

RAPPORT GESTION PROJET

APPLICATION WEB GESTION
D'ABSENCES DES ÉTUDIANTS

Prepared by: Sheila Favreau
Social Media Team 1 Manager
Date: 07/12/2024

1.INTRODUCTION

Ce projet vise à développer une application web pour gérer efficacement les absences des étudiants. L'application est conçue pour simplifier les tâches administratives liées au suivi et au rapport des absences, en s'adressant aux administrateurs, chefs de département et enseignants. Elle garantit une gestion précise des données, facilite la création de groupes et soutient la prise de décisions éclairées grâce à des rapports détaillés.

1.CONTENU PRINCIPAL DU RAPPORT

2.1 Analyse du Cahier des Charges

L'application comporte deux modules principaux :

- 1.Administration : Permet la création et la gestion des départements, des modules, des éléments de modules, ainsi que l'affectation des enseignants. Elle permet également d'importer les données des étudiants depuis des fichiers Excel et de générer des rapports d'absences semestriels.
- 2.Utilisation : Conçue pour les chefs de département et les enseignants, elle permet de gérer les groupes d'étudiants (TD/TP) et d'enregistrer les absences. Les fonctionnalités incluent la modification des types d'absences et la génération de résumés mensuels.

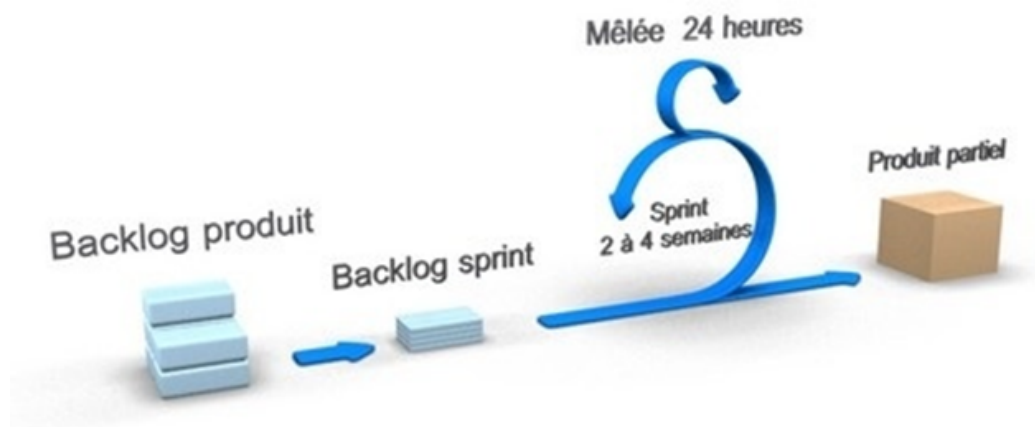
2.2 Gestion de Projet avec les Méthodes Agiles Le projet suit une méthodologie Agile itérative et collaborative :

- Initialisation : Définition de la vision du produit et identification des parties prenantes.
- Planification : Création du product backlog et définition des sprints.
- Exécution : Développement incrémental avec des sprints courts et des

- Revue Continue : Rétrospectives d'équipe et ajustements permanents.
- Clôture : Validation finale du produit et capitalisation des apprentissages.

Caractéristiques clés de l'approche Agile :

- Développement itératif
- Accent sur la collaboration
- Planification flexible
- Amélioration continue
- Approche centrée sur le client



2.3 Décomposition du Projet (WBS et PBS)

• Structure de Découpage du Travail (WBS)

La Structure de Découpage du Travail (WBS) divise le projet en tâches gérables afin de faciliter la planification, l'exécution et le suivi. Chaque tâche est associée à des livrables spécifiques.

Niveau 1 : Objectif global

- Développement de l'application de gestion des absences des étudiants.

Niveau 2 : Phases principales

1. Analyse des besoins

- Collecte des exigences auprès des utilisateurs (administrateur, chef de département, enseignants).
- Documentation des spécifications fonctionnelles et techniques.

2. Conception UI/UX

- Conception des interfaces utilisateur avec Figma.
- Validation des maquettes par les parties prenantes.

3. Développement back-end

- Mise en place de la logique métier avec PHP.
- Création des API pour gérer les données des étudiants, des absences et des rapports.

4. Développement front-end

- Développement des interfaces avec HTML, CSS et JavaScript.
- Intégration des fonctionnalités dynamiques (formulaires, tableaux interactifs).

5. Conception de la base de données

- Création du schéma relationnel dans MySQL.
- Implémentation des tables (étudiants, absences, modules, etc.).

6. Tests

- Tests fonctionnels pour vérifier les cas d'utilisation.
- Tests de performance pour évaluer la charge du système.

7. Déploiement et documentation

- Hébergement de l'application sur un serveur.
- Livraison des guides utilisateur et technique.

• Structure de Découpage des Produits (PBS)

La Structure de Découpage des Produits (PBS) identifie les livrables du projet, classés par catégories.

1. Application web

- Interfaces utilisateur.
- Modules fonctionnels (gestion des absences, importation des étudiants, rapports).

1. Base de données
 - Tables pour stocker les données des départements, étudiants, enseignants, absences.
 - Procédures SQL pour les importations et les rapports.
2. Documentation
 - Cahier des charges initial.
 - Rapport de projet.
 - Manuel utilisateur.
3. Rapports et backups
 - Rapports d'absences par filière, module, et étudiant.
 - Sauvegardes périodiques de la base de données.

2.4 Conception Logique

Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation illustre les interactions entre les utilisateurs (acteurs) et le système, mettant en évidence les fonctionnalités principales accessibles selon les rôles. Par exemple :

- Acteurs :
 - Administrateur
 - Enseignant
 - Chef de département
- Cas d'utilisation principaux :
 - Gérer les absences
 - Importer les données des étudiants
 - Générer des rapports

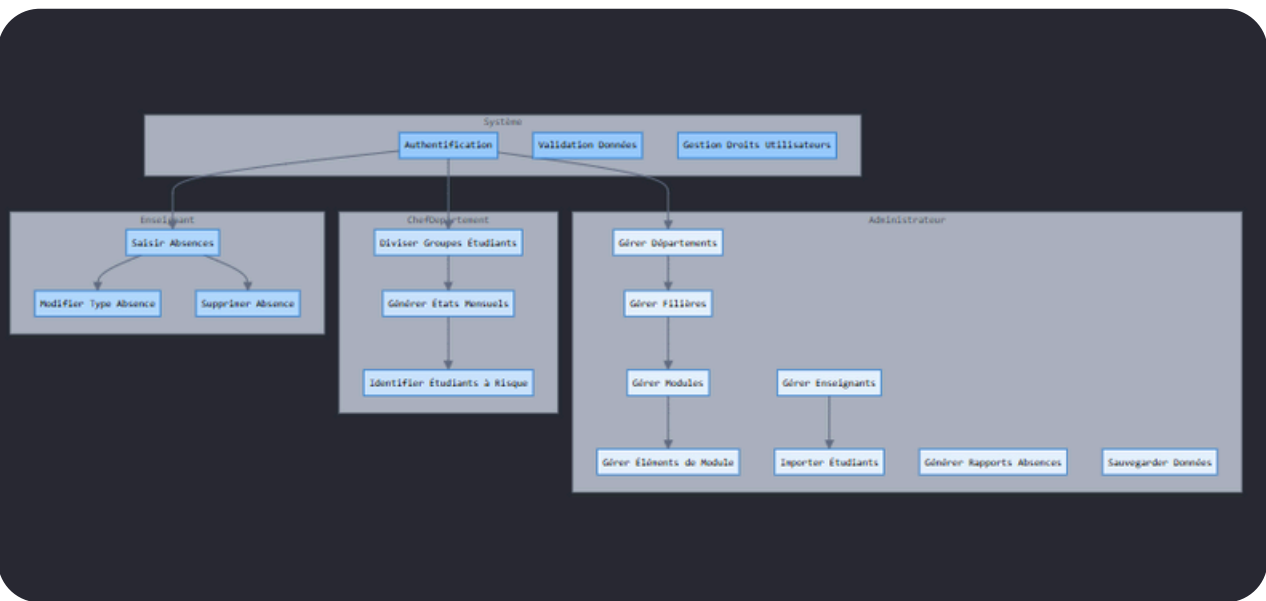


Diagramme de Cas d'Utilisation

Diagramme de Classes

Le diagramme de classes représente les entités principales du système, leurs attributs et leurs relations. Les classes clés incluent :

- Utilisateur : Attributs comme id, nom, rôle.
- Étudiant : Attributs comme id, nom, groupe.
- Absence : Attributs comme id, date, type, motif.
- Groupe : Attributs comme id, nom, module.



Diagramme de Classes

Diagramme de Séquence

Le diagramme de séquence décrit le flux d'interactions entre les différents acteurs et le système pour un scénario donné. Exemple : "Ajouter une absence" :

1. L'enseignant se connecte au système.
2. Il sélectionne un groupe d'étudiants.
3. Il saisit les informations d'absence (type, date, motif).
4. Le système enregistre l'absence dans la base de données.

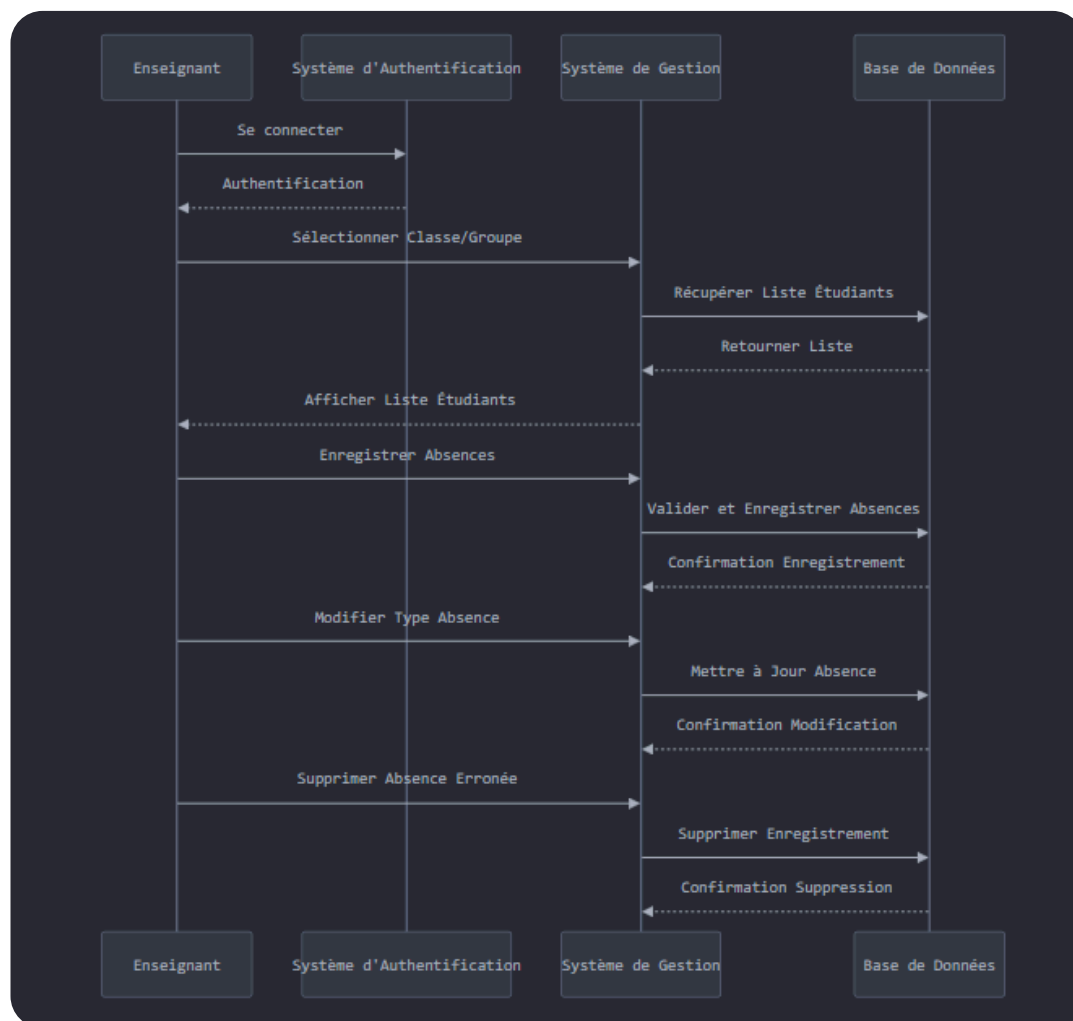


Diagramme de Classes











2.5 Conception UI avec Figma

Figma a été utilisé pour créer une interface intuitive et conviviale. Les écrans clés incluent :

- Tableau de bord administrateur
- Vues de gestion des groupes d'étudiants Les écrans priorisent l'accessibilité et la réactivité.

All Teachers List

Search by ID, Name, or Subject

<input type="checkbox"/>	Teacher Name	School ID	Subject	Class(es)	Phone Number	Address
<input type="checkbox"/>	 Ms. Elizabeth Johnson johnson@eduprohigh.edu	SJ-ENG-123	English Literature	9A, 10B	(555) 101-0101	123 Elm St, Springfield, IL
<input checked="" type="checkbox"/>	 Mr. Carlos Garcia garcia@eduprohigh.edu	GRC-HIS-456	History	8C, 11A	(555) 101-0102	456 Oak Ave, Springfield, IL
<input type="checkbox"/>	 Ms. Angela Jackson jackson@eduprohigh.edu	JCK-MATH-789	Math	7A, 12 AP Calculus	(555) 101-0103	789 Pine Rd, Springfield, IL
<input type="checkbox"/>	 Mr. Luis Rodrige rodrige@eduprohigh.edu	ROD-DRA-012	Drama	Drama Club	(555) 101-0104	012 Maple Dr, Springfield, IL
<input type="checkbox"/>	 Ms. Susan Chen chen@eduprohigh.edu	CHN-SCI-345	Science	8B, 9B Biology	(555) 101-0105	345 Birch Ln, Springfield, IL
<input type="checkbox"/>	 Mr. William Blake blake@eduprohigh.edu	THM-HIS-678	History	10A, 11 AP World History	(555) 101-0106	678 Cedar Blvd, Springfield, IL
<input type="checkbox"/>	 Ms. Emily Lee lee@eduprohigh.edu	LEE-ENG-901	English	7B, 8A	(555) 101-0107	901 Walnut St, Springfield, IL
<input type="checkbox"/>	 Mr. Fernando Davis davis@eduprohigh.edu	HER-SPA-234	Spanish	Spanish I, Spanish II	(555) 101-0108	234 Spruce Way, Springfield, IL
<input type="checkbox"/>	 Ms. Laura Lopez lopez@eduprohigh.edu	LOP-MATH-567	Math	7C, Pre-Algebra	(555) 101-0109	567 Redwood Ct, Springfield, IL
<input type="checkbox"/>	 Mr. Daniel Kim kim@eduprohigh.edu	KIM-ART-890	Art	Art I, Art II	(555) 101-0110	890 Fir St, Springfield, IL

<

1 2 3 4 ... 17

emailaddress@mail.com

+82 1234 5678

Terms of Use Privacy Policy

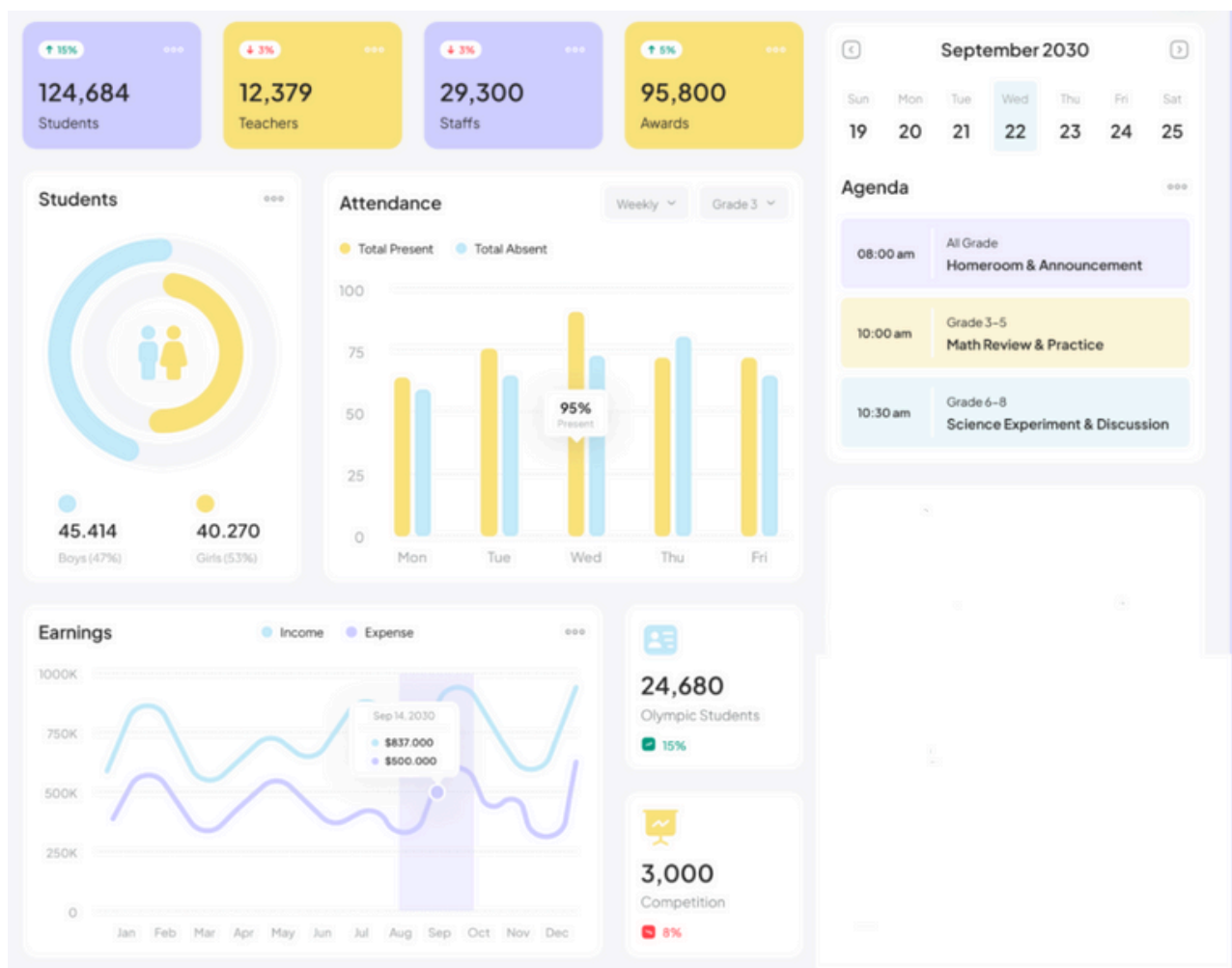


Tableau de bord administrateur

2.6 Conception de la Base de Données avec MySQL

FLe schéma de la base de données comprend des tables pour :

- Départements : Stocke les détails des départements.
- Modules et Éléments : Définit les modules liés aux départements.
- Étudiants : Gère les détails des étudiants importés depuis Excel.
- Absences : Suit les enregistrements des absences.

-- Schéma de la base de données

-- Table : Départements

```
CREATE TABLE Departements (  
    id_departement INT  
    AUTO_INCREMENT PRIMARY KEY,  
    nom_departement VARCHAR(100) NOT NULL,  
    description TEXT  
);
```

-- Table : Modules

```
CREATE TABLE Modules (  
    id_module INT  
    AUTO_INCREMENT PRIMARY KEY,  
    nom_module VARCHAR(100) NOT  
    NULL,  
    id_departement INT NOT NULL,  
    FOREIGN KEY (id_departement)  
    REFERENCES  
    Departements(id_departement) ON  
    DELETE CASCADE  
);
```

-- Table : Elements

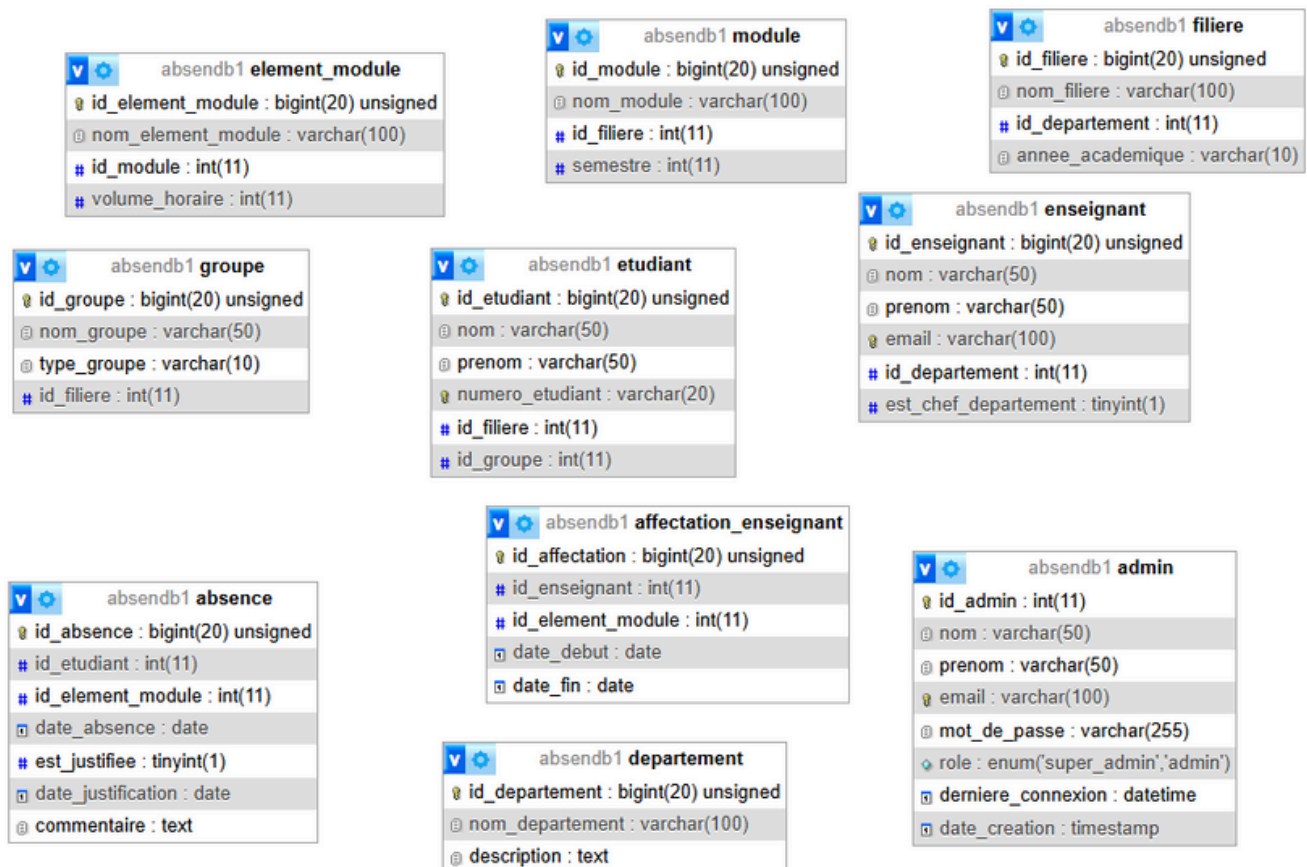
```
CREATE TABLE Elements (  
    id_element INT  
    AUTO_INCREMENT PRIMARY KEY,  
    nom_element VARCHAR(100)  
    NOT NULL,  
    id_module INT NOT NULL,  
    FOREIGN KEY (id_module)  
    REFERENCES Modules(id_module)  
    ON DELETE CASCADE  
);
```

-- Table : Étudiants

```
CREATE TABLE Etudiants (  
    id_etudiant INT  
    AUTO_INCREMENT PRIMARY KEY,  
    nom VARCHAR(100) NOT NULL,  
    prenom VARCHAR(100) NOT  
    NULL,  
    email VARCHAR(150) UNIQUE,  
    numero_etudiant VARCHAR(50)  
    UNIQUE NOT NULL  
);
```

-- Table : Absences

```
CREATE TABLE Absences (  
    id_absence INT  
    AUTO_INCREMENT PRIMARY KEY,  
    id_etudiant INT NOT NULL,  
    id_element INT NOT NULL,  
    date_absence DATE NOT NULL,  
    justification TEXT,  
    FOREIGN KEY (id_etudiant)  
    REFERENCES Etudiants(id_etudiant)  
    ON DELETE CASCADE,  
    FOREIGN KEY (id_element)  
    REFERENCES Elements(id_element)  
    ON DELETE CASCADE  
);
```



-- Schéma de la base de données

2.7 Développement (PHP)

Gestion des Départements :

Cette fonctionnalité permet de gérer les départements dans le système à travers des opérations CRUD (Créer, Lire, Mettre à jour, Supprimer). Elle permet aux administrateurs de créer de nouveaux départements, de consulter la liste des départements existants, de mettre à jour les informations des départements et de supprimer ceux qui ne sont plus nécessaires. Cette gestion facilite l'organisation et la structuration de l'entreprise.

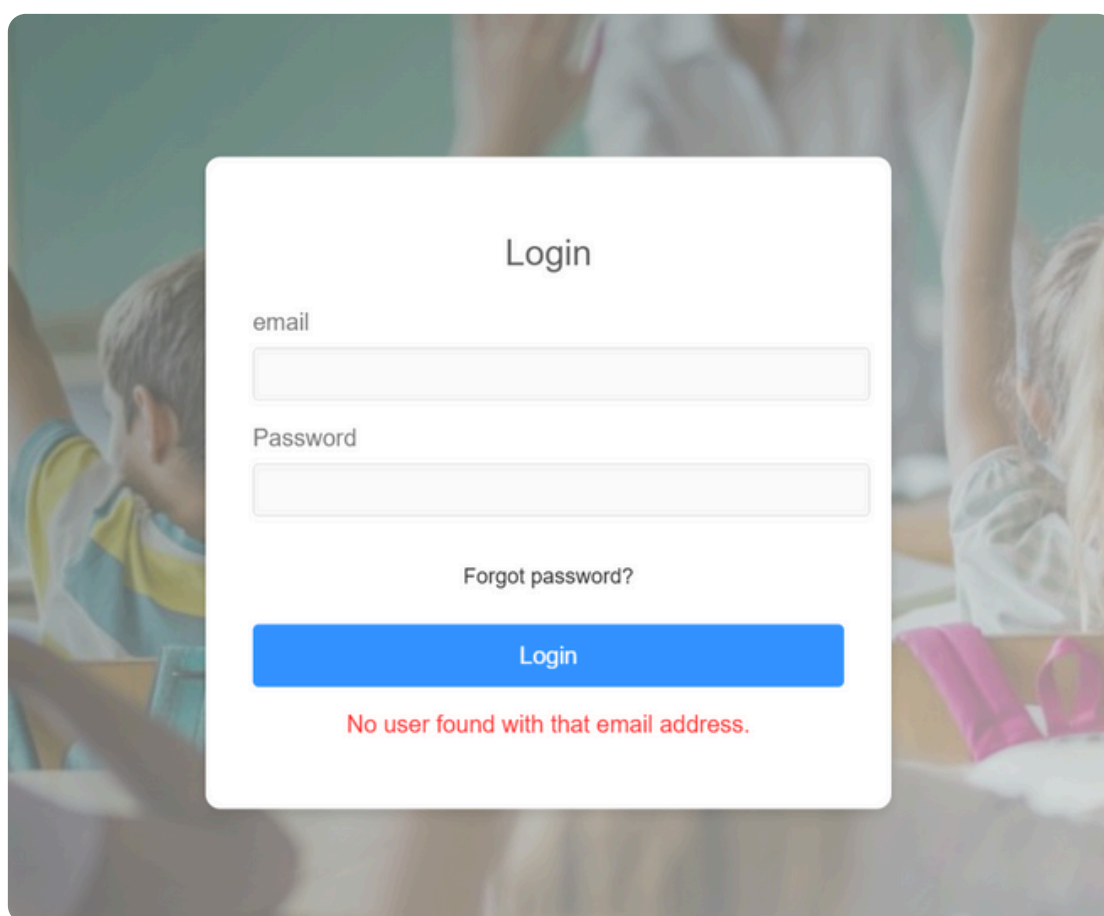
Gestion des Utilisateurs :

La gestion des utilisateurs permet aux administrateurs de gérer les comptes utilisateurs, de les créer, de consulter leurs informations, de les mettre à jour et de les supprimer. Les utilisateurs peuvent être affectés à des rôles spécifiques (par exemple, administrateur, employé, etc.) en fonction de leurs responsabilités au sein de l'organisation. Cette fonctionnalité assure une gestion sécurisée et structurée des accès et des droits au sein du système.

Connexion :

La fonctionnalité de connexion permet aux utilisateurs de se connecter à l'application de manière sécurisée. Elle inclut un mécanisme d'authentification avec un identifiant et un mot de passe, et peut inclure des options comme la réinitialisation de mot de passe et la gestion des sessions. Une fois connecté, l'utilisateur peut accéder aux fonctionnalités et données auxquelles il est autorisé en fonction de ses privilèges.

Les interface de l'application



Interface Login

Users						Create New User
ID	Name	Email	Role	Department	Actions	
1	tazi	tazi@gmail.com	2	Marketing 1	Edit	Delete
2	mohcine	mohcine@gmail.com			Edit	Delete
3	mohcine@gmail.com	ennefaimohssin@gmail.com	1	Production	Edit	Delete

Gesttionne les utilisateurs

Users

ID	Name
1	tazi
2	mohcine
3	mohcine@gmail.com

Create New User

Name

Email

Password

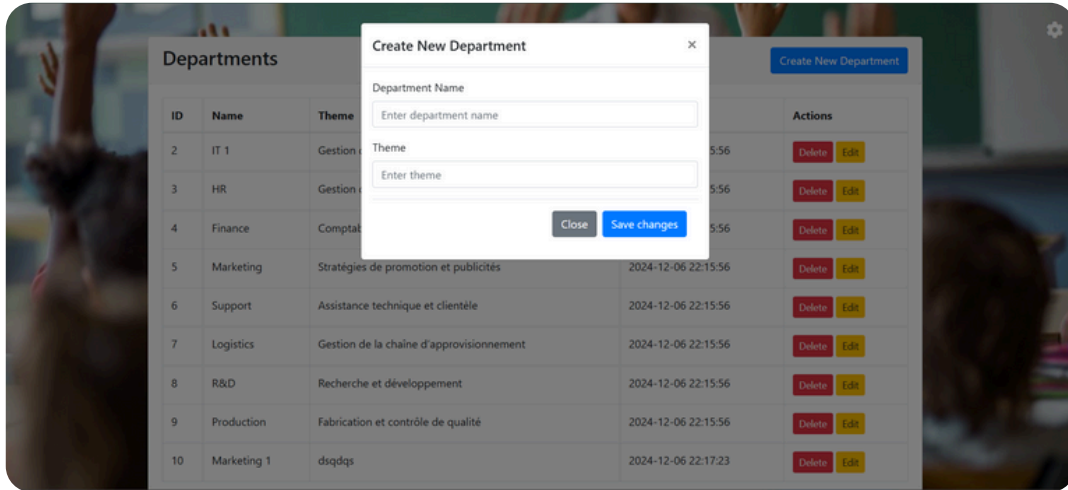
Role

Department

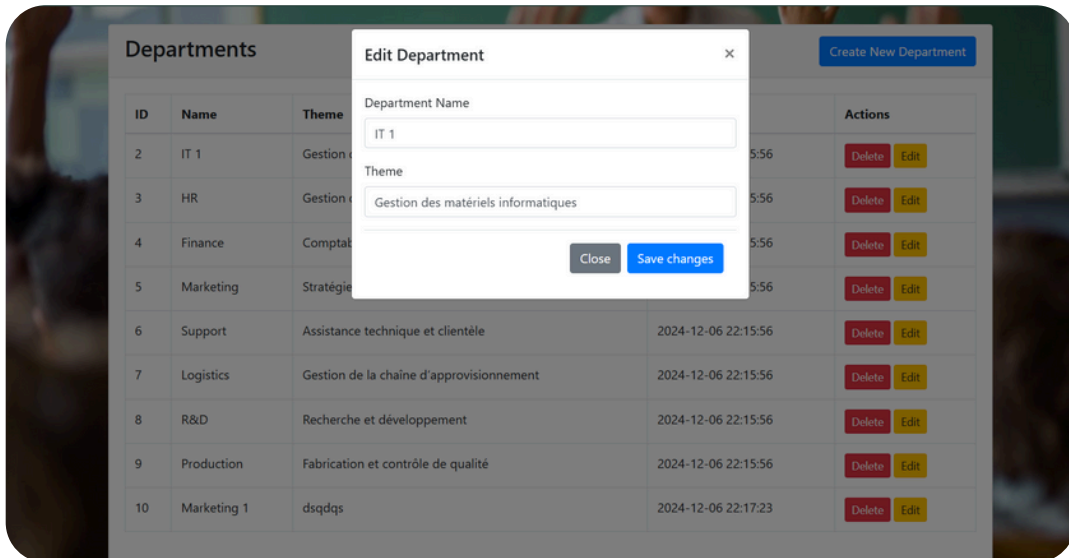
Ajouter utilisateur

Departments					Create New Department
ID	Name	Theme	Creation Date	Actions	
2	IT 1	Gestion des matériels informatiques	2024-12-06 22:15:56	Delete	Edit
3	HR	Gestion des ressources humaines	2024-12-06 22:15:56	Delete	Edit
4	Finance	Comptabilité et finances	2024-12-06 22:15:56	Delete	Edit
5	Marketing	Stratégies de promotion et publicités	2024-12-06 22:15:56	Delete	Edit
6	Support	Assistance technique et clientèle	2024-12-06 22:15:56	Delete	Edit
7	Logistics	Gestion de la chaine d'approvisionnement	2024-12-06 22:15:56	Delete	Edit
8	R&D	Recherche et développement	2024-12-06 22:15:56	Delete	Edit
9	Production	Fabrication et contrôle de qualité	2024-12-06 22:15:56	Delete	Edit
10	Marketing 1	dsqdg	2024-12-06 22:17:23	Delete	Edit

Gesttionnes les Departements



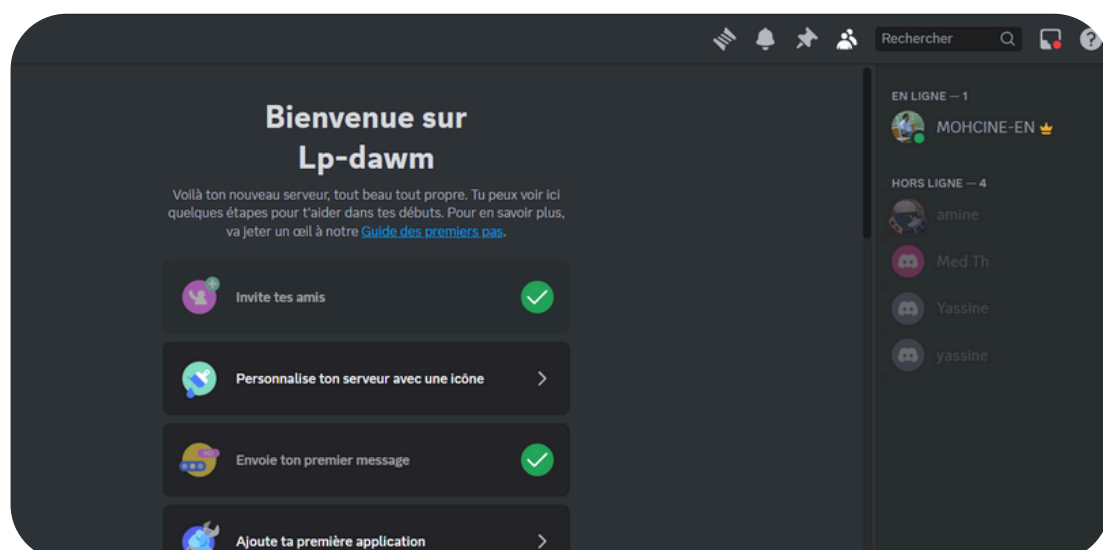
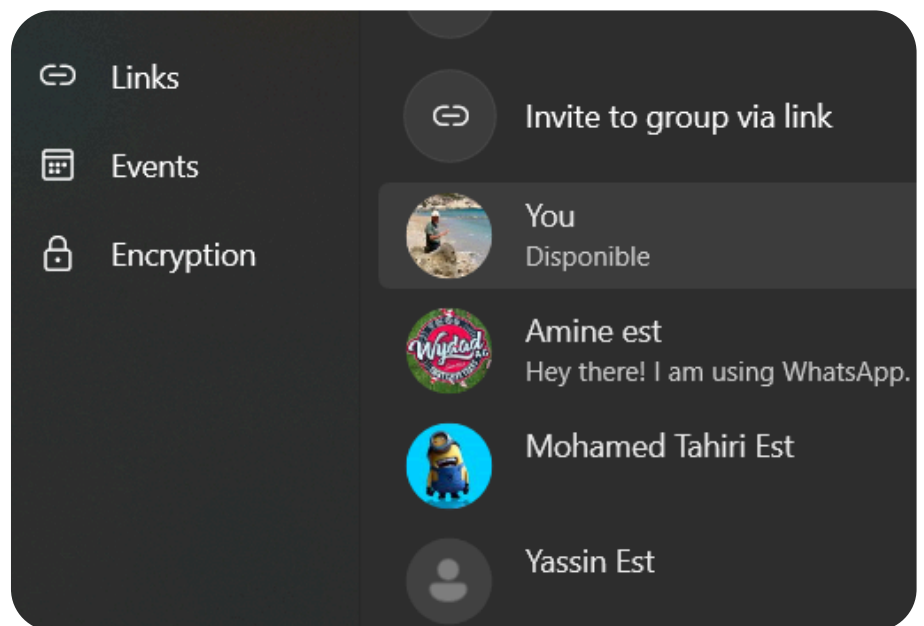
Ajouter departementss



Modifier departements

2.8 Communication et Collaboration de Groupe

La communication entre les membres de l'équipe a été facilitée grâce à WhatsApp pour les échanges instantanés et Discord pour les discussions de groupe, les appels vocaux et les réunions en ligne. De plus, nous avons eu des cours en présentiel à l'École Supérieure de Casablanca , ce qui a renforcé la collaboration et la cohésion au sein de l'équipe.



2.9 Tests de l'Application

La phase de test a assuré la fonctionnalité et l'utilisabilité :

- Tests Fonctionnels : Vérification du suivi des absences, des rapports et de l'importation des données.
- Tests d'Utilisabilité : Évaluation des interactions utilisateur et de la clarté de l'interface.
- Tests de Performance : Évaluation des temps de réponse du système sous différentes charges.

-

Exemple de cas de test :

- Scénario : Importation des données des étudiants depuis un fichier Excel.
- Résultat Attendu : Les données doivent être correctement mappées dans la base de données sans erreurs.

3. CONCLUSION

Le projet a abouti à une application web efficace pour la gestion des absences. Les principaux résultats incluent :

- Suivi et rapport des absences automatisés.
- Simplification des flux de travail administratifs.

Les défis, tels que la précision des données lors des importations, ont été résolus efficacement. Les améliorations futures pourraient inclure des analyses avancées et une intégration mobile pour étendre l'accessibilité et la fonctionnalité.