Examen de Fin de Formation (*Session Blanche*)

Préparer par : M. DAIF Othmane

Secteur : Digital et Intelligence Artificielle

Niveau : Technicien Spécialisé

Filière : Développement Digital — Option Web Full Stack

Durée : 4 h 00 **Barème :** /100

Consignes

- Lisez la totalité du sujet et le préliminaire avant de commencer.
- Répondez à toutes les questions ; une réponse cohérente est toujours valorisée.
- Pour les exercices de calcul (PERT), la méthode correcte est prise en compte même si le résultat final est faux ; justifiez toujours votre raisonnement.
- Soignez la lisibilité de vos schémas et de votre code.

Préliminaire

La société **« AgriSmart »** exploite un réseau de serres connectées destinées à la production agricole. Chaque serre est équipée de capteurs (température, humidité, luminosité…) qui mesurent en continu les conditions de culture. Pour garantir la fiabilité des mesures, AgriSmart suit l'état de charge et la précision de chaque capteur, et planifie son remplacement avant qu'il ne se dérègle. Les remplacements sont effectués par des techniciens.

Schéma du modèle logique de données (MLD) :

Serre(id, nom, localisation, superficie, #id_capteur_principal,date_derniere_inspection, date_prochaine_inspection)

Capteur(id, numero_serie, type, precision, niveau_charge,nombre_mesures, statut)

Intervention(id, #id_serre, #id_ancien_capteur, #id_nouveau_capteur,date_intervention, #id_technicien, motif)

Technicien(id, nom, telephone, email, specialite)

NB : la clé primaire est le premier champ souligné de chaque table ; la clé étrangère est précédée du signe #.

Partie Théorique (40 points)

Dossier 1 : Création d'une application Cloud native (8 points)

1- Définir le cloud public et le cloud privé, et citer un fournisseur pour chacun. (2 pts)

2- Qu'est-ce qu'un microservice ? Donner un avantage de l'architecture microservices. (2 pts)

3- On souhaite travailler avec Docker :

a- Donner la commande pour télécharger l'image officielle de MongoDB depuis Docker Hub. (2 pts)

b- Donner la commande pour lister les conteneurs en cours d'exécution. (2 pts)

Dossier 2 : Préparation d'un projet web (6 points)

Donner le diagramme de cas d'utilisation correspondant au système suivant : (6 pts)

Le système AgriSmart sera utilisé par trois acteurs : l'Administrateur, le Technicien et l'Agriculteur.

- L'agriculteur peut consulter l'état de ses serres et consulter les alertes le concernant.
- Le technicien peut consulter la liste des capteurs, enregistrer une intervention, et planifier la prochaine inspection.
- L'administrateur peut faire tout ce que fait le technicien, et peut en plus gérer les serres (ajouter, supprimer) et gérer les comptes utilisateurs.
- L'accès au système est sécurisé : chaque utilisateur doit s'authentifier pour accéder à ses fonctionnalités.

Dossier 3 : Approche Agile et gestion de projet (7 points)

1- Citer les trois rôles de la méthode Scrum. (1,5 pt)

2- Mettre dans l'ordre les étapes de gestion d'un projet : Tests, Conception, Cahier des charges, Réalisation, Analyse, Mise en production. (1,5 pt)

3- Soit le tableau des tâches du projet AgriSmart ci-dessous :

Tâche A : Analyse des besoins — durée 3 jours — antériorité : aucune.
Tâche B : Conception de la base de données — durée 2 jours — antériorité : A.
Tâche C : Développement back-end — durée 5 jours — antériorité : B.
Tâche D : Développement front-end — durée 4 jours — antériorité : B.
Tâche E : Tests et recette — durée 2 jours — antériorités : C et D.
Tâche F : Déploiement — durée 1 jour — antériorité : E.

a- Dresser le diagramme PERT en précisant les dates au plus tôt et au plus tard. (2 pts)

b- Déterminer le chemin critique et la durée minimale du projet. (2 pts)

Dossier 4 : Gestion de données NoSQL — MongoDB (11 points)

1- Créer la base de données « AgriDB » et la collection « capteurs », puis insérer le document suivant : (3 pts)

```
{
  "_id": "c1",
  "numero_serie": "CAP-100",
  "type": "Humidité",
  "niveau_charge": 85,
  "statut": "En service",
  "serre": { "id_serre": "1", "nom": "Serre A1" },
  "mesures": [
    { "_id": "m1", "valeur": 62, "unite": "%" }
  ]
}
```

On suppose que la collection « capteurs » contient un ensemble de documents. Écrire les requêtes MongoDB permettant de :

2- Afficher les capteurs triés par niveau de charge décroissant. (2 pts)

3- Afficher le nombre de capteurs dont le type est « Humidité ». (2 pts)

4- Mettre à jour le statut du capteur d'identifiant « c3 » à la valeur « Défaillant ». (2 pts)

5- Supprimer tous les capteurs dont le niveau de charge est inférieur à 10. (2 pts)

Dossier 5 : Web dynamique PHP (8 points)

On dispose d'un tableau PHP **$capteurs** où chaque élément est un tableau associatif contenant les clés **numero_serie**, **type**, **niveau_charge** et **statut**.

1- Écrire un script PHP qui parcourt le tableau $capteurs et affiche le nombre de capteurs dont le niveau de charge est inférieur à 20. (4 pts)

2- Écrire un script PHP qui calcule et affiche la moyenne des niveaux de charge de l'ensemble des capteurs. (4 pts)

Partie Pratique (60 points)

Dossier 1 : Gestion des données MySQL (12 points)

En vous basant sur le MLD du préliminaire, écrire les scripts MySQL répondant aux questions suivantes :

1- Écrire le script de création de la table « Capteur ». (2 pts)

2- Modifier la table « Capteur » en ajoutant une colonne « date_installation » de type DATE, non nulle. (2 pts)

3- Écrire le trigger « TR_charge » qui empêche l'insertion d'un capteur si son niveau de charge n'est pas compris entre 0 et 100. (2 pts)

4- Écrire la fonction « fct_chargeFaible » qui reçoit l'identifiant d'un capteur et retourne 1 si son niveau de charge est inférieur à 15, sinon 0. (2 pts)

5- Écrire la procédure « P_capteursParStatut » qui affiche le nombre de capteurs pour chaque statut. (2 pts)

6- Gestion des utilisateurs et des rôles : (2 pts)

a- Créer le rôle « RoleTechnicien ».

b- Donner les droits d'ajout, de modification et de suppression sur les tables « Capteur » et « Intervention » au rôle « RoleTechnicien ».

c- Créer l'utilisateur « sara » avec le mot de passe « 1234 ».

d- Attribuer le rôle « RoleTechnicien » à l'utilisateur « sara ».

Dossier 2 : Développement Front-End (24 points)

Soit l'état initial du store Redux suivant (le store, les actions et les reducers sont déjà créés ; il ne vous est pas demandé de les écrire) :

```
InitialState = {
  serre: {
    id: 7,
    nom: "Serre A1",
    localisation: "Meknès",
    date_derniere_inspection: "01/05/2026",
    date_prochaine_inspection: "01/08/2026",
    CapteurPrincipal: {
      id: 42,
      numero_serie: "CAP-310",
      type: "Humidité",
      precision: 98,
      niveau_charge: 76,
      nombre_mesures: 15240,
      statut: "En service"
    }
  },
  Techniciens: [ { id: ..., nom: ... }, { id: ..., nom: ... }, ... ],
  Capteurs:    [ { id: ..., numero_serie: ... }, ... ],
  StatutCapteur: [ "En service", "En maintenance", "Défaillant" ]
};
```

1- Écrire la composante « DetailsSerre.js » qui lit les informations de la serre depuis le store Redux (en utilisant useSelector) et affiche : le nom de la serre, sa localisation, ses dates d'inspection, ainsi que le numéro de série, le type et le niveau de charge du capteur principal. (4 pts)

2- Écrire la composante « Menu.js » qui définit, à l'aide de NavLink, les liens menant aux composantes : « Détails Serre » (/DetailsSerre), « Liste Capteurs » (/ListeCapteurs) et « Nouveau Capteur » (/AjouterCapteur). (4 pts)

3- Écrire le code de « App.js » qui : définit le routage (BrowserRouter, Routes, Route), appelle la composante Menu.js, et intègre le Provider du store Redux. (4 pts)

4- Écrire une composante « Formulaire.js » permettant de saisir le numéro de série, le type et le niveau de charge d'un nouveau capteur. Le clic sur le bouton « Confirmer » affiche le récapitulatif des informations saisies sur la même page. (4 pts)

5- Initialiser la variable du state « capteurs » de la composante principale App.js en appelant l'API du back-end suivante : (4 pts)

> **Méthode HTTP :** GET
>
> **URL de l'API :** http://localhost:8000/capteurs
>
> **Résultat :** un tableau d'objets capteur au format JSON (id, numero_serie, type, niveau_charge, statut).

6- Créer la composante « ListeCapteurs.js » qui reçoit le tableau des capteurs et l'affiche sous forme de liste à puces (numéro de série, type et niveau de charge de chaque capteur). (4 pts)

Dossier 3 : Développement Back-End (24 points)

1- Donner les commandes de création des fichiers de migration : « create_techniciens_table », « create_serres_table », « create_capteurs_table ». (1,5 pt)

2- Donner le code des méthodes up() et down() du fichier de migration de la table « capteurs », en définissant le champ « numero_serie » comme unique et le champ « statut » avec la valeur par défaut « En service ». (3 pts)

3- Donner la commande permettant d'appliquer les migrations. (1 pt)

4- Donner les commandes de création des modèles : Technicien, Serre, Capteur. (1,5 pt)

5- Donner le code du modèle « Capteur » sachant qu'un capteur appartient à une serre et qu'une serre possède plusieurs capteurs (préciser la propriété fillable et les relations). (4 pts)

6- Créer le contrôleur de ressource « CapteurController » en écrivant les méthodes index (afficher tous les capteurs), store (ajouter un capteur) et destroy (supprimer un capteur). (6 pts)

7- Créer un fichier seeder pour la table « capteurs » et écrire la méthode run() permettant d'ajouter un capteur avec des données de votre choix. (3 pts)

8- Donner la route de type resource permettant d'associer toutes les actions du CapteurController. (2 pts)

9- Donner le code de la méthode du modèle « Serre » permettant de récupérer la liste de tous les capteurs d'une serre donnée. (2 pts)