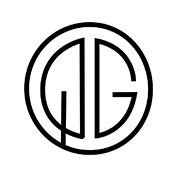
D’SOUZA Joliston BTS SIO 1 B

CRETIN Yann



**TEAM QG**

**Description de l’organisation :**

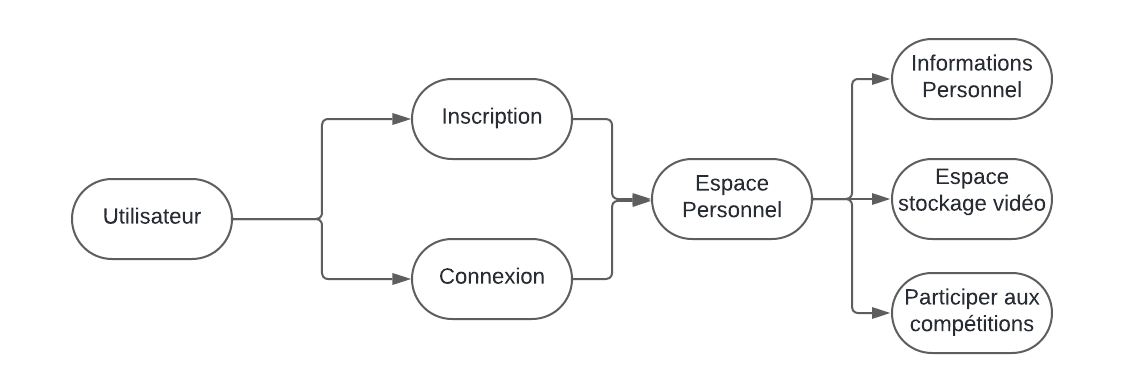
Team QG est une jeune association évoluant dans l'événementiel créée par quatre amis qui ont pour but de fédérer toutes les promotions passées et futurs de la section hip-hop du lycée Turgot dans laquelle ils se sont rencontrés. Elle est aujourd'hui une organisation créant divers liens générationnels autour de la culture hip-hop par le biais d'événements culturels et/ou compétitifs ou de prestations publiques rassemblant plusieurs générations de danseurs étalées sur presque 10 ans. Situé dans la région parisienne dans le 12e arrondissement

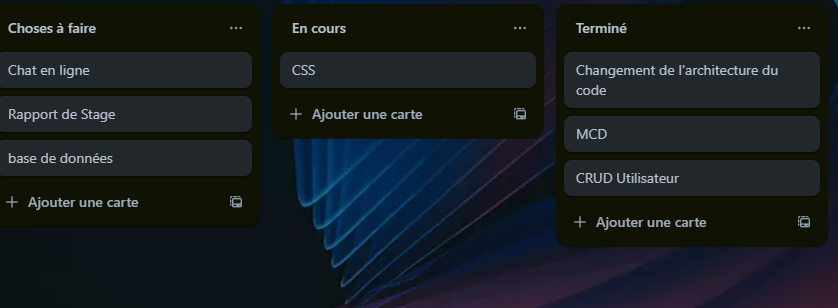


**Environnement de travail :**

Nous sommes deux stagiaires, moi et Yann CRETIN, travaillant en télétravail par appels vocaux pour développer le site web TEAM QG en utilisant les langages HTML/CSS, PHP, JavaScript et SQL. Nous communiquons entre nous et avec notre tuteur via plusieurs plateformes incluant Whatsapp et Discord. Nous utilisons Github pour la gestion de nos codes, et les tâches sont réparties de manières organisées via Trello en s'assurant de la qualité du code en réalisant des revues de code ensemble. Grâce à cela, on parvient à gérer le front-end et le back-end du site, implémentant des nouvelles fonctionnalités et une interface utilisateur plus professionnelle et attrayante.

Use case :

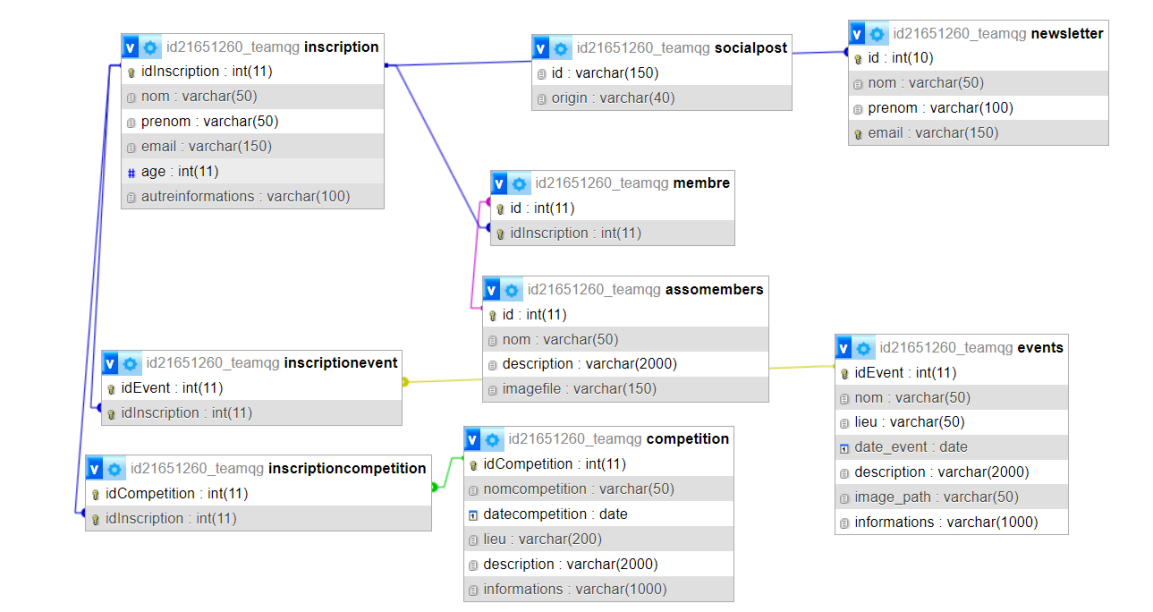


Trello : 

**Auto-formation :**

Pendant ma formation, j'ai approfondi ma compréhension du langage HTML/CSS, PHP, SQL et JavaScript. Par exemple, pour le JavaScript, cela m'a permis d'améliorer le design du site pour le rendre responsive et attrayant. J'ai donc entrepris d'apprendre JavaScript, un langage de programmation essentiel pour rendre les pages web interactif et dynamique. Nous commençons par apprendre la syntaxe de base, y compris les variables (let, const), les types de données (nombres, chaînes de caractères, objets, tableaux) et les opérateurs. En ce qui concerne le HTML/CSS, j'ai pu acquérir de nouvelles compétences pour modifier l'apparence d'une structure selon mes préférences, en utilisant des techniques de mise en page avec Flexbox et Grid afin de créer des designs réactifs.

**Description du MCD TeamQG :**



**Tables :**

Inscription : Représente les informations de base sur les personnes inscrites, y compris leur nom, prénom, email, âge et autres informations supplémentaires.

socialpost : Représente les publications sur les réseaux sociaux, identifiées par un identifiant unique et leur origine.

newsletter : Représente les abonnements à la newsletter, comprenant le nom, le prénom et l'email de l'abonné.

membre : Représente les membres de l'organisation, en reliant chaque membre à une inscription via l'identifiant d'inscription.

assomembers : Représente les membres d'une association avec des informations détaillées telles que le nom, la description et le fichier image associé.

events : Représente les événements organisés par l'organisation, avec des détails comme le nom, le lieu, la date, la description, le chemin de l'image et des informations supplémentaires.

competition : Représente les compétitions organisées, avec des détails sur le nom, la date, le lieu, la description et des informations supplémentaires.

inscriptionevent : Représente la relation entre les inscriptions et les événements, en liant chaque inscription à un ou plusieurs événements.

**Relations entre les tables :**

inscription - membre :

Relation de type 1:1 (un-à-un). Chaque inscription peut être associée à un membre. La table membre utilise idInscription comme clé étrangère pour lier un membre à une inscription.

inscription - inscriptionevent - events :

Relation de type M

(plusieurs-à-plusieurs). Une inscription peut participer à plusieurs événements, et un événement peut accueillir plusieurs inscriptions. La table inscriptionevent sert de table d'association pour gérer cette relation.

inscription - inscriptioncompetition - competition :

Relation de type M

(plusieurs-à-plusieurs). Une inscription peut participer à plusieurs compétitions, et une compétition peut accueillir plusieurs inscriptions. La table inscriptioncompetition sert de table d'association pour gérer cette relation.

assomembers :

Indépendante des autres tables, cette table contient des informations sur les membres de l'association, sans relation directe avec les autres tables dans ce diagramme.

socialpost :

Indépendante des autres tables, cette table contient des informations sur les publications sociales sans relation directe avec les autres tables dans ce diagramme.

newsletter :

Indépendante des autres tables, cette table contient des informations sur les abonnements à la newsletter sans relation directe avec les autres tables dans ce diagramme.

**Tâche occupante :**

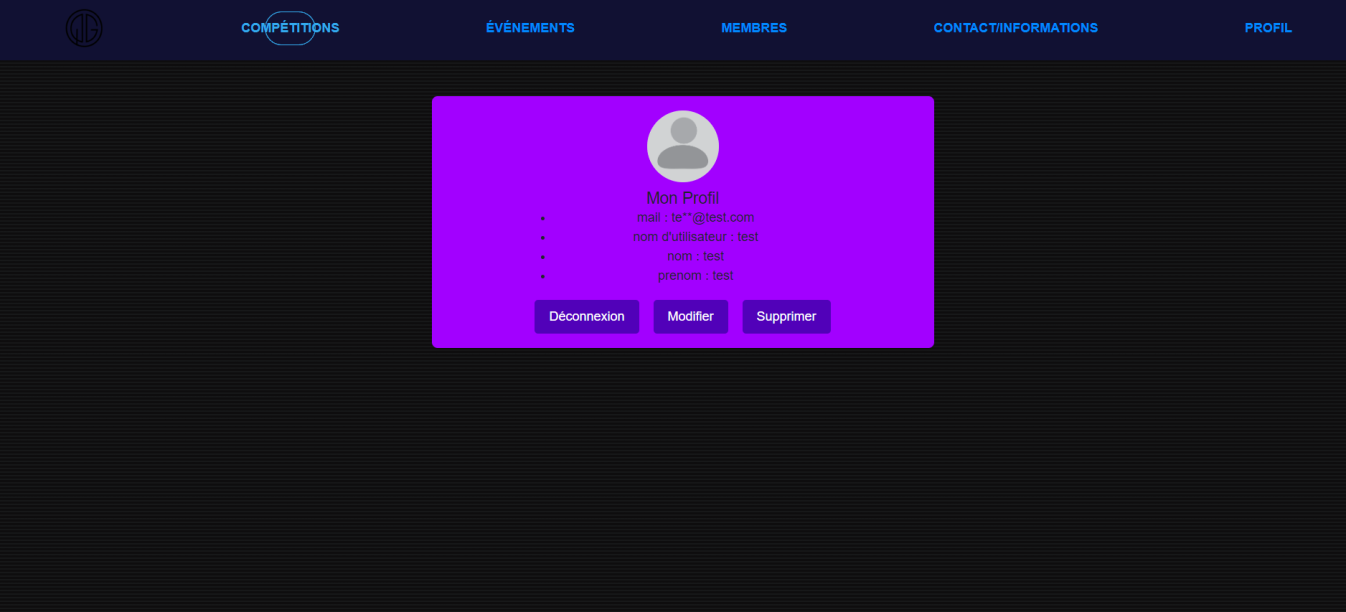
Parmi les nombreuses tâches données par notre tuteur, l'une des plus importantes était de rendre le site plus professionnel et plus attrayant.

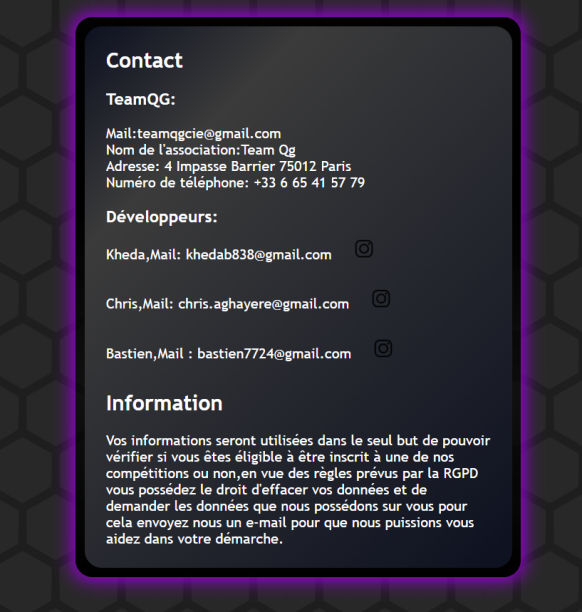
Pour donner un aspect plus professionnel au site web, plusieurs améliorations peuvent être apportées. En premier lieu, il est crucial d'opter pour un design moderne et cohérent, en utilisant une palette de couleurs harmonieuse, une typographie élégante, et en cherchant à se démarquer des autres sites. L'interface utilisateur doit être intuitive pour garantir une navigation facile. Intégrer des contenus de haute qualité, comme des images en haute résolution et des vidéos informatives, peut renforcer l'aspect professionnel du site.

Footer :



Espace Personnel :



Page Contact/Informations : 

De plus, l'ajout de fonctionnalités interactives telles que des formulaires de contact, des inscriptions en ligne et des calendriers d'événements peut stimuler l'engagement des utilisateurs. Enfin, veiller à une sécurité renforcée en prenant des mesures de protection des données est essentiel pour établir la confiance des utilisateurs. En combinant ces éléments, le site web atteindra un niveau élevé de professionnalisme, répondant ainsi aux attentes des utilisateurs.

Les tâches les plus compliquées selon moi, on était le codage en full-stack, ce n’était pas une tâche essentiellement difficile, mais assez longue et contraignante, car il y a eu un changement d’architecture à faire, relié à une mauvaise compréhension du code de ma part, alors j’ai préféré changer directement toute l’architecture de celui-ci, le problème était que le “controller hub” ou Contrôleur Principale (avec les actions) se trouvait dans les modèles et non dans le contrôler, je ne sais pas quelle est la norme ou si cela est normal. Cependant, étant donné que j’ai été habituer à travailler avec un “Contrôleur” principal dans un dossier “contrôleur”, j’ai préféré changer directement cela, ce qui m’a pris un peu plus de temps que prévu suite au changement des liens, des actions, et également le nom des dossiers (afin de les renommer ou directement reliés aux include). Hormis ce petit bémol, la partie longue était la création des fonctions back-end dans les modèles ainsi que leurs liaisons avec à l’architecture mvc, puis la base de données (la base de données m’a pris également un peu de temps, il n’a pas fallu changer grand-chose, mais encore une fois reliés à une mauvaise compréhension de ma part, je m’en suis rendu compte assez tard, et je m’étais lancé sur une mauvaise piste.).

code du CRUD utilisateur :

