

**Ne rien écrire ici**

- 5) Quels sont les capteurs dont la dernière date de mesure est effectuée avant le **15 février 2024** ?  
*On affichera les champs `idCapt`, `refCapt` et la dernière date de mesure effectuée.*

.....

.....

.....

.....

**B- Ecrire les requêtes SQL permettant de répondre aux situations suivantes :**

- 1) Pour alléger la base de données, on se propose de supprimer les alertes dont le statut est "R".

.....

.....

- 2) En raison d'un dysfonctionnement du capteur d'identifiant 4, on se propose de diminuer de 2 unités les mesures enregistrées entre le **29 janvier 2024** et le **02 février 2024**.

.....

.....

.....

- 3) Pour classer les capteurs selon leurs catégories, on se propose d'ajouter une nouvelle table nommée **Categorie** définie comme suit :

Champ	Description et type
<code>idCat</code>	Identifiant d'une catégorie, de type entier (clé primaire).
<code>nomCat</code>	Nom d'une catégorie, de type chaîne de 50 caractères au maximum.

.....

- 4) On se propose de restreindre les valeurs permises pour le champ `nomCat` de la table **Categorie**, déjà créée, aux valeurs suivantes : "**débit**", "**pression**" et "**température**".

.....

.....

- 5) On se propose d'établir la relation entre les tables **Capteur** et **Categorie** en ajoutant le champ `idCat` comme clé étrangère dans la table **Capteur**

.....

.....

.....

Ne rien écrire ici

## 2<sup>ème</sup> Partie : Programmation Web

### I. Analyse et interprétation du code HTML

En se basant sur la figure *Figure3 (page 13)*, répondre aux questions suivantes :

- 1) À la ligne N° 9, par quel évènement remplacer **onclick** pour que la fonction **demarrage()** soit appelée dès le chargement de la page ?

```
<body onclick = "demarrage()">
```

- 2) À la ligne N° 11, que représente le terme **img** de l'attribut **src** ?

```
<img id = "img" src = "img/tunisie.bmp" alt = "Carte Tunisie">
```

- 3) Quand se déclenche l'appel de la fonction **barrage(1)** figurant à la ligne N° 12 ?

```
<div class = "zone" id = "b1" onmouseover = "barrage(1)"></div>
```

- 4) À la ligne N° 16, quel est le rôle de l'expression **target = "\_ blank"** ?

```
<form action = "details.php" method = "post" target = "_ blank" >
```

- 5) À la ligne N° 17, quel est le rôle de l'attribut **hidden** ?

```
<input type = "text" name = "info1" id = "info1" hidden>
```

Voir suite ↗

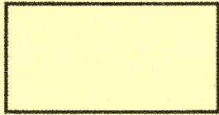


Section : ..... N° d'inscription : ..... Série : .....

Nom et Prénom : .....

Date et lieu de naissance : .....

Signatures des surveillants  
.....  
.....



Epreuve: STI - Section : Sciences de l'informatique - Session principale 2024

20

## II. Analyse et interprétation du code CSS

En se basant sur les figures *Figure3 (page 13)* et *Figure5 (page 14)* et pour chaque étiquette indiquée dans le fichier `mesStyles.css`, encercler la proposition correcte qui correspond à la description donnée.

Etiquette	Description	Proposition
1	Valeur permettant d'afficher les éléments de la classe <code>carte</code> conformément à leurs positions spécifiées dans le fichier <code>mesStyles.css</code> .	none inline-block
2	Déclaration permettant d'afficher les boîtes de la classe <code>zone</code> sous forme circulaire.	border-radius : 50% border-width : thin
3	Déclaration permettant d'afficher, en gras, le contenu de l'élément ayant l'identifiant " <code>affichage</code> ".	font-weight : bold font-family : bold

## III. Analyse et interprétation du code JavaScript

En se basant sur la figure *Figure4 (page 13)*, répondre aux questions suivantes :

- 1) Compléter le tableau ci-dessous en précisant pour chaque variable sa portée (**locale ou globale**).

Variable	debut	fin
Portée	.....	.....

- 2) Donner le rôle de chaque instruction figurant à la ligne spécifiée :

Ligne N° 8 `fin = lesBarrages.indexOf("#" , debut)`

.....  
.....

Ligne N° 12 `unBarrage = lesBarrages.substr(debut , fin - debut)`

.....  
.....

Ligne N° 18 `document.getElementById("info2").innerHTML = infoBarrage`

.....  
.....

Ne rien écrire ici

#### IV. Analyse et interprétation du code PHP

En se basant sur la figure *Figure6 (page 14)*, et pour chacune des questions ci-dessous, mettre une croix (X) dans la case correspondante à la réponse correcte.

**Important : Pour chaque question, toute réponse comportant plus d'une croix est considérée erronée.**

1) Quel est le comportement du script dans le cas où le fichier "**connexion.php**", appelé dans la ligne N° 2, est inexistant ?

- Une erreur est générée et le script s'arrête.
- Une erreur est générée et le script continue à s'exécuter.
- Le script s'exécute avec les paramètres de connexion par défaut.

2) Etant donné qu'au niveau de la ligne N° 7, la variable **\$nomBarrage** peut contenir un nom de barrage comportant le caractère '\_' (comme "Sidi\_Salem"), quelle instruction faut-il ajouter à la ligne N° 8 pour remplacer chaque caractère '\_' de la variable **\$nomBarrage** par un **espace** ?

- `$nomBarrage = str_replace('_', ' ', $nomBarrage);`
- `$nomBarrage = strcmp('_', ' ');`
- `$nomBarrage = substr($nomBarrage, strpos($nomBarrage, '_'), strlen($nomBarrage));`

3) Quel est le comportement du script en cas d'erreur syntaxique dans la requête SQL de la ligne N° 9 ?

- Le script affiche le message "Problème de connexion" et s'arrête.
- Le script affiche le message "Requête erronée" et s'arrête.
- Le script affiche le message "Barrage non existant" et s'arrête.

4) Quel est le comportement du script si on modifie l'instruction de la ligne N° 13, comme indiqué ci-dessous ?

```
{ $message = $lig[0]. " * ".$lig[1]. " * ".$lig[2]; }
```

- Le script génère une erreur de syntaxe et s'arrête.
- Le script génère une erreur de syntaxe puis affiche le message "Barrage non existant".
- Le script ne modifie pas son comportement car il est possible d'accéder aux éléments du tableau résultat soit par leurs noms ou leurs indices.

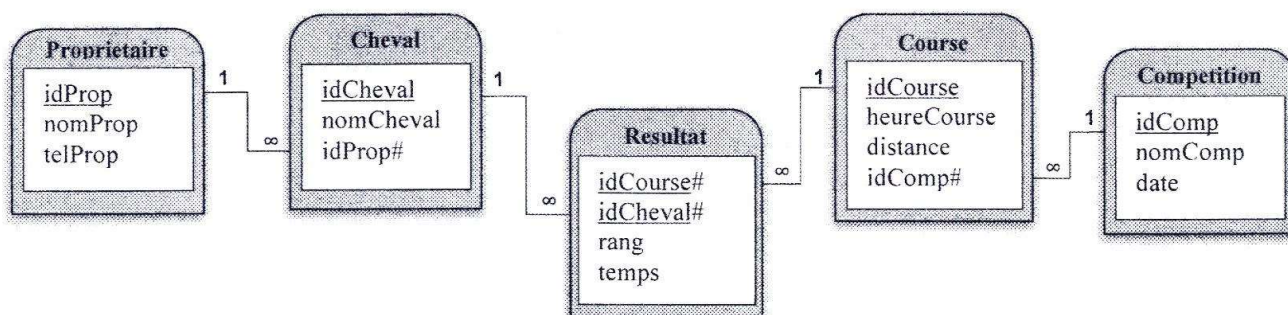
RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session principale	2024
	Épreuve : <b>Systèmes et Technologies de l'Informatique</b>		Section : <b>Sciences de l'informatique</b>
	Durée : <b>3 Heures</b>		Coefficient de l'épreuve : <b>1.8</b>

N° d'inscription

## Enoncé de l'exercice N°2

HippicTeams est une association qui regroupe des amateurs de chevaux. Elle organise des compétitions hippiques mensuelles comportant plusieurs courses qui se déroulent au cours d'une même journée.

Pour diffuser les détails des compétitions, l'association se propose de mettre en place une base de données simplifiée dont la structure est illustrée par la représentation graphique suivante :



Les champs des tables sont décrits dans le tableau suivant :

Champ	Description
idProp	Identifiant du propriétaire d'un cheval.
nomProp	Nom du propriétaire d'un cheval.
telProp	Numéro de téléphone du propriétaire d'un cheval.
idCheval	Identifiant d'un cheval.
nomCheval	Nom d'un cheval.
idComp	Identifiant d'une compétition.
nomComp	Nom d'une compétition.
date	Date de déroulement d'une compétition.
idCourse	Identifiant d'une course.
heureCourse	Heure de début d'une course.
distance	Distance à parcourir dans une course, exprimée en mètre.
rang	Classement d'un cheval dans une course ( <b>entier strictement positif</b> ).
temps	Temps réalisé par un cheval dans une course, exprimé en seconde.

## Enoncé de l'exercice N°3

Dans le cadre de la gestion des ressources hydrauliques, on dispose d'un site web qui diffuse les données des barrages des différentes régions de la Tunisie. Ces informations sont disponibles sur une cartographie interactive (voir *Figure1* et *Figure2* de la page 13) qui permet d'afficher pour chaque barrage choisi son nom, sa superficie et éventuellement d'autres informations supplémentaires enregistrées dans la base de données «Gestion\_Barrage».

Généralement, un barrage est équipé de plusieurs vannes qui sont contrôlées par des capteurs permettant de collecter des informations telles que le débit, la pression et la température.

### 1<sup>ère</sup> Partie : Gestion de la base de données

La base de données « Gestion\_Barrage » est décrite par la représentation textuelle simplifiée ci-dessous :

**Barrage** (idBar, nomBar, volume, hauteur, superficie)

**NiveauEau** (dateNiv, idBar#, niveau)

**Vanne** (idVan, etatVan, idBar#)

**Capteur** (idCapt, refCapt, idVan#)

**Mesure** (idMes, valMes, dateMes, idCapt#)

**Alerte** (idAlert, degAlert, statAlert, idMes#)

Les champs des tables sont décrits dans le tableau suivant :

Champ	Description et type
idBar	Identifiant d'un barrage, de type entier.
nomBar	Nom d'un barrage, de type chaîne de caractères.
volume	Capacité maximale d'un barrage exprimée en million m <sup>3</sup> , de type réel.
hauteur	Hauteur d'un barrage exprimée en mètre, de type réel.
superficie	Superficie d'un barrage exprimée en km <sup>2</sup> , de type réel.
dateNiv	Date et heure de prélèvement du niveau d'eau d'un barrage.
niveau	Niveau d'eau exprimé en mètre, de type réel.
idVan	Identifiant d'une vanne, de type entier.
etatVan	Etat d'une vanne, de type entier (0 = fermé, 1 = ouvert).
idCapt	Identifiant d'un capteur, de type entier.
refCapt	Référence d'un capteur, de type chaîne de caractères.
idMes	Identifiant d'une mesure enregistrée par un capteur, de type entier.
valMes	Valeur de la mesure enregistrée par un capteur, de type réel.
dateMes	Date et heure d'une mesure enregistrée par un capteur.
idAlert	Identifiant d'une alerte déclenchée suite à une mesure, de type entier.
degAlert	Degré d'une alerte, de type entier (1 : avertissement, 2 : grave, 3 : critique).
statAlert	Statut d'une alerte, de type caractère ("R" : résolu, "A" : actif, "E" : en attente).

## 2<sup>ème</sup> Partie : Programmation Web

Le site est composé des pages `index.html`, `mesStyles.css`, `controles.js` et `details.php` dont on présente les extraits ci-dessous.

### ✓ Page "index.html"

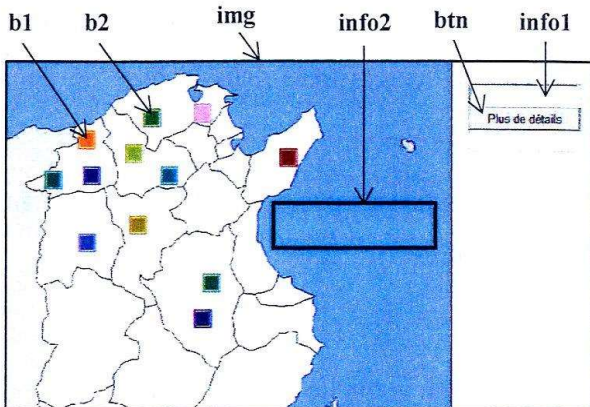


Figure1 : mode création

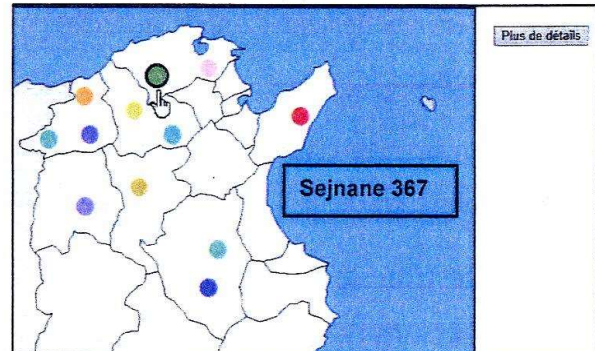


Figure2 : mode aperçu

```
3 <head>
4   <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
5   <title>Carte des barrages de la Tunisie</title>
6   <script src = "controles.js" ></script>
7   <link href = "mesStyles.css" rel = "stylesheet" type = "text/css">
8 </head>
9 <body onclick = "demarrage()">
10  <div class="carte">
11    <img id = "img" src = "img/tunisie.bmp" alt = "Carte Tunisie" >
12    <div class = "zone" id = "b1" onmouseover = "barrage(1)" ></div>
13    <div class = "zone" id = "b2" onmouseover = "barrage(2)" ></div>
14    <!-- suite des barrages -->
15    <div class = "formulaire" id = "details" >
16      <form action = "details.php" method = "post" target = "_blank">
17        <input type = "text" name = "info1" id = "info1" hidden >
18        <input name = "Submit" type = "submit" id = "btn" value = "Plus de détails">
19      </form>
20    </div>
21    <div class = "formulaire" id = "affichage"><p id = "info2"></p></div>
22  </div>
23 </body>
```

Figure3 : extrait du code HTML

### ✓ Page "controles.js"

```
1 lesBarrages = "Benmtir 103#Sejnane 367#Sidi_Salem 7950#Joumine 418#"
2
3 function extraireBarrage(y){
4   let fin = -1 // Dans cet extrait "let" est équivalente à "var"
5   i = 0
6   let unBarrage = "Non disponible"
7   do { debut = fin + 1
8     fin = lesBarrages.indexOf("#" , debut)
9     i = i+1
10    } while (i != y && fin != -1)
11   if(fin != -1)
12     unBarrage = lesBarrages.substr(debut , fin - debut )
13   return unBarrage }
14
15 function barrage(x){
16   infoBarrage = extraireBarrage(x)
17   document.getElementById("info1").value = infoBarrage
18   document.getElementById("info2").innerHTML = infoBarrage }
```

Figure4 : extrait du code JavaScript

✓ Page "mesStyles.css"

```
img { width : 700px;}

.carte { position : relative;
        display : .....; } ①

.zone { position : absolute;
        .....; } ②

.formulaire { position : absolute;}

#affichage { top: 165px;
             left: 490px;
             width: 185px;
             .....; } ③

#details { top: 20px;
           left: 720px; }

#b1 { top: 85px;
      left: 265px;
      width: 20px;
      height: 20px;
      background-color: rgba(255, 180, 0, 0.5);}

#b2 { top: 60px;
      left: 340px;
      width: 20px;
      height: 20px;
      background-color: rgba(100, 180, 55, 0.5); }
```

Figure5 : extrait du code CSS

✓ Page "details.php"

```
1 <?php
2   require ("connexion.php");
3   $connect = mysqli_connect($host,$user,$pass,$base) or die("Problème de connexion");
4
5   $barrage = $_POST["info1"];
6   $posNom = strpos($barrage, " ");
7   $nomBarrage= substr($barrage , 0 , trim($posNom));
8
9   $ch="select * from barrage where nomBar = '$nomBarrage'";
10  $req = mysqli_query($connect,$ch) or die("Requête erronée");
11  $message="Barrage non existant";
12  if($lig = mysqli_fetch_array($req))
13      {$message = $lig["idBar"]." * ".$lig["nomBar"]." * ".$lig["volume"];}
14  echo($message);
```

Figure6 : extrait du code PHP

N.B. :

- Le fichier **connexion.php** contient les paramètres d'accès à la base de données.
- L'instruction de la **ligne 3** peut être exprimée comme suit :

```
mysqli_connect($host,$user,$pass) or die("Problème de connexion");
mysqli_select_db($base);
```

- L'instruction de la **ligne 10** peut être exprimée comme suit :

```
$req = mysqli_query($ch) or die("Requête erronée");
```

- L'instruction de la **ligne 12** peut être exprimée comme suit :

```
if($lig = mysqli_fetch_array($req))
```