hr | avatar
Illuminating Talent

Test Results and Interview Guide

Candidate: **Elizabeth Wantsajob**
Assessment: Markdown
Completed: November 4, 2026
Prepared for: Sara Maple
Example Company

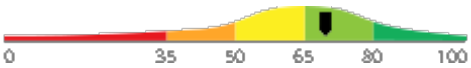
What's Included

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

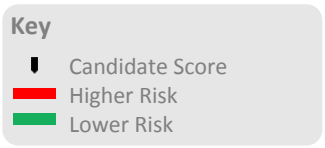
Important Note: The Markdown assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes 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.

Proprietary and Confidential

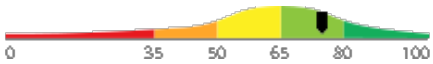
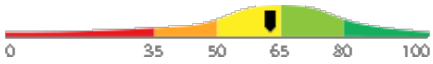
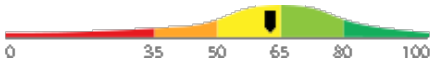
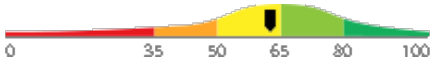
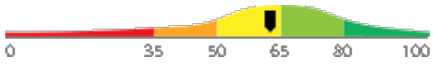
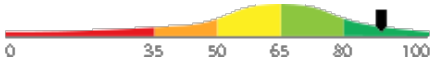
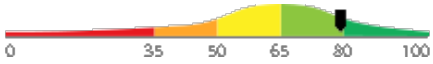
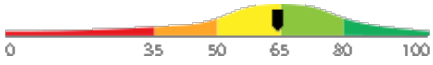
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Markdown November 4, 2026	70	

The candidate exhibits solid proficiency across a broad range of Markdown concepts, including document structure, code blocks, tables, blockquotes, and debugging malformed formatting. They are likely capable of independently writing and maintaining well-organized Markdown files, though they may have occasional gaps in areas such as platform-specific rendering nuances, comments, or integration with documentation systems and static site generators.



Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Basic Syntax and Text Formatting	75	
Basic Syntax and Text Formatting (Coding Tasks)	62	
Code Blocks, Blockquotes, and Special Syntax (Coding Tasks)	62	
Links, Images, Lists, and Tables (Coding Tasks)	62	
Reading, Debugging, and Maintaining Markdown Files (Coding Tasks)	62	
Code Blocks, Blockquotes, and Special Syntax	89	
Links, Images, Lists, and Tables	79	
Reading, Debugging, and Maintaining Markdown Files	64	

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	0 10 20 30 40 50 60 70 80 90 100
Global	70th	
North America	58th	
United States	58th	
Example Company	64th	

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

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

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Candidate:	Elizabeth Wantsajob, beth.wantsajob@gmail.com
Assessment:	Markdown
Authorized:	November 4, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
IP Address:	96.241.65.10
IP Country:	United States
IP State:	Virginia
IP City:	Gainesville
Started:	November 4, 2026, 4:39:59PM EST
Completed:	November 4, 2026, 4:39:59PM EST
Overall Score:	70

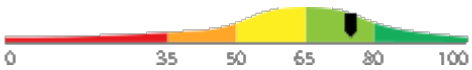
Knowledge and Skills Detail

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

Basic Syntax and Text Formatting

Score: 75



Description:

Covers the core building blocks of Markdown, including how to create headers, paragraphs, line breaks, and formatted text such as bold, italics, and strikethrough. This is the foundation of writing any Markdown document and is applied in virtually every file.

Interpretation:

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

The candidate demonstrates a solid and broadly competent understanding of Markdown, including the ability to write, debug, and maintain files across a wide range of syntax and structural concepts. They are likely capable of working independently on most Markdown tasks, with only occasional difficulty in specialized areas such as platform rendering nuances or integration with documentation tooling.

Walk me through how you would structure a Markdown document that includes a title, a few section headers, a short paragraph, and some bold and italic text. What does that look like in raw Markdown?

1	2	3	4	5
Recalls some syntax but makes errors or cannot explain the difference between header levels or inline formatting marks.		Correctly describes most syntax with minor gaps, such as confusing single vs. double asterisks for bold vs. italic.		Fluently and accurately describes all syntax, explains why certain choices are made, and anticipates common mistakes.

Can you explain what the pound sign (#) does at the start of a line in Markdown, and what happens if you use more than one?

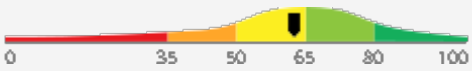
1	2	3	4	5
Cannot explain the purpose of # or gives an incorrect explanation with no self-correction.		Correctly identifies # as a header marker and understands that more symbols mean smaller headers, but cannot elaborate further.		Accurately explains header levels 1–6, mentions spacing requirements, and notes how headers affect document structure.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Basic Syntax and Text Formatting (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?

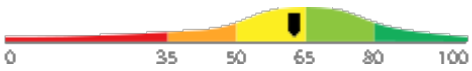
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Code Blocks, Blockquotes, and Special Syntax (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?

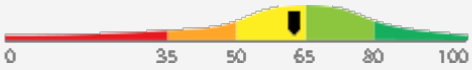
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Links, Images, Lists, and Tables (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?

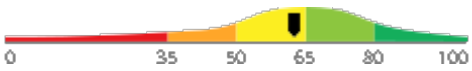
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Reading, Debugging, and Maintaining Markdown Files (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?

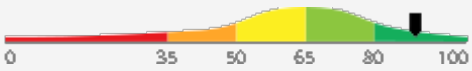
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Code Blocks, Blockquotes, and Special Syntax

Score: 89



Description:

Covers how to display inline code, fenced code blocks with language tags, blockquotes, horizontal rules, and how to escape characters that have special meaning in Markdown. These features are especially important for technical documentation and developer-facing content.

Interpretation:

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

The candidate demonstrates a strong and comprehensive command of Markdown's code blocks, blockquotes, and special syntax features, including precise use of language-tagged fenced blocks, horizontal rules, and character escaping. This individual is highly capable of authoring accurate, professional technical and developer-facing Markdown documentation with minimal oversight.

Interview Guide

You are writing technical documentation and need to show a multi-line code sample in Python, a short inline reference to a function name, and a note set apart as a blockquote. How would you write each of those in Markdown?

1	2	3	4	5
Can describe one element but confuses inline code with fenced blocks or does not know how to specify a language for syntax highlighting.		Correctly describes most elements but may omit the language tag on the fenced code block or make minor syntax errors.		Accurately and confidently writes all three elements, explains backtick usage, language identifiers, and the > syntax for blockquotes.

If you wanted to display a backtick character as plain text in a Markdown file without it being interpreted as code formatting, how would you do that?

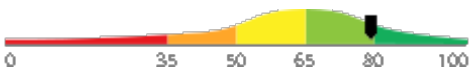
1	2	3	4	5
Does not know how to escape the backtick or suggests an approach that would not work.		Knows that escaping is possible but is unsure of the exact method, such as using a backslash or double backticks.		Accurately explains using a backslash to escape special characters and describes the double-backtick workaround for backticks inside inline code.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Links, Images, Lists, and Tables

Score: 79



Description:

Covers how to create ordered and unordered lists, nest list items, insert hyperlinks and images, and build tables with rows and columns. These elements are used constantly in documentation, README files, and technical writing.

Interpretation:

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

The candidate demonstrates a solid working knowledge of Markdown links, images, lists, and tables, and can reliably apply these elements across documentation, README files, and technical writing tasks. Minor gaps may exist in edge cases or advanced formatting scenarios, but overall competence is evident.

Interview Guide

Imagine you are writing a README file and need to include a table comparing three tools, a numbered list of setup steps, and a link to an external resource. How would you write each of those in Markdown?

1	2	3	4	5
Can describe one or two elements but struggles with table syntax or nesting, and makes notable errors.		Accurately describes most elements with minor syntax errors, such as missing the header separator row in a table.		Correctly and confidently writes all three elements, explains the pipe and dash syntax for tables, and notes alignment options.

How do you create a simple bullet point list in Markdown, and how would you add an indented sub-item under one of the bullets?

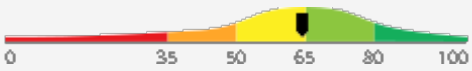
1	2	3	4	5
Cannot produce correct list syntax or does not know how to create a nested item.		Correctly creates a basic list but is unsure about the exact indentation needed for nesting.		Accurately writes list syntax, explains multiple valid markers (-, *, +), and clearly describes indentation rules for nesting.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Detail

Reading, Debugging, and Maintaining Markdown Files

Score: 64



Description:

Covers the ability to read raw Markdown source without a renderer, identify and fix formatting errors, and understand how the same file may render differently across platforms such as GitHub and standard editors. Also includes organizing documents for readability and knowing how to use Markdown comments.

Interpretation:

Candidate appears capable of average job performance in this area with little or no training.

The candidate has a foundational working knowledge of reading and debugging Markdown source files, though gaps remain in handling more complex formatting issues or cross-platform rendering considerations. They demonstrate a basic familiarity with document organization and Markdown comments but may require guidance in nuanced scenarios.

Interview Guide

You open a Markdown file and notice that a section of text that was meant to appear as a formatted table is showing up as plain text with pipes and dashes. How would you go about diagnosing and fixing the problem?

1	2	3	4	5
Offers a vague answer without identifying likely causes such as a missing header separator row or inconsistent spacing.		Identifies one or two likely causes and describes a reasonable fix, but may not mention platform rendering differences.		Systematically identifies common table formatting errors, explains how to verify the fix, and mentions that rendering behavior may vary by platform.

Have you ever had Markdown formatting that looked correct to you but did not render the way you expected? What did you do to figure out what went wrong?

1	2	3	4	5
Cannot describe a debugging process or has not encountered rendering issues and offers no hypothetical approach.		Describes a basic process such as checking syntax or previewing in a different tool, but without much depth.		Describes a clear, methodical approach including checking raw source, testing in multiple renderers, and consulting platform-specific documentation.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

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.

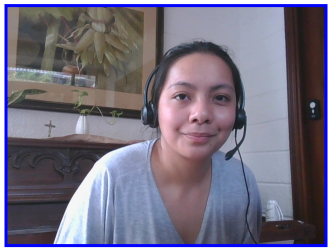
Question or Task	Response
Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries. The program declares a function called duplicate_array that: 1. Takes a const int pointer to a source array and its 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.

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Identity Confirmation Photos

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

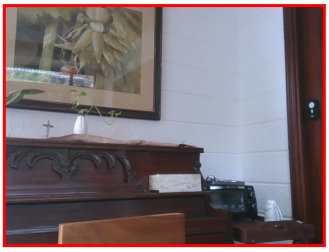
Photo Analysis Results	
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)



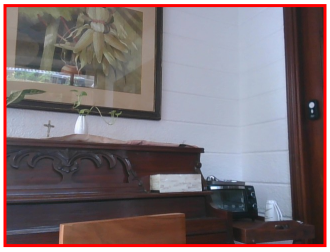
Pre/Post-Test Photo



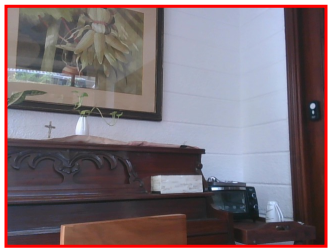
ID Photo



In-Test Error Detected (No Face Detected)



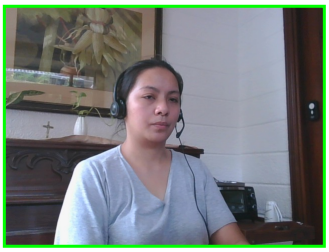
In-Test Error Detected (No Face Detected)



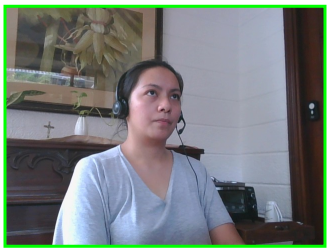
In-Test Error Detected (No Face Detected)



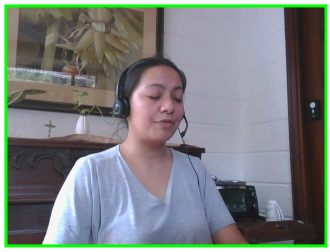
In-Test Photo



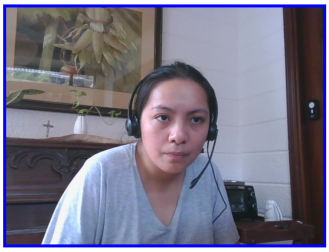
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

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

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

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: 21151-1, Key: 0-0, Rpt: 68, Prd: 10076, Created: 2026-11-04 16:40 EST
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

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 (%)
Basic Syntax and Text Formatting	75.1793	Numeric Score	75.1793	12.5000
Basic Syntax and Text Formatting (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Code Blocks, Blockquotes, and Special Syntax	89.0603	Numeric Score	89.0603	12.5000
Code Blocks, Blockquotes, and Special Syntax (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Links, Images, Lists, and Tables	79.7624	Numeric Score	79.7624	12.5000
Links, Images, Lists, and Tables (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Reading, Debugging, and Maintaining Markdown Files	64.9072	Numeric Score	64.9072	12.5000
Reading, Debugging, and Maintaining Markdown Files (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Weighted Average:				70.1028
Final Overall Score:				70

Copyright 2026, HR Avatar, Inc. All Rights Reserved.

Notes

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

Copyright 2026, HR Avatar, Inc. All Rights Reserved.