

Problem 2. Task Post

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()` (use only `appendChild()`)
- `prepend()`
- `replaceWith()`
- `replaceAll()`
- `closest()`
- `replaceChildren()`

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 image shows a web interface with a dark blue background and colorful geometric patterns. On the left, there is a light blue rounded rectangle titled "Add New Task". Inside this rectangle, there are three input fields: "Enter title here", "Enter category here", and "Add content..." (a text area). Below these fields is a red button labeled "PUBLISH". To the right of the form, there are two red rounded rectangles. The top one is labeled "Task for review:" and the bottom one is labeled "Uploaded Task:". Both are currently empty.

Your Task

Write the missing JavaScript code to make the **Task Post** work as expected:

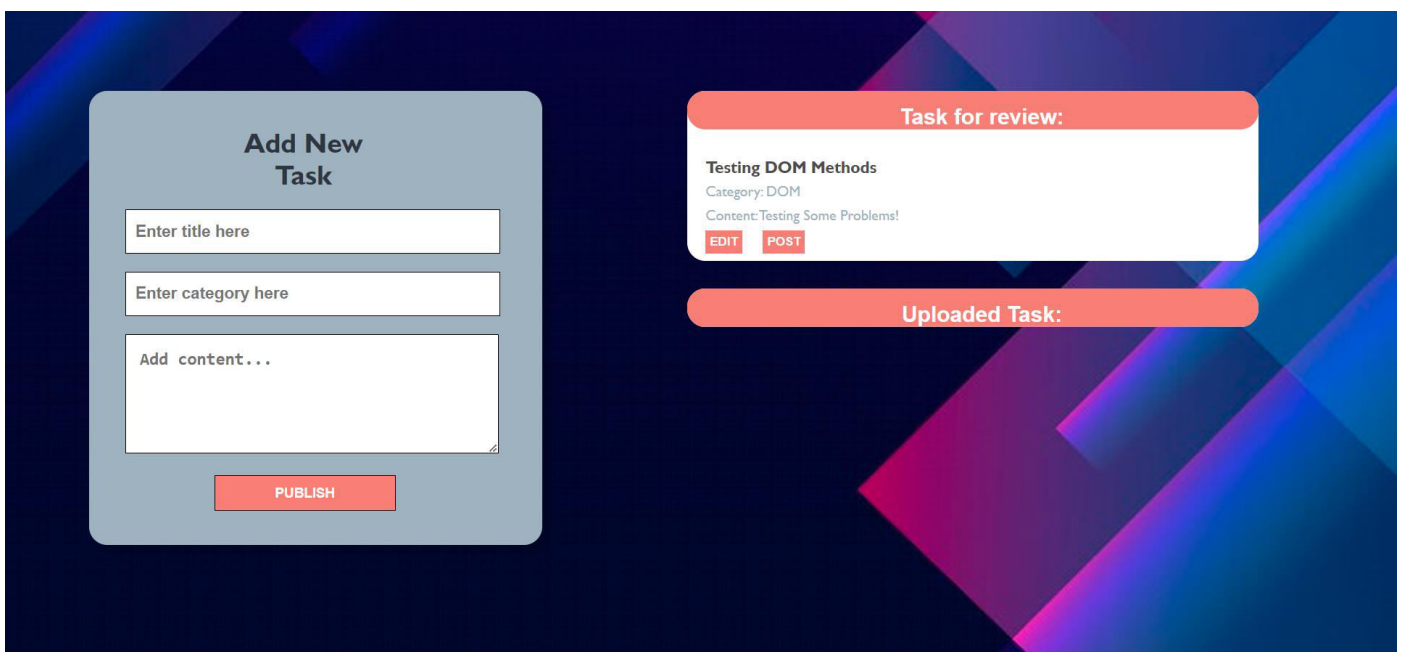
- **Title, category, and content** should be **non-empty strings**. If any of them are empty, the program should not do anything.

1. Getting the information from the task

When you click the **[Publish]** button, the information from the input fields must be added to the `` with the `id "review-list"` and the input fields should be cleared.

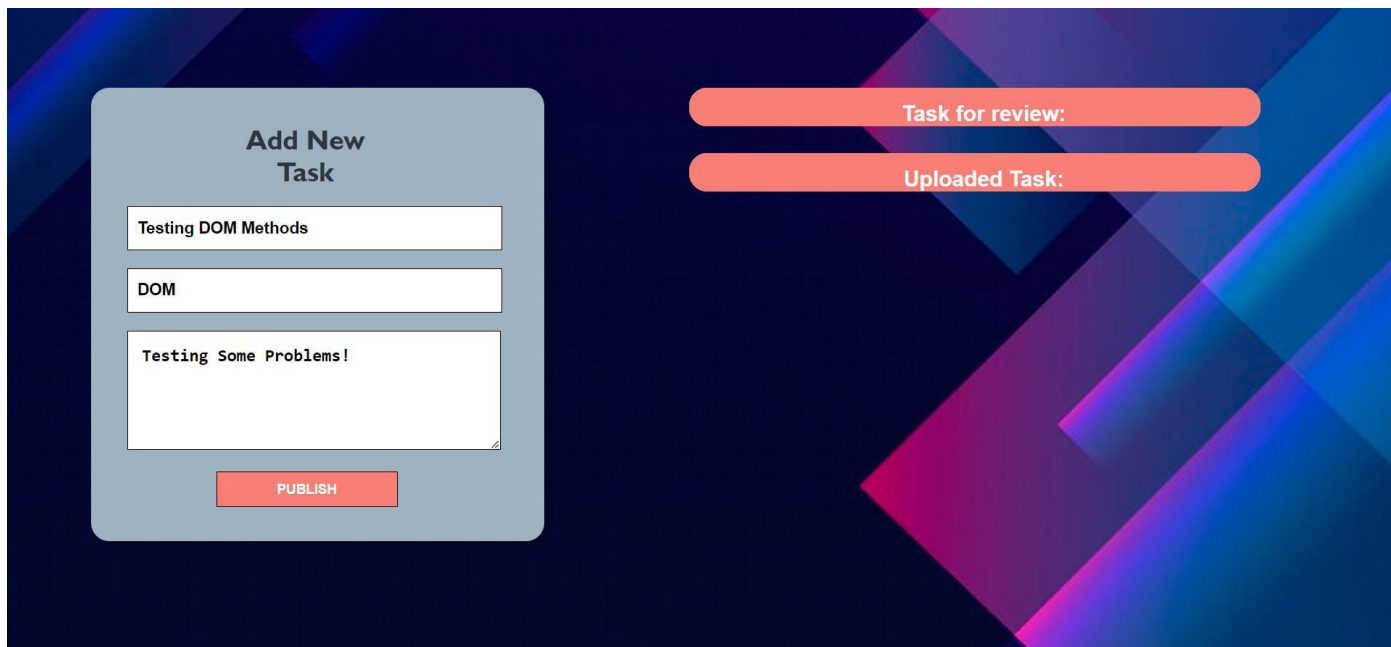
The HTML structure should look like this:

```
▼<ul id="review-list">
  ▼<li class="rpost">
    ▼<article>
      <h4>Testing DOM Methods</h4>
      <p>Category: DOM</p>
      <p>Content: Testing Some Problems!</p>
    </article>
    <button class="action-btn edit">Edit</button>
    <button class="action-btn post">Post</button>
  </li>
</ul>
```

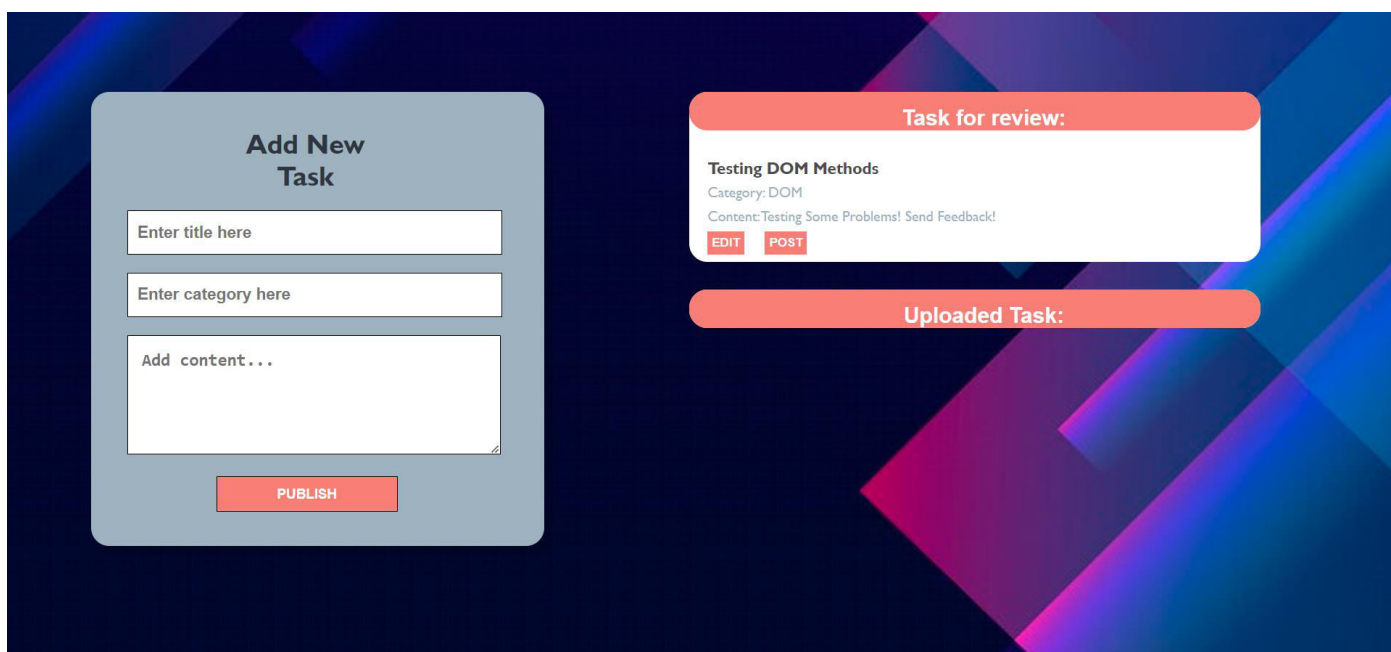


2. Edit information for task

When the **[Edit]** button is clicked, the information from the post must be sent to the input fields on the left side and the record should be deleted from the ` "review-list"`.



After editing the information, add a new item to the `` with the updated information.



3. Post Task

When you click the **[Post]** button, the task must be **deleted** from the `` with **id "review-list"** and appended to the `` with **id "published-list"**.

The **buttons** [Edit] and [Post] should be removed from the `` element.

```
▼<ul id="published-list">
  ▼<li class="rpost">
    ▼<article>
      <h4>Testing DOM Methods</h4>
      <p>Category: DOM</p>
      <p>Content: Testing Some Problems! Send Feedback!</p>
    </article>
  </li>
</ul>
```

The screenshot displays a web interface with a dark blue background and abstract geometric shapes. On the left, a light blue rounded rectangle titled "Add New Task" contains three input fields: "Enter title here", "Enter category here", and "Add content...". Below these fields is a red "PUBLISH" button. On the right, two stacked white rounded rectangles with red headers show the task details. The top header is "Task for review:" and the bottom header is "Uploaded Task:". The content of the "Uploaded Task:" box matches the HTML code shown above: "Testing DOM Methods", "Category: DOM", and "Content: Testing Some Problems! Send Feedback!".

Submission

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