Passcert

Higher Quality, better service!

Q&A

Http://www.passcert.com

We offer free update service for one year.

Exam : **GH-600**

Title : Developing in Agentic AI Systems

Version : DEMO

1.HOTSPOT

Case study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Overview

Contoso Ltd. is a software development company located in the United States.

Existing Environment

GitHub Environment

Contoso uses GitHub Enterprise and assigns GitHub Copilot Pro+ licenses to its developers. The developers use Microsoft Visual Studio Code as their IDE.

Contoso has a customer portal.

The code for the portal is stored in a GitHub repository named repo1 that contains the following:

A custom agent named agent1 that includes instructions to review specs related to best practices

A custom instruction file named validate-instructions.md that is used to validate tone of voice and applies to all .md and .txt files

A custom instruction file named codereview.instructions.md that is used by the Copilot coding agent but is excluded for use by the Copilot code review repo1 has the following structure:

The front-end is stored in the /frontend folder.

The API logic is stored in the /api folder.

Contoso has a second repository named repo2 that contains a legacy .NET application named App1 built by using .NET 6. repo2 has a multi-agent workflow for modernization tasks.

Contoso enables the Model Context Protocol (MCP) registry and allows the Microsoft Learn MCP Server. Every developer must configure their own connection to the Learn MCP Server.

Problem Statements

The developers working in repo1 report that the Microsoft Learn documentation is NOT being retrieved when they attempt to validate a design by using agent1.

The testing team at Contoso identifies that the customer portal uses inconsistent UI styles, which leads to customer confusion and branding issues. The UI inconsistencies stem from variations in the folder structure.

Agent Logs

You have the following logs for the multi-agent workflow used in repo2.

```
2026-03-20 13:32:46.705 [info] [code-referencing] Public code references are enabled.
2026-03-20 13:32:46.828 [info] Fetched model metadata in 124ms e0c73fe9-b053-4d9b-a2c3-a34d25bda491
2026-03-20 13:32:46.832 [info] ccreq:2b92362f.copilotmd | markdown
2026-03-20 13:33:08.382 [info] [FolderRepositoryManager] Workspace isolation mode selected for session untitled-432b4bb1-1406-474e-b15e-e10575a945bb, skipping worktree creation
2026-03-20 13:33:08.384 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:33:08.481 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:33:08.481 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:33:08.903 [info] Using Copilot CLI session: de04f0fe-4c01-4963-ae73-374e482eb771 (isNewSession: true, isolationEnabled: false, workingDirectory: file:///d%3A/git/code, worktreePath: undefined)
2026-03-20 13:33:09.208 [info] [CopilotCLISession] Invoking session de04f0fe-4c01-4963-ae73-374e482eb771
2026-03-20 13:33:09.209 [info] ccreq:e5987520.copilotmd | markdown
2026-03-20 13:34:49.089 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:34:49.093 [info] request done: requestId: [8fac4c58-84fa-46d3-ba8f-2a5a61119df9] model deployment ID: []
2026-03-20 13:34:49.095 [info] ccreq:e52a1868.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 606ms | [copilotLanguageModelWrapper]
2026-03-20 13:34:49.765 [info] ccreq:9ffad198.copilotmd | markdown
2026-03-20 13:35:04.476 [info] [FolderRepositoryManager] Workspace isolation mode selected for session untitled-e49263e-a119-40af-93bc-0c7e5ff093cc, skipping worktree creation
2026-03-20 13:35:04.477 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:04.538 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:04.538 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:35:04.847 [info] Using Copilot CLI session: 6852c5dc-ce11-4f86-85aa-d879812a12d0 (isNewSession: true, isolationEnabled: false, workingDirectory: file:///d%3A/git/code, worktreePath: undefined)
2026-03-20 13:35:05.049 [info] [CopilotCLISession] Invoking session 6852c5dc-ce11-4f86-85aa-d879812a12d0
2026-03-20 13:35:10.730 [info] ccreq:fb9aca1c.copilotmd | markdown
2026-03-20 13:35:21.212 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:35:21.215 [info] request done: requestId: [6e4e18a3-6196-434b-84b3-a972330e822a] model deployment ID: []
2026-03-20 13:35:21.217 [info] ccreq:75e978d7.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 618ms | [copilotLanguageModelWrapper]
2026-03-20 13:35:31.473 [info] ccreq:76a9c1aa.copilotmd | markdown
2026-03-20 13:35:31.859 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:31.909 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:31.909 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:35:32.223 [info] ccreq:adfc457f.copilotmd | markdown
2026-03-20 13:35:32.518 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:35:32.522 [info] request done: requestId: [a16313c7-578e-42c2-a7d3-e755e76ccb64] model deployment ID: []
2026-03-20 13:35:32.523 [info] ccreq:0fb9e216.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 457ms | [copilotLanguageModelWrapper]
2026-03-20 13:35:32.634 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:32.654 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:32.654 [info] [CopilotCLISession]   github: type=http
```

Requirements

Planned Changes

Contoso plans to have all agents and developers in repo1 use the Microsoft Learn MCP to ensure that reviews are validated by using the appropriate documentation. This must be implemented centrally.

Contoso plans to leverage AI-powered coding agents to implement new portal features and pages.

Technical Requirements

App1 must be upgraded to .NET 10. A previous upgrade attempt was started by using the Copilot modernization agent, but the attempt was never finalized.

You plan to retry the upgrade. You must first analyze App1 by using AI, and then generate a report that contains breaking changes and deprecated patterns before retrying the upgrade.

All AI-generated code for UI styling must adhere to a predefined folder structure.

The architects at Contoso need help building implementation plans for repo1. The company wants to implement a new agent named agent2 to analyze the code base and the code requirements, and then respond with a detailed plan. The agent must NOT be able to edit files or run local commands.

The developers must be able to delegate work to the Copilot coding agent by assigning issues to the agent.

You need to implement agent2 to meet the technical requirements.

How should you complete the YAML configuration? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content. NOTE: Each correct selection is worth one point.

Answer Area

```
---

name: implementation-planner

description: Creates detailed implementation plans and technical specifications in markdown format

tools: [

        [ 'agent', 'edit', 'search' ]
        [ 'execute', 'read', 'web' ]

  'microsoftdocs/mcp/docs_search',

  'microsoftdocs/mcp/docs_fetch'

]

---
```

Answer:

Answer Area

```
---

name: implementation-planner

description: Creates detailed implementation plans and technical specifications in markdown format

tools: [

        [ 'agent', 'edit', 'search' ]
        [ 'execute', 'read', 'web' ]

  'microsoftdocs/mcp/docs_search',

  'microsoftdocs/mcp/docs_fetch'

]

---
```

2.Case study
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Overview
Contoso Ltd. is a software development company located in the United States.

Existing Environment
GitHub Environment
Contoso uses GitHub Enterprise and assigns GitHub Copilot Pro+ licenses to its developers. The developers use Microsoft Visual Studio Code as their IDE.
Contoso has a customer portal.
The code for the portal is stored in a GitHub repository named repo1 that contains the following:
A custom agent named agent1 that includes instructions to review specs related to best practices
A custom instruction file named validate-instructions.md that is used to validate tone of voice and applies to all .md and .txt files
A custom instruction file named codereview.instructions.md that is used by the Copilot coding agent but is excluded for use by the Copilot code review repo1 has the following structure:
The front-end is stored in the /frontend folder.
The API logic is stored in the /api folder.
Contoso has a second repository named repo2 that contains a legacy .NET application named App1 built by using .NET 6. repo2 has a multi-agent workflow for modernization tasks.
Contoso enables the Model Context Protocol (MCP) registry and allows the Microsoft Learn MCP Server. Every developer must configure their own connection to the Learn MCP Server.

Problem Statements

The developers working in repo1 report that the Microsoft Learn documentation is NOT being retrieved when they attempt to validate a design by using agent1.
The testing team at Contoso identifies that the customer portal uses inconsistent UI styles, which leads to customer confusion and branding issues. The UI inconsistencies stem from variations in the folder structure.

Agent Logs
You have the following logs for the multi-agent workflow used in repo2.

```
2026-03-20 13:32:46.705 [info] [code-referencing] Public code references are enabled.
2026-03-20 13:32:46.828 [info] Fetched model metadata in 124ms e0c73fe9-b053-4d9b-a2c3-a34d25bda491
2026-03-20 13:32:46.832 [info] ccreq:2b92362f.copilotmd | markdown
2026-03-20 13:33:08.382 [info] [FolderRepositoryManager] Workspace isolation mode selected for session untitled-432b4bb1-1406-474e-b15e-e10575a945bb, skipping worktree creation
2026-03-20 13:33:08.384 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:33:08.481 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:33:08.481 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:33:08.903 [info] Using Copilot CLI session: de04f0fe-4c01-4963-ae73-374e482eb771 (isNewSession: true, isolationEnabled: false, workingDirectory: file:///d%3A/git/code, worktreePath: undefined)
2026-03-20 13:33:09.208 [info] [CopilotCLISession] Invoking session de04f0fe-4c01-4963-ae73-374e482eb771
2026-03-20 13:33:09.209 [info] ccreq:e5987520.copilotmd | markdown
2026-03-20 13:34:49.089 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:34:49.093 [info] request done: requestId: [8fac4c58-84fa-46d3-ba8f-2a5a61119df9] model deployment ID: []
2026-03-20 13:34:49.095 [info] ccreq:e52a1868.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 606ms | [copilotLanguageModelWrapper]
2026-03-20 13:34:49.765 [info] ccreq:9ffad198.copilotmd | markdown
2026-03-20 13:35:04.476 [info] [FolderRepositoryManager] Workspace isolation mode selected for session untitled-e49263e-a119-40af-93bc-0c7e5ff093cc, skipping worktree creation
2026-03-20 13:35:04.477 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:04.538 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:04.538 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:35:04.847 [info] Using Copilot CLI session: 6852c5dc-ce11-4f86-85aa-d879812a12d0 (isNewSession: true, isolationEnabled: false, workingDirectory: file:///d%3A/git/code, worktreePath: undefined)
2026-03-20 13:35:05.049 [info] [CopilotCLISession] Invoking session 6852c5dc-ce11-4f86-85aa-d879812a12d0
2026-03-20 13:35:10.730 [info] ccreq:fb9aca1c.copilotmd | markdown
2026-03-20 13:35:21.212 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:35:21.215 [info] request done: requestId: [6e4e18a3-6196-434b-84b3-a972330e822a] model deployment ID: []
2026-03-20 13:35:21.217 [info] ccreq:75e978d7.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 618ms | [copilotLanguageModelWrapper]
2026-03-20 13:35:31.473 [info] ccreq:76a9c1aa.copilotmd | markdown
2026-03-20 13:35:31.859 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:31.909 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:31.909 [info] [CopilotCLISession]   github: type=http
2026-03-20 13:35:32.223 [info] ccreq:adfc457f.copilotmd | markdown
2026-03-20 13:35:32.518 [info] message 0 returned. finish reason: [stop]
2026-03-20 13:35:32.522 [info] request done: requestId: [a16313c7-578e-42c2-a7d3-e755e76ccb64] model deployment ID: []
2026-03-20 13:35:32.523 [info] ccreq:0fb9e216.copilotmd | success | gpt-4o-mini -> gpt-4o-mini-2024-07-18 | 457ms | [copilotLanguageModelWrapper]
2026-03-20 13:35:32.634 [info] [CopilotCLIMCPHandler] loadMcpConfig called. CLIMCPServerEnabled=false
2026-03-20 13:35:32.654 [info] [CopilotCLISession] Passing 1 MCP server(s) to SDK: [github]
2026-03-20 13:35:32.654 [info] [CopilotCLISession]   github: type=http
```

Requirements
Planned Changes
Contoso plans to have all agents and developers in repo1 use the Microsoft Learn MCP to ensure that reviews are validated by using the appropriate documentation. This must be implemented centrally.
Contoso plans to leverage AI-powered coding agents to implement new portal features and pages.

Technical Requirements
App1 must be upgraded to .NET 10. A previous upgrade attempt was started by using the Copilot modernization agent, but the attempt was never finalized.
You plan to retry the upgrade. You must first analyze App1 by using AI, and then generate a report that contains breaking changes and deprecated patterns before retrying the upgrade.
All AI-generated code for UI styling must adhere to a predefined folder structure.
The architects at Contoso need help building implementation plans for repo1. The company wants to implement a new agent named agent2 to analyze the code base and the code requirements, and then respond with a detailed plan. The agent must NOT be able to edit files or run local commands.
The developers must be able to delegate work to the Copilot coding agent by assigning issues to the agent.

Before App1 is upgraded, you need to verify each individual upgrade step and whether all tests have passed.

Which file should you use?
A. assessment.md
B. plan.md
C. agent.md
D. tasks.md
Answer: A

3.DRAG DROP
You have a GitHub repository that has a GitHub Actions workflow. The workflow runs an AI agent.
You need to ensure that the default GITHUB_TOKEN permissions are read-only, and write access is granted to only the job that performs repository write operations. The workflow must be able to create and approve pull requests only when explicitly enabled.
How should you complete the workflow? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content. NOTE: Each correct selection is worth one point.

Values

- contents: read
- contents: write
- issues: write
- permissions: read-all
- pull-requests: read
- pull-requests: write

Answer Area

```
name: agent-pr

on:
  pull_request:

permissions:
      [                    ]

jobs:
  analyze:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v5
      - name: Read-only analysis
        run: echo "Analyze PR content and post a comment via API"
  update_artifacts:
    runs-on: ubuntu-latest
    permissions:
          [                    ]
    steps:
      - uses: actions/checkout@v5
      - name: Update generated artifacts
        run: |
          echo "regenerate"
      - name: Create pull request
            [                    ]
        run: echo "create PR"
```

Answer:

Values

- contents: read
- contents: write
- issues: write
- permissions: read-all
- pull-requests: read
- pull-requests: write

Answer Area

```
name: agent-pr
on:
  pull_request:
permissions:
      contents: read
jobs:
  analyze:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v5
      - name: Read-only analysis
        run: echo "Analyze PR content and post a comment via API"
  update_artifacts:
    runs-on: ubuntu-latest
    permissions:
          contents: write
    steps:
      - uses: actions/checkout@v5
      - name: Update generated artifacts
        run: |
          echo "regenerate"
      - name: Create pull request
            pull-requests: write
        run: echo "create PR"
```

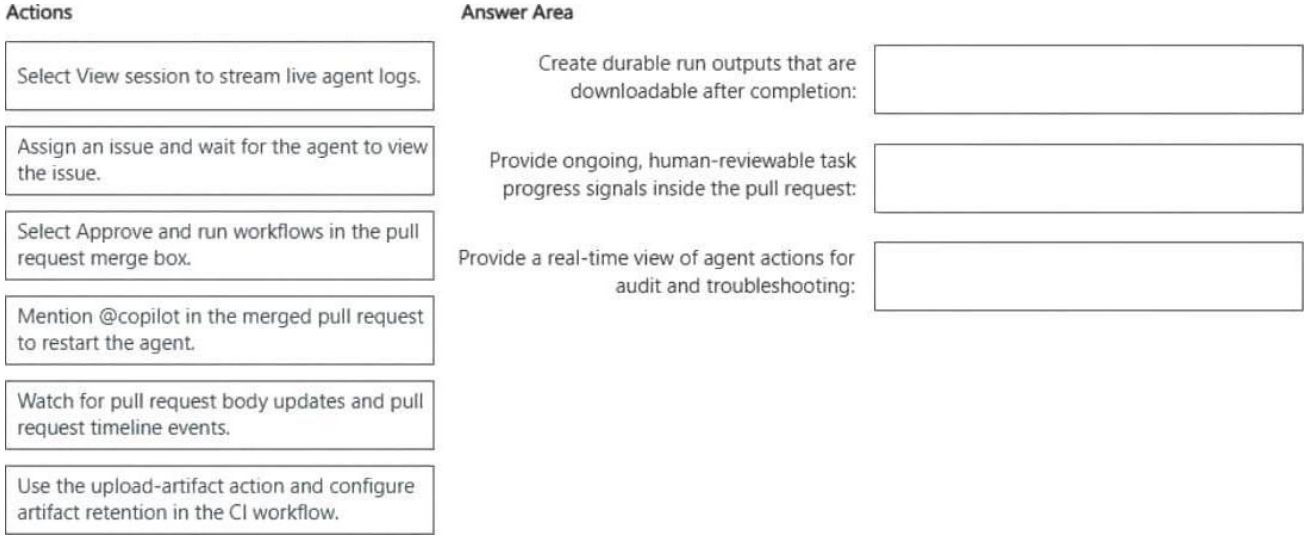
4.DRAG DROP

You have a GitHub repository that uses the GitHub Copilot coding agent to resolve issues and create draft pull requests. The repository uses GitHub Actions for CI, and reviewers rely on pull request timelines and workflow artifacts to understand what the agent did.

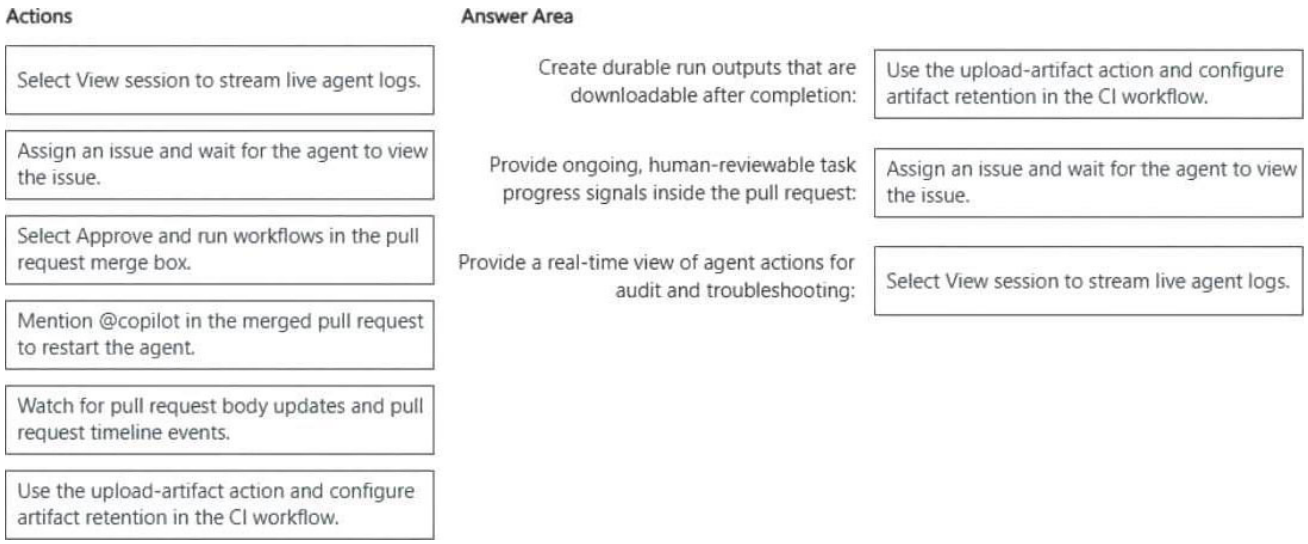
During long-running agent tasks, the reviewers lose track of decisions and validation steps, which causes repeated questions and reworks when context drifts between iterations.

You need to persist task progress and decisions as durable artifacts and ensure that the reviewers can verify what the agent did during and after execution by using GitHub as the system of record.

What should you do for each requirement? To answer, drag the appropriate actions to the correct requirements. Each action may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content. NOTE: Each correct selection is worth one point.



Answer:



5.DRAG DROP

You have a GitHub Enterprise Cloud Organization that uses the GitHub Copilot coding agent to resolve issues asynchronously.

When an issue is assigned to GitHub Copilot, the agent creates a draft pull request, but your team cannot always tell whether the agent is actively working, has completed its session, or is awaiting workflow approval.

Which execution context does each signal indicate? To answer, drag the appropriate context to the correct signals. Each signal may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content. NOTE: Each correct selection is worth one point.

Contexts	Answer Area	
The agent session timed out and must be reassigned.	The eyes emoji (👀) reaction appears on the issue:	
A human must manually approve and run the workflows.	The pull request timeline shows Copilot started work:	
The agent cannot see the repository content due to exclusions.	A draft pull request exists, but GitHub Actions checks are not running:	
The agent session is actively running and generating live logs.		
The agent comment was ignored because the user lacks write access.		
The agent acknowledges the assignment and will create the draft pull request.		

Answer:

Contexts	Answer Area	
The agent session timed out and must be reassigned.	The eyes emoji (👀) reaction appears on the issue:	The agent acknowledges the assignment and will create the draft pull request.
A human must manually approve and run the workflows.	The pull request timeline shows Copilot started work:	The agent session is actively running and generating live logs.
The agent cannot see the repository content due to exclusions.	A draft pull request exists, but GitHub Actions checks are not running:	A human must manually approve and run the workflows.
The agent session is actively running and generating live logs.		
The agent comment was ignored because the user lacks write access.		
The agent acknowledges the assignment and will create the draft pull request.		