 <h1 style="text-align: center;">EXAMEN FINAL</h1>			
Département	TI	Date	07/06/23
Matière	DEVELOPPEMENT WEB ET MULTIMEDIA 2	Durée	01h30
Enseignants	S.HAMZA, Y.HAMZA, M.MAKKES, A.SOLTANI, A.TRIKI	Doc. Autorisés	NON
Classes	TI 101, TI102, TI103, TI104, TI105, TI106	Calculat. Autorisée	NON
		Nombre de pages	04+ Annexes

Il vous est demandé de remplir intégralement l'entête de votre copie et de soigner la présentation

Exercice 1

10 Pts

La Fédération Tunisienne de Tennis (FTT) veut créer une page web interactive qui permet de suivre les joueuses de Tennis féminin classées dans le Top 10 du classement Mondial WTA.

Pour ce faire, la FTT vous demande de créer l'interface suivante :

Classement WTA		
RANG	NOM & PRENOM	NBPOINTS
1	Iga Swiatek	8940
2	Aryna Sabalenka	7541
3	Jessica Pegula	5205
4	Elena Rybakina	5090
5	Caroline Garcia	5025
6	Coco Gauff	4305
7	Ons Jabeur	3541
8	Maria Sakkari	3391
9	Daria Kasatkina	3275
10	Petra Kvitova	3162
Informations personnelles		
Jabeur, Ons		
<ul style="list-style-type: none"> • Pays: Tunisia • Professionnel depuis: 2010 • Gain Total: 7782641 		

Figure 1: Aperçu de la page "WTA.html"

L'interface doit répondre aux spécifications suivantes :

Spécification 1 : Lors du chargement de la page web, le classement des 10 premières joueuses s'affiche avec :

- leur rang actuel dans le classement mondial WTA
- leur nom et prénom
- Le nombre total de points cumulés par joueuse

Un clic sur une ligne du classement permet d'obtenir les informations personnelles de la joueuse en question.

Spécification 2 : Les informations personnelles sur une joueuse comportent :

- Son nom complet
- Son pays
- L'année depuis laquelle elle est devenue professionnelle
- Son gain total

Travail demandé :

N.B : Les méthodes *insertAdjacentHTML* et *insertAdjacentElement* **ne peuvent pas** être utilisées dans cet exercice

Compléter l'implémentation de la page « WTA.html » qui utilise l'API « All Sports API » dont la documentation est fournie en annexe. Pour cela :

- 1) Ecrire le script d'une fonction **classement()** qui permet de répondre à la spécification 1
- 2) Ecrire le script d'une fonction **afficherInfos(id)** qui prend en paramètre l'identifiant d'une joueuse et qui répond à la spécification 2, sachant que :
 - Les informations sont affichées dans une division « divDetails » qui ne devient visible que suite au clic sur une ligne du classement
 - Le nom complet, le pays, l'année et le gain total sont insérés respectivement dans les éléments *strong*, *span*, *span* et *em* déjà définis dans la partie html de la page.

Code HTML de la page « WTA.html »

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Classement WTA</title>
  <style>      .c1 {color:brown}    </style>
</head>
<body onload="classement()">
  <h2>Classement WTA</h2>
  <table>
    <thead>
      <tr><th>RANG</th><th>NOM & PRENOM </th><th>NBPOINTS</th></tr>
    </thead>
    <tbody></tbody>
  </table>
  <div id="divDetails" hidden>
    <h3 class="c1"> Informations personnelles</h3>
    <strong>.....</strong>
    <ul>
      <li>
        Pays: <span>.....</span>
      </li>
      <li>
        Professionnel depuis: <span class="c1">.....</span>
      </li>
      <li>
        Gain Total: <em>.....</em>
      </li>
    </ul>
  </div>
  <script>
    const urlBase= 'https://allsportsapi2.p.rapidapi.com/api/tennis';
    const options = {
      method: 'GET',
      headers: {
        'X-RapidAPI-Key': '255c99d668mshb475ec51acca46ap1353d0jsnf59640bae26c',
        'X-RapidAPI-Host': 'allsportsapi2.p.rapidapi.com'
      }
    };
  </script>
</body>
</html>
```

Le groupe scolaire EDS, collège et lycée privé souhaite développer un simulateur (figure 2) qui permet de calculer et afficher le total des frais de scolarité annuel d'un élève sachant que :

Le total des frais comporte les frais d'inscription, les frais d'études et les frais de restauration.

Les détails des frais se présentent comme suit :

- Frais d'inscription 500 DT
- Frais d'études 6500 DT pour le niveau collège et 7000 DT pour le niveau Lycée
- Frais de restauration : Cantine : 1650 DT/an ou Panier : 750 DT/an

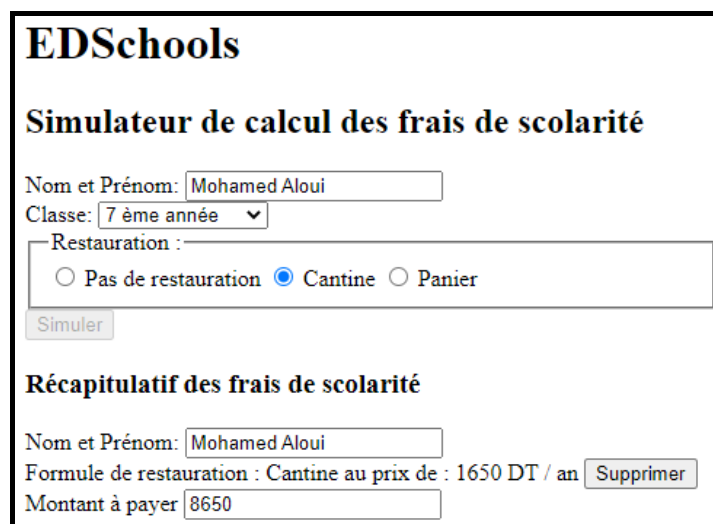


Figure 2: Aperçu de la page "Simulateur.html"

Fonctionnement :

Le parent saisit le nom et le prénom de l'élève ainsi que sa classe puis sélectionne l'un des modes de restauration et clique le bouton « Simuler » pour voir apparaître en bas de la page les informations suivantes :

- Le nom et le prénom de l'élève
- La formule de restauration s'il a choisi « Panier » ou « Cantine »
- Le montant à payer

A noter que le parent peut éliminer les frais de restauration, pour cela, il clique sur le bouton « Supprimer », le montant total change alors automatiquement et les frais sont supprimés.

Le bouton « Simuler » est désactivé après chaque simulation.

Travail demandé :

N.B : Les méthodes `insertAdjacentHTML` et `insertAdjacentElement` **ne peuvent pas** être utilisées dans cet exercice

Compléter l'implémentation de la page « Simulateur.html ». Pour cela :

- 1) Ecrire le script d'une fonction **afficherRestauration()** qui permet d'ajouter dynamiquement (en utilisant le DOM) une division ayant pour id « divR » **avant** la division « divFrais ». Elle comporte un span et un bouton :
 - Le span affiche la formule de restauration choisie selon le modèle suivant :
Formule de restauration : Panier au prix de : 750 DT / an
 - Le bouton « Supprimer » permet de supprimer la division « divR » et met à jour le montant à payer

Il est à noter que la division « divR » n'est ajoutée que si l'utilisateur choisit « Panier » ou « Cantine »

- 2) Ecrire le script d'une fonction **simuler()** invoquée lors d'un clic sur le bouton «Simuler» et qui répond à la spécification donnée.

Code HTML de la page « Simulateur.html »

```
<!doctype html>
<html>
<head>
  <title>Simulateur </title>
  <meta charset="utf-8" />
</head>
<body>
  <h1>EDSchools</h1>
  <h2> Simulateur de calcul des frais de scolarité</h2>
  <form name="fForme">
    <label>Nom et Prénom: </label>
    <input type="text" name="txtNP" /> <br>
    <label>Classe: </label>
    <select id="selClasse">
      <optgroup label="Collège">
        <option value="6500">7 ème année</option>
        <option value="6500">8 ème année</option>
        <option value="6500">9 ème année</option>
      </optgroup>
      <optgroup label="Lycée">
        <option value="7000">1 ère année</option>
        <option value="7000">2 ème année</option>
        <option value="7000">3 ème année</option>
        <option value="7000">4 ème année</option>
      </optgroup>
    </select>
    <fieldset>
      <legend> Restauration :</legend>
      <input type="radio" name="rdR" value="0" /> Pas de restauration
      <input type="radio" name="rdR" value="1650" /> Cantine
      <input type="radio" name="rdR" value="750" /> Panier
    </fieldset>
    <button type="button" name="btnSimuler" onclick="simuler()"> Simuler </button>
  </form>
  <br>
  <div id="divSimulation">
    <div>
      <h3>Récapitulatif des frais de scolarité</h3>
      <label>Nom et Prénom: </label>
      <input type="text" readonly id="tNom"> <br>
    </div>
    <div id="divFrais">
      <label>Montant à payer</label>
      <input type="text" readonly id="tMontant">
    </div>
  </div>
  <script>
    MontantTotal =500;
  </script>
</body>
</html>
```

../..

Annexe 1 : Principaux attributs et méthodes DOM

- Principales méthodes d'accès au DOM

getElementById(id), getElementsByTagName(typeBalise), getElementsByName(name), querySelector(sélecteur), querySelectorAll(sélecteur)

- Quelques propriétés DOM

innerHTML, innerText, childNodes, firstChild, lastChild, parentNode, nextSibling, previousSibling

- Quelques méthodes pour la manipulation DOM

setAttribute(attribut, valeur), createElement(balise), createTextNode(texte), appendChild(nouveauNoeud), insertBefore(nouveauNoeud, noeudRéférence), removeChild(noeudASupprimer), replaceChild(nouveauNoeud).

Annexe 2 : Documentation de l'API : « All Sports API »

Soit l'API "All Sports API" qui offre un ensemble d'informations dans le domaine sportif et qui nécessite une authentification via une clé comme suit :

Clé	Valeur
X-RapidAPI-Key'	255c99d668mshb475ec51acca46ap1353d0jsnf59640bae26c
X-RapidAPI-Host'	allsportsapi2.p.rapidapi.com

L'API offre les 2 endpoints suivants :

- GET <https://allsportsapi2.p.rapidapi.com/api/tennis/rankings/wta>

Cet endpoint fournit un ensemble d'informations sur des joueuses de tennis tels que : le rang, le nom, le nombre de points et l'id.

Exemple : Le résultat de la requête vers cet endpoint donne :

```
{rankings: Array(500), updatedAtTimestamp: 1685394015}
└─ rankings: Array(500)
  └─ [0 ... 99]
    └─ 0:
      bestRanking: 1
      bestRankingDateTimestamp: 1649030400
      ▶ country: {}
      points: 8940
      previousRanking: 1
      ranking: 1
      rankingClass: "team"
      rowName: "Iga Swiatek"
      ▼ team:
        ▶ country: {alpha2: 'PL', name: 'Poland'}
        gender: "F"
        id: 228272
```

Figure 3: Aperçu du corps de la réponse à la requête http affiché dans la fenêtre console

- GET <https://allsportsapi2.p.rapidapi.com/api/tennis/player/{id}>

Cet endpoint fournit des informations sur une joueuse dont l'identifiant est spécifié

Exemple : Le résultat de la requête suivante (en spécifiant l'id de Ons Jabeur qui est 55401) donne

GET <https://allsportsapi2.p.rapidapi.com/api/tennis/player/55401>

```
▼ {team: {...}} ⓘ  
  ▼ team:  
    ▼ country:  
      alpha2: "TN"  
      name: "Tunisia"  
      fullName: "Jabeur, Ons"  
      gender: "F"  
      id: 55401  
    ▼ playerTeamInfo:  
      birthDateTimestamp: 778032000  
      birthplace: "Ksar El Hellar, Tunisia"  
      prizeTotal: 7782641  
      turnedPro: "2010"
```

Figure 4: Aperçu du corps de la réponse à la requête http dans la fenêtre console