



# Rapport de projet

---

ING1 GI2  
Développement Web

**ARRICASTRES** Guillaume  
**ARTIGAS** Zoé  
**COQUET** Lucas  
**CONTAL** Augustin  
**KLUCZNY** Thomas

FASSI Dieudonné

# Sommaire

<b>1. Introduction et Contexte.....</b>	<b>2</b>
a. Présentation général du projet.....	2
b. Détail des consignes du Projet.....	2
c. Enjeux et objectifs.....	2
d. Choix des technologies.....	3
<b>2. Répartition des tâches.....</b>	<b>3</b>
a. Organisation.....	3
b. Travail collaboratif.....	4
c. Rôles initiaux.....	5
<b>3. Étapes réalisées.....</b>	<b>6</b>
a. Conception.....	6
b. Structure du développement.....	10
c. Développement des modules.....	11
d. Résultats obtenus.....	11
<b>4. Conclusion et perspectives.....</b>	<b>15</b>
a. Bilan général.....	15
b. Améliorations possibles.....	16

# 1. Introduction et Contexte

## a. Présentation général du projet

Dans le cadre de notre première année en école d'ingénierie, nous avons été amenés à réaliser un projet de développement web autour du thème de la "plateforme intelligente, performante et responsive". Ce projet s'inscrit dans une logique de mise en pratique des compétences acquises en cours, en lien avec des technologies web modernes, la gestion de projet, et les problématiques de l'Internet des Objets. Nous avons choisi de concentrer notre travail sur une maison connectée, un cas concret de bâtiment intelligent. Ce choix s'inscrit dans une tendance actuelle où les objets connectés prennent une place croissante dans le quotidien, permettant une automatisation et une optimisation des tâches domestiques. Notre objectif était de concevoir une plateforme centralisée permettant la visualisation, la gestion et l'optimisation de ces objets avec en plus un système d'authentification et de permission en fonction du rôle de l'utilisateur.

## b. Détail des consignes du Projet

Le cahier des charges du projet imposait un cadre strict, destiné à encadrer la conception d'une plateforme web simulant un environnement intelligent, tel qu'un bâtiment ou une ville connectée. Celle-ci devait impérativement intégrer une gestion multi-utilisateurs, permettant de différencier les accès et les fonctionnalités selon le type d'utilisateur. Plusieurs modules fonctionnels devaient être mis en place : un module d'information accessible au public pour proposer une visite libre du site sans aucun accès avancés, un module de visualisation permettant aux utilisateurs de consulter les objets connectés et de gérer leur profil, un module de gestion destiné aux utilisateurs avancés pour configurer ou surveiller les objets, ainsi qu'un module d'administration complet, offrant un contrôle total sur les utilisateurs, les objets et les paramètres du système. La base de données représentait un élément central du projet, devant permettre de stocker des objets connectés tels que des capteurs, caméras ou thermostats, tout en enregistrant les interactions des utilisateurs. Sa structure devait être rigoureuse, modulaire et facilement manipulable. Sur le plan technique, le recours à un ou plusieurs frameworks était obligatoire, en backend comme en frontend, tout en excluant l'usage de CMS tels que WordPress. Enfin, la réalisation devait respecter plusieurs bonnes pratiques professionnelles : l'utilisation de Git pour la gestion de version, l'adoption d'une méthodologie agile de type SCRUM, la mise en œuvre d'un design responsive, ainsi que l'attention portée à l'accessibilité et à la compatibilité multi-supports avec des tests et optimisations réguliers.

## c. Enjeux et objectifs

L'objectif de ce projet dépassait largement la simple création technique d'un site web simulant une maison connectée. Il s'agissait aussi d'un exercice de collaboration imposée, puisque les groupes de travail ont été constitués par tirage au sort. Cette particularité a introduit un véritable enjeu humain dès le départ : il a fallu apprendre à travailler avec des personnes que nous ne connaissions pas, aux méthodes de travail parfois très différentes.

S'ouvrir à d'autres manières de faire, construire une dynamique de groupe équilibrée, et développer une communication fluide faisaient partie intégrante des objectifs pédagogiques du projet. Sur le plan technique, il s'agissait de concevoir une plateforme complète, multi-utilisateur, respectant un cahier des charges rigoureux. Cela incluait la gestion d'une base de données alimentée par des objets connectés, l'intégration d'un système d'authentification sécurisé, la différenciation des rôles (visiteur, simple, complexe, administrateur), et l'accès à des interfaces adaptées à chaque profil. Nous devions également nous approprier des technologies que nous ne connaissions pas encore, notamment le framework Express.js avec Node.js côté backend. Le tout devait être réalisé dans un temps limité, en respectant les bonnes pratiques de développement : lisibilité du code, organisation des fichiers, modularité, et cohérence fonctionnelle. En résumé, les enjeux de ce projet mêlaient apprentissage technique, cohésion de groupe, et capacité à s'adapter pour produire un site web fonctionnel, intuitif et conforme aux attentes.

## d. Choix des technologies

Pour répondre efficacement aux enjeux techniques et organisationnels du projet, nous avons opté pour une stack technologique légère, mais robuste et adaptée à un développement rapide en équipe. Côté backend, nous avons utilisé le framework Express.js avec Node.js, reconnu pour sa simplicité, sa rapidité et sa capacité à gérer les routes, les API REST, la logique serveur et l'authentification utilisateur. Pour la gestion des données, nous avons mis en place une base de données relationnelle avec MySQL, afin de stocker de manière structurée les informations relatives aux utilisateurs, objets connectés, niveaux, et historiques.

Le frontend repose sur une base HTML, CSS et JavaScript, sans framework additionnel, ce qui nous a permis de garder un contrôle total sur le rendu visuel et de concevoir une interface responsive de manière manuelle, en tenant compte des contraintes de compatibilité. Afin d'assurer une bonne organisation au sein de l'équipe, nous avons utilisé Git et GitHub pour le versioning du code, avec des branches dédiées et des commits réguliers. La coordination des tâches et le suivi de l'avancement du projet ont été facilités par Trello, outil collaboratif que nous avons utilisé tout au long du développement. Enfin, la rédaction du rapport s'est appuyée sur Google Docs, afin de permettre une co-écriture fluide et accessible à tous les membres du groupe.

## 2. Répartition des tâches

### a. Organisation

Pour organiser efficacement le travail en équipe, nous avons pris contact dès le début du projet. Lors de cette première réunion, nous avons échangé sur la méthodologie à adopter, défini le sujet du site, et réparti les rôles entre les membres du groupe.

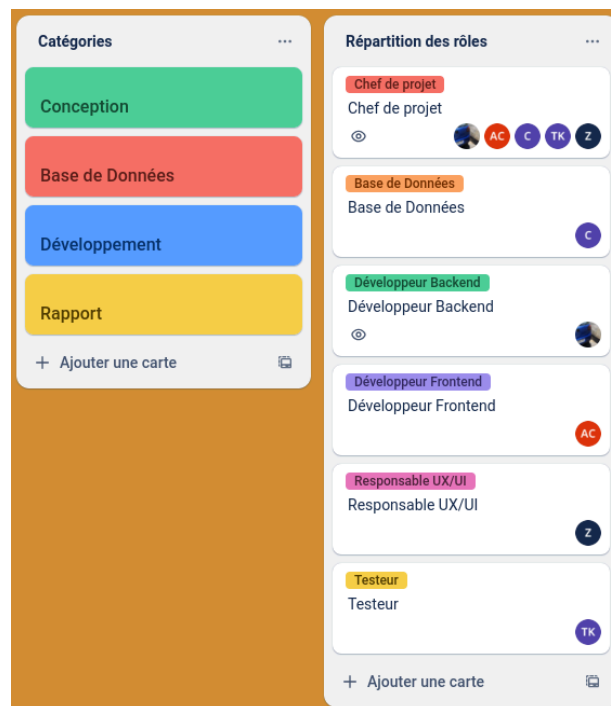
## b. Travail collaboratif

Pour organiser efficacement notre travail, nous avons décidé d'adopter la méthode SCRUM. Il s'agit d'une méthode agile particulièrement bien adaptée aux projets web, car elle permet aux développeurs de rester au plus près des besoins réels des utilisateurs. Elle évite ainsi de perdre du temps à concevoir des fonctionnalités qui ne seront finalement pas utilisées.

Afin de mettre en pratique cette méthode, nous avons choisi d'utiliser Trello, un outil de gestion de projet collaboratif. Nous y avons mis en place un ScrumBoard, qui nous a permis de visualiser l'ensemble des tâches à réaliser et de suivre l'évolution du projet au fil du temps. Ce tableau était structuré de manière à représenter les différentes étapes d'avancement du travail, organisées selon le principe des sprints, c'est-à-dire des périodes de développement courtes et définies à l'avance.

Notre ScrumBoard comportait sept colonnes distinctes : une colonne pour les catégories de tâches, une autre pour la répartition des rôles au sein de l'équipe, puis les colonnes classiques représentant l'état d'avancement, à savoir : à faire, en cours, bloqué, à vérifier et terminé. Les deux premières colonnes, consacrées aux catégories et aux rôles, faisaient office de repères visuels. Elles associaient à chaque type de tâche et à chaque membre de l'équipe une couleur spécifique, ce qui nous permettait d'identifier rapidement, d'un simple coup d'œil, le type de travail à effectuer ainsi que la personne en charge.

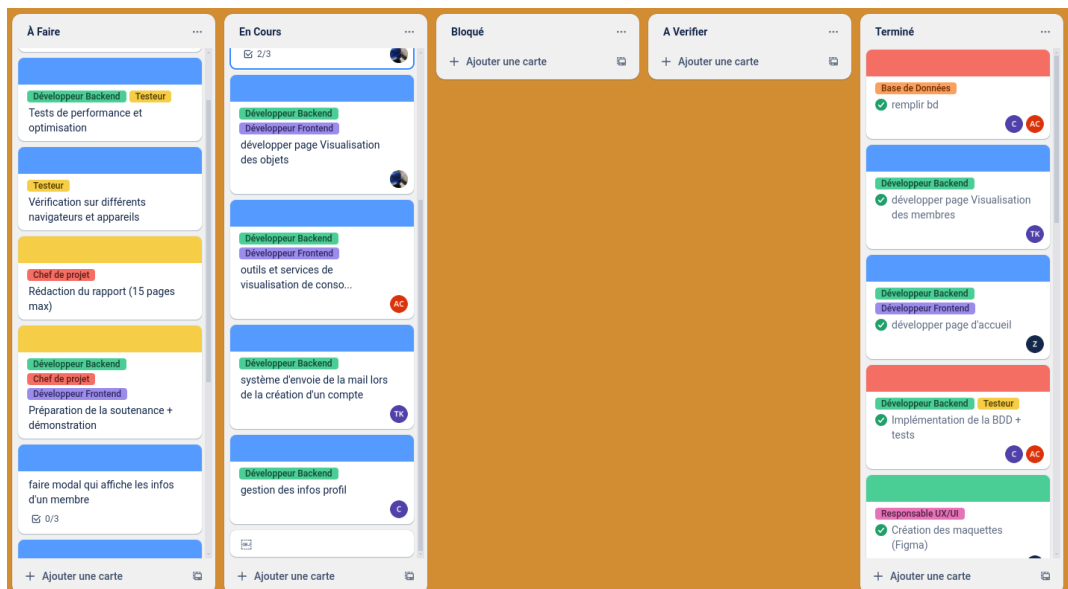
Les tâches, sous forme de cartes, étaient déplacées manuellement d'une colonne à l'autre au fur et à mesure de leur progression. Cette organisation visuelle s'est révélée très utile pour assurer un bon suivi du projet, faciliter la communication entre les membres de l'équipe et garantir une répartition équilibrée du travail.



Grâce au système de couleurs que nous avons mis en place, il est possible d'identifier immédiatement à quelle catégorie appartient une tâche, simplement en jetant un coup d'œil à sa couleur.

Les autres colonnes du tableau servent à suivre l'avancement du travail. Lorsqu'une nouvelle tâche est créée, elle est placée dans la colonne "À faire". Dès qu'elle est en cours de réalisation, elle est déplacée dans la colonne "En cours". Une fois terminée, elle passe dans la colonne "Terminé". En parallèle, la colonne "Bloqué" permet de recenser les tâches qui ne peuvent pas avancer en raison d'un problème ou d'un obstacle. Enfin, la colonne "À vérifier" regroupe les tâches achevées mais qui nécessitent une validation ou un contrôle avant d'être considérées comme définitivement finalisées.

Ce système de suivi visuel nous a permis de garder une vision claire et structurée de l'état d'avancement du projet. Voici une représentation concrète de notre ScrumBoard :



Comme on peut le voir, chaque tâche est associée à une catégorie identifiable par sa couleur, à un rôle indiqué par une étiquette, ainsi qu'à une ou plusieurs personnes chargées de la réaliser.

### c. Rôles initiaux

Comme le montre notre ScrumBoard, nous avons défini des rôles spécifiques au sein de l'équipe. L'idée était d'identifier autant de rôles qu'il y a de membres dans le groupe, soit cinq, afin que chacun soit référent sur un domaine particulier. En cas de difficulté ou de question liée à un aspect précis du projet, cela nous permet de savoir à qui nous adresser. La répartition des rôles s'est faite en prenant en compte les préférences et les compétences de chacun. Il est important de noter que ces rôles ne limitent pas les membres à un seul type de tâche. Au contraire, nous avons souhaité que chacun participe à l'ensemble des étapes du projet afin de favoriser une collaboration globale et une montée en compétences

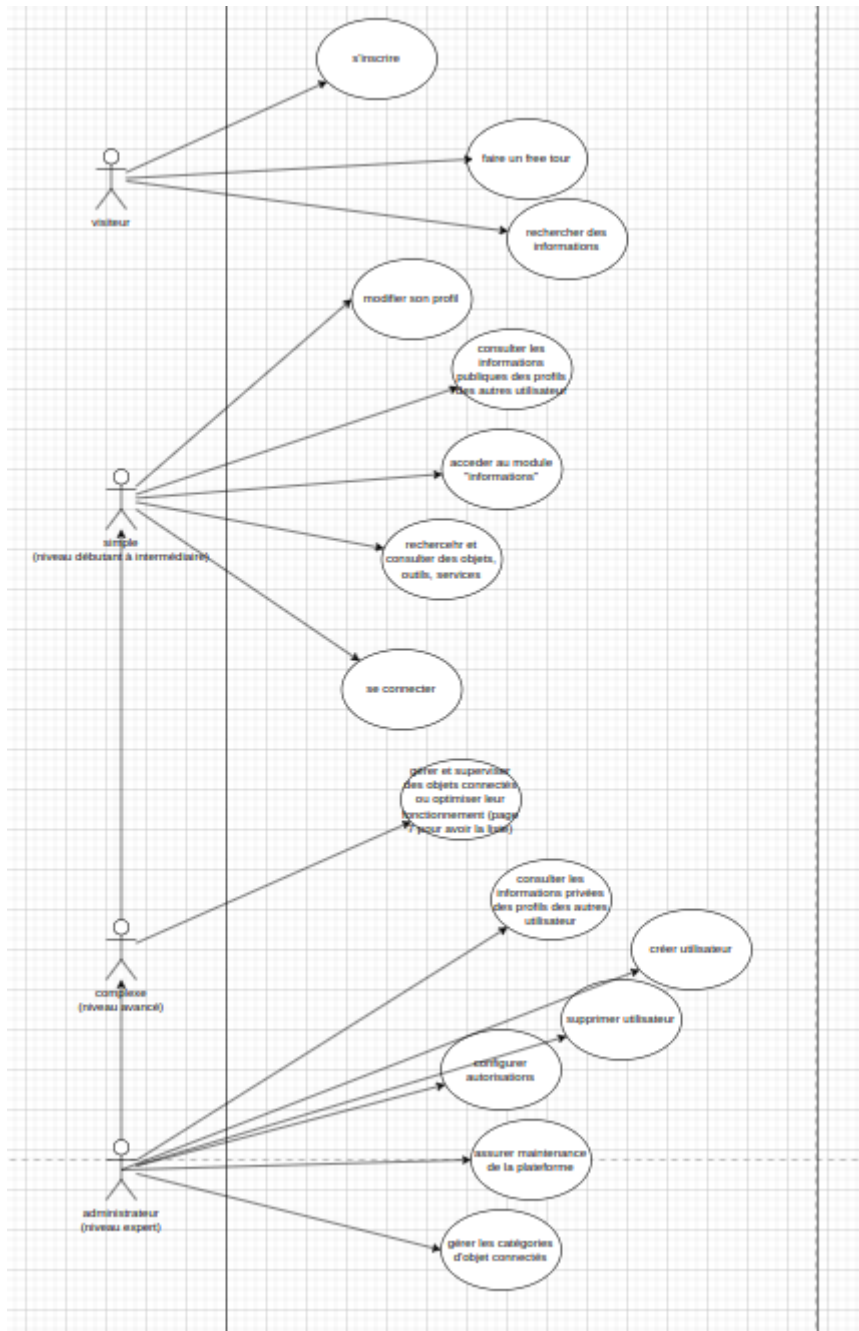
collective. Les rôles définis sont les suivants : responsable de la base de données, développeur backend, développeur frontend, responsable UI/UX et testeur. En parallèle, nous avons décidé que tous les membres participeraient à la gestion du projet en endossant collectivement le rôle de chef de projet. Concrètement, la répartition est la suivante : Lucas Coquet est en charge de la base de données, Guillaume Arricastres s'occupe du développement backend, Augustin Contal est responsable du frontend, Zoé Artigas gère la partie UI/UX, et Thomas Kluczny est chargé des tests.

## 3. Étapes réalisées

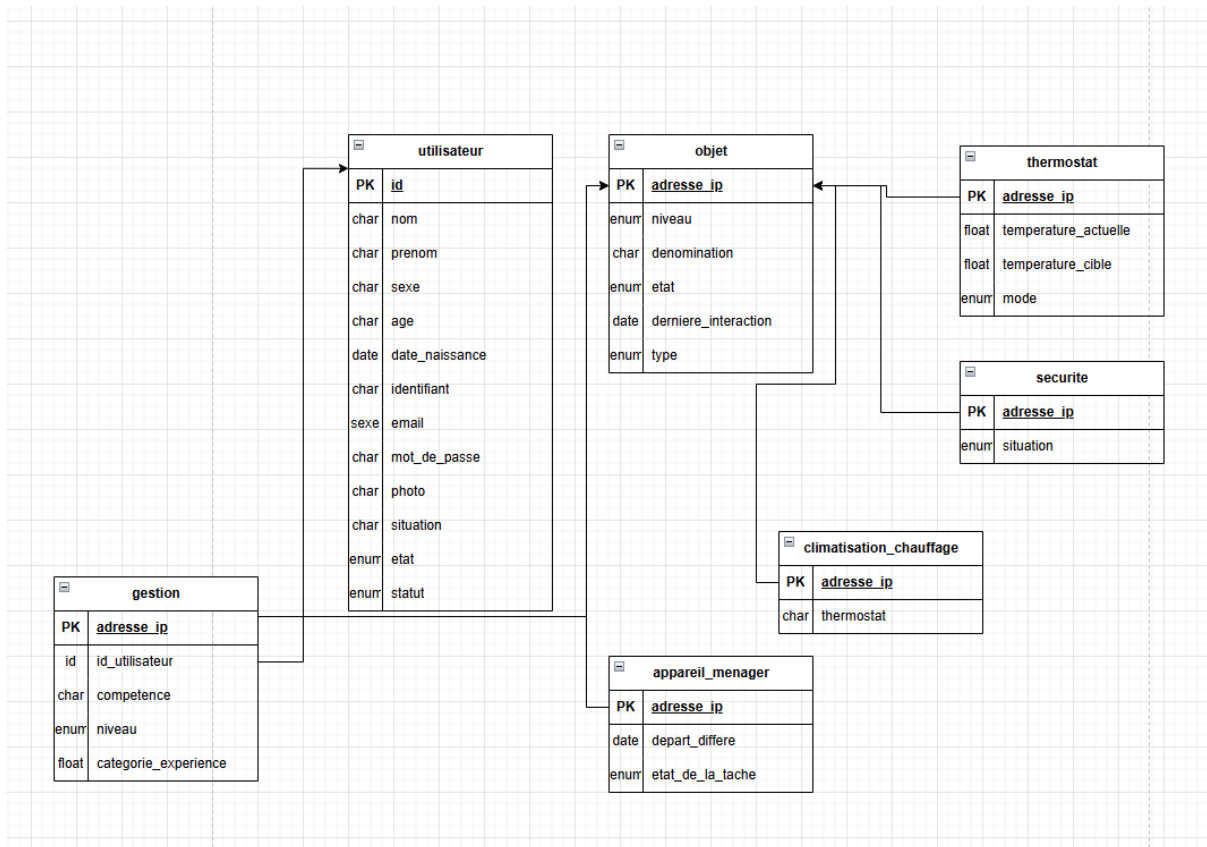
### a. Conception

Avant de passer à la phase de développement, il était essentiel de choisir le sujet de notre site web, puis de réaliser un travail de conception approfondi. Pour déterminer le thème, nous avons organisé un vote au sein du groupe afin de départager trois propositions : une ville connectée, un bâtiment connecté ou une maison connectée. C'est finalement la maison connectée qui a été retenue à l'unanimité. Une fois le sujet validé, nous avons réfléchi aux technologies à utiliser ainsi qu'à la structure générale du projet. Comme mentionné précédemment, nous avons opté pour le framework Express côté backend, et pour MySQL en tant que base de données. En parallèle, nous avons suivi la démarche vue en cours en réalisant un diagramme des cas d'utilisation ainsi qu'un diagramme UML. Ces outils nous ont permis de clarifier les fonctionnalités à développer, de définir les différents profils d'utilisateurs, et de structurer de manière cohérente notre base de données.

Voici le diagramme correspondant :

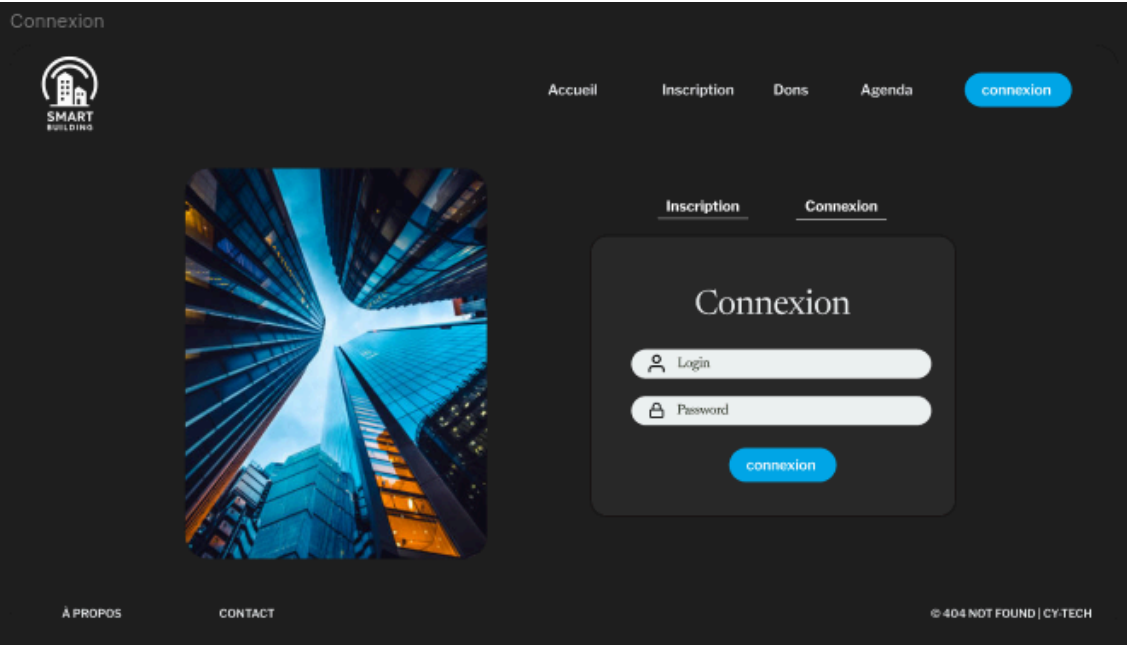
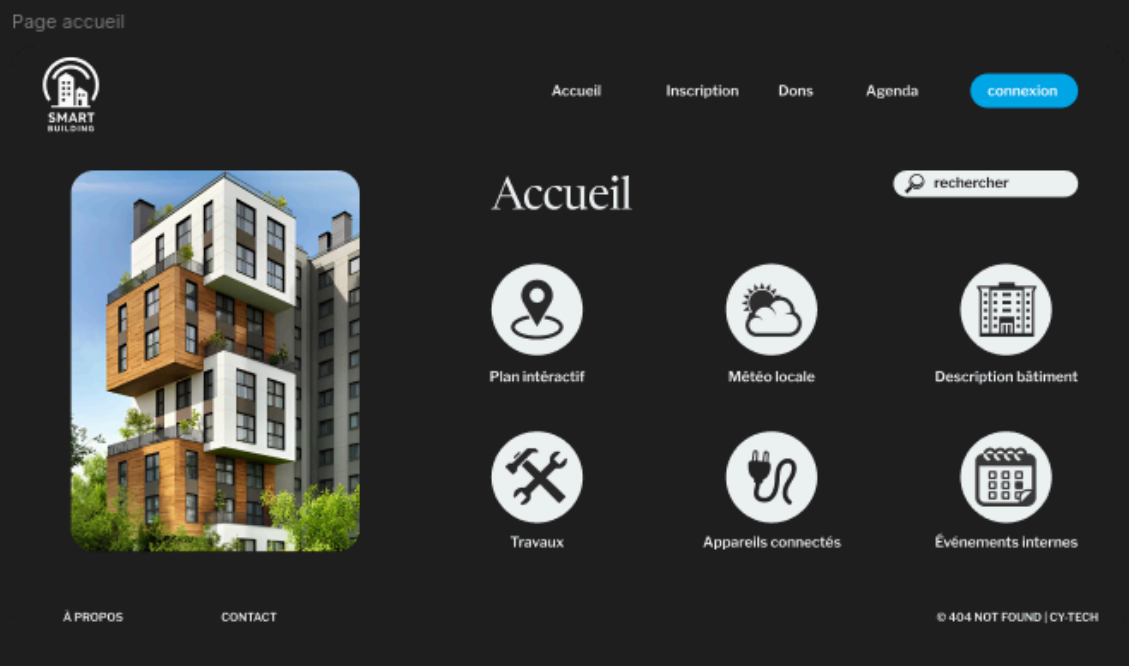


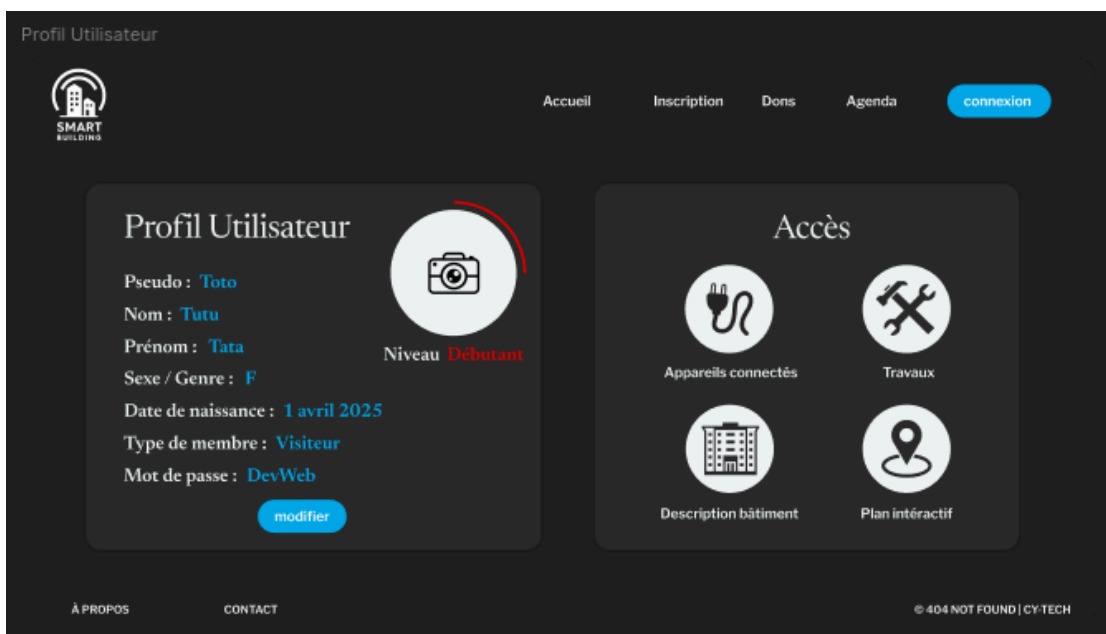
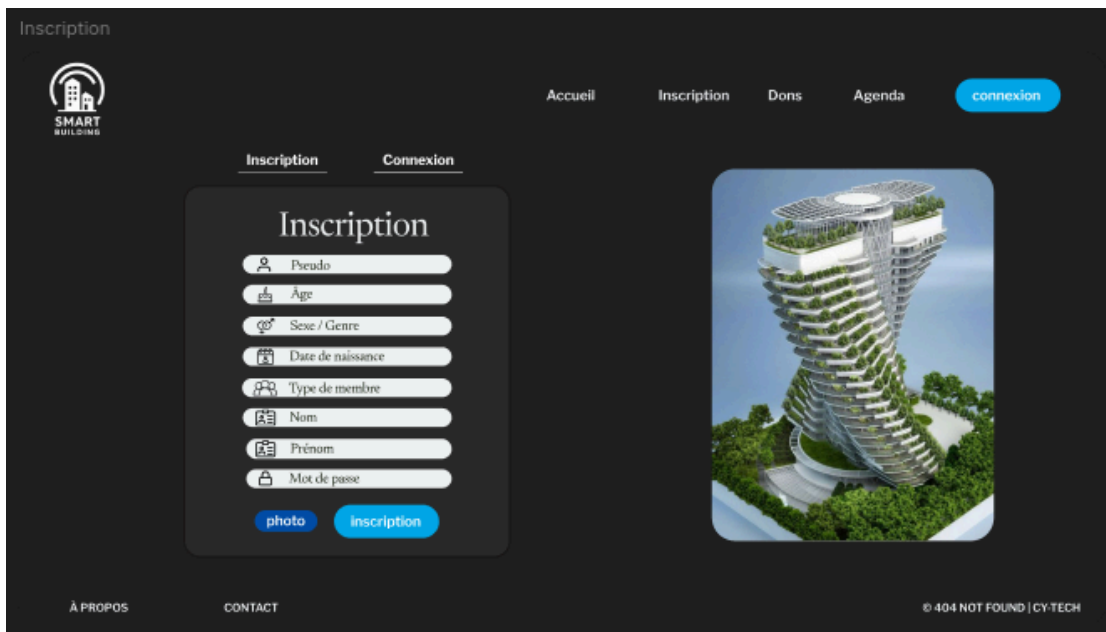
De même pour le diagramme sql, il nous a permit de bien structurer notre base de données avant de se lancer dans la rédaction du code SQL :



Avant de débuter le développement du site, nous avons naturellement pris le temps de concevoir des maquettes sur Figma. Celles-ci nous ont offert une première vision concrète de l'interface du site, en définissant notamment la charte graphique, l'ergonomie générale, ainsi que l'agencement des boutons et des différentes fonctionnalités. Ces maquettes ont servi de référence tout au long du développement, en guidant nos choix en matière de design et d'expérience utilisateur.

Vous trouverez ci-dessous les maquettes Figma qui ont servi de base à la conception visuelle du site :





## b. Structure du développement

Une fois toute la conception finie, le développement commence. Nous avons tout d'abord créé un dépôt GitHub afin que chacun puisse accéder au code source du projet et le modifier. En utilisant un framework tel que Express, la structure de notre projet est déjà prédéfinie. Nous y retrouvons le dossier "views" contenant les fichiers .ejs permettant la création et la structuration des différentes pages du site avec du HTML. Il y a aussi le dossier routes contenant les api tel que les fichiers qui définissent les chemins URL qui nous permettent ici de relier les pages à la base de données. Et enfin le dossier public contient quant à lui tous les fichiers css ainsi que les images pour la mise en forme. A partir de ça on a rajouté un fichier user.sql qui va contenir toutes la base de données du site ainsi que le fichier index.ejs qui contient le code d'initialisation et le point de départ de notre site. On a

aussi utilisé un dossier config contenant notamment les accès qui vont permettre l'exploitation de la base de donnée, ici les identifiants et mot de passe de mysql. A partir de cette structure on a pu développer notre site plus en profondeur, on a codé pour chaque nouvelle page un fichier .ejs dans le répertoire views où l'on peut retrouver du html et du javascript permettant notamment de manipuler le DOM. Dans le cas où nous voulions utiliser notre bdd nous avons donc rajouter les fichiers nécessaires dans le dossier routes afin d'établir la connexion entre notre page ejs et notre sql.

## c. Développement des modules

Le développement s'est articulé autour de quatre rôles principaux pour les utilisateurs, conformément aux exigences du cahier des charges. Chaque rôle correspond à un niveau de privilèges croissant, et repose sur un principe d'héritage : un rôle supérieur dispose automatiquement des droits associés aux rôles inférieurs, en plus de ses propres fonctionnalités spécifiques.

Après la création d'un compte sur la plateforme, l'utilisateur est initialement enregistré avec un accès limité. C'est un administrateur qui est chargé d'attribuer un rôle à chaque utilisateur, en fonction de ses besoins et responsabilités.

Le premier rôle, appelé Visiteur, est accessible sans authentification. Il permet de découvrir la plateforme à travers quatre catégories principales. Le visiteur peut consulter les objets connectés enregistrés dans la maison (tels qu'une ampoule connectée ou une cuisinière), accéder à la liste des individus présents et utiliser un moteur de recherche multi-filtres pour retrouver des objets selon différents critères (nom, type, marque, état).

Le deuxième rôle, appelé Utilisateur Simple, est attribué après inscription et validation par un administrateur. En plus des fonctionnalités disponibles pour les visiteurs, l'utilisateur simple peut gérer son profil personnel (photo, pseudo, niveau, rôle, etc.) et modifier certaines caractéristiques des objets connectés.

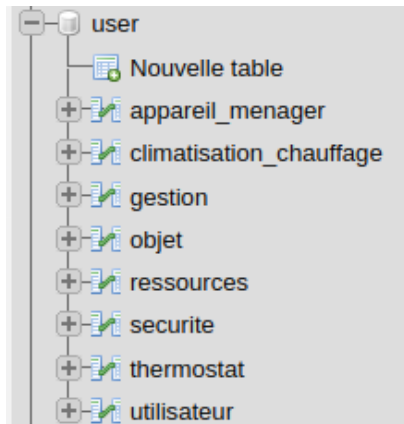
Le troisième rôle, nommé Utilisateur Complexe, est destiné aux utilisateurs ayant un niveau d'accès avancé. En complément des droits précédents, il leur permet d'ajouter de nouveaux objets connectés à la plateforme et de configurer leurs paramètres fonctionnels à l'aide d'un tableau de bord.

Enfin, le quatrième rôle, celui d'Administrateur, donne un accès complet à toutes les fonctionnalités de la plateforme. En plus de celles associées aux autres rôles, l'administrateur peut gérer les utilisateurs (création, suppression, modification des niveaux d'accès) et superviser l'ensemble des objets connectés.

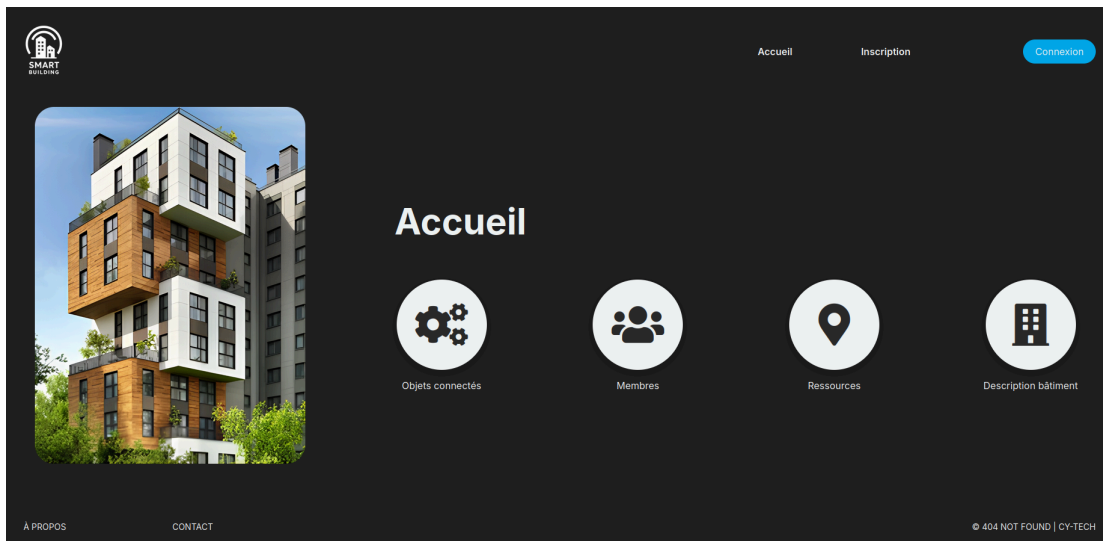
## d. Résultats obtenus

Nous avons réussi à mettre en place une plateforme fonctionnelle, structurée en prenant en compte les exigences du cahier des charges. Concrètement nous avons donc:

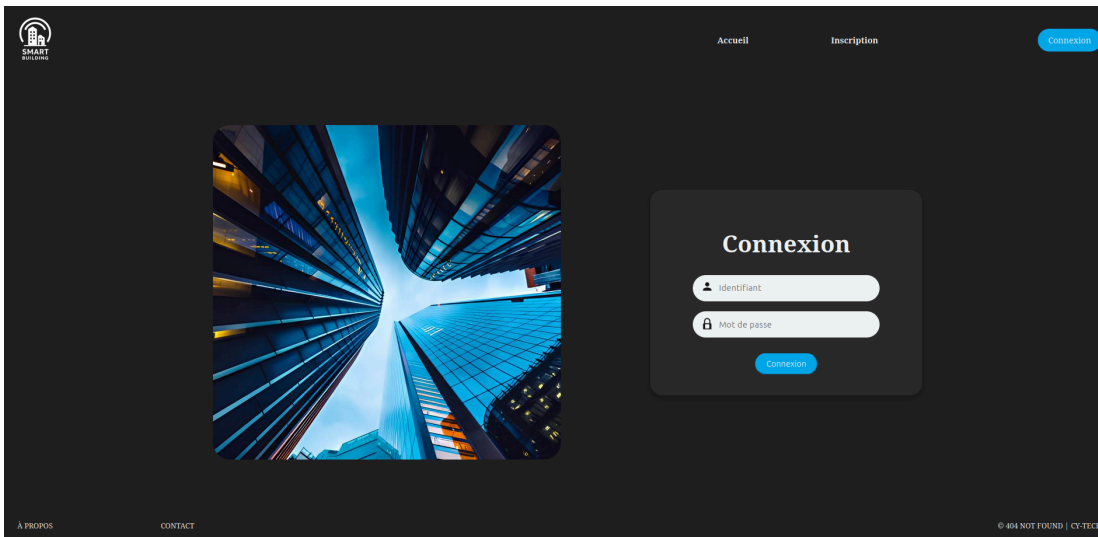
- Une base de données sur mysql dynamique contenant les utilisateurs et les objets connectés et les ressources de la maison



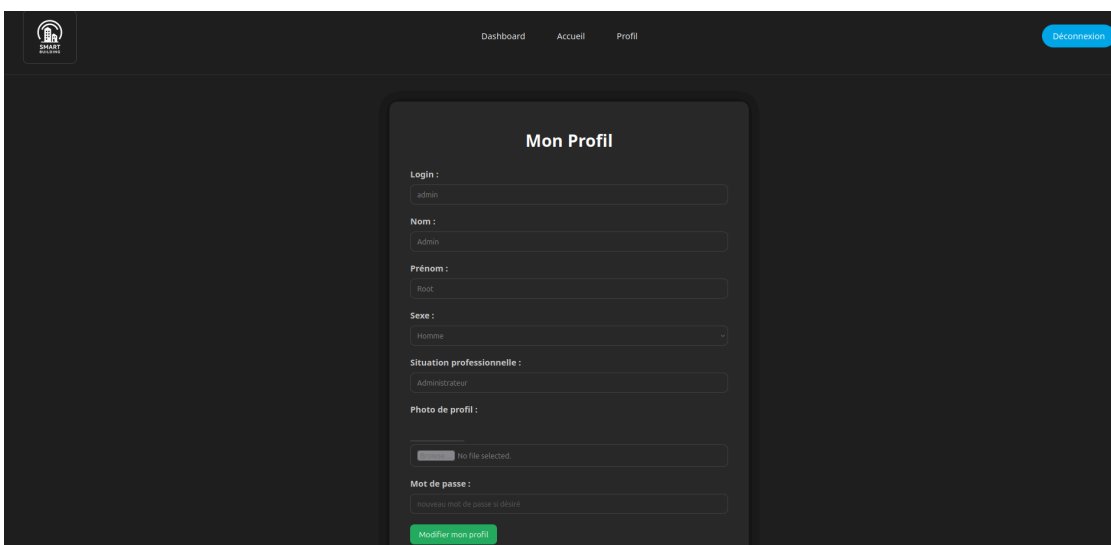
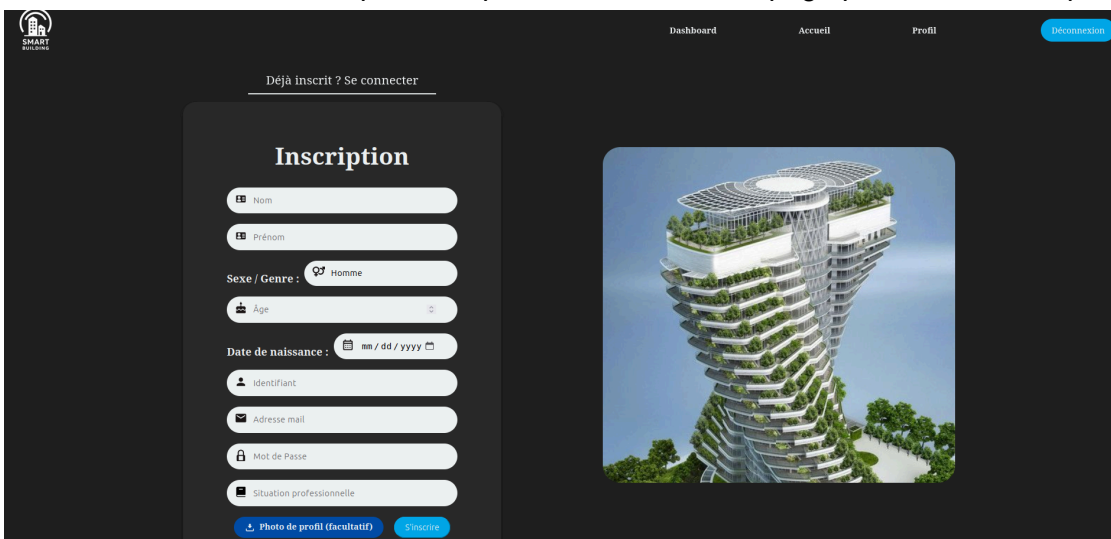
- 
- Une interface utilisateur fluide et instinctive



- Une page de connexion



- La possibilité de s'inscrire sur le site avec ses informations personnelles, une adresse mail et une photo de profil. Avec aussi une page pour consulter le profil.



- Un Dashboard pour gérer les objets et les utilisateurs

SMART BUILDING

Accueil Profil Déconnexion

## Gestion des utilisateurs

Ajouter un utilisateur +

Nom	Prénom	Statut	État	Modifier	Supprimer
Dupont	Jean	Complexe	Validé	Modifier	Supprimer
Martin	Sophie	Visiteur	En attente	Modifier	Supprimer
Durand	Paul	Administrateur	Validé	Modifier	Supprimer
Bernard	Alice	Simple	Validé	Modifier	Supprimer
Lemoine	Alex	Simple	En attente	Modifier	Supprimer
Arricastres	Gillaume	Simple	Validé	Modifier	Supprimer
Chosson	Clément	Complexe	Validé	Modifier	Supprimer
Admin	Root	Administrateur	Validé	Modifier	Supprimer

SMART BUILDING

## Gestion des objets

Ajouter un objet +

Nom	Adresse IP	Type	Niveau	État	Modifier	Supprimer
Ampoule connectée	192.168.1.10	Lumière	Débutant	Inactif	Modifier	Supprimer
Cuisinière	192.168.1.11	appareil_menager	Débutant	Inactif	Modifier	Supprimer
Machine à café	192.168.1.12	appareil_menager	Débutant	Inactif	Modifier	Supprimer
Thermostat intelligent	192.168.1.20	thermostat	Intermédiaire	Actif	Modifier	Supprimer
Caméra de surveillance	192.168.1.30	Capteur	Avancé	Actif	Modifier	Supprimer
Machine à laver	192.168.1.40	appareil_menager	Débutant	Inactif	Modifier	Supprimer
Enceinte intelligente	192.168.1.50	Enceinte connectée	Débutant	Inactif	Modifier	Supprimer
Prise connectée	192.168.1.60	Prise	Intermédiaire	Actif	Modifier	Supprimer
Serrure connectée	192.168.1.70	securite	Avancé	Actif	Modifier	Supprimer
Porte connectée salle de contrôle	192.168.1.80	securite	Expert	Actif	Modifier	Supprimer
Climatisation	192.168.1.90	Climatisation	Avancé	Actif	Modifier	Supprimer


- Une gestion des fonctionnalités des objets selon le rôle

SMART BUILDING

Dashboard Accueil Profil Déconnexion

## Membres

Rechercher




<b>Dupont</b> Jean 30 ans complexe	<b>Martin</b> Sophie 25 ans visiteur
<b>Durand</b> Paul 45 ans administrateur	<b>Bernard</b> Alice 28 ans simple
<b>Lemoine</b> Alex 32 ans simple	<b>Arricastres</b> Guillaume 26 ans simple
<b>Chosson</b> Clément 25 ans complexe	<b>Admin</b> Root 30 ans administrateur

SMART BUILDING

Dashboard Accueil Profil Déconnexion

## Objets Connectés

Rechercher



<b>Ampoule connectée</b> null Niveau: débutant	<b>Cuisinière</b> null Niveau: débutant	<b>Machine à café</b> null Niveau: débutant
<b>qcqccqcc</b> 192.168.1.20 Niveau: intermédiaire	<b>Caméra de surveillance</b> null Niveau: avancé	<b>Machine à laver</b> null Niveau: débutant
<b>Enceinte intelligente</b> null	<b>Prise connectée</b> null	<b>Serrure connectée</b> null

## 4. Conclusion et perspectives

### a. Bilan général

Ce projet de maison connectée a été une expérience enrichissante sur beaucoup de niveaux pour tous les membres du groupe. Au-delà du développement technique, il nous a permis de travailler en équipe sur un projet complet, du cadrage initial à la livraison finale, en passant par toutes les étapes de conception, de codage, de tests, et de documentation.

Nous avons pu appliquer des notions vues en cours dans un cadre concret, tout en découvrant de nouveaux outils et de nouvelles méthodes de travail afin de débloquent certaines interrogations. Même si des rôles avaient été attribués, dès le départ nous avons rapidement compris qu'une approche collaborative et polyvalente était plus efficace et plus formatrice.

De plus, la création des modules demandés par le cahier des charges (Information, Visualisation, Gestion, Administration) nous a permis d'explorer différentes facettes du développement web, en particulier la gestion des droits d'accès et la mise en lien entre le front-end et la base de données.

### b. Améliorations possibles

Plusieurs pistes d'amélioration pourraient enrichir la plateforme et renforcer sa portée fonctionnelle comme visuelle. L'intégration d'un framework frontend moderne, tel que React ou Vue.js, permettrait de rendre l'interface utilisateur plus dynamique, fluide et réactive, tout en facilitant la gestion des composants. L'ajout d'API de simulation pour les objets connectés offrirait une interaction plus réaliste avec les dispositifs virtuels, renforçant ainsi l'aspect immersif de l'environnement intelligent. Une visualisation de données plus avancée, avec des graphiques dynamiques ou des tableaux de bord en temps réel, améliorerait significativement l'expérience utilisateur, en particulier pour les profils complexes ou administrateurs. Enfin, héberger le site sur un serveur distant rendrait la plateforme accessible à tout moment, facilitant les tests à distance et permettant une démonstration plus professionnelle et fluide de l'application.

Ces améliorations permettraient d'enrichir encore l'expérience utilisateur et la crédibilité du projet tout en nous rapprochant encore plus d'un projet livrable réel et nous permettant alors de nous enrichir encore plus profondément.