SHREYA GHOLASE
23070521145

Smart Resume Data Extractor

You are asked to build a small JavaScript program that processes raw resume text (copied from a plain .txt file). Since resumes are often messy (extra spaces, inconsistent capitalization, multiple lines, etc.), your program must **clean, validate, and extract key information** using **string functions and regular expressions**.

Input (raw text string example):

```
let resume = `
 Name: jOhN doE
 Email: john.doe@Example.com
 Phone: 9876543210

 Skills: HTML, CSS, JavaScript, Python, SQL

 Extra: Contact me at john.altmail@example.org
 #developer #coder
`;
```

Expected Output (after processing):

```
{
 name: "John Doe", // Trimmed & formatted correctly
 email: "john.doe@example.com", // Validated and lowercased
 phone: "9876543210", // Validated mobile number
 skills: ["HTML", "CSS", "JavaScript", "Python", "SQL"], // Clean split
 hashtags: ["#developer", "#coder"], // Extracted using regex
 emailCount: 2, // Count of email addresses found
 wordCount: 15, // Count words in resume text
 vowelCount: 24 // Count vowels in resume text
}
```

Steps

Students must implement **all the following sub-tasks** in one program:

1. **Name Formatting**
   Extract the name, remove extra spaces (trim() + replace()),
   Capitalize the first letter of each word (charAt() + slice()).
2. **Email Validation & Extraction**
   Use regex to find *all* email addresses.
   Store the first one as the primary email.
   Count how many emails were found.

3. **Mobile Number Validation**
   Validate that the phone number starts with 6,7,8,9 and has exactly 10 digits.
   If invalid, output "Invalid phone number".

4. **Skills Processing**
   Extract skills from the line.
   Split by comma, trim spaces, and return as an array.

5. **Hashtag Extraction**
   Extract all hashtags (e.g., #developer, #coder).

6. **Word Count**
   Count total number of words in the resume.

7. **Vowel Count**
   Count total vowels (a, e, i, o, u).

The following code contains function names + comments only, so you have to fill in the logic yourself (using what you have learned in Practical 6 Part 1 & 2).

// Final Project: Smart Resume Data Extractor

```
let resume = `
 Name: jOhN doE
 Email: john.doe@Example.com
 Phone: 9876543210
```

```
 Skills: HTML, CSS, JavaScript, Python, SQL

 Extra: Contact me at john.altmail@example.org
 #developer #coder
`;
// 1. Format Name
function formatName(rawName) {
 // TODO: Trim spaces, split into words, capitalize each word properly
return formattedName;
}

// 2. Extract and Validate Emails
function extractEmails(text) {
 // TODO: Use regex to find all emails
 // Return array of emails
}

// 3. Validate Mobile Number
function validateMobile(number) {
 // TODO: Check if starts with 6/7/8/9 and has exactly 10 digits
// Return valid number or "Invalid phone number"
}

// 4. Process Skills
function processSkills(rawSkills) {
 // TODO: Split by comma, trim each skill, return array
}

// 5. Extract Hashtags
function extractHashtags(text) {
 // TODO: Use regex to find all hashtags (#...)
 // Return array of hashtags
}

// 6. Word Count
function countWords(text) {
 // TODO: Split by spaces, filter empty items, return count
}
```

```
// 7. Vowel Count
 function countVowels(text) {
  // TODO: Loop through text and count vowels (a, e, i, o, u)
 }
// ------------------------
// Main Program
// ------------------------

 function processResume(resumeText) {
let output = {};

  // Extract Name
  // (Hint: use regex or string search for "Name:")
// output.name = formatName(...);

  // Extract Emails
  // let emails = extractEmails(resumeText);  //
output.email = emails[0]; // Primary email  //
output.emailCount = emails.length;

  // Extract Phone
  // (Hint: find line with "Phone:")
  // output.phone = validateMobile(...);

  // Extract Skills
  // (Hint: find line with "Skills:")
  // output.skills = processSkills(...);

  // Extract Hashtags
  // output.hashtags = extractHashtags(resumeText);

  // Count Words
  // output.wordCount = countWords(resumeText);

  // Count Vowels
  // output.vowelCount = countVowels(resumeText);

  return output;
```

```
}

// Run Program
console.log(processResume(resume));
```

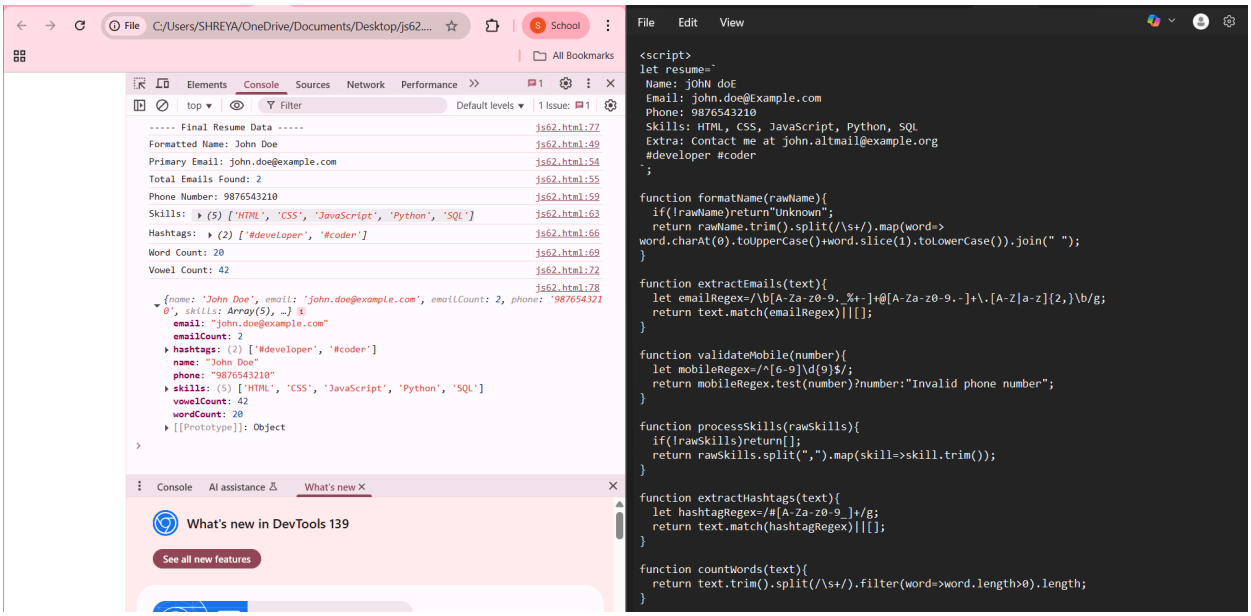
Test Input 1 (Normal, Clean Example)

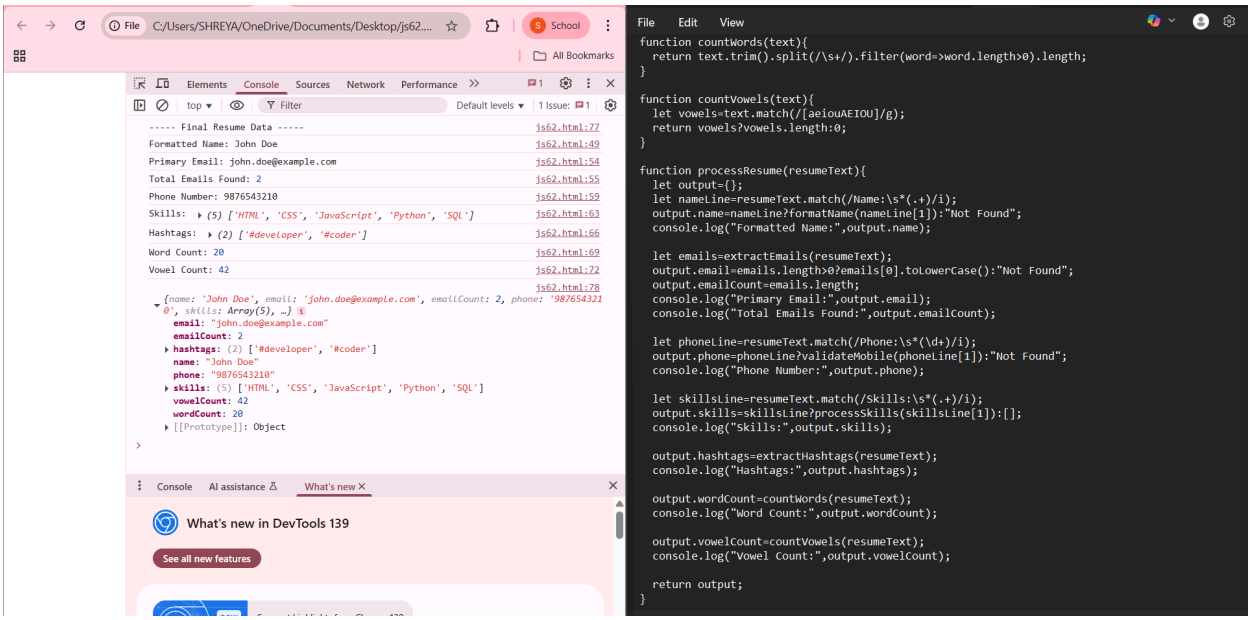
```
let resume1 = `
 Name: jOhN doE
 Email: john.doe@Example.com
 Phone: 9876543210

 Skills: HTML, CSS, JavaScript, Python, SQL

 Extra: Contact me at john.altmail@example.org
 #developer #coder
`;
```

Expected: Properly formatted name, 2 emails, valid phone, 5 skills, 2 hashtags, correct word & vowel count.





Test Input 2 (Messy Spaces & Invalid Phone)

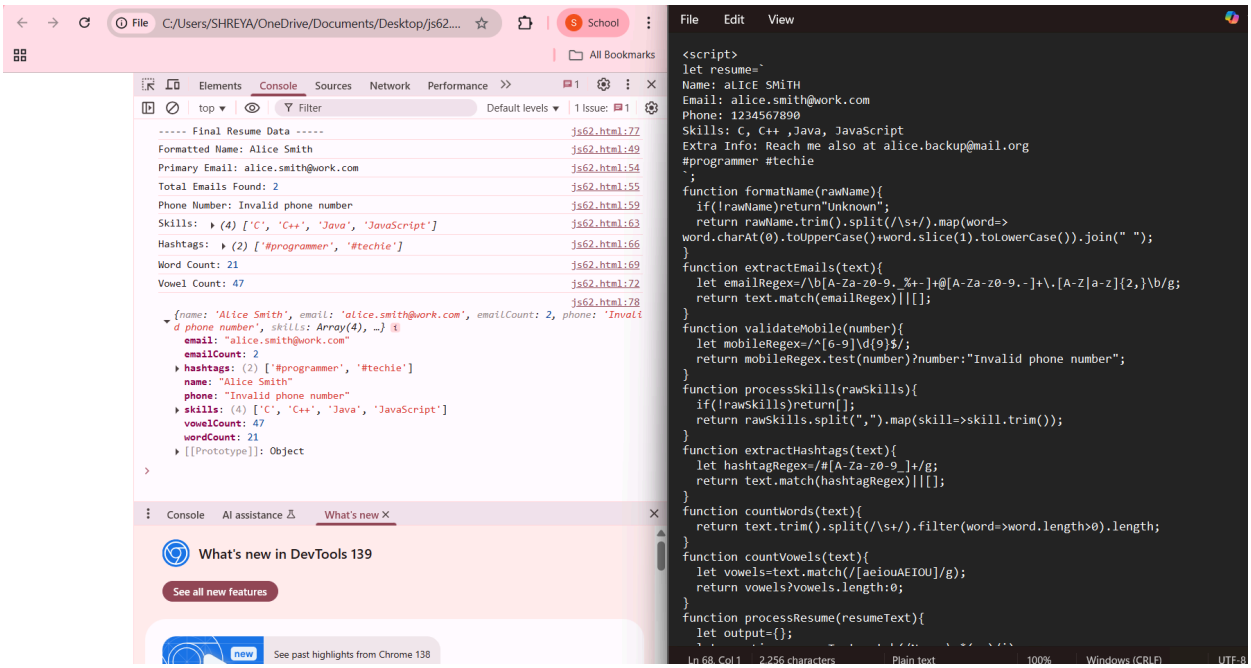
let resume2 = `
Name: aLIcE SMiTH
Email: alice.smith@work.com
Phone: 1234567890

Skills: C, C++ ,Java, JavaScript

Extra Info: Reach me also at alice.backup@mail.org
#programmer #techie
`;

Expected:

- Name → Alice Smith
- 2 emails extracted
- Phone → "Invalid phone number" (since starts with 1)
- Skills → ["C", "C++", "Java", "JavaScript"]
- Hashtags → ["#programmer", "#techie"]

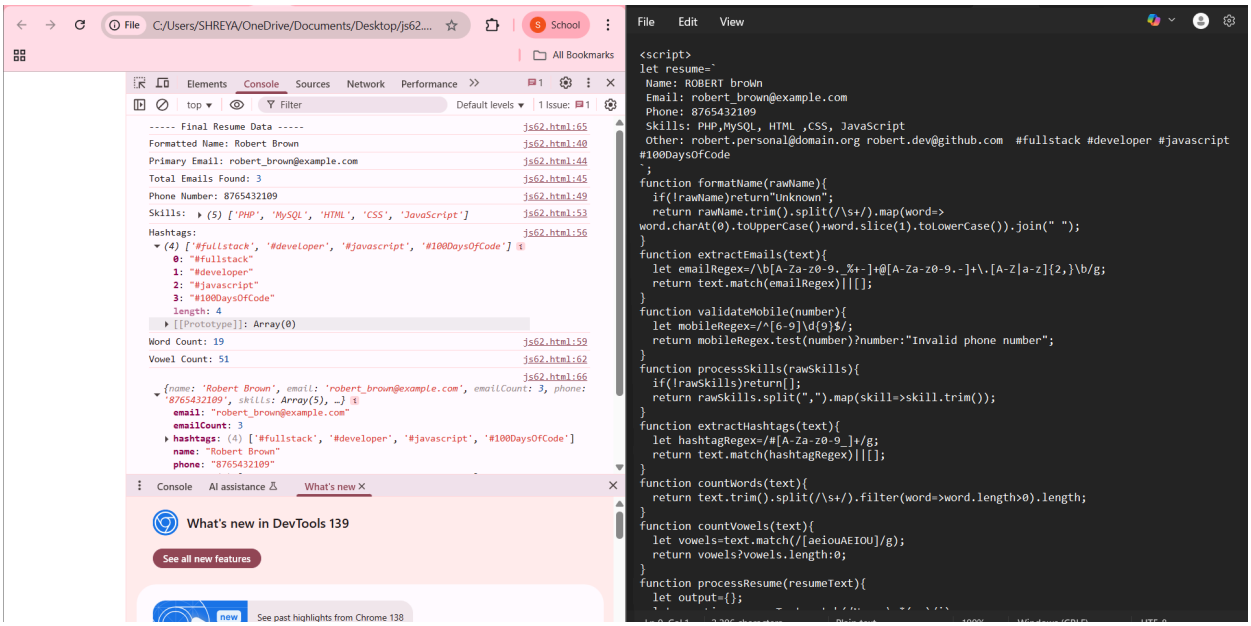


Test Input 3 (Multiple Emails & Extra Hashtags)

let resume3 = `
Name: ROBERT broWn
Email: robert_brown@example.com
Phone: 8765432109
Skills: PHP,MySQL, HTML ,CSS, JavaScript
Other: robert.personal@domain.org robert.dev@github.com
#fullstack #developer #javascript #100DaysOfCode
`;

Expected:

- Name → Robert Brown
- 3 emails extracted
- Valid phone
- Skills → ["PHP", "MySQL", "HTML", "CSS", "JavaScript"]
- 4 hashtags extracted



Test Input 4 (Extra Noise, Mixed Case, Extra Spaces)

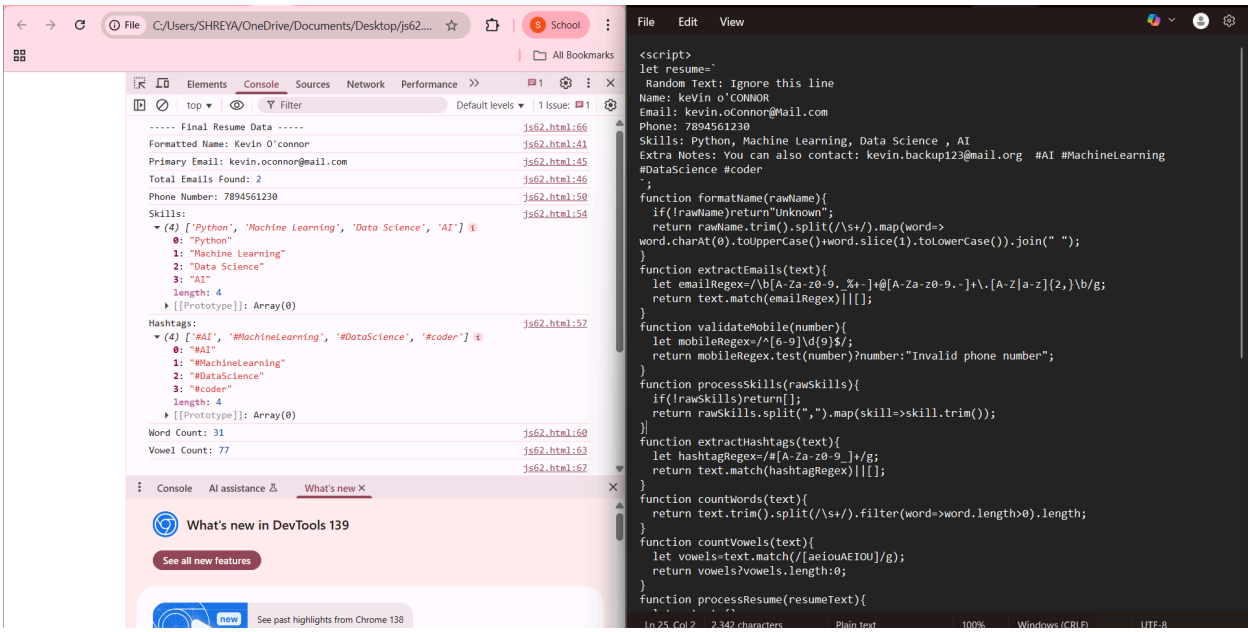
let resume4 = `
Random Text: Ignore this line

Name: keVin o'CONNOR
Email: kevin.oConnor@Mail.com
Phone: 7894561230
Skills: Python, Machine Learning, Data Science , AI

Extra Notes: You can also contact: kevin.backup123@mail.org
#AI #MachineLearning #DataScience #coder
`;

Expected:

- Name → Kevin O'Connor (handle apostrophe properly)
- 2 emails extracted
- Valid phone
- Skills → ["Python", "Machine Learning", "Data Science", "AI"] ● Hashtags → ["#AI", "#MachineLearning", "#DataScience", "#coder"]


File C:/Users/SHREYA/OneDrive/Documents/Desktop/js62....
School
All Bookmarks
Elements Console Sources Network Performance
top Filter Default levels 1 Issue: 1
----- Final Resume Data -----
Formatted Name: Kevin O'connor
Primary Email: kevin.oconnor@mail.com
Total Emails Found: 2
Phone Number: 7894561230
Skills:
(4) ['Python', 'Machine Learning', 'Data Science', 'AI']
0: "Python"
1: "Machine Learning"
2: "Data Science"
3: "AI"
length: 4
[[Prototype]]: Array(0)
Hashtags:
(4) ['#AI', '#MachineLearning', '#DataScience', '#coder']
0: "#AI"
1: "#MachineLearning"
2: "#DataScience"
3: "#coder"
length: 4
[[Prototype]]: Array(0)
Word Count: 31
Vowel Count: 77
js62.html1:66
js62.html1:41
js62.html1:45
js62.html1:46
js62.html1:50
js62.html1:54
js62.html1:57
js62.html1:60
js62.html1:63
js62.html1:67
Console AI assistance What's new
What's new in DevTools 139
See all new features
new
See past highlights from Chrome 138
File Edit View
<script>
let resume=`
Random Text: Ignore this line
Name: keVin o'CONNOR
Email: kevin.oConnor@Mail.com
Phone: 7894561230
Skills: Python, Machine Learning, Data Science , AI
Extra Notes: You can also contact: kevin.backup123@mail.org #AI #MachineLearning
#DataScience #coder
`;
function formatName(rawName){
if(!rawName)return"Unknown";
return rawName.trim().split(/\s+/).map(word=>
word.charAt(0).toUpperCase()+word.slice(1).toLowerCase()).join(" ");
}
function extractEmails(text){
let emailRegex=/\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b/g;
return text.match(emailRegex)||[];
}
function validateMobile(number){
let mobileRegex=/^[6-9]\d{9}$/;
return mobileRegex.test(number)?number:"Invalid phone number";
}
function processSkills(rawSkills){
if(!rawSkills)return[];
return rawSkills.split(",").map(skill=>skill.trim());
}
function extractHashtags(text){
let hashtagRegex=/#[A-Za-z0-9_]+/g;
return text.match(hashtagRegex)||[];
}
function countWords(text){
return text.trim().split(/\s+/).filter(word=>word.length>0).length;
}
function countVowels(text){
let vowels=text.match(/[aeiouAEIOU]/g);
return vowels?vowels.length:0;
}
function processResume(resumeText){
Ln 25, Col 2 2,342 characters Plain text 100% Windows (CRLF) UTF-8