

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 2026

20

Le sujet comporte 15 pages numérotées de 1/15 à 15/15.

Les pages de 1 à 11 sont à remettre.

Partie A : Gestion de la base de données (12.5 points)

En se référant à la représentation textuelle de la base de données "Gestion_Satellites" et au tableau de description des champs des tables à la page 12, répondre aux questions des exercices qui suivent.

Exercice 1 : Interprétation de la base de données (1.5 point)

- 1) Compléter le tableau ci-dessous en mettant une croix (X) dans la colonne "Valide" si la règle de gestion proposée est vérifiée ou dans la colonne "Invalide" dans le cas contraire.

N°	Règle de gestion	Valide	Invalide
1	Une mission est assurée par une seule navette.		
2	Dans une mission, on ne peut réaliser qu'une seule intervention.		
3	Lors d'une intervention, il est possible d'agir sur plusieurs satellites.		
4	Un astronaute peut avoir plusieurs spécialités.		
5	Un astronaute peut participer à plusieurs interventions.		
6	Un astronaute peut effectuer plusieurs rôles dans une même intervention.		
7	Chaque satellite fait l'objet d'une seule intervention.		
8	Un satellite appartient à un seul opérateur.		

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

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

- a) Pour s'assurer que les valeurs saisies dans le champ "annee" de la table "Navette" sont strictement supérieures à 1970, on doit définir une contrainte d'intégrité :

- ☐ de domaine
☐ référentielle
☐ de table (clé primaire)

- b) Les valeurs de la colonne "idOp" de la table "Satellite" :

- ☐ peuvent être nulles
☐ doivent être uniques (sans doublons)
☐ doivent exister dans la table "Operateur"

- c) Pour obliger l'utilisateur à saisir une valeur parmi "E", "P" et "S" dans le champ "statut" de la table "Intervention", on doit définir en SQL la contrainte :

- ☐ CHECK (statut BETWEEN 'E' AND 'S')
☐ CHECK (statut IN ('E', 'P', 'S'))
☐ CHECK (statut = 'E' AND statut = 'P' AND statut = 'S')

- d) La relation entre la table "Astronaute" et la table "Intervention" est de type :

- ☐ un à un
☐ un à plusieurs
☐ plusieurs à plusieurs

Exercice 2 : Interprétation des requêtes (2 points)

1) Donner le rôle de chacune des requêtes suivantes :

a) ALTER TABLE Navette MODIFY nomNav VARCHAR (50)

.....

b) ALTER TABLE Navette ADD CONSTRAINT CHECK (charge > 1000)

.....

c) ALTER TABLE Mission ADD siteLancement VARCHAR (50)

.....

2) On présente ci-dessous le contenu des deux tables "Operateur" et "Satellite".

Operateur	
idOp	libOp
A01	Arabsat
N02	Nilesat
T03	Turksat

Satellite			
idSat	nomSat	poids	idOp
101	Badr 4	2330	A01
102	Nilesat 101	1840	N02
103	Arabsat 5C	4700	A01

Les requêtes d'insertion ci-dessous ont généré des erreurs lors de leurs exécutions malgré qu'elles sont syntaxiquement correctes.

Question : Indiquer pour chacune de ces requêtes le nom du champ qui a généré l'erreur ainsi que sa cause.

a) INSERT INTO Operateur VALUES ('T03', 'Eutelsat')

Nom du champ :

Cause de l'erreur :

b) INSERT INTO Operateur VALUES ('G04', 'Nilesat')

Nom du champ :

Cause de l'erreur :

c) INSERT INTO Satellite VALUES (104, 'Nilesat 103', 200, 'N02')

Nom du champ :

Cause de l'erreur :

d) INSERT INTO Satellite VALUES (107, 'Badr 7', 1985, 'B01')

Nom du champ :

Cause de l'erreur :

Ne rien écrire ici

e) INSERT INTO Satellite (idSat, poids, idOp) VALUES (108, 600, 'A01')

Nom du champ :

Cause de l'erreur :

Exercice 3 : Manipulation des données (6.5 points)

1) Ecrire les requêtes SQL permettant de répondre aux questions suivantes :

a) Quelles sont les missions effectuées par la navette ayant l'identifiant "001" ? On affichera les champs idMis, dateMis et dateRet.

.....

.....

.....

.....

b) Quelles sont les missions ayant un retard de retour ? On affichera les champs idMis, dateMis, dateRet ainsi que le nombre de jours de retard.

.....

.....

.....

.....

.....

c) Quels sont les satellites de l'opérateur ayant le libellé "Nilesat" ? On affichera, selon l'ordre croissant des poids, les champs nomSat et poids.

.....

.....

.....

.....

.....

.....

.....

d) Quelles sont les interventions de maintenance ? On affichera les champs idInt, description et dateInt.

.....

.....

.....

.....

.....

.....

.....

Ne rien écrire ici

- e) Quels sont les astronautes ayant participé à des interventions sur le satellite dont l'identifiant est **103** ? On affichera sans répétition les champs *idAst* et *nomAst*.

.....

.....

.....

.....

.....

.....

- f) Quels sont les astronautes ayant la spécialité "**Médecin**" qui n'ont jamais participé à des interventions ?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- g) Quelles sont les navettes qui ont réalisé au moins deux missions ? On affichera les champs *idNav*, *nomNav* ainsi que le nombre de missions.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- h) Quels sont les satellites ayant le poids le plus élevé ? On affichera les champs *idSat*, *nomSat*, *poids* et *libOp*.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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 2026

2) Ecrire les requêtes SQL permettant de répondre aux situations suivantes :

a) Le satellite d'identifiant **125** a quitté son orbite et s'est détruit, d'où il n'est plus nécessaire de sauvegarder ses données.

b) Pour augmenter la performance du satellite ayant l'identifiant **124**, la société lui a ajouté un kit de télécommunication dont le poids est égal à **200** kilogrammes.

c) Lors de l'intervention ayant comme identifiant "**M0012**", l'astronaute nommé "**El qarni Ali**" a joué le rôle de "**Adjoint technique**" au lieu de "**Adjoint de communication**".

Exercice 4 : Extension et modification de la base de données (2.5 points)

1) Sachant que plusieurs opérateurs peuvent cofinancer et coexploiter un même satellite, on propose de modifier la structure de la base de données "**Gestion_Satellites**" et ceci en respectant les règles de gestion suivantes :

R1 : un opérateur peut financer plusieurs satellites.

R2 : un opérateur finance un satellite avec un seul montant.

R3 : un satellite peut être financé par plusieurs opérateurs.

Proposer une représentation textuelle des tables "**Satellite**" et "**Operateur**" ainsi que toute nouvelle table indispensable permettant de prendre en compte cette modification, tout en indiquant la **clé primaire** et la (les) **clé(s) étrangère(s)** de chaque table.

Ne rien écrire ici

- 2) Afin d'enrichir la base de données par des informations concernant la nationalité de chaque astronaute ainsi que l'agence spatiale à laquelle il est affecté, le concepteur de la base de données décide d'ajouter deux tables nommées respectivement "**Agence**" et "**Nationalite**" et de modifier la structure de la table "**Astronaute**" en respectant les règles de gestion suivantes :

R4 : Un astronaute a une seule nationalité.

R5 : Un astronaute est affecté à une seule agence.

R6 : Une agence peut avoir plusieurs astronautes.

Le concepteur de la base de données propose les représentations textuelles suivantes :

Proposition	Représentation textuelle
P1	Astronaute (<u>idAst</u> , nomAst, specialite, idAg#) Agence (<u>idAg</u> , nomAg) Nationalite (<u>idNat</u> , nomPays, idAst#)
P2	Astronaute (<u>idAst</u> , nomAst, specialite, idNat#) Agence (<u>idAg</u> , nomAg, idAst#) Nationalite (<u>idNat</u> , nomPays)
P3	Astronaute (<u>idAst</u> , nomAst, specialite, idNat#, idAg#) Agence (<u>idAg</u> , nomAg) Nationalite (<u>idNat</u> , nomPays)
P4	Astronaute (<u>idAst</u> , <u>idNat#</u> , <u>idAg#</u> , nomAst, specialite) Agence (<u>idAg</u> , nomAg) Nationalite (<u>idNat</u> , nomPays)

Questions :

- a) Pour chaque proposition (**P1**, **P2**, **P3** et **P4**) et pour chaque règle de gestion (**R4**, **R5** et **R6**), compléter le tableau suivant par la lettre (**V**) si la règle est vérifiée ou la lettre (**F**) dans le cas contraire.

Important : Toute réponse autre que **V** ou **F** est considérée erronée.

Règle de gestion			
	R4 : Un astronaute a une seule nationalité.	R5 : Un astronaute est affecté à une seule agence.	R6 : Une agence peut avoir plusieurs astronautes.
P1			
P2			
P3			
P4			

- b) En déduire parmi les propositions **P1**, **P2**, **P3** et **P4** celle qui satisfait toutes les règles de gestion (**R4**, **R5** et **R6**).

Partie B : Programmation Web (7.5 points)

On présente dans les pages 13, 14 et 15 :

- un aperçu de la page "intervention.html" (Fig1, page 13),
- le code incomplet du fichier "intervention.html" (Fig2, page 13),
- le code incomplet du fichier "mesControles.js" (Fig3, page 14),
- le code incomplet du fichier "intervention.php" (Fig4, page 15),
- une liste de quelques propriétés CSS (page 15).

1) Interprétation et complétion du code HTML (1.5 point)

En se référant à la figure Fig1 et aux lignes du code HTML (Fig2), mettre une croix (X) dans la case correspondante à la proposition correcte.

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

- a) A la ligne 4, pour afficher correctement les caractères accentués (é, è, ê, ...) dans cette page, on ajoute à l'élément **meta** :

- ☐ **lang** = "fr"
☐ **charset** = "utf-8"
☐ **content** = "fr"

- b) A la ligne 6, la définition **rel** = "stylesheet" permet de spécifier :

- ☐ le chemin d'accès de la ressource téléchargée.
☐ le type de la ressource téléchargée.
☐ la nature de la relation entre la page HTML et la ressource externe.

- c) A la ligne 9, pour afficher le résultat retourné par l'appel du fichier "intervention.php" dans l'élément **iframe** défini à la ligne 35, on ajoute :

- ☐ **target** = "zone"
☐ **target** = "cible"
☐ **target** = "conteneur"

- d) A la ligne 11, pour avoir le texte "Nouvelle intervention" comme présenté dans la figure Fig1, on ajoute :

- ☐ **<figcaption>** Nouvelle intervention **</figcaption>**
☐ **<title>** Nouvelle intervention **</title>**
☐ **<legend>** Nouvelle intervention **</legend>**

- e) La déclaration au niveau des lignes 13 et 14 permet de cocher un seul bouton radio à la fois car les deux boutons ont :

- ☐ le même **name**
☐ des **id** différents
☐ des **value** différentes

- f) Au niveau des lignes 13 et 14, pour appeler la fonction "**desactiver**" lorsque l'utilisateur coche avec la souris l'un des deux boutons radio, on ajoute :

☐ **onblur** = "desactiver(false)"
☐ **onclick** = "desactiver(false)"
☐ **onmouseover** = "desactiver(false)"

- g) A la ligne 15, pour obliger l'utilisateur à choisir une valeur supérieure ou égale à "**01/01/2020**" pour le champ du formulaire nommé "**dateInt**", on ajoute :

☐ **min** = "2020-01-01"
☐ **max** = "2020-01-01"
☐ **value** = "2020-01-01"

- h) A la ligne 18, la fonction **afficherLogo()** s'exécute lorsque :

☐ le pointeur de la souris survole un élément de la liste déroulante
☐ le pointeur de la souris quitte la liste déroulante
☐ un nouvel élément de la liste est sélectionné

- i) A la ligne 26, pour afficher par défaut le texte indicatif "**Une lettre suivie de 4 chiffres**" comme présenté dans la figure **Fig1**, on ajoute à l'élément **input** :

☐ **start** = "Une lettre suivie de 4 chiffres"
☐ **placeholder** = "Une lettre suivie de 4 chiffres"
☐ **list** = "Une lettre suivie de 4 chiffres"

- j) A la ligne 28, pour obliger l'utilisateur à saisir un texte dans le champ "**Description**", on ajoute à l'élément **input** l'attribut :

☐ **readonly**
☐ **disabled**
☐ **required**

- k) A la ligne 31, pour afficher le texte "**problème de chargement de l'image**" lorsque l'image ne se charge pas, on ajoute :

☐ **title** = "problème de chargement de l'image"
☐ **alt** = "problème de chargement de l'image"
☐ **name** = "problème de chargement de l'image"

- l) A la ligne 35, l'élément **iframe** est utilisé pour :

☐ définir un cadre en ligne qui affiche une autre ressource
☐ regroupe des éléments HTML dans un bloc de contenu
☐ grouper plusieurs éléments HTML de type block

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 2026

2) Complétion du code CSS (2 points)

En se référant à la figure **Fig2** et à la liste des propriétés CSS (page 15), compléter les règles CSS ci-dessous en tenant compte des situations suivantes :

- a) Mettre en gras tous les éléments de type **label**, les **options** de la liste déroulante ainsi que le bouton **"Enregistrer"**.

..... { font-weight : bold ; }

- b) Définir l'opacité à **50%** pour toute image lorsqu'elle est cliquée.

..... { opacity : 50% ; }

- c) Appliquer aux boutons **"Annuler"** et **"Enregistrer"** une bordure **double** avec des coins arrondis d'un rayon de **10 pixels** et une taille de **16 pixels** pour le texte.

.bouton { : double ;

..... : 10px ;

..... : 16px ; }

- d) Répéter l'image d'arrière-plan **"fond.jpg"** uniquement dans le sens vertical.

form { background-image : url('fond.jpg') ;

background-repeat : ; }

- e) Agrandir la taille des labels de **2 fois** lorsqu'ils sont survolés par la souris.

label : { transform : ;

display : inline-block ; }

- f) Définir une largeur de **180 pixels**, une hauteur de **20 pixels** et une couleur d'arrière-plan **#F7EC75** à la classe nommée **saisie**.

..... { : ;

..... : 20 px ;

..... : #F7EC75 ; }

3) Interprétation et complétion du code JavaScript (2.5 points)

En se référant aux figures **Fig2** et **Fig3**, répondre aux questions suivantes :

- a) Ecrire le corps de la fonction "**initialiser**" permettant d'appeler la fonction "**cacherLogo**" qui permet de rendre invisible une image et la fonction "**desactiver**" qui permet de rendre les éléments du formulaire inactifs à l'exception des deux boutons radio.

function initialiser()

{
..... }

- b) Réécrire la ligne numéro **8** du code HTML (**Fig2**) afin d'appeler la fonction "**initialiser**" lors du chargement de la page.

- c) La fonction "**validerMission**" (**Fig3**, ligne **18**), dont le code est incomplet, permet de vérifier la validité de l'identifiant d'une mission qui doit être une chaîne de 5 caractères dont le premier est une lettre suivie de 4 chiffres.

Question : Pour chaque ligne de la fonction "**validerMission**", compléter le tableau suivant par le code JavaScript adéquat à la tâche concernée.

N°	Tâche	Code JavaScript
①	Remplacer le contenu de la variable " idMis " par son équivalent en majuscule.	
②	Vérifier que le 1 ^{er} caractère du contenu de la variable " idMis " est une lettre alphabétique.	
③	Modifier le style de la bordure de l'élément dont l'identifiant est " idMis " par des tirets.	
④	Afficher le contenu de la variable " msg " dans le paragraphe ayant l'identifiant " erreur ".	

- d) Suite à la sélection du satellite "**Nilesat 101**" de la liste déroulante, la fonction "**afficherLogo**" sera exécutée. Compléter le tableau ci-dessous par la valeur finale de chacune des variables suivantes :

Variable	Valeur
sat	
p	
source	
apparence	

4) Interprétation et complétion du code PHP (1.5 point)

En se référant à la figure **Fig4**, mettre la lettre (**V**) dans la case correspondante à la proposition correcte ou la lettre (**F**) dans le cas contraire.

Important : Toute réponse autre que **V** ou **F** est considérée erronée.

- a) A la ligne **6**, pour stocker dans la variable **\$dateHeure** la concaténation de la date et de l'heure récupérées à partir du formulaire, séparées par le caractère espace, on utilise l'expression :

☐	\$dateInt . "_" . \$heureInt ;	// "_" : désigne le caractère espace
☐	\$_POST["dateInt"] . "_" . \$_POST["heureInt"] ;	// "_" : désigne le caractère espace
☐	\$_POST["dateInt"] + "_" + \$_POST["heureInt"] ;	// "_" : désigne le caractère espace

- b) A la ligne **8**, pour stocker dans la variable **\$desc** la conversion en minuscules du contenu de la variable **\$description** sans les espaces du début et de fin, on utilise :

☐	\$desc = trim (strtolower (\$description)) ;
☐	\$desc = strtolower (trim (\$description)) ;
☐	\$desc = strtolower (\$description) ; \$desc = trim (\$desc) ;

- c) A la ligne **11**, pour vérifier que la mission récupérée à partir du formulaire existe dans la base de données, on utilise :

☐	if (mysqli_num_rows (\$res) > 0)
☐	if (mysqli_affected_rows (\$res) > 0)
☐	if (\$lig = mysqli_fetch_array (\$res))

- d) A la ligne **24**, pour stocker dans la variable **\$numInt**, uniquement les 4 derniers caractères de son contenu, on utilise :

☐	\$numInt = substr (\$numInt , strlen (\$numInt) - 4 , 4) ;
☐	while (strlen (\$numInt) > 4) { \$numInt = substr (\$numInt , 1 , strlen (\$numInt) - 1) ; }
☐	\$numInt = substr (\$numInt , strlen (\$numInt) , 4) ;

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT * * * Session principale 2026	
	Épreuve : Systèmes et Technologies de l'Informatique	Section : Sciences de l'informatique
	Durée : 3h	Coefficient de l'épreuve : 1.8

N° d'inscription

SpaceAst est une société aérospatiale qui organise des missions spatiales à bord de navettes, équipées par des astronautes, pour assurer des interventions de déploiement (mise en place) et de maintenance des satellites. Pour gérer les différentes interventions, SpaceAst exploite la base de données simplifiée nommée "**Gestion_Satellites**" décrite par la représentation textuelle suivante :

Navette (idNav, nomNav, annee, charge)
Mission (idMis, dateMis, datePre, dateRet, idNav#)
Operateur (idOp, libOp)
Satellite (idSat, nomSat, poids, idOp#)
Intervention (idInt, description, dateInt, statut, idMis#, idSat#)
Astronaute (idAst, nomAst, specialite)
Participation (idAst#, idInt#, role)

Sachant que :

- **Navette** : contient les informations relatives aux navettes spatiales.
- **Mission** : contient les informations relatives aux missions des navettes.
- **Operateur** : contient les informations relatives aux opérateurs des satellites.
- **Satellite** : contient les informations relatives aux satellites.
- **Intervention** : contient les informations relatives aux interventions de mise en place ou de maintenance des satellites.
- **Astronaute** : contient les informations relatives aux astronautes.
- **Participation** : contient les informations relatives aux astronautes participant aux interventions réalisées.

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

Champ	Description	Type et propriété(s)
idNav	Identifiant d'une navette spatiale.	Chaîne de 3 caractères
nomNav	Nom d'une navette spatiale.	Chaîne non vide de 20 caractères et unique
annee	Année de fabrication d'une navette spatiale.	Entier strictement supérieur à 1970
charge	Charge maximale d'une navette exprimée en kg.	Entier strictement positif
idMis	Identifiant d'une mission.	Chaîne de 5 caractères commençant par une lettre
dateMis	Date de départ d'une navette pour réaliser une mission.	Date non vide
datePre	Date prévue de retour d'une navette après une mission.	Date non vide
dateRet	Date réelle de retour d'une navette après une mission.	Date non vide
idOp	Identifiant d'un opérateur qui finance un satellite.	Chaîne non vide de 3 caractères
libOp	Libellé d'un opérateur qui finance un satellite.	Chaîne non vide de 50 caractères et unique
idSat	Identifiant d'un satellite.	Entier strictement positif
nomSat	Nom d'un satellite.	Chaîne non vide de 50 caractères et unique
poids	Poids d'un satellite exprimé en kg.	Entier strictement supérieur à 500
idInt	Identifiant d'une intervention.	Chaîne de 5 caractères commençant par la lettre "M" ou la lettre "D" suivie de 4 chiffres. ("M": Maintenance; "D": Déploiement)
description	Description d'une intervention.	Chaîne non vide de 100 caractères
dateInt	Date et heure d'une intervention.	Date et heure non vide
statut	Statut d'une intervention.	Un caractère ("E" : Echec , "P" : Prévue, "S" : Succès)
idAst	Identifiant d'un astronaute.	Chaîne de 3 caractères
nomAst	Nom et prénom d'un astronaute.	Chaîne non vide de 50 caractères et unique
specialite	Spécialité d'un astronaute.	Chaîne de 50 caractères
role	Rôle d'un astronaute dans une intervention.	Chaîne non vide de 50 caractères

N.B. : Les règles d'intégrité référentielles entre les tables sont maintenues avec la clause SQL "**ON DELETE CASCADE**".

Annexe de la partie B (Programmation Web)

➤ Fig1 : aperçu de la page *intervention.html*

Nouvelle intervention

Nature intervention	☐ Déploiement ☐ Maintenance	
Date	jj/mm/aaaa	Heure --:--
Satellite	Sélectionner un satellite	
Identifiant mission	Une lettre suivie de 4 chiffres	
Description		
		Annuler Enregistrer

➤ Fig2 : code incomplet du fichier *intervention.html*

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta name = "Interventions de spaceAst" ..... >
5     <script src = "mesControles.js"> </script>
6     <link href = "mesStyles.css" rel = "stylesheet" type = "text/css" >
7 </head>
8 <body>
9     <form action = "intervention.php" method = "post" onsubmit = " return verifier( )" ..... >
10     <fieldset>
11         .....
12     <label> Nature intervention </label>
13     <input name = "natInt" type = "radio" id = "rd1" value = "D" ..... > Déploiement
14     <input name = "natInt" type = "radio" id = "rd2" value = "M" ..... > Maintenance <br>
15     <label> Date </label> <input name = "dateInt" type = "date" id = "dateInt" ..... >
16     <label> Heure </label> <input name = "heureInt" type = "time" id = "heureInt"> <br>
17     <label> Satellite </label>
18     <select name = "listSat" id = "listSat" onchange = " afficherLogo( )" >
19         <option > Sélectionner un satellite </option>
20         <option value = "Arabsat 5C"> Arabsat 5C </option>
21         <option value = "Nilesat 101"> Nilesat 101 </option>
22         <option value = "Badr 4"> Badr 4 </option>
23         <option value = "Turksat 6A"> Turksat 6A </option>
24     </select> <br>
25     <label> Identifiant mission </label>
26     <input name = "idMis" type = "text" id = "idMis" class = "saisie" onblur = "validerMission( )" ..... ><br>
27     <label> Description </label>
28     <input name = "desInt" type = "text" id = "desInt" class = "saisie" ..... ><br>
29     <input type = "reset" id = "exit" class = "bouton" value = "Annuler" >
30     <input type = "submit" id = "save" class = "bouton" value = "Enregistrer" >
31     <img src = "" id = "logo" ..... >
32     <p id = "erreur"> </p>
33 </fieldset>
34 </form>
35 <iframe name = "zone" id = "cible" class = "conteneur"> </iframe>
36 </body>
37 </html>
```

➤ Fig3 : code incomplet du fichier mesControles.js

```

1  /* cacherLogo permet de cacher l'image ayant l'identifiant "logo" en attribuant la valeur "hidden" à la
2  propriété de style "visibility" */
3  function cacherLogo() {
4      document.getElementById( "logo" ).style.visibility = "hidden"
5  }
6
7  /* desactiver permet de modifier l'état d'activation de tous les éléments du formulaire sauf les 2 boutons
8  radio, les éléments seront inactifs lorsque la valeur du paramètre "etat" est égale à true */
9  function desactiver(etat) {
10     document.getElementById( "dateInt" ).disabled = etat
11     document.getElementById( "heureInt" ).disabled = etat
12     document.getElementById( "listSat" ).disabled = etat
13     document.getElementById( "idMis" ).disabled = etat
14     document.getElementById( "desInt" ).disabled = etat
15     document.getElementById( "exit" ).disabled = etat
16     document.getElementById( "save" ).disabled = etat
17 }
18 function validerMission() {
19     const idMis = document.getElementById( "idMis" ).value
20     ❶
21     const taille = idMis.length
22     if ( ( taille == 5 ) && ( ❷ ) && ( !isNaN ( idMis.replace( idMis[0] , "" ) ) ) )
23     { msg = ""
24       document.getElementById( "idMis" ).value = idMis    }
25     else
26     { msg = "Identifiant non conforme"
27       document.getElementById( "idMis" ) ❸ = "dashed" }
28     ❹
29 }
30 function rechercherLogo( ch , t ) {
31     let pos = -1
32     do
33         pos = pos + 1
34     while ( pos < t.length && t[pos] != ch )
35     return pos
36 }
37 function afficherLogo() {
38     let source = ""
39     let apparence = "hidden"
40     const idSat = [ "Arabsat 5C" , "Badr 4" , "Nilesat 101" , "Turksat 6A" ]
41     const logos = [ "op1.jpg" , "op2.jpg" , "op3.jpg" , "op4.jpg" ]
42     let sat = document.getElementById( "listSat" ).value
43     let p = rechercherLogo( sat , idSat )
44     if ( p < idSat.length )
45     { source = logos[p]
46       apparence = "visible"
47     }
48     document.getElementById( "logo" ).src = source
49     document.getElementById( "logo" ).style.visibility = apparence
50 }

```

➤ Fig4 : code incomplet du fichier intervention.php

```

1 <?php
2 $connect = mysqli_connect ("localhost" , "root" , "" , "Gestion_satellites") or die ("Problème de connexion") ;
3 $nature = $_POST["natInt"] ;
4 $identifiantM = $_POST["idMis"] ;
5 $nom = $_POST["listSat"] ;
6 $dateHeure = .....
7 $description = $_POST["desInt"] ;
8 .....
9 $req1 = " SELECT * FROM mission WHERE idMis = '$identifiantM' " ;
10 $res = mysqli_query ( $connect , $req1 ) or die ( " Vérifier la requête de recherche des missions " ) ;
11 .....
12 {
13     $req2 = " SELECT idSat FROM Satellite WHERE nomSat = '$nom' " ;
14     $res = mysqli_query ( $connect , $req2 ) or die ("Vérifier la requête de recherche des satellites") ;
15     $lig = mysqli_fetch_array ( $res ) ;
16     $identifiantS = $lig["idSat"] ;
17     $req3 = " SELECT MAX(idInt) AS max FROM Intervention WHERE idInt LIKE '$nature%' " ;
18     $res = mysqli_query ( $connect , $req3 ) or die ( " Requête de recherche erronée " ) ;
19     $numInt = "0001" ;
20     if ( $lig = mysqli_fetch_array ( $res ) )
21     {
22         $vmax = $lig["max"] ;
23         $numInt = "000" . ( substr ( $vmax , 1 ) + 1 ) ;
24         .....
25     }
26     $numInt = $nature . $numInt ;
27     $req4 = " INSERT INTO Intervention
28             VALUES ( '$numInt' , '$desc' , '$dateHeure' , 'P' , '$identifiantM' , '$identifiantS' ) " ;
29     $res = mysqli_query ( $connect , $req4 ) or die ( " Requête d'insertion erronée " ) ;
30     $message = " Insertion effectuée " ;
31 }
32 else
33 { $message = " Vérifier l'identifiant de la mission " ; }
34 echo $message ;
35 mysqli_close ( $connect ) ;
36 ?>

```

➤ Liste de quelques propriétés CSS

font-family	background-color	border	display
font-weight	background-image	width	top
font-style	background-repeat	height	bottom
font-size	background-size	position	left
font	background	float	right
text-align	border-color	padding	overflow
text-shadow	border-style	margin	opacity
text-transform	border-radius	box-shadow	filter
color	border-width	transition	transform