

Prénom, Nom : Ambre Bafour Équipe : 7						
Niveau / Qualité de mobilisation: Compétences Mobilisés par le projet (simplifiés / adaptés)*	Pas mobilisé	Découverte	Notions	Application	Maitrise	Expertise
Application / Démo						
C02.1	Complexité : Résoudre un problème complexe en mobilisant les concepts, méthodes et outils informatiques et mathématiques adaptés.		Application statique.	API sample sans identification utilisateur.	Appli avec authentification et API protégés.	Appli avec schéma des données avancées et/ fonctionnalités originales (son, cartes, vidéos...).
C03.2	Schema des Données & BackEnd : Concevoir et modéliser le stockage de données		UML ou schema papier simple.	UML réaliste.	Schema implementé et BD instanciée.	BD optimisé.
C04.8	FrontEnd : système utile, avec interface cohérente et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation)		Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complète.	Une front basic mais qui fonctionne.	Front poussé avec composants complexes (ex animations, site responsives) et/ou réflexion et re-conception après retours des utilisateurs.
Développement / Code						
C04.10	Qualité Code : Code stable, et qui répond aux besoins demandés [...]		Code qui se n'exécute pas.	Code qui tourne avec quelques bugs.	Code complexe qui tourne sans bugs.	Code avec des bonnes pratiques respectées (structure, genericité, noms et contenus des fonctions, ...), et bien commenté.
C04.7	Test : Tester un logiciel : concevoir, planifier et exécuter un plan de validation logiciel [...]		Code pas testé.	Test partielle manuel et/ou avec des utilisateurs.	Tests unitaires faites.	Cahier des tests.
Rapport						
C04.3 C06.1	Cahier : Traduire des fonctionnalités attendues en cahiers des charges. [...] Avoir une réflexion sur le cahier initial les évolutions. Veille : Tenir en compte de la concurrence (veille scientifique / technologique).		Liste des fonctionnalités haute niveau, pas de planification.	Cahier des charges non-exhaustive, planification pas refléchi. Idée de ce qui existe dans le marché (concurrence).	Cahier des charges exhaustive, sans planification ou planification pas refléchi.	Cahier des charges exhaustive avec planification détaillée, avec perspectives d'évolution dans le cahier des charges, étude des concurrences.
C14.1 C04.4	Réflexion : Développer une pratique réflexive sur son projet. Argumenter pour ces décisions: Bien argumenter sur la pertinence du projet, les choix technologique et fonctionnels, l'organisation du travail. Donner des références/citations des sources utilisées.		Pas de réflexion personnel et/ou pas de réflexion en groupe.	Réflexion superficielle sur les choix (technologiques, conception, structure).	Réflexion approfondie sur les choix (techno, conception, solution), sur la démarche (travail en équipe, gestion, planification), sur le cahier des charges (honnêteté). Bien citer les ressources utilisées.	Etre capable de faire une Retour d'Expérience (REX) sur chaque partie du projet (technologie, gestion, front/back, groupe) et être capable d'attirer des leçons dès le prochain projet.
Communication						
C11.2	Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...]		Équipe pas préparé.	Avoir fait effort de préparation, mais présentation dehors les limites / contraintes données.	Être capable de présenter le projet dans les contraintes données.	Être capable de convaincre de la maîtrise de son projet (son valeur, les choix faits, etc).
Travail en équipe						
C04.2	Coordination et planification: Mettre en œuvre une méthodologie de projet, planifier votre travail [...]. Communiquer régulièrement la progression auprès des acteurs (ex professeurs). Communiquer la progression à l'écrit.		Pas présent en cours, pas de communication.	Communiquer dans la classe.	Communiquer dans la classe et mini-rapports envoyés.	Communiquer clairement (en verbal + mini-rapport) sur l'avancement et la répartition des tâches à chaque séance. Mettre-en-place de cérémonies de type agile et l'expliquer.
C04.9	DevOps: Gérer le cycle de vie logiciel tout au long des phases de planification, de développement, [...] selon les pratiques DevOps, et mettre en place des architectures orientées services		Absence des outils de gestion (code, tâches).	Outils initiés mais peu utilisés.	Outils de gestion code+projet mis-en-place et utilisés régulièrement.	Avoir un git propre (branches, ...) et connection avec Jira et tâches, mise en place d'un Docker.
						Avoir une CI/CD - Chaîne d'intégration Chaîne de déploiement complet.