

JS Advanced Final Exam

Problem 1. Music Site

Environment Specifics

Please, be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

The following actions are **NOT** supported:

- `.forEach()` with **NodeList** (returned by `querySelector()` and `querySelectorAll()`)
- `.forEach()` with **HTMLCollection** (returned by `getElementsByClassName()` and `element.children`)
- Using the **spread-operator** (`...`) to convert a **NodeList** into an array
- `append()` in Judge (use only `appendChild()`)
- `replaceWith()` in Judge
- `replaceAll()` in Judge
- `closest()` in Judge

If you want to perform these operations, you may use `Array.from()` to first convert the collection into an array.

Use the provided skeleton to solve this problem.

Note: You **can't** and you have no permission to **change** directly the given HTML code (index.html file).

The screenshot shows a web application titled "MUSIC FOR EVERYONE." The interface is divided into several sections:

- Add song.** A form with the following fields:
 - Genre:
 - Name of the song:
 - Author of the song:
 - Date of creation:
 -
- Collection of songs** (pink box)
- Saved songs** (orange box)
- Total Likes: 0** (pink box with a thumbs up icon)

At the bottom of the page, there is a footer that reads "@JS Advanced Exam - Music Site".

Your Task

Write the missing JavaScript code to make the **Music Site** work as expected:

- All fields (genre, name, author, and date) are filled with the correct input
 - Genre, name, author, and date are non-empty strings
- The program should not do anything if any of the input fields are empty.

1. Getting the information about a new song



Add song.
Join the world of music, add the latest hits.
Genre:

Name of the song:

Author of the song:

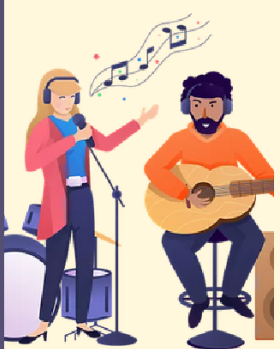
Date of creation:

No spams included

Collection of songs

Saved songs

- When you click the ["Add"] button, the information from the input fields must be added to the **div** with the class "all-hits-container" and then clear input fields.



Add song.
Join the world of music, add the latest hits.
Genre:

Name of the song:

Author of the song:

Date of creation:

No spams included

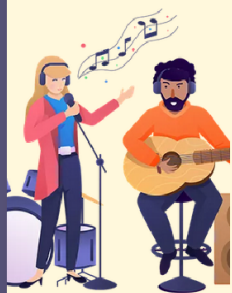
Collection of songs
Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Saved songs

- The HTML structure looks like this:

```
<section id="all-hits">
  <div class="all-hits-container">
    <h1>Collection of songs</h1>

    <div class="hits-info">
      
      <h2>Genre: Pop</h2>
      <h2>Name: Pon de Replay</h2>
      <h2>Author: Rihanna</h2>
      <h3>Date: 26.11.2009</h3>
      <button class="save-btn">Save song</button>
      <button class="like-btn">Like song</button>
      <button class="delete-btn">Delete</button>
    </div>
  </div>
</section>
```



Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:


No spams included

Collection of songs

Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Saved songs

Total Likes: 0




- When the ["Like song"] button is clicked:
 - You need to take a value of the current number of **likes** inside the paragraph in the section with the id "total-likes" and increase the value by one.

```

<section id="total-likes">
  <div class="likes">
    <p>Total Likes: 0</p>
    
  </div>
</section>

```

- The button ["Like song"] for the current song must then be **disabled**, as the user has the right to like the song only once (Once the button is **disabled**, its color will turn gray).



Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:


No spams included

Collection of songs

Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Saved songs

Total Likes: 1





Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:

Add

No spams included

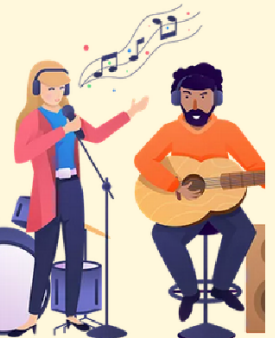
Collection of songs

Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Save song Like song Delete

Saved songs

- When the ["Save song"] button is clicked, you need to move the current song in the **div** with class "saved-container".



Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:

Add

No spams included

Collection of songs

Saved songs

Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Delete

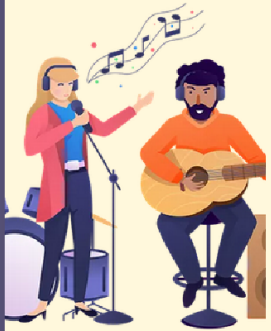
- The HTML structure looks like this:

```

<section id="saved-hits">
  <div class="saved-container">
    <h1>Saved songs</h1>

    <div class="hits-info">
      
      <h2>Genre: Pop</h2>
      <h2>Name: Pon de Replay</h2>
      <h2>Author: Rihanna</h2>
      <h3>Date: 26.11.2009</h3>
      <button class="delete-btn">Delete</button>
    </div>
  </div>
</section>

```



Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:

Add

No spams included

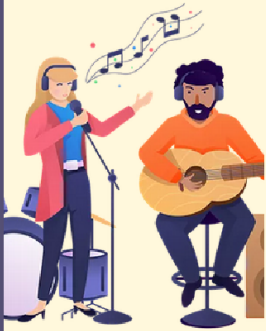
Collection of songs

Saved songs

Genre: Pop
Name: Pon de Replay
Author: Rihanna
Date: 26.11.2009

Delete

- When you click the **["Delete"]** button, the song should be removed from the current section.
Note: When deleting a song, you should not reduce the value of the current number of likes.



Add song.

Join the world of music, add the latest hits.

Genre:

Name of the song:

Author of the song:

Date of creation:

Add

No spams included

Collection of songs

Saved songs

- The HTML structure looks like this:

```

<section id="saved-hits">
  <div class="saved-container">
    <h1>Saved songs</h1>
  </div>
</section>

```

Submission

Submit only your **solve()** function.

GOOD LUCK... 😊