

Ne rien écrire ici

3) Compléter les phrases ci-dessous par les mots convenables à partir de la liste suivante :

composée – de domaine – fille – de table – primaire – référentielle – mère – simple – étrangère

- a) La clé étrangère dans la table "Stand" assure l'intégrité avec la table "Halle".
- b) La table "Location" possède une clé primaire
- c) La contrainte d'intégrité permet de s'assurer que le champ "prix" de la table "Location" ne contient que des valeurs strictement positives.
- d) Le champ "codeCl" dans la table "Location" est une clé, toutefois il est une clé dans la table "Client".

Exercice 2 : Interprétation des requêtes

(1,5 point)

1. Soit la liste des rôles suivante :

- a) Supprimer toutes les manifestations dont le "codeManif" contient le caractère "3".
- b) Modifier la valeur par défaut d'un champ de la table "Halle".
- c) Afficher le nom et la date d'ouverture des manifestations ayant le code de type "Salon".
- d) Modifier le numéro de téléphone d'un client dont le code est spécifié.
- e) Supprimer toutes les manifestations dont le "codeManif" se termine par le caractère "3".
- f) Ajouter un nouveau client avec des valeurs spécifiques.
- g) Ajouter une contrainte de domaine sur un champ de la table "Halle".
- h) Supprimer un client ayant des valeurs spécifiques.
- i) Afficher le nom et la date d'ouverture des manifestations ayant le libellé de type "Salon".
- j) Modifier le type du champ "telCl" de la table "Client".

Pour chacune des requêtes suivantes, compléter la colonne "Réponse" avec la lettre correspondante à son rôle parmi la liste présentée ci-dessus.

Requête	Réponse
INSERT INTO Client VALUES ('C0001', 'Ali ABIDI', '33556677', 'Ali.ABIDI@gmail.com', 'Tunisie');	
SELECT nomManif, dateOuv FROM Manifestation F, Type_Manif T WHERE (F.codeType = T.codeType) AND (libType = 'Salon');	
UPDATE Client SET telCl = '66333111' WHERE codeCl = 'C0023';	
DELETE FROM Manifestation WHERE codeManif Like '%3';	
ALTER TABLE Halle ADD CHECK (nbStandsMax > 0);	
ALTER TABLE Halle MODIFY nbStandsMax int DEFAULT 1;	

Exercice 3 : Manipulation des données de la base

(6 points)

Écrire en SQL les requêtes permettant de répondre aux questions suivantes :

1) Quels sont les différents types de manifestations ? *On affichera les champs "codeType" et "libType".*

2) Quelles sont les manifestations qui ont une durée strictement inférieure à 5 jours ? *On affichera les champs "codeManif", "nomManif" et "dateOuv".*

3) Comment réduire de **15%** les prix de location des stands de la manifestation ayant le code "**M0001**" ?

4) Quels sont les clients qui ont loué un ou plusieurs stands ? *On affichera sans répétition les champs "nomCl", "telCl" et "emailCl".*

5) Quels sont les stands libres dans la halle de libellé "**Halle A**", triés par superficie décroissante ? *On affichera les champs "codeStand", "designation" et "superficie".*

Ne rien écrire ici

- 6) Quel est le nombre de manifestations pour chaque type ? On affichera les champs "codeType" et "libType" ainsi que le nombre de manifestations de chaque type.

- 7) Quelles sont les manifestations dont le revenu total de location de stands dépasse strictement 10000 Dinars ? On affichera les champs "codeManif", "nomManif" ainsi que le total des revenus.

- 8) Quelles sont les désignations des stands qui n'ont pas été loués pendant l'année **2025** ? On affichera uniquement le champ "designation".

- 9) Quels sont les stands ayant le prix de location le plus élevé ? On affichera les champs "codeStand", "designation" et "superficie".

Section : N° d'inscription : Série :

Nom et Prénom :

Date et lieu de naissance :

Signature des
surveillants

Epreuve : STI - Section : Sciences de l'informatique – Session principale 2025

Exercice 4 : Extension et modification de la structure de la base de données (1,5 point)

Important : À chaque représentation textuelle demandée, indiquer la **clé primaire** et la (les) **clé(s) étrangère(s)** de chaque table.

1) Pour éliminer l'incohérence des données dans le champ "**pays**" de la table "**Client**", on propose de modifier la structure de la base de données et ceci en respectant les deux règles de gestion suivantes :

- **R1** : Un client appartient à un seul pays.
- **R2** : Il est possible d'avoir plusieurs clients provenant du même pays.

Proposer une représentation textuelle de la table "**Client**" ainsi que toute nouvelle table indispensable permettant de prendre en compte cette modification.

2) Pour chaque manifestation, on désire évaluer d'autres services tels que la sécurité, la publicité, etc. Pour cela, on propose d'apporter les modifications suivantes :

- Création de deux nouvelles tables :
 - **Service** : contient les informations relatives aux différents services.
 - **Note_Service** : contient les notes attribuées aux différents services lors d'une évaluation.
- Ajout de nouvelles règles de gestion :
 - **R3** : Pour une évaluation, on peut évaluer plusieurs services.
 - **R4** : Un service peut être évalué plusieurs fois dans différentes évaluations.
 - **R5** : Une évaluation concerne une seule manifestation.
 - **R6** : Un service ne peut être évalué qu'une seule fois par évaluation avec l'attribution d'une note.

a) Le tableau ci-dessous contient quatre représentations textuelles erronées. En se référant aux règles de gestion ci-dessus et pour chaque représentation textuelle, compléter la colonne "**Règle**" par le nom de la règle (**R3**, **R4**, **R5** ou **R6**) non respectée.

Représentation textuelle	Règle
Service (codeSer, libSer) Evaluation (codeEval, dateEval) Note_Service (codeEval#, codeSer#, note, commentaire, codeManif#)	
Service (codeSer, libSer) Evaluation (codeEval, dateEval, codeManif#, codeSer#, note) Note_Service (codeEval#, commentaire)	
Service (codeSer, libSer) Evaluation (codeEval, dateEval, codeManif#) Note_Service (codeEval#, codeSer#, commentaire)	
Service (codeSer, libSer, codeEval#, note) Evaluation (codeEval, dateEval, codeManif#) Note_Service (codeEval#, commentaire)	

b) En déduire une nouvelle représentation textuelle correcte des tables "**Service**", "**Evaluation**" et "**Note_Service**" tout en respectant les règles de gestion **R3**, **R4**, **R5** et **R6**.

Partie B : Programmation Web

(8 points)

Exercice 1 : Analyse et interprétation du code HTML

(2,5 points)

La page "EvaluationManif.html", dont l'aperçu est présenté dans la figure "Fig.1" (page 12) et dont le code source est présenté dans la figure "Fig.2" (page 13), permet d'évaluer la qualité des quatre services de base offerts par la société "Dynamik Expo" (accueil, confort, signalétique et logistique).

En se basant sur ces deux figures, répondre aux questions suivantes :

- 1) Indiquer devant chaque rôle présent dans le tableau ci-dessous le numéro de la ligne contenant le code HTML qui lui correspond.

Rôle	N° ligne
Créer un bouton permettant d'envoyer les données du formulaire à un serveur.	
Créer un conteneur en ligne permettant d'afficher un contenu.	
Spécifier l'encodage des caractères de la page web.	
Lier une feuille de style CSS externe à la page web.	
Exécuter une fonction JavaScript au moment où la page web est entièrement chargée.	
Créer une liste déroulante.	
Regrouper un ou plusieurs éléments d'un formulaire.	
Fournir des informations supplémentaires sur la page web qui seront utilisées par les moteurs de recherche.	

- 2) Pour chacune des affirmations ci-dessous, mettre une croix (X) dans la case correspondante à la proposition correcte.

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

- a) L'attribut **method** utilisé au niveau de la ligne 11 du code HTML de la figure "Fig.2" permet de :

- ☐ spécifier l'adresse URL vers laquelle les données du formulaire seront envoyées
☐ spécifier la méthode HTTP utilisée pour envoyer les données du formulaire
☐ définir le nom du conteneur où la réponse du serveur sera affichée

- b) La balise **<hr>** utilisée au niveau de la ligne 21 du code HTML de la figure "Fig.2" permet d'insérer :

- ☐ une ligne horizontale
☐ une ligne dans un tableau
☐ un retour à la ligne

- c) L'élément **<input type="range">** utilisé au niveau de la ligne 24 du code HTML de la figure "Fig.2" permet de :

- ☐ saisir librement un texte dont la valeur est comprise entre min et max
☐ sélectionner une valeur dans une plage définie
☐ saisir une valeur numérique

- d) L'attribut **alt** utilisé au niveau de la ligne 26 du code HTML de la figure "Fig.2" sert à :

- ☐ spécifier une légende de l'image
☐ définir un texte qui s'affiche lorsqu'on survole l'image
☐ définir une description textuelle alternative pour l'image

Ne rien écrire ici

Exercice 2 : Analyse et interprétation du code CSS

(1,5 point)

En se basant sur la figure "Fig.1" (page 12) et la figure "Fig.2" (page 13), écrire dans la colonne "Code CSS" du tableau ci-dessous le code CSS adéquat à chaque mise en forme demandée et ce en utilisant les instructions fournies dans la liste suivante :

Liste d'instructions

background-color : yellow | border : 3px solid red | font-style : italic | color : red | #labserv | legend | font-weight : bolder | border-radius : 5px | display : inline-block | border : 3px dashed red | .labserv | font-size : 20px | text-align : center | padding : 5px | margin : 5px | color : yellow | width : 150px |

Mise en forme	Code CSS
Le titre "Évaluation d'une manifestation" doit avoir : - une couleur de texte rouge. - une taille de caractères égale à 20 pixels. - un style de caractères gras.	<pre> { ; ; ; } </pre>
Le titre "Service" doit avoir : - un texte centré. - un arrière-plan jaune. - une bordure rouge hachée de 3 pixels. - une largeur de 150 pixels. - un style de caractères italique.	<pre> { ; ; ; ; ; } </pre>
Les libellés existant dans le formulaire doivent avoir : - un comportement d'affichage bloc en ligne. - une marge intérieure haut de 5 pixels.	<pre> label { ; ; } </pre>

Exercice 3 : Analyse et interprétation du code JavaScript

(2 points)

1) En se basant sur la figure "Fig.2" (page 13) et pour chacune des affirmations ci-dessous, répondre par la lettre (V) si la proposition est correcte ou par la lettre (F) dans le cas contraire.

Important : Une case vide ou contenant des lettres autres que (V) ou (F) ou des symboles est considérée comme une réponse *erronée*.

a) Le code incomplet de la fonction "verif", dont l'appel figure à la ligne 11 du code HTML, est le suivant :

```

function verif ()
{
    if (document.getElementById ("manif") ..... ==0)
    { alert ("Il faut sélectionner une manifestation.");
      return false; }
}

```

La propriété à ajouter à la fonction "verif" afin de vérifier qu'une manifestation est sélectionnée est :

☐ .checked
 ☐ .selectedIndex
 ☐ .value
 ☐ .indexOf

- b) Le code incomplet de la fonction **"imagenote"**, dont les appels figurent aux lignes **25, 29, 33** et **37** du code HTML, est le suivant :

```
function imagenote (idnote, idimage)
{
    note=document.getElementById (idnote).value;
    ;
}
```

L'instruction à ajouter à la fonction **"imagenote"** afin d'afficher l'image correspondante à la note attribuée (**"0.png"** pour la note 0, **"1.png"** pour la note 1, ..., **"5.png"** pour la note 5) est :

- ☐ document.getElementById (idimage).name=note+'.png';
- ☐ document.getElementsByName (idimage).value=note+'.png';
- ☐ document.getElementById (idimage).src=note+'.png';
- ☐ document.getElementsByName (idimage).src=note+'.png';

- c) La fonction **"calcultotal"**, dont les appels figurent aux lignes **25, 29, 33** et **37** du code HTML, permet de calculer et d'afficher le total des quatre notes attribuées. Le script de cette fonction est :

☐

```
function calcultotal ()
{
    let s=0;
    for (let i=1; i<=4; i++)
    {
        idnote="serv"+i;
        s=s+parseInt(document.getElementById(idnote).value);
    }
    document.getElementById ("total").innerHTML=String(s) +"/20";
}
```

☐

```
function calcultotal ()
{
    let s=0;
    for (const i=1; i<=4; i++)
    {
        idnote="serv"+i;
        s=s+parseInt (document.getElementById (idnote).value);
    }
    document.getElementById ("total").innerHTML=String(s)+"/20";
}
```

☐

```
function calcultotal ()
{
    for (i=1; i<=4; i++)
    {
        idnote="serv"+i;
        s=s+parseInt (document.getElementById (idnote).value);
    }
    document.getElementById ("total").innerHTML=String(s)+"/20";
}
```

☐

```
function calcultotal ()
{
    let s=0;
    for (i=1; i<=4; i++)
    {
        const idnote="serv"+i;
        s=s+Number(document.getElementById (idnote).value);
    }
    document.getElementById ("total").innerHTML=String(s)+"/20";
}
```


**Signature des
surveillants**

Nom et Prénom :

Date et lieu de naissance :

Epreuve : STI - Section : Sciences de l'informatique – Session principale 2025

2) Le code de la fonction "**inconnue**", dont l'appel figure à la ligne 10 du code HTML, est le suivant :

```
function inconneue ()
{
    for (i=1; i<=4; i++)
    {
        idimage="imgser"+i;
        document.getElementById (idimage).src='0.png';
        idnote="serv"+i;
        document.getElementById (idnote).value=0;
    }
}
```

En se basant sur ce code et sur le code HTML de la figure "Fig.2" (page 13), donner le rôle de cette fonction.

Exercice 4 : Analyse et interprétation du code PHP

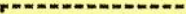
(2 points)

En se référant aux codes des fichiers "**Evaluation.php**" (page 14), "*EvaluationManif.html*" (page 13) et suite au clic sur le bouton "**Valider**" de l'exemple d'exécution suivant, et pour chacune des affirmations ci-dessous, mettre une croix (X) dans la case correspondante à la réponse correcte.

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

Évaluation d'une manifestation

Manifestation : Séminaire des médecins ▾

Service	Note
Accueil :	 ★★★★★
Confort :	 ★★★★★
Signalétique :	 ★★★★★
Logistique :	 ★★★★★
Note totale :	10/20

Valider Annuler

1) Suite à l'exécution de l'instruction qui figure à la ligne 2 du code PHP, la variable "*\$manif*" contient :

- ☐ Séminaire des médecins
☐ M0003
☐ manif

2) Suite à l'exécution de l'instruction qui figure à la ligne 5 du code PHP, le navigateur affiche :

- ☐ Le total des notes attribuées est 10 / 20
- ☐ Le total des notes attribuées est 0.5
- ☐ Un message d'erreur relatif au champ "total" et le message : "Le total des notes attribuées est / 20"

3) L'instruction qui figure à la ligne 3 du code PHP permet de :

- ☐ copier tous les éléments du tableau associatif "\$_POST" dans le tableau nommé "Snote"
- ☐ créer un tableau indexé nommé "Snote" contenant les notes récupérées à partir du formulaire
- ☐ créer un tableau associatif nommé "Snote" contenant les notes récupérées à partir du formulaire

4) Si la date du jour est **10 Juin 2025**, la variable "\$datejour" qui figure à la ligne 6 du code PHP contient :

- ☐ 2025-06-10
- ☐ 06-10-2025
- ☐ 10-06-2025

5) L'instruction qui figure à la ligne 7 du code PHP peut être remplacée par :

- ☐ \$con=mysqli_connect("localhost","root","","Gestion_Exposition") ;
if (!\$con) { echo "Erreur B.D."; }
- ☐ \$con=mysqli_connect("localhost","root","","Gestion_Exposition") ;
if (!isset(\$con)) { die ("Erreur B.D."); }
- ☐ \$con=mysqli_connect("localhost","root","","Gestion_Exposition") ;
if (!\$con) { die ("Erreur B.D."); }

6) L'instruction qui figure à la ligne 10 du code PHP peut être remplacée par l'instruction :

- ☐ if (\$nreg=mysqli_num_rows(\$resInsert))
- ☐ if (\$nreg=mysqli_fetch_row(\$resSelect))
- ☐ if (\$nreg=mysqli_fetch_rows(\$resSelect))

7) L'instruction qui figure à la ligne 24 du code PHP peut être remplacée par l'instruction :

- ☐ if (mysqli_num_rows(\$resInsert)!=0)
- ☐ if (mysqli_fetch_row(\$resInsert))
- ☐ if (mysqli_affected_rows (\$con)!=0)

8) La fonction "**mysqli_error(\$con)**" qui figure à la ligne 27 du code PHP retourne en cas d'erreur générée par la dernière requête SQL exécutée :

- ☐ une chaîne contenant le message d'erreur
- ☐ un tableau contenant la liste des erreurs
- ☐ un entier contenant le code de l'erreur

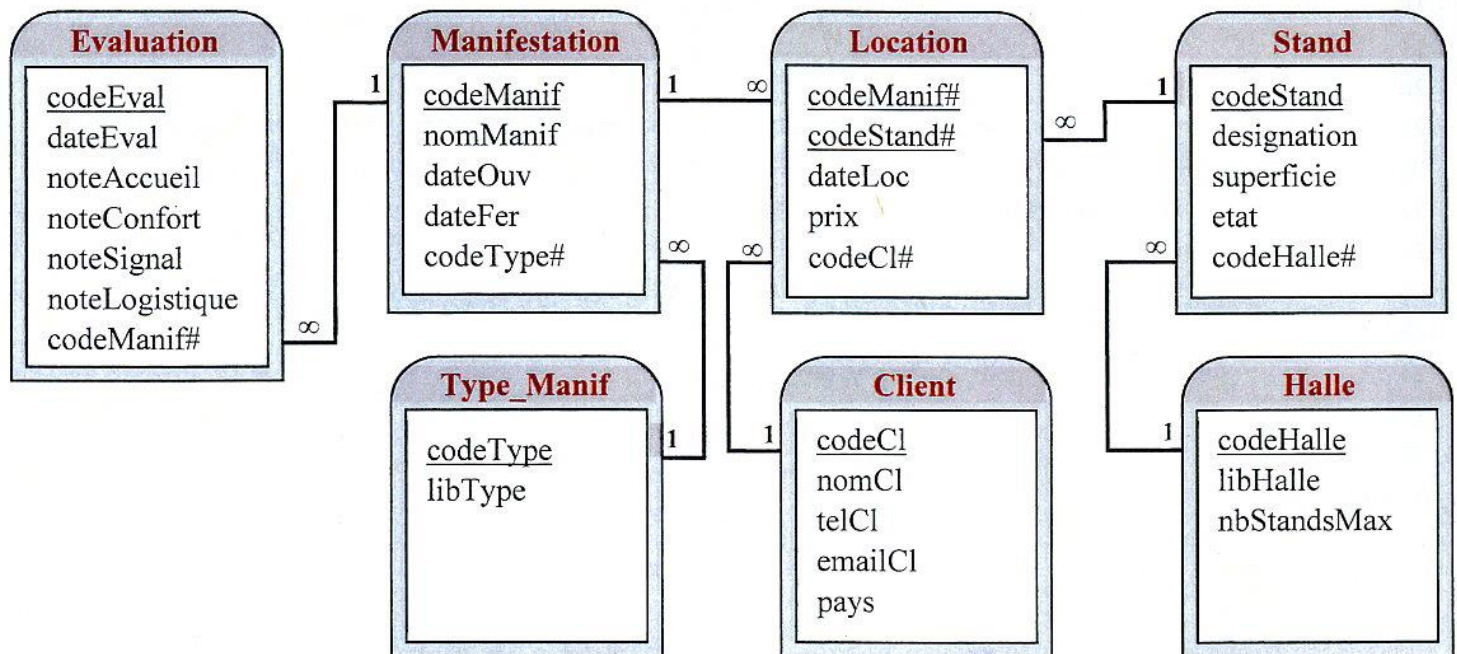
RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session principale 2025
	Épreuve : Systèmes et Technologies de l'Informatique	Section : Sciences de l'informatique
	Durée : 3h	Coefficient de l'épreuve: 1.8

N° d'inscription

Afin de faciliter la gestion de ses manifestations, la société "**Dynamik Expo**" utilise une base de données simplifiée intitulée "**Gestion_Exposition**". Chaque manifestation est associée à un type (foire, salon, séminaire, etc.). Les manifestations se déroulent dans des halles qui sont subdivisées en plusieurs stands disponibles à la location aux clients.

Pour chaque manifestation, des évaluations sont réalisées de manière anonyme en attribuant des notes aux services de base offerts (accueil, confort, signalétique et logistique).

La base de données "**Gestion_Exposition**" est décrite par la présentation graphique suivante :



Sachant que :

- **Type_Manif** : contient les informations relatives aux types de manifestations.
- **Manifestation** : contient les informations relatives aux manifestations.
- **Halle** : représente les différentes halles disponibles.
- **Stand** : contient les informations des différents stands.
- **Client** : contient les informations des clients qui louent des stands.
- **Location** : représente les locations de stands par les clients pour des manifestations spécifiques.
- **Evaluation** : contient les informations relatives aux évaluations des quatre services de base offerts pour chaque manifestation.

Soit la description des champs des tables de cette base de données :

Champ	Description	Type et propriété(s)
codeType	Code d'un type d'une manifestation	Chaîne de 3 caractères
libType	Libellé d'un type d'une manifestation	Chaîne de 20 caractères et unique
codeManif	Code d'une manifestation	Chaîne de 5 caractères
nomManif	Nom d'une manifestation	Chaîne de 50 caractères et unique
dateOuv	Date d'ouverture d'une manifestation	Date
dateFer	Date de fermeture d'une manifestation	Date
codeHalle	Code d'une halle	Chaîne de 3 caractères
libHalle	Libellé d'une halle	Chaîne de 50 caractères et unique
nbStandsMax	Nombre maximal de stands d'une halle	Entier strictement positif
codeStand	Code d'un stand	Chaîne de 3 caractères
designation	Désignation d'un stand	Chaîne de 50 caractères
superficie	Superficie d'un stand exprimé en m ²	Entier compris entre 30 et 100
etat	Etat d'un stand (Libre/Occupé)	Un caractère contenant la lettre "L" si le stand est libre ou "O" s'il est occupé
codeCl	Code d'un client	Chaîne de 5 caractères
nomCl	Nom d'un client	Chaîne de 50 caractères
telCl	Numéro de téléphone d'un client	Chaîne de 8 caractères et unique
emailCl	Email d'un client	Chaîne de 50 caractères et unique
pays	Pays d'un client	Chaîne de 50 caractères
dateLoc	Date de location d'un stand	Date
prix	Prix de location d'un stand	Réel strictement positif
codeEval	Code d'une évaluation	Entier auto-incrémenté
dateEval	Date d'une évaluation	Date
noteAccueil	Note attribuée au service accueil	Entier compris entre 0 et 5
noteConfort	Note attribuée au service confort	Entier compris entre 0 et 5
noteSignal	Note attribuée au service signalétique	Entier compris entre 0 et 5
noteLogistique	Note attribuée au service logistique	Entier compris entre 0 et 5

Le site est composé des pages "EvaluationManif.html", "mesStyles.css", "mesScripts.js" et "Evaluation.php".

Un aperçu de la page web "EvaluationManif.html", de son code ainsi que le code du fichier "Evaluation.php" sont présentés ci-après.

N.B. : Le fichier "Evaluation.php" est développé autour de la base de données décrite dans la page 11, sans tenir compte des modifications apportées à cette base dans l'exercice N° 4 de la partie A.

❖ Page "EvaluationManif.html"

Évaluation d'une manifestation

Manifestation : Choisir une manifestation ▼

Service	Note
Accueil :	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
Confort :	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
Signalétique :	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
Logistique :	☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐
Note totale :	0/20

- Fig.1 : Aperçu de la page "EvaluationManif.html" -


```

1 <!DOCTYPE HTML>
2 <html lang="fr">
3 <head>
4   <title>Évaluation d'une manifestation</title>
5   <meta charset="utf-8">
6   <meta name="Description" content="Cette page permet l'évaluation des manifestations.">
7   <link rel="stylesheet" type="text/css" href="mesStyles.css">
8   <script src="mesScripts.js"> </script>
9 </head>
10 <body onload="inconnue()">
11   <form id="f" name="f" action="evaluation.php" method="POST" onsubmit="return verif ()">
12     <fieldset>
13       <legend>Évaluation d'une manifestation</legend>
14       <label>Manifestation :</label>
15       <select id="manif" name="manif">
16         <option selected>Choisir une manifestation</option>
17         <option value="M0001">Foire du livre</option>
18         <option value="M0002">Salon de l'Art</option>
19         <option value="M0003">Séminaire des médecins</option>
20       </select><br>
21       <hr>
22       <label class="labserv">Service</label> <label id="labnote">Note</label><br><br>
23       <label class="service">Accueil :</label>
24       <input type="range" id="serv1" name="serv1" min="0" max="5"
25         onchange="imagenote('serv1', 'imgser1'); calcultotal();"/>
26       <br>
27       <label class="service">Confort : </label>
28       <input type="range" id="serv2" name="serv2" min="0" max="5"
29         onchange="imagenote('serv2', 'imgser2'); calcultotal();"/>
30       <br>
31       <label class="service">Signalétique :</label>
32       <input type="range" id="serv3" name="serv3" min="0" max="5"
33         onchange="imagenote('serv3', 'imgser3'); calcultotal();"/>
34       <br>
35       <label class="service">Logistique :</label>
36       <input type="range" id="serv4" name="serv4" min="0" max="5"
37         onchange="imagenote('serv4', 'imgser4'); calcultotal();"/>
38       <br>
39       <label class="service">Note totale :</label>
40       <span id="total">0/20</span><br>
41     <div>
42       <input name="Valider" type="submit" value="Valider">
43       <input name="Annuler" type="reset" value="Annuler">
44     </div>
45   </fieldset>
46 </form>
47 </body>
48 </html>

```

- Fig.2 : Code HTML de la page "EvaluationManif.html" -

❖ Fichier "Evaluation.php"

```
1 <?php
2 $manif=$_POST["manif"];
3 $note=array($_POST["serv1"], $_POST["serv2"], $_POST["serv3"], $_POST["serv4"]);
4 $tot=$_POST["total"];
5 echo "Le total des notes attribuées est $tot / 20 <br>" ;
6 $datejour=date('Y-m-d') ;
7 $con=mysqli_connect("localhost","root","", "Gestion_Exposition") or die ("Erreur B.D.");
8 $reqSelect="select DateOuv , DateFer from manifestation where (CodeManif ='$manif') ";
9 $resSelect=mysqli_query ($con, $reqSelect) or die ("La requête est erronée.");
10 if ($resreg=mysqli_fetch_array($resSelect))
11 {
12     $date1=$resreg[0];
13     $date2=$resreg[1];
14     if ($date1>$datejour)
15         { echo "La manifestation sélectionnée n'a pas encore commencé."; }
16     else
17     if ($date2<$datejour)
18         { echo "La manifestation sélectionnée est déjà clôturée."; }
19     else
20     {
21         $reqInsert="insert into evaluation (dateEval, noteAccueil, noteConfort , noteSignal, noteLogistique,
22         codeManif) values ('$datejour', $note[0], $note[1], $note[2], $note[3], '$manif')";
23         $resInsert=mysqli_query($con,$reqInsert) ;
24         if ($resInsert)
25             { echo "L'évaluation est enregistrée avec succès."; }
26         else
27             { echo "Echec d'insertion" . mysqli_error($con); }
28     }
29 }
30 else
31     echo "La manifestation n'existe pas !";
32 mysqli_close($con) ;
33 ?>
```

- Fig.3 : Code du fichier "Evaluation.php" -