```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>P&B UI by RCV - Simple</title>
    <!-- Feuille de style -->
    <link href="./style/style.css" rel="stylesheet">
    <!-- Addons -->
    <link href="https://fonts.googleapis.com/css?family=Roboto:700&display=swap" rel="stylesheet">
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" defer></script>
    <script src="../../frontend/frontend-lib.js" defer></script>
    <!-- Scripts -->
    <script type="module" src="./Script/script.js" defer></script>
</head>
<body>
    <div id="overlay">
        <!-- <img src="./Overlay/overlay.png" alt="overlay" style="z-index: 100;" /> -->
        <!-- <img src="./Overlay/underlay.png" alt="underlay" style="z-index: -100;" /> -->

        <!-- Partie Teams & Scores/timers -->
        <div id="bar">
            <div class="blue text flex-vcenter">
                <span class="big-text" id="blue_name">KARMINE CORP</span>
            </div>
            <div class="blue timer flex-vcenter">
                <span class="big-text" id="blue_timer">:00</span>
            </div>
            <div class="middle">
                <div class="top flex-vcenter">
                    <span class="score-text" id="score">0 - 0</span>
                </div>
                <div class="bot flex-vcenter">
                    <span class="phase-text" id="phase">PICK PHASE 2</span>
                </div>
            </div>
            <div class="red timer flex-vcenter">
```

```html
                <span class="big-text" id="red_timer">:00</span>
            </div>
            <div class="red text flex-vcenter">
                <span class="big-text" id="red_name"> TEAM VITALITY</span>
                <span class="textlol"></span>
            </div>
        </div>

        <!-- Partie sharescreen -->
        <a href="champions.html">
            <div class="container">
                <img src="./Image/gameplay_lol.png" alt="">
            </div>
        </a>

        <!-- Partie pickups gauche & droite -->
        <div class="boxes1">
            <div class="block boxB box1"></div>
            <div class="block boxB box2"></div>
            <div class="block boxB box3"></div>
            <div class="block boxB box4"></div>
            <div class="block boxB box5"></div>
        </div>

        <div class="boxes2">
            <div class="block boxR box6"></div>
            <div class="block boxR box7"></div>
            <div class="block boxR box8"></div>
            <div class="block boxR box9"></div>
            <div class="block boxR box10"></div>
        </div>
```

```
        <!-- Partie bans gauche et droite -->
        <div class="boxes3 blue">
            <div class="ban">
                <div class="ban1 ban1-blue"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-blue"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-blue"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-blue"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-blue"></div>
            </div>
        </div>
        <div class="lc-logo">
            <img src="./Image/LOGO_LIGUECORPO_Small.png" alt="">
        </div>
        <div class="boxes3 red">
            <div class="ban">
                <div class="ban1 ban1-red"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-red"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-red"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-red"></div>
            </div>
            <div class="ban">
                <div class="ban1 ban1-red"></div>
            </div>
        </div>
        <!-- Partie bans & pickup automatique -->
        <!-- <div class="picks blue" id="picks_blue"></div>
        <div class="picks red" id="picks_red"></div>

        <div class="bans blue" id="bans_blue"></div>
        <div class="bans red" id="bans_red"></div> -->
    </div>
</body>
</html>
```

```
// Récupérer les informations (nom + image) des champions côtés ban bleu
for (let i = 0; Array.isArray(BansBlueArray) && BansBlueArray.length > i; i++) {
  const BansBlue = BansBlueArray[i];
  const BansBoxesBleu = blueBansBoxes[i];

  // On récupère dans le fichier le nom et l'image du champion
  const championID = BansBlue;
  const championImage = `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/${championID}_0.jpg`;

  // Créer un élément span pour le nom et l'image du champion
  const championTitle2 = document.createElement("span");
  championTitle2.textContent = championID;
  championTitle2.classList.add("champion-title2-blue");

  // Insérer les valeurs de champion dans la boîte
  BansBoxesBleu.appendChild(championTitle2);
  BansBoxesBleu.style.backgroundImage = `url(${championImage})`;
  BansBoxesBleu.style.backgroundSize = "cover";
  BansBoxesBleu.style.backgroundPosition = "center";

  console.log('Blue ban side :');
  console.log(`Name: ${championID} | Image: ${championImage}`);
  console.log('-------------------------------');
}
// Récupérer les informations (nom + image) des champions côtés ban rouge
for (let i = 0; Array.isArray(BansBlueArray) && BansBlueArray.length > i; i++) {
  const BansRed = BansRedArray[i];
  const BansBoxesRed = redBansBoxes[i];

  // On récupère dans le fichier le nom et l'image du champion
  const championID = BansRed;
  const championImage = `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/${championID}_0.jpg`;

  // Créer un élément span pour le nom et l'image du champion
  const championTitle2 = document.createElement("span");
  championTitle2.textContent = championID;
  championTitle2.classList.add("champion-title2-red");

  // Insérer les valeurs de champion dans la boîte
  BansBoxesRed.appendChild(championTitle2);
  BansBoxesRed.style.backgroundImage = `url(${championImage})`;
  BansBoxesRed.style.backgroundSize = "cover";
  BansBoxesRed.style.backgroundPosition = "center";

  console.log('Blue ban side :');
  console.log(`Name: ${championID} | Image: ${championImage}`);
  console.log('-------------------------------');
}
```

```
// Récupérer les informations (nom + image) des champions côtés pick bleu
for (let i = 0; Array.isArray(PicksBlueArray) && PicksBlueArray.length > i; i++) {
  const PicksBlue = PicksBlueArray[i];
  const PicksBoxesBleu = bluePicksBoxes[i];

  // On récupère dans le fichier le nom et l'image du champion
  const championID = PicksBlue;
  const championImage = `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/${championID}_0.jpg`;

  // Créer un élément span pour le nom et l'image du champion
  const championTitle = document.createElement("span");
  championTitle.textContent = championID;
  championTitle.classList.add("champion-title");

  // Insérer les valeurs de champion dans la boîte
  PicksBoxesBleu.appendChild(championTitle);
  PicksBoxesBleu.style.backgroundImage = `url(${championImage})`;
  PicksBoxesBleu.style.backgroundSize = "cover";
  PicksBoxesBleu.style.backgroundPosition = "center";

  console.log('Blue pick side :');
  console.log(`Name: ${championID} | Image: ${championImage}`);
  console.log('------------------------------');
}
// Récupérer les informations (nom + image) des champions côtés pick rouge
for (let i = 0; Array.isArray(PicksRedArray) && PicksRedArray.length > i; i++) {
  const PicksRed = PicksRedArray[i];
  const PicksBoxesRed = redPicksBoxes[i];

  // On récupère dans le fichier le nom et l'image du champion
  const championID = PicksRed;
  const championImage = `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/${championID}_0.jpg`;

  // Créer un élément span pour le nom et l'image du champion
  const championTitle = document.createElement("span");
  championTitle.textContent = championID;
  championTitle.classList.add("champion-title");

  // Insérer les valeurs de champion dans la boîte
  PicksBoxesRed.appendChild(championTitle);
  PicksBoxesRed.style.backgroundImage = `url(${championImage})`;
  PicksBoxesRed.style.backgroundSize = "cover";
  PicksBoxesRed.style.backgroundPosition = "center";

  console.log('Red pick side :');
  console.log(`Name: ${championID} | Image: ${championImage}`);
  console.log('------------------------------');
}
```

```
// Récupérer les valeurs du localStorage
const picksRedJSON = localStorage.getItem('selectedPicksRed');
const picksBlueJSON = localStorage.getItem('selectedPicksBlue');
const bansRedJSON = localStorage.getItem('selectedBansRed');
const bansBlueJSON = localStorage.getItem('selectedBansBlue');

// Convertir les chaînes JSON en tableaux
const picksRed = picksRedJSON ? JSON.parse(picksRedJSON) : [];
const picksBlue = picksBlueJSON ? JSON.parse(picksBlueJSON) : [];
const bansRed = bansRedJSON ? JSON.parse(bansRedJSON) : [];
const bansBlue = bansBlueJSON ? JSON.parse(bansBlueJSON) : [];

// Utiliser les tableaux récupérés pour restaurer l'état précédent
const selectedPicksRed = new Set(picksRed);
const selectedPicksBlue = new Set(picksBlue);
const selectedBansRed = new Set(bansRed);
const selectedBansBlue = new Set(bansBlue);

console.log("Tous les champions ont été sélectionnés !");
console.log("Champions sélectionnés par l'équipe rouge :", Array.from(selectedPicksRed));
console.log("Champions sélectionnés par l'équipe bleue :", Array.from(selectedPicksBlue));
console.log("Champions bannis par l'équipe rouge :", Array.from(selectedBansRed));
console.log("Champions bannis par l'équipe bleue :", Array.from(selectedBansBlue));

// Récupérer le tableau du localStorage
const PicksBlueArray = JSON.parse(localStorage.getItem('selectedPicksBlue'));
const PicksRedArray = JSON.parse(localStorage.getItem('selectedPicksRed'));
const BansBlueArray = JSON.parse(localStorage.getItem('selectedBansBlue'));
const BansRedArray = JSON.parse(localStorage.getItem('selectedBansRed'));

// Déclaration variables
const bluePicksBoxes = document.getElementsByClassName("boxB");
const redPicksBoxes = document.getElementsByClassName("boxR");
const blueBansBoxes = document.getElementsByClassName("ban1-blue");
const redBansBoxes = document.getElementsByClassName("ban1-red");
```


KARMINE CORP
:00
0 - 0
PICK PHASE
:00
TEAM VITALITY
LIGUECORPO
KARMINE CORP
:00
0 - 0
PICK PHASE
:00
TEAM VITALITY
Amumu
Anivia
Annie
Brand
Blitzcrank
Aatrox
Ahri
Akali
Akshan
Alistar
LIGUECORPO

BONUS :

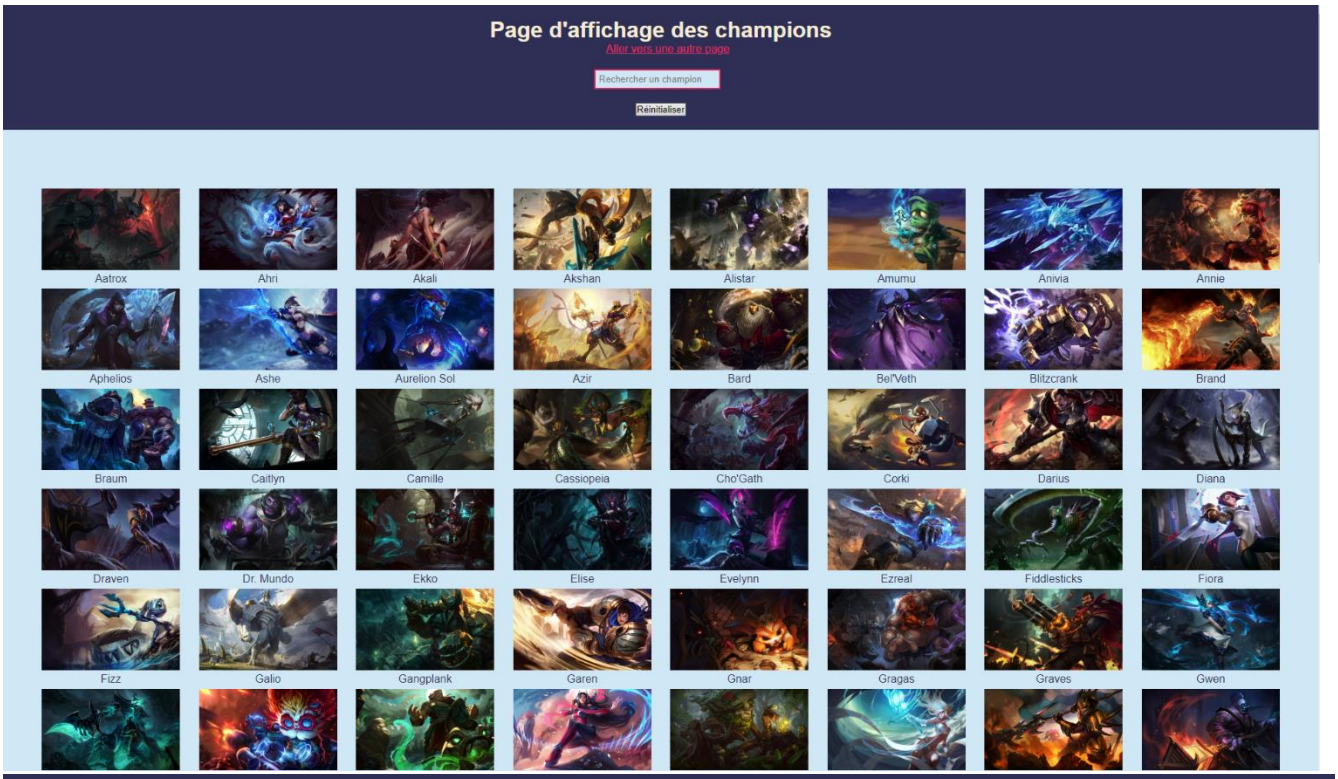
```
fetch("../assets/champions.json")
  .then(response => response.json())
  .then(data => {
    const champions = data.data;

    const championGrid = document.getElementById("champion-grid");
    const searchInput = document.getElementById("search-input");
    const selectedPicksRed = new Set(); // Utilisation d'un Set pour stocker les champions rouges sélectionnés à prendre (évite les doublons)
    const selectedPicksBlue = new Set(); // Utilisation d'un Set pour stocker les champions bleus sélectionnés à prendre (évite les doublons)
    const selectedBansRed = new Set(); // Utilisation d'un Set pour stocker les champions rouges sélectionnés à bannir (évite les doublons)
    const selectedBansBlue = new Set(); // Utilisation d'un Set pour stocker les champions bleus sélectionnés à bannir (évite les doublons)
    const maxPicks = 5; // Maximum de champions à prendre pour chaque équipe
    const maxBans = 5; // Maximum de champions à bannir pour chaque équipe
    let pickCounter = 1; // Compteur pour les champions sélectionnés

    // Fonction pour afficher les champions correspondant à la recherche
    const displayChampions = (searchText) => {
      championGrid.innerHTML = ""; // Efface la grille avant d'afficher les résultats

      Object.values(champions).forEach(champion => {
        const championName = champion.name;
        const championID = champion.id;
        const championImage = `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/${championID}_0.jpg`;

        if (championName.toLowerCase().includes(searchText.toLowerCase())) {
          // Création de la div du champion (même code que précédemment)
          const championDiv = document.createElement("div");
          championDiv.classList.add("champion");

          // Création de l'image du champion
          const championImageElement = document.createElement("img");
          championImageElement.src = championImage;
          championImageElement.alt = championName;

          // Création du nom du champion
          const championNameElement = document.createElement("p");
          championNameElement.textContent = championName;

          // Ajout des éléments à la div du champion
          championDiv.appendChild(championImageElement);
          championDiv.appendChild(championNameElement);
```

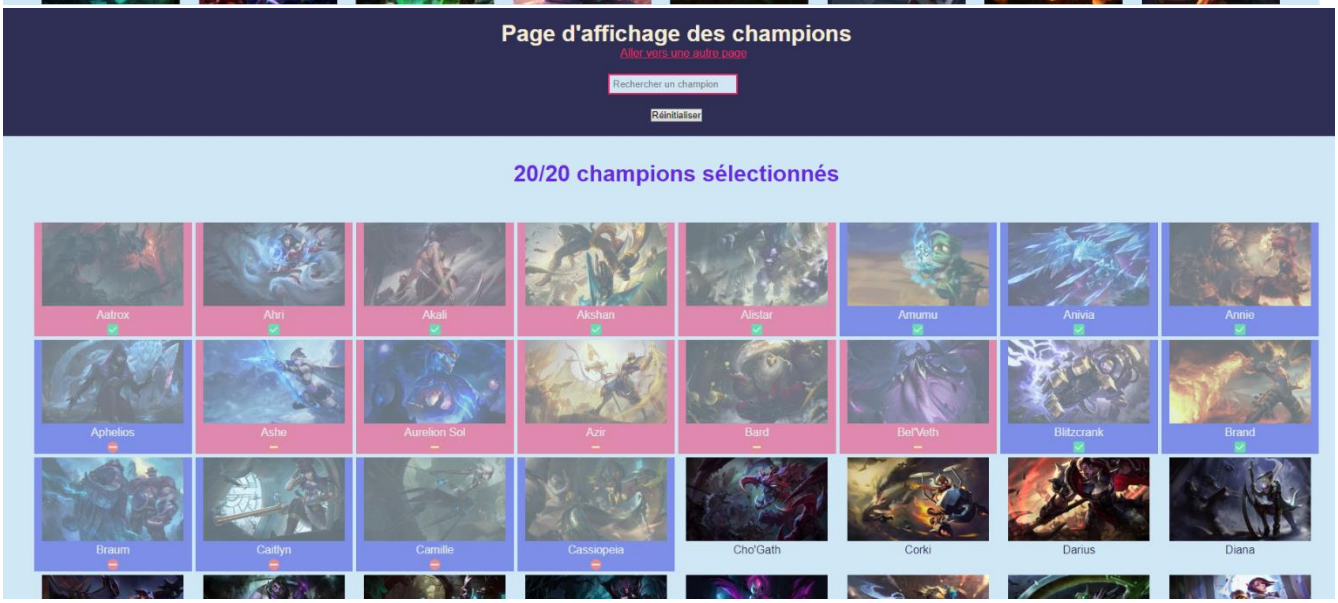
```
if (!selectedPicksRed.has(championID) && !selectedPicksBlue.has(championID) && !selectedBansRed.has(championID) && !selectedBansBlue.has(championID
  if (selectedPicksRed.size < maxPicks) {
    selectedPicksRed.add(championID);
    console.log(`Choisissez le ${ordinalSuffix(pickCounter)} champion de l'équipe rouge :`, selectedChampion);

    // Appliquer la classe CSS pour griser le champion sélectionné
    championDiv.classList.add("picked", "red");

    // Ajouter le logo "✅" pour les champions pick
    const pickLogo = document.createElement("span");
    pickLogo.textContent = "✅";
    championDiv.appendChild(pickLogo);
  } else if (selectedPicksBlue.size < maxPicks) {
    selectedPicksBlue.add(championID);
    console.log(`Choisissez le ${ordinalSuffix(pickCounter - maxPicks)} champion de l'équipe bleue :`, selectedChampion);

    // Appliquer la classe CSS pour griser le champion sélectionné
    championDiv.classList.add("picked", "blue");

    // Ajouter le logo "✅" pour les champions pick
    const pickLogo = document.createElement("span");
    pickLogo.textContent = "✅";
    championDiv.appendChild(pickLogo);
  } else if (selectedBansRed.size < maxBans) {
    selectedBansRed.add(championID);
    console.log(`Choisissez le ${ordinalSuffix(pickCounter - (2 * maxPicks))} champion de l'équipe rouge à bannir :`, selectedChampion);

    // Appliquer la classe CSS pour griser le champion sélectionné
    championDiv.classList.add("banned", "red");

    // Ajouter le logo "⛔" pour les champions ban
    const banLogo = document.createElement("span");
    banLogo.textContent = "⛔";
    championDiv.appendChild(banLogo);
  } else if (selectedBansBlue.size < maxBans) {
    selectedBansBlue.add(championID);
    console.log(`Choisissez le ${ordinalSuffix(pickCounter - (2 * maxPicks) - maxBans)} champion de l'équipe bleue à bannir :`, selectedChampion);

    // Appliquer la classe CSS pour griser le champion sélectionné
    championDiv.classList.add("banned", "blue");

    // Ajouter le logo "⛔" pour les champions ban
    const banLogo = document.createElement("span");
    banLogo.textContent = "⛔";
    championDiv.appendChild(banLogo);
  }
```

```
    // Vérifier si le champion est déjà sélectionné ou non
    const isChampionPickedRed = selectedPicksRed.has(championID);
    const isChampionPickedBlue = selectedPicksBlue.has(championID);
    const isChampionBannedRed = selectedBansRed.has(championID);
    const isChampionBannedBlue = selectedBansBlue.has(championID);

    if (isChampionPickedRed) {
      championDiv.classList.add("picked", "red");
      // Ajouter le logo "✅" pour les champions pick
      const pickLogo = document.createElement("span");
      pickLogo.textContent = "✅";
      championDiv.appendChild(pickLogo);
    } else if (isChampionPickedBlue) {
      championDiv.classList.add("picked", "blue");
      // Ajouter le logo "✅" pour les champions pick
      const pickLogo = document.createElement("span");
      pickLogo.textContent = "✅";
      championDiv.appendChild(pickLogo);
    } else if (isChampionBannedRed) {
      championDiv.classList.add("banned", "red");
      // Ajouter le logo "⛔" pour les champions ban
      const banLogo = document.createElement("span");
      banLogo.textContent = "⛔";
      championDiv.appendChild(banLogo);
    } else if (isChampionBannedBlue) {
      championDiv.classList.add("banned", "blue");
      // Ajouter le logo "⛔" pour les champions ban
      const banLogo = document.createElement("span");
      banLogo.textContent = "⛔";
      championDiv.appendChild(banLogo);
    } else {
      // Ajout de l'événement de clic pour ajouter le champion sélectionné
      championDiv.addEventListener("click", () => {
        const selectedChampion = { name: championName, id: championID, image: championImage };
```

```
            pickCounter++; // Incrémenter le compteur de choix
            updateSelectedCount(); // Mettre à jour le compteur de champions sélectionnés
          }
        });
      }

      // Ajout de la div du champion à la grille
      championGrid.appendChild(championDiv);
    }
  });
};


// Fonction pour mettre à jour le compteur de champions sélectionnés
const updateSelectedCount = () => {
  const totalCount = selectedPicksRed.size + selectedPicksBlue.size + selectedBansRed.size + selectedBansBlue.size;
  const selectedCountElement = document.getElementById("selected-count");
  selectedCountElement.textContent = `${totalCount}/${(maxPicks + maxBans) * 2} champion${totalCount === 1 ? '' : 's'} sélectionné${totalCount === 1 ? '' : 's'}`;

  if (totalCount > 0) {
    document.querySelector("h1").classList.add("selected");
  } else {
    document.querySelector("h1").classList.remove("selected");
  }

  // Vérifier si tous les champions ont été sélectionnés
  if (totalCount === (maxPicks + maxBans) * 2) {
    console.log("Tous les champions ont été sélectionnés !");
    console.log("Champions sélectionnés par l'équipe rouge :", Array.from(selectedPicksRed));
    console.log("Champions sélectionnés par l'équipe bleue :", Array.from(selectedPicksBlue));
    console.log("Champions bannis par l'équipe rouge :", Array.from(selectedBansRed));
    console.log("Champions bannis par l'équipe bleue :", Array.from(selectedBansBlue));
  }
  // Convertir les tableaux en chaînes de caractères JSON
  const picksRedJSON = JSON.stringify(Array.from(selectedPicksRed));
  const picksBlueJSON = JSON.stringify(Array.from(selectedPicksBlue));
  const bansRedJSON = JSON.stringify(Array.from(selectedBansRed));
  const bansBlueJSON = JSON.stringify(Array.from(selectedBansBlue));

  // Stocker les valeurs dans le localStorage
  localStorage.setItem('selectedPicksRed', picksRedJSON);
  localStorage.setItem('selectedPicksBlue', picksBlueJSON);
  localStorage.setItem('selectedBansRed', bansRedJSON);
  localStorage.setItem('selectedBansBlue', bansBlueJSON);
};
const resetButton = document.getElementById("reset-button");
```

```
    resetButton.addEventListener("click", () => {
      // Réinitialiser les sélections en vidant les ensembles
      selectedPicksRed.clear();
      selectedPicksBlue.clear();
      selectedBansRed.clear();
      selectedBansBlue.clear();
      pickCounter = 1;

      // Mettre à jour l'affichage des champions
      displayChampions("");
      updateSelectedCount();
      console.log("Champions réinitialisés avec succès !")
    });

    // Événement d'écoute sur l'input de recherche
    searchInput.addEventListener("input", (event) => {
      const searchText = event.target.value;
      displayChampions(searchText);
    });

    // Afficher tous les champions au chargement de la page
    displayChampions("");
})
.catch(error => console.log(error));


// Fonction pour ajouter le suffixe ordinal (1er, 2e, 3e, etc.)
const ordinalSuffix = (num) => {
  const suffixes = ["e", "er", "e", "e", "e", "e", "e", "e", "e", "e"]; // Suffixes pour les nombres ordinaux
  const lastDigit = num % 10;
  const secondLastDigit = Math.floor(num / 10) % 10;

  if (secondLastDigit === 1) {
    return num + "e";
  } else {
    return num + suffixes[lastDigit];
  }
};
```


Page d'affichage des champions
Aller vers une autre page
Rechercher un champion
Réinitialiser
Aatrox
Ahri
Akali
Akshan
Alistar
Amumu
Anivia
Annie
Aphelios
Ashe
Aurelion Sol
Azir
Bard
Bel'Veth
Blitzcrank
Brand
Braum
Caitlyn
Camille
Cassiopeia
Cho'Gath
Corki
Darius
Diana
Draven
Dr. Mundo
Ekko
Elise
Evelynn
Ezreal
Fiddlesticks
Fiora
Fizz
Galio
Gangplank
Garen
Gnar
Gragas
Graves
Gwen


Page d'affichage des champions
Aller vers une autre page
Rechercher un champion
Réinitialiser
20/20 champions sélectionnés
Aatrox
Ahri
Akali
Akshan
Alistar
Amumu
Anivia
Annie
Aphelios
Ashe
Aurelion Sol
Azir
Bard
Bel'Veth
Blitzcrank
Brand
Braum
Caitlyn
Camille
Cassiopeia
Cho'Gath
Corki
Darius
Diana