hr avatar
Illuminating Talent

Test Results and Interview Guide

Candidate:	**Elizabeth Wantsajob**
Assessment:	Angular Programming (Short)
Completed:	June 28, 2026
Prepared for:	Sara Maple Example Company

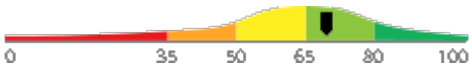
What's Included

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

Important Note: The Angular 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.

Proprietary and Confidential

Overall

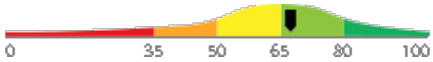
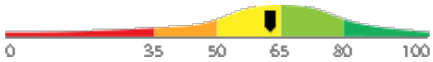
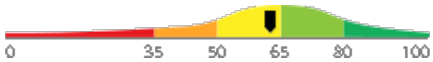
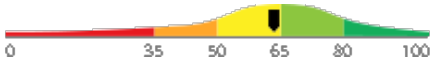
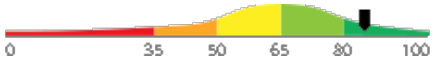
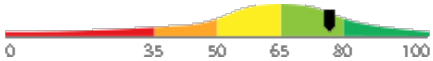
Candidate	Score	Interpretation
Elizabeth Wantsajob	69	

beth.wantsajob@gmail.com
Angular Programming (Short)
June 28, 2026

The candidate demonstrates a solid working knowledge of Angular, including components, templates, directives, services, dependency injection, routing, and forms, and is capable of writing and maintaining Angular applications with limited supervision. Some gaps may exist in more advanced areas such as complex RxJS usage, lifecycle hook optimization, or nuanced HTTP client handling.

Key
- Candidate Score
- Higher Risk
- Lower Risk

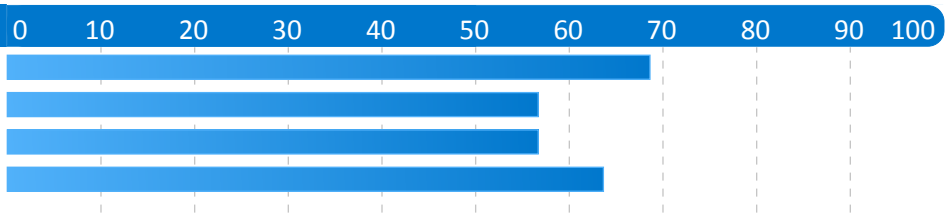
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Components, Templates, and Data Binding	67	
Components, Templates, and Data Binding (Coding Tasks)	62	
Directives and Pipes (Coding Tasks)	62	
Directives and Pipes	63	
Routing and Navigation	85	
Services, Modules, and Dependency Injection	76	

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



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:	Angular Programming (Short)
Authorized:	June 28, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 28, 2026, 7:00:37PM EDT
Completed:	June 28, 2026, 7:00:37PM EDT
Overall Score:	69

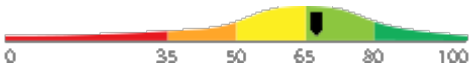
Knowledge and Skills Detail

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

Components, Templates, and Data Binding

Score: 67



Description:

Components are the core building blocks of Angular applications. This topic covers how to create and configure components, write templates using HTML and Angular syntax, and bind data between the component class and the view using property binding, event binding, and two-way binding.

Interpretation:

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

The candidate demonstrates a solid working knowledge of Angular programming, including competency in components, templates, data binding, routing, services, and dependency injection, with reasonable familiarity across most tested subject areas. Some advanced topics such as complex RxJS usage, fine-grained lifecycle hook application, or nuanced debugging techniques may still present occasional challenges. This individual is well-suited for entry-level to mid-level Angular development roles and is capable of contributing effectively to real-world web application projects.

Describe the differences between property binding, event binding, and two-way binding in Angular. When would you use each one?

1	2	3	4	5
Confuses or cannot distinguish between the three binding types; unable to give appropriate use cases.		Correctly identifies the binding types and gives basic use cases but explanation lacks depth or precision.		Clearly distinguishes all three, uses correct syntax examples, and gives practical, well-reasoned use cases for each.

Can you explain what a component is in Angular and describe how you would display a value from your component's TypeScript class in its HTML template?

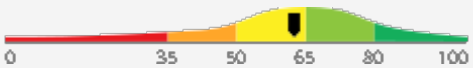
1	2	3	4	5
Vague or incorrect description of components; cannot explain how to display class data in a template.		Correctly describes a component and mentions interpolation or property binding but lacks detail or clarity.		Clearly explains components, accurately describes interpolation and binding syntax with a concrete example.

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

Detail

Components, Templates, and Data Binding (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?

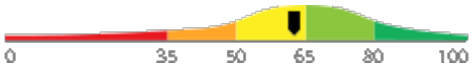
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

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

Detail

Directives and Pipes (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?

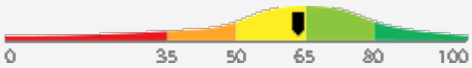
1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

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

Detail

Directives and Pipes

Score: 63



Description:

Directives and pipes are used directly in Angular templates to control how content is displayed. This topic covers built-in structural directives like *ngIf and *ngFor for controlling the DOM, attribute directives for changing element appearance or behavior, and pipes for transforming and formatting data displayed in templates.

Interpretation:

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

The candidate possesses a foundational understanding of Angular directives and pipes, demonstrating working knowledge of common use cases. They can apply basic structural directives and pipes in templates but may lack consistency or depth when handling more complex scenarios.

Interview Guide

How would you use a pipe in an Angular template, and can you describe a situation where you would create a custom pipe instead of using a built-in one?

1	2	3	4	5
Cannot demonstrate pipe usage in a template or gives an incorrect explanation of when to create a custom pipe.		Correctly shows basic pipe usage in a template and gives a reasonable but generic scenario for a custom pipe.		Demonstrates confident pipe usage with syntax, gives a specific and practical use case for a custom pipe with a clear rationale.

What is the purpose of *ngIf and *ngFor in an Angular template? Can you give a simple example of how you would use one of them?

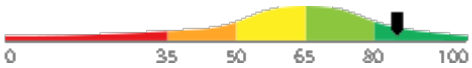
1	2	3	4	5
Cannot explain the purpose of either directive or provides an incorrect example.		Correctly explains one directive and provides a basic but functional example.		Accurately explains both directives, distinguishes their purposes, and provides clear, correct examples for each.

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

Detail

Routing and Navigation

Score: 85



Description:

Angular's router enables navigation between different views or pages within a single-page application. This topic covers how to define routes, configure the RouterModule, use router directives like routerLink and router-outlet in templates, and pass data between routes using route parameters.

Interpretation:

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

The candidate exhibits a comprehensive and advanced mastery of Angular Routing and Navigation. They are highly proficient in all aspects of the subject, including RouterModule configuration, router directives, route parameters, and navigation within single-page applications, and can be expected to apply this knowledge effectively with minimal oversight.

Interview Guide

How would you pass a value, such as a record ID, from one route to another, and how would you read that value in the destination component?

1	2	3	4	5
Cannot describe how to define or read route parameters; confuses query params, route params, or other methods.		Correctly describes using a route parameter in the route definition and mentions ActivatedRoute but lacks implementation detail.		Accurately explains defining a parameterized route, navigating to it, and using ActivatedRoute to read the parameter with a clear example.

How would you set up basic navigation between two pages in an Angular application so that clicking a link loads a different view?

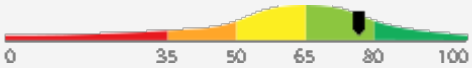
1	2	3	4	5
Cannot describe how to configure routes or use router directives to navigate between views.		Describes setting up routes in an array and using routerLink but omits router-outlet or key configuration details.		Accurately describes defining routes, configuring RouterModule, placing router-outlet, and using routerLink with a clear example.

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

Detail

Services, Modules, and Dependency Injection

Score: 76



Description:

Services are classes used to share data and logic across multiple components. This topic covers how to create services, how Angular's dependency injection system provides services to components and other classes, and how NgModules are used to organize an application and declare which components, directives, and pipes belong together.

Interpretation:

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

The candidate demonstrates a solid and competent understanding of Angular services, NgModules, and dependency injection. They are capable of creating and providing services, organizing application modules, and effectively managing component dependencies in most practical situations.

Interview Guide

Can you explain what the @NgModule decorator does and describe the role of its declarations, imports, and providers arrays?

1	2	3	4	5
Cannot explain the purpose of @NgModule or confuses the role of its main arrays.		Correctly explains the overall purpose of @NgModule and the role of at least two of the three main arrays.		Accurately explains @NgModule and clearly distinguishes the purpose of declarations, imports, and providers with practical examples.

What is a service in Angular, and how would you make a service available to a component that needs to use it?

1	2	3	4	5
Cannot define a service or describe how to provide or inject it into a component.		Correctly defines a service and mentions injection via the constructor but lacks detail on how to provide it.		Clearly defines a service, explains the @Injectable decorator, providedIn options, and constructor injection with confidence.

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

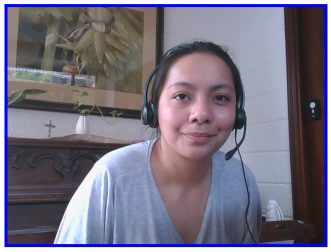
Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries.
 	
The program declares a function called duplicate_array that:
1. Takes a const int pointer to a source array and its length as parameters.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

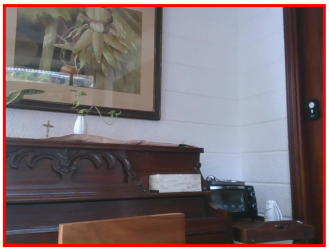
The following photos of the candidate and any identification were uploaded during the assessment session.

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%)

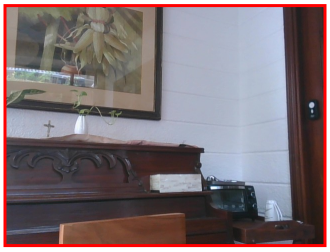


Pre/Post-Test Photo

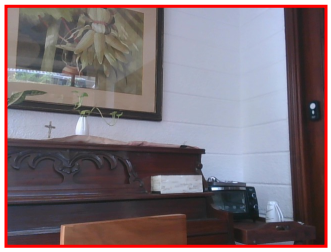
ID Photo



In-Test Error Detected (No Face Detected)

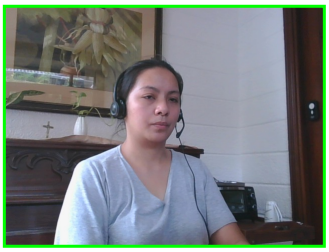


In-Test Error Detected (No Face Detected)

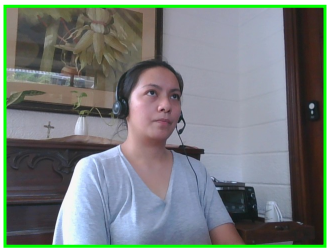


In-Test Error Detected (No Face Detected)

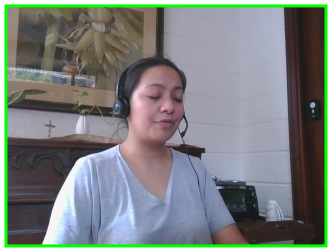
In-Test Photo



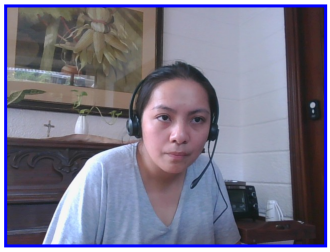
In-Test Photo



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: 20825-1, Key: 0-0, Rpt: 68, Prd: 9647, Created: 2026-06-28 19:00 EDT
- 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 (%)
Components, Templates, and Data Binding	67.8835	Numeric Score	67.8835	16.6667
Components, Templates, and Data Binding (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Directives and Pipes	63.8276	Numeric Score	63.8276	16.6667
Directives and Pipes (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Routing and Navigation	85.4417	Numeric Score	85.4417	16.6667
Services, Modules, and Dependency Injection	76.8353	Numeric Score	76.8353	16.6667
Weighted Average:				69.9908
Final Overall Score:				69

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.