Contenu

[Dédicaces 3](#_Toc106045166)

[Remerciements 3](#_Toc106045167)

[**Résumé** 4](#_Toc106045168)

[Introduction générale 5](#_Toc106045169)

[Chapitre 1 : Contexte Générale du projet 5](#_Toc106045170)

[1.1 Introduction 5](#_Toc106045171)

[1.2 Présentation du projet : 5](#_Toc106045172)

[1.3 Objectif du projet 6](#_Toc106045173)

[Conclusion : 6](#_Toc106045174)

[Chapitre 2 : Analyses et spécification 7](#_Toc106045175)

[2.1 Problématique : 7](#_Toc106045176)

[2.2 Etude de l’existant : 7](#_Toc106045177)

[2.3 Les solutions proposées : 8](#_Toc106045178)

[2.4 Les besoins fonctionnels 8](#_Toc106045179)

[2.5 Les besoins non fonctionnels 9](#_Toc106045180)

[Fiabilité : 9](#_Toc106045181)

[Les erreurs : 9](#_Toc106045182)

[La convivialité : 9](#_Toc106045183)

[La sécurité : 9](#_Toc106045184)

[La maintenabilité : 9](#_Toc106045185)

[Exploitabilité : 9](#_Toc106045186)

[Chapitre 3 : Conception 10](#_Toc106045187)

[3.1 Introduction 10](#_Toc106045188)

[3.2 Choix de la méthodologie de modélisation 10](#_Toc106045189)

[3.3 Présentation d’UML 10](#_Toc106045190)

[3.4 Pourquoi choisir UML 11](#_Toc106045191)

[3.5 Analyse 11](#_Toc106045192)

[3.6 Les scenarios des cas d’utilisation 11](#_Toc106045193)

[3.6.1 Spécification des cas d’utilisation global : 11](#_Toc106045194)

[3.6.2 Spécification des scénarios de chaque cas d’utilisation global pour « ADMINISTRATEUR » : 12](#_Toc106045195)

[3.6.3 Spécification des scénarios de chaque cas d’utilisation global pour un proffesseur : 14](#_Toc106045196)

[3.6.4 Spécification des scénarios de chaque cas d’utilisation global pour « eleve »: 15](#_Toc106045197)

[3.7 Diagramme de classe 16](#_Toc106045198)

[Conclusion : 18](#_Toc106045199)

[Chapitre 4 : Réalisation 19](#_Toc106045200)

[4.1 Introduction : 19](#_Toc106045201)

[4.2 Architecture de l’application : 19](#_Toc106045202)

[4.2.1 Front-End 19](#_Toc106045203)

[4.2.2. Back-End 20](#_Toc106045204)

[4.3 Technologies et Framework adoptées : 20](#_Toc106045205)

[4.3.1 Framework laravel : 20](#_Toc106045206)

[4.3.2 JSON : 20](#_Toc106045207)

[4.3.3 Framework Bootstrap : 21](#_Toc106045208)

[4.3.4 Laravel: 21](#_Toc106045209)

[4.3.5 MySQL 21](#_Toc106045210)

[4.4 Environnement de développement : 22](#_Toc106045211)

[4.4.1 Visual Studio Code : 22](#_Toc106045212)

[4.4.2 GitHub : 22](#_Toc106045213)

[4.5 Présentation des interfaces : 22](#_Toc106045214)

[4.5.1 Interface d’accueil: 23](#_Toc106045215)

[4.6 Interface d’Administrateur: 23](#_Toc106045216)

[4.6.3 les classes 26](#_Toc106045217)

[4.6.3.1 ajouter un ou plusieur classe 26](#_Toc106045219)

[4.6.3.2 suprimer un ou plusieur classe 27](#_Toc106045221)

[4.6.4 les sous-classe 27](#_Toc106045223)

[4.6.4.1 liste des sous-classe 27](#_Toc106045224)

[4.6.4.2 ajouter des sous-classe 28](#_Toc106045226)

[4.6.5 les élèves 28](#_Toc106045228)

[4.6.5.1 liste des élèves 28](#_Toc106045229)

[4.6.5.2 ajouter un élève 29](#_Toc106045231)

[4.6.5.3 liste des promotions 29](#_Toc106045233)

[4.6.5.4 ajouter un promotion 30](#_Toc106045235)

[**4.6.6** enseignants 30](#_Toc106045237)

[**4.6.7** les parents 32](#_Toc106045238)

[4.6.8 les factures 33](#_Toc106045239)

[**4.6.9** la présence et l’absence 33](#_Toc106045240)

[**4.6.10** les matiere 33](#_Toc106045241)

[***4.6.11*** *liste des examens* 34](#_Toc106045242)

[**4.6.13** ajouter un examen 35](#_Toc106045243)

[4.6.13 liste des cours en ligne 35](#_Toc106045244)

[4.6.14 ajouter des cours en ligne 36](#_Toc106045245)

[**4.7** interface l’enseigant 36](#_Toc106045246)

Dédicaces

A nos chers parents pour tous leurs sacrifices toujours consentis pour le bien-être de leurs chers enfants.

A mon oncle paternelle pour son effort sans limites, la confiance qu’il nous accorde et pour tout le soutien qu’il n’a jamais cessé de nous apporter.

Remerciements

Nous remercions Dieu le tout puissant de nous avoir donné la santé et la volonté d’entamer et de terminer ce travail.

Tout d’abord, ce travail ne serait pas aussi riche et n’aurait pas pu avoir le jour sans l’aide et l’encadrement du professeur Nom, nous le remercions pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant la préparation de ce Projet.

Au terme de ce travail, il nous est vraiment agréable de nous acquitter d’une dette de reconnaissance auprès de toutes les personnes, dont l’intervention au cours de notre Projet, a favorisé son aboutissement.

**Résumé**

Le travail est présenté dans le cadre d'un mémoire de licence en Réseaux informatique et télécommunications, il a pour objet de mettre en œuvre une application web permettant la gestion de plate-fomre d’éducation à des biens externes pour faciliter l’éducation a celle-ci.

La conception de notre application a été faite par le langage de modélisation Unified Modeling Language (UML). Concernant l’implémentation, on a utilisé plusieurs technologies en se basant sur laravel pour la partie Back-end, vuejs et Bootstrap pour la partie front-end qui sont des Framework permettant de créer d’une maniere efficace des applications web, et MYSQL, PHPMyAdmin comme Système de Gestion de Base de données (SGBD).

Ce document définit le projet et son contexte et décrit les étapes nécessaires à sa réalisation.

# Introduction générale

Tealem est une plate-forme éducative, comme edX, qui vise à fournir un enseignement de niveau collégial et une formation de haute qualité à l'échelle mondiale.

Il propose des diplômes en ligne, des cours spécialisés et des activités ponctuelles proposés par des universités partenaires renommées du monde entier. La plupart des cours Coursera nécessitent une participation active, des commentaires, des devoirs écrits et la participation à des conférences. L'effort est payant, car après avoir terminé le cours, on peut demander un certificat avec un tampon approprié dessus.

Mais des logiciels multifonctions permettent de gérer l’activité de la plate-forme dans toute sa globalité. Ces logiciels offrent en effet l’avantage de donner accès à différents outils précieux pour les étudiants , faire de l’inter-agence et ainsi partager en temps réel de précieuses informations entre professionnels.

C’est dans ce cadre que situe notre projet de fin d’étude qui consiste à concevoir et de développer une application web pour la plate-forme d’éducation .Ce rapport décrit les étapes de développement de notre projet.

# Chapitre 1 : Contexte Générale du projet

## Introduction

Ce chapitre présente l’application, le contexte général du projet ainsi que l’objectif du projet

## Présentation du projet :

Le projet vise principalement la création d'une application web pour une plate-forme éducative. Le projet peut être divisé en trois parties :

1. **Gestion administrateur :**

C'est un aspect très important . Il permet a **administrateur** de gérer et de visualiser tous la plate-forme car il fournit une interface qui le permet d'effectuer leurs tâches de manière fluide.

1. **Gestion des professeurs:**

Le professeur il permet de gérer ces élevés et de faire ces cours online

D’autre part il permet de faire les examens et la présence et l’absence et il donne des rapports a l’administrateur

1. **Gestion des élevés:**

**(en cour …)**

## Objectif du projet

C’est un secret de polichinelle de dire que notre pays est très mal lotis en matière d’éducation à distance, même si nous constatons depuis quelques années l’émergence de certaine tentatives visant à rattraper ce retard, il n’en demeure pas moins que notre société est loin de tirer profit des avantages qu’offre l’internet en matière d’éducation pour les citoyens du monde au vingtième siècle. C’est dans ce contexte que cette application a été développée et elle vise essentiellement à mettre en place une plateforme centré sur l’éducation et qui constitue, je l’espère, une contribution de ma part dans les efforts visant à aider notre pays à relever les défis et à occuper la place qu’il mérite.

Notre application est une plateforme qui sert à gérer toutes les activités de formation que peut offrir un centre de formation ou institution éducative, de ce fait, elle peut être considérer comme une alternative à un enseignement présentiel, et peut être également aperçu comme un outil facilitateur qui apporte un concours précieux aux étudiants mauritaniens quels que soient la place ou l’institution dans lesquelles ils sont inscrits.

## **Conclusion :**

Ce chapitre nous permet de présenter le contexte de notre projet et d’expliquer les objectifs. Pour le chapitre qui suit, nous l’avons consacré à l’analyse et spécification de l’application.

# Chapitre 2 : Analyses et spécification

## Problématique :

Aujourd'hui, obtenir un bien immobilier dans la ville de Nouakchott sans vous déplacer est une véritable victoire, commençant par le processus de recherche d'un bien immobilier et se terminant par la présence d'un bien immobilier répondant aux aspirations.

La pluparts des client ont des difficultés envisagées actuellement lors de la recherche de vente ou location d’un bien immobilier.

Le secteur de la gestion et de la commercialisation des biens immobiliers sont confrontés à plusieurs problèmes car ils s'appuient sur un système de gestion traditionnelle qui aboutit souvent à un problème avec les clients en raison d’absence de l’automatisation des processus immobilier.

En ce qui concerne les propriétaires, la plupart d’eux rencontrent également des problèmes pour diffuser leurs annonces immobiliers au bon client en raison du manque d'informations liées à la publication des offres immobilières, en particulier pour les personnes qui n’ont aucun canal publicitaire ou qui sont novices dans le domaine et qui ne possèdent pas tant de maisons.

## Etude de l’existant :

Aujourd'hui, plusieurs milieux utilisent la technologie web afin de rendre facile les taches liées à la communication dont les annonces y font parties.

Si on regarde le secteur immobilier, on constate un net recul du secteur. Ce recul impact sur l’approchement des services vers les clients qui recherchent un bien et les propriétaires qui souhaitent exposer leurs biens.

Le manque de numérisation du secteur est considéré comme un principal problème, et l’absence de centres de service immobiliers avancés dans les agences, où le client est contraint de quitter son domicile et de se renseigner sur chaque agence.

Il y a aussi un manque de satisfaction des propriétaires avec les services fournis par les ces agences.

## Les solutions proposées :

Pour résoudre ces problèmes, on propose de développer une application web de gestion immobilière et de publication des annonces. Cette application doit :

1. Répondre aux besoins des utilisateurs selon leur catégorie, car elle permettra aux employés de l'établissement de gérer entièrement toutes leurs propriétés, en plus de permettre au client de visualiser les annonces et d'envoyer des demandes d’achat ou de réservations qui peuvent se transformer en une vente ou location si la demande est acceptée.
2. Offrir la possibilité d'authentification pour certains propriétaires qui gèrent leur maison, pour accéder à leur propre plateforme : gèrent leurs demandes et communiquent avec leurs clients.
3. Fournir la sécurité des données en permettant aux administrateurs de gérer les utilisateurs du système ainsi que leurs rôles.
4. Fournir une interface très responsive et comprenant une carte électronique pour faciliter le processus de recherche pour le client, car elle lui permet de prévisualiser les immobiliers annones disponibles dans chaque région.

## **Les besoins fonctionnels**

L’application actuelle couvre un large espace présentant une gamme de services assurant certaines fonctionnalités. Nous allons définir ici les différents besoins fonctionnels du système

* Les classes
* Les étudiants
* Gérer les professeurs
* Gérer les parents
* Gérer les comptes
* Gérer la présence et l’absence
* Gérer les matières scolaires
* Gérer les examens
* Gérer la bibliothèque
* Gérer les classes en ligne
* Gérer la réglage
* Gérer les utilisateurs

## Les besoins non fonctionnels

Présentation les objectifs liés aux performances du système et aux contraintes de son environnement.

### Fiabilité :

L’application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

### Les erreurs :

Les ambigüités doivent être signalées par des messages d’erreurs bien organisés pour bien guider l’utilisateur et le familiariser avec notre site web.

### La convivialité :

Le système doit avoir une interface conviviale permettant un dialogue simple entre l’utilisateur et l’application.

### La sécurité :

Le système doit être fiable et sécurisé : il impose l’authentification pour accéder à l’application à eleve, propriétaire ou professeur qui veut avois des cours ou de conferance. Les mots de passe ne sont enregistrés dans la base qu’après un chiffrement.

### La maintenabilité :

Le code doit être compréhensible par simple lecture, notamment en respectant les règles de gestion et les normes de développement.

### Exploitabilité :

Les impacts, en termes de performances, doivent être pris en compte lors de développement, ainsi que la consommation des ressources (CPU, mémoire, etc.) qui doit être minimisée.

# Chapitre 3 : Conception

## Introduction

Dans le chapitre précédent, nous avons défini notre champ d’étude et ses limites, nous avons aussi présenté la solution préconisée. Nous présentons dans ce chapitre l’étape cruciale qui consiste à concevoir notre projet « plate-forme d’éducation». Nous commencerons par la présentation d’UML, pour la modélisation de notre système (phase analyse et spécification des besoins). Par la suite nous présenterons une vue qui exprime les modules et les exécutables physiques sans aller à la réalisation concrète du système. Nous allons mettre l’accent sur les diagrammes de cas d’utilisation, classe et relationnel.

## Choix de la méthodologie de modélisation

Dans la cadre de notre projet, nous avons opté pour le langage UML comme une approche de conception. Ci-dessous, nous présentons ce langage puis nous justifions notre choix.

## Présentation **d’UML**

UML se définit comme un langage de modélisation graphique et textuel, il est destiné à comprendre et décrire des besoins, spécifier et documenter les systèmes, et sert aussi à esquisser des architectures logicielles, concevoir des solutions et communiquer des points de l’UML unifie à la fois les notations et les concepts orientés objet. Il ne s’agit pas d’une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d’un langage.

## Pourquoi choisir UML

UML est un langage formel et normalisé, il permet le gain de précision, encourage l’utilisation d’outils et constitue à cet effet un gage de stabilité. UML est un support de communication performant : il cadre l’analyse et facilite la compréhension de représentations abstraites complexe. Son caractère polyvalent et sa souplesse en font un langage universel.

## **Analyse**

La phase d’analyse a pour objectif d’identifier les acteurs qui interagissent avec le système et spécifier les fonctionnalités du système. Cette phase sert aussi à déterminer pour chaque acteur ses droits d’accès à l’utilisation du système.

## Les scenarios des cas d’utilisation

### Spécification des cas d’utilisation global :

|  |  |
| --- | --- |
| Les acteurs | Les cas d’utilisation |
| Administrateur | * Les classes * Les etudiants * Gérer les proffesseurs * Gérer les parents * Gérer comptes * Gérer la présence et l’absence * Gérer les matières scolaires * Gérer les examens * Gérer la bibliothéque * Gérer les classes en ligne * Gérer la réglage * Gérer les utilisateurs |
| Professeur | * Authentifier * Les classes * Gérer Les étudiants * Gérer les examens * Gérer les rapports * Gérer les profil personnelle |
| etudiant | * Authentification * Presance dans la cours en ligne * profil personnelle * le note * faire des examens * obtenir un diplôme |
|  |  |

Tableau 1: Spécification des cas d’utilisation global

### Spécification des scénarios de chaque cas d’utilisation global pour « ADMINISTRATEUR » :

|  |  |
| --- | --- |
| Cas D’utilisation | Scénarios |
| S’authentifier dans l’espace | * saisir le login * saisir le mot de passe |
| * Gérer les classes | * ajouter un classes * liste des classes * modifier les classes * Supprimer des classes |
| * Les étudiants | * ajouter un étudiants * liste les étudiants * modifier les étudiants * ajouter des images à un etudiants |
| * Gérer les professeurs | * ajouter une professeurs * modifier les professeurs * supprimer des professeur * liste des professeurs * affecter un professeur à un classe |
| * Gérer les matières scolaires * Gérer les matières scolaires | * ajouter un matières * liste les matières * modifier les matières * ajouter des matières * supprimer des matières * affecter un matières à un classe |
| Gérer comptes | * cliquer sur le menu(gestion contrats) * ajouter un contrat (location, vente) * modifier les contrats (location, vente) * supprimer des contrats (location, Vente) |
| bibliothéque | * ajouter un livres * modifier les livres * liste des livres * supprimer des livres |
| Gérer les examens | * cliquer sur le menu(gestion location) * ajouter un examens * modifier examens * supprimer examens * affecter des question a un examen |
|  | * liste de classe de ce proffesseur |
| Gérer les clients | * cliquer sur le menu(gestion clients) * ajouter client * modifier client * supprimer client |

Tableau 2: Spécification des scénarios de chaque cas d’utilisation global pour Administrateur

### Spécification des scénarios de chaque cas d’utilisation global pour un professeur :

|  |  |
| --- | --- |
| Cas d’utilisation | Scénarios |
| S’authentifier dans l’espace | * saisir le login * saisir le mot de passe |
| Gérer les classes | * liste de classe de ce professeur |
| Gérer les examen | * liste de classe de ce professeur * liste de question de ce professeur |

Tableau 3: Spécification des scénarios de chaque cas d’utilisation global pour un proffesseur

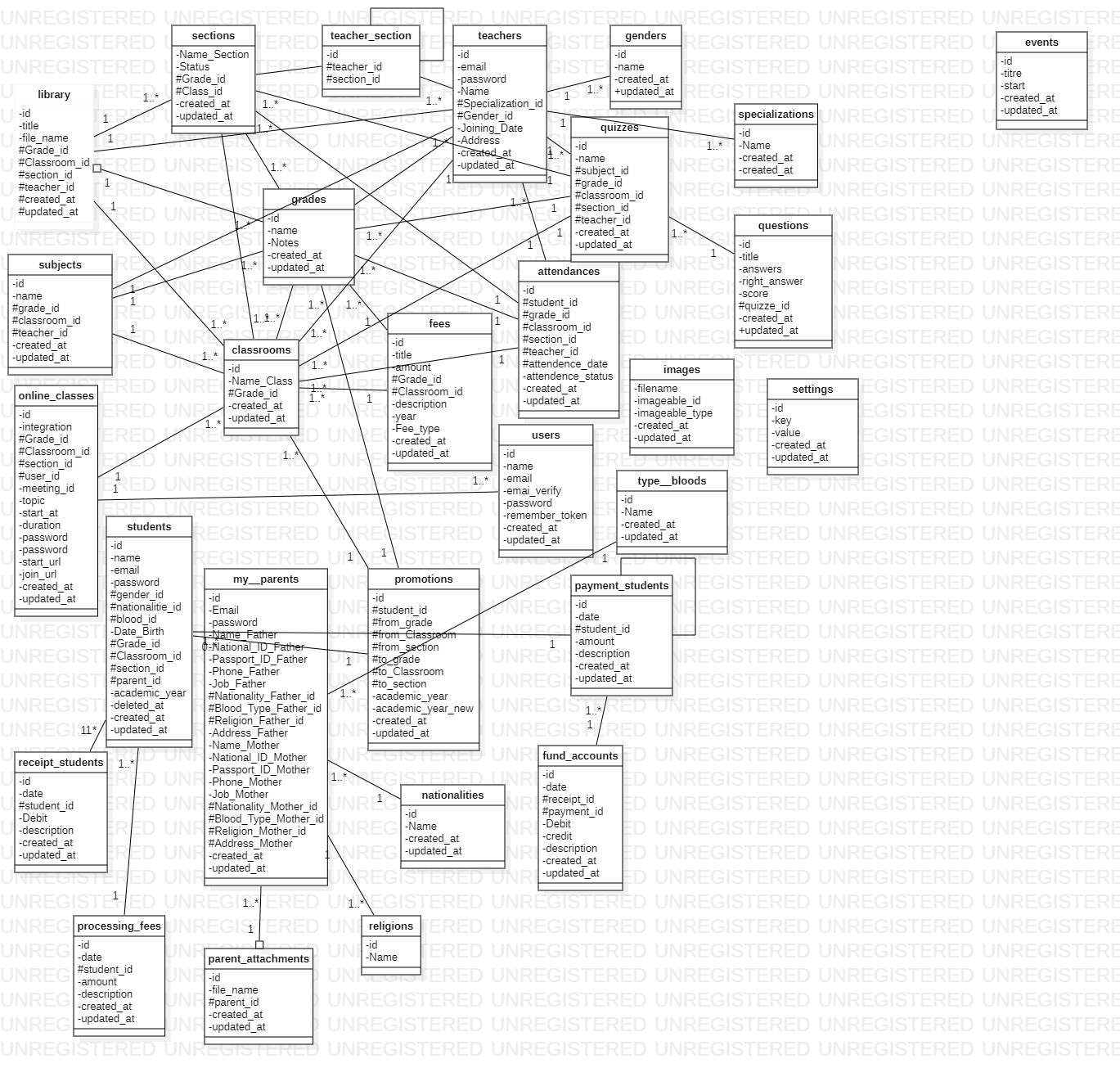
### Spécification des scénarios de chaque cas d’utilisation global pour « élève »:

|  |  |
| --- | --- |
| Cas d’utilisation | Scénarios |
| S’authentifier dans l’espace | * saisir le login * saisir le mot de passe |
| Gérer classe | * accepter une demande de la réunion * profil personnelle * voir le résultat * faire le paiement |
| Gérer bibliothèque | * télécharger un livre |

Tableau 4: Spécification des scénarios de chaque cas d’utilisation global pour eleve

## Diagramme de classe

Le diagramme de classe est considéré comme le plus important dans le développement orienté objet, il représente l’architecture conceptuelle du système, est utilisé pour représenter la vue de conception statique. Son intérêt est de modéliser les entités de l’application et de représenter l’ensemble des informations gérées par l’application.

****

## Conclusion :

Dans ce chapitre, nous avons conçu et documenté le code que nous devons produire. Dans cette phase, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. Dans le chapitre suivant nous avons étudier les outils et les langages utilisés durant la phase de construction

# Chapitre 4 : Réalisation

## Introduction :

Après avoir finaliser l’étape d’analyse et conception, nous passons à l’implémentation de notre application web. Nous commençons tous d’abord par la présentation des différents outils utilisés pour l’implémentation. Puis nous présentons quelques interfaces de notre application. Environnement du développement.

## Architecture de l’application :

Dans les phases préliminaires du développement d’une application, la définition de l’architecture technique est importante, qui consiste à faire les choix de technologies et d’organisation de composants logiciels les plus adaptés aux besoins et aux contraintes de l’organisation d’accueil, et décrit la manière dont seront agencés les différents éléments de l’application et comment ils interagissent entre eux.

### 4.2.1 Front-End

Le Front-end correspondant au côté client sera mis en place à l'aide du framework Angular. L'interface utilisateur (UI) de niveau présentation est l'interface utilisateur dans laquelle elle peut interagir avec l'application. L'UI est construite avec l'engagement de description (HTML, CSS). L'UI sera connectée au composant (construit avec TypeScript) qui gère les données et ce composant aura accès à un ou plusieurs services et dans ce service qui aura accès au Back-end (serveur). Le service contient plusieurs méthodes qui peuvent envoyer des requêtes http au back-end. Et le résultat renvoyé par les serveurs sera récupéré par le service qui le trasmet au composant et s'affichera par la suite dans l'UI.

### 4.2.2. Back-End

Laravel est un Framework web opensource écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur Git Hub.

Cette architecture offre un avantage essentiel du fait que, comme chaque niveau fonctionne sur sa propre infrastructure, chaque niveau peut être développé simultanément par une équipe de développement distincte, et peut être mis à jour ou mis à l'échelle selon les besoins sans avoir d'incidence sur les autres niveaux.

## Technologies et Framework adoptées :

### Framework laravel :

Laravel est un Framework web opensource écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur Git Hub.

### JSON :

JSON (JavaScript Object Notation) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l’information structurée comme le permet XML par exemple. Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

### Framework Bootstrap :

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

Figure 13: Bootstrap logo

### Laravel:

Laravel est un Framework web opensource écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur Git Hub.



### MySQL

Figure 15: logo MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR) fonctionnant sous Windows et Linux. Il fait partie des logiciels de gestion de base des données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

## Environnement de développement :

### Visual Studio Code :

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l’auto-complétions, la coloration syntaxique, le débogage, et les commandes git.

Figure 18: logo VS Code

### GitHub :

GitHub est une plateforme [open source](https://www.lemagit.fr/definition/Open-Source) de gestion de versions et de [collaboration](https://whatis.techtarget.com/fr/definition/Collaboration) destinée aux développeurs de logiciels. Livrée en tant que logiciel à la demande ([SaaS](https://www.lemagit.fr/definition/SaaS), Software as a Service), la solution GitHub a été lancée en 2008. Elle repose sur Git, un système de gestion de code open source créé par Linus Torvalds dans le but d'accélérer le développement logiciel. Git permet de stocker le code source d'un projet et de suivre l'historique complet de toutes les modifications apportées à ce code. Grâce aux outils qu'elle fournit pour gérer les conflits éventuels résultant des changements apportés par plusieurs développeurs, il est possible de collaborer efficacement sur un même projet. GitHub facilite la programmation collaborative en mettant une interface Web à disposition du référentiel de code de Git, ainsi que des outils d'administration favorisant la [collaboration](https://whatis.techtarget.com/fr/definition/Collaboration).

Figure 21: logo GitHub

## Présentation des interfaces :

Dans notre application on a quatre type utilisateurs. Par conséquent, nous avons prévu une page d’accueil dans laquelle chaque utilisateur peut le consulter. Il composer de trois phases principales :

### Interface d’accueil:

L'interface suivante s'adresse principalement aux utilisateurs de notre plate-forme.

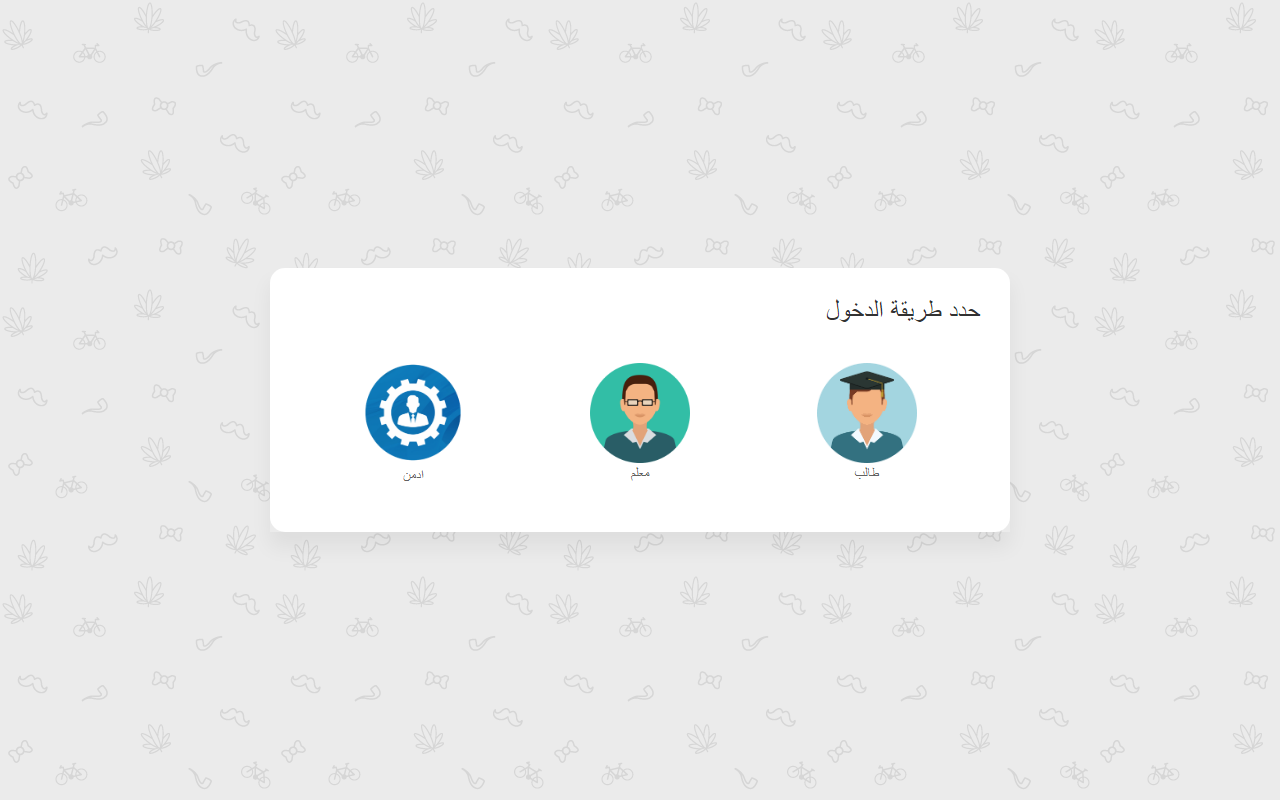
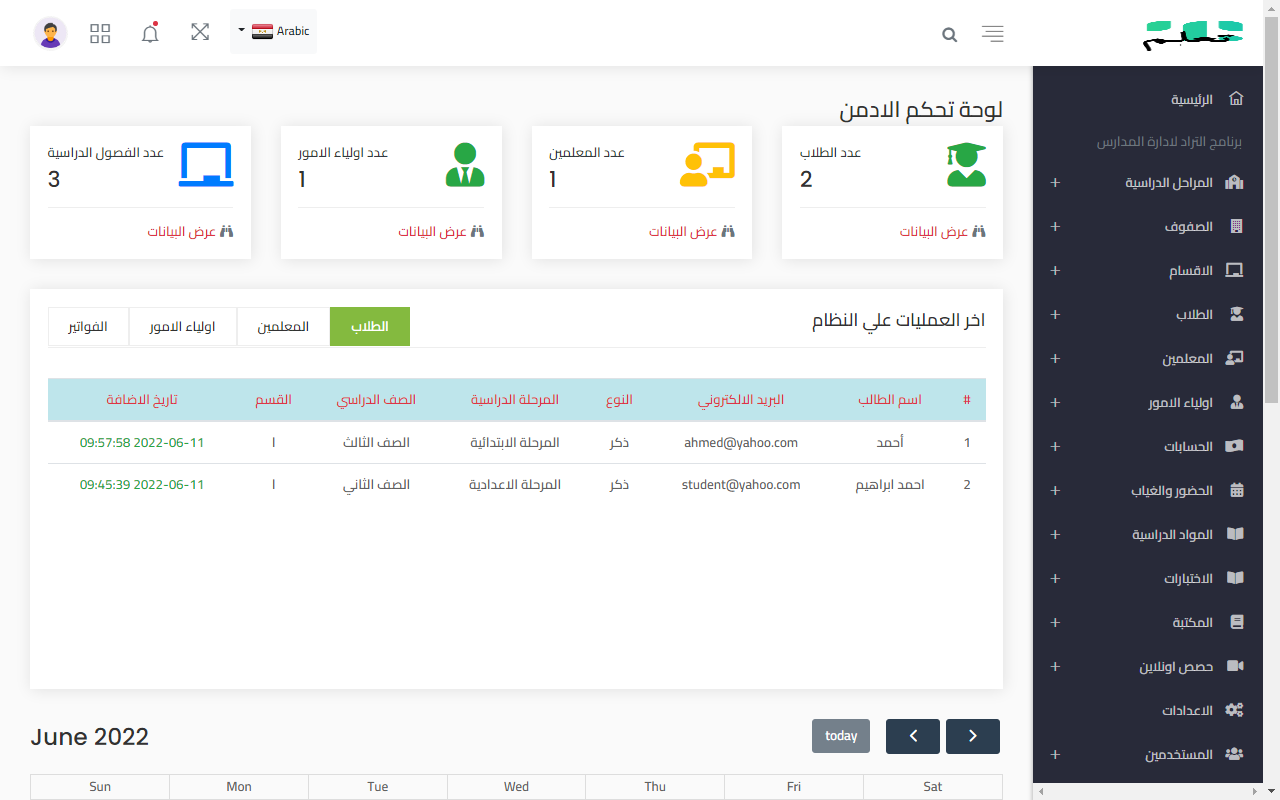


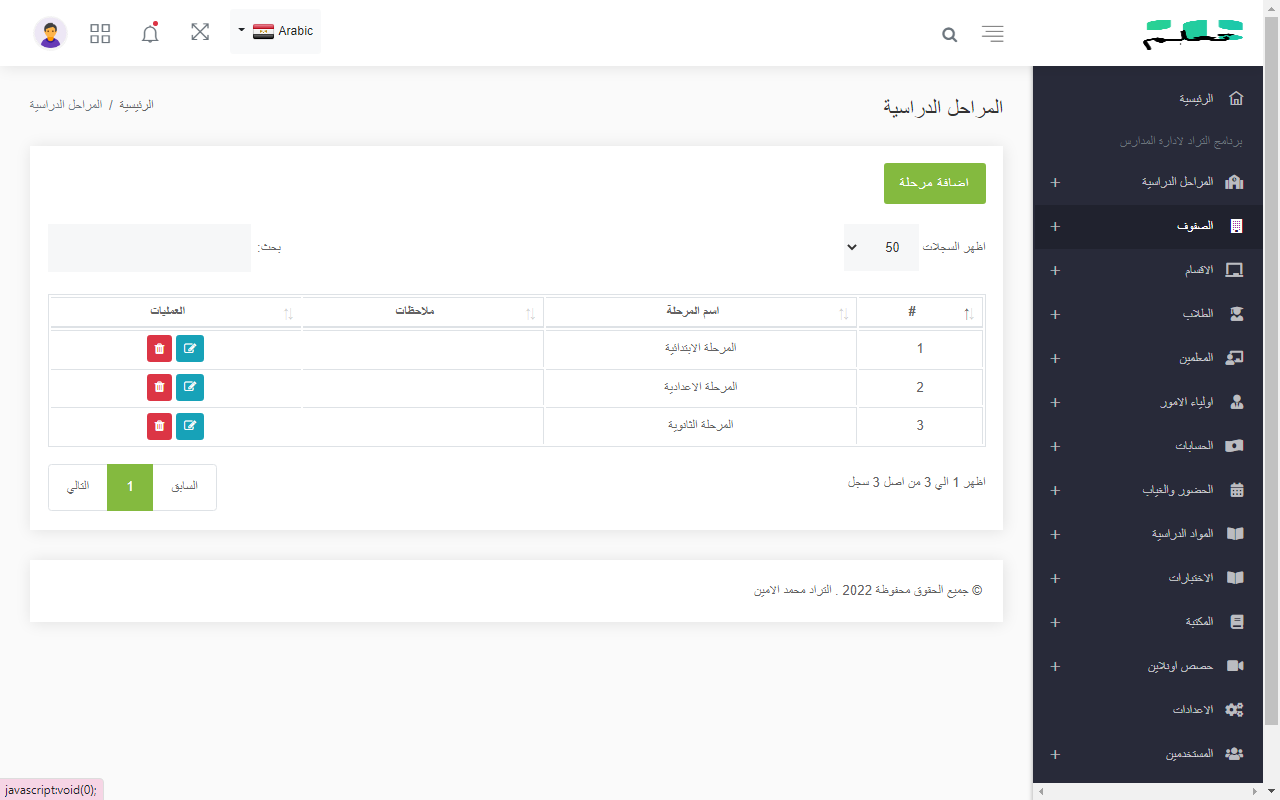
Figure 1: Interface d’accueil – List annonces

### Interface d’Administrateur:

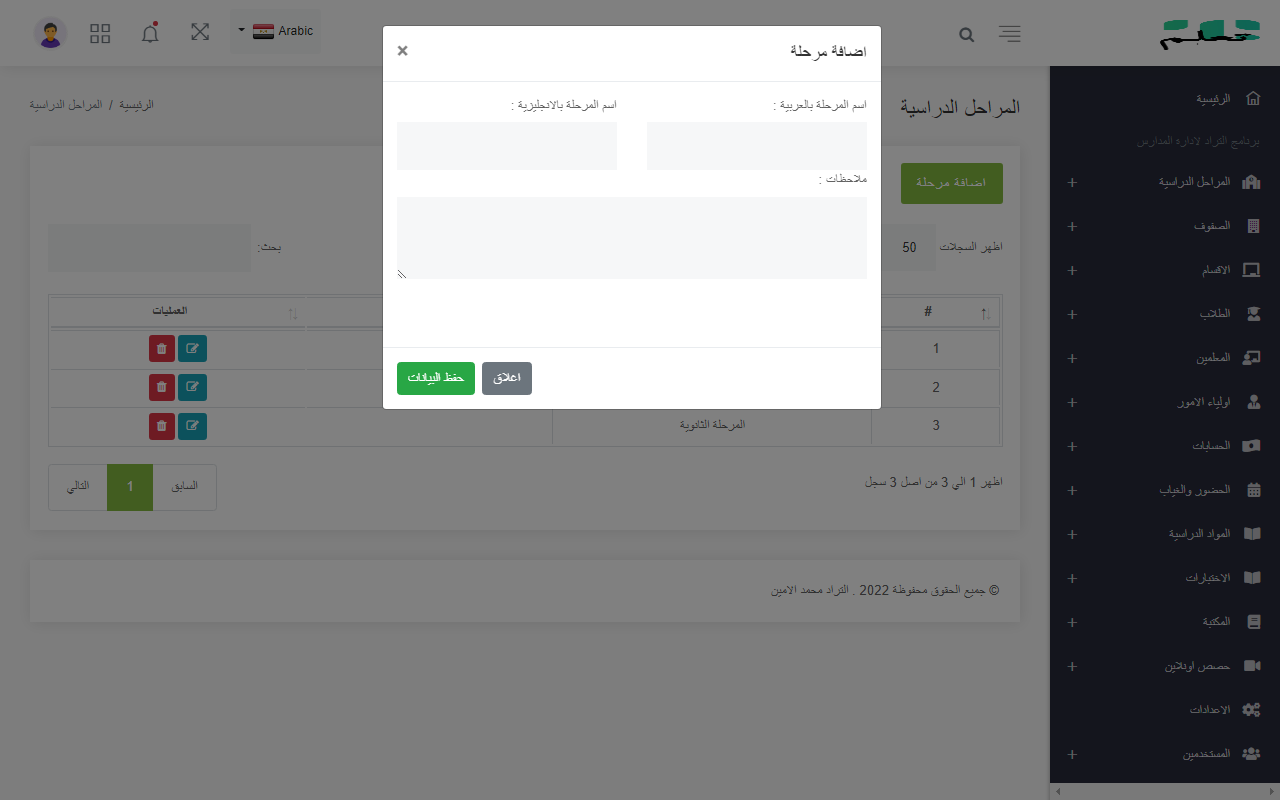
* + 1. login

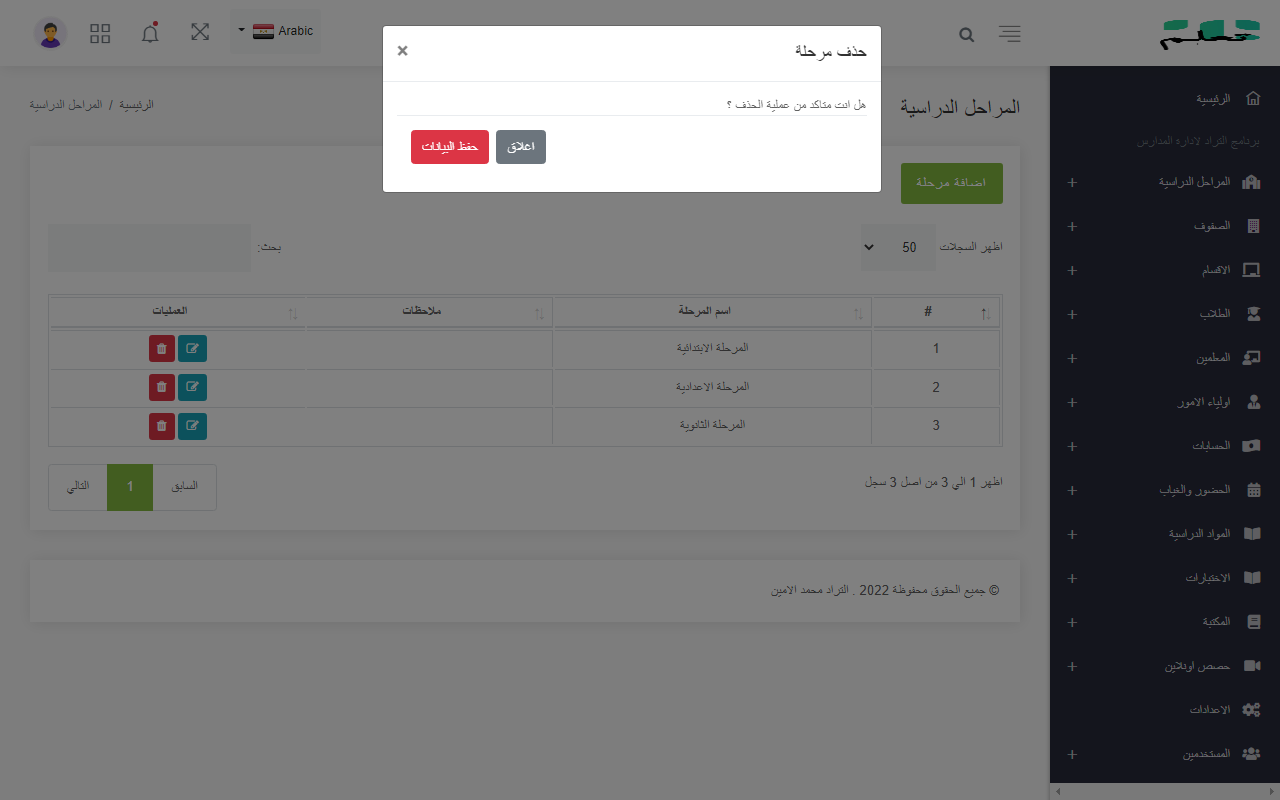


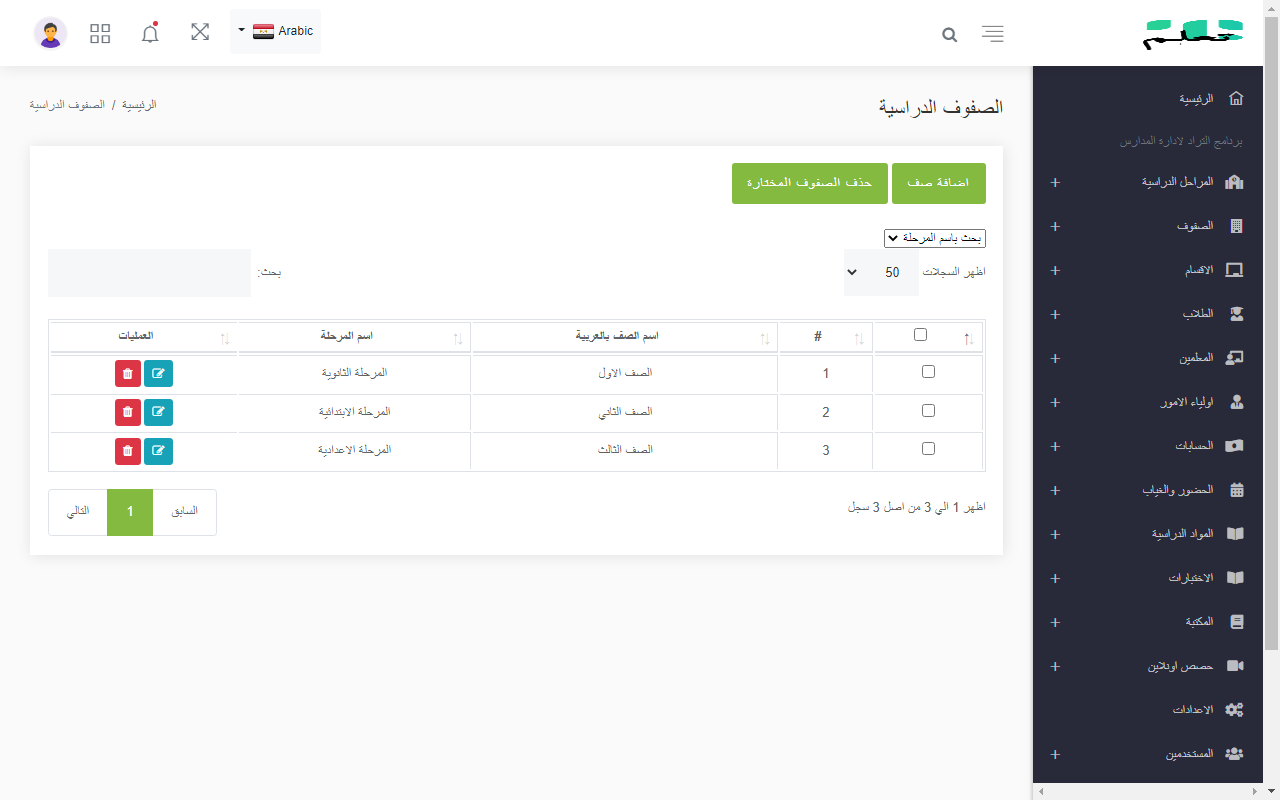
****



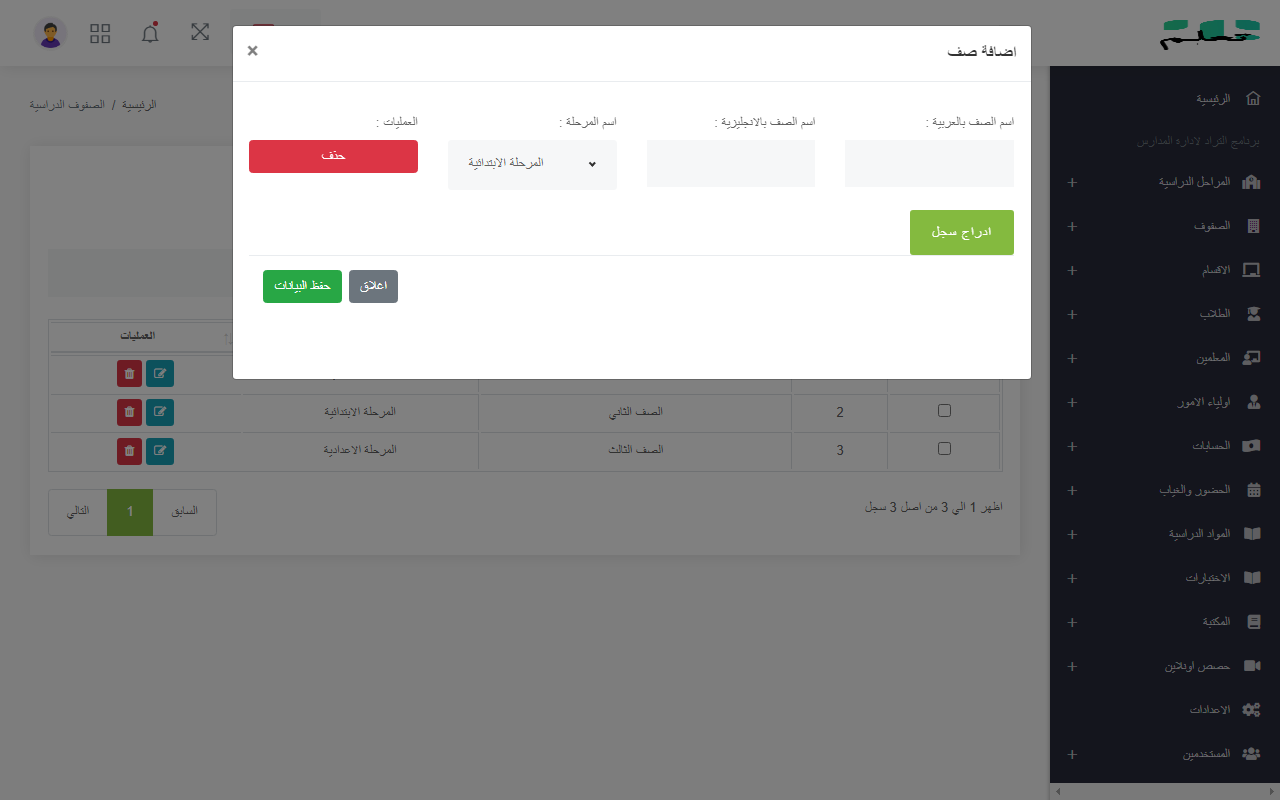
**4.6.2.2** ajouter



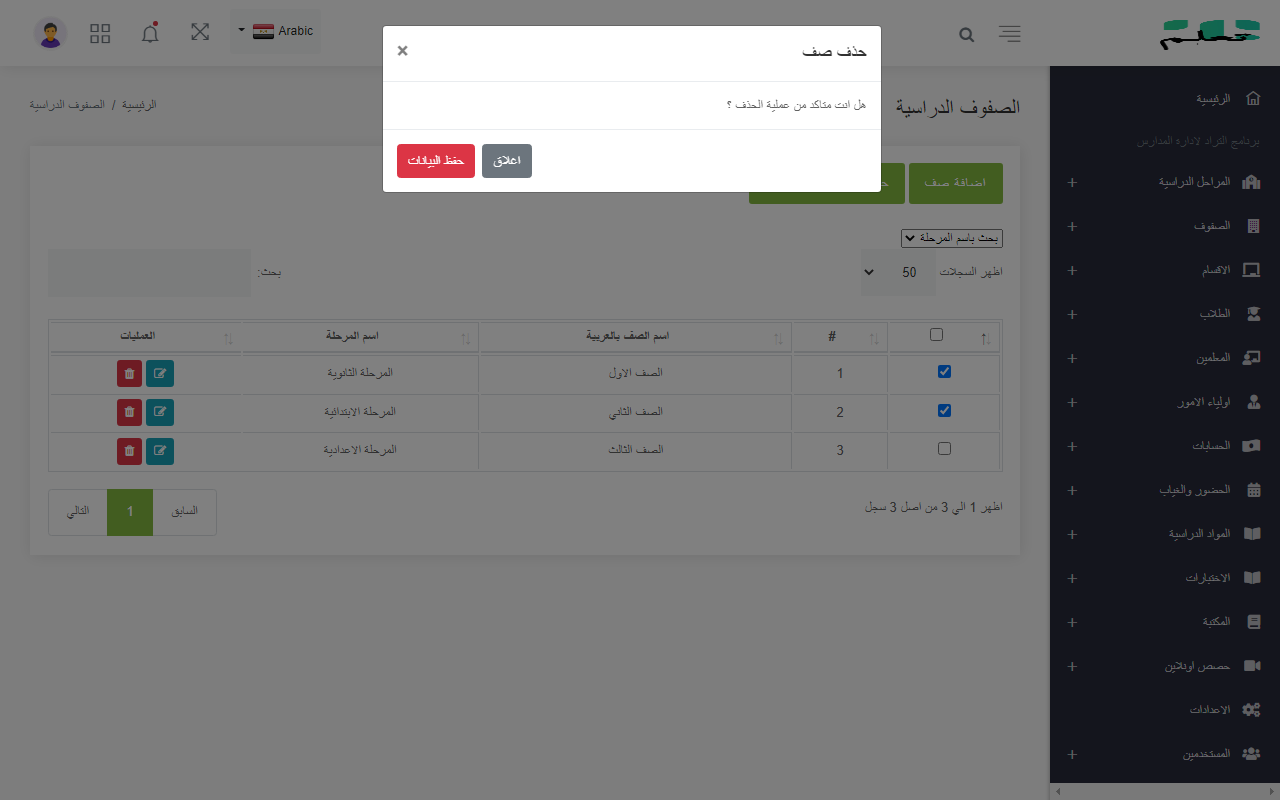
* + - 1. supprimer
    1. les classes



4.6.3.1 ajouter un ou plusieur classe

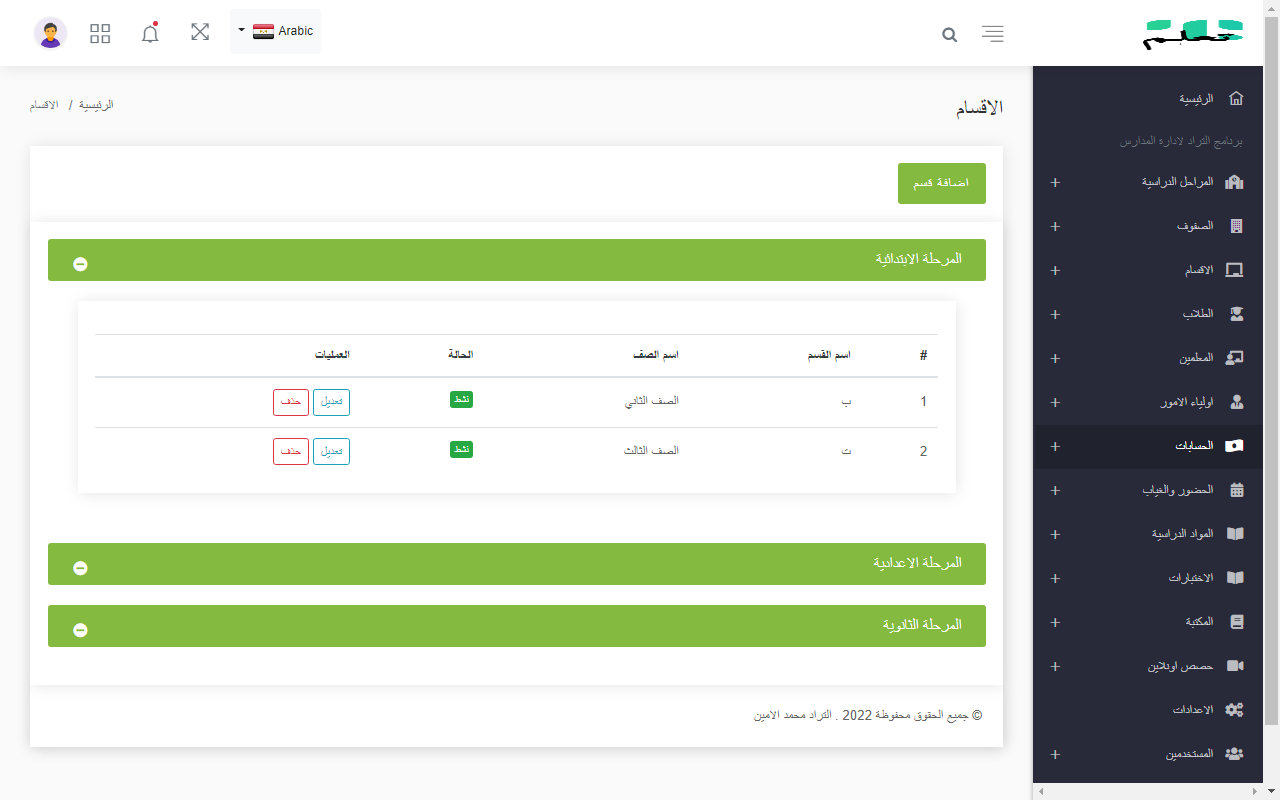


4.6.3.2 suprimer un ou plusieur classe

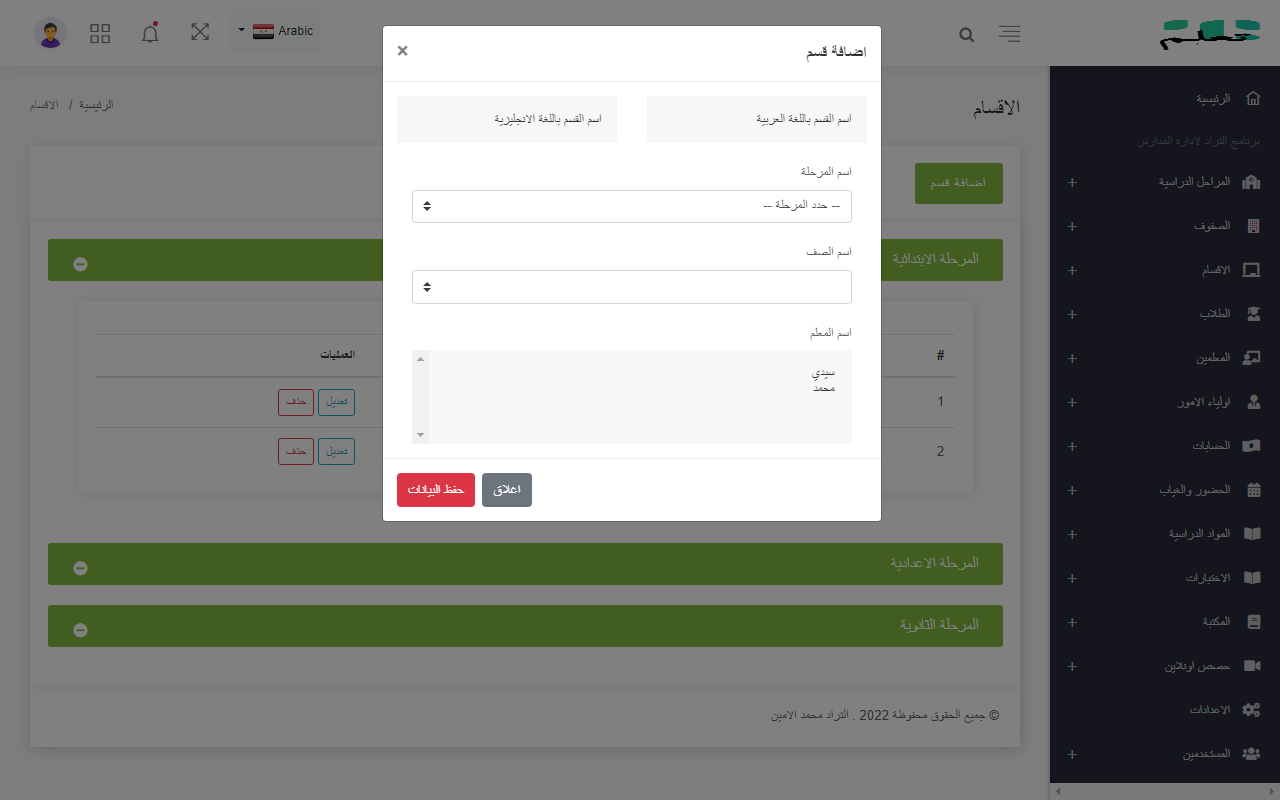


* + 1. les sous-classe

4.6.4.1 liste des sous-classe

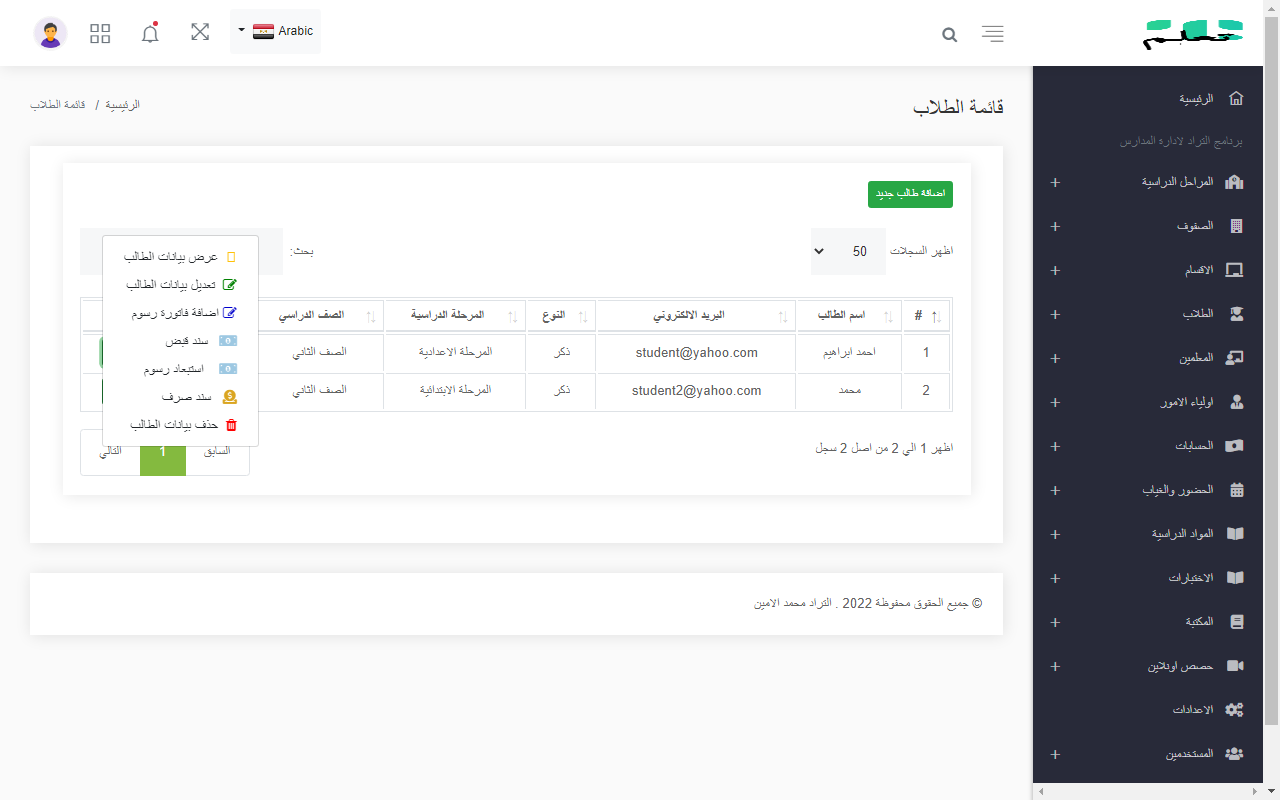


4.6.4.2 ajouter des sous-classe

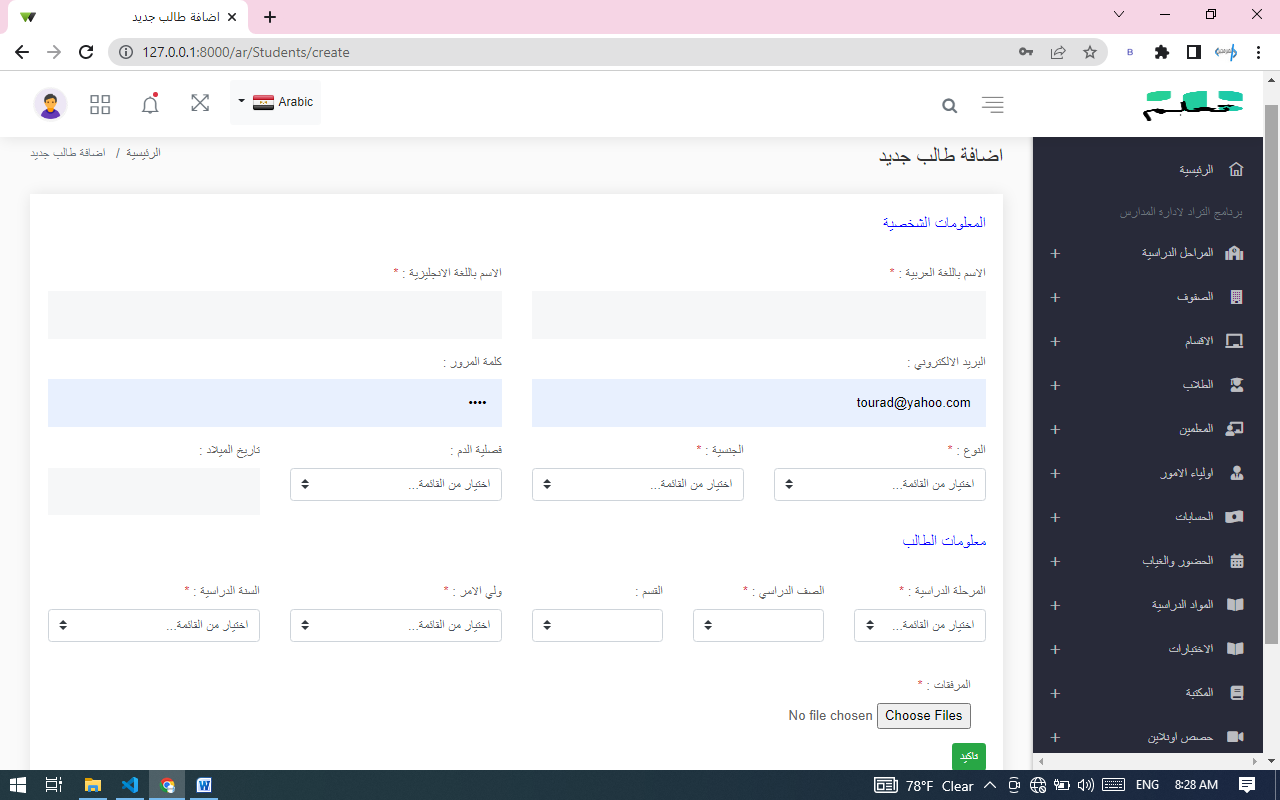


* + 1. les élèves

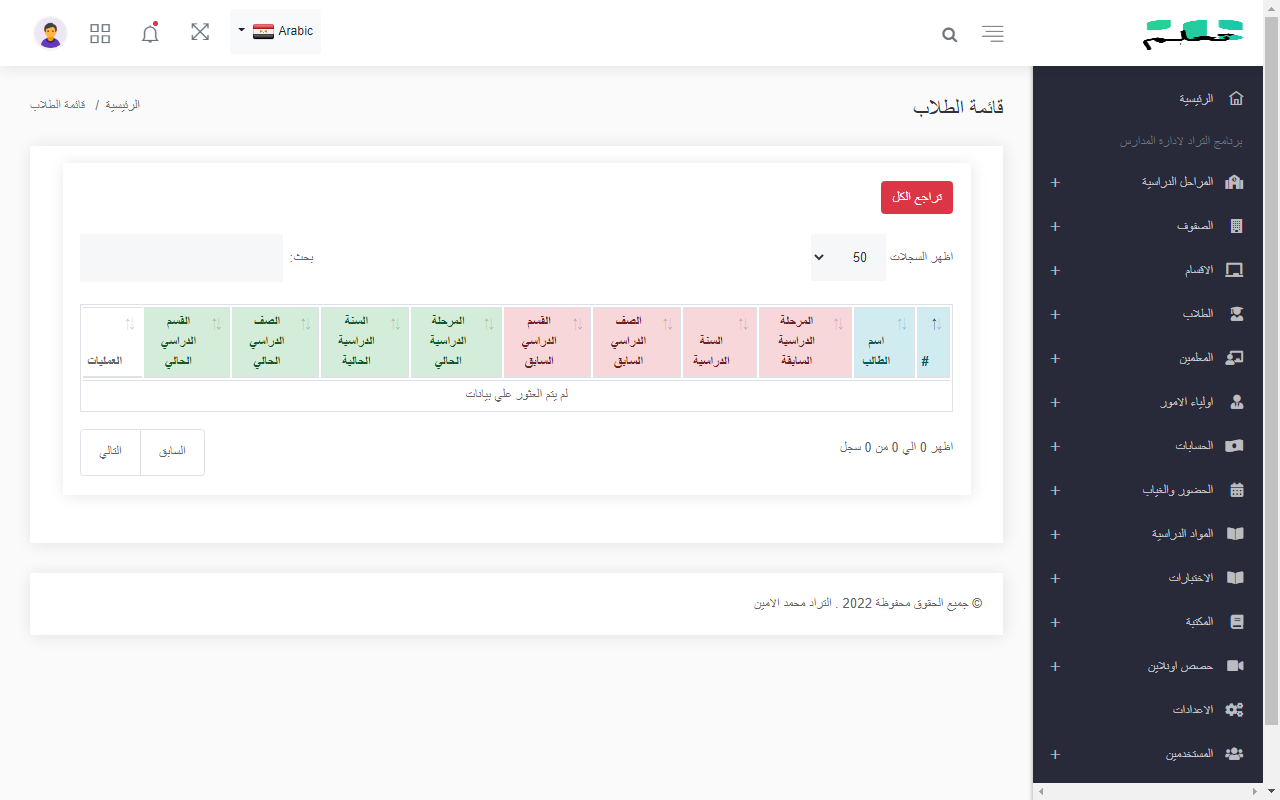
4.6.5.1 liste des élèves



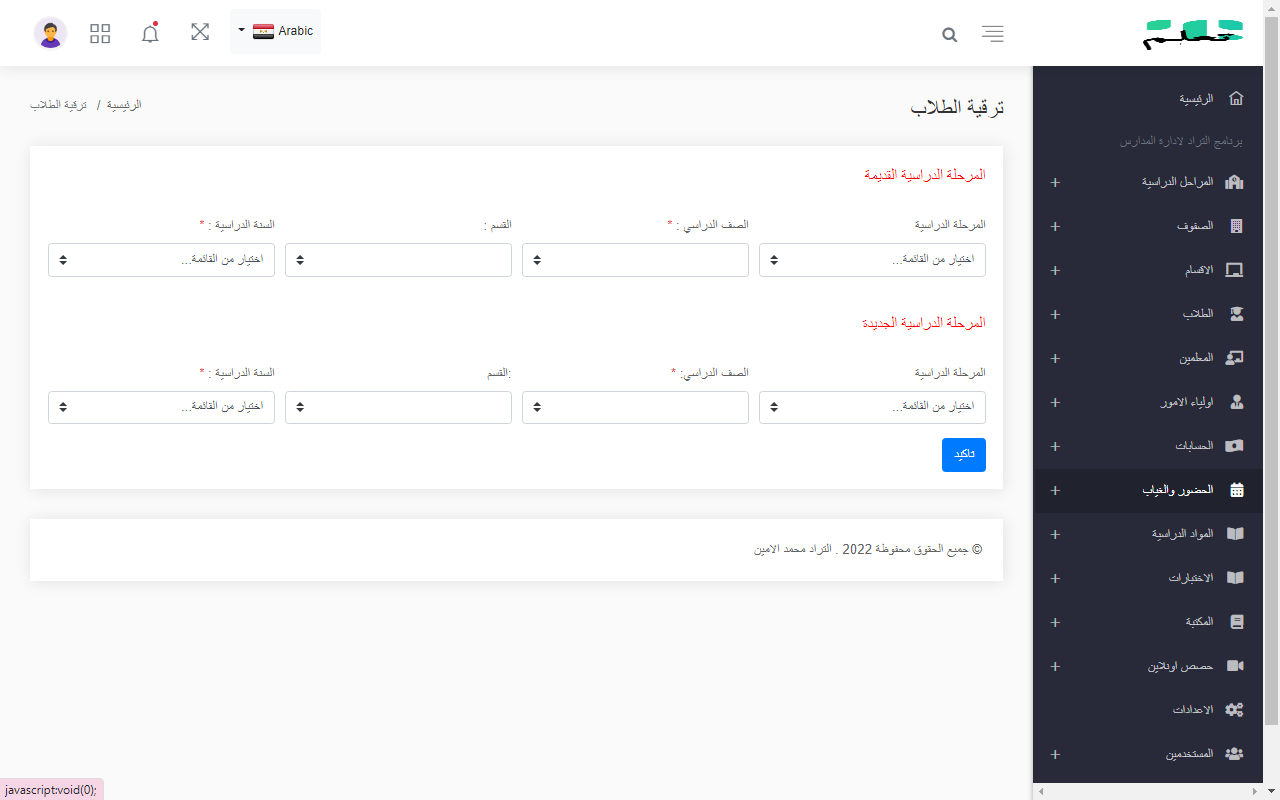
4.6.5.2 ajouter un élève



* + - 1. liste des promotions

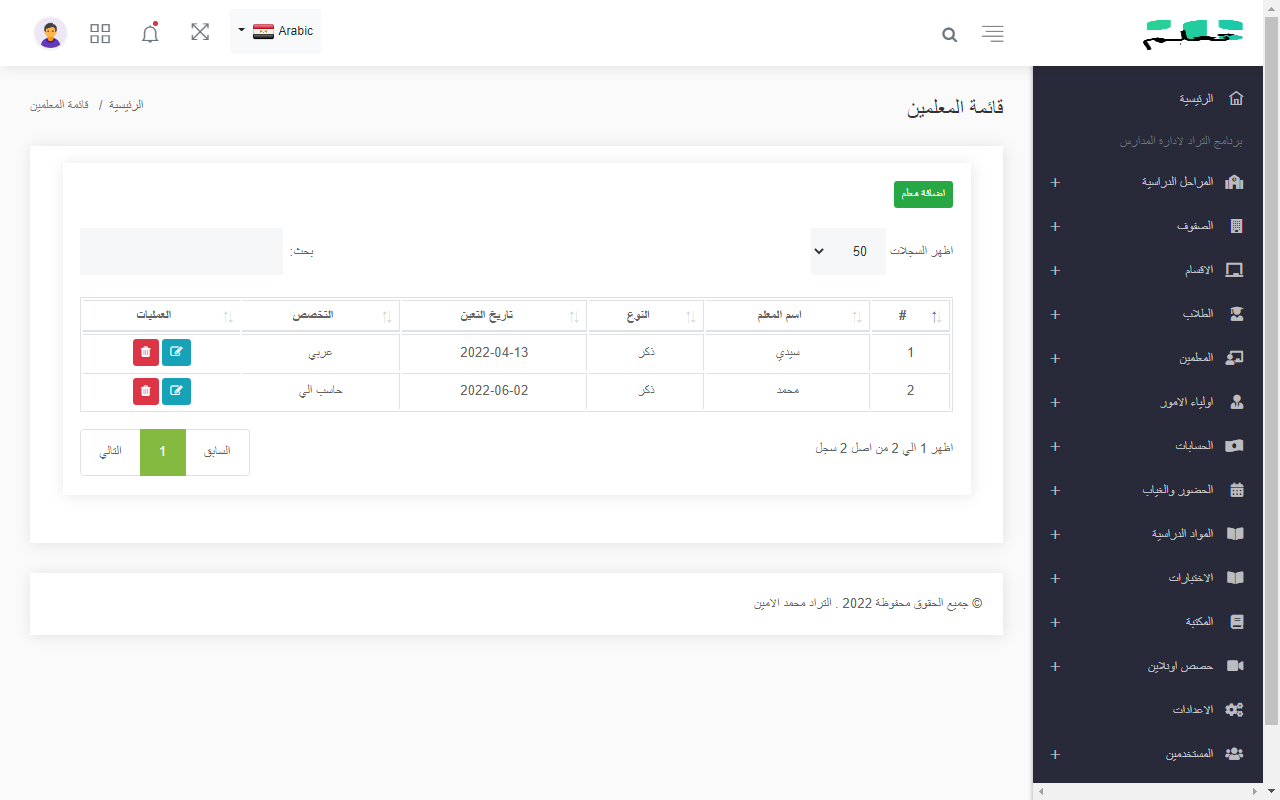


* + - 1. ajouter un promotion

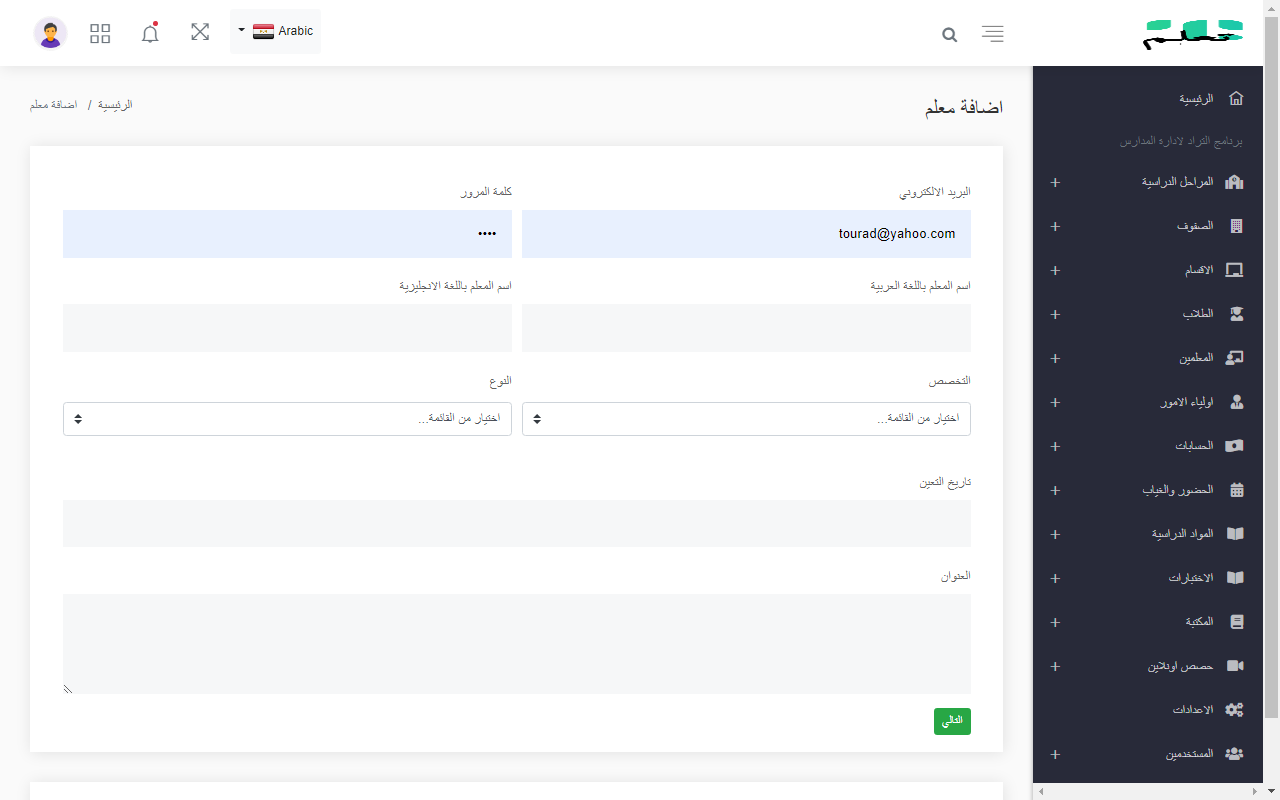


* + 1. enseignants

4.6.6.1 liste des enseignants

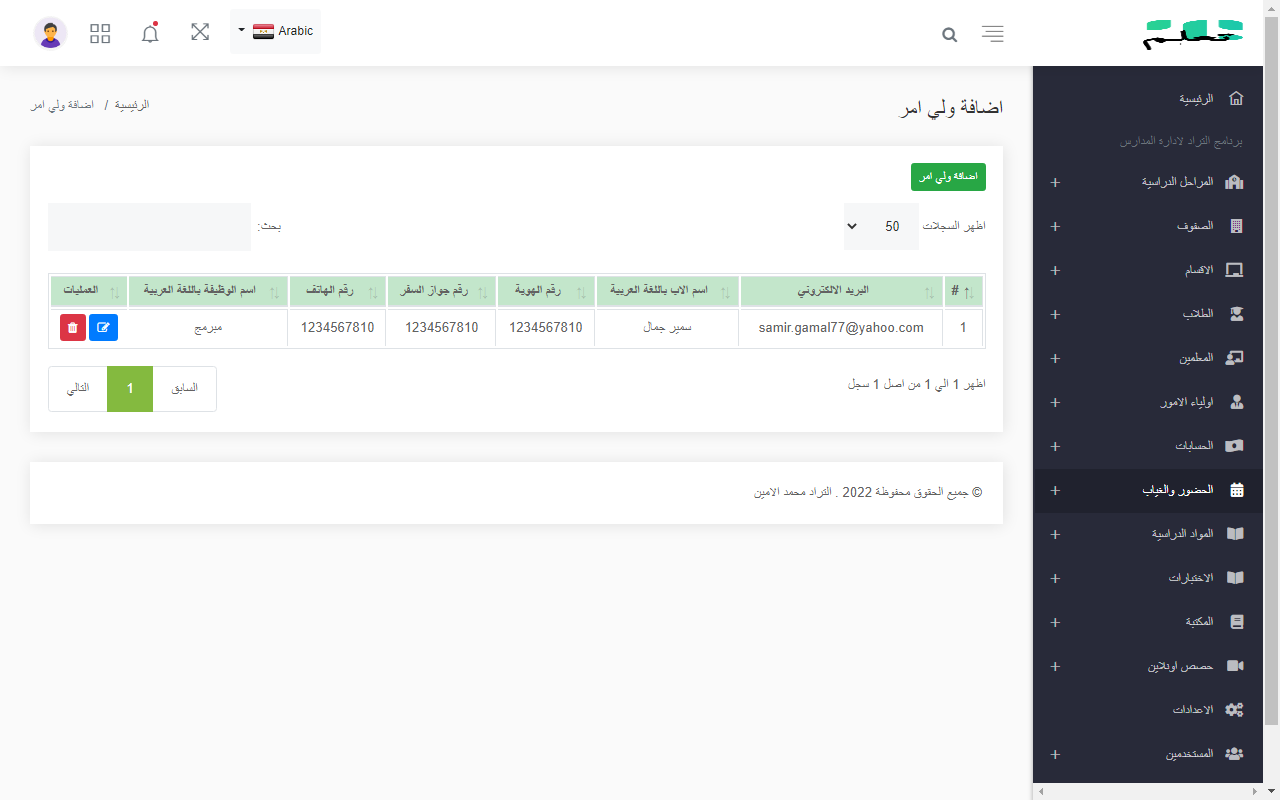


4.6.6.2 ajouter un enseignant

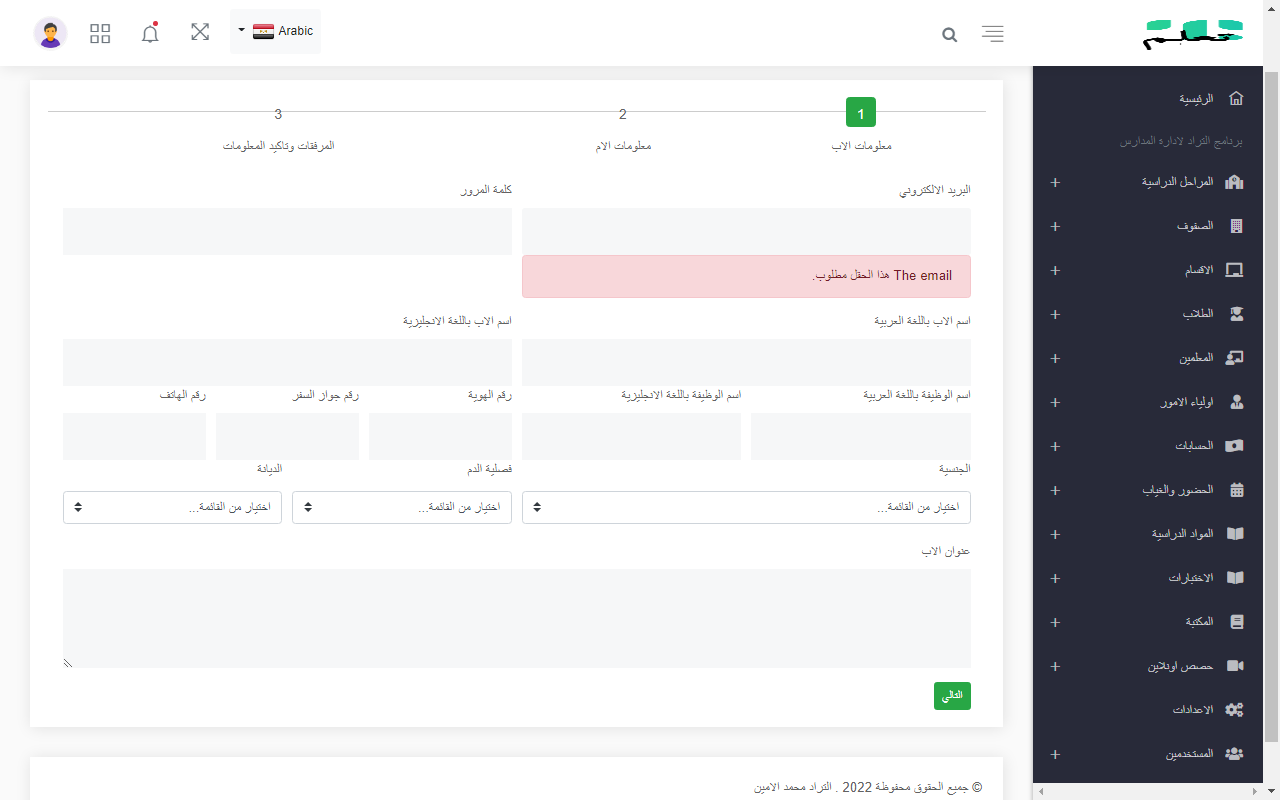


* + 1. les parents

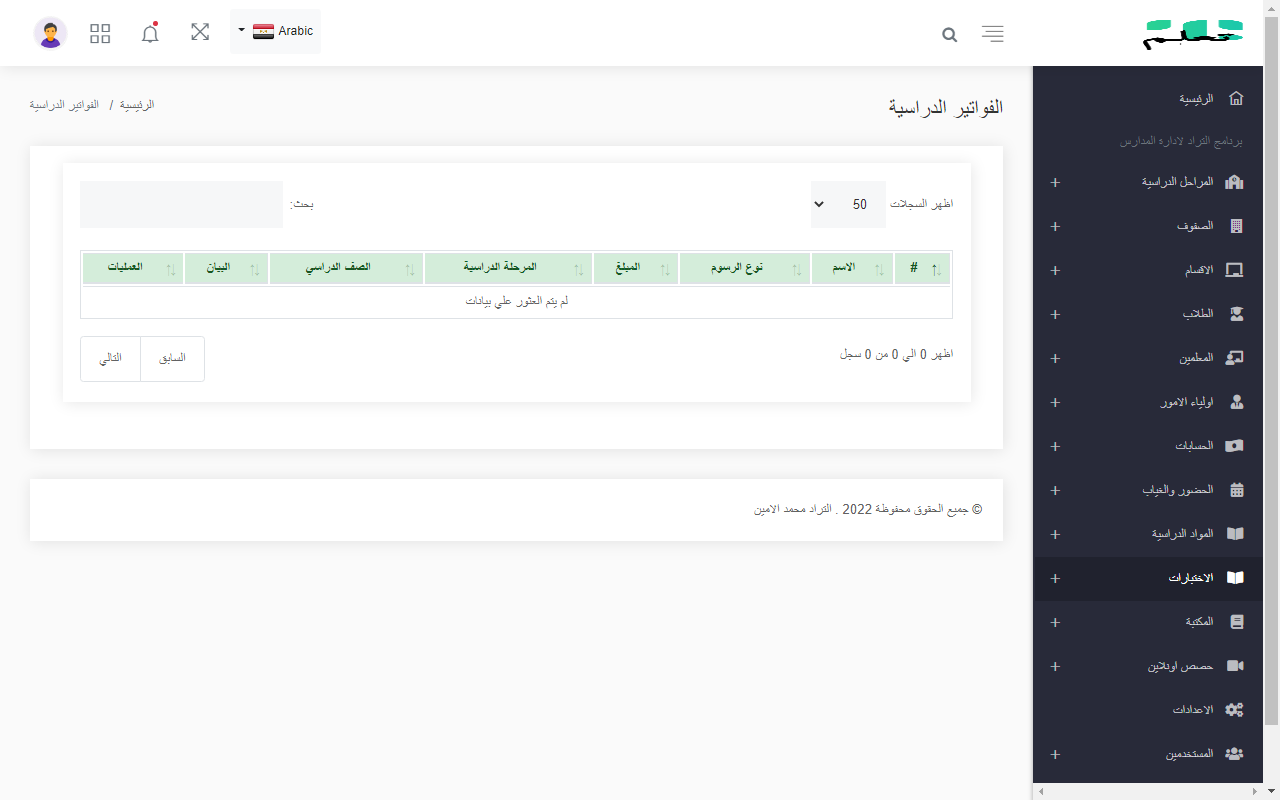
4.6.7.1 liste des parents



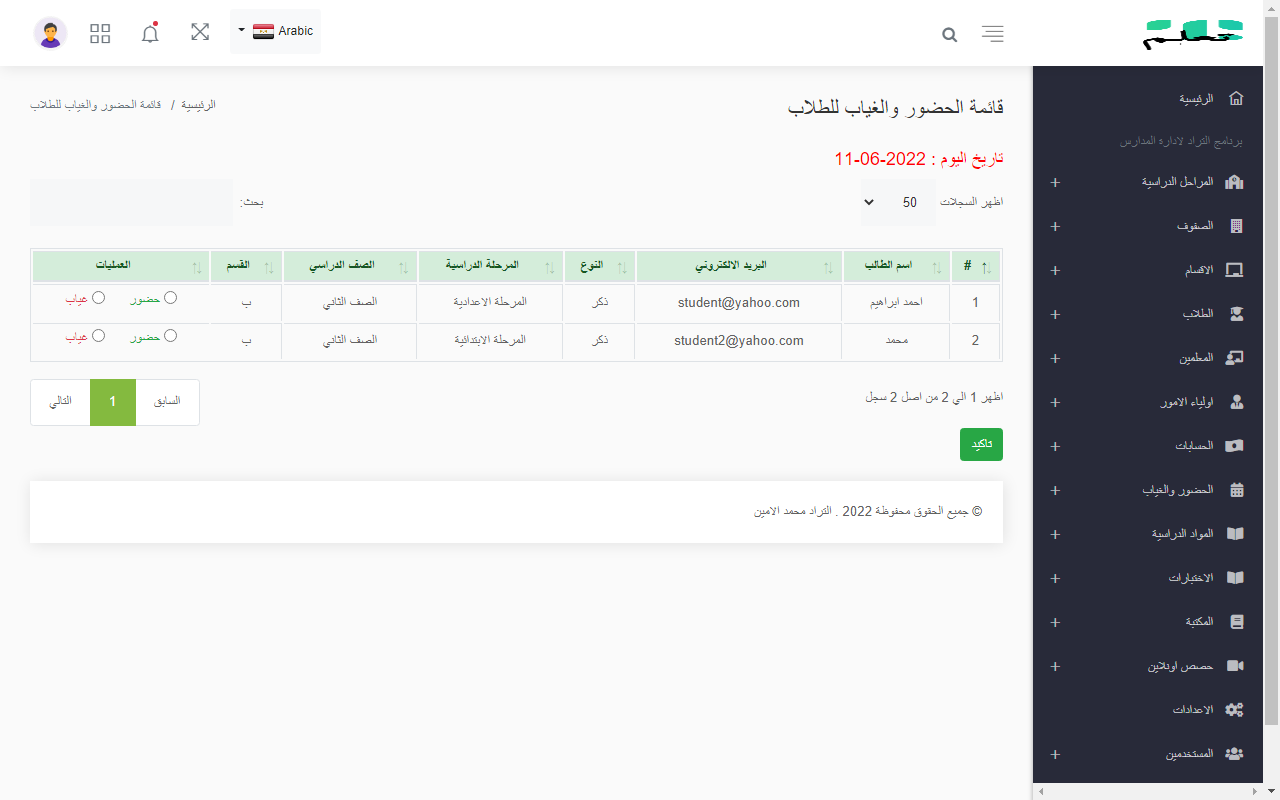
4.6.7.2 ajouter un parent



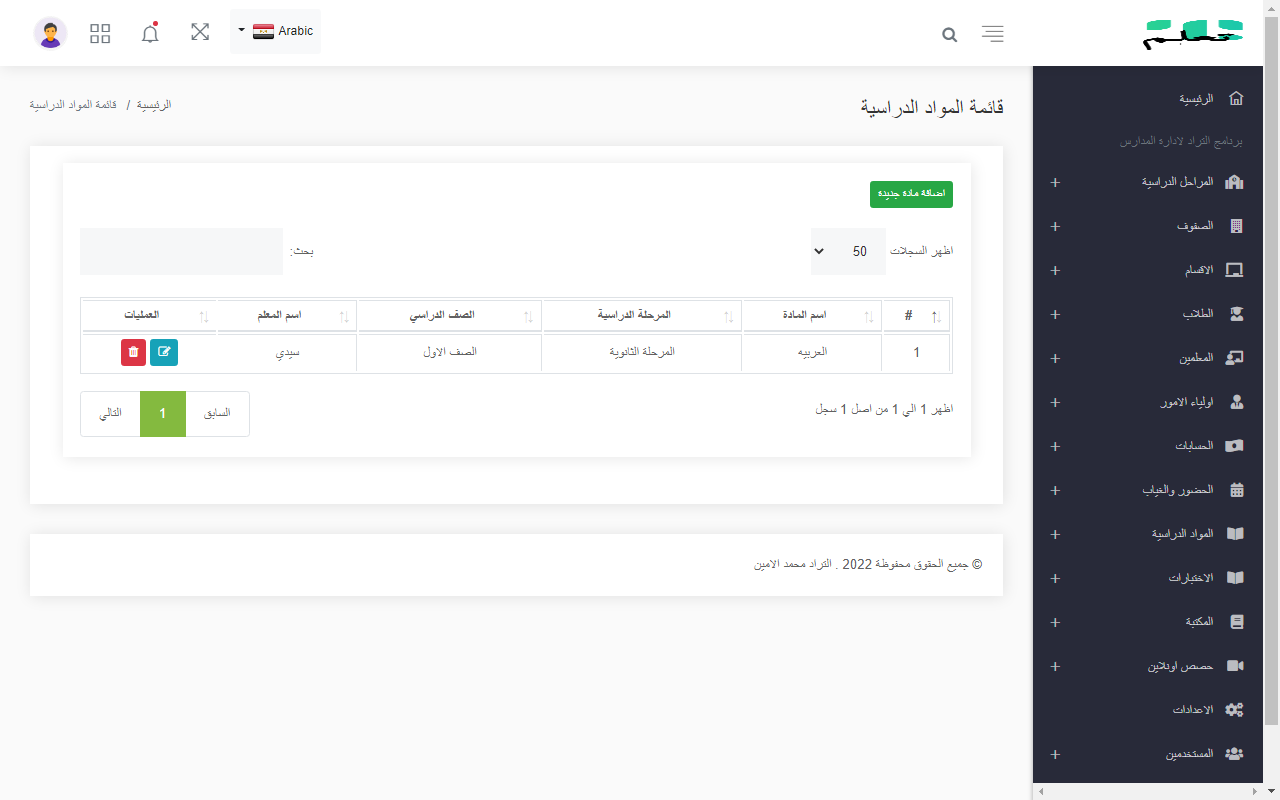
### les factures



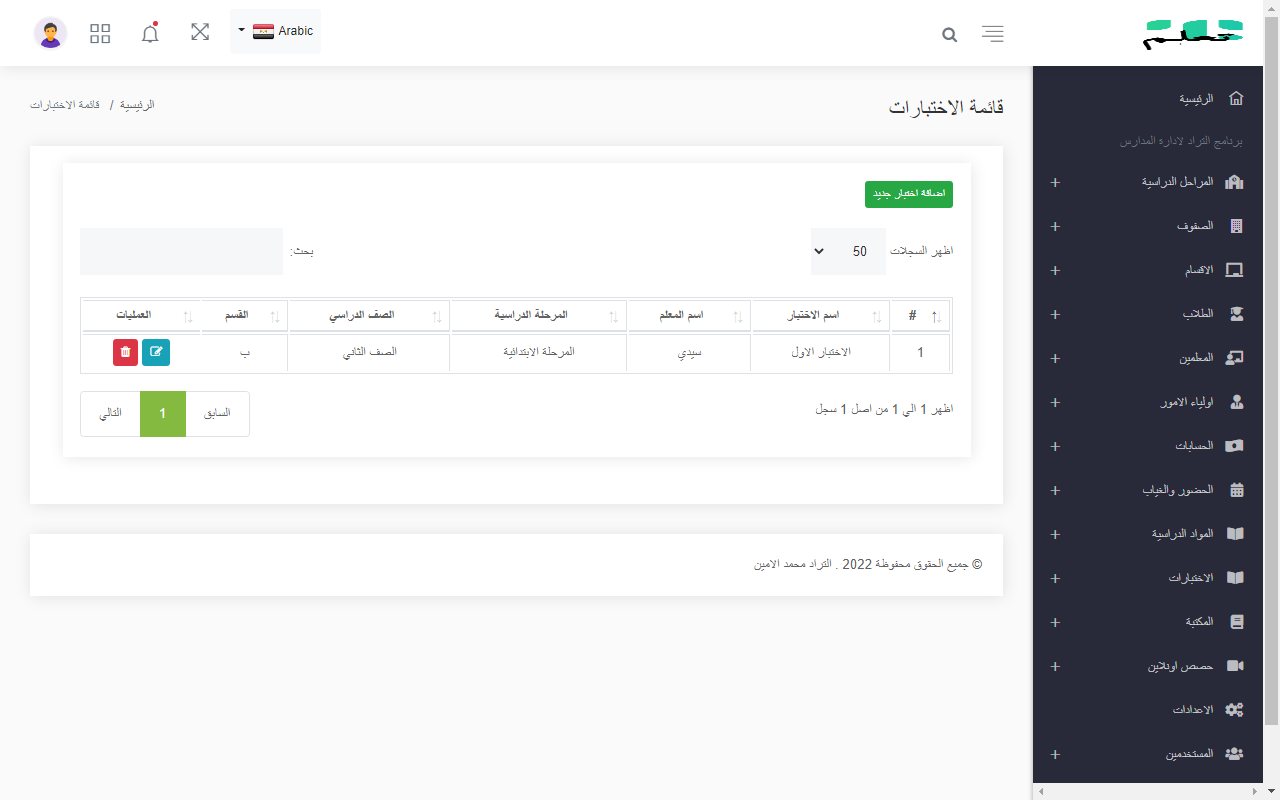
* + 1. la présence et l’absence



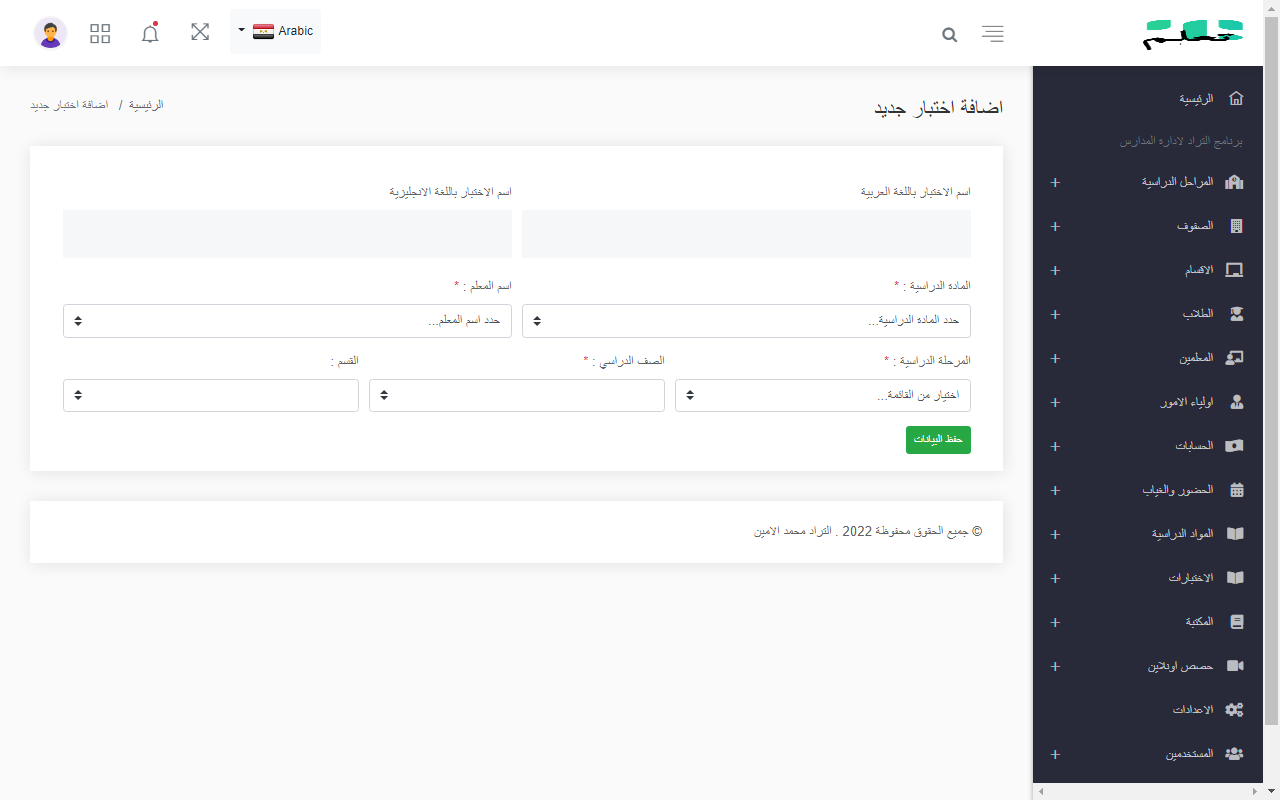
* + 1. les matiere



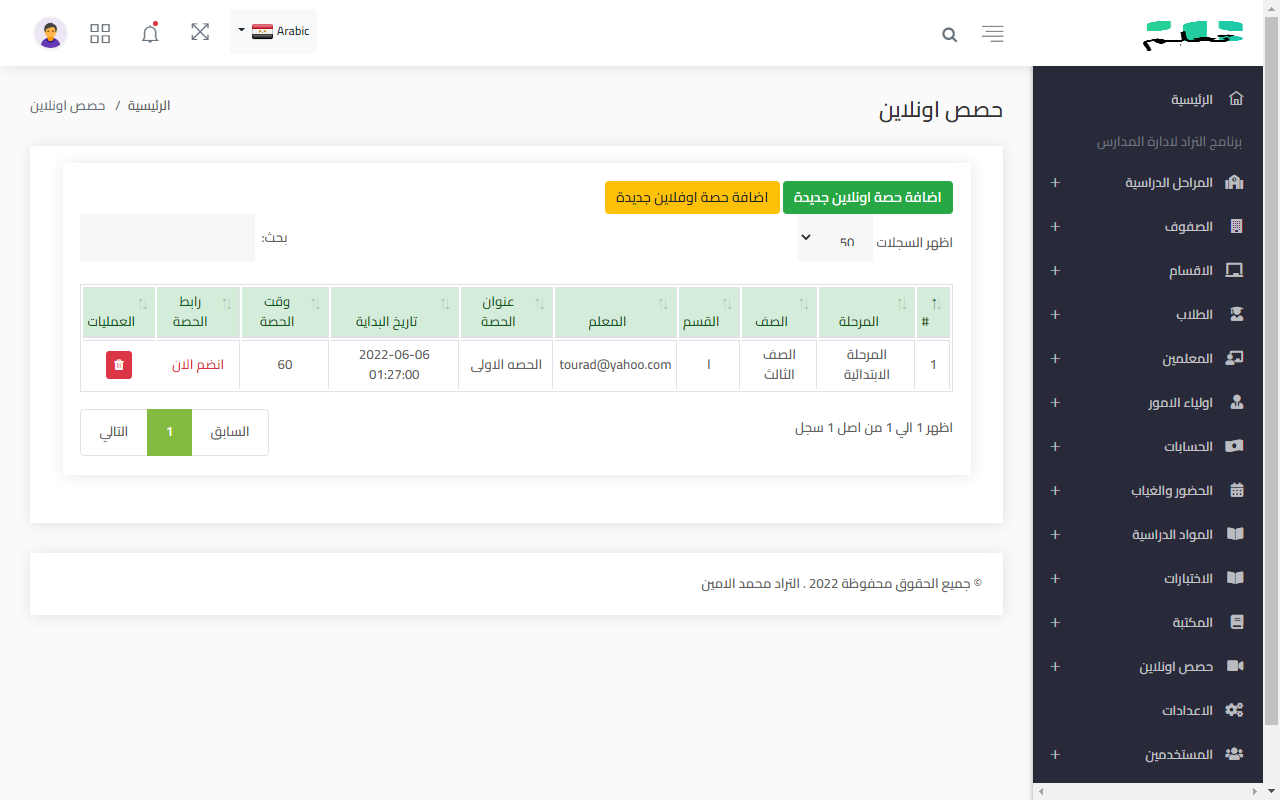
* + 1. *liste des examens*

**

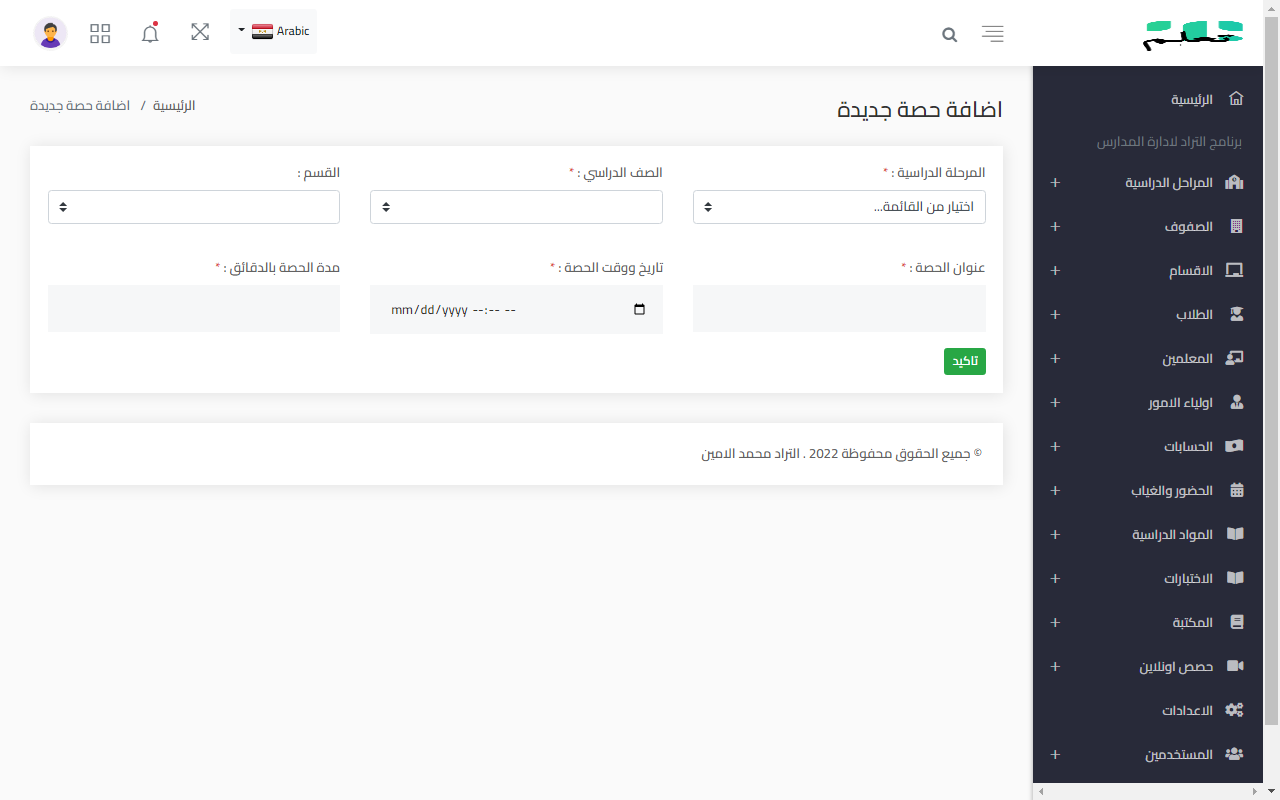
* + 1. ajouter un examen

**

### 4.6.13 liste des cours en ligne



### ajouter des cours en ligne

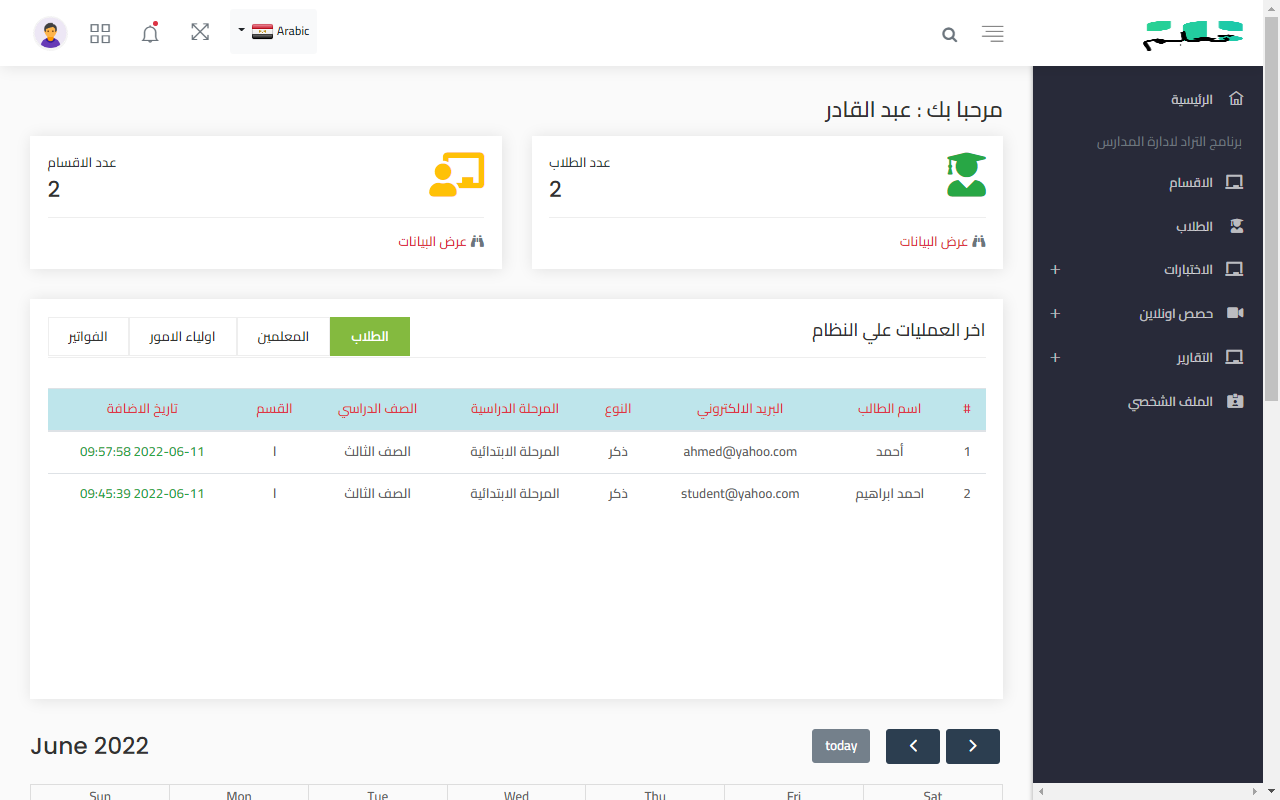


* 1. interface l’enseigant

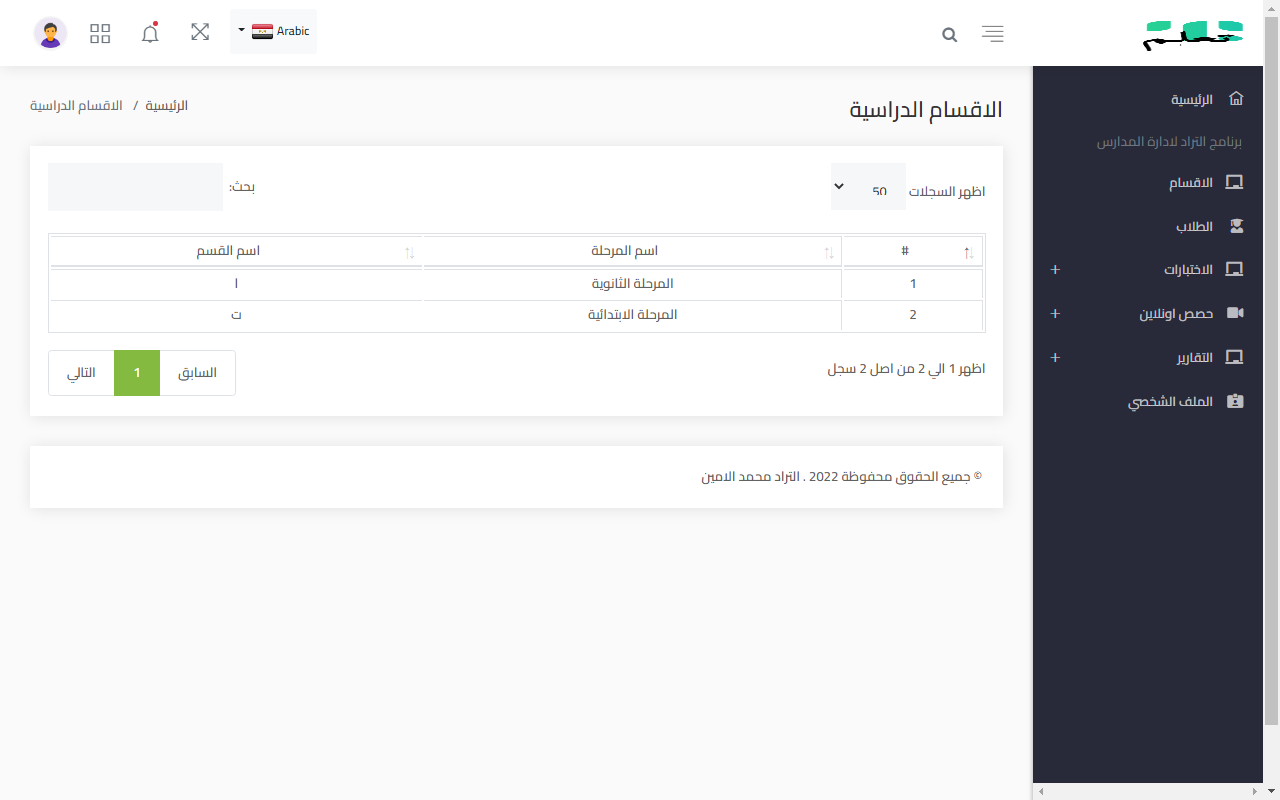
4.7.1 login



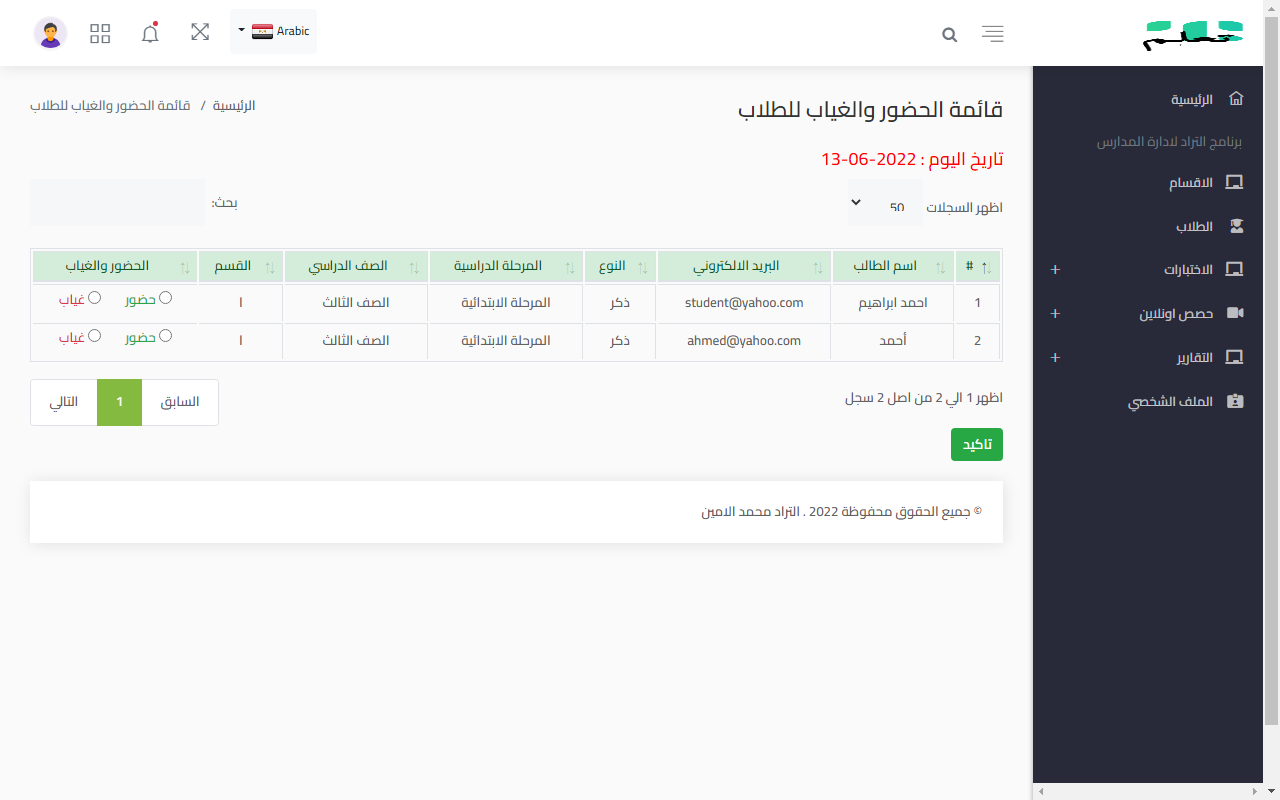
* + 1. Acceil



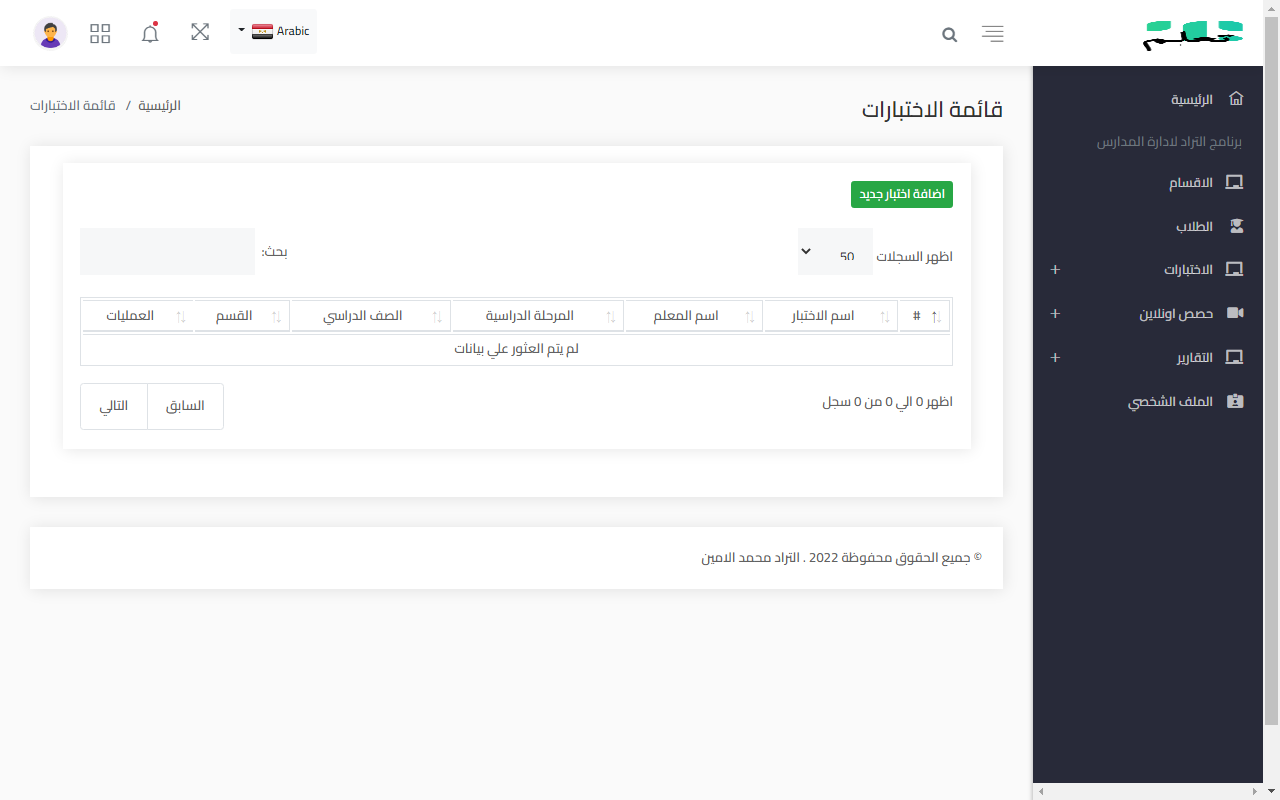
* + - liste des classes



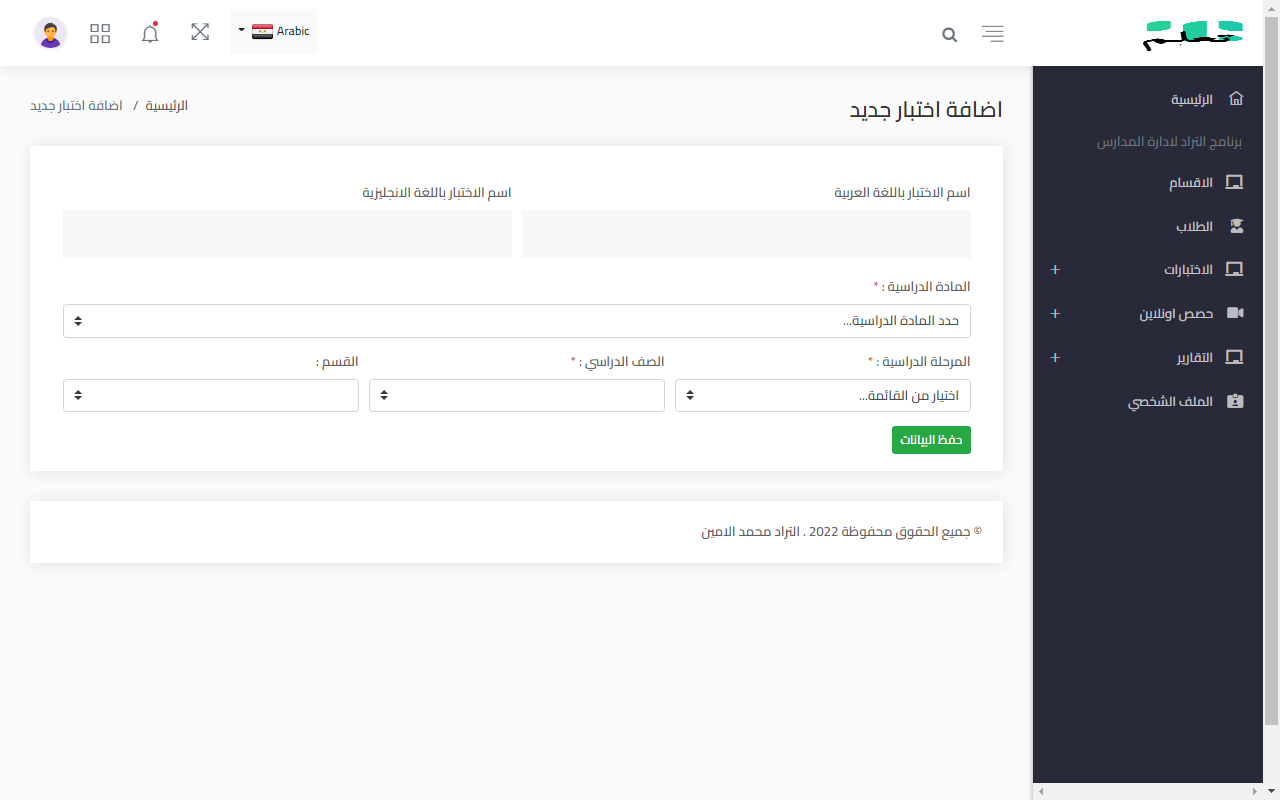
* + 1. la presence et l’absence



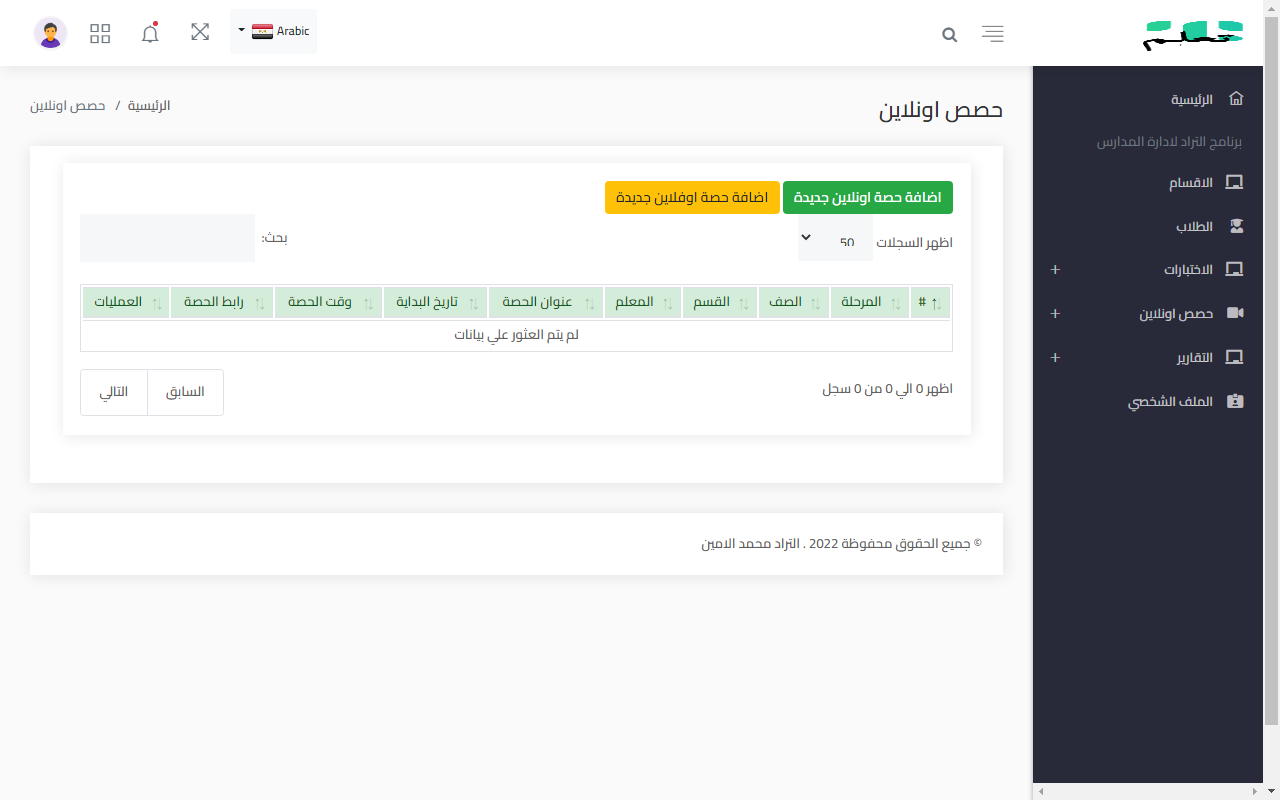
* + 1. liste des examens



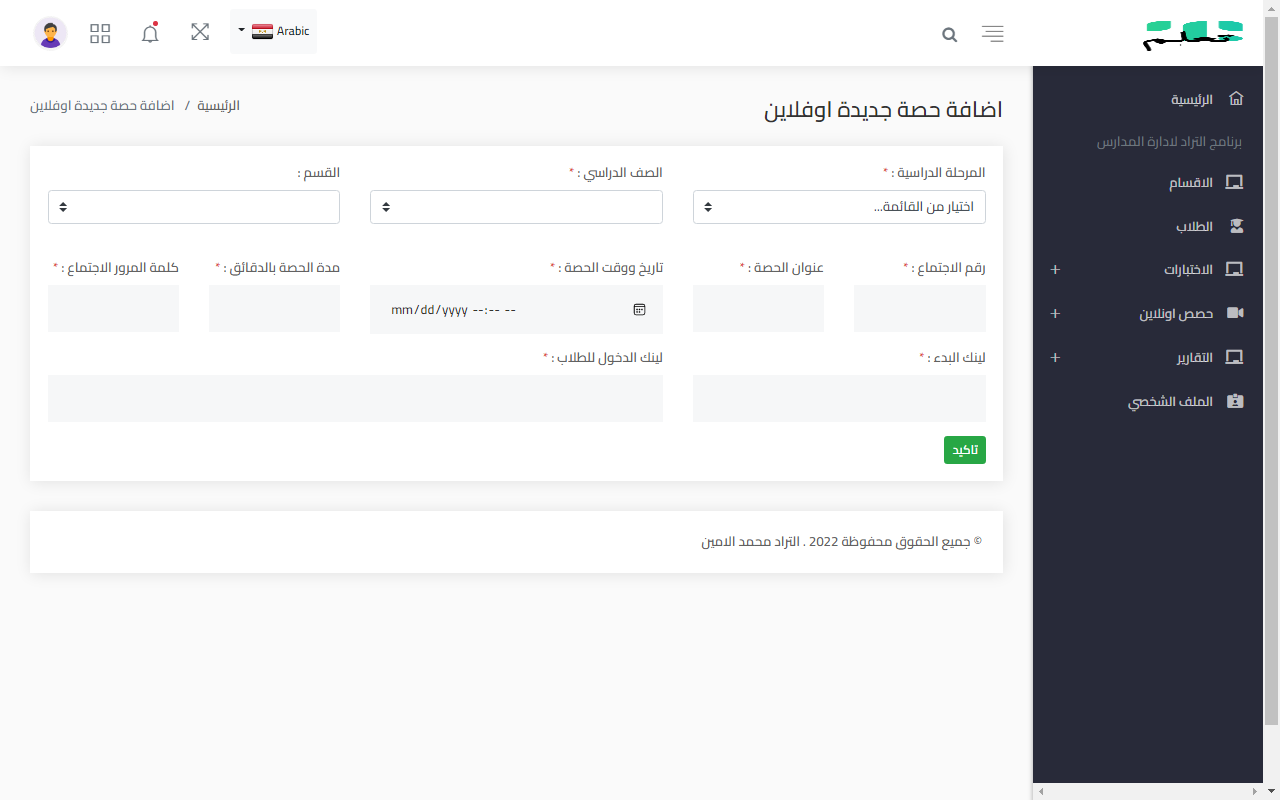
* + 1. ajouter un examen



* + 1. liste des cour en ligne



* + 1. ajouter un cour en ligne



**Conclusion générale**

Ce projet nous a permis de mettre en pratique notre esprit d’étude, d’analyse et de critique, de mettre en application certaines de nos connaissances et notre savoir acquis lors de la période de la formation a ISCAE et de découvrir la différence entre les projets professionnels et ceux à caractère pédagogique. L’analyse des besoins, la conception, et la réalisation du système de gestion du parc informatique, ont été les phases du cycle de développement de notre projet. Les difficultés majeures que nous avons rencontrées durant ce projet résident essentiellement dans la nouveauté du Framework LARAVEL. L’application réalisée répond à la plupart des besoins actuels qui ont été formulé par les différents citoyens et employées de l’état Civil qu’on a rencontrés. Par ailleurs, l’application est encore améliorable, on pourrait lui ajouter des fonctionnalités supplémentaires pour qu’il puisse remplace le system d’état civil elle-même.

**Bibliographie**

<https://laravel.fr>

<https://drow.io>

<https://www.valuecoders.com>

[fr.wikipedia.org](https://fr.wikipedia.org/wiki/Wikip%C3%A9dia)