**

Plusieurs sites d’apprentissage en ligne non dédient aux créations de votre propre projet.

La difficulté de transformer nos études en un exemple concret.

Perds de temps en répéter même code en plusieurs projet.

Le souci principal de chaque futur développeur se réduit essentiellement au choix d'un site d'apprentissage qui lui offre cours simple avec les tests pour lui pouce à continuer et lui facilite le développement afin de maintenir une ambiance favorable à la productivité et l’évolution des études.

D’autre part le professionnels développeurs souffre de répéter même code ou le copier puis le changer ce qu’implique un temps perdu qui augmente les délais de programmation et réduire la performance des employés.

**I.2.2 Etude de l’existant**

Les apprenants rencontrent souvent le problème de choix de site d'apprentissage ou de la langue à étudier, et surtout le premier pas de développement d'un projet.

Le choix se fait dans ce cas par l’une ou la combinaison des méthodes suivantes :

- chercher les plus langage demander à étudier.

- chercher dans plusieurs endroit (YouTube, Google) un cours.

- chercher des idées simples pour le développer.

Les plus fameux sites pour apprendre à coder sont :

-Open Classrooms

-Codecademy

-Khan Academy

-PluralSight

-Coursera

-CodeWars

-W3 School

Ceux sites ont à quelques exceptions des points faibles comme :

-Contenu léger avec des questions du quizz assez faible.

-Le faible nombre de cours ouverts ou gratuit.

-Apprentissage sans bien tester ou développer.

-Absence d'app mobile.

-Manque de code source.

**I.3 Objectifs du projet**

Les problèmes mentionnés précédemment peuvent être résolus par un moyen qui garantit :

-cours en ligne standard et gratuits

-Première expérience d’éducation simple en création de votre propre code.

-le bon intégration et l’accessibilité facile au monde programmation.

-cours de programmation accessibles à tous supports : ordinateur, tablette, smartphone.

-la liberté de créer facilement des applications, des sites web et d’autres projets.

De plus on propose un générateur de code qui facilite le développement et réduire le temps consommer pour le grand projet.

**I.4 Solution proposée**

Ce projet consiste à concevoir et développer une application web/mobile permettant la bonne et simple éducation. Elle doit aussi offrir la possibilité de développer via une interface graphique ainsi qu’elle lui fournir le code source.

Ce site vise essentiellement à diminuer le temps de création de vaste projet. Ceci va donc améliorer considérablement la productivité de développeurs afin de garantir une meilleure performance du travail.

Le projet présente une Platform accessible à tous supports (site web et application mobile). Cette méthode facilite l’accessibilité au monde programmation.

Notre site est destiné essentiellement aux apprenants de programmation quel que soit leur niveau d’éducation.

**Conclusion**

Tout au long de ce chapitre, nous avons mis notre projet dans son cadre général, à savoir la présentation du projet et ses objectifs, et nous allons spécifier dans le prochain chapitre la conception notre site.

***Chapitre II : Architecture et conception***

**II.1 Introduction**

La réussite de tout projet dépend de la qualité de son départ. De ce fait, l’étape de spécification des besoins constitue la base de départ de notre travail, elle doit décrire sans ambiguïté le logiciel à développer.

Pour assurer les objectifs attendus, il est essentiel que nous parvenions à une vue claire des différents besoins escomptés de notre projet.

Au cours de ce chapitre, nous allons dégager les fonctionnalités attendues du notre site en définissant les différents cas d’utilisation et quelques scénarios qui expliquent ces cas.

De plus, nous allons entamer une partie importante du développement de site qui constitue un pont entre la spécification et la réalisation. Nous allons présenter dans un premier temps la conception générale, puis la conception détaillée comprenant les vues statiques via les diagrammes de classes.

**II.2 Besoins fonctionnels et non-fonctionnels**

* Fonctionnels

-Un administrateur peut créer, modifier ou supprimer des tâches.

-Un administrateur peut visualiser les statistiques relatives (tâches, rendement de site).

-Un apprenant peut accéder à la liste de cours proposée par l'administrateur.

-Un apprenant peut télécharger tous documents proposés par les cours.

-Un apprenant peut créer et modifier son cursus.

-Un apprenant peut visualiser les statistiques relatives à ses études.

-Un apprenant peut communiquer avec l'administrateur à travers un service de messagerie.

* Non-fonctionnels

-La site/app doit être facile à utiliser.

-Le temps de réponse de la site/app doit être le plus rapide possible.

-La site doit être portable.

-Possibilité d’optimiser La site/app plus tard.

-La site/app doit sécuriser les informations de ses utilisateurs.

**II.3 Traduction des besoins en diagramme de cas d'utilisation**

Le diagramme des cas d’utilisation est un modèle simplifié du fonctionnement de la site/app qui spécifie les acteurs qui l’utilisent et les services qu’il leur offre.

**II.3.1 Description des acteurs**

Un acteur représente une abstraction d'un rôle joué par des entités externes au système qui interagissent directement avec lui.

Dans cette application, nous identifions deux types d’acteurs : l’administrateur et l'internaute.

Administrateur : C’est l’acteur qui gère les tâches ainsi que les apprenants et le contenue de site. Il se charge aussi de contrôler les fonctionnalités de site.

Internaute : Il s’agit d’un apprenant qui chercher de cours programmation à étudier ou un développeur qui cherche à générer son code plus vite.

**II.3.2. Diagramme de cas d'utilisation global**

Le diagramme de la figure 2 illustre le rôle de chaque acteur ainsi que l’étendue de ses responsabilités.

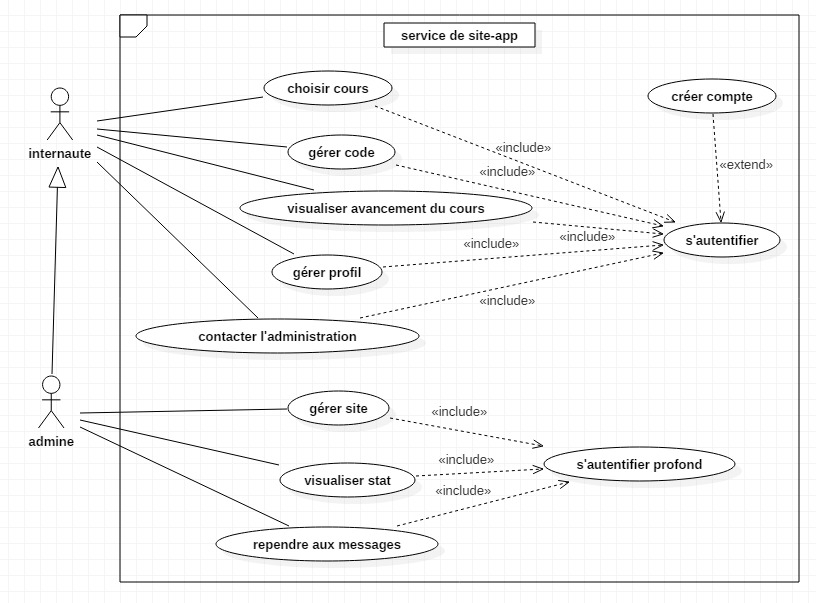


Figure 2

Ce diagramme est développé davantage par le tableau suivant :

|  |  |
| --- | --- |
| Titre | Accéder aux services de la site |
| Description | Ce cas d’utilisation permet aux utilisateurs ayant le rôle internaute ainsi que le rôle administrateur d’accéder à leurs comptes sur le site. |
| Acteurs | Internaute, administrateur |
| Préconditions | -L’utilisateur doit être authentifié.  -L’utilisateur doit avoir le rôle « Administrateur » pour gérer les tâches de tous les utilisateurs. |
| Scénario nominal | 1. L’utilisateur s’authentifie.  2. L’utilisateur choisit le cours dans le menu de la page d’accueil.  3. L’utilisateur est conduit à la page principale du cours où il peut visualiser ses différents modules ainsi que des statistiques sur leur progrès. |

**II.4 Conception générale**

Notre application consiste en un module web et un autre mobile qui sont en interaction à travers des services web.

- La site est plus simple à entretenir, parce qu’elle sépare la partie client de la partie serveur.

- Elle permet aussi de ne pas avoir à maintenir une connexion permanente entre le client et le serveur.

- Le serveur peut ainsi répondre à d'autres requêtes venant d'autres clients sans saturer l'ensemble de ses ports de communication.

- Une meilleure évolutivité et tolérance aux pannes (un serveur peut être ajouté facilement pour augmenter la capacité de traitement, ou pour en remplacer un autre).

- Dans un contexte Web :

- l'utilisation du protocole HTTP permet de tirer parti de son enveloppe et ses en-têtes.

- l'utilisation d'URI comme représentant d'une ressource permet d'avoir un système universel d'identification des éléments de site.

Pour l’application mobile, nous avons choisi de travailler avec le patron de Conception MVC.

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l’anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l’interface homme-machine (IHM) d’une application logicielle. Ce paradigme divise l’IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l’interface.

Nous l’avons choisi vu ses avantages nombreux :

-Il permet de formaliser l’application et de faciliter la compréhension de son mode de fonctionnement.

-Il permet de délimiter les différents modules de l’application et de simplifier leur création.

-La séparation des modules nous permet d’effectuer des changements sur l’un d’eux sans affecter le reste du code ou le fonctionnement global de l’application.

Les composant du modèle MVC sont :

- Model : La base de l’application. Il décrit les données et contrôle leur intégrité. Il comporte la base de données.

- Vue : L’interface avec laquelle interagit l’utilisateur. Elle est en interaction avec le modèle ainsi que le contrôleur.

- Contrôleur : Responsable de la synchronisation entre le modèle et la vue. Il gère les événements reçus par la vue et demande les changements nécessaires au modèle.

**II.5 Diagramme de classes**

La figure 3 représente le diagramme de classes des entités utilisées dans notre système :

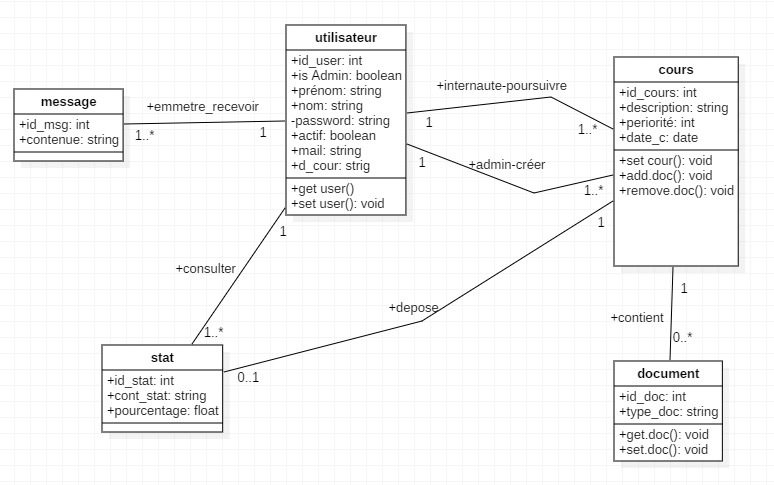


Figure 3

-Utilisateur : c’est la classe qui représente un utilisateur du site qui est soit un internaute soit un administrateur.

-cours : c’est l’entité qui représente la liste des cours proposée par l’administrateur et présentée à l’internaute.

-Document : certaines cours peuvent être accompagnées de documents qui servent soit à clarifier certains détails ou qui serve le code source réalisée par l'internaute.

-Message : c’est la classe qui définit les messages échangés entre les utilisateurs à travers un service de messagerie dans l’application.

-Stat : la class stat désigne le compte-rendu retourné aux utilisateurs qui décrit le déroulement de site et cours (pour l'apprenant).

**Conclusion**

Ce chapitre nous a permis de couvrir les différents besoins fonctionnels et non fonctionnels auxquels doit satisfaire notre application.

De plus, nous avons modélisé le fonctionnement de l’application afin d’avoir une vue globale et simplifiée du système. Nous avons aussi détaillé les différents modules de l’application ce qui nous a permis d’organiser le travail et d’avoir une idée claire sur le travail à réaliser.

Ce travail est décrit plus précisément dans le chapitre qui suit.

***Chapitre III : Réalisation***

**Introduction**

Après l’étape de conception de l’application, nous allons, dans ce chapitre, décrire la phase de réalisation. Nous allons présenter, en premier lieu, l’environnement du travail utilisé pour le développement de site, ensuite, nous allons donner un aperçu sur le travail accompli à travers des captures d’écran.

**III.1 Réalisation**

**III.1.2. Environnement logiciel**

Dans ce paragraphe nous décrivons les différents outils et logiciels utilisés pour la mise en œuvre de site.

* Notepad ++ : est un éditeur de texte et de code source à utiliser avec Microsoft Windows. Il prend en charge l'édition par onglets, ce qui permet de travailler avec plusieurs fichiers ouverts dans une seule fenêtre. Le nom du projet provient de l'opérateur d'incrémentation C. Notepad ++ est distribué sous forme de logiciel libre.
* WampServer : il fait référence à une pile de solutions pour le système d'exploitation Microsoft Windows, créée par Romain Bourdon et composée du serveur Web Apache, d'OpenSSL pour le support SSL, de la base de données MySQL et du langage de programmation PHP.
* PhpStorm : est un IDE commercial multiplateforme pour PHP, développé par la société tchèque JetBrains. PhpStorm fournit un éditeur pour PHP, HTML et JavaScript avec une analyse de code à la volée, une prévention des erreurs et des refactorisations automatisées pour le code PHP et JavaScript.
* SGBD choisie : MySQL :

MySQL est un système de gestion de base de données

(SGBD). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle et Microsoft SQL Server.

* Outil pour la modélisation avec UML : Enterprise Architect v7

Enterprise Architect est un outil de modélisation UML2 flexible, complet et puissant conçu pour les plateformes Windows. Ce logiciel vous offre un avantage concurrentiel pour le développement des systèmes, la gestion de projet et l’analyse commerciale. Un vaste éventail de langages de développement sont pris en charge, y compris Action Script, C, C + +, C # et VB. NET, Java, Visual Basic 6, Python, PHP, XSD, WSDL et plus encore.

* Android Studio : est l'environnement de développement intégré officiel du système d'exploitation Android de Google, basé sur le logiciel IntelliJ IDEA de JetBrains et conçu spécifiquement pour le développement Android.

**III.1.3. Environnement technique**

* UML : Unified Modeling Language :

Un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. C’est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG).

* HTML5/CSS3 :

Ce sont les deux langages utilisés pour créer et mettre en forme (respectivement) les pages de l’application web.

* -SQL : Structured Query Language :

Un langage complet de gestion de bases de données relationnelles. Il a été conçu par IBM dans les années 70. Il est devenu le langage standard des systèmes de gestion de bases de données (SGBD) relationnelles (SGBDR).

* -JavaScript :

Un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

* -PHP :

Un langage de scripts libre4 principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP3, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes5. En raison de la richesse de sa bibliothèque, on désigne parfois PHP comme une plateforme plus qu'un simple langage.

**III.2 Travail réalisé :**

**Introduction :**

Cette partie est consacrée à présenter quelques interfaces de l’application réalisée.

**III.2.1. Interfaces Web**

Les différentes interfaces de l’application web ont été créées de façon dans lequel nous avons choisi des éléments clairs et visuellement simples.

Les couleurs choisies pour le coté web ainsi que mobile sont principalement :

- un bleu sombre (code hex. #000080)

- un bleu cyan (code hex. #0099ff)

* Page d’accueil :

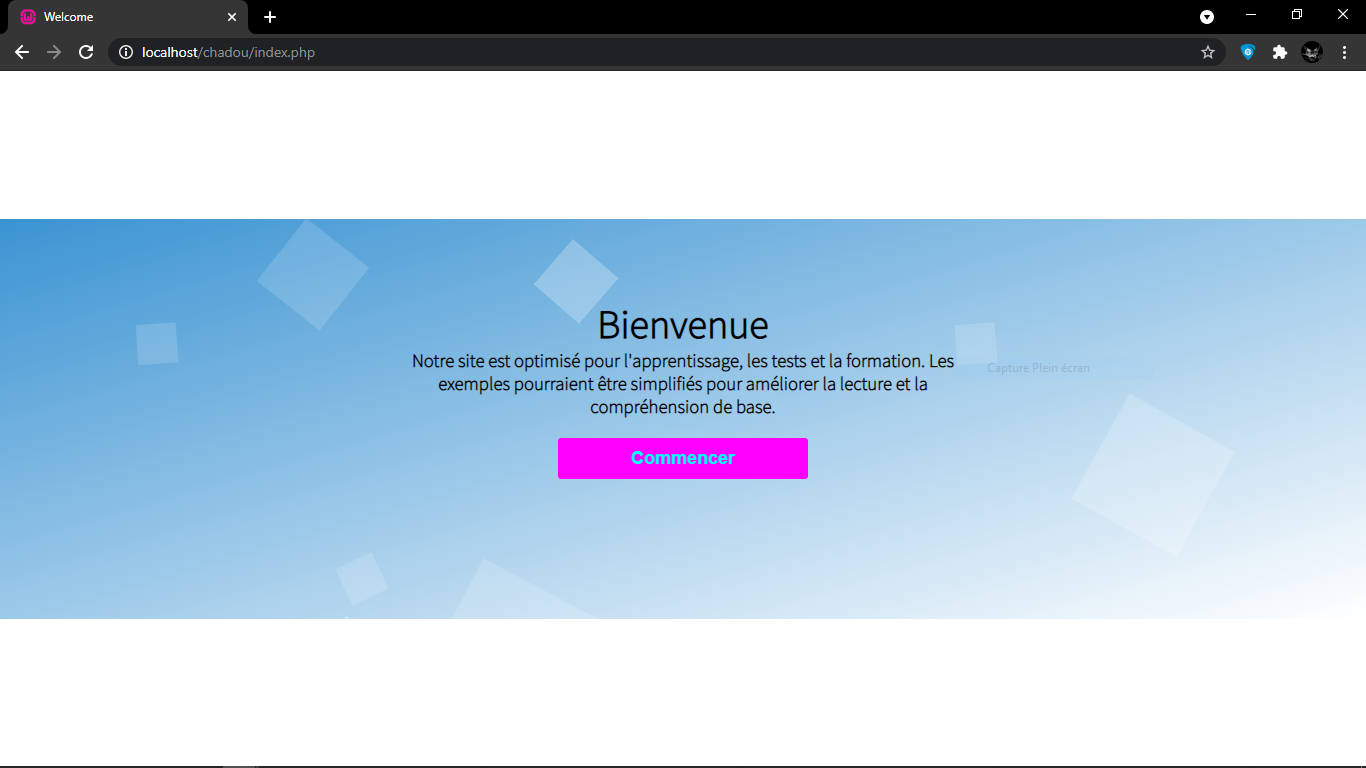


Figure 4

* Page de connexion :

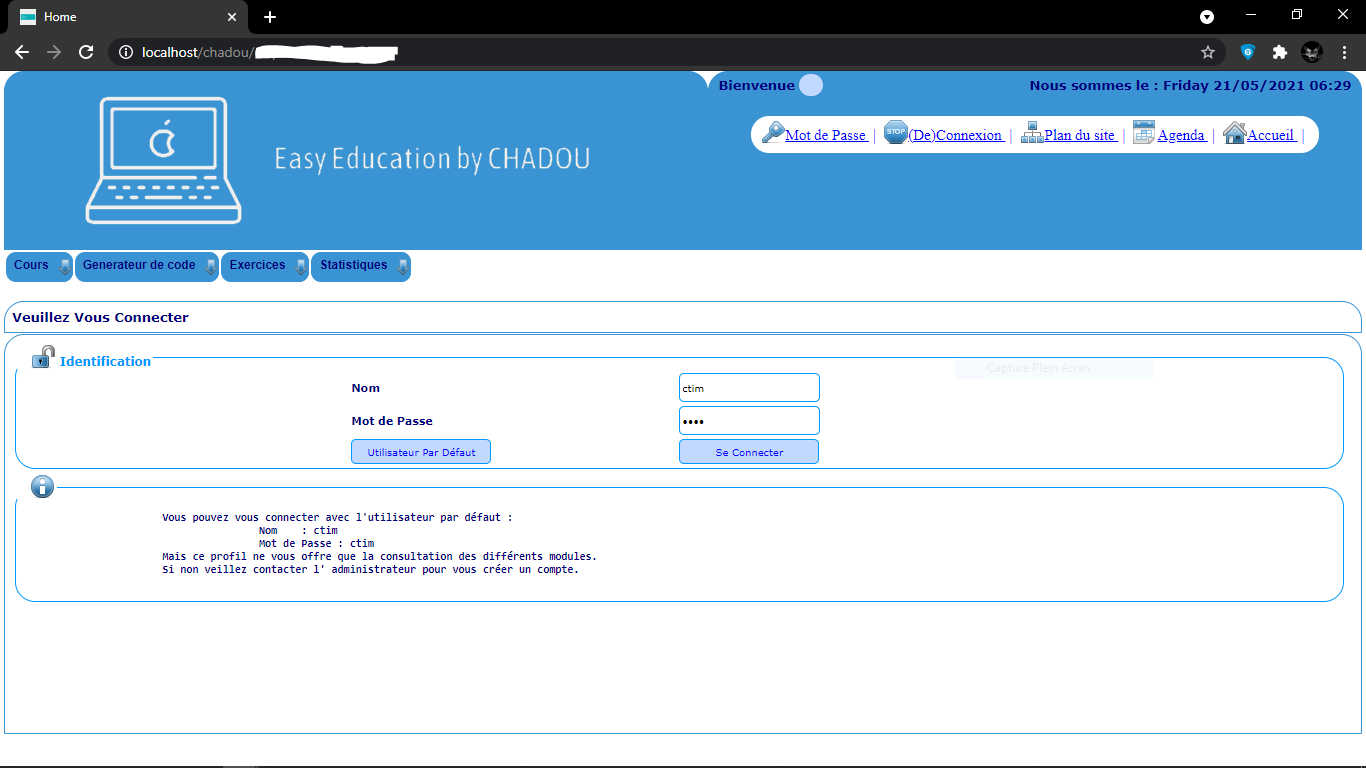


Figure 5

* Page principale :

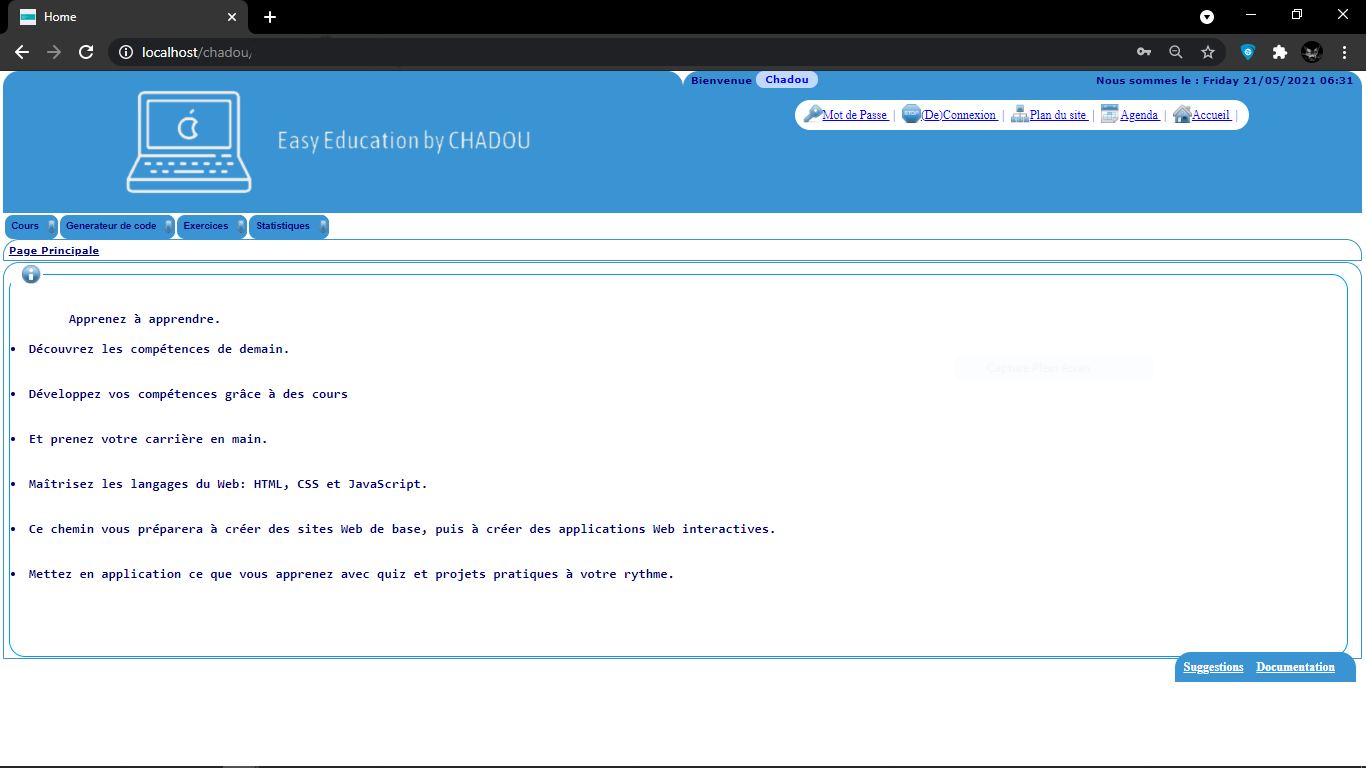


Figure 6

* Page exemple de cours :

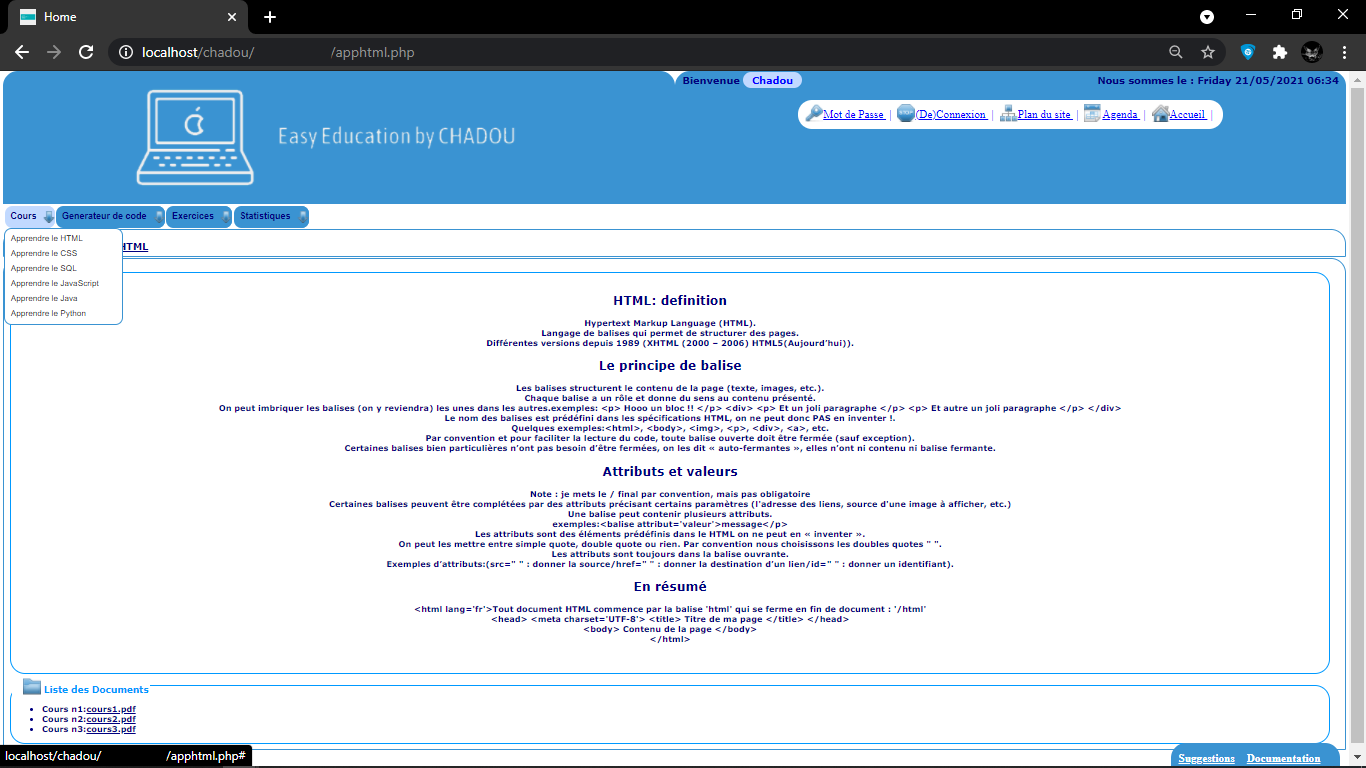


Figure 7

* Page de générateur de code :



Figure 8

* Page de suggestions :

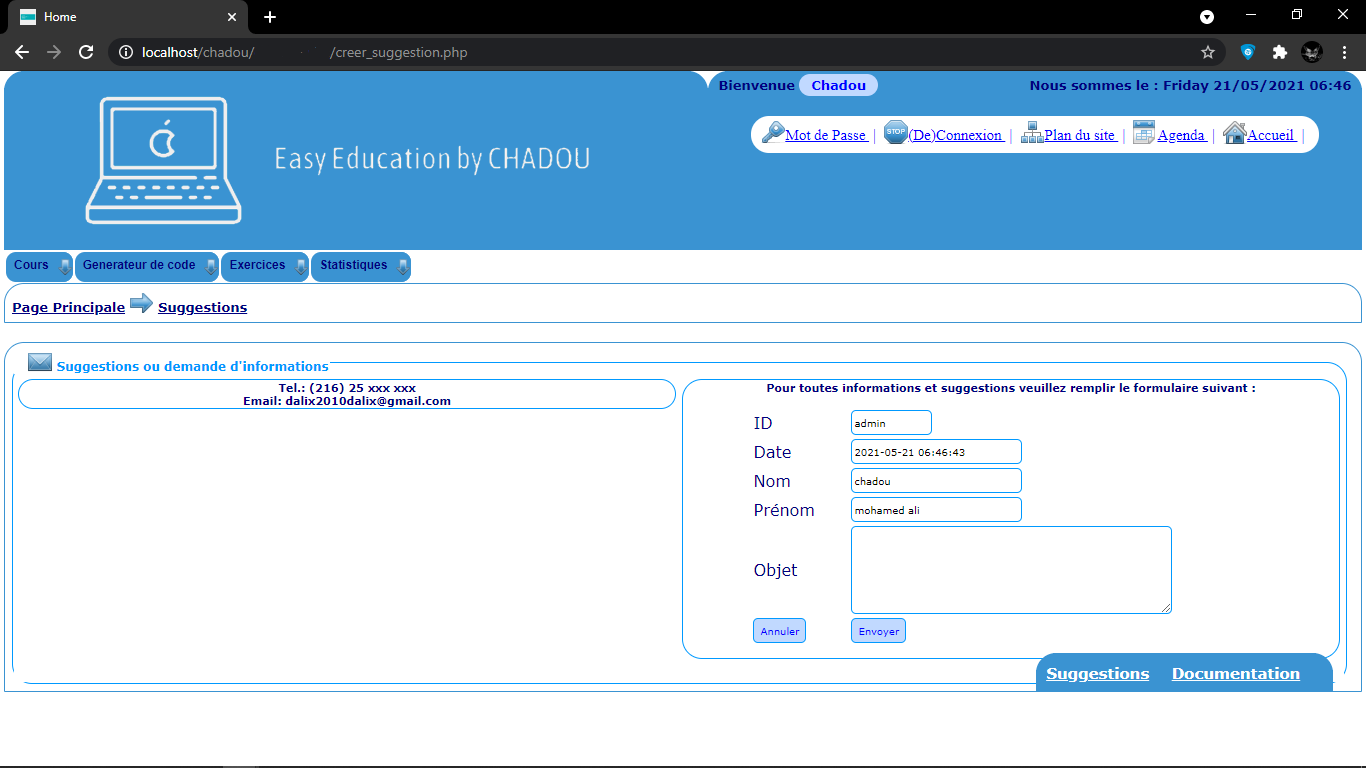


Figure 9

* Page de suivi suggestions :

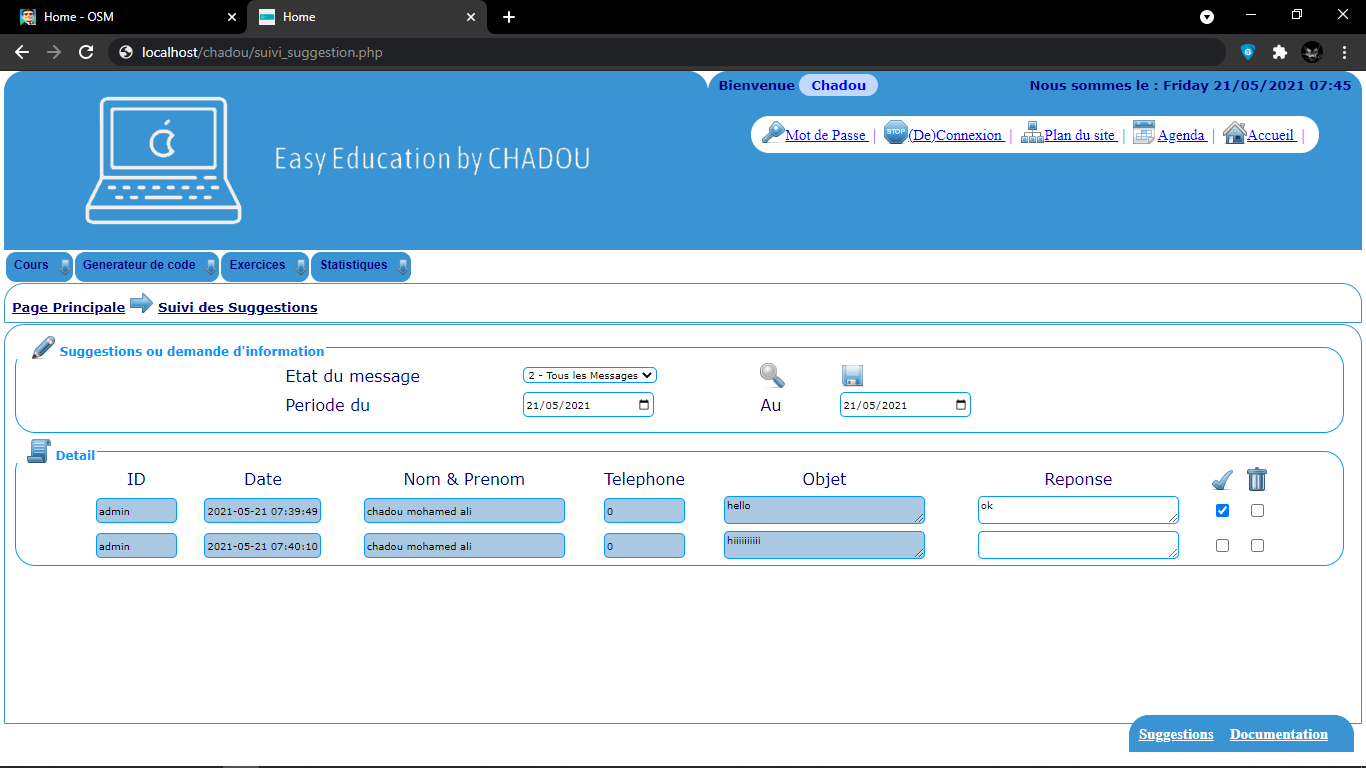


Figure 10

* Page documentation :

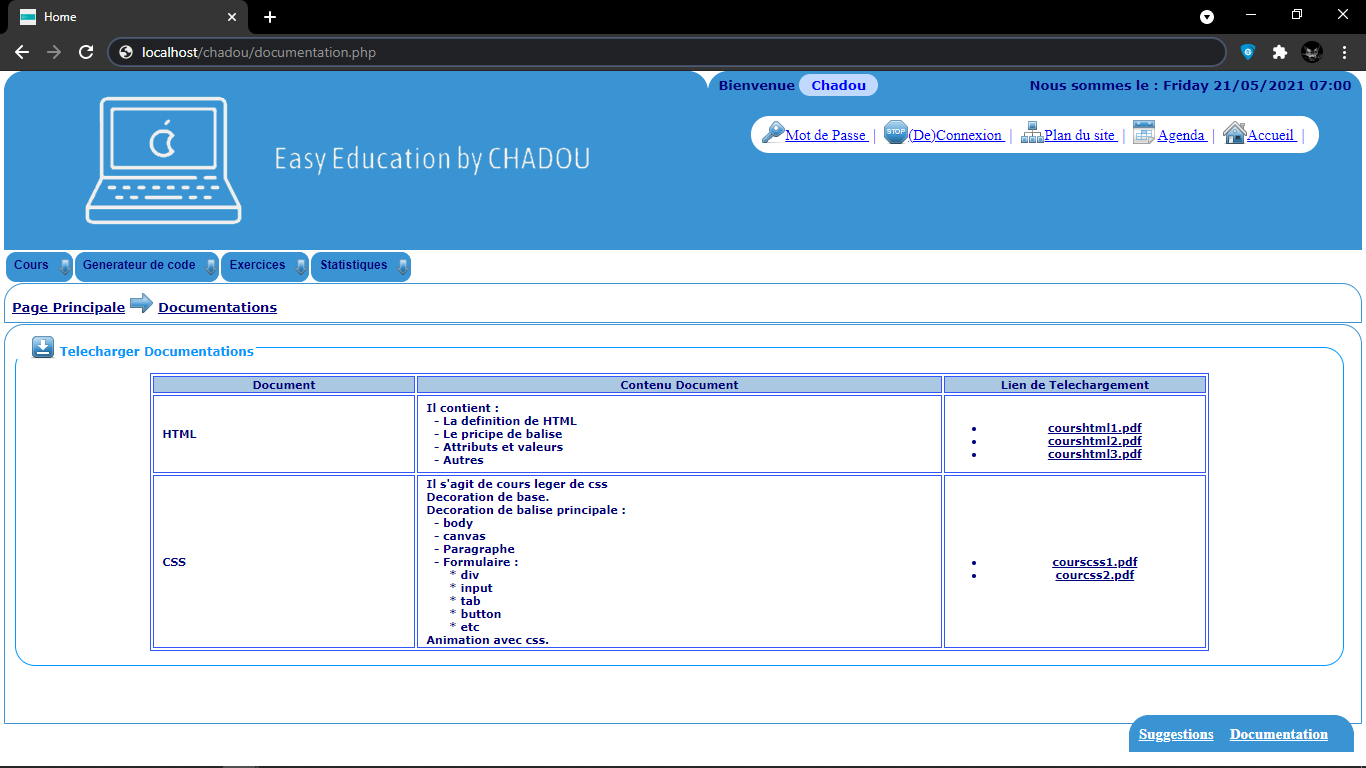


Figure 11

* Page pour plan de site :

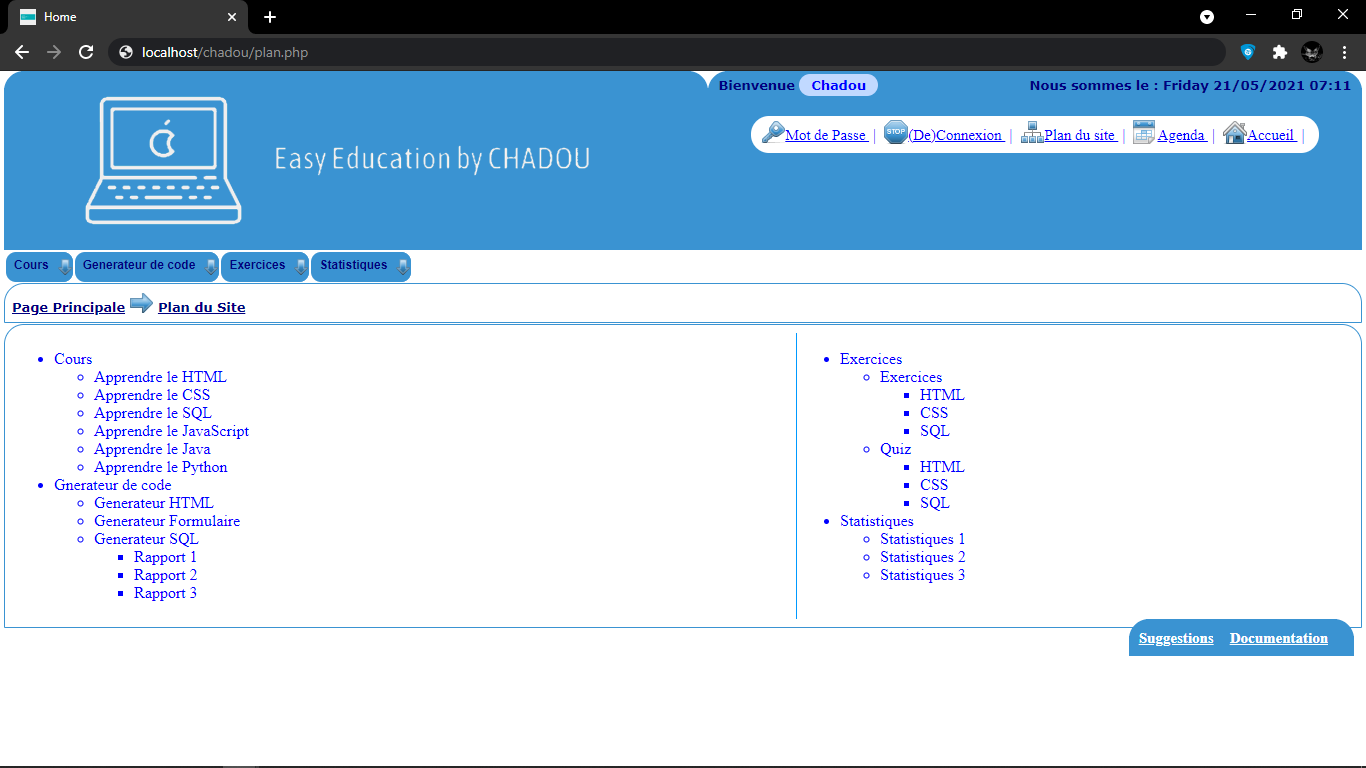


Figure 12

* Capture de base de données :

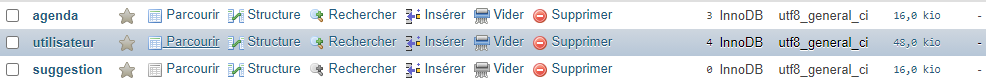


Figure 13

**Conclusion**

Dans ce chapitre, nous avons présenté quelques interfaces de du coté web de site et nous avons démontré quelques scénarios de fonctionnement que nous avons appliqué pour vérifier que notre site fonctionne correctement.

***Conclusion et perspectives***

Dans ce rapport, nous avons exposé les étapes de conception et de développement de notre cite qui consiste à créer une application mobile/web d’apprentissage. C'est un exemple concret du rôle d’apprentissage dans le domaine du développement utile dans la vie courante.

Notre travail s'est déroulé sur trois étapes. Nous avons commencé par une étude de l'existant, suivie de la proposition d'une solution adéquate.

Dans la deuxième phase, nous avons spécifié les besoins fonctionnels et non fonctionnels que doit respecter l’application suivis par des diagrammes des cas d’utilisation mettant en jeu les acteurs qui interagissent avec le système.

La troisième étape concernait la conception de notre application en utilisant des diagrammes de classes.

Dans la dernière phase, nous avons évoqué les différentes technologies utilisées ainsi que l'implémentation de notre système.

Ce projet se situe en effet, dans le cadre du projet de fin de la deuxième année du cycle ingénieur en informatique. Ce projet était une véritable expérience de développement, qui me permis de bien gérer mes connaissances.

Cependant, nous pouvons encore améliorer cette application en ajoutant autre langage de développement ainsi d’autre outils de générateur de code.de plus, on peut ajouter un Forum pour l’utilisateurs pour facilite la communication entre eux ainsi que le partage de connaissance.

Une autre amélioration concernerait les statistiques du progrès de chaque utilisateur. On peut certainement améliorer l’algorithme de calcul de ce progrès.