

Rapport d'activité – Projet *All Together*

Date : 10 octobre 2025

Avancement de la journée

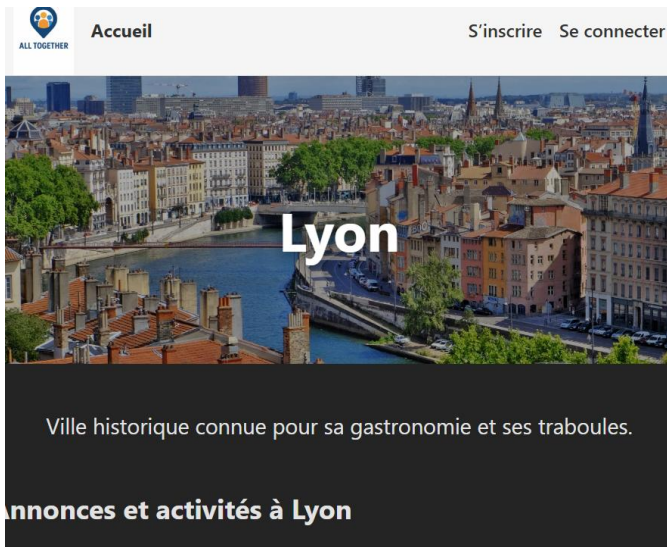
- **Pablo :**
 - Avancement sur le font:

création des pages se connecter et s'inscrire avec liaison entre les deux page



The image shows a web application mockup for 'All Together'. At the top, there is a navigation bar with a logo on the left and links for 'Accueil', 'S'inscrire', and 'Se connecter' on the right. Below the navigation bar is a large black rectangular area containing a white login form. The form is titled 'Se connecter' in bold. It features two input fields: 'Email' and 'Mot de passe', each with a dark grey placeholder bar. Below these fields is a blue button with the text 'Se connecter'. At the bottom of the form, there is a link that says 'Pas encore de compte ? S'inscrire'.

créations des pages des différentes villes (basé sur la futur base de donnée) un seul fichier est créer pour ces pages et se base sur des données provisoir pour ne pas avoir à créer une page par ville



Type : Bar

Auto évaluation:

Prénom, Nom :							
Équipe :							
Niveau / Qualité de mobilisation:		Pas mobilisé	Découverte	Notions	Application	Maîtrise	Expertise
Compétences Mobilisées par le projet (simplifiées / adaptées)*							
Application / Démonstration							
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 protégées.	Appli avec schéma des données avancés et/ou fonctionnalités originales (son, cartes, vidéos...).	Appli déployé.
C03.2		Schema des Données & BackEnd : Concevoir et modéliser le stockage de données.	UML ou schéma papier simple.	UML réalisée.	Schema implementé et BD instantie.	BD optimisé.	Efforts au niveau moteur de recherche, schema complexe, etc.
C04.8		FrontEnd : système utile, avec interface cohérent et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation).	Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complete.	Une front basic mais qui fonctionne.	Un front complete, testé par d'utilisateurs.	Front poussé avec composants complexes (ex animations, site responsives) et/ou reflection 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 respectés (structure, generateurs, noms et contenu des fonctions, ...), et bien commenté.	Code optimisé et possibilité de communication avec des services extérieurs.
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.	Tests end-to-end (e2e), tests TNR.
Rapport							
C04.3		Cahier : Traduire des fonctionnalités attendues en cahiers de charges. [...].	Liste des fonctionnalités haute niveau, pas de planification.	Cahier des charges non-exhaustive, planification pas réfléchi.	Cahier des charges exhaustive, sans planification ou planification pas réfléchi, l'absence de ce qui existe dans le marché (concurrence).	Cahier des charges exhaustive avec planification, connaissance des concurrences.	Cahier des charges exhaustive avec planification détaillée, dans le cahier des charges, étude des concurrences.
C04.4		Reflexion : Développer une pratique réflexive sur son projet. Argumenter pour ces décisions: Bien argumenter sur la pertinence du projet, les choix technologiques et fonctionnels, l'organisation du travail. Donner des références/citations des sources utilisés.	Pas de réflexion personnel et/ou pas de réflexion en groupe.	Reflexion superficielle sur les choix technologiques, conception, structure).	Reflexion approfondis sur les choix (techno, conception, solution), sur la démarche (travail en équipe, gestion, planification), sur le cahier des charges (honnêtes). Bien citer les ressources utilisés.	Justifier les choix du projet et personnel avec pertinence, esprit critique, auto-évaluation.	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 des le prochaine projet.
Communication							
C11.2		Communication: Communiquer et convaincre en s'adaptant aux objectifs et contraintes [...].	Equipe pas préparé.	Avoir fait effort de preparation, mais presentation dehors les limites / contraintes données.	Etre capable de présenter le projet dans les contraintes données.	Etre capable de convaincre de la maîtrise de son projet (son valeur, les choix faites, etc).	Etre capable de convaincre de la maîtrise de son projet et aussi son propre expertise sur le sujet.
Travail en équipe							
C04.2		Coordination et planification: Mettre en œuvre une méthodologie de projet, planifier votre travail [...]. Communiquer régulièrement la progrès auprès des acteurs (ex professeurs). Communiquer la progression à l'écart. 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.	Pas present en cours, pas de communication.	Communiquer dans la classe.	Communiquer dans la classe et mini-rapports envoyés.	Communiquer sur l'avancé et identification des points bloquants.	Communiquer clairement (en verbal mini-rapport) sur l'avancé et la répartition des tâches à chaque séance. Mettre en place de cérémonie de type agile et l'expliquer.
C04.9			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 CICD - Chaîne d'intégration Chaîne de Déploiement complet.

- **Paolo :**

- Sécurité et Authentification

AuthController : Endpoints register/login avec gestion JWT

JwtTokenProvider : Génération et validation des tokens JWT

SecurityConfig : Configuration Spring Security avec rôles

Gestion des rôles : Différenciation USER vs ADMIN

- Gestion des Utilisateurs

UserProfile : Entité principale avec rôles et sécurité Spring

UserProfileService : Service complet avec CRUD et validation

UserProfileController : API REST avec permissions admin

DTOs spécialisés : UserCreateDTO, UserUpdateDTO, UserResponseDTO

- Gestion des Lieux

PlaceService : Service de gestion des lieux avec catégories

PlaceController : API REST avec recherche par ville/catégorie

Sécurisation : Lecture pour tous, écriture réservée aux admins

- Système de Participations

ParticipationService : Gestion des inscriptions aux événements

ParticipationController : API pour gérer les participations

Prévention des doublons : Un utilisateur ne peut pas s'inscrire deux fois au même événement le même jour

Contrainte d'unicité : Implémentée au niveau base de données et métier

- Sécurité mise en place

Permissions :

- USER : Lecture lieux + Gestion propres participations
- ADMIN : Tous les droits (CRUD utilisateurs, lieux, etc.)

Protections :

- Validation des données (email, âge, mots de passe)

- Tokens JWT avec expiration
- Gestion fine des permissions par endpoint
- Prévention des injections et accès non autorisés

Auto évaluation:

Niveau / Qualité de mobilisation: Compétences Mobilisées par le projet (simplifiées / adaptées)*	Pas mobilisé	Découverte	Notions	Application	Maîtrise	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 protégées.	Appli avec schéma des données avancés et fonctionnalités originales (son, cartes, vidéos...).	Appli déployé.
C03.2	Schema des Données & BackEnd : Concevoir et modéliser le stockage de données	UML ou schéma papier simple.	UML réaliste.	Schema implémenté et BD instancié.	BD optimisée.	Efforts au niveau moteur de recherche, schéma complexe, etc...
C04.8	FrontEnd : système utile, avec interface cohérent et utilisable, construite avec une approche centrée-utilisateur (conception, évaluation)	Prototype papier ou page Figma simple.	Figma (ou autre prototype) site complet.	Une front basic mais qui fonctionne.	Un front complet, testé par d'utilisateurs.	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 [...] Test : Tester un logiciel : concevoir, planifier et exécuter un plan de validation logiciel [...]	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 contenu des fonctions, ...) et bien commenté.	Code optimisé et possibilité de communication avec des services extérieurs.
C04.7		Code pas testé.	Test partielle manuel et/ou avec des utilisateurs.	Tests unitaires faites.	Cahier des tests.	Tests end-to-end (e2e), tests TNR.
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 concurrence (veille scientifique / technologique).	Liste des fonctionnalités haute niveau, pas de planification.	Cahier des charges non-exhaustive, planification pas réfléchi.	Cahier des charges exhaustive, sans planification ou planification pas réfléchi. Idée de ce qui existe dans l'existant (concurrents).	Cahier des charges exhaustive avec planification, connaissance des concurrences.	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 utilisés.	Pas de réflexion personnel et/ou pas de réflexion en groupe.	Reflexion superficielle sur les choix (technologiques, conception, structure).	Reflexion approfondis 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.	Justifier les choix du projet et personnel avec pertinence, esprit critique, auto-évaluation.	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 faites, etc).	Être capable de convaincre de la maîtrise de son projet et aussi son propre expertise sur le sujet.
Travail en équipe						
C04.2	Coordination et planification: Mettre en oeuvre une méthodologie de projet, planifier votre travail [...]. Communiquer régulièrement la progrès 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 sur l'avancé et identification des points bloquants.	Communiquer clairement (en verbal + mini-rapport) sur l'avancement et la répartition des tâches à chaque séance. Mettre-en-place de cérémonie 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 (codé, 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 CICD - Chaîne d'Intégration Chaîne de Déploiement complet.

Jira:

The screenshot displays a Jira board with three columns: 'A FAIRE' (4 items), 'EN COURS' (2 items), and 'TERMINÉ(E)' (11 items). Each column contains task cards with titles, due dates, and status indicators.

Column	Count	Task Title	Due Date	Status	Assignee
A FAIRE (4)	1	carte interactive	11 nov. 2025	Not Started	AL-20
	2	fin projet	11 nov. 2025	In Progress	AL-14
	3	créer filtre de recherche		Not Started	AL-31
EN COURS (2)	1	passer tests http		In Progress	AL-29
	2	améliorer squelette page		In Progress	AL-30
TERMINÉ(E) (11)	1	Ajout planning détaillé au cahier des charges	26 sept. 2025	Completed	AL-24
	2	Ajouter les endpoints manquants :		Completed	AL-26
	3	authentification	1 nov. 2025	Completed	AL-15
	4				