

## 1 INFORMATIONS GENERALES

Candidat :	Nom :	Prénom :
Lieu de travail :	ETML	
Chef de projet :	Nom :	Prénom :
Nombre de périodes :	24p	
Procédure :	<p>Les projets sont réalisés individuellement.</p> <p>Le projet est évalué selon la grille d'évaluation.</p> <p>Les apprentis ont de la semaine 1 à la semaine 5 pour travailler et finir le projet.</p> <p>A la semaine 5 du projet, les apprentis devront fournir leur projet (code et rapport) et leur auto-évaluation.</p> <p>Lors des semaines 6 et 7, les apprentis seront en remédiation du projet ou en approfondissement et devront rendre leur rapport final à la fin de la semaine 7.</p>	

## 2 PROCÉDURE

- Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le premier jour de l'épreuve.
- Le cahier des charges est commenté et discuté avec l'élève. Par sa signature, l'élève accepte le travail proposé.
- L'élève a connaissance de la grille d'évaluation avant de débiter le travail.
- Il est entièrement responsable de la sécurité de ses données.
- En cas de problèmes graves, le candidat avertit son chef de projet au plus vite.
- Le candidat a la possibilité d'obtenir de l'aide externe et doit le mentionner dans son rapport final.
- Les projets sont réalisés seul
- Le projet est évalué selon la grille d'évaluation.
- Les apprentis ont de la semaine 1 à la semaine 5 pour travailler et finir le projet.
- A la semaine 5 du projet, les apprentis devront fournir leur projet et leur auto-évaluation.
- Lors des semaines 6 et 7, les apprentis seront en remédiation du projet ou en approfondissement et devront rendre leur rapport au chef de projet (semaine 7).
- Le projet compte pour 24p.

### 3 Création d'un site d'e-commerce sécurisé (webstore)

Les apprentis choisiront chacun un thème de commerce en ligne et réaliseront le site. Ce projet se concentre sur la page d'authentification (des clients et des administrateurs du site). Pour la création du compte admin, vous avez le choix de la procédure, mais celle-ci devra être justifiée.

Dans une première étape du projet, les apprentis réaliseront eux-mêmes l'authentification (en émettant un token JWT). L'émission du token JWT ne se fera que si l'utilisateur a été correctement identifié par votre serveur. Le mot de passe sera protégé par un hashage et un salage.

Pour cette protection du mot de mot de passe, il sera interdit d'utiliser bcrypt, il faudra coder cela vous-même cette protection. Lorsque ceci sera codé et fonctionnel, vous implémenterez aussi bcrypt et comparerez les deux solutions dans votre rapport.

L'architecture de votre site respectera les normes que vous avez étudié lors des modules précédents.

Le projet doit impérativement se concentrer sur les aspects login et sécurité du site.

Résumé des pages :

- Une page d'authentification (pour tous les utilisateurs) qui permet de se connecter
- Une page d'administration (visible pour les administrateurs uniquement) et qui permet d'effectuer une recherche parmi les utilisateurs
- La page de bienvenue affiche les détails de l'utilisateur connecté (elle appellera la route backend /api/users/nom\_utilisateur)

### 4 OBJECTIF DU PROJET

Au terme du projet, l'apprenti sera capable de construire une application node.js offrant un accès sécurisé et une gestion des rôles.

### 5 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un ordinateur standard de la section informatique avec Docker Desktop

### 6 CAHIER DES CHARGES

#### 6.1 Dockerisation

L'ensemble des services web sera conteneurisé.

#### 6.2 Profil du client

Le client peut accéder à son propre profil depuis la page de bienvenue.

Seul son profil lui sera rendu visible et les routes du backend seront protégées.

#### 6.3 HTTPS

Il doit être possible d'accéder à votre site de e-commerce de manière sécurisée (<https://localhost>). Le port utilisé sera le 443. Le certificat sera auto-signé par OpenSSL.

## 6.4 Authentification par mot de passe

L'utilisateur devra s'authentifier par la page <https://localhost/login>

Le mot de passe sera haché et salé avant d'être stocké dans la base de données (table `t_users`).

## 6.5 Vérification du token JWT

Le token devra être vérifié lorsque l'utilisateur vous l'envoie et la signature du jeton devra aussi être utilisée pour vérifier que le token n'a pas été modifié.

## 6.6 Administration

Une page d'administration devra avoir un champ de recherche (Nom du visiteur) et permettre d'afficher tous les utilisateurs ayant tout ou partie de ce nom.

## 6.7 Protection contre les injections SQL

Votre page d'administration devra être protégée contre les injections SQL.

Sans utiliser `sequelize` ou tout autre ORM, votre site devra être robuste face aux injections. L'utilisation de `sequelize` (ou autre ORM) sera autorisée dans l'approfondissement du projet.

## 6.8 Utilisation de `bcrypt`

Une fois l'authentification faite avec votre propre mécanisme (mot de passe en base de données, salé et haché). Vous modifierez votre code pour utiliser **`bcrypt`**. Votre code doit prévoir une condition pour utiliser soit `bcrypt` soit votre propre implémentation.

## 6.9 Versioning

Votre code sera versionné sur Github et un `.gitignore` empêchera de versionner les binaires npm (dossiers `node_modules`, documents word/excel). Votre dépôt sera partagé avec votre chef de projet dès le début du projet.

## 6.10 Documentation

- Journal de travail (1 ligne par quart d'heure de travail)
- Rapport, contenant :
  - Votre conceptualisation (schéma)
  - Des explications sur votre code
  - Une conclusion sur le travail fourni et sur l'attitude face au projet

## 6.11 Utilisation de l'IA

Une utilisation injustifiée de l'IA ou une absence de maîtrise du code, induira un non-acquis dans les compétences techniques du projet.

**7 VALIDATION**

	Lu et approuvé le :	Signature :
Candidat :		
Chef de Projet :		