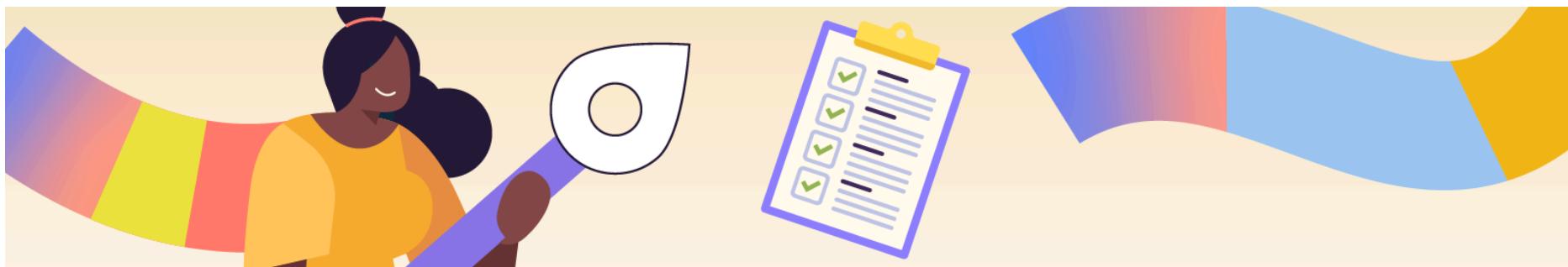


Autoévaluation

Améliorez et ajoutez des fonctionnalités



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 !

Étape	Indicateurs de réussite de l'activité	Notes
<i>Étape 1 – Analysez le code existant</i>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai installé toutes les technologies nécessaires : Java, Angular, Docker <input type="checkbox"/> J'ai réussi à lancer le back-end et le front-end <input type="checkbox"/> L'application fonctionne correctement en créant un utilisateur agent. <input type="checkbox"/> J'ai exploré la structure du projet (dossiers, fichiers, logique générale). <input type="checkbox"/> J'ai pris des notes ou fait un schéma pour mieux comprendre l'architecture full-stack. <input type="checkbox"/> Je peux expliquer la structure et le fonctionnement du code existant à orale (par ex. lors d'une session de mentorat). 	
<i>Étape 2 – Corrigez l'API d'authentification des utilisateurs côté back-end</i>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai analysé le code existant avant de faire toute modification. <ul style="list-style-type: none"> <input type="checkbox"/> J'ai identifié les fichiers contenant des bugs <input type="checkbox"/> J'ai noté les comportements inattendus ou les erreurs visibles. <input type="checkbox"/> J'ai apporté les corrections nécessaires pour que l'API <code>/api/login</code> fonctionne. Désormais, elle retourne un token JWT. <input type="checkbox"/> J'ai testé cette API avec succès sur Postman. 	

<p><i>Étape 3 – Implémentez l'interface d'authentification côté front-end</i></p>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai implémenté un écran d'authentification simple avec login et mot de passe. <input type="checkbox"/> J'ai connecté le nouvel écran à la route d'authentification <ul style="list-style-type: none"> <input type="checkbox"/> J'ai implémenté un service Angular pour appeler l'API. <input type="checkbox"/> J'ai vérifié la sécurisation de l'écran. <input type="checkbox"/> J'ai testé la fonctionnalité du nouvel écran. <input type="checkbox"/> J'ai géré les erreurs serveur (ex. mauvais identifiants) et les états (chargement, erreur, succès). 	
<p><i>Étape 4 – Ajoutez de nouvelles fonctionnalités côté back-end</i></p>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai implémenté les routes nécessaires pour gérer les étudiants (fonctions CRUD) : <ul style="list-style-type: none"> <input type="checkbox"/> ajouter un étudiant <input type="checkbox"/> consulter la liste des étudiants et les informations détaillées d'un étudiant <input type="checkbox"/> modifier un étudiant <input type="checkbox"/> supprimer un étudiant <input type="checkbox"/> Toutes les routes de l'API (existantes et nouvelles) fonctionnent correctement. <input type="checkbox"/> J'ai vérifié la sécurisation des routes pour assurer l'accès uniquement aux utilisateurs authentifiés. <input type="checkbox"/> J'ai testé chaque route avec succès sur Postman. <input type="checkbox"/> J'ai respecté les bonnes pratiques de développement back-end en Java, notamment le découpage du code en suivant le modèle en couche. 	

*Étape 5 –
Implémentez les
écrans côté
front-end*

- J'ai implémenté les écrans nécessaires pour gérer les étudiants (en consommant les API CRUD du back-end) :
 - ajouter** un étudiant
 - consulter** la liste des étudiants et les informations détaillées d'un étudiant
 - modifier** un étudiant
 - supprimer** un étudiant
- J'ai vérifié la sécurisation des écrans pour assurer l'accès uniquement aux utilisateurs authentifiés.
- J'ai testé tous les écrans et vérifié leur fonctionnement côté l'utilisateur.
- J'ai respecté les bonnes pratiques de développement front-end en Angular, notamment l'utilisation des services Angular.

(Pour ce projet, il n'a pas besoin de respecter de règles particulières concernant l'apparence visuelle ou l'accessibilité.)

Autoévaluation

Effectuez des tests unitaires et d'intégration



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 !

Étape	Indicateurs de réussite de l'activité	Notes
<i>Étape 1 – Analysez le code de test fourni dans le back-end</i>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai relu les fichiers de tests déjà présents dans le back-end. <input type="checkbox"/> J'ai identifié les classes, services et controllers déjà testés. <input type="checkbox"/> J'ai compris les logiques de test utilisées dans ces fichiers. <input type="checkbox"/> J'ai exécuté les tests fournis pour vérifier leur comportement. <input type="checkbox"/> J'ai pris des notes sur les bonnes pratiques à réutiliser pour mes futurs tests. 	
<i>Étape 2 – Écrivez un plan de tests</i>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai listé les cas de tests nécessaires pour valider le fonctionnement du back-end, du front-end et du bout au bout. <input type="checkbox"/> J'ai ajouté des tests unitaires pour les composants et les services. <input type="checkbox"/> J'ai organisé mes cas de test par priorité (simple → complexe). <input type="checkbox"/> J'ai distingué les entrées attendues et les sorties attendues pour chaque test. 	
<i>Étape 3 – Implémentez les tests unitaires et d'intégration du back-end</i>	<ul style="list-style-type: none"> <input type="checkbox"/> J'ai lu les documentations sur JUnit et sur Mockito pour monter en compétence sur ces technologies. <input type="checkbox"/> J'ai écrit des tests unitaires pour les nouveaux services du back-end. <input type="checkbox"/> J'ai écrit des tests d'intégration pour les nouveaux controllers. 	

	<input type="checkbox"/> J'ai documenté mes tests avec des commentaires utiles. <input type="checkbox"/> J'ai atteint au moins 80 % de couverture de test sur le back-end. <input type="checkbox"/> J'ai généré de rapports de couverture de test indiquant le seuil de 80 % minimum.	
<i>Étape 4 – Implémentez les tests unitaires et d'intégration du front-end</i>	<input type="checkbox"/> J'ai lu les documentations sur Jest pour monter en compétence sur cette technologie. <input type="checkbox"/> J'ai écrit des tests unitaires pour les services Angular et pour les composants Angular. <input type="checkbox"/> J'ai vérifié que tous les tests s'exécutent avec succès. <input type="checkbox"/> J'ai atteint au moins 80 % de couverture de test sur le front-end avec Jest. <input type="checkbox"/> J'ai généré de rapports de couverture de test indiquant le seuil de 80 % minimum.	
<i>Étape 5 – Implémentez les tests E2E du front-end</i>	<input type="checkbox"/> J'ai lu les documentations sur Cypress pour monter en compétence sur cette technologie. <input type="checkbox"/> J'ai écrit des tests E2E pour les écrans principaux de l'application. <input type="checkbox"/> J'ai commencé par les formulaires simples (inscription, connexion). <input type="checkbox"/> J'ai vérifié que tous les tests E2E passent sans erreur. <input type="checkbox"/> J'ai atteint au moins 80 % de couverture E2E sur les parcours utilisateurs. <input type="checkbox"/> J'ai généré de rapports de couverture de test indiquant le seuil de 80 % minimum.	