

Roomia



Dufour Ezéchiél, Delattre Gauthier, Penot Natys,
Vanneufville Ian

Le projet étant trop lourd pour être rendu sur Junia Learning et Teams
voici un lien GitHub pour consulter le fichier :

https://github.com/DelattreGauthier/G01_Dufour_Vanneufville_Delattre_Penot

1) Thématique :

Notre idée était de créer un site web utile en répondant à un besoin concret rencontré par la communauté de Junia. Nous avons donc développé Roomia, une plateforme complète de gestion et de réservation de salles.

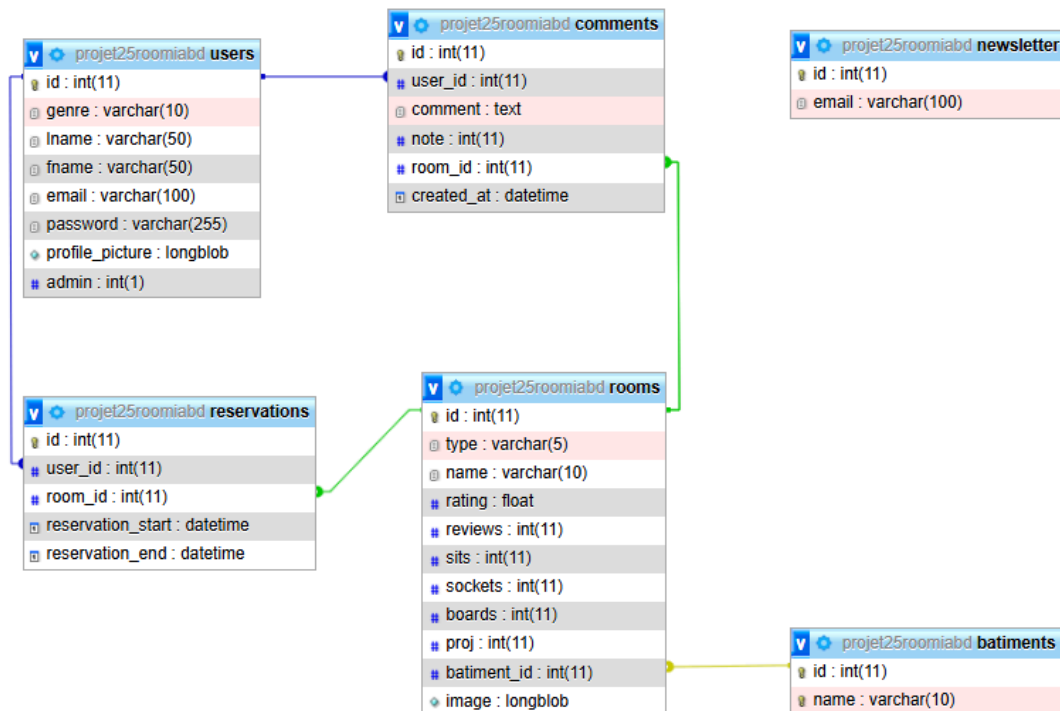
Le problème initial était simple : il est souvent difficile et chronophage de trouver une salle libre pour travailler, se réunir ou se détendre. Il fallait parfois parcourir de nombreux bâtiments et entrer dans plusieurs salles déjà occupées, ce qui dérangeait les personnes présentes et faisait perdre du temps à tout le monde.

Roomia apporte une solution moderne et collaborative à ce problème :

- Le site permet de visualiser en temps réel la disponibilité des salles et amphithéâtres dans tous les bâtiments de Junia.
- Les utilisateurs peuvent réserver une salle, annuler une réservation, ou consulter l'historique des réservations.
- Chaque salle dispose d'une fiche détaillée avec ses équipements, ses horaires, des photos, et les commentaires/notes laissés par la communauté.
- Un système d'authentification permet à chacun de gérer son profil, ses réservations, et de personnaliser son expérience (photo de profil, préférences...).
- Les administrateurs disposent d'un panel dédié pour gérer les utilisateurs, les salles, les réservations et modérer les contenus.
- Roomia propose également une inscription à une newsletter, une gestion transparente des cookies, et une interface responsive adaptée à tous les appareils.

L'objectif de Roomia est donc de simplifier la recherche de salles disponibles, d'optimiser l'utilisation des espaces, et de favoriser la collaboration et le confort au sein de Junia, tout en offrant une expérience utilisateur moderne, sécurisée et personnalisée.

2) Diagramme de classe d'UML :



3) Contenu de chaque page :

connexion_bd.php : Ce fichier gère la connexion entre le site web et la base de données. Il s'agit d'un script de traitement utilisé en arrière-plan par d'autres pages ; il n'est donc pas accessible directement via l'URL.

log_out.php : Script de traitement permettant à un utilisateur de se déconnecter du site. Il est uniquement accessible via le bouton "Se déconnecter" présent sur la page profil.php.

ajouter_salle.php : Page contenant un formulaire permettant aux administrateurs d'ajouter une nouvelle salle à la liste existante. Elle est accessible uniquement depuis la page panel_admin.php.

annulation_reservation.php : Script de traitement permettant l'annulation d'une réservation. Il n'est accessible que depuis la page salle.php, à condition qu'une réservation ait déjà été effectuée et que l'utilisateur clique sur le bouton "Annuler".

batiment.php : Page qui affiche toutes les salles et amphithéâtres d'un bâtiment donné (via l'ID passé en GET). Elle liste les salles avec leur image, leurs horaires de disponibilité, et permet d'accéder à la page de chaque salle. Accessible depuis la page batiments.php.

batiments.php : Page listant les différents bâtiments disponibles (IC1, IC2, ALG, NS) sous forme de liens avec images. Permet de choisir un bâtiment pour voir ses salles. Accessible depuis le menu principal.

connexion.php : Page de connexion pour les utilisateurs. Elle vérifie les identifiants, gère les erreurs, et crée la session utilisateur si la connexion réussit. Accessible depuis le menu ou lors d'une action nécessitant d'être connecté.

html/cookies.php : Page d'information sur la politique de cookies du site Roomia. Explique les types de cookies utilisés et la gestion du consentement. Accessible via un lien dans le bandeau cookies ou le footer.

php/cookies.php : Script inclus dans les pages pour gérer l'affichage du bandeau de consentement aux cookies et le traitement de l'acceptation. Utilisé en arrière-plan, pas destiné à être accédé directement.

inscription.php : Page d'inscription pour créer un nouveau compte utilisateur. Vérifie la validité des champs, gère l'upload de la photo de profil, et ajoute l'utilisateur à la base de données. Accessible depuis la page de connexion.

newsletter.php : Page permettant aux utilisateurs de s'inscrire à la newsletter de Roomia en saisissant leur email. Vérifie la validité et l'unicité de l'email avant de l'ajouter à la base. Accessible via le footer.

panel_admin.php : Panel d'administration réservé aux utilisateurs admin. Permet de visualiser, modifier, supprimer ou ajouter des utilisateurs, salles, réservations, commentaires, etc. Offre aussi un champ pour exécuter des requêtes SQL manuelles.

profil.php : Page de profil de l'utilisateur connecté. Affiche ses informations, sa photo de profil (modifiable), ses réservations (avec possibilité de suppression), ses commentaires, et un bouton de déconnexion. Accessible uniquement si connecté.

salle.php : Page détaillée d'une salle ou d'un amphithéâtre. Affiche ses infos, ses horaires de réservation (avec possibilité de réserver/annuler), l'historique des réservations, et les commentaires des utilisateurs. Accessible depuis batiment.php ou via un lien direct.

user.php : Page publique affichant le profil d'un utilisateur (nom, photo), son historique de réservations et ses commentaires. Accessible en cliquant sur le nom ou la photo d'un utilisateur depuis la page d'une salle.

envoi_commentaire.php : Script de traitement pour l'envoi d'un commentaire/note sur une salle. Vérifie les champs, ajoute le commentaire à la base, met à jour la note moyenne de la salle, puis redirige vers la page de la salle. Accessible uniquement via le formulaire de commentaire.

fonctions.php : Fichier utilitaire contenant toutes les fonctions PHP réutilisables du projet (nettoyage des données, validation, génération des horaires, formatage des dates, gestion des cookies, etc.). Inclus dans la plupart des pages PHP.

footer.php : Fichier inclus affichant le pied de page du site, la gestion des cookies, et le formulaire d'inscription à la newsletter. Utilisé sur toutes les pages principales.

getSalleImg.php : Script qui sert l'image d'une salle depuis la base de données (format binaire), en détectant le type MIME. Utilisé pour afficher dynamiquement les images des salles.

getUserAvatar.php : Script qui sert la photo de profil d'un utilisateur depuis la base de données, en détectant le type MIME. Utilisé pour afficher dynamiquement les avatars.

header.php : Fichier inclus affichant l'en-tête du site : logo, menu de navigation, bouton de connexion/profil, menu déroulant responsive. Utilisé sur toutes les pages principales.

modifier.php : Page permettant à un administrateur de modifier les informations d'une salle existante (nom, équipements, image, etc.). Accessible uniquement depuis le panel admin.

picture.php : Script qui sert la photo de profil de l'utilisateur connecté (pour l'affichage dans le header ou le profil). Utilisé en interne, pas destiné à être accédé directement.

remplirBDD.php : Script utilitaire (désactivé/commenté) pour remplir la base de données avec des salles et amphithéâtres fictifs lors du développement. Non utilisé en production.

reservation.php : Script de traitement pour effectuer une réservation de salle. Vérifie les règles métier (plages horaires, durée, conflits, etc.), ajoute la réservation à la base, et gère les messages de succès/erreur. Accessible uniquement via le formulaire de réservation.

supprimer.php : Script de traitement pour supprimer un élément (utilisateur, salle, réservation, commentaire, etc.) de la base de données. Accessible uniquement via le panel admin ou le profil utilisateur.

styles.php : Fichier CSS dynamique généré en PHP, qui adapte les couleurs et styles du site selon le thème choisi (clair/sombre). Inclus dans toutes les pages principales.

update_profile_picture.php : Script de traitement permettant à l'utilisateur de mettre à jour sa photo de profil depuis la page profil.php. Gère l'upload, la validation et la mise à jour en base.

index.php : Page d'accueil du site Roomia. Présente le projet, l'équipe, les fonctionnalités, et propose des liens vers les bâtiments, la connexion, la newsletter, etc.

4) Utilisation du PHP dans notre site :

Le site Roomia exploite pleinement les capacités du langage PHP pour assurer l'ensemble de ses fonctionnalités côté serveur. PHP est utilisé pour établir la connexion avec la base de données MySQL, permettant ainsi de stocker et de récupérer toutes les informations essentielles : utilisateurs, salles, réservations, commentaires, images, etc. Chaque action réalisée par l'utilisateur (connexion, inscription, réservation, ajout ou modification de salle, envoi de commentaire, gestion du profil, etc.) est traitée par des scripts PHP dédiés qui valident les données reçues, appliquent les règles métier (par exemple, éviter les conflits de réservation ou vérifier les droits d'accès), et effectuent les opérations nécessaires en base.

PHP gère également la sécurité du site, notamment à travers la gestion des sessions pour l'authentification, la vérification des rôles (utilisateur ou administrateur), la protection contre les accès non autorisés à certaines pages ou scripts, et la validation/sanitation des entrées utilisateurs pour éviter les failles courantes (injections SQL, XSS...). Les scripts PHP sont aussi responsables de la gestion des fichiers uploadés, comme les photos de profil ou les images des salles, en assurant leur stockage sécurisé et leur affichage dynamique via des scripts spécialisés.

De plus, PHP permet de générer dynamiquement le contenu des pages en fonction du contexte : affichage des salles disponibles en temps réel, personnalisation de l'interface selon l'utilisateur connecté, affichage des réservations, des commentaires, ou encore adaptation des styles selon les préférences. Les fichiers d'inclusion (header, footer, styles dynamiques) facilitent la réutilisation du code et garantissent une cohérence sur l'ensemble du site. Enfin, PHP intervient dans la gestion des cookies, du consentement utilisateur, et dans l'envoi d'emails (par exemple pour la newsletter).

Grâce à cette utilisation complète de PHP, Roomia propose une expérience interactive, fluide et sécurisée, tout en assurant la fiabilité et la cohérence des données et des interactions sur l'ensemble de la plateforme.

5) Répartition des tâches :

Ian : Dans le cadre de ce projet, j'ai apporté une contribution significative à plusieurs éléments clés. J'ai participé à la conception et la mise en place de certains fichiers, en développant plusieurs codes PHP adaptés à tous types d'utilisateurs, incluant des vitesses de traitement fluides et de qualité afin d'améliorer l'UX. J'ai également créé une structure qui a servi de base pour la deuxième partie de ce projet, en veillant à assurer des dépendances cohérentes avec l'ensemble des données.

Par ailleurs, j'ai réalisé les pages des utilisateurs, salles et bâtiments dynamiquement, en m'assurant qu'elles soient à la fois attrayantes mais surtout fonctionnelles. J'ai effectué un important travail sur le traitement des informations des différentes pages, avec un accent particulier sur la page de choix des salles pour garantir une adaptation optimale à toutes heures et tous utilisateurs. J'ai également été responsable de la création des salles et de leur contenu.

En ce qui concerne l'aspect dynamique, j'ai conçu la base de données globale en utilisant une approche via JSON, ce qui a permis une structure claire, flexible et pratique. Ce choix technique a été déterminant pour garantir un dynamisme fluide et une utilisation uniforme.

Natys : Dans le cadre de ce projet, j'ai pris en charge plusieurs développements importants, principalement liés à la gestion des données et à l'interaction avec la base de données.

Tout d'abord, j'ai réalisé l'intégralité des tableaux dynamiques présents dans `panel_admin.php`. Pour cela, j'ai conçu et exécuté des requêtes SQL permettant de récupérer efficacement les données pertinentes depuis la base. J'ai ensuite structuré l'affichage en HTML en intégrant directement les résultats, en veillant à ce que les tableaux soient à la fois lisibles, complets et facilement exploitables par les administrateurs. Chaque entrée des tableaux est associée à des liens dynamiques vers les pages `modifier.php` et `supprimer.php`. Le lien vers `supprimer.php` utilise la méthode GET pour transmettre l'identifiant unique de l'enregistrement à supprimer, garantissant ainsi le bon traitement de la requête.

J'ai développé entièrement les logiques de ces deux pages :

- `modifier.php` permet de récupérer les informations actuelles de l'entrée, les afficher dans un formulaire pré-rempli, et de mettre à jour la base après validation de l'utilisateur.

- supprimer.php gère la suppression sécurisée des enregistrements.

En parallèle, j'ai également participé activement à la création de la page profil.php. Ce travail a nécessité une analyse minutieuse des données disponibles : j'ai rédigé plusieurs requêtes SQL pour récupérer l'ensemble des informations liées aux utilisateurs, puis opéré un tri sélectif pour ne conserver que les données réellement pertinentes à afficher sur leur profil (par exemple, nom, prénom, photo de profil, droits d'accès, historique de réservation, historique de commentaires, etc.). Enfin, j'ai contribué à la fonctionnalité d'affichage des horaires libres ou réservés de chaque salle dans bâtiment.php. Pour cela, j'ai dû extraire les créneaux horaires depuis la base de données, analyser leur statut (libre ou réservé) et concevoir un affichage visuel clair pour l'utilisateur.

Ezéchiél : Au cours de ce projet, j'ai eu l'occasion de participer à l'élaboration de plusieurs fonctionnalités de notre site web. J'ai tout d'abord été responsable de la mise en place des pages de connexion et d'inscription. Cela a nécessité la création de plusieurs règles de validation avancées pour les formulaires, afin de s'assurer que les données saisies respectaient un format précis et sécurisé.

L'objectif était de garantir un premier niveau de sécurité dès la saisie des informations par les utilisateurs. Par la suite, j'ai participé à la conception de la page de profil utilisateur. Mon travail a principalement consisté à développer la fonctionnalité de changement d'image de profil, en gérant aussi bien l'upload sécurisé des fichiers que la mise à jour correspondante dans la base données. Ce développement a nécessité une attention particulière à la sécurité des données transférées ainsi qu'à la bonne gestion des formats de fichiers.

En parallèle, j'ai également contribué à l'élaboration de la newsletter, en permettant aux utilisateurs d'entrer leur adresse mail dans la base de données.

Enfin, bien que l'axe principal du projet ne portait pas sur l'esthétique du site, je me suis investi dans la réalisation du layout de plusieurs nouvelles pages, notamment le panel d'administration. J'ai pris soin d'assurer une certaine cohérence visuelle, afin d'améliorer l'expérience utilisateur.

Gauthier : Au cours de ce projet, j'ai pris en charge de nombreuses tâches. J'ai commencé par la mise en place de la base de données, en créant l'ensemble des tables ainsi que tous leurs éléments. J'ai ensuite développé le système de gestion des comptes, incluant les formulaires de connexion et d'inscription, tout en intégrant la fonctionnalité de photo de profil.

Par la suite, j'ai consacré beaucoup de temps à aider les autres membres de l'équipe et à intervenir dans tous les domaines du projet. J'ai notamment participé à la création de la page du panneau d'administration, ainsi qu'à l'amélioration du design général du site en travaillant sur le CSS afin de le rendre plus agréable visuellement.

J'ai également mis en place le système de newsletter ainsi que la page dédiée à son traitement. Un travail important a aussi été réalisé sur la gestion des redirections, afin d'empêcher les utilisateurs d'accéder à des pages non autorisées. J'ai développé le système de gestion des cookies, en rédigeant la page de politique d'utilisation des cookies et en concevant le pop-up d'acceptation.

Enfin, j'ai été à l'origine de nombreuses pages de scripts annexes telles que `log_out.php`, et j'ai rédigé l'intégralité du rapport de stage.