Group Anagrams

You are given an array of strings strs[]. Your task is to group all the strings that are anagrams of each other. An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once. The goal is to return the grouped anagrams as a list of lists, where each sublist contains words that are anagrams of each other.

Input:
An array of strings strs[] consisting of lowercase English letters.

Output:
● A list of lists, where each sublist contains strings that are anagrams of each other. The order of the output groups does not matter.

Examples:
● Example 1
  Input: strs[] = ["eat", "tea", "tan", "ate", "nat", "bat"]
  Output: [["eat", "tea", "ate"], ["tan", "nat"], ["bat"]]
  Explanation:
  i. "eat", "tea", and "ate" are anagrams of each other.
  ii. "tan" and "nat" are anagrams of each other.
  iii. "bat" has no anagram in the array, so it forms its own group.

Constraints:
● 1 ≤ strs.length ≤ 10^4 (The array can contain up to 10,000 strings)
● 0 ≤ strs[i].length ≤ 100 (Each string can have up to 100 characters)
● All strings consist of lowercase English letters.

Test Cases:
1. Input: strs[] = ["eat", "tea", "tan", "ate", "nat", "bat"]
   Output: [["eat", "tea", "ate"], ["tan", "nat"], ["bat"]]
2. Input: strs[] = [""]
   Output: [[""]]
3. Input: strs[] = ["a"]
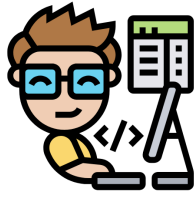
Output: [["a"]]
4. Input: strs[] = ["abc", "bca", "cab", "xyz", "zyx", "yxz"]
   Output: [["abc", "bca", "cab"], ["xyz", "zyx", "yxz"]]
5. Input: strs[] = ["abc", "def", "ghi"]
   Output: [["abc"], ["def"], ["ghi"]]

Edge Cases:
1. Single string: If the array contains only one string, the output should be a list containing that string.
2. Empty strings: If the array contains empty strings, they should be grouped together.
3. All anagrams: If all strings in the array are anagrams of each other, they should form a single group.