

## 1. OBJECTIF

Créer une **Application de Suivi de Tutorats d'Apprentis (ASTA)**.

## 2. EXPRESSION DU BESOIN

Vous devez remplacer un système existant de suivi du tutorat d'apprentis, basé sur un fichier Excel, par une application Web.

			Remarques / Exemples de valeur
<b>Apprenti</b>	Programme		M2-PRO
	Année académique		2023-2024
	Majeure		Digital Transformation
	Nom		
	Prénom		
	Adresse électronique		
	Téléphone		10 chiffres
<b>Entreprise</b>	Raison sociale		
	Adresse		
	Informations utiles pour accéder aux locaux		Badge, pièce d'identité, étage...
<b>Maître d'apprentissage</b>	Nom		
	Prénom		
	Poste		
	Adresse électronique		
	Téléphone		
	Remarques		
<b>Mission</b>	Mots-clés		DevOps, node.js, CI/CD, Tests
	Métier cible (réf. Cigref)		
	Commentaires		
<b>Visite</b>	Date		
	Format		Visio / Présentiel
	Commentaires		
<b>Evaluation-école</b>	<b>Mémoire / Rapport</b>		
		Thème/Sujet	
		Note finale	
		Commentaires	
	<b>Soutenance</b>		
		Date	
		Note finale	
		Commentaires	
<b>Remarques</b>			
<b>Feedback du tuteur-enseignant</b>			Commentaires libres du TE

Dans le tableau ci-dessus vous trouverez l'ensemble des données disponibles dans le système actuel.

- ✓ A minima toutes ces données doivent être intégrées dans la nouvelle application.
- ✓ Vous pouvez proposer de nouveaux champs si vous le souhaitez.
- ✓ Un seul profil utilisera l'application : le tuteur enseignant

### 3. FONCTIONNALITES PRINCIPALES A REALISER

1. Ouverture d'une session via un écran de connexion permettant de saisir le login et le mot de passe :
  - 1.1. Réaffichage de la même page en cas d'erreur de connexion
  - 1.2. Message d'erreur si les informations de connexion ne sont pas valides
  - 1.3. Une fois la connexion validée, affichage en haut à droite de chaque page de :  
Bonjour [Prénom] | Déconnectez-vous
  - 1.4. Un clic sur Déconnectez-vous provoque la fermeture de la session et l'affichage de la page de connexion.
2. Affichage d'un tableau de bord synthétique avec la liste des apprentis suivis par le tuteur-enseignant pour l'année en cours.
  - 2.1. Si aucun apprenti ne figure dans le tableau, affichez le message "La liste est vide. Ajoutez au moins un apprenti" (en rouge)
3. Pour chaque apprenti, possibilité d'afficher toutes les informations sur une page dédiée.
4. Modification de tous les champs dans la page de détails (page dédiée du §3) :
  - 4.1. Affichage du message "Modification réussie !" en cas de réussite de la modification
  - 4.2. Et affichage du message "Échec de la modification" en cas d'erreur en indiquant idéalement le(s) champ(s) qui sont en erreur et un message d'erreur clair et utile pour chacun de ces champs.
5. Ajout d'apprenti.
  - 5.1. Ajout via la saisie
  - 5.2. Ajout via l'import de données (bonus)
6. Création d'une nouvelle année académique.
  - 6.1. Les apprentis de I3 sont archivés.
  - 6.2. Les apprentis de I1 et I2 sont promus : I1 vers I2 et I2 vers I3
  - 6.3. Les apprentis archivés ne doivent plus remonter dans la liste des apprentis de l'année en cours.
7. Recherche.
  - 7.1. Lancement d'une recherche à partir d'un des critères suivants :
    - 7.1.1. Nom
    - 7.1.2. Entreprise
    - 7.1.3. Mot-clé de la mission
    - 7.1.4. Année académique.

### REMARQUES :

- 1) Vous pouvez proposer de nouvelles fonctionnalités, mais pensez toujours à faire valider votre idée par l'enseignant avant toute mise en œuvre.
- 2) Pour le design visuel, vous avez carte blanche. Vous pouvez donc donner libre cours à votre créativité quant au volet UX Design (esthétique, navigation, structure des pages, etc...)

### 4. EXIGENCES TECHNIQUES

- 1) L'utilisation de Spring Boot est obligatoire.
- 2) Les stacks de votre application doivent être *a minima* identiques aux stacks du TP fil rouge.
- 3) Documentez vos API avec Swagger ou tout autre outil équivalent.
- 4) Déployez votre application sur un serveur de votre choix. Elle sera ainsi accessible via une URL et ce, à partir de n'importe quel ordinateur relié à Internet.
- 5) Gestion des **exceptions** :
  - a. Suivez les consignes contenues dans le fichier *ALTN72 - Clean code - Gestion des exceptions*.
  - b. Entre *ControllerAdvice* et *ResponseStatusException* choisissez l'option qui vous semble la plus pertinente, cohérente et efficace pour gérer les exceptions.
  - c. Dans tous les cas, vous devez créer des exceptions personnalisées avec des messages (en français) pertinents.
  - d. Si et seulement si l'utilisation de *ResponseEntity* vous paraît pertinente, alors exploitez cette classe dans votre code et soyez prêts à expliquer ce choix lors de la soutenance.
- 6) Persistance
  - a. Utilisez un mélange de requêtes prédéfinies et de requêtes explicites (en JPQL)
  - b. Utilisez une seule fois une requête SQL standard. Dans votre rapport, faites une analyse critique de cette façon de faire.
  - c. Si vous utilisez l'annotation *@Transactional*, alors faites-le de manière pertinente et soyez prêts à justifier chaque utilisation de cette annotation.
- 7) BONUS : (éligible => non automatique)
  - a. Réalisez une IHM exceptionnelle, intuitive, ergonomique et cohérente et visuellement remarquable : *Max. +1*.
  - b. Une idée originale ? Par exemple, vous souhaitez construire la Vue avec une technologie différente de Thymeleaf ? Vous voulez mettre en œuvre une technologie non vue dans le cadre du cours mais pertinente et efficace ? Votre approche par rapport au Clean Code est conforme aux meilleures pratiques en vigueur ? Vous avez pensé à des fonctionnalités innovantes ? : *Max.+2*

### 5. LIVRABLES

1. Le **code source** de votre projet Spring Boot.
2. Un script SQL pour recréer vos tables (à intégrer dans votre projet).
3. Un fichier *Rapport.md*, intégré dans votre projet, et contenant les informations suivantes :
  - 3.1. Un login et un mot de passe pour tester votre application.
  - 3.2. Des informations relatives à votre outillage :
    - 3.2.1. Quel IDE avez-vous utilisé ?
    - 3.2.2. Quel SGBD avez-vous choisi ? Indiquez le nom et la version. Par exemple *MariaDB 10.4.28*
  - 3.3. Toutes les informations que vous estimez utiles pour lancer et tester votre application.
  - 3.4. La réponse aux questions suivantes :
    - a) Sur quel aspect de votre travail souhaitez-vous attirer l'attention du correcteur ?
    - b) Quelle est la plus grande difficulté que vous avez rencontrée ? Comment avez-vous géré/solutionné/contourné cette difficulté ?
    - c) Quelle a été la contribution de chaque membre de l'équipe ?
    - d) Si vous deviez retenir trois points de ce cours en général et de ce projet en particulier, quels seraient ces trois points ?
    - e) Les fonctionnalités que vous n'avez pas eu le temps de mettre en œuvre et pourquoi.
    - f) A quel niveau, dans votre projet, avez-vous réussi à respecter entièrement ou partiellement les principes SOLID ?

### Que faut-il uploader dans l'espace dédié au dépôt, sur Moodle ?

Au choix :

- Un fichier texte que vous devrez nommer de manière que je puisse l'identifier sans ambiguïté (exemple : *ALTN72\_projet\_Jacques\_AUGUSTIN.txt*).

**OU**

- Un fichier \*.zip contenant votre code, si vous ne souhaitez pas le déposer sur un repository

Un seul et unique dépôt par équipe, s'il vous plaît !

### 6. CRITERES D'EVALUATION

- ✓ Respect des instructions
- ✓ Nombre de fonctionnalités réalisées
- ✓ Degré de finition des fonctionnalités réalisées
- ✓ Réponses aux questions durant la soutenance (35 %)
- ✓ Autonomie
- ✓ Capacité à travailler harmonieusement en équipe

### 7. DATE LIMITE POUR DEPOSER VOTRE PROJET

**30 octobre 2025 à 23h59.**

**Amusez-vous et bon projet à toutes et à tous !**

