

**Examen National de Fin de Formation**  
**Session de Juin 2025**

**Examen de Fin de Formation (Epreuve de Synthèse)**

<b>Secteur :</b>	<b>Digital et Intelligence Artificielle</b>	<b>Niveau :</b>	<b>Technicien Spécialisé</b>
<b>Filière :</b>	<b>Développement Digital option Web Full Stack</b>		
<b>Variante</b>	<b>V2</b>	<b>Durée :</b>	<b>4H00</b>
		<b>Barème</b>	<b>/100</b>

**Consignes et Conseils aux candidats :**

- Toutes les réponses devront être justifiées avec le détail des calculs qui doit être indiqué sur la copie ;
- Apporter un soin particulier à la présentation de votre copie ;

**Document(s) et Matériel(s) autorisés :**

- Les documents ne sont pas autorisés ;
- Calculatrice simple (non programmable) autorisée.

@LearningM3ana

0774 03 15 01

**Détail du Barème :**

N° Des Dossiers	Travaux à réaliser	Barème
<b>Partie Théorique</b>		
<b>Dossier 1</b>	Création d'une Application Cloud native	<b>8 pts</b>
<b>Dossier 2</b>	Préparation d'un projet web	<b>6 pts</b>
<b>Dossier 3</b>	Approche Agile	<b>7 pts</b>
<b>Dossier 4</b>	Gestion de données NOSQL	<b>11 pts</b>
<b>Dossier 5</b>	Web Dynamique PHP	<b>8 pts</b>
<b>Total partie Théorique</b>		<b>/40 points</b>
<b>Partie Pratique</b>		
<b>Dossier 1</b>	Gestion des données MySQL	<b>12 pts</b>
<b>Dossier 2</b>	Développement Front End	<b>24 pts</b>
<b>Dossier 3</b>	Développement Back End	<b>24 pts</b>
<b>Total partie Pratique</b>		<b>/60 points</b>
<b>Total Général</b>		<b>/100 points</b>

**Dossier 1 Création d'une Application Cloud native (8 pts) :**

1. A quoi sert l'outil Postman ? (2pts)
2. Comment télécharger une image Docker depuis Docker Hub ? (2pts)
3. Comment arrêter un conteneur Docker en cours d'exécution ? (2pts)
4. Quels sont les services cloud (as a service) offerts par Azure ? (2pts)

**Préliminaire :**

La société « **FastFoodDelivery** » fournit à ses employés des trottinettes électriques pour assurer la livraison des colis auprès de ses clients. Pour garantir la continuité et la qualité de ses services, elle a décidé de mettre en place un système de suivi de l'état des batteries de ses trottinettes. Ce système permettra d'anticiper le remplacement des batteries avant qu'elles ne se détériorent ou tombent en panne.

**Trottinette**(id, numero\_serie, #id\_batterie, date\_derniere\_maintenance, date\_prochaine\_maintenance)  
**Batterie**(id, Numero\_serie, capacite, sante\_batterie, nombre\_cycles, statut)  
**Remplacement**(id, #id\_trottinette, #id\_ancienne\_batterie, #id\_nouvelle\_batterie, date\_replacement, #id\_reparateur, raison)  
**Reparateur**(id, nom, telephone, email, specialite)

\*NB : La clé primaire est écrite en souligné, la clé étrangère est écrite en #

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

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

Le système sera utilisé par trois utilisateurs : **Administrateur, Réparateur et Livreur**

Le réparateur peut **signaler** qu'une trottinette a été maintenue, **enregistrer** les remplacements des batteries, **saisir** une planification de la prochaine maintenance et **consulter** la liste des trottinettes

L'administrateur en plus des fonctionnalités du réparateur peut gérer les trottinettes en les **ajoutant** au système, **consulter** leur liste et **relancer** le contrôle des trottinettes s'ils ne sont pas faits à temps

L'accès au système est sécurisé, et chaque utilisateur doit s'authentifier pour accéder à ses fonctionnalités respectives.

Le livreur à un accès libre en **consultation** des trottinettes

Filière	DDOWFS	Variante	2	Page 2 sur 8
Session	Fin de Formation	Session	Juin 2025	

### Dossier 3 : Approche Agile (7 pts)

Soit la liste des tâches suivantes :

Tâche	Durée En jours	Tâches Antérieurs
A	2	-
B	1	A
C	3	A
D	4	B, C
E	2	D
F	1	E

- 1- Dresser le diagramme PERT. (3pts)
- 2- Trouver le Chemin critique. (1pt)
- 3- Déterminer la durée d'exécution du projet. (1pt)
- 4- La tâche B a pris 4 jours au lieu de 1 jour prévu : (2pts)
  - a. Déterminer le nouveau chemin critique
  - b. Déterminer la nouvelle durée du projet

### Dossier 4 : Gestion de données NOSQL (11 pts)

- 1) Créer une base de données appelée **FastFoodDelivery** et une collection nommée **batteries**. (1pt)
- 2) Insérer les documents suivants dans la collection batteries : (2pts)

```
[  
  {"_id":"1", "numéro_série":"s12", "capacité":"20%",  
   "santé_batterie":"bonne", "nombre_cycle":300, "statut":"en service"},  
  
  {"_id":"2", "numéro_série":"s245", "capacité":"30%",  
   "santé_batterie":"moyenne", "nombre_cycle":700, "statut":"en panne"}  
]
```

- 3) Modifier la capacité et le statut de la batterie dont l'id est 2 par les valeurs respectives suivantes : 45%, « Entretien ». (2 pts)
- 4) Afficher les attributs **nombre\_cycles**, **santé\_batterie** et **capacité** des batteries dont la capacité est différente de 50% classées par **nombre\_cycles** dans un ordre décroissant. (2pts)
- 5) Afficher le nombre de batteries regroupées par **nombre\_cycles**. (2pts)
- 6) Supprimer les batteries qui n'ont pas le statut *en service* ou *entretenu*. (2pts)

### Dossier 5 : Web Dynamique PHP (8 pts)

- 1) Donner le code PHP de la création de la classe **Batterie** avec le champ **nombre\_cycles** défini privée. (1pt)
- 2) Implémenter le getter **getNombreCycles** et le setter **setNombreCycles** pour la propriété **nombre\_cycles**. (1pt) :
- 3) Ajouter à la classe **Batterie** une méthode **batterieDétériorée**, qui retourne vrai si sa capacité est inférieure à 30% et retourne faux dans le cas contraire. (2pts)
- 4) Créer un tableau de deux objets **Batteries** avec des données de votre choix. (2pts)
- 5) Parcourir le tableau de la question 4 pour afficher le nombre de batterie détériorée en utilisant la méthode **batterieDétériorée** de la question 3. (2 pts)

**Dossier 1 : Gestion des données MySQL (12 pts)**

En se basant sur le schéma de base de données ci-dessus, écrire les scripts MySQL qui répondent aux questions suivantes :

- 1- Ecrire le script de la création de la table 'Trottinette'. (2pts)
- 2- Modifier la table 'Trottinette' en ajoutant la colonne 'nb\_batteries' qui est de type entier positif et non nul. (2pts)
- 3- Ecrire un trigger TR1 qui contrôle la valeur de la colonne 'sante\_batterie' qui doit être un nombre compris entre 0 et 100, lors de l'ajout d'une nouvelle batterie. (2pts)
- 4- Ecrire un trigger TR2, qui modifie la valeur de la colonne statut de 'En Utilisation' à 'Retiré'. Lors d'un ajout dans la table 'Remplacement'. (2pts)
- 5- Ecrire une fonction FCT1 qui retourne le nombre total de batteries consommées par une Trottinette dont l'id est passé en paramètre. (2pts)
- 6- Gestion des utilisateurs/Rôles : (2pts)
  - a. Créer le rôle 'RoleReparateur'
  - b. Donner les droits d'ajout, suppression et modification sur les tables 'Trottinette' et 'Remplacement' au rôle 'RoleReparateur'
  - c. Créer l'utilisateur 'Fatima' avec le mot de passe 1111
  - d. Attribuer le rôle 'RoleReparateur' à l'utilisateur 'Fatima' déjà créée

Filière	DDOWFS	Variante	2	Page 4 sur 8
Examen	Fin de Formation	Session	Juin 2025	

## Dossier 2 : Développement Front End (24 pts)

Soit l'état initial du *store* redux :

```
InitialState = {  
  Trottinette : {  
    id: 14,  
    Matricule: "1 A 4321",  
    Date_derniere_maintenance: "01/06/2025",  
    Date_prochaine_maintenance: "01/08/2025",  
    BatterieUtilisee: {  
      Id : 135,  
      Capacite : 73,  
      Numero_serie : "BAT-202",  
      sante_batterie: 95,  
      nombre_cycles: 2127,  
      statut: "En Utilisation",  
    }  
  },  
  ,Reparateurs : [ {id :...,Nom :...,...}, {id :...,Nom :...;...},...],  
  ,Batteries : [ {id :...,Numero_serie :...,....}, {id :..., Numero_serie:...,....},],  
  ,StatutBatterie : [ "En Stock","En Utilisation","Retiré"]  
};
```

Remarque : Il n'est pas demandé à vous de créer le store Redux (actions, reducer, store).

1. Ecrire la composante **DetailsTrottinette.js** qui lit les informations depuis le store Redux -En utilisant `useSelector`- et les affiche comme indiqué dans l'image ci-dessous. -(voir figure 1). (4pts)

# Trottinette Details

**Matricule: 1 A 4321**

**Capacité Batterie Actuelle: 73%**

**Santé Batterie Actuelle: 95%**

**Date Dernière Maintenance: 01-06-2025**

**Date Prochaine Maintenance: 01-08-2025**

Figure 1- Détails d'une Trottinette (Composante `DetailTrottinette.js`)

Filière	DDOWFS	Variante	2	Page 5 sur 8
Examen	Fin de Formation	Session	Juin 2025	

2. Ecrire le code de la composante **AjouterRemplacement.js** qui ajoute un Remplacement de la batterie dans la base de données à travers l'API en utilisant les données du store pour remplir le formulaire (voir figure 2) : (6pts)

Méthode http	POST
URL de l'API	http://localhost:8000/api/AjouterRemplacement
Structure Objet	<pre>{   "id_trottinette":...,   "id_ancienne_batterie":...,   "id_nouvelle_batterie":...,   "date_Remplacement":...,   "id_Reparateur":...,   "raison":... }</pre>

## Remplacement de Batterie

Trottinette: 1 A 4321

Ancienne Batterie: BAT-202

Nouvelle Batterie:

Date de Remplacement:

Reparateur:

Raison:

**Ajouter**

PS :

- Tous les champs sont obligatoires.
- Les valeurs du champs (Trottinette, ancienne batterie, liste batteries, liste Reparateurs) sont lues depuis le store redux

Figure 2- Ajout d'une Remplacement d'une batterie

Filière	DDOWFS	Variante	2	Page 6 sur 8
Examen	Fin de Formation	Session	Juin 2025	

3. Créer la composante **ListeBatteries.js** qui lit la liste des batteries depuis le store redux et l'affiche selon la forme du figure 3 : (6pts)

## Liste Batteries

**Choisir le Statut de la Batterie:**

Id	Numero Serie	Statut	Nombre Cycles
232	BAT-11	En Stock	6
16	BAT-19	En Stock	24

**2 batteries Trouvées**

**La moyenne des nombres de cycles est 15**

*Figure 3 : Liste des Batteries - Lue depuis le store Redux*

- La liste déroulante (select) lit les données depuis le store redux (StatutBatterie).
- Après clique sur le bouton 'Filtrer' Les données sont filtrées selon l'option choisie au niveau de la liste déroulante.
- En Bas du tableau, on affiche le nombre total d'éléments trouvés ainsi que la moyenne des nombres de cycles

4. Ecrire la composante **Menu.js** qui définit les liens qui mènent vers les composantes : (4pts)

Lien	Composante	Titre
/AjouterRempl	AjouterRemplacement.js	Nouveau Remplacement
/ListeBatteries	ListeBatteries.js	Liste Batteries
/DetailsTrottinette	DetailsTrottinette.js	Détails Trottinette

5. Ecrire le code **App.js** qui : (4pts)
- Définit le routage (BrowserRouter, Routes, ...)
  - Appel la composante Menu.js crée ci-dessus.
  - Integer le provider du store redux

### Dossier 3 : Développement Back End (24 pts)

- 1) Donner la commande de la création de migration pour la table Batteries. (1pt)
- 2) Donner le code des méthodes *up* et *down* du fichier de la migration de la table Batteries en définissant le champ *numéro\_série* comme unique et le champ *capacité* par défaut à «100%». (2pts)
- 3) Donner la commande pour appliquer la migration. (1pt)
- 4) Donner le code pour créer un fichier *seeder* pour la table Batterie. (1pt)
- 5) Donner le code de la méthode *run* pour assurer l'ajout d'une batterie avec des données de votre choix. (2pts)
- 6) Donner la commande pour appliquer le seeder et peupler la table batteries. (1pt)
- 7) Créer les modèles des tables Batteries et Trotinettes avec leurs attributs et fonctions de relations. (2pts)
- 8) Donner la commande de la création du contrôleur TrottinetteController. (1pt)
- 9) Créer la méthode *GetBatterie* dans le contrôleur TrottinetteController qui retourne les informations de la batterie d'une trottinette dont l'id est passé en paramètre. (2pts)
- 10) Créer la méthode *AfficherBatterie* dans le contrôleur TrottinetteController qui retourner une vue nommée *InfosBatterie*(serra créer dans la question 11) avec les données de la méthode *GetBatterie* de la question 9 (2pts)
- 11) Créer la vue *InfosBatterie* qui affiche les informations de la batterie sous forme d'un tableau comme le montre l'exemple ci-dessous avec les liens modifier et supprimer (2pts)

id	Numéro série	Capacité	Santé Batterie	Nombre Cycle	Statut	Modifier	Supprimer
1	S2	90	Bonne	270	En Panne	<u>Modifier</u>	<u>Supprimer</u>

- 12) Créer la méthode *SupprimerBatterie* dans le contrôleur TrottinetteController qui supprime une batterie dont l'id est passé en paramètre et qui redirige l'utilisateur vers la vue *Welcome*. (2pts)
- 13) Donner la commande pour créer un middleware sous le nom «*autorisationSuppression*». (1pt)
- 14) Donner le corps de la méthode *handle* pour que le middleware *autorisationSuppression* accepte les requêtes entre 8H30 et 18H30 et dans le cas contraire on affiche un code d'erreur 403 avec le message « Les suppression sont permises entre 8H30 et 18H30 ». (2pts)
- 15) Donner le code de la route qui mène vers l'action de suppression de batterie en la protégeant par le middleware «*autorisationSuppression*». (2pts)