**Compte rendue**

**Projet TER**



Ce compte rendu est le fruit

du travail des membres suivants :

**SOMMAIRE**

1. **Introduction**

* **Présentation du projet et adoption du modèle MVC.**

1. **Développement**

* **Analyse globale des fichiers clés et leur contribution au projet.**

1. **Base de Donnée**

* **Exeplication de la conseption**

1. **Répertoire VIEWS**

* **Aperçu des fichiers et leur impact sur l'interface utilisateur.**

1. **Répertoire Controllers**

* **Description des contrôleurs et de leur rôle**

1. **Répertoire Public**

* **Vue d'ensemble de la validation des données et des styles visuels.**

# **1) Introduction**

Le projet abordé dans ce rapport concerne la gestion de post-it en mode multi-utilisateurs, visant à offrir une plateforme interactive qui permet aux utilisateurs de créer, éditer, partager et consulter des post-it de façon collaborative. Ce système s'appuie sur le modèle architectural MVC (Modèle-Vue-Contrôleur), choisi pour structurer le code de manière efficace et maintenable, facilitant ainsi le développement et l'évolution de l'application.

Le modèle MVC est un paradigme de conception largement adopté dans le développement d'applications web. Il segmente l'application en trois composants interconnectés : le Modèle, qui gère les données et la logique métier ; la Vue, qui s'occupe de l'affichage et de l'interaction utilisateur ; et le Contrôleur, qui réagit aux entrées des utilisateurs, communique avec le modèle et met à jour la vue. Cette structuration permet de séparer les différentes préoccupations du code, favorisant ainsi sa clarté, sa réutilisabilité et sa maintenabilité.

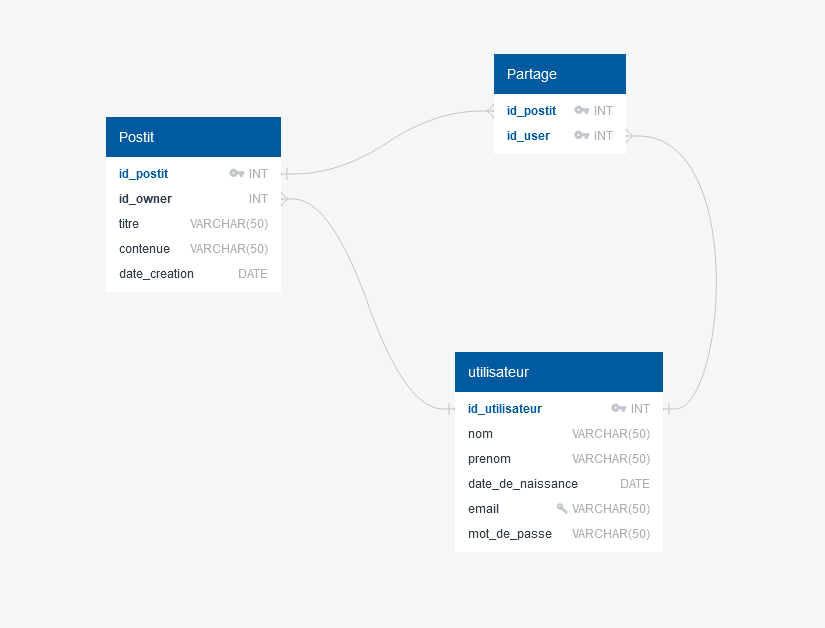
Dans le contexte de ce projet, le modèle MVC permet de clarifier la répartition des responsabilités au sein de l'application, assurant ainsi une organisation cohérente et une meilleure collaboration entre les développeurs. Ce compte rendu se propose d'analyser en détail l'impact de chaque composant MVC sur la structure et la fonctionnalité du projet de gestion de post-it.

# **2) Développement**

La section Développement du compte rendu se concentre sur l'analyse approfondie des fichiers essentiels situés à la racine du projet, mettant en exergue leur conception robuste et la manière dont ils orchestrent les fonctionnalités clés de l'application.

1. Le fichier **install.php** joue un rôle fondamental dans la mise en place de l'application. Il est responsable de l'initialisation et de la configuration de la base de données, une étape cruciale pour assurer le bon fonctionnement du système. Ce fichier prend en charge la création de la base de données si elle n'existe pas, établit une connexion sécurisée avec celle-ci et définit la structure des tables nécessaires, comme les tables utilisateurs, post-it et partage, garantissant ainsi l'intégrité et la cohérence des données.
2. Le fichier **index.php** sert de point d'entrée principal de l'application, illustrant son architecture modulaire et sa capacité à gérer dynamiquement les flux d'interactions. Il initialise une session PHP, vérifie l'état de connexion de l'utilisateur et oriente les requêtes vers les contrôleurs et les vues appropriés, facilitant ainsi une navigation fluide et intuitive. De plus, il gère différentes actions utilisateur telles que l'ajout, la modification et la suppression de post-it, contribuant à une expérience utilisateur cohérente et agréable.
3. Le fichier **config.php** est déterminant pour la configuration de l'application, centralisant les paramètres de connexion à la base de données. Sa structure modulaire et son accessibilité en font un élément clé pour l'organisation et la maintenance des informations de configuration, soutenant ainsi l'efficacité et la sécurité de l'application.

# **3) Base de données**



## Table utilisateurs

### Description :

Cette table stocke les informations des utilisateurs du site.

### Colonnes :

* **id\_utilisateur** (clé primaire) (int(11)) : Identifiant unique de l'utilisateur.
* **nom** (VARCHAR(100)): Nom de l'utilisateur.
* **prenom** (VARCHAR(255)): Prénom de l'utilisateur.
* **date\_de\_naissance** (date): Date de naissance de l'utilisateur.
* **email** (VARCHAR(100)): Adresse email de l'utilisateur.
* **mot\_de\_passe** (VARCHAR(255)): Mot de passe haché de l'utilisateur.

## Table postit

### Description :

Cette table stocke les informations des post-its.

### Colonnes :

* **id\_postit** (clé primaire) (int(11)) : Identifiant unique du post-it.
* **id\_owner** (int(11)): Identifiant de l'utilisateur propriétaire du post-it (clé étrangère référençant la table utilisateurs).
* **titre** (VARCHAR(100)): Titre du post-it.
* **contenu** (VARCHAR(255)):): Contenu du post-it.
* **date\_creation** (DATETIME): Date de création du post-it (sert également de date de modification).

### Remarques :

* La clé étrangère id\_owner permet de relier chaque post-it à son propriétaire.
* La date de création est également utilisée comme date de modification.

## Table partage

### Description :

Cette table gère les partages de post-its entre utilisateurs.

### Colonnes :

* **id\_postit** (clé primaire composée, référence la table postit): Identifiant du post-it partagé.
* **id\_utilisateur** (clé primaire composée, référence la table utilisateurs): Identifiant de l'utilisateur avec lequel le post-it est partagé.

### Remarques :

Une ligne dans la table partage représente un partage d'un post-it avec un utilisateur spécifique.

### Exemple :

Si l'utilisateur U1 souhaite partager son post-it P1 avec les utilisateurs U2 et U3, la table partage contiendra les lignes suivantes :

* (P1, U2)
* (P1, U3)

# **4) Répertoire VIEWS**

La section Répertoire VIEWS du compte rendu détaille les différents fichiers inclus dans ce dossier et leur contribution spécifique à l'interface utilisateur de l'application de gestion de post-it multi-utilisateurs.

1. **accueil.php** : Ce fichier est central pour offrir une vue d'ensemble des post-its à l'utilisateur. Il organise l'affichage des post-its propres à l'utilisateur et ceux partagés, intégrant des fonctionnalités pour les modifier ou les supprimer. Il fournit également des liens pour l'ajout de nouveaux post-its, facilitant l'interaction de l'utilisateur avec l'application.
2. **ajoutpostits.php** : Conçu pour simplifier l'ajout de nouveaux post-its, ce fichier présente un formulaire intuitif qui guide l'utilisateur à travers les différentes étapes d'ajout, y compris la sélection des utilisateurs avec lesquels partager le post-it. La validation côté client est mise en place pour améliorer l'expérience utilisateur et minimiser les erreurs.
3. **connexion.php** : Ce fichier met en avant un formulaire de connexion minimaliste, gérant l'authentification des utilisateurs et offrant un accès direct aux pages d'inscription. La gestion des messages d'état et des erreurs contribue à une expérience utilisateur claire et sans frustration.
4. **error.php** : Servant à informer les utilisateurs des erreurs rencontrées, ce fichier assure la communication des problèmes de manière simple et directe, permettant aux utilisateurs de revenir facilement à la page d'accueil.
5. **inscription.php** : Ce fichier est dédié au processus d'inscription, offrant un formulaire détaillé pour recueillir les informations des nouveaux utilisateurs. Il oriente les utilisateurs à travers chaque étape de l'inscription, assurant une intégration fluide au système.
6. **modifierpostit.php** : Permettant la modification des post-its existants, ce fichier affiche un formulaire pré-rempli avec les détails actuels du post-it, offrant la possibilité de changer le titre, le contenu et les utilisateurs partagés.
7. **visualiserPostit.php** : Ce fichier fournit une interface pour examiner en détail les post-its, mettant en avant les informations clés et distinguant clairement les post-its personnels des partagés.

# **5) Répertoire Controllers**

La section Controllers du compte rendu décrit le rôle crucial que joue ce répertoire dans la gestion de la logique métier de l'application, orchestrant les interactions entre les utilisateurs, la base de données et l'interface utilisateur.

1. **accueilController.php** : Ce fichier est essentiel pour la gestion de la page d'accueil, offrant des fonctions pour récupérer les post-its détenus par l'utilisateur et ceux qui sont partagés avec lui. Il emploie des requêtes SQL optimisées pour extraire ces informations rapidement et avec précision, contribuant ainsi à une expérience utilisateur fluide et personnalisée.
2. **inscriptionController.php** : Ce contrôleur gère l'inscription des nouveaux utilisateurs, en s'assurant de la validité et de l'unicité des données fournies. Il traite et stocke soigneusement les informations dans la base de données, jouant un rôle fondamental dans l'expansion et la gestion de la communauté d'utilisateurs.
3. **loginController.php** : Ce fichier supervise l'authentification des utilisateurs, validant leurs informations de connexion et établissant des sessions sécurisées. Il est déterminant pour maintenir la sécurité et l'accès approprié aux fonctionnalités de l'application.
4. **postitController.php** : Ce contrôleur est au cœur de la gestion des post-its, orchestrant leur création, modification, visualisation et suppression. Il assure la persistance des données de post-its et gère les interactions complexes, comme le partage entre utilisateurs, en veillant à la cohérence et à la sécurité des opérations.

# **6) Répertoire Public**

La section Répertoire Public du compte rendu met en lumière les fichiers clés qui influencent directement l'expérience utilisateur et la fonctionnalité de l'application de gestion de post-it.

1. **validation-jquery.js** : Ce fichier joue un rôle critique dans la validation des données utilisateur sur les formulaires d'inscription et de connexion, garantissant ainsi que les informations soumises respectent les exigences spécifiées. Il aide à prévenir les erreurs de saisie et renforce la sécurité de l'application en validant les formats d'adresse e-mail et de mot de passe.
2. **script-recherche.js** : Ce script améliore l'expérience utilisateur en facilitant la recherche dynamique au sein de l'application. Il permet une récupération rapide et efficace des informations recherchées, optimisant l'interaction des utilisateurs avec la base de données et améliorant l'accessibilité des fonctionnalités de partage et de gestion des post-it.
3. **style.css et styles.css** : Ces fichiers CSS sont essentiels pour définir l'apparence visuelle de l'application, assurant une présentation cohérente et attrayante à travers les différentes pages. Ils contribuent à l'ergonomie et à l'esthétique de l'interface utilisateur, en définissant des styles pour le positionnement, la couleur et la mise en forme des éléments, ce qui favorise une expérience utilisateur agréable et engageante.