East West University
Department of Computer Science and Engineering
Course: CSE479- Web Programming
Summer 2025

Lab Work -03

Objectives:

You will write and execute

- Code that uses the JavaScript String data type
- Code that uses the JavaScript Array data type
- Basic JavaScript instructions

Exercise 1: Identify Empty Entries in Resume Fields

Function Name: isFieldMissing(input)

Write a function that takes a string input from a resume (e.g., candidate.name, candidate.email) and returns:

- "Missing" if the field is blank or only contains spaces.
- "OK" otherwise.

❗ Must trim leading/trailing whitespace before checking.

Test case:

1. **console.log(isFieldMissing("")); // "Missing"**
2. **console.log(isFieldMissing(" ")); // "Missing"**
3. **console.log(isFieldMissing("John Doe")); // "OK"**

Exercise 2: Extract Skill Tags from String

Function Name: extractSkills(skillsString)

The resume contains a string like: "JavaScript, Python, HTML,css , node.js "

Your function should:

- Split the string by comma,
- Trim each item,
- Return the **array of skills in lowercase**.

Test case:

1. **console.log(extractSkills("JavaScript, Python, HTML,css , node.js ")); // ["javascript", "python", "html", "css", "node.js"]**

Exercise 3: Format Candidate Initials

Function Name: getInitials(fullName)

A resume provides the full name in the format: "Sadia K. Rahman"

You must:

- Extract the **first letter** of each part (excluding middle initials like "K."),
- Return it in the format: "S.R."

❗ Ignore any parts that are only one letter or end with a dot.

Test case:

1. ***console.log(getInitials("Sadia K. Rahman")); // "S.R."***
2. ***console.log(getInitials("Hasibul Islam")); // "H.I."***

Exercise 4: Title Case the Resume Headline

Function Name: formatHeadline(headline)

Convert the input string (resume headline) to title case: capitalize the **first letter of each word**, lowercase the rest.

Test case:

1. ***console.log(formatHeadline("full stack DEVELOPER & javascript ENTHUSIAST")); // "Full Stack Developer & Javascript Enthusiast"***

Exercise 5: Top N Preferred Locations

Function Name: preferredLocations(locationsArray, n)
The resume has a list of preferred work locations:
["Dhaka", "Chittagong", "Sylhet", "Rajshahi"]
Return:

- First n locations if n is given.
- If n is not given, return just the first.
- If n is negative, return empty array.

Test case:

1. ***console.log(preferredLocations(["Dhaka", "Sylhet", "Barisal"], 2)); // ["Dhaka", "Sylhet"]***
2. ***console.log(preferredLocations(["Dhaka"], -1)); // []*** []

Exercise 6: List Recently Updated Projects

- **Function Name:** findSkillPair(skills, targetLength)

You are given an array of skill names. Find a pair of consecutive skills such that:

- The total number of **characters** in the two skill names equals targetLength.

Return the indices of that pair.

Test case:

1. ***console.log(findSkillPair(["java", "python", "js", "html"], 10)); // [0, 1] because "java" (4) + "python" (6) = 10***

Exercise 7: Rearrange Skill by Priority

Function Name: rearrangeSkill(skills, fromIndex, toIndex)

Move a skill from one index to another in the resume's skill list.

Test case**:**

1. **console.log(rearrangeSkill(["html", "css", "js", "react"], 0, 2)); // ["css", "js", "html", "react"]**

Exercise 8: Final Challenge - Clean and Analyze Resume Data

Check the following input:

```
const candidate = {
  name: "  Nahid Islam ",
  email: "    ",
  skills: "JavaScript, Python,  HTML,css  ,   node.js ",
  headline: " aspiring FULLSTACK developer  ",
  projects: ["ecommerce", "gamebot", "portfolio", "chatapp"]
};
```

Now, write a function analyzeResume(candidate) that:

1. Cleans name and email fields (trim and check if missing),
2. Converts skills to an array of lowercase skills,
3. Converts headline to title case,
4. Returns a new cleaned object.
