

Examen National de Fin d'année
Session de Report 2023

8/11/2023

Examen de Fin de Formation (Epreuve de Synthèse)

Secteur :	Digital et Intelligence Artificielle	Niveau :	Technicien Spécialisé
Filière :	Développement Digital Option Web Full Stack		

Variante	V2	Durée :	4h00	Barème	/100
-----------------	----	----------------	------	---------------	------

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.

Détail du Barème :

Question	Barème
Dossier 1	8pts
Q1	2
Q2	3
Q3	1.5
Q4	1.5
Dossier 2	14pts
Q1	2.5
Q2	2.5
Q3	5
Q4	4
Dossier 3	12pts
Q1	2.5
Q2	2.5
Q3	2
Q4	2.5
Q5	2.5
Dossier 4	6pts
Q1	6

Question	Barème
Dossier 5	12pts
Q1	3
Q2	3
Q3	3
Q4	3
Dossier 6	23
Q1	2
Q2	2
Q3	3
Q4a	2
Q4b	2
Q4c	2
Q4d	2
Q5	4
Q6	4

Question	Barème
Dossier 7	25pts
Q1	4
Q2a	3
Q2b	3
Q3a	5
Q3b	5
Q4	5

Total	100
--------------	------------

Mohammed TAHER

DEVOWFS 201

N° 23

Partie théorique:(40PTs)

Dossier 1 : Créer une application Cloud native (8pts)

1. Comparez les 2 modèles du Cloud Computing :SaaS, et IaaS du point de vue des utilisateurs du service par rapport au fournisseur. Vous pouvez expliquer avec un schéma (2pts)
2. Une application Cloud native se base sur des piliers, parmi lesquels : DevOps et Conteneurs. Donner une définition de ces deux piliers (3pts)
3. Que veut dire « instancier une image Docker dans un conteneur »? (1.5pts)
4. Donner la commande pour lister tous les conteneurs disponibles (1.5pts)

Dossier2 : Adopter l'approche Agile(14pts)

Lors de la réunion sprint planning, Toute l'équipe Scrum (Product Owner, Scrum Master et les Développeurs) était présente pour déterminer les tâches à réaliser lors du prochain sprint :

TÂCHE	DURÉE	PRÉDÉCESSEURS
A	5	-
B	2	A
C	2	-
D	4	B,C
E	1	D
F	5	B,C
G	1	F
H	2	E,G

Avec :

A :Analyse et conception	E : test unitaire du composant 1
B : Création des tables	F : Développement du composant 2
C :Création des interfaces html	G : test unitaire du composant 2
D : Développement du composant 1	H : Test d'intégration

Questions :

1. Définir un sprint. (2,5 pts)
2. C'est quoi le rôle du scrum master ? (2,5 pts)
3. Réaliser le diagramme de Gantt (5 pts)
4. Identifier le chemin critique (4 pts)

Dossier 3: Gérer les données (partie : NoSQL) (12pts)

Nous utiliserons une base de données contenant des informations sur les terrains(agricole lotissement). Chaque terrain est caractérisé par:

- o numT : identifiant du terrain.
- o type : type du terrain(agricole ou lotissement)
- o prix: prix de vente terrain
- o titre_foncier : titre foncier du terrain
- o cin : le code du propriétaire du terrain
- o adresse : adresse du terrain
- o superficie : superficie du terrain
- o notaire : un document décrivant le notaire du terrain contenant les attributs suivants
 - Numn : identifiant du notaire.
 - nom : nom du notaire.
 - prenom : du notaire

Filière	DDOWFS	Variante	V2	Page	1 sur 5
Examen	Fin de Formation	Session	Report 2023		

- Adresse: adresse du notaire
- Tel : telephone du notaire

Écrivez les requêtes suivantes pour :

1. Afficher les notaires qui travaillent à « Rabat ». trier par nom croissant. (2,5pts)
2. Afficher le nombre total des terrains pour chaque type. (2,5pts)
3. Afficher le prix maximum de tous les terrains de type agricole. (2pts)
4. Augmenter de 1% le prix de tous les terrains agricoles des propriétaires cin=P1 et cin=P4. (2,5pts)
5. Supprimer l'adresse et le téléphone du notaire numn=N2(2,5pts)

Dossier4: préparer un projet web(6pts)

On souhaite faire la conception et l'analyse d'une application de gestion des dossiers des terrains (agricole ou lotissement) traités par les notaires avec la méthode UML. Les travaux de l'ancienne équipe chargée de l'analyse et de la conception de l'application ont abouti au modèle relationnel suivant :

- Cabinet(id_cab,nom,adresse,tel,ville)
- Notaire (numn, nom,prenom,age, tel,email, # id_cab)
- Dossier_terrain (iddossier, date_ouverture, date_cloture,#numter, #numnotaire, etat)
- vendeur (cin, nom, prenom, adresse ddn,tel)
- Terrain (numter,typeTerrain,prixvente,#cin, titre_foncier,adresse,ville,surface)

N.B :

- Le champ état prend l'une des valeurs : « traité », « nontraité »
- Le champ typeTerrain prend l'une des valeurs : « Terrain agricole », « terrain en lotissement »
- Les champs marqués en gras et soulignés représentent les clés primaires des table, les champs marqués par # représentent les clés étrangères.
- Toutes les clés sont de type chaines de caractères.

On précise que :

- Chaque notaire travaille dans un cabinet de notariat.
- Un vendeur peut être propriétaire de plusieurs terrains.
- Un dossier concerne un seul terrain et traité par un seul notaire.

En se basant sur le MLD déjà établi et les points précisés, réaliser le digramme de classe équivalent

1. Établir le diagramme de classes à partir du schéma ci-dessus. (6pts)

Filière	DDOWFS	Variante	V2	Page	2 sur 5
Examen	Fin de Formation	Session	Report 2023		

Partie Pratique:(60PTs)

Dans cette partie vous devez utiliser le modèle relationnel de la partie théorique.

Dossier 5 :Gérer les données (partie : MySql) (12pts)

1. Écrire une requête SQL qui affiche tous les cabinets de la ville Casablanca qui ont clôturés des dossiers entre le 2 janvier et le 31 Mai 2005 (3pts).
2. Écrire une fonction qui retourne le nombre de notaires qui n'ont traité aucun dossier concernant un type de terrain passé en paramètre (3pts).
3. Écrire une procédure qui affiche les notaires (nom, prénom) qui ont traité plus de 200 dossiers (3pts).
4. Écrire un déclencheur qui empêche de supprimer tous les dossiers sauf ceux qui datent de plus de 2 ans et non clôturés. (3pts).

Dossier 6: Back end(laravel) :(23 PTS).

On veut développer une application web de gestion des cabinets notariaux. Pour la partie backend, on utilisera les tables **Dossier_Terrain** et **Notaire** du modèle relationnel de la partie théorique. (Voir

Dossier 4 – Préparer un projet web)

1. Écrire le code de la méthode **up()** et la méthode **down()** de fichier migration de la table **Dossier_Terrain** .(2pts).
2. Écrire le code qui permet de peupler la table **Dossier_Terrain** par des jeux d'enregistrement en utilisant seeder. (2pts)
3. Écrire le code des deux modèles associés à la table **Dossier_terrain** et **Notaire**. Et définir les relations entre eux. (3pts).

N.B : On suppose que les autres modèles sont déjà créés.

4. Écrire le code du contrôleur **NotariatControleur** qui contient les méthodes :(8pts).
 - a) **Recherche** : qui permet de rediriger vers une page **vueRecherche.blade.php** qui affiche la liste des dossiers d'un cabinet passé comme paramètre. Et devant chaque dossier, la vue affiche une case à cocher « checkbox » pour supprimer le dossier coché. (2pts)
 - b) **SupprimerTous**: qui permet de supprimer tous les dossiers dont les numéros sont passés en paramètres. (2pts)
 - c) **Ajouter** : qui insère un nouveau notaire dans la table **Notaire** en respectant les contraintes suivantes :(2pts)
 - ✓ le champ numnn doit être une chaîne de caractères qui commence par N et se termine par E.

Filière	DDOWFS	Variante	V2	Page	3 sur 5
Examen	Fin de Formation	Session	Report 2023		

- ✓ le champ email doit être un email valide
- ✓ l'âge de notaire doit être supérieur à 22 ans

d) **Verifier** : permet de vérifier si le notaire a bien traité le dossier d'un terrain dont le numéro (dossier) est passé en paramètre. Si le dossier a été traité, elle retourne, sous format json, toutes les informations du terrain concerné, sinon elle retourne le message «dossier encours de traitement».(2pts)

5. Écrire les routes à ajouter dans **web.php** pour accéder aux 3 méthodes « **recherche** », «**SupprimerTous**» et « **ajouter** » du contrôleur «**NotariatControleur**» et la route à ajouter dans **api.php** pour accéder à la méthode « **Verifier**» (4pts).
6. Écrire le code de la vue **rechercher.blade.php** (4pts).

Dossier 7: développer en front end: (25 PTS)

Dans ce dossier, on veut développer avec **React** la partie front end d'une application de gestion des notariaux. L'équipe chargée de cette partie a décomposé la page souhaitée en composants React suivants.

- un composant simulateur '**Calcul_frais.js** (*Redux et React*)
 - un composant '**Verification.js**'(*React et Laravel :Appel apis*)
 - un composant **Menu.js**
1. Écrire le code du composant **Menu.js** contenant un menu de navigation sous forme de liens permettant d'accéder aux 2 composants précédents via les routes de React. (4 pts)

Le composant simulateur '**Calcul_frais.js**' permet aux utilisateurs de calculer les frais de notaire d'un terrain agricole ou d'un terrain en lotissement.

Ces frais incluent:

- Droits d'enregistrement: 4% du prix de vente du terrain
- Conservation foncière: 1,5% du prix de vente + 100 DH de droit fixe + 100 DH pour le certificat de propriété
- Frais de dossier (timbres, copies...) : 1500 DH
- Honoraires du notaire : entre 0,5% et 1,5% détaillés comme suit :

Montant de vente du terrain	Taux
Moins de 300 000,00 DH	4 000,00 DH (Frais fixes)
Entre 300 001,00 et 1 000 000,00 DH	1,5%
Entre 1 000 001,00 et 5 000 000,00 DH	1,25%
Entre 5 000 001,00 et 10 000 000,00 DH	0,75%
Plus de 10 000 000,00 DH	0,5%

- TVA sur les honoraires du notaire : 10%

Exemple :

Montant de vente du bien:	<input type="text" value="3000000"/>	<input type="button" value="calculer"/>
Droits d'enregistrement (4% de prix de vente) :		120 000,00
Conservation foncière (1,5% du prix de vente + 200 DH) :		45 200,00
Frais de dossier (timbres, copies...) :		1 500,00
honoraires du notaire (1,25% du prix de vente dans ce cas) :		37 500,00
TVA sur les honoraires du notaire (10%) :		3 750,00
TOTAL=		207 950,00

On veut sauvegarder les calculs de la simulation dans un store REDUX.

L'état initial de ce store va suivre la structure suivante :

```
constinitState={ListeSimulation :[  
  {  
    id : montant de vente,  
    Droits_en : droits d'enregistrement du terrain,  
    Conserve : le montant de la conservation foncière calculé,  
    Date : valeur de la date de calcul,  
    total : le total calculé  
  }  
]}
```

2. Créer le store Redux décrit ci-dessus avec son état initial et un reducer pour les deux actions suivantes : (6pts)
 - Ajouter : ajouter une simulation de calcul dans le tableau « ListeSimulation »(3pts)
 - Vider : vider le tableau « ListeSimulation » .(3pts)
3. Écrire le code du composant **Calcul_frais.js** :
 - a. le clic sur le bouton « calculer » calcule et affiche les droits d'enregistrement, les Honoraires du notaire, Frais de dossier, Conservation foncière, TVA sur les honoraires du notaire et le total calculé. (5pts)
 - b. Le calcul doit être sauvegardé dans le store (5pts)

4.Écrire le composant 'Verification.js' qui permet de vérifier si le notaire a bien traité un dossier dont son idossier est passé en paramètre et ceci en utilisant l'api crée dans la partie backend ont les informations suivantes (5PTS)

URL de l'api	http://127.0.0.1:8000/api/verifier/{iddemande}
Méthode HTTP	GET
Résultat	Résultat de la fonction Verifier du backend