

Commissariat régional de l'éducation de Ariana		
Devoir de synthèse N°3	<i>Section</i> : Sciences de l'Informatique	
	<i>Épreuve</i> : Systèmes & Technologies de l'Informatique	
<i>Niveau</i> : 4 ^{ème} année	<i>Date</i> : Mardi 13 mai 2025	<i>Durée</i> : 3 H
Nom & Prénom :		Classe : Note : /20

*Le sujet comporte 11 pages numérotées de 1/11 à 11/11.
Le candidat est appelé à répondre sur cette même feuille du devoir qui sera remise à la fin de l'épreuve.*

EXERCICE N°1 (1.75 points)

I- Dans un contexte de base de données et de développement Web et pour chacun des rôles proposés ci-dessous, mettre une croix (X) dans la case correspondante à la **proposition correcte**.

Important : Toute réponse comportant *plus d'une croix* est considérée *faussee*.

Rôle	Propositions		
	toLowerCase()	lowercase	strtolower()
Méthode en JavaScript qui convertit tous les caractères d'une chaîne en minuscule.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fonction en PHP qui retourne une sous-chaîne à partir d'une chaîne.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sélecteur d'état en CSS qui met en forme un élément HTML qui obtient le focus.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fonction en PHP qui retourne un nombre aléatoire.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Commande SQL qui désactive une contrainte.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Événement en HTML qui se déclenche lorsque le pointeur de la souris survole un élément.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Commande SQL qui retourne le jour du mois.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

EXERCICE N°2 (10 points)

Dans le cadre de la transition vers une mobilité durable, plusieurs villes mettent en place des services de location de vélos afin de faciliter les déplacements et réduire l'empreinte carbone.

Les vélos loués aux clients sont attachés à une seule station. A la fin de chaque location le client est appelé à rendre le vélo dans la station de départ.

Une application web est proposée pour permettre aux clients de réserver un vélo.

Après avoir autorisé la géolocalisation, l'application suggère la station la plus proche avec les vélos disponibles. Le client choisit un modèle de vélo et l'horaire, puis confirme sa réservation, ce qui marque le vélo comme « En location » dans le système.

Partie Gestion des données

Soit une **première version** d'une base de données simplifiée intitulée "**BD_BikeRent**" décrite par la représentation textuelle suivante :

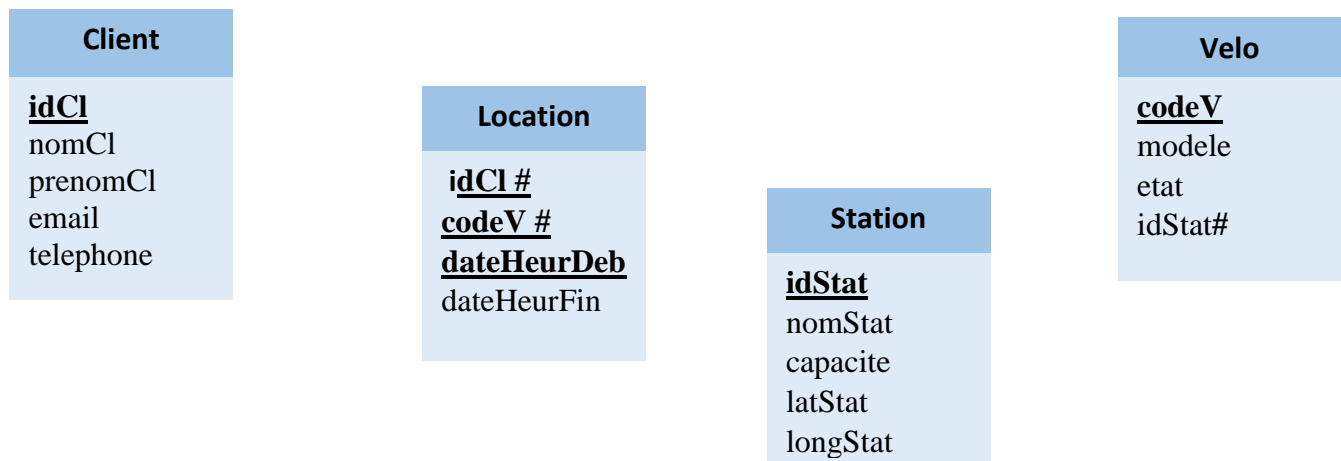
Version 1

Client (idCl, nomCl, prenomCl, email, telephone)
Station (idStat, nomStat, capacite, latStat, longStat)
Velo (codeV, modele, etat, idStat#)
Location (idCl#, codeV#, dateHeurDeb, dateHeurFin)

La description des champs des tables de cette base de données est la suivante :

Champ	Description
idCl	Identifiant d'un client
nomCl	Nom d'un client
prenomCl	Prénom d'un client
email	Email d'un client
telephone	Téléphone d'un client
codeV	Code d'un vélo
modele	Modèle d'un vélo (City, Electrique, Enfant, Pliant)
etat	Etat d'un vélo (Disponible, En location, En maintenance)
idStat	Identifiant d'une station
nomStat	Nom d'une station
capacite	Nombre maximal de vélos d'une station
latStat	Latitude de la localisation GPS d'une station
longStat	Longitude de la localisation GPS d'une station
dateHeurDeb	Date et heure de début de location
dateHeurFin	Date et heure de fin de location

- 1) Compléter le schéma graphique de la base de données "**BD_BikeRent**" en liant les tables et en indiquant les cardinalités de chaque relation (**types de relations**).



- 2) En se référant à la représentation textuelle de cette base de données, compléter la colonne "**Validité**" du tableau ci-dessous par "**Oui**" lorsque la règle de gestion est valide ou par "**Non**" dans le cas contraire.

Règle de gestion	Validité
Un client peut faire plusieurs locations au cours d'une journée.	
Un vélo peut être loué plusieurs fois pour la même date et la même heure.	
Chaque vélo est associé à une seule station.	
Une station regroupe plusieurs vélos.	

3) Pour faciliter la location des vélos, le client n'est pas obligé de rendre le vélo à la même station de location. Pour ce faire, le concepteur de la base de données propose de modifier la base de données en ajoutant les règles de gestion suivantes :

- **R1** : Une station peut être le point de départ de plusieurs locations et chaque location a un seul point de départ.
- **R2** : Une station peut également être le point de retour de plusieurs locations et chaque location n'a qu'un seul point de retour (ou **NULL** si elle n'est pas encore terminée).
- **R3** : Chaque vélo doit contenir l'identifiant de la dernière station où il est stationné.

Proposer une représentation textuelle des tables **Velo**, **Station** et **Location**, en tenant compte de cette modification et des règles de gestion **R1**, **R2** et **R3**.

Station (.....)

Velo (.....)

Location (.....)

Partie développement Web

L'application web propose le formulaire illustré par la "Figure 1", permettant de détecter et d'afficher la station la plus proche du client connecté ainsi que les numéros des vélos disponibles pour un modèle choisi. Le code du formulaire est fourni par la "Figure 2" en annexe (page 10).

Figure 1 : Formulaire de la page "Reservation.php"

A. Partie I

1) Pour chacune des propositions d'amélioration suivantes, cocher (X) la réponse adéquate.

a. Pour rendre la zone de saisie "**Station détectée**" non modifiable par l'utilisateur, on ajoute l'attribut :

- | | | | |
|--------------------------|----------|--------------------------|----------|
| <input type="checkbox"/> | hidden | <input type="checkbox"/> | readonly |
| <input type="checkbox"/> | disabled | <input type="checkbox"/> | blocked |

b. Pour que les boutons radio fassent partie du même groupe, on leur affecte la même valeur de l'attribut :

- | | | | |
|--------------------------|-------|--------------------------|------|
| <input type="checkbox"/> | value | <input type="checkbox"/> | id |
| <input type="checkbox"/> | name | <input type="checkbox"/> | type |

c. Pour rendre les options de la liste "**Vélo disponible**" modifiables lors de la saisie, on remplace la balise <select> par une zone de saisie <input> associée à une balise :

<input type="text"/>	textarea
<input type="list"/>	datalist

<input type="text"/>	fieldset
<input type="text"/>	details

d. Pour définir un format de date par défaut dans le champ de saisie et spécifier la date initiale, on utilise l'attribut :

<input type="text"/>	value
<input type="text"/>	name

<input type="text"/>	default
<input type="text"/>	fixed

e. Dans le cas où aucune valeur initiale n'est définie pour le contrôle de type range "**Durée estimée en heures**", sa valeur prend par défaut :

<input type="text"/>	La valeur minimale spécifiée
<input type="text"/>	La valeur maximale spécifiée

<input type="text"/>	Une valeur au hasard
<input type="text"/>	La valeur moyenne (min + max) / 2

2) Étant donné une feuille de style intitulée "**mesStyles.css**" contenant des règles CSS, remplir le tableau ci-dessous par le rôle de la règle CSS fournie ou par les éléments manquants de la règle CSS correspondante à la description donnée.

Description/Rôle	Règle CSS
Sélecteur qui met en forme les champs de type radio sans utiliser les id et les classes. { margin-right: 10px; }
	#logo{ width: 150px; height: 150px; margin-left: 150px; border-radius: 50%; animation : animlogo 2s infinite ; } @keyframes animlogo{ 0%{ transform: rotate(0deg);} 100%{ transform: rotate(360deg);} }
Transition de 0.3 s qui permet de changer la couleur d'arrière-plan lorsqu'on survole le bouton "Confirmer" par la souris.	#btn-confirm { background-color: #153354; color: white; border: none; padding: 12px 20px; font-size: 1rem; border-radius: 6px; cursor: pointer; width: 100%; ; } { background-color: #0056b3; }

- 3) Soit le code incomplet de la fonction JavaScript "**verifierModele()**" ci-dessous, permettant de vérifier si un modèle de vélo est choisi ou non, compléter le corps de la fonction en se référant aux commentaires fournis.

```

function verifierModele() {
    //récupérer les boutons radios dans la variable radios
    let radios = document. .... ("mv");
    for (let i = 0; i < .....; i++) {
        if (.....) {
            return ..... ; // Un modèle est sélectionné }
        }
    alert("Veuillez sélectionner un modèle de vélo.");
    return ..... ; // Aucun modèle sélectionné
}

```

- 4) Pour afficher la valeur de la durée estimée de la location du vélo, on utilise une fonction JavaScript "**dureEstime()**".

- a. Choisir l'événement le plus adapté pour appeler la fonction à chaque modification effectuée par le client :

<input type="checkbox"/>	onkeyup	<input type="checkbox"/>	onplay
<input type="checkbox"/>	onmouseover	<input type="checkbox"/>	oninput

- b. Compléter cette fonction en se référant aux commentaires fournis :

```

function dureEstime() {
    // récupérer la valeur du champ d'id "duree"
    ..... ;
    // copier la valeur dans l'étiquette d'id "valHeure"
    ..... = nbh;
}

```

B. Partie II

Lors du chargement de la page "**Reservation.php**" illustrée précédemment, on veut ajouter un script PHP pour remplir le formulaire à partir des données extraites de la base "**BD_BikeRent**", sachant qu'une fonction PHP **selectStation(lat, long)** a été développée et testée dans le fichier "**source.php**" permettant de rechercher et de retourner la station la plus proche.

- 1) Donner l'instruction PHP permettant d'associer le fichier "**source.php**" à la page "**Reservation.php**".

.....

2) Donner la commande **mysqli** permettant d'établir une connexion à la base de données hébergée sur un serveur nommé "**MonServeur**", sachant que la base est accessible à l'utilisateur "**user1**" ayant le mot de passe "**helloWord**".

`$cnx = ;`

3) En se référant à la "Figure 2" de l'annexe, écrire les instructions PHP permettant de récupérer les données correspondantes au modèle de vélo choisi ainsi qu'au code de la station.

`$modele = ;`
`$station = ;`

4) En utilisant la **version 1** de la base de données "**BD_BikeRent**" décrite précédemment à la page 2, compléter la requête SQL permettant de sélectionner les vélos disponibles correspondants au modèle souhaité et se trouvant dans la station indiquée.

`$sql = "SELECT
.....
..... " ;`
`$Res = mysqli_query($cnx,$sql);`

5) Pour parcourir les données retournées dans la variable **\$Res**, on propose deux solutions :

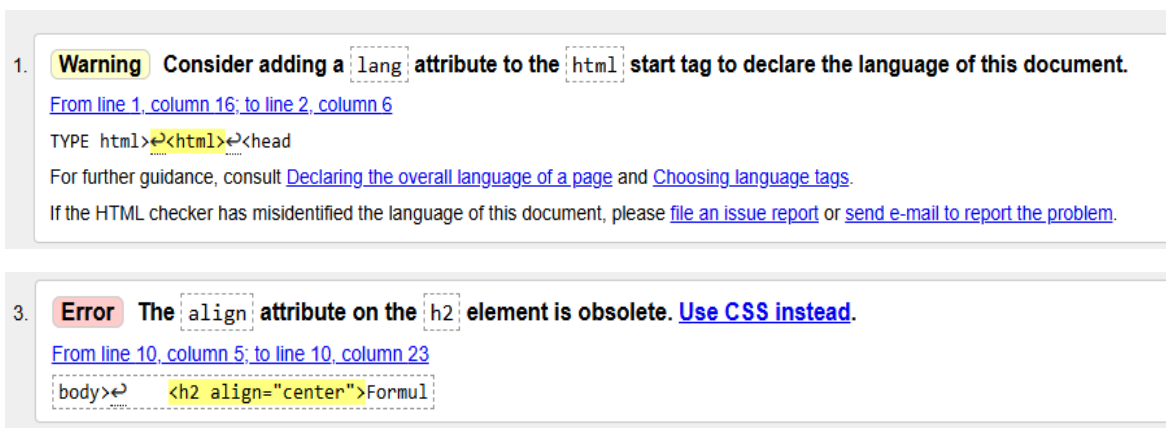
`$T = mysqli_fetch_row($Res)`
et
`$T = mysqli_fetch_array($Res).`

Expliquer la différence entre ces deux solutions.

.....
.....
.....

C. Partie III

Après avoir finalisé la création de toutes les pages web de l'application, l'administrateur les a soumises à un outil de validation des pages Web qui a généré les messages suivants :



1) Expliquer l'utilité de l'utilisation d'un outil de validation.

.....
.....
.....

- 2) Expliquer les deux messages affichés tout en précisant la différence entre **Warning** (avertissement) et **Error** (erreur).

.....

EXERCICE N°3 (8.25 points)

La Journée Internationale des Monuments et des Sites est célébrée le 18 avril de chaque année. Pour fêter cette occasion, l'Institut National du Patrimoine Tunisien encourage les musées à échanger des tableaux pendant une période fixe et les exposer dans leurs locaux. Pour ce faire, on utilise la base de données intitulée "**BD_Gestion_musee**" pour assurer cette action.

L'implémentation de la base de données "**BD_Gestion_musee**" est décrite par les requêtes SQL suivantes :

<pre>CREATE TABLE artiste (id_artiste int(11) NOT NULL AUTO_INCREMENT, nom_artiste varchar(30) NOT NULL, date_naiss_artiste date NOT NULL, date_deces_artiste date NULL, PRIMARY KEY (id_artiste)) ;</pre>	<pre>CREATE TABLE tableau (id_tab int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, titre_tab varchar(50) NOT NULL, annee_realisation int(4) NOT NULL, provenance varchar(15) NOT NULL, id_artiste int(11) NOT NULL REFERENCES artiste (id_artiste) ON DELETE CASCADE ON UPDATE CASCADE);</pre>
<pre>CREATE TABLE tiers (id_tiers varchar(5) NOT NULL PRIMARY KEY, nom_tiers varchar(30) NOT NULL, type_tiers varchar(30) NOT NULL) ;</pre>	
<pre>CREATE TABLE pret (id_pret int(11) NOT NULL AUTO_INCREMENT, id_tab int(11) NOT NULL, id_tiers varchar(5) NOT NULL, date_deb_pret date NOT NULL, date_fin_pret date NOT NULL, PRIMARY KEY (id_pret), CONSTRAINT pret_ibfk_1 FOREIGN KEY (id_tab) REFERENCES tableau (id_tab) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT pret_ibfk_2 FOREIGN KEY (id_tiers) REFERENCES tiers (id_tiers) ON DELETE CASCADE ON UPDATE CASCADE) ;</pre>	

La description des champs des tables de cette base de données est la suivante :

Champ	Description
id_artiste	Identifiant d'un artiste.
nom_artiste	Nom d'un artiste.
date_naiss_artiste	Date de naissance d'un artiste.
date_deces_artiste	Date de décès d'un artiste ou Null s'il n'est pas décédé.
id_tab	Identifiant d'un tableau.
titre_tab	Titre d'un tableau.
annee_realisation	Année de réalisation d'un tableau.
provenance	Région d'origine d'un tableau.
id_tiers	Identifiant d'un tiers (celui qui va prêter un tableau).
nom_tiers	Nom d'un tiers.
types_tiers	Type d'un tiers (Fondation, Musée, Galerie).
id_pret	Identifiant d'un prêt.
date_deb_pret	Date de début d'un prêt.
date_fin_pret	Date de fin d'un prêt.

N.B. : Des extraits du contenu des tables est fournie en annexe (page 11).

1) En se basant sur les requêtes **SQL** citées ci-dessus, déduire la représentation textuelle de la base de données "**BD_Gestion_musee**".

.....

.....

.....

.....

.....

.....

2) Pour chacune des requêtes **SQL** contenant des erreurs syntaxiques, données dans le tableau ci-dessous, réécrire la requête en effectuant les corrections nécessaires dans la colonne **Correction** et indiquer le type du langage utilisé (*LDD ou LMD*) dans la colonne **Type Langage**.

Requêtes erronées	Correction	Type Langage
DELETE pret WHERE date_deb_pret < `2023-03-15` ;		
UPDATE pret WHERE date_deb_pret = NOW();		
ALTER TABLE tableau CHECK (annee_realisation > 1980);		
ALTER TABLE artiste RENAME TO date_naissance_artiste;		
INSERT INTO tableau ("mémoire berbère", 1982, "Tozeur", 3) ;		

3) En se référant aux extraits des données (Annexe page 11), compléter les tableaux ci-dessous par le résultat d'exécution de chacune des deux requêtes suivantes :

```
SELECT id_artiste, COUNT(*) AS
nb_tableaux
FROM tableau
GROUP BY id_artiste
HAVING COUNT(*) >= 3;
```

```
SELECT t.id_tiers, t.nom_tiers, p.id_pret
FROM tiers t, pret p
WHERE t.id_tiers = p.id_tiers
and DATEDIFF(p.date_fin_pret, p.date_deb_pret) <= 30;
```


4) Écrire les requêtes **SQL** permettant d' :

- a) afficher les informations (**identifiant du prêt, titre du tableau, nom de tiers**) des tableaux prêtés à des fondations ou à des galeries.

.....

.....

.....

.....

- b) afficher la liste des prêts qui sont en cours.

.....

.....

.....

.....

- c) afficher, pour chaque année de réalisation, le nombre des tableaux créés.

.....

.....

.....

.....

- d) afficher, dans l'ordre chronologique, la liste des tableaux créés par des artistes de la famille "**Turki**".

.....

.....

.....

.....

- e) afficher les informations (**titre du tableau, provenance, nom d'artiste**) des tableaux qui n'ont pas été prêtés.

.....

.....

.....

.....

- f) afficher le nombre total de tableaux créés, l'année de réalisation la plus ancienne et l'année la plus récente de chaque artiste.

.....

.....

.....

.....

Annexe EXERCICE N°2

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Réservation de vélo</title>
  <link rel="stylesheet" href="mesStyles.css">
</head>
<body>
<h2 align="center">Formulaire de réservation</h2>
<form action="confirmation.php" method="post">
  <!-- Station détectée automatiquement avec le script Php-->
  <div class="form-group">
    <label for="stat">Station détectée :</label>
    <input type="text" name="stat" id="stat">
  </div>
  <!-- Type de vélo -->
  <fieldset>
    <legend>Choisissez un Modèle de vélo :</legend>
    <div>
      <label><input type="radio" name="mv" value="City">Vélo City</label><br>
      <label><input type="radio" name="mv" value="Electrique">Vélo Électrique</label><br>
      <label><input type="radio" name="mv" value="Enfant">Vélo Enfant</label><br>
      <label><input type="radio" name="mv" value="Pliant">Vélo Pliant</label>
    </div>
    <div>
      
    </div>
  </fieldset>
  <!-- Modèle de vélo disponible -->
  <div class="form-group">
    <label for="velo">Vélo disponible :</label>
    <select name="velo" id="velo">
      <option value="">-- Choisir un vélo --</option>
    </select>
  </div>
  <!-- Date et heure de début -->
  <div class="form-group">
    <label for="dhdeb">Date et heure de début :</label>
    <input type="datetime" name="dhdeb" id="dhdeb">
  </div>
  <!-- Durée estimée -->
  <div class="form-group">
    <label for="duree">Durée estimée (en heures) :</label>
    <input type="range" name="duree" id="duree" min="1" max="24" value=3>
    <span id="valHeure">3</span> h
  </div>
  <!-- Bouton de soumission -->
  <div class="form-group">
    <input type="submit" value="Confirmer" id="btn-confirm">
  </div>
</form>
<script src="mesControles.js"></script>
</body>
</html>
```

Figure 2 : Code de la page "Reservation.php"

Annexe EXERCICE N°3

Ci-dessous le contenu des tables de la base de données "BD_Gestion_musee"

Table artiste

id_artiste	nom_artiste	date_naiss_artiste	date_deces_artiste
1	Aly Ben salem	1910-03-01	2001-07-20
2	Hédi Turki	1922-05-15	2019-03-31
3	Abdelaziz Gorgi	1928-06-01	2008-11-10
4	Zoubeir Turki	1924-11-05	2009-10-23
5	Safia farhat	1924-02-29	2004-02-07

Table tableau

id_tab	titre_tab	annee_realisation	provenance	id_artiste
1	Danse des chevaux	1975	Tunis	1
2	Lumières nomades	1985	Djerba	1
3	Souffle de désert	1992	Sfax	2
4	Bleu méditerranéen	1990	Tunis	2
5	Femme au jasmin	1978	Sidi Bou Said	3
6	Mémoire berbère	1982	Tozeur	3
7	Scène de la médina	1980	Tunis	4
8	Marché aux épices	1983	Kairouan	5
9	Tapisseries de lumière	1970	Nabeul	5
10	L'élan de la palme	1972	Gabès	1

Table tiers

id_tiers	nom_tiers	type_tiers
F01	Fondation Abdelaziz Gorgi	Fondation
F02	Fondation kamal Lazaar	Fondation
G01	Galerie La Marsa	Galerie
G02	Le violon bleu	Galerie
M01	Musée National du Bardo	Musée
M02	Musée d'Art Moderne de Tunis	Musée

Table pret

id_pret	id_tab	id_tiers	date_deb_pret	date_fin_pret
1	1	M01	2023-05-01	2023-05-25
2	2	G01	2023-04-01	2023-07-01
3	3	M02	2023-03-15	2023-08-01
4	2	G02	2023-07-15	2023-07-30
5	5	F01	2024-04-03	2025-05-05
6	9	F02	2024-01-05	2024-07-05