

AKYUREK Eren

BTS SIO SLAM

SESSION 2026

# PAPPE



Liens :

TimeTracker : [https://github.com/Eren-Akyurek/TimeTracker\\_GSB](https://github.com/Eren-Akyurek/TimeTracker_GSB)

Portfolio : <https://eren-akyurek.github.io/portfolio/>

<b>BTS SERVICES INFORMATIQUES AUX ORGANISATIONS</b>	<b>SESSION 2025</b>
<b>ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)</b>	
<b>Épreuve E5 - Conception et développement d'applications (option SLAM)</b>	

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>	<b>N° réalisation : 1</b>
<b>Nom, prénom : AKYUREK Eren</b>	<b>N° candidat : 02249605006</b>
<b>Épreuve ponctuelle</b> <input checked="" type="checkbox"/> <b>Contrôle en cours de formation</b> <input type="checkbox"/>	<b>Date : 19 / 12 / 2025</b>
<b>Organisation support de la réalisation professionnelle</b> La réalisation s'inscrit dans le cadre d'un Projet Personnalisé Encadré (PPE), fondé par le centre de formation. Il concerne une entreprise pharmaceutique fictive, nommée Laboratoire Galaxy Swiss Bourdin (GSB). L'objectif était de concevoir des outils destinés à améliorer le suivi de l'activité.	
<b>Intitulé de la réalisation professionnelle</b> Développement de l'application ' TimeTracker GSB ' : Gestion centralisée des fiches de temps et des demandes d'absences	
<b>Période de réalisation : Oct .2025 – Jan. 2026</b> <b>Lieu :</b> Dans le cadre de la formation <b>Modalité :</b> <input checked="" type="checkbox"/> <b>Seul(e)</b> <input type="checkbox"/> <b>En équipe</b>	
<b>Compétences travaillées</b> <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données	
<b>Conditions de réalisation<sup>5</sup> (ressources fournies, résultats attendus)</b> Ressources fournies : Contexte GSB et cahier des charges. Résultats attendus : Une application web fonctionnelle permettant l'authentification sécurisée par rôles, la saisie et validation des temps (calcul automatique des jours ouvrés), la gestion des congés et un tableau de bord pour le workflow de validation RH.	
<b>Description des ressources documentaires, matérielles et logicielles utilisées<sup>6</sup></b> <ul style="list-style-type: none"> <li>- <b>Environnement de développement (IDE) :</b> Cursor (éditeur basé sur VS Code).</li> <li>- <b>Serveur web local :</b> XAMPP (Apache).</li> <li>- <b>Système de Gestion de Base de Données (SGBD) :</b> MySQL (géré via l'interface phpMyAdmin incluse dans XAMPP).</li> <li>- <b>Langages utilisés :</b> PHP (Back-end), HTML5, CSS3, JavaScript Vanilla (Front-end), SQL.</li> <li>- <b>Gestion de versions et collaboration :</b> Git (en ligne de commande) et GitHub (dépôt distant).</li> </ul>	
<b>Modalités d'accès aux productions<sup>7</sup> et à leur documentation<sup>8</sup></b> Repository Github : <a href="https://github.com/Eren-Akyurek/TimeTracker_GSB">https://github.com/Eren-Akyurek/TimeTracker_GSB</a> Portfolio professionnel : Consultable en ligne à l'adresse suivante : <a href="https://Eren-Akyurek.github.io/portfolio/">https://Eren-Akyurek.github.io/portfolio/</a>	

<sup>5</sup> En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>6</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

<sup>7</sup> Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

<sup>8</sup> Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

**ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)****Épreuve E5 - Conception et développement d'applications (option SLAM)****Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

**1. Contexte et objectifs du projet** Dans le cadre de la modernisation de son système d'information, le laboratoire GSB a exprimé le besoin de remplacer sa gestion papier/Excel des temps de travail par une solution intranet. Le projet "TimeTracker GSB" est une application web sur-mesure développée en PHP/MySQL permettant de dématérialiser la saisie des heures et d'intégrer un workflow de validation des absences.

**2. Fonctionnalités développées (selon les profils utilisateurs)****Espace Collaborateur (Employé) :**

- **Authentification sécurisée** : Connexion via identifiants uniques hachés en base de données.
- **Gestion du temps** : Saisie et suivi des heures de travail effectuées.
- **Demandes d'absences** : Formulaire de déclaration de congés ou d'absences avec sélection des dates et motifs.
- **Suivi** : Tableau de bord personnel permettant de consulter l'état de ses demandes (En attente, Validée, Refusée).

**Espace Manager / Administrateur :**

- **Supervision** : Tableau de bord global avec vue sur les compteurs de l'équipe.
- **Workflow de validation** : Interface dédiée permettant d'accepter ou de refuser les demandes de congés des collaborateurs (changement d'état dans la base de données).
- **Gestion des utilisateurs** : (CRUD) Création, modification et suppression des comptes employés et de leurs rôles.

**3. Architecture et Base de données**

- Modélisation et conception d'une base de données relationnelle (MCD / MLD).
- Création de tables interconnectées pour gérer les utilisateurs, les rôles (droits d'accès), les feuilles de temps et les statuts de validation des congés.
- Sécurisation des requêtes SQL (prévention des injections).

**4. Livrables et productions réalisées**

- L'application web fonctionnelle (déployée sur environnement local XAMPP).
- Le script SQL de création et d'alimentation de la base de données (Jeu d'essai).
- Le dossier complet de documentation (PAPPE) comprenant la documentation utilisateur (manuel avec captures d'écran) et la documentation technique (dictionnaire de données, architecture, règles de gestion).

# Sommaire

## 1. Avant-Projet et Cahier des Charges

- 1.1. Présentation de l'organisation : Galaxy Swiss Bourdin (GSB)
- 1.2. Contexte et origine de la demande
- 1.3. Les acteurs du projet et profils utilisateurs
- 1.4. Besoins fonctionnels (Fonctionnalités attendues)
- 1.5. Contraintes techniques et architecture
- 1.6. Organisation et Planification

## 2. Analyse et Conception (Modélisation)

- 2.1. Diagramme des Cas d'Utilisation (Use Cases)
- 2.2. Architecture Applicative : Le Modèle MVC
- 2.3. Dictionnaire de données
- 2.4. Modèle Conceptuel de Données (MCD)

## 3. Manuel Utilisateur (Présentation des interfaces)

- 3.1. L'authentification et gestion des sessions
- 3.2. Espace Employé : Tableau de bord et saisie des temps
- 3.3. Espace Employé : Demande de congés
- 3.4. Espace Manager : Workflow de validation
- 3.5. Espace Administrateur : Gestion des comptes (CRUD)

#### 4. Réalisation Technique (Extraits de code)

- 4.1. Environnement de développement et déploiement local
- 4.2. Sécurité : Authentification et protection CSRF
- 4.3. Logique métier : Le Workflow de validation
- 4.4. Interaction Base de données : Requêtes préparées et Upsert

#### 5. Phase de Tests et Recettage

- 5.1. Méthodologie de tests
- 5.2. Tableaux des tests fonctionnels

#### 6. Bilan et Perspectives

- 6.1. Bilan de la réalisation
- 6.2. Perspectives d'évolution

# 1. Avant-Projet et Cahier des Charges

## 1.1. Présentation de l'organisation : Galaxy Swiss Bourdin (GSB)

Le laboratoire pharmaceutique Galaxy Swiss Bourdin (GSB) est une entité née de la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin.

L'entreprise est spécialisée dans la recherche, le développement et la distribution de produits pharmaceutiques.

Par sa nature, GSB emploie une très grande variété de collaborateurs (chercheurs, visiteurs médicaux, personnel administratif, techniciens) répartis sur de multiples sites géographiques et laboratoires.

Cette dispersion géographique et la diversité des statuts rendent la gestion des ressources humaines particulièrement complexe.

## 1.2. Contexte et origine de la demande

Jusqu'à présent, la gestion des emplois du temps, des temps de travail et des demandes de congés au sein de GSB s'effectuait de manière décentralisée et souvent asynchrone (utilisation de tableurs Excel, formulaires papier, échanges d'emails).

Cette méthode de fonctionnement présentait plusieurs limites majeures :

- Perte de temps considérable pour le service des Ressources Humaines.
- Risque élevé d'erreurs de saisie ou de perte de documents.
- Manque de visibilité en temps réel pour la direction sur le "staffing" (affectation des employés).
- Frustration des employés face à l'opacité du suivi de leurs demandes de congés.

Pour pallier ces dysfonctionnements, la Direction des Systèmes d'Information (DSI) de GSB a commandité la création de "**TimeTracker**", une application

web intranet centralisée, afin de dématérialiser et d'automatiser l'ensemble de ces processus RH.

### 1.3. Les acteurs du projet et profils utilisateurs

L'application a été pensée pour répondre aux besoins spécifiques de trois catégories d'utilisateurs distinctes, chacune possédant des droits d'accès et des vues différentes :

#### 1. **Les Employés (Personnel de terrain, technique et administratif) :**

Ils sont les utilisateurs principaux de la plateforme. Leur besoin est de pouvoir consulter leur planning, déclarer leurs heures et soumettre des demandes d'absences de manière autonome.

#### 2. **Les Managers et le service RH :**

Ils encadrent les équipes. Leur rôle est d'avoir une vue globale, de gérer la planification (affectation sur les projets/sites) et de valider ou refuser les demandes soumises par les employés.

#### 3. **Les Administrateurs :** Le service informatique. Ils ont la charge du maintien de la plateforme, de la gestion de la base de données et de la configuration des comptes utilisateurs (CRUD complet).

### 1.4. Besoins fonctionnels (Fonctionnalités attendues)

Sur la base des besoins exprimés par le client, les fonctionnalités suivantes ont été définies et implémentées :

#### **Pour l'espace Employé :**

- Authentification sécurisée par identifiant (email) et mot de passe.
- Tableau de bord (Homepage) affichant les alertes et notifications récentes.

- Interface de visualisation de l'emploi du temps (jours travaillés, affectations aux projets/sites).
- Module de soumission des jours de travail via une vue mensuelle.
- Formulaire dédié pour la demande de congés/absences et suivi en temps réel du statut de ces demandes (En attente, Validée, Refusée).
- Consultation du temps de travail hebdomadaire cumulé.

#### **Pour l'espace RH / Manager :**

- Tableau de bord global permettant le suivi de l'ensemble des collaborateurs.
- Interface de Workflow : Validation ou refus des demandes de congés en attente.
- Planification de l'affectation des employés sur les différents projets de l'entreprise (Staffing).
- Module de statistiques : Consultation détaillée sur le temps de travail effectif et les compteurs de congés pris par chaque employé.

#### **Pour l'espace Administrateur :**

- Panneau de gestion des utilisateurs : Création de nouveaux comptes, modification des rôles et des informations, et suppression sécurisée des anciens collaborateurs.

#### 1.5. Contraintes techniques et architecture (Besoins non-fonctionnels)

Pour garantir la pérennité, la sécurité et la maintenabilité de l'application, des choix technologiques stricts ont été arrêtés en phase de conception :

- **Architecture Client-Serveur et Modèle MVC :**

L'application est développée en PHP natif en respectant rigoureusement le patron de conception MVC (Modèle-Vue-Contrôleur).

L'architecture repose sur un contrôleur frontal (index.php) agissant comme un mini-routeur qui distribue les requêtes vers les bons contrôleurs.

- **Front-End :**

L'interface utilisateur est construite en HTML5, CSS3 et JavaScript Vanilla, garantissant une compatibilité avec les navigateurs standards de l'entreprise. Un layout global (base.php) assure la cohérence visuelle.

- **Base de données :**

L'exploitation des données (personnel, temps, congés) repose sur un SGBD relationnel MySQL.

- **Sécurité :**

L'accès à l'intranet est strictement protégé.

L'authentification utilise des sessions PHP sécurisées.

Les mots de passe ne sont jamais stockés en clair mais sont hachés cryptographiquement en base de données.

Les requêtes SQL utilisent l'extension PDO avec des requêtes préparées pour annihiler tout risque d'injection SQL.

## 1.6. Organisation et Planification

Le développement a été mené selon une approche itérative inspirée des méthodes agiles.

Le découpage du projet s'est articulé autour des grands jalons suivants :

1. Analyse du besoin et rédaction du cahier des charges.
2. Conception de la base de données (Dictionnaire de données, MCD, MLD).
3. Maquettage des interfaces (UI/UX).
4. Développement du Back-End (Contrôleurs, Modèles et routage PHP).
5. Développement du Front-End et intégration des vues.

6. Implémentation des règles métiers complexes (Workflow de validation RH).
7. Phase de tests de validation et déploiement local.

## 2. Analyse et Conception (Modélisation)

### 2.1. Diagramme des Cas d'Utilisation (Use Cases)

Afin de modéliser les interactions entre les différents acteurs (Employé, Manager/RH, Administrateur) et le système TimeTracker, un diagramme des cas d'utilisation a été réalisé au standard UML.

Ce schéma permet d'isoler les frontières du système et de s'assurer que les restrictions d'accès sont correctement définies avant le développement.

### 2.2. Architecture Applicative : Le Modèle MVC

Pour répondre aux exigences de maintenabilité et de clarté du code, l'application a été structurée autour du patron de conception MVC (Modèle-Vue-Contrôleur) "Fait maison" en PHP natif.

- Le Routeur (Contrôleur Frontal) :

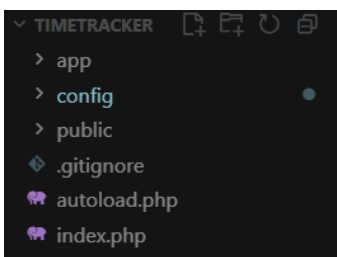
Le fichier `index.php` intercepte toutes les requêtes de l'utilisateur. Il analyse l'URL et instancie le bon contrôleur.

- Les Contrôleurs (/Controller) :

Ils contiennent la logique métier. Par exemple, l'`AbsenceController` reçoit la demande de congé, vérifie les droits de l'utilisateur, et fait le lien avec le Modèle.

- Les Modèles (/Model) : Ils sont les seuls autorisés à communiquer avec la base de données MySQL via l'objet PDO. Ils s'occupent des requêtes SQL préparées (ex: `AbsenceSheetModel`).

- Les Vues (/View) : Elles s'occupent de l'affichage HTML/TailwindCSS. Elles reçoivent les données du contrôleur et les injectent dans le gabarit principal (`base.php`).

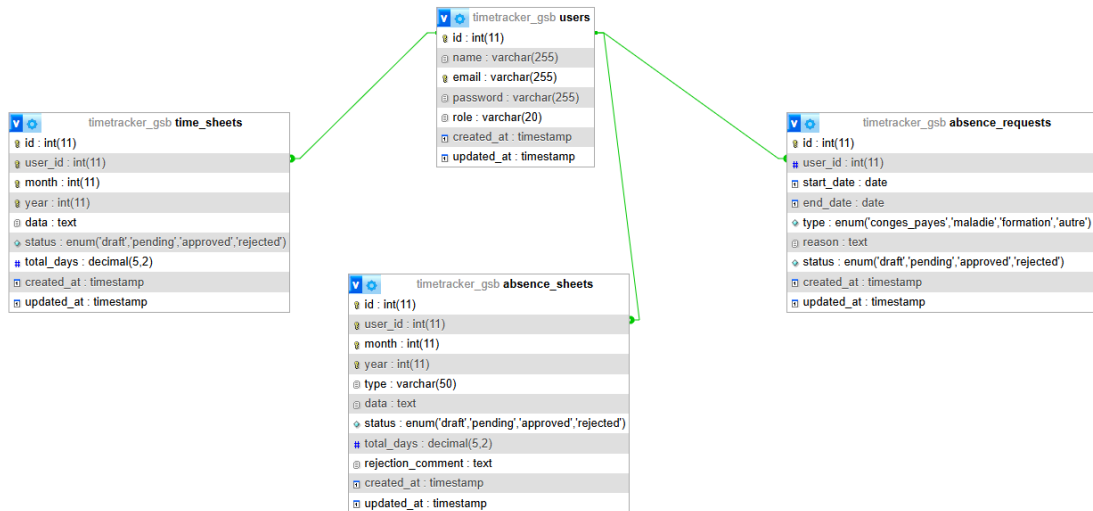


### 2.3. Le Dictionnaire de Données

Nom de la donnée	Code (Nom du champ)	Type	Taille	Description
Identifiant Utilisateur	id_user	Entier (INT)	-	Clé primaire, identifiant unique de l'employé
Nom	nom	Chaîne (VARCHAR)	50	Nom de famille de l'employé
Prénom	prenom	Chaîne (VARCHAR)	50	Prénom de l'employé
Adresse Email	email	Chaîne (VARCHAR)	100	Email de connexion
Mot de passe	password	Chaîne (VARCHAR)	255	Mot de passe haché
Rôle de l'utilisateur	role	Chaîne (VARCHAR)	20	ex: "employe", "manager", "admin"
Identifiant Absence	id_absence	Entier (INT)	-	Clé primaire, identifiant unique de la demande
Date de début	date_debut	Date (DATE)	-	Premier jour de l'absence
Date de fin	date_fin	Date (DATE)	-	Dernier jour de l'absence
Motif	motif	Chaîne (VARCHAR)	50	ex: "Congé", "Maladie", "RTT"
Statut de la demande	statut	Chaîne (VARCHAR)	20	ex: "En attente", "Validé", "Refusé"

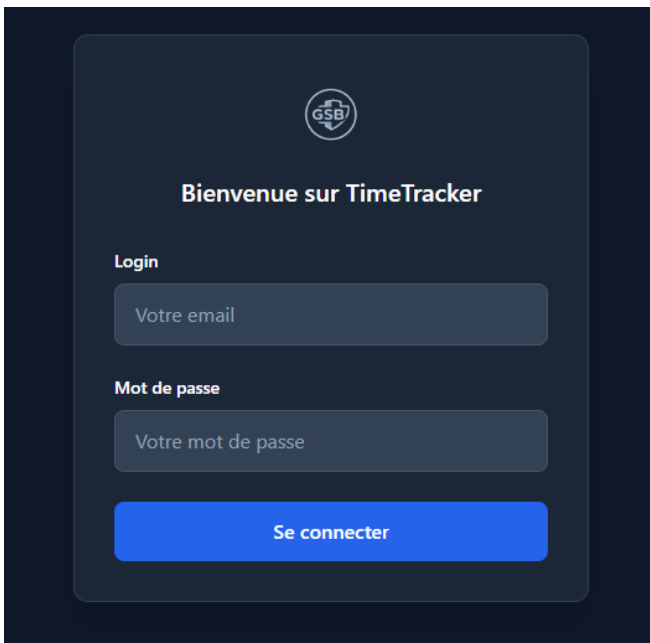
## 2.4. Le Modèle Conceptuel de Données (MCD) / Modèle Logique (MLD)

MCD \*



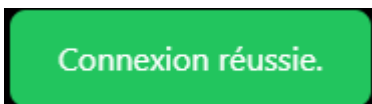
### 3. Présentation de l'application (Manuel Utilisateur)

#### 3.1. L'authentification



L'accès à l'intranet TimeTracker est restreint. Chaque collaborateur doit s'authentifier à l'aide de son adresse email professionnelle et de son mot de passe.

Le système vérifie ces identifiants en base de données et redirige l'utilisateur vers l'espace correspondant à son rôle (Employé ou Manager)



Notification de validation de connexion



## Bienvenue sur TimeTracker

Email ou mot de passe incorrect.

### Login

### Mot de passe

Se connecter

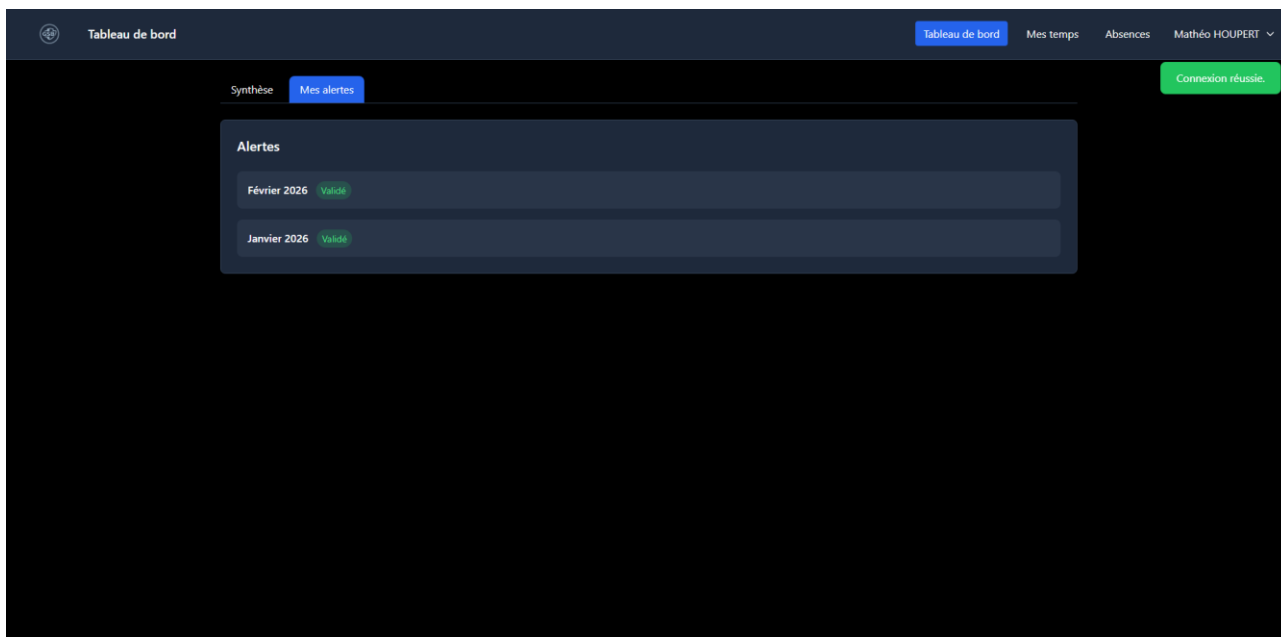
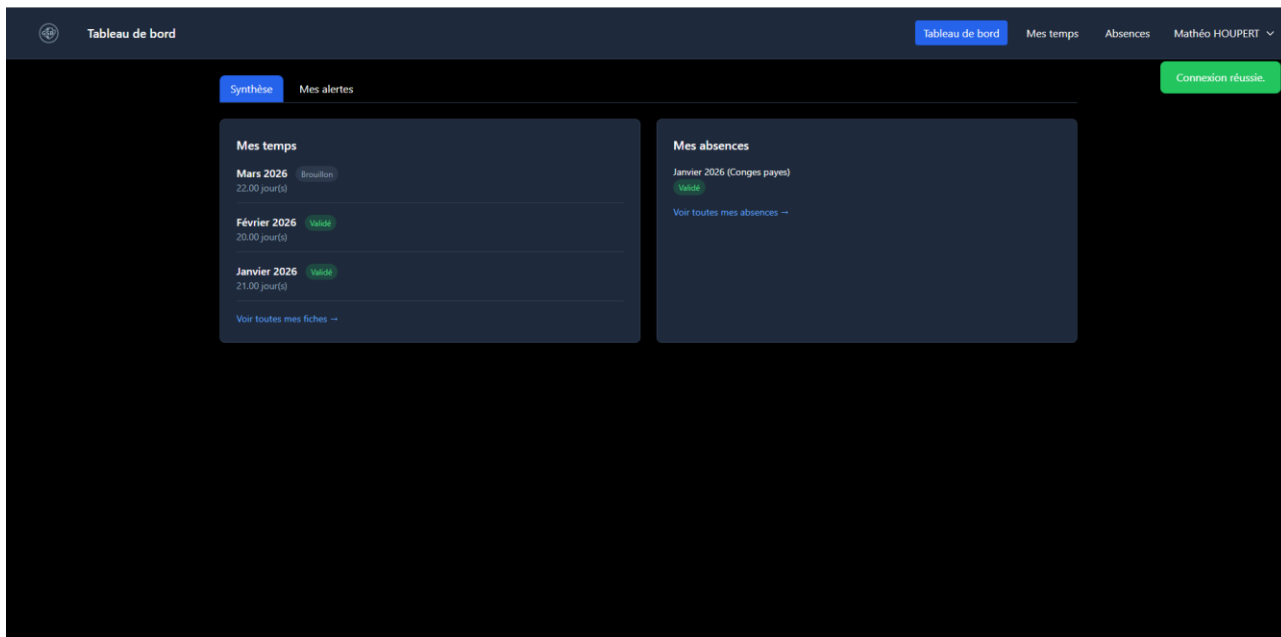
### Login



Veillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "eren.akyurek.gsb.fr".

En cas d'erreur de mot de passe, erreurs gérées.

### 3.2. Espace Collaborateur : Le Tableau de bord personnel

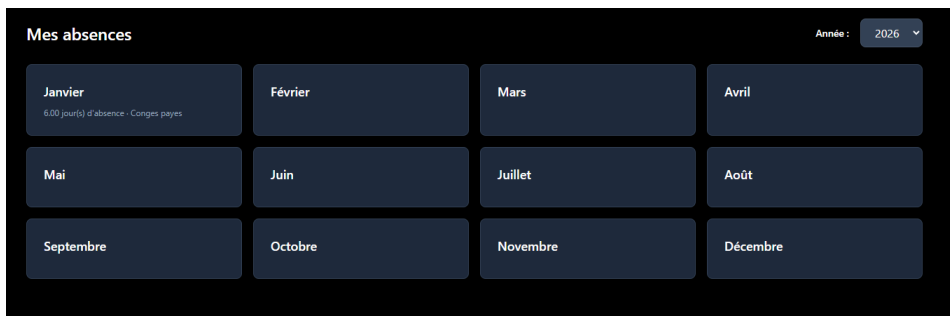


Une fois connecté, le collaborateur accède à son tableau de bord.

Cette interface lui permet d'avoir une vue d'ensemble sur ses compteurs (heures travaillées, solde de congés) et de naviguer facilement vers les formulaires de saisie.



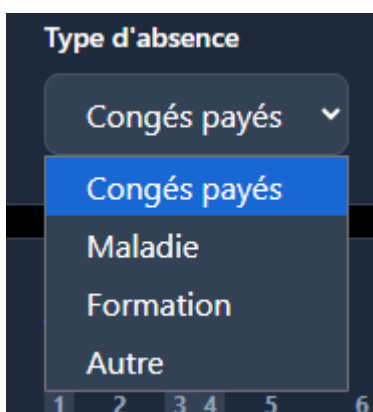
### 3.3. Espace Collaborateur : Saisie d'une demande d'absence



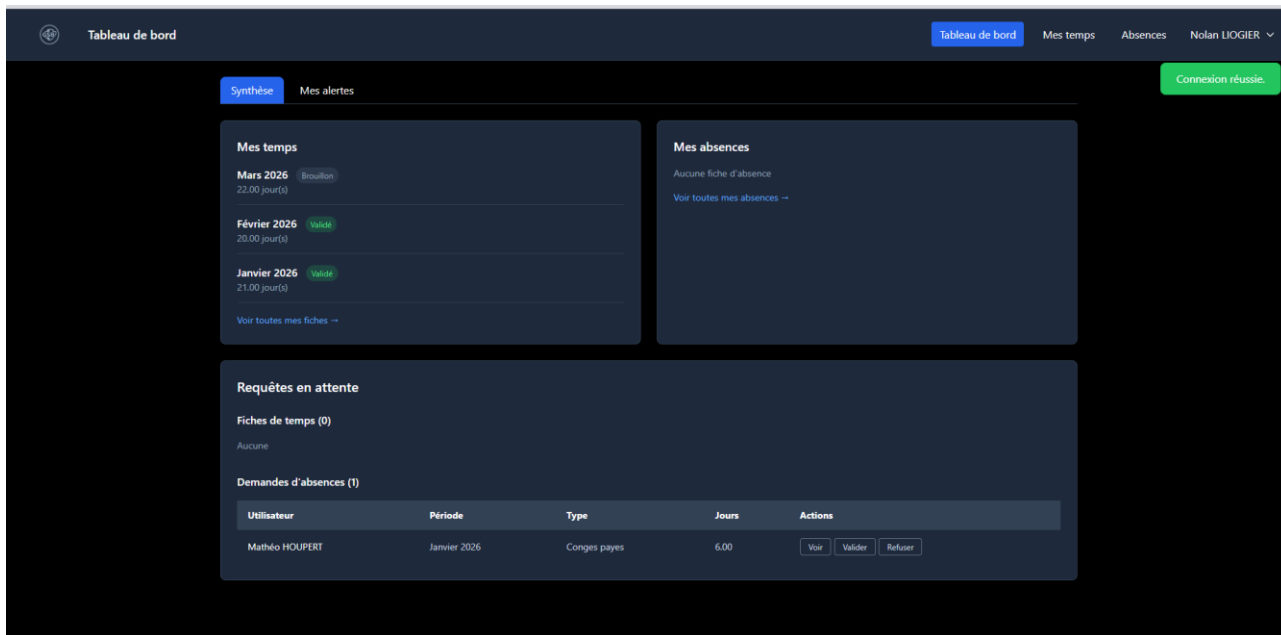
Pour déclarer une absence, l'employé remplit un formulaire en précisant les dates de début et de fin, ainsi que le motif (Congé payé, RTT, Maladie). Une fois soumise, la demande est enregistrée en base de données avec le statut "En attente" par défaut.

1 étant jour remplie, 0,5 demi-journée et 0 journée non remplie.

Donc dans l'onglet congés, pour crée une demande de congé on sélectionne le jour en question et on remplit 1 après avoir choisir le type d'absence.



### 3.4. Espace Manager : Le Workflow de validation



Utilisateur	Période	Type	Jours	Actions
Mathéo HOUPERT	Janvier 2026	Conges payés	6.00	<a href="#">Voir</a> <a href="#">Valider</a> <a href="#">Refuser</a>

L'utilisateur disposant des droits "Manager" possède une vue globale sur les demandes de son équipe.

Il a la capacité de traiter les demandes "En attente" en cliquant sur les actions de validation ou de refus.

Cette action met instantanément à jour le statut de la demande dans la base de données, informant ainsi l'employé de la décision.



### 3.5. Espace Administrateur : Gestion des utilisateurs

ID	NOM	EMAIL	RÔLE	DATE DE CRÉATION	ACTIONS
4	Léa SOURMAIL	lea.sourmail@gsb.fr	Administrateur	2026-03-04 10:51:23	Voir Modifier Supprimer
1	Eren AKYUREK	eren.akyurek@gsb.fr	Administrateur	2026-03-02 15:13:25	Voir Modifier Supprimer
8	Stella DURRAY	stella.durray@gsb.fr	Ressources Humaines	2026-03-05 16:17:02	Voir Modifier Supprimer
5	Nolan LIOGIER	nolan.liogier@gsb.fr	Ressources Humaines	2026-03-04 10:52:48	Voir Modifier Supprimer
2	Noa VIOLLET	noa.viollet@gsb.fr	Ressources Humaines	2026-03-02 15:26:01	Voir Modifier Supprimer
9	Léonard DEMON	leonard.demon@gsb.fr	Ressources Humaines	2026-03-05 16:18:10	Voir Modifier Supprimer
7	Jules ROLAND	jules.roland@gsb.fr	Ressources Humaines	2026-03-05 16:16:16	Voir Modifier Supprimer
16	Nicolas BOURGOIN	nicolas.bourgoin@gsb.fr	Utilisateur	2026-03-05 16:25:02	Voir Modifier Supprimer
13	Nazaret LETOCARD	nazaret.letocard@gsb.fr	Utilisateur	2026-03-05 16:22:57	Voir Modifier Supprimer
6	Mathéo HOUPERT	matheo.houpert@gsb.fr	Utilisateur	2026-03-04 10:53:19	Voir Modifier Supprimer

L'application intègre un panneau d'administration réservé aux utilisateurs possédant le rôle "Admin".

Depuis cet espace, l'administrateur a la charge de gérer l'ensemble des collaborateurs du laboratoire GSB.

Il dispose d'une interface complète (CRUD) lui permettant de créer de nouveaux comptes, de mettre à jour les informations existantes (comme le changement de rôle d'un employé vers manager), ou de supprimer le compte d'un collaborateur ayant quitté l'entreprise.

Utilisateurs > créer

#### Créer un utilisateur

Nom

Email

Mot de passe

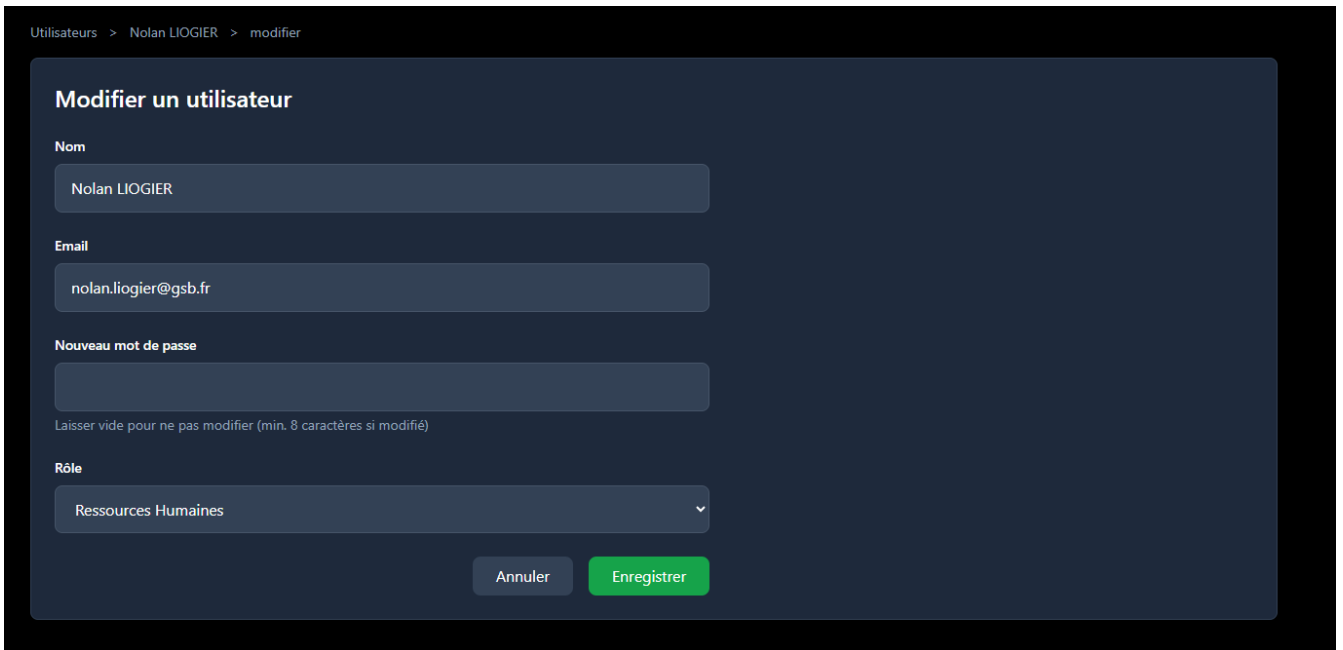
Minimum 8 caractères

Rôle

Utilisateur

Annuler Créer

En tant qu'admin on peut également ajouter un utilisateur grâce à cette interface



The screenshot shows a web interface for modifying a user. At the top, there is a breadcrumb trail: "Utilisateurs > Nolan LIOGIER > modifier". The main heading is "Modifier un utilisateur". Below this, there are several input fields: "Nom" with the value "Nolan LIOGIER", "Email" with the value "nolan.liogier@gsb.fr", and "Nouveau mot de passe" which is currently empty. A note below the password field states "Laisser vide pour ne pas modifier (min. 8 caractères si modifié)". There is also a "Rôle" dropdown menu currently set to "Ressources Humaines". At the bottom right, there are two buttons: "Annuler" and "Enregistrer".

On peut également changer les informations d'un utilisateur déjà enregistré.

## 4. Réalisation Technique et Extraits de Code

### 4.1. Sécurité et Authentification des utilisateurs

Analyse technique de l'authentification : Ce fragment de code, issu du contrôleur AuthController, illustre la mise en œuvre de plusieurs bonnes pratiques de sécurité :

1. Protection contre les attaques CSRF : La méthode validateCSRF() s'assure que la requête POST provient bien de notre formulaire et non d'un site tiers malveillant.
2. Hachage des mots de passe : L'application ne stocke aucun mot de passe en clair. L'authentification utilise la fonction native PHP password\_verify() qui compare le mot de passe saisi par l'utilisateur avec le hash sécurisé (Argon2ID) stocké dans la base de données.
3. Gestion des sessions (Fixation de session) : En cas de succès, la méthode loginUser() initialise les variables de session (\$\_SESSION) et utilise session\_regenerate\_id(true) pour créer un nouvel ID de session. Cela permet de se prémunir contre les attaques par fixation de session.

```
49     $this->validateCSRF('auth/login');
50
51     $email = trim((string) ($_POST['email'] ?? ''));
52     $password = (string) ($_POST['password'] ?? '');
53
54     if ($email === '' || $password === '') {
55         $_SESSION['login_error'] = 'Veuillez remplir tous les champs.';
56         $this->redirect('auth/login');
57     }
58     if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
59         $_SESSION['login_error'] = 'Format d\'email invalide.';
60         $this->redirect('auth/login');
61     }
62
63     $model = new UserModel($this->pdo);
64     $user = $model->findByEmail($email);
65     if (!$user || !password_verify($password, (string) $user['password'])) {
66         $_SESSION['login_error'] = 'Email ou mot de passe incorrect.';
67         $this->redirect('auth/login');
68     }
69
```

## 4.2. Logique métier : Le Workflow de validation d'un congé par le Manager

Contrôle des habilitations (Contrôleur) :

Avant tout traitement, le système vérifie le rôle de l'utilisateur (RH ou Admin) et valide le jeton CSRF.

Application des règles de gestion (Service) :

Le système fait appel au ValidationService pour appliquer une règle métier stricte : la séparation des tâches.

Un manager ne peut en aucun cas valider sa propre demande d'absence.

Persistance et Sécurité (Modèle) :

Une fois le nouveau statut déterminé (Validé ou Refusé avec commentaire), les données sont envoyées au modèle AbsenceSheetModel.

La mise à jour en base de données s'effectue via une requête SQL UPDATE utilisant des requêtes préparées (PDO), garantissant ainsi une protection totale contre les failles par injection SQL.

```
$check = $this->validationService->checkValidationPermission($validatorUserId, $submitterUserId, $validatorRole);
if (!$check['allowed']) {
    $_SESSION['error_message'] = $check['message'];
    $this->redirect('dashboard/index');
}
$status = $action === 'approve' ? 'approved' : 'rejected';
$updateData = ['status' => $status];
if ($action === 'reject') {
    $updateData['rejection_comment'] = trim((string) ($_POST['rejection_comment'] ?? ''));
}
$this->model->update($id, $updateData);
$_SESSION['success_message'] = $action === 'approve' ? 'Fiche validée.' : 'Fiche refusée.';
$this->redirect('dashboard/index');
}
```

```
public function update(int $id, array $data): bool
{
    $allowed = ['type', 'data', 'status', 'total_days', 'rejection_comment'];
    $fields = [];
    $params = ['id' => $id];
    foreach ($allowed as $field) {
        if (!array_key_exists($field, $data)) {
            continue;
        }
        $fields[] = "{$field} = :{$field}";
        if ($field === 'data' && is_array($data[$field])) {
            $params[$field] = json_encode($data[$field]);
        } else {
            $params[$field] = $data[$field];
        }
    }
    if ($fields === []) {
        return false;
    }
    $sql = 'UPDATE absence_sheets SET ' . implode(', ', $fields) . ' WHERE id = :id';
    $stmt = $this->pdo->prepare($sql);
    return $stmt->execute($params);
}
```

### 4.3. Interaction avec la base de données : Saisie d'une absence

Analyse technique de l'insertion en base de données : La soumission d'une demande d'absence par un collaborateur démontre une gestion optimisée des enregistrements.

#### 1. Logique conditionnelle (Contrôleur) :

Plutôt que d'insérer aveuglément de nouvelles données, le contrôleur vérifie au préalable si une fiche d'absence existe déjà pour l'utilisateur sur le mois et l'année concernés.

Si c'est le cas, il met à jour la fiche existante. Sinon, il déclenche la création d'une nouvelle entrée.

#### 2. Requête sécurisée (Modèle) :

L'insertion s'effectue via la méthode create() du modèle.

On y retrouve l'utilisation de requêtes préparées (PDO) pour contrer les injections SQL.

#### 3. Flexibilité des données :

On note l'utilisation pertinente de la fonction json\_encode() pour stocker les détails spécifiques des jours d'absence (le champ data) sous forme de chaîne JSON, ce qui offre une grande flexibilité à la structure de la base de données.

```
if ($existing) {
    $success = $this->model->update($existing['id'], $payload);
} else {
    $success = $this->model->create([
        'user_id' => $userId,
        'month' => $month,
        'year' => $year,
        'type' => $type,
        'data' => $data,
        'total_days' => $totalDays,
        'status' => $status
    ]);
}
```

```
public function create(array $data): bool
{
    $sql = 'INSERT INTO absence_sheets (user_id, month, year, type, data, status, total_days) VALUES (:user_id, :month, :year, :type, :data, :status, :total_days)';
    $stmt = $this->pdo->prepare($sql);
    return $stmt->execute([
        'user_id' => $data['user_id'],
        'month' => $data['month'],
        'year' => $data['year'],
        'type' => $data['type'] ?? 'conges_payes',
        'data' => is_array($data['data'] ?? []) ? json_encode($data['data']) : (string) ($data['data'] ?? ''),
        'status' => $data['status'] ?? 'draft',
        'total_days' => $data['total_days'] ?? 0
    ]);
}
```

## 5. Phase de Tests et Recettage

### 5.1. Méthodologie de tests

Afin de garantir la fiabilité et la sécurité de l'application "TimeTracker", une campagne de tests fonctionnels et techniques a été menée avant le déploiement.

La méthodologie adoptée repose principalement sur des tests manuels de type "boîte noire" (validation des cas d'utilisation depuis l'interface) ainsi que sur des tests de sécurité (vérification des failles courantes).

Pour chaque fonctionnalité critique, nous avons testé :

- **Le chemin nominal (Happy Path)** : Le comportement attendu lorsque l'utilisateur effectue l'action correctement.
- **Les chemins d'erreur** : La réaction du système face à des saisies incorrectes, des tentatives d'accès non autorisées ou des champs obligatoires manquants.

## 5.2. Tableaux des tests fonctionnels

Les tableaux ci-dessous répertorient un échantillon des scénarios de tests exécutés lors de la phase de recette.

N°	Scénario de test	Préconditions	Action réalisée	Résultat attendu	Résultat obtenu	Statut
1.1	Connexion réussie	L'utilisateur possède un compte valide	Saisie de l'email et du mot de passe corrects, puis clic sur "Connexion"	Redirection vers le tableau de bord, initialisation de la session (\$_SESSION)	Redirection OK. Session créée avec le rôle correspondant	<b>Succès</b> ☑
1.2	Mot de passe incorrect	L'utilisateur possède un compte valide	Saisie de l'email correct mais d'un mauvais mot de passe	Refus de connexion, affichage du message "Email ou mot de passe incorrect"	Message d'erreur affiché. Aucune session créée	<b>Succès</b> ☑
1.3	Format d'email invalide	Aucune	Saisie de "jean.dupont@gsb" (sans le .fr)	Le système bloque la soumission et affiche "Format d'email invalide"	Blocage par la fonction filter_var de PHP	<b>Succès</b> ☑
1.4	Accès page protégée sans session	L'utilisateur n'est pas connecté	Tentative d'accès direct à l'URL ?r=dashboard/index	Redirection forcée vers la page de login	Redirection immédiate par le BaseController	<b>Succès</b> ☑

N°	Scénario de test	Préconditions	Action réalisée	Résultat attendu	Résultat obtenu	Statut
2.1	Soumission d'une demande valide	Connecté en tant qu'Employé	Remplissage du formulaire (Dates, Motif : Congés Payés) et clic sur Envoyer	Insertion en BDD. Le statut passe à "En attente". Message flash de succès	Requête SQL INSERT réussie. Affichage dans l'historique	Succès ✓
2.2	Soumission d'une demande vide	Connecté en tant qu'Employé	Clic sur "Envoyer" sans remplir les dates	Rejet du formulaire, obligation de remplir les champs requis	Blocage HTML5 (required) et vérification Back-End	Succès ✓
2.3	Modification d'un brouillon	L'employé a une demande au statut "Draft"	Clic sur "Sauvegarder" après modification des dates	Mise à jour de la fiche existante sans créer de doublon	Déclenchement de la méthode update() au lieu de create()	Succès ✓

N°	Scénario de test	Préconditions	Action réalisée	Résultat attendu	Résultat obtenu	Statut
3.1	Validation d'une absence par le RH	Connecté en tant que Manager/RH	Clic sur le bouton "Valider" d'une demande "En attente"	Mise à jour du statut en "Approved" dans la base de données	Statut mis à jour via AbsenceSheetModel. Disparition de la liste "À traiter"	Succès ✓
3.2	Refus avec commentaire	Connecté en tant que Manager/RH	Clic sur "Refuser", saisie du motif "Effectif insuffisant"	Mise à jour du statut en "Rejected" et enregistrement du commentaire	Statut à jour. L'employé voit le motif de refus sur son tableau de bord	Succès ✓
3.3	Séparation des tâches (Auto-validation)	Le Manager soumet sa propre demande de congé	Le manager tente de valider sa propre demande de congé	Le système bloque l'action. Message d'erreur d'habilitation	Le ValidationService intercepte l'action et retourne une erreur	Succès ✓

## 6. Bilan et Perspectives

### 6.1. Bilan de la réalisation

Le développement de l'application web "TimeTracker GSB" a permis de répondre avec succès au cahier des charges initial. L'application offre une plateforme centralisée, sécurisée et ergonomique pour la gestion des temps et des absences. D'un point de vue technique, ce projet a été l'occasion de mettre en pratique et d'approfondir mes compétences en :

- **Architecture logicielle :**  
Mise en place d'un modèle MVC strict permettant une séparation claire entre la logique métier, l'accès aux données et l'affichage.
- **Sécurité des applications web :**  
Implémentation de mécanismes de protection contre les failles courantes (injections SQL via requêtes préparées, attaques CSRF, hachage Argon2ID des mots de passe).
- **Base de données :**  
Conception d'un modèle relationnel complet et manipulation des données via l'extension PDO.

### 6.2. Perspectives d'évolution

Bien que l'application réponde aux besoins actuels, plusieurs évolutions pourraient être envisagées pour les futures versions :

- **Génération de rapports :**  
Intégration d'une fonctionnalité d'export (PDF ou Excel) des feuilles de temps mensuelles pour faciliter le travail du service comptabilité.
- **Notifications :**  
Mise en place d'un système d'envoi d'emails automatiques pour alerter un employé lorsque sa demande de congé a été validée ou refusée par son manager.
- **Responsive Avancé :**  
Optimisation poussée de l'interface pour une utilisation sur tablette, voire le développement d'une application mobile compagne (actuellement en cours de développement avec le projet StockTracker).

## Accès :

Mails : [prenom.nom@gsb.fr](mailto:prenom.nom@gsb.fr)

Mot de passe en fonction des rôles :

User – PrenomUser123!

RH – PrenomRH123!

Admin – PrenomAdmin123!

Exemple pour vous connecter en tant qu'administrateur :

- email : [eren.akyurek@gsb.fr](mailto:eren.akyurek@gsb.fr)
- password : ErenAdmin123!

Exemple pour vous connecter en tant que RH :

- email : [nolan.liogier@gsb.fr](mailto:nolan.liogier@gsb.fr)
- password : NolanRH123!

Pour vous connecter en tant qu'utilisateur

- email : [matheo.houpert@gsb.fr](mailto:matheo.houpert@gsb.fr)
- password : MatheoUser123!

## Liste des comptes :

### - Admins (2) :

Eren AKYUREK  
Léa SOURMAIL

### - RH (5) :

Noa VIOLLET  
Nolan LIOGIER  
Stella DURRAY  
Léonnard DEMON  
Jules ROLAND

### - Users (10) :

Mathéo HOUPERT  
Léon BAGHDASSARIAN  
Nazaret LETOCARD  
Alyssa STARK  
Connor LISSARD  
Elian BALESTRO  
George HIDALGO  
Jeanne KIOSK  
Manuel MEZINO  
Nicolas BOURGOIN