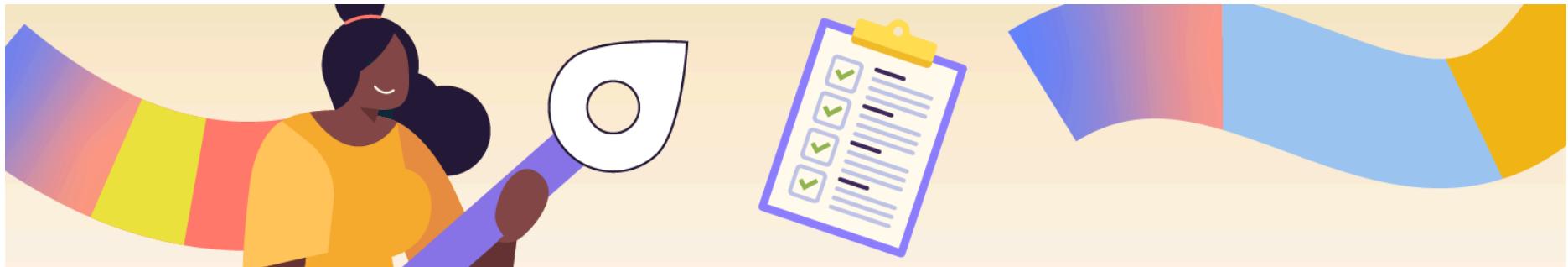


# Auto-évaluation

Mettez en place un site de mise en relation avec PHP

Option B – scénario fictif



Un dernier doute avant l'envoi de vos livrables ?

Pour vérifier la qualité de votre travail :

- cochez les cases ci-dessous : elles indiquent que vous avez bien pris en compte chaque indicateur de réussite ;
- renseignez, si besoin, la colonne "Notes" avec des commentaires sur vos livrables / vos étapes. Ils seront des points de discussion avec votre mentor pendant votre session de bilan / soutenance.

Quand toutes les cases de ce document seront cochées, vous pourrez déposer vos livrables sur la plateforme.

Bonne réussite !

Compétence + Livrable	Indicateurs de réussite de l'activité	Notes
<b>Développer et intégrer l'interface</b> Site web	<p><b>Conformité des interfaces graphiques</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Les pages correspondent aux maquettes Figma fournies.</li> <li><input type="checkbox"/> Toutes les pages demandées sont présentes.</li> <li><input type="checkbox"/> Il n'y a aucune règle CSS directement dans le code HTML/PHP, seulement l'utilisation de classes.</li> </ul> <p><b>Qualité des interfaces graphiques</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Le site passe le test sur W3C sans erreur (warnings autorisés).</li> <li><input type="checkbox"/> Le site passe le test sur WCAG niveau 2 (warnings autorisés).</li> <li><input type="checkbox"/> Les images, logos, icônes, polices, couleurs et textes sont exactement les bons et disposés aux mêmes endroits que sur la maquette.</li> <li><input type="checkbox"/> Les tailles et distances entre les éléments sont les mêmes que sur la maquette (avec une marge de tolérance de +/-10 %).</li> </ul> <p><b>Opérationnalité des interfaces</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Tous les liens présents sur le site sont fonctionnels.</li> <li><input type="checkbox"/> Accéder à une page qui n'existe pas amène sur une page d'erreur "404".</li> <li><input type="checkbox"/> Aucune page ne génère d'erreurs PHP.</li> <li><input type="checkbox"/> La partie admin n'est pas accessible aux utilisateurs lambda.</li> </ul> <p><i>Ne peut pas faire l'objet d'un refus de soutenance : le fait que le responsive ne soit pas ou mal réalisé.</i></p>	

<p><b>Développer les règles de gestion des données</b></p> <p>Site web</p>	<p><b>Conformité des règles de gestion des données</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Une base de données relationnelle a été utilisée (MySQL, MariaDB...). Les bases noSQL (ex. : mongoDB) sont proscrites.</li> <li><input type="checkbox"/> Les tables de la base de données possèdent des relations entre elles qui garantissent l'intégrité de la base de données (ex. : au niveau de la BDD, il est impossible de créer un post sans utilisateur valide).</li> </ul> <p><b>Opérationnalité des règles de gestion des données</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Le code PHP respecte les PSR : une marge d'erreur est tolérée tant que l'indentation est cohérente sur l'ensemble du projet.</li> <li><input type="checkbox"/> Les noms des classes, méthodes, variables et autres constantes sont cohérents et explicites.</li> <li><input type="checkbox"/> Le projet respecte OBLIGATOIUREMENT l'architecture MVC avec au minimum : <ul style="list-style-type: none"> <li><input type="checkbox"/> Un routeur : toutes les pages passent par un routeur (la POO n'est pas obligatoire, même un simple switch / if/else est autorisé).</li> <li><input type="checkbox"/> Contrôleurs : il y a un ou plusieurs contrôleurs, écrits en POO.</li> <li><input type="checkbox"/> Modèles : l'architecture entité/manager (ou entité/repository) a été utilisée. Les entités et les managers sont écrits en POO.</li> <li><input type="checkbox"/> Vue : un système a été mis en place pour éviter de dupliquer les header/footer sur chaque page (la POO n'est pas obligatoire).</li> </ul> </li> <li><input type="checkbox"/> Aucune librairie PHP n'a été utilisée. En particulier Twig, des routeurs clefs en main ou encore Doctrine sont interdits ici. Les outils pour la gestion du CSS sont en revanche autorisés (Tailwind, Bootstrap, Sass, Less...).</li> <li><input type="checkbox"/> Un fichier README est présent pour expliquer comment déployer le site.</li> </ul>	
--	---	--

	<p><i>Ne peuvent pas faire l'objet d'un refus de soutenance :</i></p> <ul style="list-style-type: none"> <li>• <i>l'utilisation ou non d'un autoloader ou des namespaces ;</i></li> <li>• <i>le fait que le projet soit codé en totalité, partiellement ou pas du tout en anglais ;</i></li> <li>• <i>la présence ou l'absence de commentaires.</i></li> </ul> <p><b>Respect des bonnes pratiques en gestion des données</b></p> <p><input type="checkbox"/> Des types cohérents sont utilisés dans la base de données (pas de varchar pour un champ qui doit contenir une date, par exemple).</p>	
<b>Utiliser un environnement de développement</b>  Site web	<p><b>Choix / maîtrise de l'environnement de travail</b></p> <p><input type="checkbox"/> Un IDE adapté au projet est installé sur le poste de l'étudiant (par exemple VSCode, PhpStorm, IntelliJ...).</p>	
<b>Collaborer avec l'équipe projet en utilisant Git et GitHub</b>  Site web	<p><b>Qualité du code</b></p> <p><input type="checkbox"/> Le projet est versionné sur Git.</p> <p><input type="checkbox"/> Le projet a été stocké sur un repo en ligne (ex. : GitHub ou GitLab).</p> <p><input type="checkbox"/> Le projet a été versionné régulièrement.</p> <p><input type="checkbox"/> Le projet présente bien un minimum de 5 commits durant le projet. Si l'étudiant ne valide pas ce critère, demandez-lui d'expliquer l'intérêt du versioning et demandez-lui de faire un commit sous 48 h après la soutenance.</p>	

	<p><b>Conformité / qualité du versioning</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Les messages de commit sur l'outil de versioning sont complets et permettent de tenir l'équipe technique informée de l'avancement du développement (à défaut, l'étudiant doit pouvoir expliquer l'intérêt d'avoir des messages de commit explicites).</li> </ul>	
<b>Effectuer des requêtes sur des données</b>  Site web	<p><b>Conformité de la communication avec la base de données</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Les données sont stockées dans une base de données relationnelle.</li> </ul> <p><b>Opérationnalité de la connexion à la base de données</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Il existe dans le code des requêtes SQL permettant de créer, récupérer, mettre à jour et supprimer des données de la base de données (CRUD).</li> </ul> <p><b>Respect des bonnes pratiques en échanges de données</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Les mots de passe stockés dans la base de données sont encodés.</li> <li><input type="checkbox"/> Les identifiants de connexion à la base de données ne sont pas visibles sur la dernière version du projet en ligne sur GitHub.</li> </ul> <p><i>Ne peut pas faire l'objet d'un refus de soutenance :</i></p> <ul style="list-style-type: none"> <li>• <i>Le fait qu'il soit possible de trouver les mots de passe à l'intérieur d'anciens commits.</i></li> </ul>	