Fiche projet

Écrivez une fiche pour chacun de vos trois à cinq projets retenus, en 150 à 300 mots au total, liens et étiquettes non compris.
Le nom entre parenthèses est la colonne de la base où ira chaque rubrique. Vous créerez ces colonnes au chapitre 1.06.
Une première version imparfaite suffit : vous la relirez avant de la mettre en ligne.

La fiche

Titre (`title`) · 2 à 6 mots

……

Résumé (`summary`) · une phrase, 15 à 25 mots

……

……

Contexte : pour qui, pourquoi, quand (`content`) · 25 à 50 mots

……

……

……

Mon rôle : seul ou en équipe, ce que j'ai fait moi-même (`content`) · 10 à 25 mots

……

……

Le problème (`content`) · 25 à 50 mots

……

……

……

Les choix techniques et pourquoi (`content`) · 40 à 70 mots

……

……

……

……

Le résultat : ce qui marche, ce qui reste inachevé, les données fictives (`content`) · 20 à 40 mots

……

……

Ce que je referais autrement (`content`) · 20 à 40 mots

……

……

Les captures

Largeur de fenêtre utilisée pour toutes vos captures : …… px

	Capture d'ensemble (`cover_path`)	Capture de détail
Nom du fichier		
Ce qu'elle montre		
Légende (une phrase)		
Texte alternatif (une phrase)		

Les liens et les étiquettes

	Votre réponse	Lien ouvert et vérifié
Lien démo (`demo_url`)		☐
Lien code (`repo_url`)		☐
README à jour dans le dépôt	☐ oui ☐ non	
Étiquettes, 3 à 6 technologies (`tags`)		

Nombre de mots de la fiche : ……

Exemple rempli

Si le refuge fait partie de vos projets, votre fiche raconte vos choix et vos difficultés, pas celles de l'exemple.

Titre · Le refuge du Brabant wallon

Résumé · Un site Laravel où trois refuges publient leurs animaux à adopter, et où les bénévoles tiennent les fiches à jour.

Contexte · Projet du cours Framework côté serveur (5XCOS) à l'IFOSUP Wavre, à l'automne 2026. Le scénario : trois refuges, à Wavre, Rixensart et Ottignies, partagent un même site pour présenter leurs chiens, leurs chats et leurs lapins.

Mon rôle · Seul. Le dépôt de départ fournissait la base de données, les modèles et les tests. J'ai écrit les routes, les contrôleurs, les vues et la validation.

Le problème · Un bénévole doit ajouter un animal, corriger sa fiche et le marquer adopté sans toucher au code. Un visiteur doit voir tout de suite les animaux encore disponibles, et seulement eux.

Les choix techniques et pourquoi · Blade et Tailwind CSS étaient imposés par le dépôt de départ. J'ai choisi une Form Request, pour écrire les règles une seule fois pour l'ajout et la modification. La liaison de modèle dans les routes répond par une erreur 404 à une adresse inconnue, sans code en plus.

Le résultat · Tous les tests fournis passent et la CI GitHub Actions est verte. La liste se filtre par espèce, et un animal adopté disparaît de la liste publique. Le site est en ligne chez OVH. Les refuges et les animaux sont fictifs.

Ce que je referais autrement · N'importe quel visiteur peut ajouter ou supprimer un animal, car le site n'a pas de compte. Je protégerais ces formulaires par une connexion. J'écrirais aussi un test pour le filtre, que j'ai seulement vérifié à la main.

Largeur de fenêtre utilisée pour toutes vos captures : 1280 px

	Capture d'ensemble (`cover_path`)	Capture de détail
Nom du fichier	`refuge-liste.png`	`refuge-erreurs.png`
Ce qu'elle montre	la page `/animaux` filtrée sur les chats	le formulaire d'ajout envoyé vide
Légende (une phrase)	La liste des animaux disponibles, filtrée sur les chats.	Le formulaire affiche une erreur en français sous chaque champ.
Texte alternatif (une phrase)	Grille de six cartes de chats avec leur nom, leur âge et leur refuge.	Formulaire d'ajout d'un animal avec trois messages d'erreur rouges.

	Votre réponse	Lien ouvert et vérifié
Lien démo (`demo_url`)	`https://refuge.example.com`	☒
Lien code (`repo_url`)	`https://github.com/votre-compte/refuge-26`	☒
README à jour dans le dépôt	☒ oui ☐ non	

	Votre réponse	Lien ouvert et vérifié
Étiquettes, 3 à 6 technologies (`tags`)	Laravel, Blade, Tailwind CSS, Pest, GitHub Actions	

Nombre de mots de la fiche : 239