# JS Advanced Exam

# Problem 01.Tech Troubleshoot Hub

**Environment Specifics**

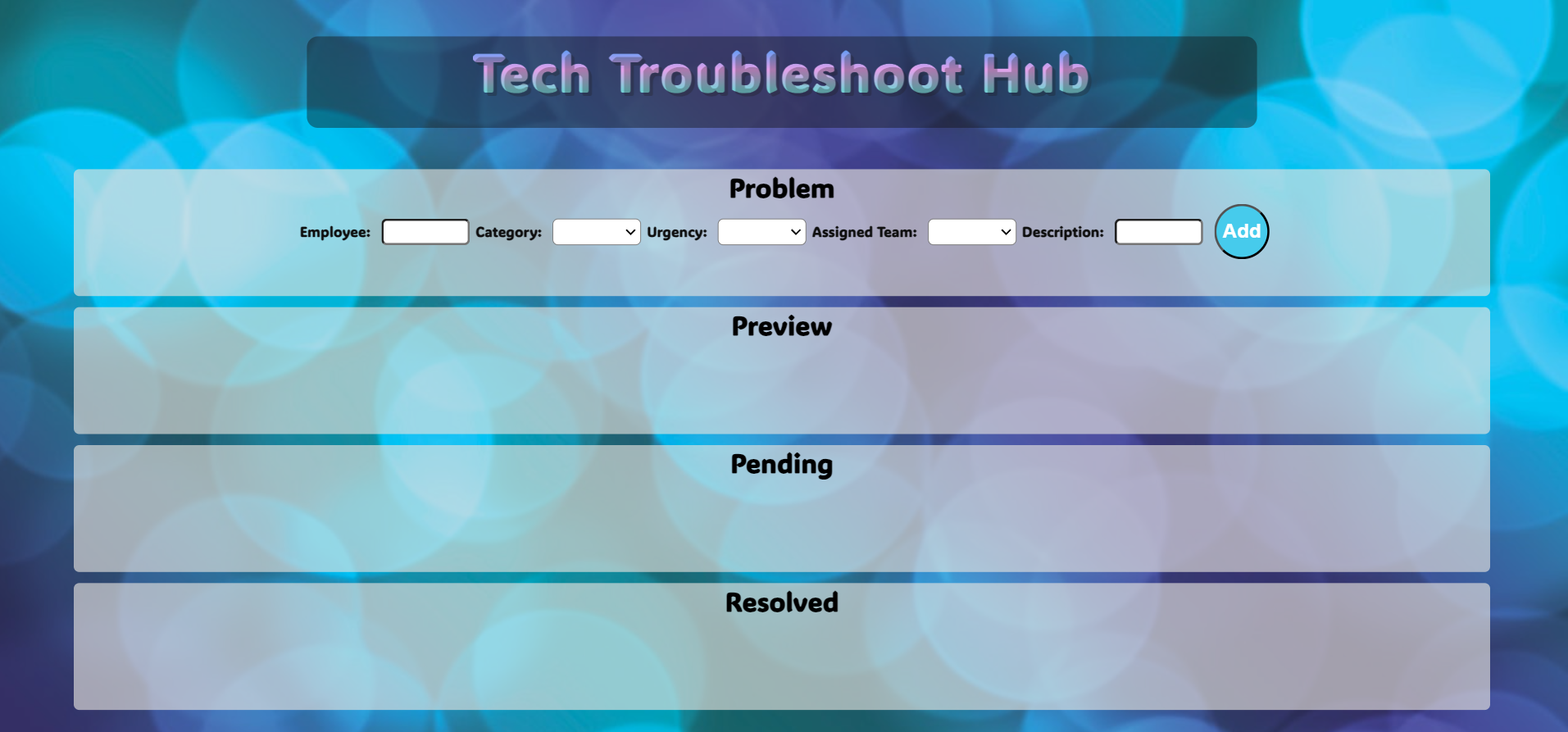
Please be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

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()**)
* **prepend()**
* **replaceWith()**
* **replaceAll()**
* **closest()**
* **replaceChildren()**
* Always turn the collection into a **JS array** (forEach, forOf, et.)

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.**

**Write the missing functionality** of this user interface. The functionality is divided in the following steps:

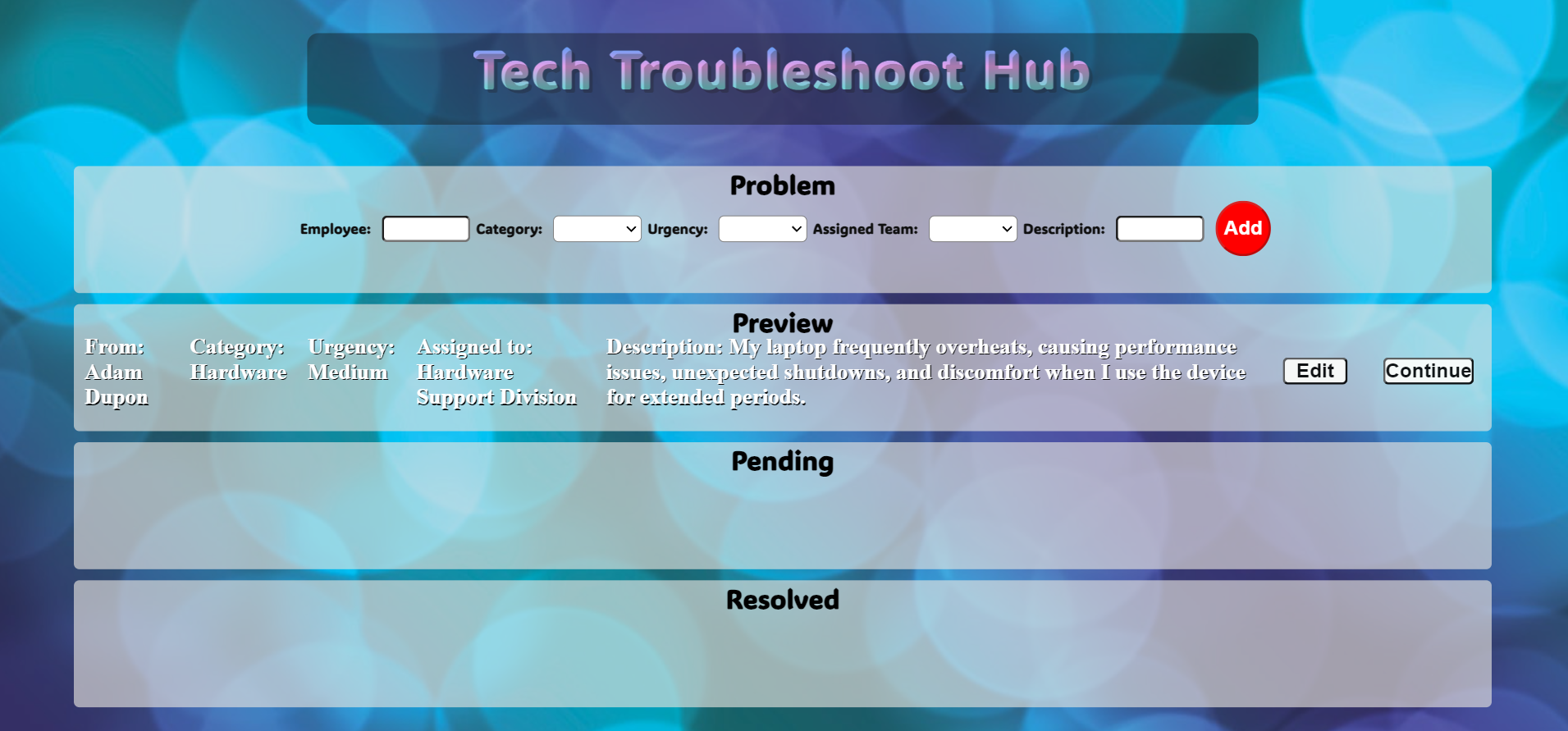
**Your Task**

**Write the missing JavaScript code** to make the **Tech Troubleshoot Hub** work as expected:

All fields **(Employee, Category, Urgency, Assigned Team,** and **Description)** are **filled with the correct input**

* **Employee, Category, Urgency, Assigned Team,** and **Description** are **non**-**empty** **strings**. If any of them is empty, the program should not do anything.

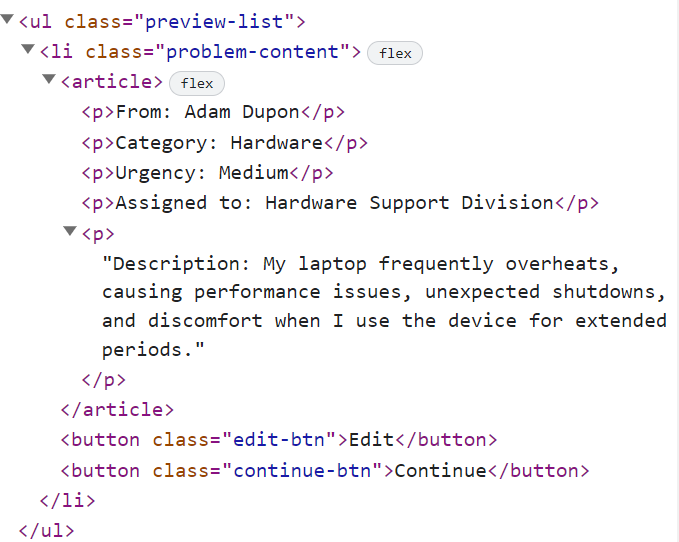
1. **Getting the information from the form**



• On clicking the **["Add"]** button, the information from the input fields is listed in the "**Preview**" section by adding a **list item** to the **".preview-list"** unordered list.

* The list item should follow the same text format and order as shown in the provided picture.
* Upon clicking the button, the **input** fields must be **cleared**, and the **["Add"]** button should be **disabled**. Additionally, the **"Edit"** and **"Continue"** buttons need to be added.

The HTML structure looks like this:



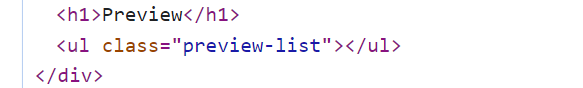
1. **Editing the information**

**The functionality here is the following:**

* **When the "**Edit" **button is clicked, all of the information is loaded in the input fields from step 1 and all buttons in Preview section are removed while the** ["Add"] **button is enabled again.**

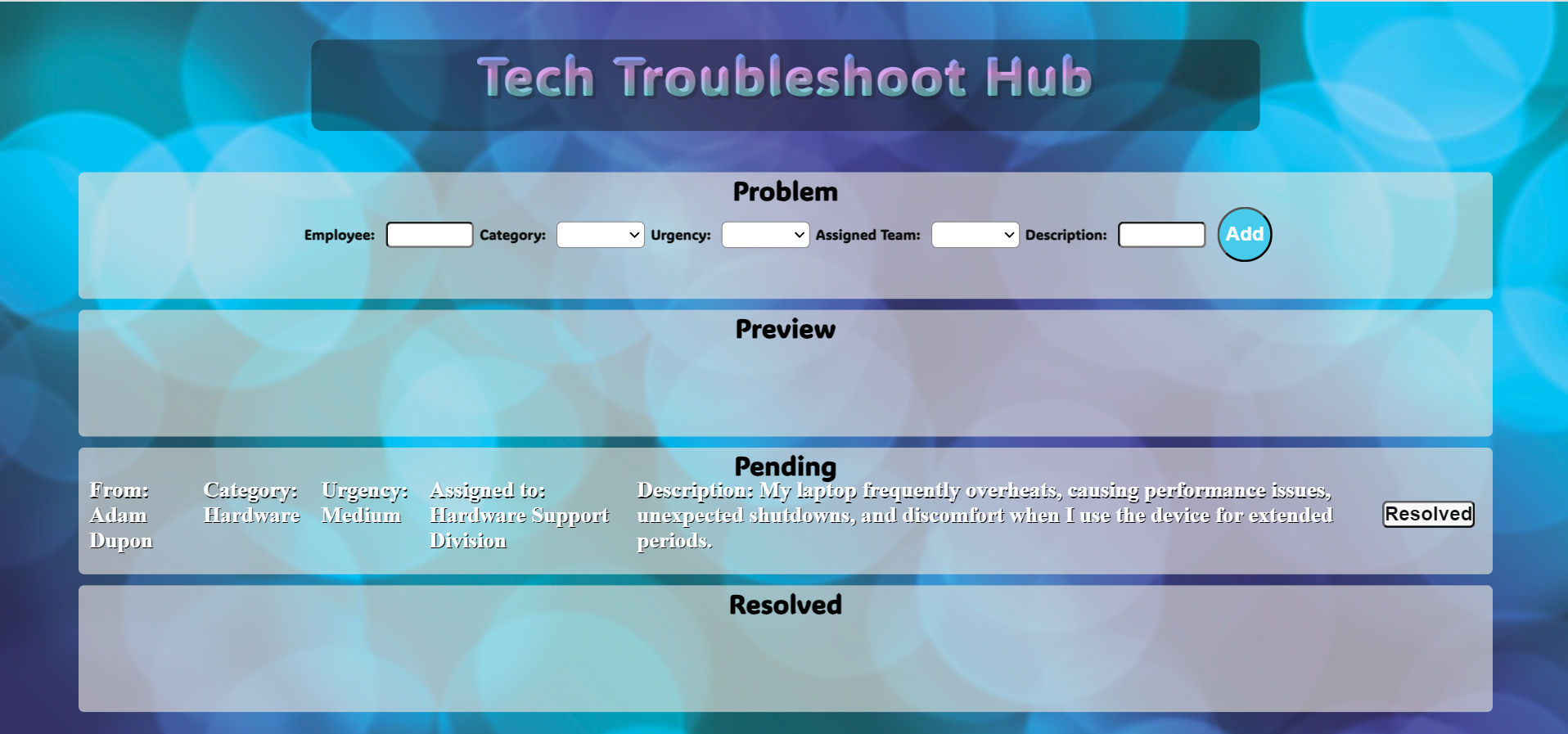
****

* **The list items must be removed from the** "preview-list" and **all of the information must go back to the input fields again**.

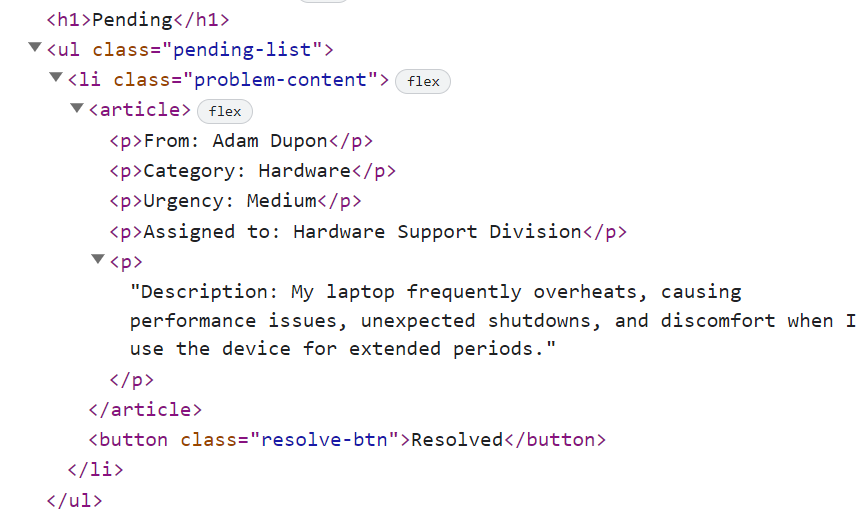
****

1. **Continue to pending**

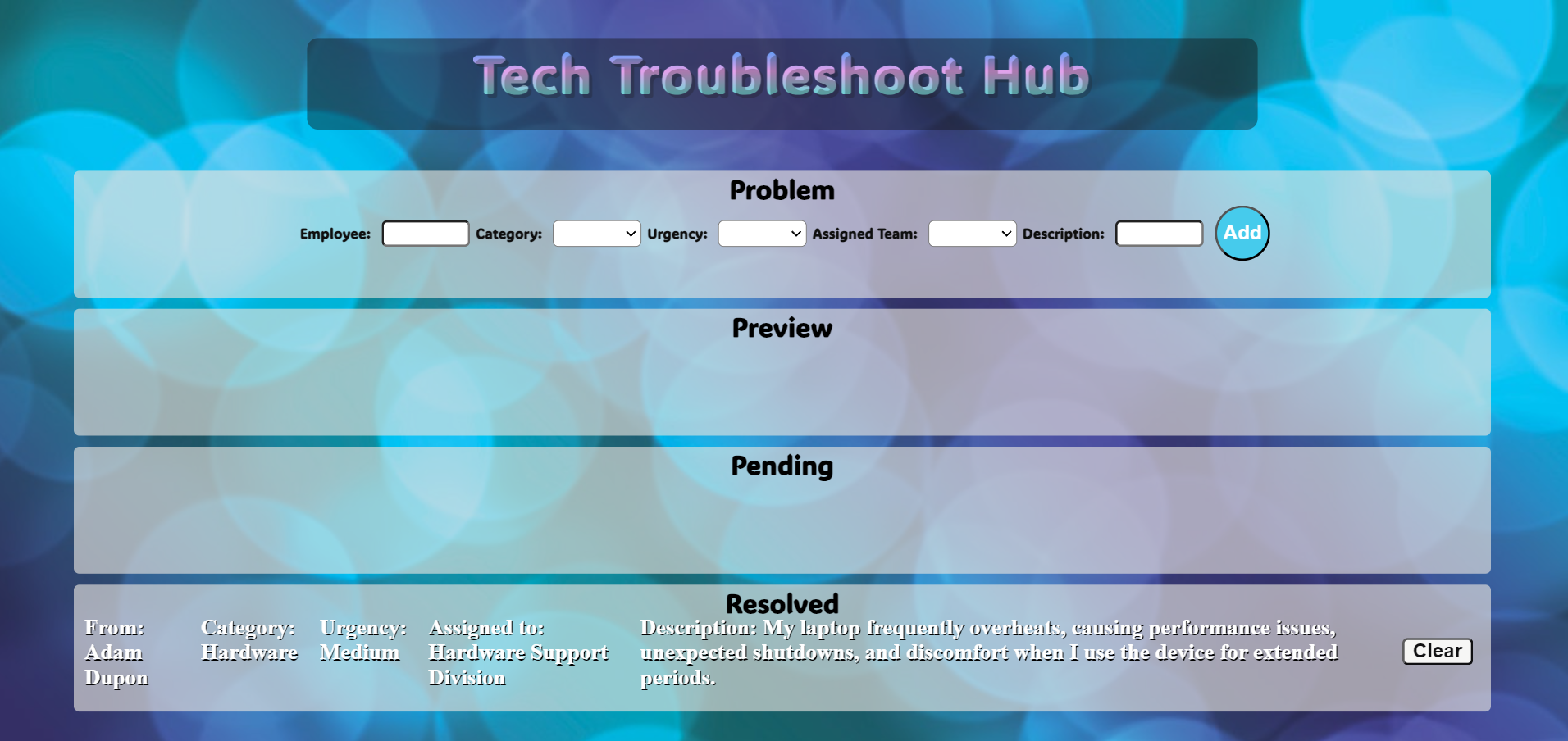
* **When the** "Continue" **button is clicked,** **the information from** "preview-list" unordered list must be transferred to "pending-list" in the same HTML structure**. For you, this means removing everything inside of the ul with class =** "preview-list" **and adding in** "pending-list", **the list item with same information** and the "Resolved" button must be **added.**

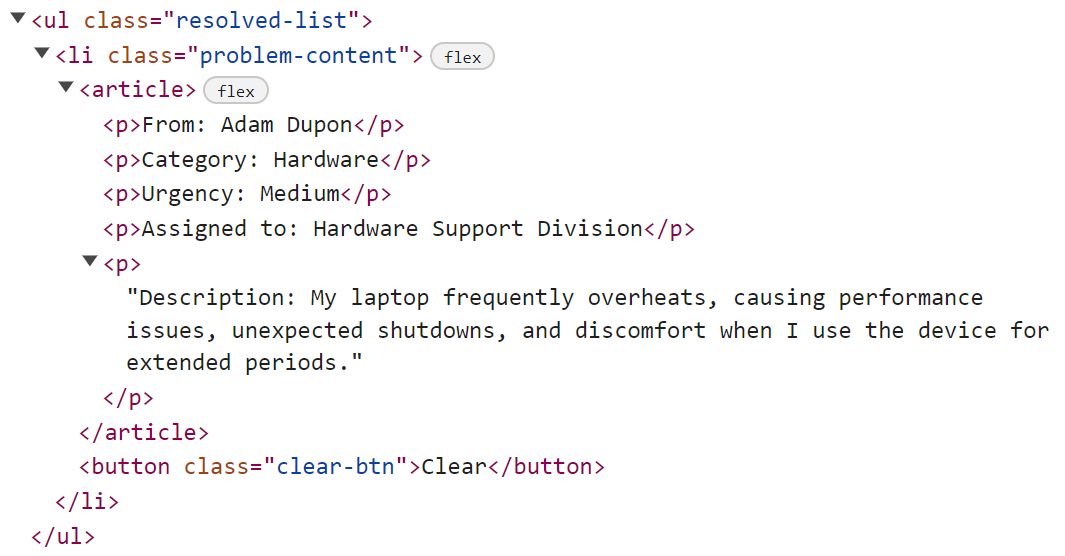
****

* **This is HTML** structure **of** "pending-list" unordered list**:**



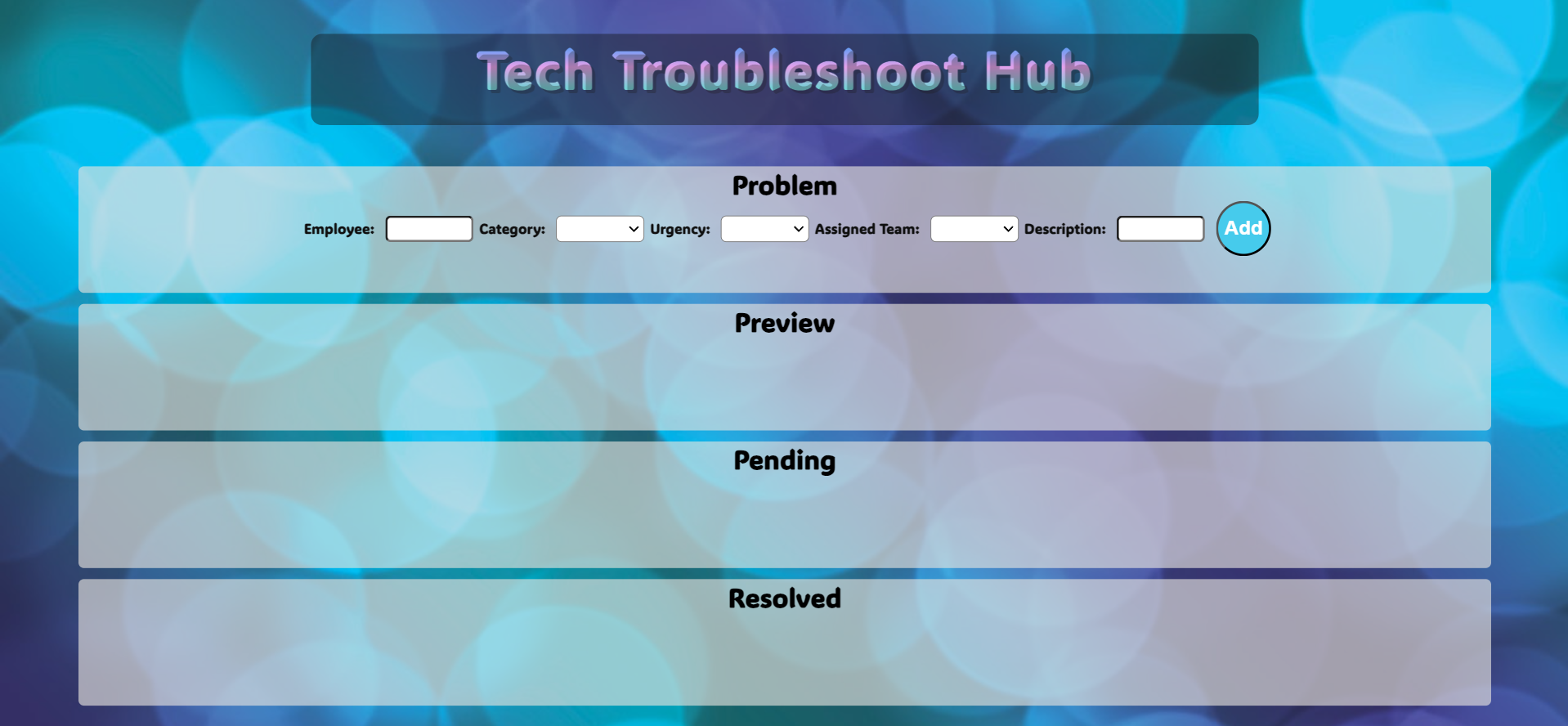
1. **Add to resolved**

* **When the** "Resolved" **button is clicked,** **the information from** "pending-list" unordered list must be transferred to "resolved-list" in the same HTML structure**. For you, this means removing everything inside of the ul with class =** "pending-list" **and adding in** "resolved-list", **the list item with same information** and the "Clear" button must be **added.**
* **This is HTML** structure **of** "resolved-list" unordered list**:**



1. **Clear information**

* **When the** "Clear" **button is clicked,** **the list item must be removed, from the** "resolved-list.
* "Clear" button:

****

## Submission

Submit only yours **solution()** function.

*GOOD LUCK… ☺*