hr | avatar
Illuminating Talent

Test Results and Interview Guide

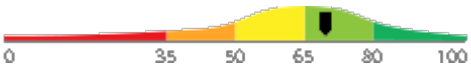
Candidate: **Elizabeth Wantsajob**
Assessment: HTML Programming (Short)
Completed: June 27, 2026
Prepared for: Sara Maple
Example Company

What's Included

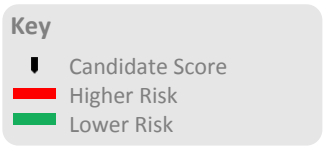
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The HTML Programming (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.

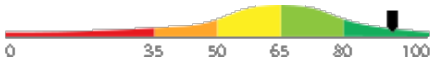
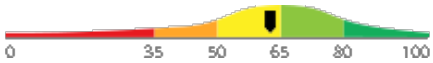
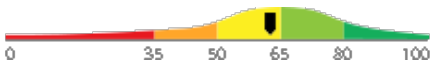
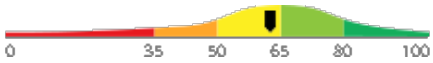
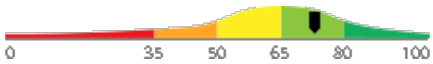
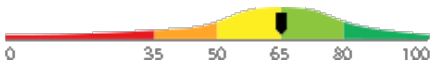
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com HTML Programming (Short) June 27, 2026	69	

The candidate exhibits a solid and competent grasp of modern HTML web programming, including document configuration, content layout, hyperlinking, media elements, and form construction using standard specifications. The candidate is likely capable of independently writing and maintaining business webpages, though occasional gaps may exist in advanced areas such as accessibility compliance, external resource referencing, or markup validation techniques.



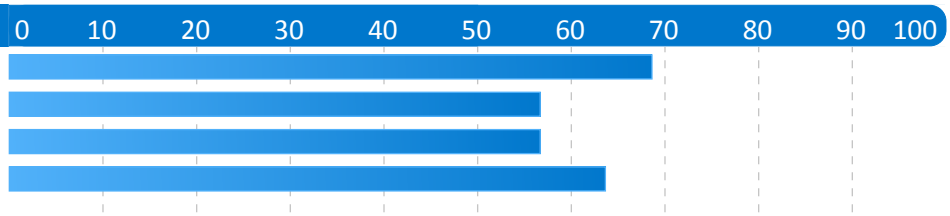
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Content Layout and Structural Elements	91	
Content Layout and Structural Elements (Coding Tasks)	62	
HTML Document Structure and Head Configuration (Coding Tasks)	62	
Hyperlinks, Images, and Media Elements (Coding Tasks)	62	
HTML Document Structure and Head Configuration	73	
Hyperlinks, Images, and Media Elements	65	

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	69th
North America	57th
United States	57th
Example Company	64th



Detail

Candidate:	Elizabeth Wantsajob, beth.wantsajob@gmail.com
Assessment:	HTML Programming (Short)
Authorized:	June 27, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 27, 2026, 2:23:32PM EDT
Completed:	June 27, 2026, 2:23:32PM EDT
Overall Score:	69

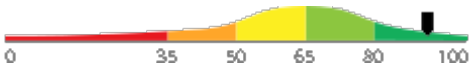
Knowledge and Skills Detail

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

Content Layout and Structural Elements

Score: 91



Description:

Covers the use of HTML elements to organize and display page content, including headings, paragraphs, lists, and structural grouping elements such as header, nav, main, section, article, aside, and footer. Focuses on using the right element for the right purpose to create clear, well-organized page layouts.

Interpretation:

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

The candidate exhibits a strong and comprehensive command of HTML content layout and structural elements. They consistently demonstrate the ability to select and apply the correct semantic elements — including headings, lists, and structural grouping elements — to produce clear, well-organized, and purposeful page layouts.

How would you mark up a webpage that has a site navigation menu, a main content area with multiple articles, and a sidebar with related links?

1	2	3	4	5
Uses only divs or cannot identify appropriate structural elements for each section.		Identifies most correct elements but misplaces one or two, such as using section instead of nav.		Correctly uses nav, main, article, and aside and explains the purpose of each choice.

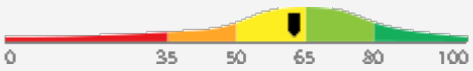
What is the difference between a div element and a semantic element like section or article, and when would you use one over the other?

1	2	3	4	5
Cannot distinguish between div and semantic elements or uses them interchangeably without reason.		Knows semantic elements carry meaning but struggles to explain when to choose one over a div.		Explains that semantic elements improve readability, accessibility, and SEO, and gives clear use-case examples for each.

Detail

Content Layout and Structural Elements (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.

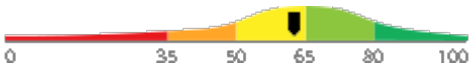
Can you explain what a pointer is in C and describe a situation where you would use one?

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

HTML Document Structure and Head Configuration (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.

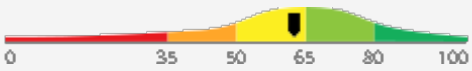
Can you explain what a pointer is in C and describe a situation where you would use one?

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

Hyperlinks, Images, and Media Elements (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.

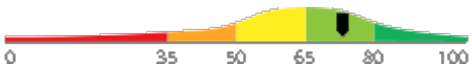
Can you explain what a pointer is in C and describe a situation where you would use one?

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

HTML Document Structure and Head Configuration

Score: 73



Description:

Covers the foundational building blocks of an HTML document, including the doctype declaration, the root html element, and the head section. Includes configuring metadata, character encoding, viewport settings, and linking external stylesheets and scripts.

Interpretation:

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

The candidate exhibits a solid and competent understanding of HTML programming, including proficiency with document structures, content layout, hyperlinks, media elements, and form creation using standard modern specifications. The individual is likely capable of independently writing and maintaining business webpages with only occasional need for reference or support on more specialized topics such as validation compliance or advanced accessibility integration.

Interview Guide

Why is the viewport meta tag important, and what happens to a webpage on a mobile device if it is left out?

1	2	3	4	5
Cannot explain the viewport tag or its effect on mobile rendering.		Knows the viewport tag affects mobile display but cannot explain specific consequences of omitting it.		Explains that without it, mobile browsers render at desktop width, causing unintended zooming and layout issues.

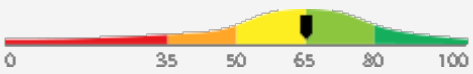
Can you walk me through the basic structure of an HTML document from top to bottom, and explain what goes inside the head element?

1	2	3	4	5
Cannot name basic elements or confuses head and body sections.		Names doctype, html, head, and body but gives vague explanations of head contents.		Clearly explains doctype, meta charset, viewport, title, and linking stylesheets and scripts.

Detail

Hyperlinks, Images, and Media Elements

Score: 65



Description:

Covers how to create hyperlinks using the anchor element to connect pages and external resources, and how to embed images using the img element with proper source paths and alternative text. Includes basic use of attributes that control link behavior and image display.

Interpretation:

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

The candidate exhibits a solid working knowledge of HTML hyperlinks, images, and related media elements. They can competently use anchor and img elements with appropriate attributes, including source paths, alternative text, and link behavior controls, with only minor gaps in proficiency.

Interview Guide

What is the difference between an absolute URL and a relative URL in an href or src attribute, and when would you use each one?

1	2	3	4	5
Cannot define or distinguish between absolute and relative URLs.		Can define both types but gives an incomplete or unclear explanation of when to use each.		Clearly explains that absolute URLs reference external resources and relative URLs reference internal files, with practical examples.

How do you create a link that opens in a new browser tab, and how do you add an image to a webpage with a text description for users who cannot see it?

1	2	3	4	5
Cannot recall the target attribute or the alt attribute, or confuses their syntax.		Knows how to write a basic link and image tag but forgets the target blank or alt attribute details.		Correctly writes both elements with target blank for the link and a descriptive alt attribute for the image.

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.

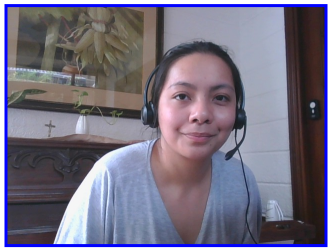
Question or Task	Response

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

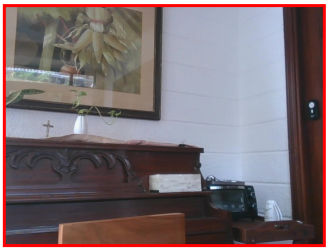
The following photos of the candidate and any identification were uploaded during the assessment session.

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%)

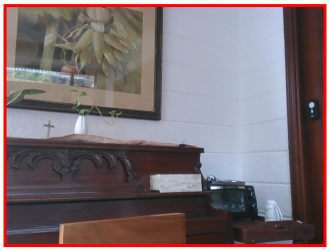


Pre/Post-Test Photo

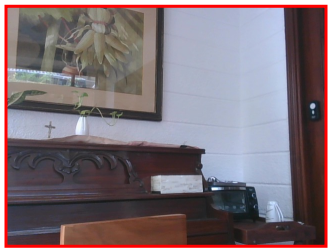
ID Photo



In-Test Error Detected (No Face Detected)

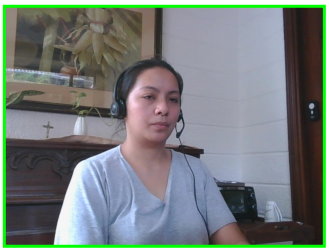


In-Test Error Detected (No Face Detected)

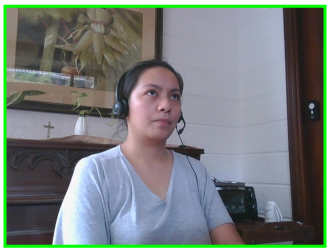


In-Test Error Detected (No Face Detected)

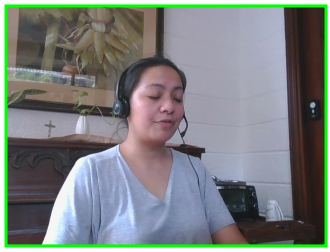
In-Test Photo



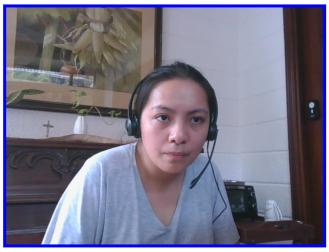
In-Test Photo



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: 20791-1, Key: 0-0, Rpt: 68, Prd: 9613, 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 (%)
Content Layout and Structural Elements	91.7307	Numeric Score	91.7307	16.6667
Content Layout and Structural Elements (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
HTML Document Structure and Head Configuration	73.4373	Numeric Score	73.4373	16.6667
HTML Document Structure and Head Configuration (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Hyperlinks, Images, and Media Elements	65.4700	Numeric Score	65.4700	16.6667
Hyperlinks, Images, and Media Elements (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Weighted Average:				69.9289
Final Overall Score:				69

Notes

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