

POUR ATTEINDRE LE SEUIL DE RÉUSSITE, L'ÉTUDIANT.E PROUVERA QU'IL/ELLE EST CAPABLE :

face au matériel et au logiciel adéquats et face à une structure informatique opérationnelle connectée à Internet, disposant des logiciels appropriés et de la documentation nécessaire, en utilisant le vocabulaire technique et l'orthographe adéquate, en respectant la législation, les normes et standards en vigueur, et au départ d'une proposition de projet individuel ou collectif d'application web dynamique formulée par le chargé de cours : - exploitant un « template » existant, - développant, par exemple, une des thématiques suivantes : e-commerce, site d'entreprise, sites institutionnel, culturel, associatif,... - prenant en compte les principes généraux d'accessibilité des applications web et de leurs contenus,

Acquis d'apprentissage	ACQUIS	NON ACQUIS
AA 1 : de produire un cahier des charges et son dossier technique (à titre d'exemples : planning, échéancier, techniques de développement, services à implémenter, développement futur, maintenance, formation, sécurité,...) AA 1.1 : Rédiger un rapport complet ☒ Niveau 1: Le rapport PDF est absent, incomplet (moins de 3 sections), ou ne respecte pas le format professionnel, rendant la compréhension du projet impossible. ☐ Niveau 2: Le rapport est présent mais manque de clarté, de détails ou de screenshots pertinents, ou la gestion des erreurs n'est pas documentée ou est rudimentaire. ☐ Niveau 3: Le rapport est complet, structuré professionnellement avec des screenshots de qualité, et inclut une section sur les difficultés/solutions et l'utilisation des outils IA, ainsi qu'une gestion des exceptions basique mais fonctionnelle. ☐ Niveau 4: Le rapport est exemplaire, détaillé, avec une analyse approfondie des choix techniques et des solutions, une gestion robuste des exceptions documentée, et un système de logs efficace est mis en place pour le débogage et la maintenance. AA 1.2 : Élaborer le diagramme de classe UML de la base de données ☒ Niveau 1: Le diagramme UML de la base de données est absent ou ne représente pas les tables essentielles (moins de 3), ou le schéma est fondamentalement incorrect. ☐ Niveau 2: Le diagramme UML est présent mais comporte des relations incorrectes, des champs essentiels manquants, ou n'est pas conforme aux standards, rendant le fonctionnement partiel ou erroné. ☐ Niveau 3: Le diagramme UML est clair, complet et conforme aux standards, représentant toutes les tables de base avec les champs nécessaires (identifiants, timestamps, contenu) et les relations correctes. ☐ Niveau 4: Le schéma de base de données est complet, optimisé (relations, cardinalités, indexes pertinents) et les choix de conception sont justifiés pour l'évolutivité et la performance.	☐	☒
AA 2 : de déployer l'application web dynamique répondant aux consignes/objectifs figurant dans la proposition du chargé de cours. AA 2.1 : Gérer le code source avec Git ☒ Niveau 1: Le dépôt GitHub est absent, privé ou l'historique est incompréhensible (ex: un seul commit final, mélange de code non pertinent). ☐ Niveau 2: Le dépôt GitHub est public et contient le code, mais l'historique est désordonné (commits génériques, branches non gérées), rendant le suivi difficile. ☐ Niveau 3: Le dépôt GitHub est public, avec un historique clair et des messages de commit pertinents, utilisant des branches de manière appropriée (ex: main, dev, feature). ☐ Niveau 4: Le dépôt GitHub est public, bien structuré, avec un historique détaillé et des messages de commit précis, incluant une gestion avancée des branches et des pull requests si applicable, démontrant une maîtrise des bonnes pratiques Git. AA 2.2 : Déployer l'application en ligne ☒ Niveau 1: L'application n'est pas déployée en ligne ou le lien fourni est non fonctionnel/inaccessible. ☐ Niveau 2: L'application est déployée, mais elle présente des erreurs critiques au démarrage, est très lente ou certaines fonctionnalités de base sont inopérantes en production. ☐ Niveau 3: L'application est déployée en ligne et est fonctionnelle, accessible et les fonctionnalités obligatoires opèrent correctement dans l'environnement de production. ☐ Niveau 4: L'application est déployée de manière robuste, avec une configuration optimisée pour la production (environnement, performances), une gestion des logs accessible et une haute disponibilité. AA 2.3 : Développer une application fonctionnelle, utilisable et sans bug critique ☒ Niveau 1: L'application est non fonctionnelle, inutilisable ou présente des bugs critiques majeurs qui empêchent l'utilisation des fonctionnalités obligatoires, ou utilise l'Options API de Vue.js (éliminatoire). ☐ Niveau 2: L'application est fonctionnelle pour les cas d'usage principaux, mais présente des bugs mineurs, des lacunes d'UX ou certaines fonctionnalités obligatoires ne sont pas entièrement implémentées ou stables. ☐ Niveau 3: L'application est entièrement fonctionnelle, utilisable, exempte de bugs critiques, et toutes les fonctionnalités obligatoires sont implémentées correctement, y compris le streaming des réponses et l'utilisation exclusive de la Composition API de Vue.js. ☐ Niveau 4: L'application est très robuste, hautement utilisable, exempte de bugs, avec une UX soignée et intègre des fonctionnalités supplémentaires qui dépassent les attentes, démontrant une maîtrise exceptionnelle des technologies et des bonnes pratiques de développement. AA 2.4 : Implémenter et optimiser la base de données ☒ Niveau 1: Les migrations sont absentes ou non fonctionnelles, empêchant la création ou l'évolution de la base de données, et/ou aucune contrainte d'intégrité référentielle n'est mise en place. ☐ Niveau 2: Les migrations sont présentes mais basiques, sans seeders, nécessitant une intervention manuelle significative pour initialiser les données, et/ou des contraintes d'intégrité basiques sont définies, mais les requêtes ne sont pas optimisées. ☐ Niveau 3: Les migrations sont complètes et fonctionnelles, permettant la création de la base de données, accompagnées de seeders simples pour les données de base, et l'intégrité référentielle est correctement appliquée avec quelques optimisations de requêtes. ☐ Niveau 4: Les migrations sont optimisées, utilisant des factories et des seeders pour générer des jeux de données de test réalistes et diversifiés, et la base de données est optimisée par l'utilisation judicieuse de contraintes CASCADE, soft deletes, eager loading et/ou mécanismes de cache. AA 2.5 : Appliquer une architecture logicielle robuste et tester l'application ☒ Niveau 1: La logique métier est directement imbriquée dans les contrôleurs, et le code Vue est désorganisé, sans aucune réutilisabilité ou structure claire, et/ou aucun test automatisé n'est implémenté. ☐ Niveau 2: Une tentative de séparation est visible, mais la logique métier reste partiellement dans les contrôleurs, ou les composants Vue manquent de modularité et de cohérence, et/ou des tests sont présents mais sont incomplets, non fonctionnels ou ne couvrent pas les scénarios critiques. ☐ Niveau 3: Les services Laravel sont extraits des contrôleurs pour gérer la logique métier, et les composants Vue sont modulaires et réutilisables, améliorant la maintenabilité, et des tests automatisés (Laravel Dusk) sont implémentés pour vérifier la navigation et les fonctionnalités principales. ☐ Niveau 4: L'application présente une architecture claire avec une séparation complète et cohérente des responsabilités (MVC, services, repositories, etc.), démontrant une maîtrise avancée des patterns de conception et de la réutilisabilité, et des tests automatisés complets couvrent les pages, les formulaires, le streaming des réponses et les cas limites (edge cases), assurant une robustesse maximale.	☐	☒

Tous les AA sont « acquis » RÉUSSITE → Voir degré de maîtrise	Au moins un AA non-acquis (en 1ère session) AJOURNEMENT → Voir justification ajournement	Au moins un AA non-acquis (en 2ème session) REFUS → Voir justification refus
--	---	---

DEGRÉ DE MAÎTRISE (% ≥ 50) - 0= Acquis, 1 = Bien, 2 = TB, 3 = TTB, 4 = Excellent et 5 = Parfait

Critères	Indicateurs	Niveau
N°1 Qualité et Maintenabilité du Code	☒ 0/5 (0/10) : Code illisible, aucune convention, duplications massives, rendant le projet incompréhensible. 1/5 (2/10) : Code difficile à lire, conventions ignorées/inconsistantes, duplications majeures, maintenance complexe. 2/5 (4/10) : Efforts d'organisation mais inconsistances, quelques duplications, "code smells" impactant clarté/maintenabilité. 3/5 (6/10) : Code globalement propre, lisible, respecte les conventions, principe DRY, améliorations mineures possibles. 4/5 (8/10) : Code très propre, cohérent, adhère aux standards (ex: PSR-12), bonne gestion des duplications, maintenance facile. 5/5 (10/10) : Code exemplaire, hautement lisible, maintenable et extensible, reflétant une attention méticuleuse aux détails et bonnes pratiques.	0/10
N°2 Innovation et Originalité	☒ 0/5 (0/10) : Aucune fonctionnalité supplémentaire ou touche personnelle. 1/5 (2/10) : Tentative de fonctionnalité supplémentaire non fonctionnelle ou cassée. 2/5 (4/10) : Une fonctionnalité supplémentaire basique et fonctionnelle (ex: export de chat, recherche simple). 3/5 (6/10) : Deux à trois fonctionnalités supplémentaires fonctionnelles, ajoutant une valeur notable ou une touche personnelle. 4/5 (8/10) : Plusieurs fonctionnalités innovantes (ex: outils LLM intégrés, multi-modalité, sujet original) implémentées de manière robuste. 5/5 (10/10) : Implémentation exceptionnelle de fonctionnalités, démontrant créativité, maîtrise technique et une vision originale (ex: agents complexes, application "coach/psy" unique).	0/10
N°3 Expérience Utilisateur (UX) et Design	☒ 0/5 (0/7.5) : Interface cassée, inutilisable, ou design inexistant. Page de marketing absente ou illisible. 1/5 (1.5/7.5) : Interface minimale, utilisation incorrecte/limitée de Tailwind, non responsive. Page de marketing basique ou non professionnelle. 2/5 (3/7.5) : Design correct mais générique, réactivité partielle, UX basique. Page de marketing simple mais fonctionnelle. 3/5 (4.5/7.5) : Interface agréable, entièrement responsive, animations fluides, bonne UX. Page de marketing claire et attractive. 4/5 (6/7.5) : UX soignée, entièrement responsive, micro-interactions pertinentes, attention aux détails. Page de marketing professionnelle et engageante. 5/5 (7.5/7.5) : Design exceptionnel, réactivité parfaite, UX comparable aux meilleures applications. Page de marketing créative, impactante et parfaitement réalisée.	0/7.5
N°4 SEO et Accessibilité	☒ 0/5 (0/7.5) : Aucun effort pour le SEO ou l'accessibilité. Mentions légales/cookie banner absents. 1/5 (1.5/7.5) : Tentative minimale de SEO (ex: un titre de page) ou accessibilité. Mentions légales/cookie banner incomplets ou non fonctionnels. 2/5 (3/7.5) : Balises méta (titre, description) et OG tags basiques. Quelques efforts d'accessibilité (ex: contraste). Mentions légales/cookie banner présents mais basiques.	0/7.5

Critères	Indicateurs	Niveau
	3/5 (4.5/7.5) : SEO correct (méta-informations complètes, OG tags pertinents). Bonnes pratiques d'accessibilité (ARIA, navigation clavier) pour les éléments clés. Mentions légales et cookie banner fonctionnels. 4/5 (6/7.5) : SEO optimisé (schéma.org, sitemap, robots.txt). Accessibilité avancée (tests WCAG basiques, gestion du focus, rôles ARIA complets). Mentions légales détaillées et cookie banner personnalisable. 5/5 (7.5/7.5) : Stratégie SEO exemplaire et intégrée. Accessibilité de niveau expert (conformité WCAG, tests automatisés). Implémentation irréprochable des mentions légales et du consentement aux cookies.	
N°5 Qualité du Rapport Technique	✓ 0/5 (0/7.5) : Rapport bâclé, incomplet, captures floues, structure chaotique. 1/5 (1.5/7.5) : Rapport minimaliste, nombreuses fautes, diagrammes rudimentaires, présentation non professionnelle. 2/5 (3/7.5) : Structure acceptable, contenu superficiel. Diagramme UML basique, manque de détails ou conformité. 3/5 (4.5/7.5) : Rapport bien structuré, diagrammes corrects, analyse pertinente mais manque de profondeur ou professionnalisme. 4/5 (6/7.5) : Rapport professionnel, diagramme UML détaillé et conforme, analyse incluant auto-critique constructive des choix et difficultés. 5/5 (7.5/7.5) : Qualité du rapport digne d'une publication, diagrammes UML parfaits, analyse approfondie des choix, défis, solutions, présentation impeccable.	0/7.5
N°6 Maîtrise Théorique et Justification des Choix (Oral)	✓ 0/5 (0/7.5) : Réponses hors sujet ou incapacité à répondre aux questions théoriques ou sur le projet. 1/5 (1.5/7.5) : Confusion sur les concepts de base (ex: MVC, ORM) ou incapacité à justifier les choix techniques. 2/5 (3/7.5) : Réponses correctes but superficielles, sans profondeur technique ni explication claire des avantages/inconvénients. 3/5 (4.5/7.5) : Bonne compréhension des concepts, mais manque de détails techniques ou de précision dans la justification des choix. 4/5 (6/7.5) : Réponses précises, vocabulaire technique approprié, choix techniques bien justifiés avec arguments solides. 5/5 (7.5/7.5) : Expertise démontrée, référence aux meilleures pratiques, justification approfondie et critique des choix, même sur sujets complexes.	0/7.5

CALCUL DE LA NOTE FINALE		
Seuil de réussite (tous AA acquis) :		0 / 50
Total des critères de maîtrise :		0 / 50
NOTE FINALE :		0 / 100