hr | avatar
Illuminating Talent

Test Results and Interview Guide

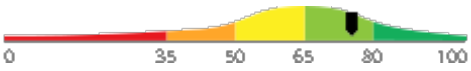
Candidate: **Elizabeth Wantsajob**
Assessment: Cascading Style Sheets (CSS) (Short)
Completed: June 27, 2026
Prepared for: Sara Maple
Example Company

What's Included

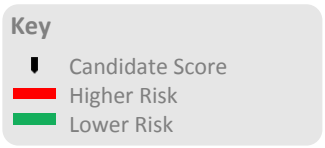
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The Cascading Style Sheets (CSS) (Short) assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes a one page summary, followed by detailed results with an embedded interview guide. Note that these results should always be used as a part of a balanced candidate selection process that includes independent evaluation steps, such as interviews and reference checks.

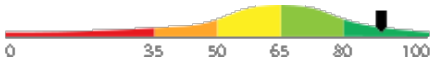
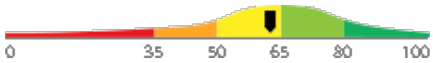
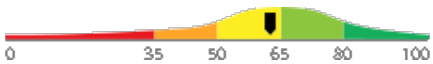
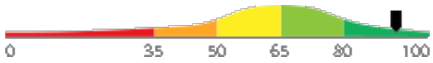
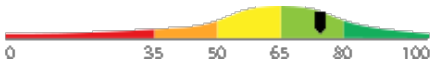
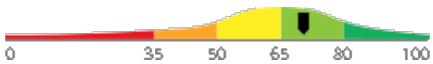
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Cascading Style Sheets (CSS) (Short) June 27, 2026	75	

The candidate demonstrates a solid working knowledge of CSS, including selectors, the box model, layout systems, and responsive design principles, with the ability to apply these skills to practical web application development scenarios. Minor gaps may exist in advanced areas such as cross-browser compatibility, CSS variable management, or resolving complex rendering conflicts, but overall competence is suitable for most entry-level to mid-level responsibilities.



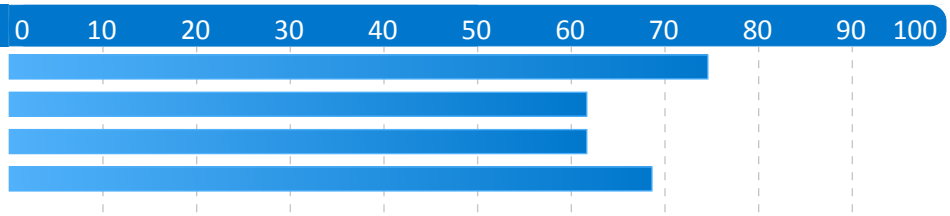
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Box Model, Spacing, and Borders	89	
Box Model, Spacing, and Borders (Coding Tasks)	62	
Selectors, Specificity, and Inheritance (Coding Tasks)	62	
Layout with Flexbox and Grid	92	
Selectors, Specificity, and Inheritance	74	
Typography, Colors, and Visual Formatting	70	

Comparison

Percentile scores indicate how the candidate compares to other test-takers within various groups. The candidate scored equal to or better than the fraction of test-takers indicated by the percentile.

Test-Taker Group	Percentile
Global	75th
North America	62nd
United States	62nd
Example Company	69th



Artificial Intelligence (AI) Generated Scores

This table includes one or more scores derived from a large language model AI query. AI-derived scores are non-deterministic. That is, they are not precisely repeatable. Therefore, these scores should always be treated as supplementary information and should never be used exclusively or compared to hard cutoff values.

Estimated Value	Score	Confidence	Interpretation
Knowledge, Skills, and Abilities Summary	-	-	Summary Points (AI): • (Generic Text for Sample Report) Strong performer in Drag and Drop Files tasks, indicating comfort with file management and basic computer interactions. • Demonstrates solid numerical accuracy in Recognizing and Confirming Numbers, a valuable asset in detail-oriented roles. • Moderate overall performance in Analytical Thinking and Attention to Detail, with adequate grammar skills but room for improvement. • Struggles with Reading and Analyzing Problems, which may limit effectiveness in roles requiring critical reading and complex problem-solving. • Lowest performance in Navigating Between Screens, suggesting difficulty with multi-screen software workflows that could impact productivity in computer-intensive roles. Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies. Elizabeth shows a strong aptitude in Drag and Drop Files, performing well on this technical task and suggesting she is comfortable with this type of computer interaction. This is a notable strength that would translate well into roles requiring file management and basic computer navigation tasks. In the area of Analytical Thinking and Attention to Detail, Elizabeth performs at a moderate level. She demonstrates solid ability in Recognizing and Confirming Numbers, which suggests she is careful and accurate when working with numerical data — a valuable skill in detail-oriented work environments. Her Grammar performance is adequate but leaves room for improvement, indicating she may occasionally make written communication errors. Her weakest area within this competency is Reading and Analyzing Problems, where she struggled to consistently interpret and work through written problem scenarios. This may impact her effectiveness in roles that require critical reading, written comprehension, or complex problem-solving. Elizabeth's most significant area for development is Navigating Between Screens, where she scored considerably lower than the other competencies. This suggests she may have difficulty efficiently moving through software interfaces or multi-screen workflows, which could slow productivity in roles that rely heavily on navigating computer applications or data entry systems. Overall, Elizabeth brings some useful technical strengths, particularly in file management and numerical accuracy, but would benefit from targeted development in software navigation and analytical problem-solving to be fully effective in roles that demand these skills. Computed on: April 2, 2026, 11:09:49PM EDT

Detail

Candidate:	Elizabeth Wantsajob, beth.wantsajob@gmail.com
Assessment:	Cascading Style Sheets (CSS) (Short)
Authorized:	June 27, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 27, 2026, 2:23:48PM EDT
Completed:	June 27, 2026, 2:23:48PM EDT
Overall Score:	75

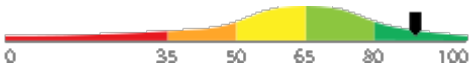
Knowledge and Skills Detail

This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail | Interview Guide

Box Model, Spacing, and Borders

Score: 89



Description:

Covers the CSS box model, which defines how every HTML element is rendered as a rectangular box with content, padding, border, and margin areas. Understanding this model is essential for controlling spacing, sizing, and layout of elements on a page.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate exhibits a comprehensive and advanced understanding of the CSS box model, including the precise control of content areas, padding, borders, and margins to achieve accurate and complex page layouts. They are well-equipped to apply these concepts proficiently across a wide range of design and development scenarios.

What is the difference between box-sizing: content-box and box-sizing: border-box, and why might you prefer one over the other when building a layout?

1	2	3	4	5
Cannot explain the difference or gives an inaccurate description of how either value works.		Explains the difference correctly but cannot clearly articulate a practical reason to prefer one over the other.		Accurately explains both values, notes that border-box includes padding and border in the stated width, and gives a practical layout reason for preferring it.

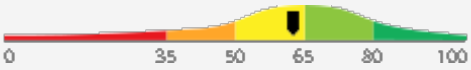
Can you describe what the CSS box model is and name the four layers that make up an element's total space on the page?

1	2	3	4	5
Cannot name or correctly describe the four layers of the box model.		Names most layers but confuses their order or purpose, such as mixing up padding and margin.		Accurately names all four layers (content, padding, border, margin) and correctly describes what each one controls.

Detail

Box Model, Spacing, and Borders (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Interview Guide

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?

1	2	3	4	5
Cannot write correct allocation code; unaware of free() or memory leak risks.		Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.		Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.

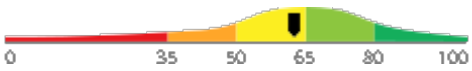
Can you explain what a pointer is in C and describe a situation where you would use one?

1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Detail

Selectors, Specificity, and Inheritance (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Interview Guide

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?

1	2	3	4	5
Cannot write correct allocation code; unaware of free() or memory leak risks.		Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.		Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.

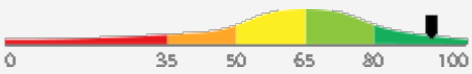
Can you explain what a pointer is in C and describe a situation where you would use one?

1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Detail

Layout with Flexbox and Grid

Score: 92



Description:

Covers the two primary CSS layout systems used to arrange and align elements on a page: Flexbox, which handles one-dimensional layouts along a row or column, and CSS Grid, which handles two-dimensional layouts with rows and columns. Both are widely used in modern web development for building page structure and UI components.

Interpretation:

Candidate should achieve superior job performance in this area with little or no training.

The candidate demonstrates an advanced and comprehensive mastery of CSS Flexbox and Grid layout systems. They are well-equipped to design, implement, and optimize complex one- and two-dimensional layouts for professional web development projects with a high degree of confidence and accuracy.

Interview Guide

When would you choose CSS Grid over Flexbox for a layout, and can you walk through how you would set up a simple two-column, three-row grid using CSS Grid properties?

1	2	3	4	5
Cannot distinguish when to use Grid versus Flexbox or cannot describe how to set up a basic grid.		Gives a reasonable explanation of when to use Grid but only partially describes how to configure a basic grid layout.		Clearly explains Grid is better for two-dimensional layouts, and accurately describes using grid-template-columns, grid-template-rows, and related properties to build the layout.

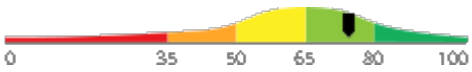
Can you describe what Flexbox is used for and name at least two CSS properties you would use to align items inside a flex container?

1	2	3	4	5
Cannot describe Flexbox's purpose or cannot name any relevant alignment properties.		Gives a general description of Flexbox and names one relevant property but cannot explain how it works.		Clearly describes Flexbox as a one-dimensional layout tool and correctly names and explains properties such as justify-content and align-items.

Detail

Selectors, Specificity, and Inheritance

Score: 74



Description:

Covers how CSS selectors are written to target HTML elements, how specificity rules determine which styles are applied when rules conflict, and how style properties are inherited from parent to child elements. This is foundational knowledge required for writing predictable, maintainable stylesheets.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid and broadly competent understanding of CSS, including layout systems such as flexbox and grid, responsive design through media queries, and the application of typography and spacing for user interface formatting. Minor gaps may exist in advanced topics such as CSS variable management, specificity conflict resolution, or thorough cross-browser style verification.

Interview Guide

If you have an element styled by both a class selector and an ID selector with conflicting property values, which style wins and why? How would you override it without changing the HTML?

1	2	3	4	5
Cannot identify which selector wins or gives an incorrect explanation of why.		Correctly identifies the ID selector wins but struggles to explain how to override without changing HTML.		Correctly identifies the ID wins, explains specificity scoring, and offers a valid override strategy such as using a more specific selector or a targeted rule.

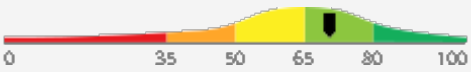
Can you explain what CSS specificity is and give an example of a situation where it might cause a style not to be applied as expected?

1	2	3	4	5
Cannot define specificity or gives an inaccurate explanation with no useful example.		Defines specificity correctly but gives a vague or partially correct example.		Clearly defines specificity, explains the hierarchy (inline > ID > class > element), and gives a concrete, accurate example.

Detail

Typography, Colors, and Visual Formatting

Score: 70



Description:

Covers the CSS properties used to style text, set colors, and control the visual appearance of UI elements. This includes font families, sizes, weights, line heights, text alignment, color formats, and background styles. These properties are used constantly when building and maintaining any web interface.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid and well-rounded understanding of CSS typography, color, and visual formatting, including font weights, line heights, color formats, and background styles. They are likely capable of independently styling and maintaining web interfaces with a high degree of accuracy and consistency.

Interview Guide

What are the differences between using px, em, and rem units for font sizes, and in what situations would you choose one over the others?

1	2	3	4	5
Cannot explain the differences between the units or confuses how they are calculated.		Correctly describes px as fixed and one of em or rem as relative, but cannot fully explain the difference between em and rem or when to use each.		Accurately explains all three units, notes that em is relative to the parent element and rem to the root, and gives practical guidance on when to use each for scalable, accessible design.

What CSS properties would you use to change the font, size, and color of a paragraph of text on a web page?

1	2	3	4	5
Cannot name the relevant properties or names them incorrectly.		Names one or two relevant properties correctly but misses one or uses incorrect syntax.		Correctly identifies font-family, font-size, and color, and can describe valid values for each such as named fonts, px or rem units, and hex or RGB color values.

IT Coding Tasks

During the assessment, the candidate was asked to write one or more programs or scripts. Their responses are included below for review.

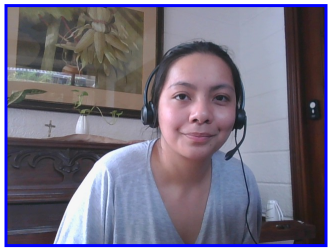
Question or Task	Response

Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries.
 	
The program declares a function called duplicate_array that:
1. Takes a const int pointer to a source array and its length as parameters.
2. Uses calloc to allocate a new int array of the same length.
3. Returns NULL if calloc fails.
4. Copies each element from the source array into the new array using pointer arithmetic (not array subscript notation).
5. Returns the pointer to the newly allocated copy.
 	
In main, the program:
1. Declares and initializes a stack array of 4 integers with values 5, 15, 25, 35.
2. Calls duplicate_array to create a heap-allocated copy.
3. Checks for NULL and prints an error and returns 1 if the call failed.
4. Prints each element of the duplicate using a loop.
5. Frees the duplicate array.
 	
Use appropriate indentation, common C coding conventions, and add brief inline comments where needed. Type your completed source code as your response.
	#include <stdio.h>
	#include <stdlib.h>
	
	int *duplicate_array(const int *src, int length) {
	/* TODO: Use calloc to allocate a new array of 'length' integers, return
	NULL if calloc fails, copy elements from src using pointer arithmetic,
	and return the new pointer. */
	calloc(303);
	}
	
	int main(void) {
	/* TODO: Declare and initialize a stack array of 4 integers: 5, 15, 25, 35,
	then call duplicate_array and store the result. Check for NULL and
	print an error message returning 1 if it failed. */
	array[4]={5,15,25,35};
	
	int i;
	
	/* Print each element of the duplicate */
	for (i = 0; i < 4; i++) {
	printf("duplicate[%d] = %d\n", i, *(duplicate + i));
	}
	
	/* Free the duplicate array */
	free(duplicate);
	return 0;
	}
	
	Comments (AI): The code segment has several syntax errors and incomplete implementation. The duplicate_array function does not correctly allocate memory or copy elements. The main function has syntax errors and does not properly call the duplicate_array function. However, the structure and intent of the code are somewhat clear, and the code attempts to follow the requirements.

Identity Confirmation Photos

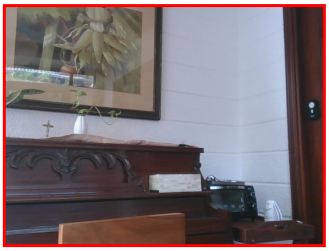
The following photos of the candidate and any identification were uploaded during the assessment session.

Photo Analysis Results	
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)

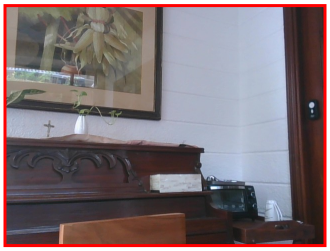


Pre/Post-Test Photo

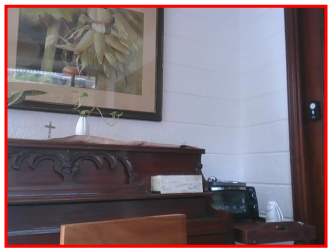
ID Photo



In-Test Error Detected (No Face Detected)

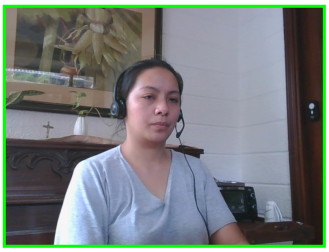


In-Test Error Detected (No Face Detected)

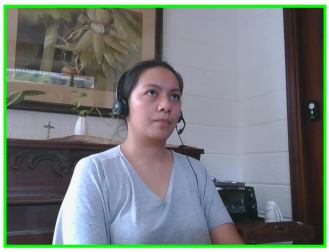


In-Test Error Detected (No Face Detected)

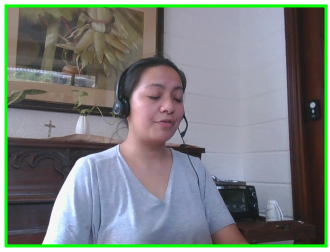
In-Test Photo



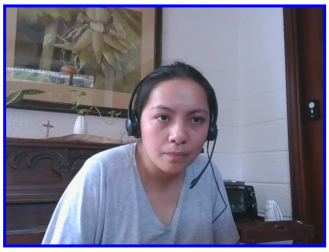
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Resume or CV

Summary Updated on

Motivated career professional with extensive experience in office administration and management. Proven track record of improving efficiency, reducing costs, and enhancing office operations through strategic initiatives and technology implementation.

Objective

I am seeking a role where I can use my many skills and my exceptional judgment and empathy for customers to make a difference to a growing company.

Education

- Associate of Applied Science in Office Administration, Portland Community College, 2020

Experience

- General Office Clerk, Paramount Office Management, 09/2023 – Present
- Administrative Assistant, Global Enterprises Inc., 04/2021 – 08/2023
- Administrative Assistant, Innovative Business Solutions Ltd., 07/2019 – 03/2021

Other Qualifications

- Microsoft Office Specialist (MOS) Certification
- Certified Administrative Professional (CAP)
- International Association of Administrative Professionals (IAAP) Certification

Report Preparation Notes

- Hiring decisions should never be based on a single source of information. The most effective use of this assessment report is as a part of a multi-faceted program of candidate evaluation that includes resume review, interviews, and reference checks.
- Overall vs Percentiles Scores: The overall score reflects the success in the test, based on the mean (average) and standard deviation of the test scores. The percentile score reflects the percentage of test-takers who scored equal or below this overall score. We recommend you use the Overall Score as your primary evaluation criteria. However, percentile scores can often be useful in comparing specific candidates against one another and with a group, such as for test takers in a certain organization or within a certain account.
- Note that comparison information is calculated based on completed instances of this assessment at that time the assessment is scored. As additional instances are completed, the comparative data may change. You can always update a report to the current values by clicking on 'Recalculate Percentiles' within the online results viewing pages at www.hravatar.com.
- Most competency scores are norm-based, which means that they can be interpreted in terms of their distance from the average or mean score. For all scales, a score equal to the mean receives a score of 65 and scores above and below this value are set so that a score change of 15 equals one standard deviation.
- For linear competencies, higher is better across the entire scale. For these scales a score between 65 and 80 (light green) represents 0 to 1 standard deviation above the mean and a score above 80 (dark green) represents more than one standard deviation above the mean. Similarly, a score of 50 - 65 (yellow) represents 0 to 1 standard deviation below the mean, while a score of 35 - 50 (orange) equates to 1 to 2 standard deviations below the mean, and a score below 35 represents more than 2 standard deviations below the mean.
- Sim ID: 20794-1, Key: 0-0, Rpt: 68, Prd: 9616, Created: 2026-06-27 14:23 EDT
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Score Calculation Detail

The following table provides a summary of how the overall score was calculated from each of the individual competency scores. First, all competency scores are calculated on a scale of 0-100. Note that some competencies use their color category rather than their actual numeric score in the overall calculation. For these, a standard score associated with the assigned color category is used in the overall score calculation rather than the actual numeric score. This is reflected in the ''Score Value Used'' column. Next, a weighted average of scores is computed using individual competency weights, typically set using job analysis data provided by the US Government Occupational Information Network (O*Net).

Competency	Score	How applied to overall	Score Value Used	Weight (%)
Box Model, Spacing, and Borders	89.2719	Numeric Score	89.2719	16.6667
Box Model, Spacing, and Borders (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Layout with Flexbox and Grid	92.9379	Numeric Score	92.9379	16.6667
Selectors, Specificity, and Inheritance	74.6626	Numeric Score	74.6626	16.6667
Selectors, Specificity, and Inheritance (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Typography, Colors, and Visual Formatting	70.8423	Numeric Score	70.8423	16.6667
Weighted Average:				75.6119
Final Overall Score:				75

Notes

(This area is intentionally blank - it's reserved as space for your notes.)