**ASSESSMENT PLAN**

**Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs**

**Non-WSQ**

All rights reserved. This document is provided for the explicit use and guidance of parties approved by NTUC LearningHub Pte Ltd as information resource only. Any other use of this document or parts thereof, including reproduction, publication, distribution, transmission, re-transmission or storage in a retrieval system in any form, electronic or otherwise, for purposes other than that expressly stated above without the express permission of NTUC LearningHub Pte Ltd is strictly prohibited. Printed in Singapore.

Table of Contents

[1. Overview of Assessment Plan 3](#_Toc166236243)

[2. Matrix of Assessment Methods and Evidence Gathering Plan 5](#_Toc166236244)

[3. Assessment Method Specifications 7](#_Toc166236245)

[3a. Written Assessment 8](#_Toc166236246)

[3b. Practical Performance 9](#_Toc166236247)

[4. Assessment Code of Practice 10](#_Toc166236248)

[5. Assessment Management Organisation 12](#_Toc166236249)

[6. Instructions to Assessor 14](#_Toc166236250)

[7. Instructions to Candidate 16](#_Toc166236251)

[8. Checklist for Assessor 17](#_Toc166236252)

[9. Individual Assessment Record 18](#_Toc166236253)

[9a. Individual Assessment Record for Written Assessment 18](#_Toc166236254)

[9b. Individual Assessment Record for Practical Performance 19](#_Toc166236255)

[10 Assessment Summary Record 20](#_Toc166236256)

[Annex A 22](#_Toc166236257)

[Annex B 24](#_Toc166236258)

[Annex C 26](#_Toc166236259)

[Annex D 28](#_Toc166236260)

# 1. Overview of Assessment Plan

To attain competency in the module, **Efficient Software Development Workflow-Version Control, GitHub, Build Images, and Swagger Design for RESTful APIs**, participants are required to undertake and successfully complete each of the assessment activities listed for the competency unit.

A summary of the key assessment specifications is shown below:

**Assessment Method**

|  |  |
| --- | --- |
| **Knowledge and Abilities** | **Assessment Method(s)** |
| K1: Design requirements for simple, basic software components. | **WA** |
| K2: Basic software design tools and techniques. | **PP** |
| K3: Types of controls, elements and features in software. | **WA** |
| K4: Indicators of software functionality and interoperability. | **PP** |
| K5: Documentation of design details. | **PP** |
| A1: Design a simple software component or interface according to functional specifications and business requirements. | **WA** |
| A2: Utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles. | **WA** |
| A3: Identify relevant controls, elements and features to be included in the software to meet its design objectives. | **WA** |
| A4: Assess functionality and interoperability of different elements or components in the software design. | **PP** |
| A5: Produce detailed design documentation mapped to user specifications. | **PP** |

**Legend**

**WA:**  Written Assignment, **PP:** Practical Performance, **RP**: Role Play

**Assessment Duration (per candidate)**

|  |  |
| --- | --- |
| **Assessment method(s)** | **Duration** |
| Written Assessment | **30 mins** |
| Practical Performance | **90 mins** |
| **Total time** | **120 mins** |

**Assessment Venue**

|  |  |
| --- | --- |
| **Assessment Method(s)** | **Venue** |
| Written Assessment | **NTUC LearningHub, Client’s Premise, or Online** |
| Practical Performance |

# 2. Matrix of Assessment Methods and Evidence Gathering Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Abilities (A)**  **And**  **Knowledge (K)** | **Evidence Requirement** | **Assessment Methods**  (Assessment Tools) | |
| **WA** | **PP** |
| **K3**  Types of controls, elements and features in software.  **A2**  Utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles.  **A3**  Identify relevant controls, elements and features to be included in the software to meet its design objectives. | **Tell me (Knowledge)**  Candidate is able to understand the types of controls, elements and features in software.  **Show me (Product)**  Candidate is able to utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles.  Candidate is able to identify relevant controls, elements and features to be included in the software to meet its design objectives.  **Question 1A:**  Show how you can review the code from the “new-feature-modal” branch. After the code has been reviewed and the Pull Request has been approved, show how your team member can merge the new branch into the main branch and update the main branch.  **Question 1B**:  List at least 3 best practices for managing code changes and resolving conflicts | ü |  |
| **K1**  Design requirements for simple, basic software components.  **A1**  Design a simple software component or interface according to functional specifications and business requirements. | **Tell me (Knowledge)**  Candidate is able to design requirements for simple, basic software components.  **Show me (Product)**  Candidate is able to design a simple software component or interface according to functional specifications and business requirements.  **Question 2**:  What are the benefits of containerization? List at least 3. | ü |  |
| **K2**  Basic software design tools and techniques.  **K4**  Indicators of software functionality and interoperability.  **A4**  Assess functionality and interoperability of different elements or components in the software design. | **Tell me (Knowledge)**  Candidate is able to show the usage of basic software design tools and techniques.  Candidate is able to show the indicators of software functionality and interoperability.  **Show me (Product)**  Candidate is able to assess functionality and interoperability of different elements or components in the software design.  **Question 1**:  You have developed a React app that you want to containerize using Docker. Assume you have already set up Docker on your local development environment. Show the steps involved:   * Create and submit your Dockerfile. * Show screenshot where the dockerfile and dockerignore should be in your file structure. * Build a docker image. Show screenshot in Docker Desktop that you have successfully created a docker image. * Build the container to run the React app. Show screenshot that the react app is compiled and running successfully. * Show screenshot in Docker Desktop that the docker container is running. |  | ü |
| **K5**  Documentation of design details.  **A5**  Produce detailed design documentation mapped to user specifications. | **Tell me (Knowledge)**  Candidate is able to work with the documentation of design details.  **Show me (Process)**  Candidate is able to produce detailed design documentation mapped to user specifications.  **Question 2**:  You are working on a project where you need to define a RESTful API using Swagger for documentation purposes. Define User API endpoints and document them using Swagger editor. |  | ü |

# 3. Assessment Method Specifications

**Flow of conducting the assessment**

List the sequence, the flow of assessment.

* Written Questioning (30 mins)
* Practical Performance (90 mins)

In the event if the evidence gathered during WQ and PP is insufficient, the assessor may ask the candidate supplementary question to gather evidence prior to conducting OQ.

**Assessment ratio**

|  |  |
| --- | --- |
| **Assessment Method** | **Assessor: Candidate Ratio** |
| Written Assessment | 1: 20 |
| Practical Performance | 1: 20 |

## 3a. Written Assessment

|  |  |  |
| --- | --- | --- |
| **S/N** | **Specifications** | **Guidelines** |
| 1 | Abilities | A1: Design a simple software component or interface according to functional specifications and business requirements.  A2: Utilize appropriate software design methods and tools, in line with the organization's software design practice and principles.  A3: Identify relevant controls, elements and features to be included in the software to meet its design objectives. |
| 2 | Knowledge | K1: Design requirements for simple, basic software components.  K3: Types of controls, elements and features in software. |
| 3 | Duration | 30 mins |
| 4 | Assessment Venue | Online classroom (Virtual Learning Classroom) through Zoom |
| 5 | Assessment Resources | * Laptop * Canvas (LHUB LMS) |
| 6 | Assessment Set-up Requirements | * Online classroom (Virtual Learning Classroom) through Zoom * Canvas (LHUB LMS) |
| 7 | Conducting the Written Assessment | * The assessment is an open-book assessment. * The learner is to complete the assessment within 30 mins. |
| 8 | Recording the Assessment Result | Record findings and conclusions in the Individual Assessment Record.  Where the assessment for each specific Abilities is completed, place a tick in the “C” column, for “Competent” or in the “NYC” column for “Not-Yet-Competent” as an indication of outcome.  Record reasons for “NYC” in remarks column. |
| 9 | Candidate Summary and Feedback | Record feedback and findings into the Assessment Summary Record. |
| 10 | Required Assessment Tools | * Course Assessment Booklet (refer to Annex A)/ Online via LHUB LMS Canvas |
| 11 | Addressing Candidate’s Special Needs | * Let the candidate be seated in the front row of the classroom * Set the font size for the assessment paper to be bigger |

## 3b. Practical Performance

|  |  |  |
| --- | --- | --- |
| **S/N** | **Specifications** | **Guidelines** |
| 1 | Abilities | A4: Assess functionality and interoperability of different elements or components in the software design.  A5: Produce detailed design documentation mapped to user specifications. |
| 2 | Knowledge | K2: Basic software design tools and techniques.  K4: Indicators of software functionality and interoperability.  K5: Documentation of design details. |
| 3 | Duration | 90 mins |
| 4 | Assessment Venue | Online classroom (Virtual Learning Classroom) through Zoom |
| 5 | Assessment Resources | * Laptop * Canvas (LHUB LMS) |
| 6 | Assessment Set-up Requirements | * Online classroom (Virtual Learning Classroom) through Zoom * Canvas (LHUB LMS) |
| 7 | Conducting the Practical Performance | * The assessment is an open-book assessment. * The learner is to complete the assessment within 90 mins. |
| 8 | Recording the Assessment Result | Record findings and conclusions in the Individual Assessment Record.  Where the assessment for each specific Abilities is completed, place a tick in the “C” column, for “Competent” or in the “NYC” column for “Not-Yet-Competent” as an indication of outcome.  Record reasons for “NYC” in remarks column. |
| 9 | Candidate Summary and Feedback | Record feedback and findings into the Assessment Summary Record. |
| 10 | Required Assessment Tools | * Course Assessment Booklet (refer to Annex A)/ Online via LHUB LMS Canvas |
| 11 | Addressing Candidate’s Special Needs | * Let the candidate be seated in the front row of the classroom * Set the font size for the assessment paper to be bigger |

# 4. Assessment Code of Practice

4.1 This code of practice provides:

* Assessors with direction on the standard of practice.
* Candidates with assurance of the standards of practice expected of assessors.
* Employers with assurance of the standards maintained by NTUC LearningHub Pte Ltd in the conduct of assessment.

4.2 The following code of practice is based on the international code of ethics and practice (The National Council for Measurement in Education [NCME]).

* 1. The differing needs and requirements of the person(s) being assessed, the local enterprise(s) and/or industry are identified and handled with sensitivity.
  2. Potential forms of conflict of interest in the assessment process and/or outcomes are identified and appropriate referrals are made, if necessary.
  3. All forms of harassment are avoided throughout the planning, conducting, reviewing and reporting of the assessment outcomes.
  4. The rights of the candidate(s) are protected during and after the assessment.
  5. Personal or interpersonal factors that are not relevant to the assessment of competency must not influence the assessment outcomes.
  6. The candidate is made aware of rights and processes of appeal.
  7. Evidence that is gathered during the assessment is verified for validity, reliability, authenticity, sufficiency and currency.
  8. Assessment decisions are based on available evidence that can be produced and verified by another assessor.
  9. Assessments are conducted within the boundaries of the assessment system policies and procedures.
  10. Formal agreement is obtained from both the candidate(s) and the assessor that the assessment was carried out in accordance with agreed procedures.
  11. Assessment tools, systems, and procedures are consistent with equal opportunity legislation.
  12. The candidate is informed of all assessment reporting processes prior to the Assessment.
  13. The candidate is informed of all known potential consequences of decisions arising from an assessment, prior to the assessment.
  14. Confidentiality is maintained regarding assessment results.
  15. Results are only released with the written permission of the candidate(s).
  16. The assessment results are used consistently with the purposes explained to the candidate.
  17. Self-assessments are periodically conducted to compare current competencies against the required competencies.
  18. Professional development opportunities are identified and sought.
  19. Opportunities for networking amongst assessors are created and maintained.
  20. Opportunities are created for technical assistance in planning, conducting and reviewing assessment procedures and outcomes.

# 5. Assessment Management Organisation

**Assessment Appeal Panel**

|  |
| --- |
| Assessment  Appeal Panel |
| I |
| Assessment Manager |
| I |
| Assessors |

**5.1 Assessment Appeal Panel**

* The Assessment Appeal Panel is responsible for reviewing and giving a decision on appeals against a Not-Yet-Competent award
* The panel is comprised of a Management Representative, the Assessment Manager and one Assessor who is independent of the case in question.

**5.2 Assessment Manager**

The Assessment Manger is responsible:

* For the administration and conduct of assessment
* Monitoring and continuous improvement of the assessment process and tools
* For the continuous professional development of the pool of assessors
* Managing the process and outcome of the appeal cases.

**5.3 Assessor**

* The Assessor is responsible for conducting the assessment according to the Assessment Plan.

**5.4 Appeal Procedure**

1. The candidate has the right to challenge the assessment decision made by the assessor.

When giving feedback to the candidate about the assessment, the Assessor must ask the candidate if he/she agrees with the outcome.

1. The assessor and the candidate **MUST** sign the Individual Assessment Summary Record regardless of outcome.
2. The candidate should notify the Assessor if he/she is not satisfied with the assessment outcome and intends to appeal against the decision. The Assessor should report the candidate’s intention in the Feedback section of the Individual Assessment Summary Record.
3. The Assessor should notify the Assessment Manager regarding the candidate’s intention to lodge an appeal.
4. The candidate must lodge the appeal in writing giving reasons for the appeal together with any appropriate fee.
5. The Assessment Manager will collect information from the candidate and Assessor and give a decision.
6. A record of the appeal and any subsequent actions and findings will be made.

# 6. Instructions to Assessor

**6a. Instructions to Assessor (Pre - Assessment)**

**General instructions for assessment**

* There are 2 parts to the assessment indicated above.
* The assessments are executed as indicated above and they are open-book assessments.
* The trainer-to-trainee ratio is as indicated above.
* The assessments are to be conducted during the scheduled time.
* Candidates must be made aware of the need to complete the whole assessment within
* Candidates unable to attend the scheduled session must make arrangements with LHUB for undertaking the assessment at an alternative time.
* At the end of the assessment, feedback will be given to the candidate regarding their C/NYC status.
* Appeal procedures must be reviewed prior to the assessment.
* All documents must be completed and submitted to LHUB upon completion of the entire assessment.

**6b. Instructions to Assessor (During Assessment)**

**Preparation for Written Assessment**

* Candidates are to complete the assessment without any consultation/discussion.
* Should candidate have difficulties writing, the candidate can narrate while the assessor writes down the content in the forms on a one-to-one basis.

**Preparation for Practical Performance**

* Candidates are to complete the assessment without any consultation/discussion.
* Should candidate have difficulties writing, the candidate can narrate while the assessor writes down the content in the forms on a one-to-one basis.

# 7. Instructions to Candidate

**General instructions for assessment**

* Candidate is to be punctual for the assessment.
* There are 2 parts to the assessment indicated above.
* The trainer-to-trainee ratio is as indicated above.
* The assessment will be conducted during the scheduled time.
* Candidates will be made aware of the need to complete the whole assessment time within the time indicated above.
* Candidates unable to attend the scheduled session must make arrangements with LHUB for undertaking the assessment at an alternative time.
* At the end of the assessment, feedback will be given to the candidate regarding their C/NYC status.
* Appeal procedures will be shared prior to the assessment.

**Conduct of Written Assessment**

* This is an open-book assessment.
* Candidate is to complete the assessment within the time indicated above.

**Conduct of Practical Performance**

* This is an open-book assessment.
* Candidate is to complete the assessment within the time indicated above.

# 8. Checklist for Assessor

|  |  |  |
| --- | --- | --- |
| **A. Prior to the Assessment, I...** | **Tick** | **Remarks** |
| 1. ensured that the candidate is informed regarding the venue and schedule of assessment |  |  |
| 2. reviewed the current copies of the performance statement to be assessed, the assessment and evidence gathering plan, the assessment checklist and the organisations standard operating procedures (SOP) |  |  |
| 3. reviewed the performance statement and evidence plan to ensure that I clearly understood the instructions and the requirements of the assessment process. |  |  |
| 4. identified and accommodated any special needs candidate |  |  |
| 5. checked the logistics set up and resources for the assessment. |  |  |
| **B. During the Assessment, I:** | | |
| 1. introduced myself and confirmed the identities of the candidates. |  |  |
| 2. put the candidates at ease by being friendly and helpful. |  |  |
| 3. explained to the candidates the purpose, context and benefits of the assessment. |  |  |
| 4. ensured the candidates understood the process of the assessment. |  |  |
| 5. gave the candidates an overview of performance statement to be assessed. |  |  |
| 6. explained the appeal procedure and the results reporting procedure. |  |  |
| 7. encouraged candidates to seek clarifications if in doubt. |  |  |
| 8. asked candidates for feedback on the assessment. |  |  |
| 9. explained appropriate legal, safety and ethical issues |  |  |
| **C. After the Assessment, I:** | | |
| 1. ensured that the candidate was given constructive feedback. |  |  |
| 2. completed and signed the Individual Assessment Summary Record and the Assessment Record – Class Summary |  |  |
| 3. thanked the candidate for participating in the assessment. |  |  |

# 9. Individual Assessment Record

## 9a. Individual Assessment Record for Written Assessment

This is a record on the competency assessment of each individual candidate which must be completed and signed by both the assessor and the candidate at the end of the assessment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Candidate Name** | | **NRIC** | | |
| **Job Title** | | **Tel** | | |
| **Company** | | | | |
| **Course Title: Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs** | | | | |
|  | | | | |
| **Abilities and Knowledge** | **Assessment Criteria** | **Please Tick** | | **Remarks** |
| **C** | **NYC** |
| **K3**  Types of controls, elements and features in software.  **A2**  Utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles.  **A3**  Identify relevant controls, elements and features to be included in the software to meet its design objectives. | **Tell me (Knowledge)**  Candidate is able to understand the types of controls, elements and features in software.  **Show me (Product)**  Candidate is able to utilise appropriate software design methods and tools, in line with the organisation's software design practice and principles.  Candidate is able to identify relevant controls, elements and features to be included in the software to meet its design objectives. |  |  |  |
| **K1**  Design requirements for simple, basic software components.  **A1**  Design a simple software component or interface according to functional specifications and business requirements. | **Tell me (Knowledge)**  Candidate is able to design requirements for simple, basic software components.  **Show me (Product)**  Candidate is able to design a simple software component or interface according to functional specifications and business requirements. |  |  |  |

## 9b. Individual Assessment Record for Practical Performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Title:** **Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs** | | | | |
|  | | | | |
| **Abilities and Knowledge** | **Assessment Criteria** | **Please Tick** | | **Remarks** |
| **C** | **NYC** |
| **K2**  Basic software design tools and techniques.  **K4**  Indicators of software functionality and interoperability.  **A4**  Assess functionality and interoperability of different elements or components in the software design. | **Tell me (Knowledge)**  Candidate is able to show the usage of basic software design tools and techniques.  Candidate is able to show the indicators of software functionality and interoperability.  **Show me (Product)**  Candidate is able to assess functionality and interoperability of different elements or components in the software design. |  |  |  |
| **K5**  Documentation of design details.  **A5**  Produce detailed design documentation mapped to user specifications. | **Tell me (Knowledge)**  Candidate is able to work with the documentation of design details.  **Show me (Process)**  Candidate is able to produce detailed design documentation mapped to user specifications. |  |  |  |

# 10 Assessment Summary Record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Knowledge and Abilities** | **Assessment Method** | | **Competence**  (Please Tick) | |
| **WA** | **PP** | **C** | **NYC** |
| K1  Design requirements for simple, basic software components. | ü |  |  |  |
| K2  Basic software design tools and techniques. |  | ü |  |  |
| K3  Types of controls, elements and features in software. | ü |  |  |  |
| K4  Indicators of software functionality and interoperability. |  | ü |  |  |
| K5  Documentation of design details. |  | ü |  |  |
| A1  Design a simple software component or interface according to functional specifications and business requirements. | ü |  |  |  |
| A2  Utilize appropriate software design methods and tools, in line with the organization's software design practice and principles. | ü |  |  |  |
| A3  Identify relevant controls, elements and features to be included in the software to meet its design objectives. | ü |  |  |  |
| A4  Assess functionality and interoperability of different elements or components in the software design. |  | ü |  |  |
| A5  Produce detailed design documentation mapped to user specifications. |  | ü |  |  |

**Legend**

**WA:**  Written Assignment, **PP:** Practical Performance, **RP**: Role Play

**NA**: Not Applicable, **C:** Competent, **NYC:** Not Yet Competent

This candidate has been assessed for the TSC: **Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs** as being

**Competent 0**

**Not Yet Competent 0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Candidate Name**  **(As in NRIC)** |  | **Assessor Name:** |  |
| **NRIC:** |  | **NRIC:** |  |
| **Candidate Signature:** |  | **Assessor Signature:** |  |
| **Date:** |  | **Date:** |  |

**Note:** By signing, the candidate is agreeing to accept the assessment outcome.

**Assessment feedback:**

|  |
| --- |
| *{Feedback on the overall performance or in the case of NYC; any area of skills gap and improvement needed}* |

# Annex A

**Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs**

**Written Assessment**

**Instructions to candidate:**

State the instructions to candidate (for example)

1. This is an individual exercise.

2. There are a total of 2 questions.

3. A total of 30 minutes are given to complete this written assessment.

4. Submit your answers on the papers provided.

5. Write your full name, NRIC number and date on this page.

|  |  |
| --- | --- |
| **Candidate Name** | **NRIC** |
| **Job Title** | **Tel** |
| **Company** | |
| **BATCH ID** | |

**Question 1A**

You are working on a collaborative React project hosted on GitHub with your team. One of your team members has made a new feature branch called “new-feature-modal” and pushed some changes to it.

You want to review their code before the new feature is merged into the main branch.

Assume that you have already cloned the repo in your local environment.

* Show how you can review the code from the “new-feature-modal” branch
* After the code has been reviewed and the Pull Request has been approved, show how your team members can merge the new branch into the main branch and update the main branch.

|  |
| --- |
|  |

**Question 1B**

List at least 3 best practices for managing code changes and resolving conflicts

|  |
| --- |
|  |

**Question 2**

What are the benefits of containerization? List at least 3.

|  |
| --- |
|  |

# Annex B

**Efficient Software Development Workflow: Version Control, GitHub, Build Images and Swagger Design for RESTful APIs**

**Practical Performance**

**Instructions to candidate:**

State the instructions to candidate (for example)

1. This is an individual exercise.

2. There are a total of 02 questions.

3. A total of 90 minutes are given to complete this written assessment.

4. Submit your answers on the papers provided.

5. Write your full name, NRIC number and date on this page.

|  |  |
| --- | --- |
| **Candidate Name** | **NRIC** |
| **Job Title** | **Tel** |
| **Company** | |
| **BATCH ID** | |

**Question 1**

You have developed a React app that you want to containerize using Docker. Assume you have already set up Docker on your local development environment. Show the steps involved:

* Create and submit your docker file
* Show screenshot where the Dockerfile and dockerignore should be in your file structure.
* Build a docker image. Show screenshot in Docker Desktop that you have successfully created a docker image.
* Build the container for the react app. Show screenshot that the react app is compiled and running successfully.
* Show screenshot in docker desktop that the container is running.

|  |
| --- |
|  |

**Question 2**

You are working on a project where you need to define a RESTful API using Swagger for documentation purposes. Define User API endpoints and document them using Swagger editor with the following specification:

* The User API need to contain:
  + User id
  + Username
  + Email address
* There should be 2 paths: /user and /user/{id}
* GET request to retrieve all users
* POST request to add new users
* GET request to retrieve a specific user by their ID.
* PUT request to update an existing user
* DELETE request to delete a user by their ID

Submit the YAML file and take screenshots of the Swagger editor for the different endpoints.

|  |
| --- |
|  |

# Annex C

**SUGGESTED ANSWERS TO**

**WRITTEN ASSESSMENT**

**Suggested Answer (Written Assessment)**

**Question 1A (A2, A3, K3)**

You are working on a collaborative React project hosted on GitHub with your team. One of your team members has made a new feature branch called “new-feature-modal” and pushed some changes to it.

You want to review their code before the new feature is merged to the main branch.

Assume that you have already cloned the repo in your local environment.

* Show how you can review the code from the “new-feature-modal” branch
* After the code has been reviewed and the Pull Request has been approved, show how your team members can merge the new branch into the main branch and update the main branch.

|  |
| --- |
| Assuming that a cloned repo already exists in your local environment:  you need to switch to the new feature branch by using the following git command:  git checkout new-feature-modal  Then you need to make sure you have the latest update by using:  git pull origin new-feature-modal  Now you can review the code from the new feature branch.  After the code has been reviewed and the Pull Request has been approved, to merge the new branch into the main branch:   * Make sure you are on the main branch:   git checkout main   * To merge:   git merge new-feature-modal   * To update main branch:   git push origin main |

**Question 1B (A2, A3, K3)**

List at least 3 best practices for managing code changes and resolving conflicts

|  |
| --- |
| Best practices for managing code changes and resolving conflicts:   * Branching Strategy: Adopt a clear branching strategy, such as Gitflow or GitHub flow, based on the project's needs. Use feature branches for new features, hotfix branches for critical patches, and develop/main branches for ongoing development. * Commit Guidelines: Follow commit message conventions to maintain a clear and informative Git history. Use meaningful commit messages that describe the purpose and context of the changes. * Regular Pull Requests: Create smaller, focused pull requests instead of large, monolithic ones. This makes it easier for reviewers to understand and approve changes. Regularly update your branch with the latest changes from the main branch to avoid conflicts. * Conflict Resolution: When conflicts arise during merges or pull requests, address them promptly. Communicate with collaborators to understand conflicting changes and find consensus on resolutions. * Automated Testing: Integrate automated testing into your workflow to catch issues early. GitHub Actions or other CI/CD tools can be used to run tests automatically on pull requests. |

**Question 2 (A1, K1)**

What are the benefits of containerization? List at least 3.

|  |
| --- |
| Benefits of Containerization:   * Portability: Containers run consistently across different environments. * Isolation: Applications and dependencies are isolated, reducing conflicts. * Scalability: Easily scale applications up or down based on demand. * Resource Efficiency: Containers share the host OS kernel, minimizing overhead. * Version Control: Container images allow versioning and easy rollbacks. |

# Annex D

**SUGGESTED ANSWERS TO**

**PRACTICAL PERFORMANCE**

**Suggested Answer (Practical Performance)**

**Question 1(K2, K4, A4)**

You have developed a React app that you want to containerize using Docker. Assume you have already set up Docker on your local development environment. Show the steps involved:

* Create and submit your docker file
* Show screenshot where the Dockerfile and dockerignore should be in your file structure.
* Build a docker image. Show screenshot in Docker Desktop that you have successfully created a docker image.
* Build the container for the react app. Show screenshot that the react app is compiled and running successfully.
* Show screenshot in docker desktop that the container is running.

|  |
| --- |
| The Dockerfile and dockerignore should be in the root directory of the react app:    The Dockerfile:    The dockerignore file:    To build a docker image, run the following in the terminal:  docker build -t docker-react .    This screenshot shows that the image is being used in a container:    To run the container with port mapping:  docker run -p 3000:3000 docker-react  To show that the react app is running:    This screenshot shows that the container is running: |

**Question 2 (A5, K5)**

You are working on a project where you need to define a RESTful API using Swagger for documentation purposes. Define User API endpoints and document them using Swagger editor with the following specification:

* The User API need to contain:
  + Userid
  + Username
  + Email address
* There should be 2 paths: /user and /user/{id}
* GET request to retrieve all users
* POST request to add new users
* GET request to retrieve a specific user by their ID.
* PUT request to update an existing user
* DELETE request to delete a user by their ID

Submit the YAML file and take screenshots of the Swagger editor for the different endpoints.

|  |
| --- |
| The YAML file content:  openapi: 3.0.0  info:  title: User API  version: 1.0.0  description: API for managing user data.  paths:  /users:  get:  summary: Get a list of all users  responses:  '200':  description: Successful response with a list of users  content:  application/json:  example:  - userid: 1  username: JohnDoe  userEmail: john.doe@example.com  - userid: 2  username: JaneSmith  userEmail: jane.smith@example.com  post:  summary: Create a new user  requestBody:  description: User data to be created.  required: true  content:  application/json:  schema:  type: object  properties:  username:  type: string  description: User's username.  userEmail:  type: string  format: email  description: User's email address.  required:  - username  - userEmail  responses:  '201':  description: User created successfully  '400':  description: Invalid request data  /user/{id}:  get:  summary: Get a user by ID  parameters:  - name: id  in: path  description: ID of the user to retrieve  required: true  schema:  type: integer  responses:  '200':  description: Successful response with the user's details  content:  application/json:  example:  userid: 1  username: JohnDoe  userEmail: john.doe@example.com  '404':  description: User not found    put:  summary: Update an existing user  parameters:  - name: id  in: path  description: ID of the user to update  required: true  schema:  type: integer  requestBody:  description: User data to be updated.  required: true  content:  application/json:  schema:  type: object  properties:  username:  type: string  description: User's new username.  userEmail:  type: string  format: email  description: User's new email address.  required:  - username  - userEmail  responses:  '200':  description: User updated successfully  '400':  description: Invalid request data  '404':  description: User not found  delete:  summary: Delete a user by ID  parameters:  - name: id  in: path  description: ID of the user to delete  required: true  schema:  type: integer  responses:  '200':  description: User deleted successfully  '404':  description: User not found  In this YAML file:   * We define an OpenAPI 3.0 document with a title, version, and description. * We specify two paths: /users for both GET and POST operations. * The GET operation retrieves a list of users with a 200 OK response and a JSON example. * The POST operation creates a new user with a 201 Created response and a JSON request body schema. * We've added a new path /user/{id} for GET requests to retrieve a specific user by their ID. * The id parameter is defined in the path and marked as required. * The responses include a 200 OK response with the user's details and a 404 Not Found response if the user with the specified ID is not found.   The screenshots: |