# Projet IO2: Réalisation d'un site de notation

Année 2021-2022

## 1 Organisation

**Équipes :** Vous devez réaliser ce projet par binômes. **Aucune équipe de plus de deux personnes ne sera acceptée**. Vous pouvez faire le projet avec un·e étudiant·e d'un autre groupe. Les monômes sont interdits sauf exceptions[1] et ils ne seront autorisés qu'au cas par cas par les enseignants.

**Inscription :** L'inscription s'effectuera sur Moodle. Les membres d'une même équipe doivent s'inscrire dans le même binôme (identifié par un numéro) sur Moodle. Les numéros n'ont aucune signification, l'ordre de passage des groupes sera fixé plus tard.

**Rendu :** Vous devrez rendre votre travail dans une archive sur Moodle.

Si vous travaillez sur vos propres machines, pensez à tester votre site web au Script pour le cas où vous auriez un problème avec votre ordinateur.

**Soutenances :** L'objectif de ce projet est de réaliser un site complet et d'être capable de présenter votre travail lors d'une soutenance de 20 à 30 minutes.

Il est important de travailler **en groupe** de manière équitable: chacun des membres du binôme devra avoir programmé une partie significative du projet. Par exemple, si un·e étudiant·e ne code que le HTML et le CSS et son·sa camarade tout le reste, ce n'est pas équitable. Par ailleurs, chacun des membres du binôme doit pouvoir répondre à des questions sur chaque partie du projet.

## 2 Travail demandé

L'objectif de ce projet est de développer un prototype de site de notation. Ce peut être, par exemple, un site qui recueille des avis et des notes sur

[1] Si le nombre d'étudiants est impair, par exemple.

des restaurants, des films, des voitures, .... Vous êtes libres de choisir la thématique et le contenu du site; soyez créatifs!

Certaines fonctionnalités sont imposées (dans la suite, on prend l'exemple des restaurants).

1. *Page d'accueil*: La page d'accueil devra permettre de choisir de s'inscrire, de se connecter avec un compte déjà créé ou de faire des recherches. Ces fonctionnalités pourront être effectuées via des redirections vers d'autres pages.

2. *Page d'inscription*: Une page avec formulaire pour créer un nouveau compte. L'utilisateur pourra au minimum choisir un pseudo et un mot de passe (qui sera stocké de manière **chiffrée**).

3. *Page profile*: Chaque utilisateur connecté aura une page associée à son compte qui montrera les derniers avis ou notes qu'il a donnés.

4. *Fonction de recherche*: il sera possible de faire des recherches selon plusieurs critères: par exemple, le nom du restaurant et/ou la ville et/ou la note minimale.

   On pourra au choix considérer que tous les avis sont visibles par tous, ou que seuls les restaurants sont visibles par les utilisateurs non connectés mais pas les avis ou encore que seule une sélection est visible aux utilisateurs non connectés. (voir extension)

5. *Comptes administrateurs*: Certains comptes pourront être désignés comme administrateurs. Ces administrateurs pourront effacer les avis ou notes de tous les autres utilisateurs, ou supprimer des utilisateurs.

6. *Publier des avis*: Les utilisateurs inscrits et connectés pourront donner des avis et/ou des notes. Par exemple, une fois qu'il a sélectionné un restaurant, il pourra dire ce qu'il en pense.

Dans la configuration minimale (avant extension),

- il est possible pour un utilisateur de s'inscrire et, s'il a un compte, de se connecter.
- la base est préremplie avec un certain nombre de restaurants, il n'y a pas de formulaire prévu pour en ajouter un.
- Il faut qu'il soit possible de mettre soit des avis soit des notes, mais pas forcément les deux.
- Pour la fonction de recherche, il doit y avoir au moins deux critères possibles combinables entre eux. (Par exemple, chercher les restaurants à Lyon qui ont eu une note supérieures à 4)

- Les avis sont tous visibles par tous les utilisateurs.
- il y a un compte administrateur qui peut supprimer des comptes ou des avis.

**Extensions:** Le nombre d'extensions doit être égal au nombre de membres du groupe (donc normalement 2).

1. *avis publics et privés* Faire en sorte que certains avis ou notes soient visibles par les utilisateurs non connectés, mais pas tous. Le critère peut être la date de l'avis; on peut aussi ne montrer que le dernier avis sur un restaurant, film, ...
2. *Ajouts de restaurants*: Permettre l'ajout de restaurant, soit par les utilisateurs connectés, soit uniquement par les administrateurs.
3. *Likes, réactions ou commentaires*: Ajouter un bouton "Le même" à chaque note, qui permet à un utilisateur connecté de donner automatiquement la même note pour ce restaurant.
4. *Signalement de contenu à un administrateur*: Ajouter un bouton à chaque avis permettant de le signaler à un compte administrateur. Les comptes administrateurs pourront consulter une liste des avis signalés dans une nouvelle page. Pour chaque avis signalé, on indiquera l'utilisateur qui l'a donné et le choix d'ignorer le signalement ou d'effacer l'avis.
5. Votre imagination...

**Langages.** Le site devra utiliser les langages vus en cours et TP: HTML, CSS, PHP et MySQL. L'utilisation de Javascript est autorisée mais pas requise et uniquement pour de petites portions de code, PHP doit rester majoritaire. L'utilisation d'un framework CSS (bootstrap, etc.) ou PHP (Symfony, etc.) n'est pas autorisée.

**Cibles.** Le site devra avant tout être pensé pour être utilisé par des êtres humains. Il devra en particulier être accessible en navigateur, et le ou la visiteur·se ne devra jamais avoir à deviner où se trouve ce qu'il ou elle cherche.

# 3 Évaluation

L'archive rendue devra contenir:

- Le code intégral de votre site;

- Un document décrivant comment le faire fonctionner, et notamment:
  - Comment préremplir la base de données, créer les tables, etc. Il est recommandé de fournir un fichier `.sql` tout fait pour cela.
- Un document décrivant le schéma de la base de données, il peut être manuscrit, il ne vous est pas demandé de faire un schéma UML, donnez juste les tables, leurs colonnes et si une colonne d'une table correspond à une colonne d'une autre, indiquez-le. Si vous aviez prévu un schéma pour plus d'extensions que réalisées, indiquez les parties qui ne servent pas, nous tiendrons compte de cette partie du travail.

L'évaluation portera en particulier sur les critères suivants:

**Modularité du code.** Le code devra être aussi modulaire que possible. On séparera les fonctionnalités distinctes dans des fonctions, classes différentes ou fichiers différents.

**Factorisation du code.** On évitera la répétition de code en préférant créer une nouvelle fonction, classe ou un nouveau fichier.

**Clarté du code.** Le code devra être clair et lisible facilement par l'évaluateur·rice. Il est recommandé qu'il soit organisé, aéré, indenté. L'utilisation de commentaires explicatifs est extrêmement recommandée. De plus, il est attendu que le code soit valide et respecte les standards.

**Sécurité.** On fera attention aux failles de sécurité et notamment aux injections SQL et au stockage des mots de passe. La session d'un utilisateur connecté doit aussi être maintenue de manière chiffrée.

**Esthétique du site.** Un site élégant sera apprécié, mais ce n'est pas le sujet du cours. L'évaluation portera avant tout sur les fonctionnalités. Il en va de même pour l'utilisation du Javascript: cela peut apporter un plus, mais ne sera pas au centre de l'évaluation; vous serez avant tout notés sur la maîtrise des quatre langages du programme du cours.

**Compréhension du projet.** Lors de la soutenance, la répartition des tâches entre les membres du binôme devra être claire. On attendra de chacun·e qu'il ou elle soit capable de parler de la partie dont il ou elle s'est occupé et au moins vaguement de la partie laissée à l'autre. Les notes seront attribuées personnellement et non pas aux binômes.

**Paternité du code.** La seule personne autorisée à travailler avec vous sur votre projet est votre binôme. Vous êtes toujours autorisé à consulter les sources publiques telles que la documentation PHP, Stackoverflow, Github. Cependant, toute recopie de bout de code devra être précisé et ne devra concerner qu'une petite partie du code du projet. De plus, tout code réutilisé doit l'être légalement et doit porter la mention de sa source.

# 4 Recommandations

**Sur la sauvegarde : Pensez à faire des sauvegardes régulières** et à toujours avoir une version qui fonctionne sous la main. Parfois, en voulant corriger un bug, on casse autre chose et on est très soulagé d'avoir une ancienne version qui marche depuis laquelle repartir. Pensez également à faire des sauvegardes sur plusieurs supports : sur vos ordinateurs, au SCRIPT, sur une clé USB, sur un git ou svn... en cas de panne de l'un de ces supports, vous aurez toujours les autres en remplacement. Les enseignants de ce cours vous recommandent *fortement* de stocker votre code sur un serveur git, par exemple Gaufre, Gitlab ou Github.

**Sur la planification:** Le projet fait appel à des contenus du cours qui n'auront pas tous déjà été vus au moment où cet énoncé est donné. Vous ne pourrez donc pas, a priori, travailler sur tous les aspects du projet dès le début.

Que ceux qui veulent étaler leur effort se rassurent : c'est tout à fait possible.

Tout d'abord, avant même de commencer à coder, vous pouvez commencer par chercher le thème de votre projet, réflêchir à l'enchaînement des pages.

Pour vos mises en page HTML et CSS, faites un schéma du résultat désiré et identifiez ses sous-parties avant d'essayer de le coder.

Ensuite, les pages HTML et CSS peuvent être écrites avant d'y intégrer du PHP, puis la base de données. Le projet est aussi une opportunité de réviser son cours en le mettant en pratique au fur et à mesure que celui-ci avance.

Pour la partie en lien avec la base de données, réfléchissez aux extensions que vous souhaitez mettre en place **avant de commencer à coder**. Si vous ne vous y prenez pas en amont, certaines d'entre elles vous demanderont d'adapter lourdement votre base de données, ce qui vous fera perdre beaucoup de temps...

**Questions** Si vous avez des questions ou des suggestions, parlez-en à votre chargé(e) de TP. Un forum sera également mis en place sur Moodle.