

Proposition de Projet : Gestion des Employés en PHP

Introduction

Titre du projet : Gestion des Employés en PHP

Étudiants impliqués :

Étudiant 1: Farah El Khatib 108791

Étudiant 2: Hadi Baghdadi 109035

Étudiant 3: Bahaa Eldine Mezher 107667

Description du projet :

Le Système de Gestion des Employés est une application web dédiée à la gestion centralisée des informations des employés au sein des entreprises. Elle permet aux administrateurs RH de suivre les performances des employés, gérer les tâches, les présences et leurs compétences, tout en offrant aux employés un accès à leurs informations personnelles et à leurs tâches. L'objectif est de centraliser et simplifier la gestion des employés, optimisant ainsi la productivité et l'efficacité de l'entreprise.

Problème résolu

Problème actuel :

De nombreuses entreprises gèrent encore les informations des employés manuellement ou avec des systèmes désuets. Cela conduit à une gestion inefficace des données, des erreurs humaines fréquentes, et une perte de temps considérable. Les employés, les responsables des ressources humaines (RH), et les administrateurs doivent souvent faire face à des processus lents et peu intuitifs.

Solution proposée :

Le projet phpemployee résout ce problème en introduisant une solution automatisée, moderne et intuitive qui offre une gestion centralisée des informations des employés, améliore la productivité et réduit le risque d'erreurs humaines. L'application offre également une interface sécurisée, permettant un contrôle d'accès précis et une gestion optimale des données.

Public cible

Le public cible de cette application inclut :

Administrateurs d'entreprise :

Les **administrateurs** seront responsables de la gestion globale de l'application. Ils auront la capacité d'ajouter, modifier et supprimer les informations des employés, gérer les rôles et permissions des utilisateurs pour garantir la sécurité des données, et assumer la supervision des tâches et des performances des employés. Ils pourront assigner des tâches aux employés en fonction de leurs compétences et niveaux, et suivre les progrès à travers des rapports de performance et des statistiques détaillées. En outre, les administrateurs géreront les présences et absences des employés, et pourront ajuster les affectations de tâches selon les besoins.

Employés :

Les **employés**, quant à eux, auront un accès direct à leurs informations personnelles et professionnelles. Ils pourront consulter et mettre à jour leurs données, accepter ou rejeter les tâches qui leur sont attribuées en fonction de leurs compétences et de leur disponibilité. De plus, ils pourront marquer les tâches comme "complètes" ou "incomplètes" selon leur avancement, en respectant des dates spécifiques. Ils enregistreront leur présence et suivront leur historique de tâches et de performances, contribuant ainsi à une meilleure organisation de leur emploi du temps et de leurs responsabilités au sein de l'entreprise.

Fonctionnalités principales:

Gestion des utilisateurs : Inscription, authentification et gestion des rôles (administrateurs, employés) avec des permissions spécifiques.

Gestion des employés (CRUD) : Création, lecture, mise à jour et suppression des profils des employés, avec des informations personnelles, professionnelles et de performance.

Gestion des tâches (CRUD) : Création, attribution, suivi et mise à jour des tâches. Les employés peuvent accepter ou rejeter les tâches, et marquer leur progression.

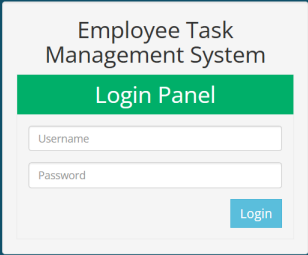
Gestion des présences : Enregistrement des présences et des absences, suivi des demandes de congé, et génération de rapports d'absentéisme.

Suivi de la performance : Suivi des indicateurs de performance liés aux tâches (taux de complétion, qualité du travail, délais) et génération de rapports analytiques.

Analyses et rapports : Génération de rapports personnalisés sur les présences, la performance des employés et d'autres données pertinentes pour la gestion.

Authentification des utilisateurs :

Seuls les utilisateurs authentifiés peuvent accéder à l'application. Cette fonctionnalité est cruciale pour garantir la sécurité des informations.



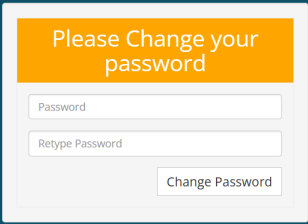
Employee Task Management System

Login Panel

Username

Password

Login



Please Change your password

Password

Retype Password

Change Password

Gestion des employés (CRUD) :

Créer (Create) :

- Lorsqu'un nouvel employé rejoint l'entreprise, l'administrateur peut facilement ajouter ses informations dans le système. Cela inclut des détails tels que le nom complet, email, les compétences, et le niveau d'expérience (débutant, intermédiaire, avancé). Le système génère un identifiant unique pour chaque employé afin de faciliter la gestion et la recherche des informations. Une fois créés, les profils des employés peuvent être associés à des tâches et des performances.

Lire (Read) :

- L'administrateur peut consulter et visualiser les informations détaillées de chaque employé à travers une interface intuitive. Cela inclut la possibilité de voir l'historique des tâches assignées, les performances passées, les absences et les présences. Par exemple, un tableau de bord centralisé permet à l'administrateur de rechercher des employés par nom, rôle, ou département, et de cliquer sur leurs profils pour obtenir plus de détails. Les filtres de recherche aident à naviguer rapidement dans la base de données d'employés.

Mettre à jour (Update) :

- Si un employé change de rôle, de département, ou si ses informations personnelles sont mises à jour, l'administrateur peut modifier facilement ces informations dans son profil. Par exemple, si un employé a suivi une formation et a acquis de nouvelles compétences, ces informations peuvent être ajoutées dans le système. L'administrateur peut aussi mettre à jour le statut des employés en fonction des changements dans leur carrière ou leur performance.

Supprimer (Delete) :

- Lorsqu'un employé quitte l'entreprise ou si son profil doit être supprimé pour toute autre raison, l'administrateur peut le retirer du système. Cependant, avant de supprimer un employé, il est important de s'assurer que ses tâches en cours soient réassignées à d'autres membres de l'équipe. De même, les informations de présence et les rapports doivent être soit archivés, soit transférés. La suppression des données doit se faire en respectant les normes de confidentialité et de sécurité des informations personnelles des employés.

1



2



ETMS

Task Mangement

Attendance

Administration

Statistics

Logout

Manage Admin

Manage Employee

Serial No.	Name	Email	Username	Details
1	Admin	admin@gmail.com	admin	

3

ETMS

Task Mangement

Attendance

Administration

Statistics

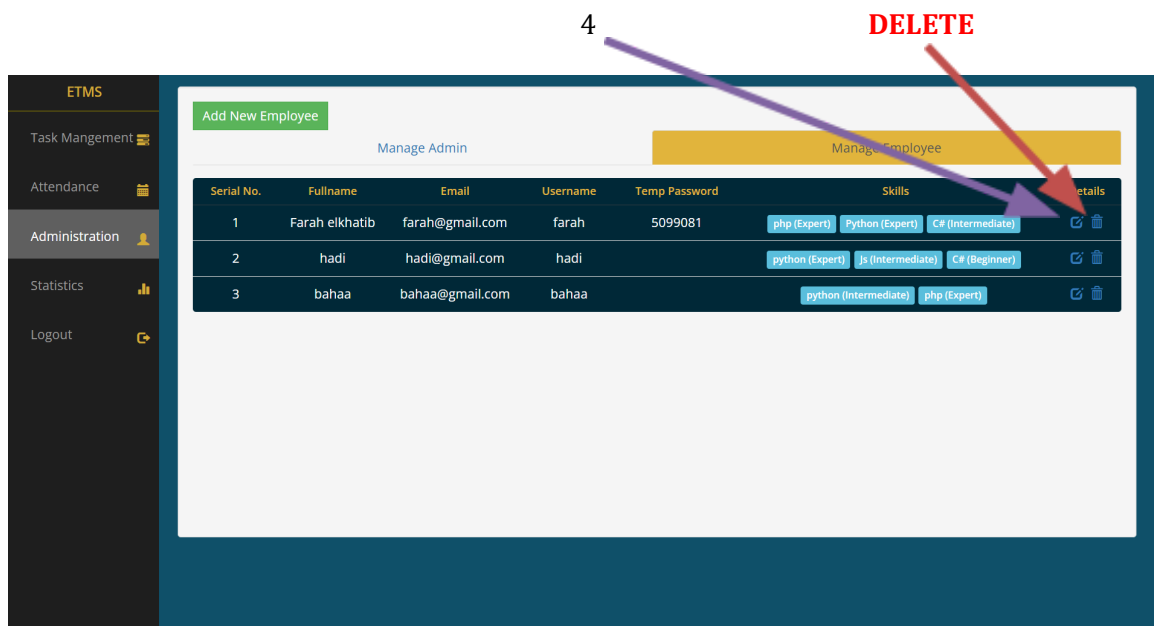
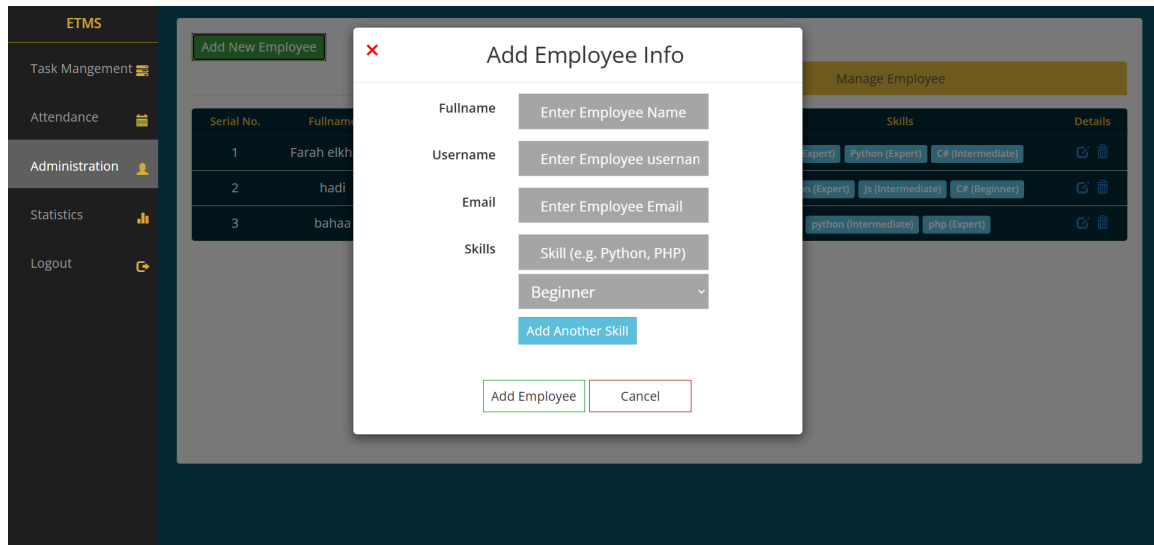
Logout

Add New Employee

Manage Admin

Manage Employee

Serial No.	Fullname	Email	Username	Temp Password	Skills	Details
1	Farah elkhatib	farah@gmail.com	farah	5099081	php (Expert) python (Expert) C# (Intermediate)	
2	hadi	hadi@gmail.com	hadi		python (Expert) js (Intermediate) C# (Beginner)	
3	bahaa	bahaa@gmail.com	bahaa		python (Intermediate) php (Expert)	



Gestion des tâches des employés (CRUD) :

Créer (Create) :

- Les administrateurs peuvent **créer** de nouvelles tâches en spécifiant des détails tels que le titre, la description, la date d'échéance et l'employé assigné. Chaque tâche peut être attribuée à un ou plusieurs employés en fonction de leurs compétences et de leur disponibilité.

Lire (Read) :

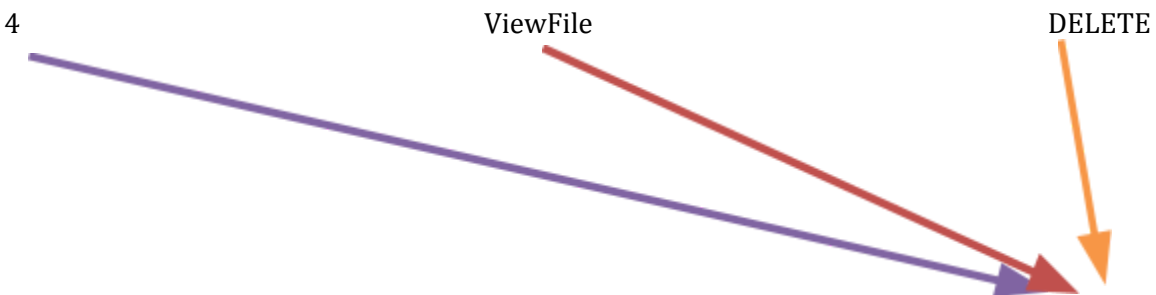
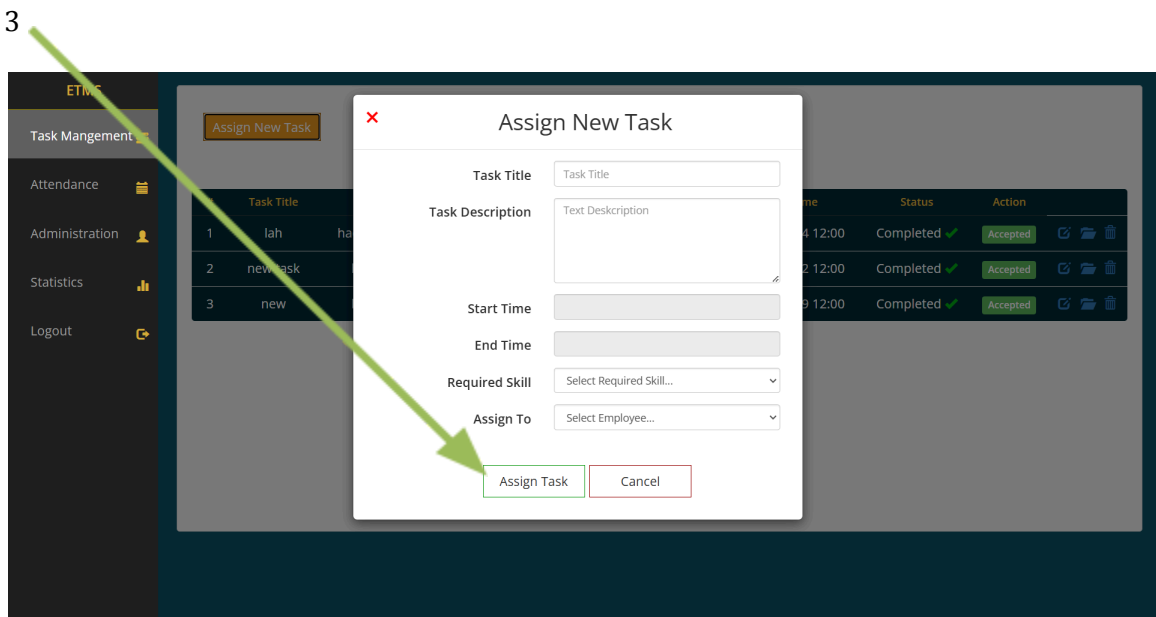
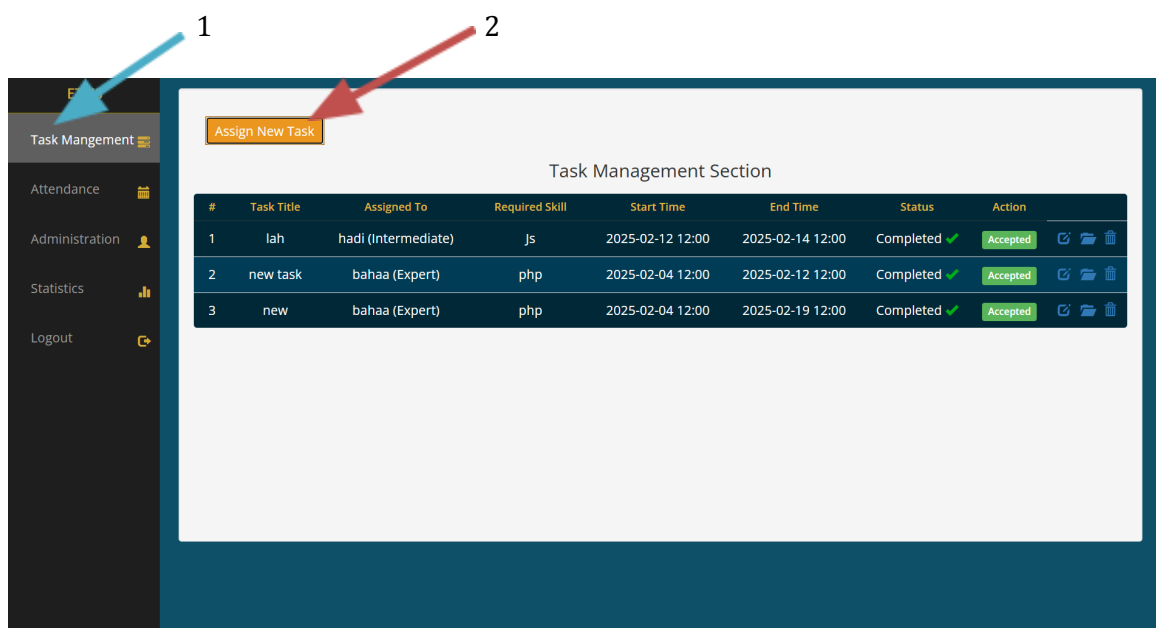
- Les administrateurs peuvent **consulter** l'ensemble des tâches assignées aux employés. Cela inclut la possibilité de voir les tâches en cours, terminées ou rejetées. Des filtres permettent de trier les tâches par date, employé, ou statut pour faciliter la gestion.

Mettre à jour (Update) :

- Les administrateurs peuvent **mettre à jour** les informations de chaque tâche. Par exemple, ils peuvent modifier le statut d'une tâche (en cours, terminé, rejeté) ou attribuer une nouvelle tâche à un employé si nécessaire. De plus, ils peuvent ajuster les délais ou apporter des modifications au contenu de la tâche.

Supprimer (Delete) :

- Les administrateurs peuvent **supprimer** des tâches qui ne sont plus nécessaires ou qui ont été annulées. Cela permet de maintenir la base de données des tâches propre et organisée, en s'assurant que seules les tâches pertinentes et actives restent.



ETMS

Task Mangement

Attendance

Administration

Statistics

Logout

Assign New Task

Task Management Section

#	Task Title	Assigned To	Required Skill	Start Time	End Time	Status	Action
1	lah	hadl (Intermediate)	Js	2025-02-12 12:00	2025-02-14 12:00	Completed	<div>Accepted</div> <div></div> <div></div> <div></div>

| 2 | new task | bahaa (Expert) | php | 2025-02-04 12:00 | 2025-02-12 12:00 | Completed | Accepted |

Gestion de l'assiduité des employés :

Suivi des présences, des absences, et des congés des employés. Cette fonctionnalité aidera les responsables RH à surveiller les horaires et la ponctualité.

1

2

The screenshot shows the ETMS (Employee Time Management System) interface. The left sidebar contains the following menu items: ETMS, Task Management, Attendance, Administration, Statistics, and Logout. The main content area is titled 'Manage Attendance' and features a 'Clock In' button at the top left. Below the button is a table with the following data:

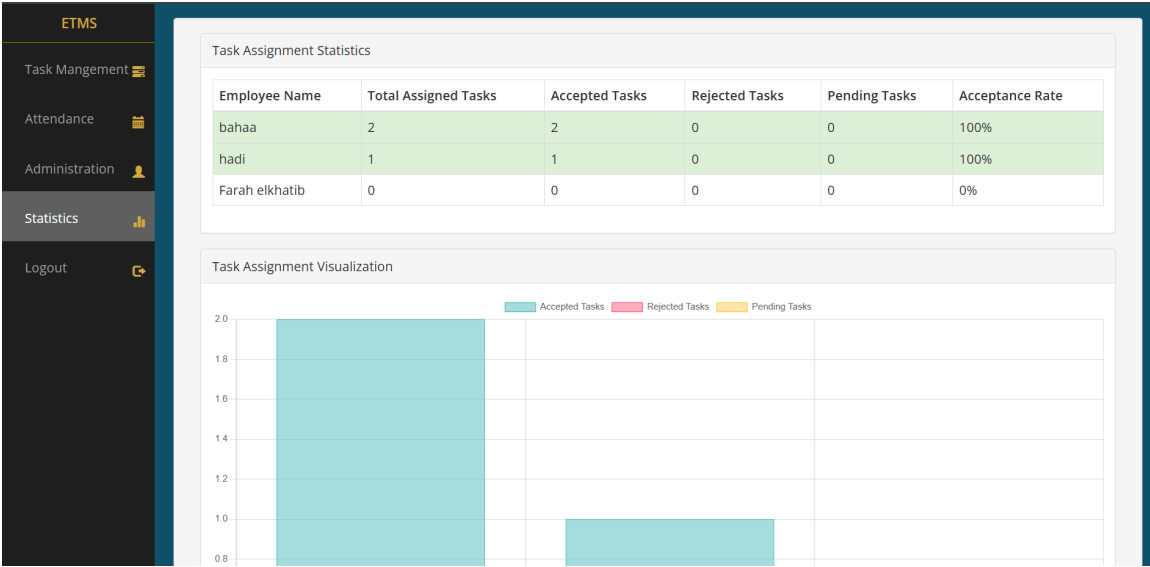
S.N.	Name	In Time	Out Time	Total Duration	Status	Action
1	Admin	13-02-2025 01:55:29	13-02-2025 01:55:32	00:00:03		
2	bahaa	04-02-2025 11:49:31	04-02-2025 11:49:34	00:00:03		
3	Admin	04-02-2025 11:48:07	04-02-2025 11:49:56	00:01:49		
4	Admin	22-03-2021 22:01:43	04-02-2025 11:14:29	13:12:46		

Gestion des compétences :

Créer et suivre les compétences des employés (ex. : PHP, Réseau, Base de données, HTML/CSS, etc.).

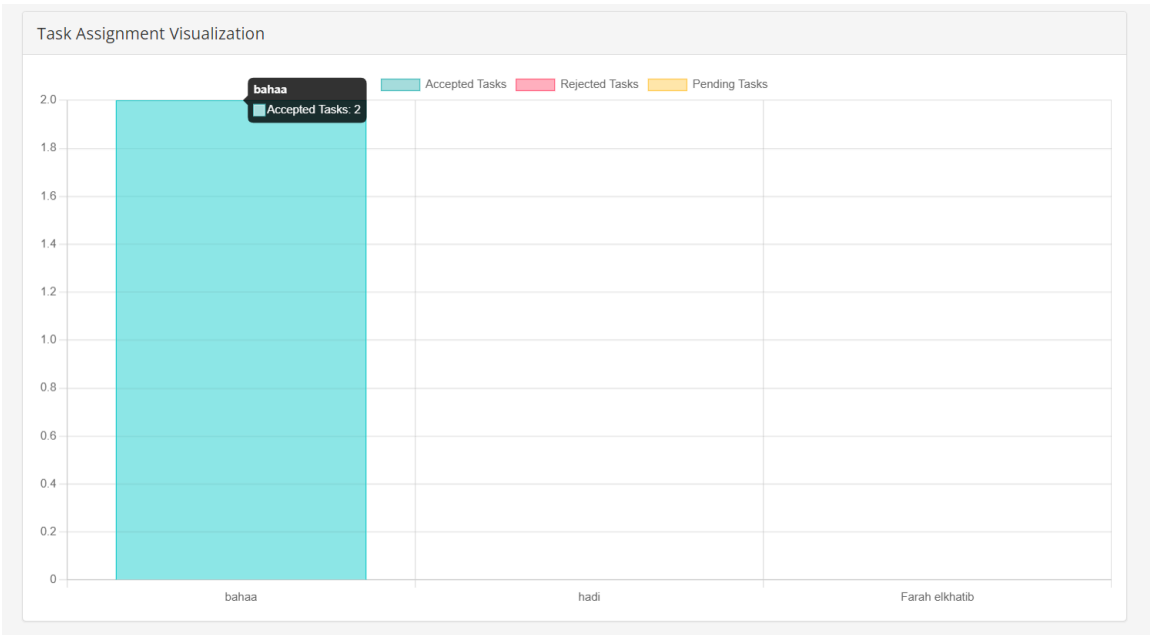
Suivi des niveaux de compétence des employés pour mieux orienter les formations et les missions.





Task Assignment Statistics

Employee Name	Total Assigned Tasks	Accepted Tasks	Rejected Tasks	Pending Tasks	Acceptance Rate
bahaa	2	2	0	0	100%
hadi	0	0	0	0	0%
Farah elkhatib	0	0	0	0	0%



Conception de la base de données

Entités clés du système :

skill_id : Identifiant des compétences des employés et niveaux de compétence (PHP, Réseau, Base de données, HTML/CSS).

attendance_id : Identifiant de suivi des présences des employés, y compris les dates et le statut (présent, absent, etc.).

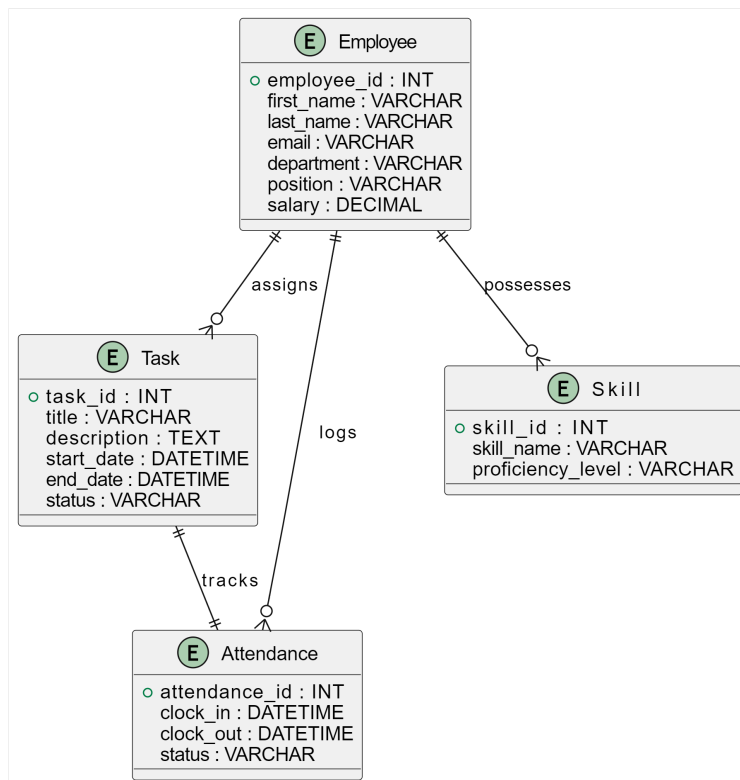
task_id : Identifiant des tâches assignées aux employés, avec des descriptions et des dates limites.

user_id : Identifiant pour gérer les comptes utilisateurs, avec les informations de connexion.

Modèles et migrations :

Modèles:

- **Employee ↔ Task**: An employee can have multiple tasks.
- **Employee ↔ Attendance**: An employee can have multiple attendance records.
- **Employee ↔ Skill**: An employee can have multiple skills with proficiency levels.
- **Task ↔ Attendance**: A task may track the attendance for the employee assigned to it.



MIGRATIONS

```
CREATE TABLE employees (  
    employee_id INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) UNIQUE NOT NULL,  
    department VARCHAR(255),  
    position VARCHAR(255),  
    salary DECIMAL(10,2)  
);
```

```
CREATE TABLE tasks (  
    task_id INT AUTO_INCREMENT PRIMARY KEY,  
    employee_id INT,  
    title VARCHAR(255) NOT NULL,  
    description TEXT,  
    start_date DATETIME,  
    end_date DATETIME,  
    status VARCHAR(50),  
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id) ON DELETE CASCADE  
);
```

```
CREATE TABLE attendance (  
    attendance_id INT AUTO_INCREMENT PRIMARY KEY,  
    employee_id INT,  
    clock_in DATETIME,  
    clock_out DATETIME,  
    status VARCHAR(50),  
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id) ON DELETE CASCADE
```

```
);
```

```
CREATE TABLE skills (  
    skill_id INT AUTO_INCREMENT PRIMARY KEY,  
    employee_id INT,  
    skill_name VARCHAR(255),  
    proficiency_level VARCHAR(50),  
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id) ON DELETE CASCADE  
);
```

Conception du Front-End

Le projet utilise HTML pour structurer les pages, CSS pour le design, et JavaScript pour l'interactivité et la validation des formulaires côté client. Le framework Bootstrap est intégré pour rendre l'interface responsive et moderne, assurant une expérience utilisateur fluide et accessible sur tous les appareils. Cette approche garantit également que les informations soumises par les utilisateurs soient validées avant d'être envoyées au serveur, minimisant ainsi les erreurs.

Interaction utilisateur :

Les utilisateurs interagiront avec l'application via un tableau de bord centralisé où ils pourront visualiser et gérer les informations des employés. Des formulaires de création et de modification seront mis à disposition pour faciliter l'édition des informations.

Wireframes :

Des wireframes doivent être créés pour chaque page clé de l'application. Cela inclut, par exemple, une page de connexion où l'utilisateur peut saisir ses identifiants, un tableau de bord de gestion des employés pour afficher et gérer les informations des employés, ainsi qu'une page dédiée à la gestion des tâches. Ces wireframes aideront à visualiser l'interface et à planifier l'organisation des différentes sections de l'application avant de procéder au développement.

Conception du Back-End

Le back-end de l'application sera développé en PHP. La logique de l'application gèrera toutes les demandes des utilisateurs, telles que l'authentification, la gestion des employés et des tâches, et le suivi des présences.

Mécanismes de sécurité et de middleware :

La sécurité sera mise en œuvre en utilisant des mécanismes d'authentification basés sur des sessions et des jetons pour garantir que seuls les utilisateurs autorisés aient accès à l'application.

Ces jetons seront utilisés pour vérifier l'identité et maintenir la session de l'utilisateur active. Les erreurs et les événements seront enregistrés pour assurer une gestion efficace des incidents et une traçabilité des actions réalisées dans l'application.

Rôles et autorisations

Il y aura plusieurs types d'utilisateurs, avec des niveaux d'accès différents :

Administrateur : Peut gérer tous les aspects de l'application (ajouter/modifier/supprimer des employés, tâches, etc.).

Employé : Peut uniquement consulter ses propres informations et ses tâches.

Défis techniques

Les principaux défis techniques peuvent inclure : La gestion des rôles complexes et des autorisations, L'optimisation des requêtes pour éviter les problèmes de performance avec de grandes bases de données, L'implémentation d'une interface utilisateur fluide et intuitive pour une expérience agréable.

Calendrier de développement (1.5 mois) :

- **Semaine 1-2** : Analyse des besoins, conception de la base de données, développement des fonctionnalités de gestion des utilisateurs et des employés.
- **Semaine 3-4** : Implémentation de la gestion des tâches, des présences et des statistiques des employés.
- **Semaine 5** : Tests fonctionnels, débogage, et amélioration de l'interface.
- **Semaine 6** : Documentation du projet et préparation de la présentation finale

Calendrier

Phase 1 : Configuration et migrations de la base de données - Durée : 2 semaines

Phase 2 : Développement du back-end (logique de gestion des employés et des tâches) - Durée : 4 semaines

Phase 3 : Conception du front-end et intégration avec le back-end - Durée : 3 semaines

Phase 4 : Tests et déploiement - Durée : 2 semaines

Conclusion

Le projet phpemployee représente une solution efficace et moderne pour la gestion des employés dans les entreprises. Il permet d'automatiser de nombreuses tâches administratives et d'améliorer la productivité tout en garantissant la sécurité et l'intégrité des données.