MODULE NAME:	MODULE CODE:
PROGRAMMING 3B	PROG7312
ADVANCED APPLICATION DEVELOPMENT	AAPD7112/w

ASSESSMENT TYPE: POE (PAPER)

TOTAL MARK ALLOCATION: 300 MARKS

TOTAL HOURS: A minimum of 45 HOURS is suggested to complete this assessment.

By submitting this assignment, you acknowledge that you have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IIE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity and Property Rights Policy (IIE023), as well as any rules and regulations published in the student portal.

INSTRUCTIONS:

1. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks. No more than 10% of the assignment may consist of direct quotes.***
2. ***Please ensure that you submit your assignment through Turnitin. Please make sure you attach a similarity report to your POE if you are required to submit a hard copy of your PoE.***
3. ***Make a copy of your assignment before handing it in.***
4. *Assignments must be typed unless otherwise specified.*
5. *Begin each section on a new page.*
6. *Follow all instructions on the PoE cover sheet.*
7. *This is an individual assignment.*

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high-quality academic work. Part of achieving this quality is referencing in a way that is consistent and congruent with the requirements of the referencing style being used.

Therefore, inconsistent and/or incongruent referencing will result in a penalty of **a maximum of ten percent being deducted from the overall percentage** awarded to your assessment submission.

Please note that **evidence of plagiarism** in the form of copied or unreferenced work, absent reference lists, or exceptionally poor referencing **may result in action being taken in accordance with The IIE's Intellectual Integrity and Property Rights Policy (IIE023)**. Similarly, **evidence of excessive AI usage may result in action being taken in accordance with The IIE's Student Conduct, Discipline and Safety Policy (IIE015)**.

Markers are required to provide feedback to students by **circling/underlining the information in the table below that best describes the student's work and by adding constructive commentary where appropriate.** The examples provided are not exhaustive but illustrate the errors.

Deductions

- Where the student's work contains **five or more errors** aligned to the **minor errors column** below, **deduct 5% from the overall percentage.**
- Where the student's work contains **five or more errors** aligned to the **major errors column** below, **deduct 10% from the overall percentage.**
- **Where both minor and major errors** (e.g. two minor and three major, etc.) are present, **deduct 10% only** (and not 5% or 15%) from the overall percentage.

Required: **Consistent and congruent referencing**	**Minor errors** **Deduct 5% from overall percentage.** **Example: if the response receives 70%, deduct 5%. The final mark is 65%.**	**Major errors** **Deduct 10% from the overall percentage.** **Example: if the response receives 70%, deduct 10%. The final mark is 60%.**
Consistency The **correct referencing style** for the **discipline** – i.e., either **Harvard**, OR **APA** (for Psychology)**,** OR **Law**, OR **IEEE** (for ICT/Engineering) – has been used consistently for all in-text references and in the bibliography/reference list. **Concepts** and **ideas** that are **quoted** and/or **paraphrased** are referenced consistently throughout. **Position of the in-text reference**: an in-text reference is positioned consistently where appropriate for every quote and paraphrase.	**Minor inconsistencies:** • The referencing style used is generally consistent with what is required, but there are one or two changes/errors in the format of in-text referencing and/or in the bibliography/reference list. • For example, page numbers for direct quotes in-text have been provided for one source, but not in another. Or, two book chapters in the bibliography/reference list have been referenced in two different formats. Or, the publication year has been placed after the author name in one bibliography/reference list entry, and after the source title in another, etc. • Concepts and ideas in quotes and/or paraphrases are typically referenced, but a full in-text reference is missing or incomplete from one or two small sections of the work. • Position of the references: in-text references are only given at the beginning and/or end of every paragraph.	**Major inconsistencies:** • Poor and wholly inconsistent referencing style used in-text and/or in the bibliography/reference list. • Multiple referencing styles for the same source types have been used. • For example, the format for direct quotes in-text and/or book chapters in the bibliography/reference list and/or year of publication in the bibliography/reference list is different across multiple instances. • Concepts and ideas in quotes and/or paraphrases are haphazardly referenced in-text. • Position of the references: in-text references are only given at the beginning or end of large sections of work.
Feedback on referencing consistency:		
Congruency • **Each source** reflected within **in-text references** is included accurately in the **bibliography/reference list.** • All **bibliography/reference list** entries are in the **required order** for the referencing style used (e.g. alphabetical, alphabetical under subheadings, numerical). • All **direct quotes** and **paraphrases** have been **integrated appropriately** into the text using introductory phrases, accurate grammar, etc.	**Minor incongruences:** • There is largely a match between the sources presented in-text and those in the bibliography/reference list, but one or two sources that appear in-text do not appear in the bibliography/reference list, or vice versa. Or key source information is missing from one or two in-text references or bibliography/reference list entries only (e.g. publication year, city of publication, URL date accessed, etc.). • There is a clear and largely accurate ordering of sources in the bibliography/reference list as required by the referencing style used, but with one or two references out of order. • An attempt has been made for source integration into the text using appropriate introductory phrases and grammar, but one or two quotes or paraphrases do not flow as clearly or logically within the sentence structure as they could.	**Major incongruences:** • No relationship/several incongruencies between the in-text referencing and the bibliography/reference list. • For example, multiple sources are included in-text, but not in the bibliography, and/or vice versa. Key source information is missing from multiple in-text references and/or reference list entries. A URL link, rather than the actual reference, is provided in the bibliography. Sources are repeated in the reference list, etc. • Most sources are listed in a haphazard order throughout the bibliography/reference list. • Few to no appropriate introductory phrases or rules of grammar have been applied, and many direct quotes and/or paraphrases feel disconnected from the flow of the text.
Feedback on referencing congruency:		
Overall feedback on referencing, with suggested improvements:		

c#
App used my resdients talk to goverment about issue
login save issue report all issue and anlayse those
issues

Portfolio of Evidence (PoE) — Background

In the PoE project, you will develop a C# .NET Framework software application to streamline municipal services in South Africa. The application aims to provide an efficient and user-friendly platform for citizens to access and request various municipal services.

Scenario:

A South African municipality is seeking to improve citizen engagement and service delivery through the implementation of a comprehensive municipal services application. The application should enable residents to:

- Report issues and request services.
- Access information about local events and announcements.
- Receive updates on the status of their service requests.

Note to Students:

Ensure that your application is fully functional and meets all outlined specifications. Additionally, consider the objectives outlined above as key indicators of the success of your Municipal Services Application. Aim to create a user-centric experience that adds value to the lives of citizens in your municipality.

user centred experince
easy of use

Instructions

Complete the parts below to provide the required software. A list of items to be submitted for each part is specified – make sure you submit everything required!

Part 1 — Municipal Services Application for South Africa (Report Issues) (Marks: 100)

Learning Units: LU1 – LU2

This part has two tasks – **Research** (20 marks) and **Implementation** (80 marks).

Task 1: RESEARCH (20 Marks)

The municipality is interested in incorporating user engagement features into the application. Conduct online research on user engagement strategies suitable for a municipal services application, especially in the South African context. Refer to the following article to start your research:

Hart, Tim G. B., et al. "Innovation for Development in South Africa: Experiences with Basic Service Technologies in Distressed Municipalities." Forum for Development Studies, vol. 47, no. 1, 20 Aug. 2019,pp. 2347.
Hart_etal_FDS_2020_Innovation_for_development_in_South_Africa_experiences_with_basic_service.pdf. [Accessed 20 February 2025]

In a Word document:

- List five user engagement strategies considered during your research.
- Provide a 500-word explanation of the chosen user engagement strategy and justify why it was selected. The line spacing on the page should be 1.5. The font should be Times New Roman or Arial fonts. This should be at an 11 or 12-point size for readability.

Remember to reference the sources used.

Note: If the explanation exceeds 500 words, any content beyond this point will not be marked.

Task 2: IMPLEMENTATION (.NET Framework Window Application) (80 Marks)

The municipality requires a C# software application to facilitate citizen reporting of issues and service requests. The application should be user-friendly and provide a seamless experience for residents to engage with municipal services.

Requirements:

1. On startup, the application shall present the user with three tasks:
    a. **Report Issues (to be implemented).**
    b. Local Events and Announcements (to be implemented later).
    c. Service Request Status (to be implemented later).
2. Only the "Report Issues" task will be implemented initially; disable the other two options.
3. Upon selecting "Report Issues," the application shall prompt users to provide details about the issue, including location and category.
4. Users should be able to attach images or documents related to the issue.
5. Implement the chosen user engagement strategy to encourage active participation.

Technical Requirements:

- Utilise appropriate data structures to store user-reported issues and relevant details.
- Create a readme file explaining how to compile, run, and use the programme.

Guidelines for Report Issues Functionalities

User Interface Specifications:

1. **Main Menu (Form):**
    - The main menu should be presented upon startup, providing the following options:
        a. Report Issues (to be implemented).
        b. Local Events and Announcements (to be implemented later).
        c. Service Request Status (to be implemented later).
2. **Report Issues Page (Windows Form):**
    - After selecting "Report Issues," create a new Windows Form that includes the following elements:

a. **Location Input (Textbox):** A textbox for users to input the location of the reported issue.
b. **Category Selection (Dropdown or ListBox):** A dropdown or list for users to select the category of the reported issue (e.g., sanitation, roads, utilities).
c. **Description Box (RichTextBox):** A RichTextBox control allowing users to provide a detailed description of the issue.
d. **Media Attachment (Button for File Dialog):** A button enabling users to attach images or documents related to the reported issue. Implement OpenFileDialog for efficient media attachment.
e. **Submit Button (Button):** A clearly labelled "Submit" button that users click to finalise the report.
f. **Engagement Feature (Label or ProgressBar):** Integrate a dynamic engagement feature, such as a label displaying encouraging messages or a ProgressBar indicating the progress of the reporting.
g. **Navigation Buttons (Button):** Include navigation buttons (e.g., "Back to Main Menu") for users to easily return to the main menu or navigate to other sections of the application.

Design Considerations:

1. **Consistency:**
   - Maintain a consistent colour scheme and layout throughout the application to enhance user familiarity.
2. **Clarity:**
   - Ensure that labels, buttons, and instructions are clear and easily understood by a diverse user base.
3. **User Feedback:**
   - Implement feedback mechanisms (e.g., MessageBox for success messages, error alerts) to keep users informed about the status of their reporting.
4. **Responsiveness:**
   - Design the interface to be responsive, accommodating various screen sizes and resolutions.

Additional Requirements:

1. **Form Interactions (Event Handling):**
   - Implement event handlers for button clicks and user interactions to ensure seamless functionality.
2. **Data Handling (Data Structures):**
   - Utilise appropriate data structures (e.g., list for storing issues) to efficiently manage and organise the reported issues.

Note: If the code does not **compile** and **run**, no marks will be awarded for any application functionality.

Submit the following items for this part:

1. A **Word document** containing your **research**.
2. **Source code** for the application.
3. The **readme file** with instructions for how to compile, run, and use the software.

Important! You will build on this application in Part 2 and the PoE. So, keep a copy of your code in a safe place!

Total: 100

Part 2 — Municipal Services Application for South Africa (Collaboration) (Marks: 100)

Learning Units: LU1 – LU4

Introduction:

In Part 2, you will continue developing the Municipal Services Application for South Africa. The focus is on advanced data structures and algorithms, including stacks, queues, priority queues, hash tables, dictionaries, sorted dictionaries, sets, and an additional recommendation feature based on user searches.

Scenario:

The Municipal Services Application aims to be a comprehensive platform, integrating various features for citizens to access local events and announcements efficiently.

Task 1: Implementation (100 Marks)

2.1. Develop a C# application:

a. **Main Menu (Form)** (30 Marks)

- Implement a Windows Form with an organised menu presenting the following options:
- Report Issues (implemented in Part 1).
- **Local Events and Announcements (to be implemented in this part).**
- Service Request Status (to be implemented in Task 3).

b. **Local Events and Announcements Page (Windows Form)** (70 Marks)

- Upon selecting "Local Events and Announcements," create a Windows Form with the following features:
- Display upcoming local events and announcements in an aesthetically pleasing manner.
- Implement a search functionality allowing users to efficiently find events based on categories and dates.
- Utilise advanced data structures, such as sorted dictionaries, to optimise event organisation.

Technical Requirements for Local Events and Announcements Page **(40 Marks)**

Mark allocation breakdown:

Stacks, Queues, Priority Queues (15 Marks):

- Implement stacks, queues, or priority queues as needed to manage event-related data structures effectively.

Hash Tables, Dictionaries, Sorted Dictionaries (15 Marks):

- Utilise hash tables, dictionaries, or sorted dictionaries for organising and retrieving event information.

Sets (10 Marks):

- Incorporate sets to handle unique categories or dates efficiently.

Additional Recommendation Feature (30 Marks):

Implement a recommendation feature based on user searches:

- Analyse user search patterns and preferences.
- Use an appropriate algorithm or data structure to suggest related or recommended events.
- Present recommendations in a user-friendly manner within the application.

Note: If the code does not **compile** and **run**, no marks will be awarded for any application functionality.

Submit the following items for this part:

1. **Source code** for the application.
2. The **readme file** with instructions for how to compile, run, and use the software.

Important! You will build on this application in the PoE. So, keep a copy of your code in a safe place!

PoE — Municipal Services Application for South Africa (Full Functioning App) (Marks: 100)

Learning Units: All Learning Units

Introduction:

Task 3 focuses on the final implementation of the Municipal Services Application, emphasising the integration of advanced data structures and algorithms, including basic trees, binary trees, binary search trees, AVL trees, red-black trees, heaps, graphs, graph traversal, and minimum spanning trees.

Scenario:

The Municipal Services Application is designed to be a comprehensive platform for residents, encompassing features such as reporting issues, accessing local events, and tracking service requests.

Implementation (100 Marks)

3.1. Develop a C# application:

a. Implement a Windows Form that presents users with a menu for selecting:
   - Report Issues (implemented in Part 1).
   - Local Events and Announcements (implemented in Part 2).
   - Service Request Status (to be implemented in this task).

b. Service Request Status Page (Windows Form) (100 Marks):

When choosing "Service Request Status," create a Windows Form with the following features:

- Display a well-organised list of submitted service requests, including their status.
- Allow users to track the progress of their service requests using unique identifiers.
- **Utilise advanced data structures such as graphs, binary search trees, or heaps to manage and display service request information efficiently.**

Technical Requirements (50 Marks):

Basic Trees, Binary Trees, Binary Search Trees, AVL Trees, Red-Black Trees (20 Marks):

- Implement these tree structures effectively for organising and retrieving service request information.

Heaps, Graphs, Graph Traversal, Minimum Spanning Tree (30 Marks):

- Utilise these structures to manage complex relationships and optimise the display of service request status.

Implementation Report (20 Marks):

- Compile a detailed readme file explaining how to compile, run, and use the programme.
- For each implemented data structure, provide an in-depth explanation of its role and contribution to the efficiency of the "Service Request Status" feature, including relevant examples.

Project Completion Report (20 Marks):

- Write a comprehensive report detailing the completion of the entire project.
- Discuss the challenges faced during the implementation of Task 3 and how they were overcome.
- Share insights into the key learnings acquired throughout the project, including new skills, problem-solving approaches, and programming techniques.

Technology Recommendations (10 Marks):

- Suggest additional technologies or tools that could enhance the functionality or performance of the Municipal Services Application.
- Justify the recommendations based on potential benefits and compatibility with the project.

Note: If the code does not **compile** and **run**, no marks will be awarded for any application functionality.

Submit the following items for this part:

1. A **Word document** containing the report.
2. **Source code** for the application, which must include the **complete code of the functioning application**.
3. The **readme file** with instructions for how to compile, run, and use the software.
4. A file listing the **updates** that you have made based on **feedback** from your lecturer.

Appendix A - PoE Marking Rubrics

Assessment Sheet (Marking Rubric)

Please note: Tear off this section and **attach** it to your work when you submit it/ If this is an online submission, then this information needs to be included in the online submission.

MODULE NAME:	MODULE CODE:
PROGRAMMING 3B	PROG7312/w
ADVANCED APPLICATION DEVELOPMENT	AAPD7112/w

STUDENT NAME:
STUDENT NUMBER:

PART 1 -Task 1					
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Research: List of five user engagement strategies **[5 Marks]**	• No user engagement strategies are listed or are completely unrelated.	• Only one or two user engagement strategies are listed, with limited relevance to municipal services applications.	• Three to four user engagement strategies are listed, demonstrating a good understanding of the topic.	• The list includes five well-defined user engagement strategies relevant to municipal services applications.	
	0 Mark	**1 - 2 Marks**	**3 – 4 Marks**	**5 Marks**	

Research:	Does not meet the required standard	Meets the required standard	Partially exceeds the required standard	Greatly exceeds the required standard	Feedback
Explanation and justification of the chosen strategy **[10 Marks]**	• No explanation or justification is provided, or it is completely illogical.	• Some details are provided, but the explanation and justification lack depth or logical coherence.	• A 500-word explanation is included, providing a good level of detail and justification for the chosen user engagement strategy.	• A comprehensive 500-word explanation is provided, clearly justifying the chosen user engagement strategy with a deep understanding of its benefits for municipal services.	
	0 – 3 Marks	**4 – 6 Marks**	**7 – 8 Marks**	**9 – 10 Marks**	
Referencing and Citations **[5 Marks]**	• No proper referencing is provided.	• Referencing is present but lacks accuracy or proper citation format.	• References are mostly accurate, with minor issues in citation format.	• Proper referencing and citations are used, following the given article and other relevant sources.	
	0 Mark	**1 - 2 Marks**	**3 – 4 Marks**	**5 Marks**	

PART 1 -Task 2					
	Does not meet the required standard	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
App Functionality: **Task presentation on startup**	• The main menu is not implemented, or it does not work at all.	• The main menu is implemented, but there are notable bugs affecting user experience.	• The main menu is well-implemented, with minor issues or bugs that do not significantly impact functionality.	• The main menu is presented flawlessly on startup, with all options working perfectly without any errors.	
[10 Marks]	**0 – 3 Marks**	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 -Task 2					
App Functionality: **Report Issues task implementation.**	• App Functionality: Report Issues task implementation	• The "Report Issues" task is fully implemented, meeting all requirements without any errors.	• The "Report Issues" task is well-implemented with only minor bugs that do not hinder functionality.	• The "Report Issues" task is fully implemented, meeting all requirements without any errors.	
[10 Marks]	**0 – 3 Marks**	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	

PART 1 -Task 2					
App Functionality: User input for issue details **[10 Marks]**	• User input functionality is not implemented or does not work at all.	• User input functionality is implemented, but there are notable issues affecting user interaction.	• User input functionality is well-implemented with only one or two minor bugs.	• User input functionality for issue details works perfectly without any errors.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 -Task 2					
App Functionality: **Media attachment functionality** **[10 Marks]**	• The media attachment feature is not implemented, or it does not work at all.	• The media attachment feature is implemented, but there are significant bugs affecting usability.	• The media attachment feature is well-implemented with only one or two minor bugs.	• The media attachment feature works flawlessly without any errors.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 -Task 2					
App Functionality: **Implementation of user engagement strategy**	• The user engagement strategy is not implemented, or it does not work at all.	• The user engagement strategy is implemented, but there are notable issues impacting its success.	• The user engagement strategy is well-implemented, with minor issues that do not hinder its effectiveness.	• The chosen user engagement strategy is seamlessly integrated, positively influencing user participation.	

[10 Marks]	0 – 3 Marks	4 - 6 Marks	7 - 8 Marks	9 - 10 Marks	
PART 1 -Task 2					
App Logic: **Use of appropriate data structures** **[5 Marks]**	• A list is not used at all to store user-reported issues.	• A list is used only in some places, with arrays or different data structures being used in others, affecting efficiency.	• A list is mostly used, with some instances of other data structures, but it does not significantly impact functionality.	• A list is consistently and appropriately used throughout the application to store user-reported issues.	
	0 Mark	**1 - 2 Marks**	**3 - 4 Marks**	**5 Marks**	
PART 1 -Task 2					
Coding Standards: **Readme file quality** **[5 Marks]**	• No readme file is submitted.	• The readme file contains very little useful information, making it challenging to understand how to use the software.	• The readme file contains sufficient information but may lack completeness or detail in some areas.	• The readme file is excellent, providing all relevant information for compiling, running, and using the software.	
	0 Mark	**1 - 2 Marks**	**3 - 4 Marks**	**5 Marks**	
PART 1 -Task 2					
Design Considerations: **Consistency, clarity, user feedback, and responsiveness**	• The interface is poorly designed, with significant inconsistencies, unclear labels, and no effective	• The interface lacks consistency, clarity, and effective feedback mechanisms,	• The interface is mostly consistent, with minor inconsistencies in color or layout. • Labels and instructions are generally clear, but	• The interface maintains a consistent color scheme and layout, enhancing user familiarity. • Labels, buttons, and instructions are	

[10 Marks]	feedback mechanisms. • It is not responsive, making it challenging for users with various screen sizes.	impacting user understanding. • Responsiveness is limited, affecting user experience on different screens.	some users may find them confusing. • Feedback mechanisms are present but may need improvement. • The interface is responsive but may have issues with certain screen sizes.	clear and easily understood. • Feedback mechanisms are implemented effectively, keeping users informed. • The interface is responsive, accommodating various screen sizes.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 -Task 2					
Additional Requirements: **Form interactions and data handling.** **[10 Marks]**	• Event handlers are not implemented, or they do not work, making the application non-functional. • Inappropriate data structures are used or not used at all.	• Event handlers are implemented, but there are notable issues affecting functionality. • Data structures are not used efficiently, impacting the organisation of user-reported issues.	• Event handlers work well, with minor issues that do not significantly impact functionality. • Data structures are mostly appropriate but may need refinement.	• Event handlers for button clicks and user interactions are implemented seamlessly, ensuring flawless functionality. • Appropriate data structures (e.g., List) are used efficiently to manage and organise user-reported issues.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 TOTAL					**/100**

Notes to Students:

PART 2 -Task 1					
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Main Menu (Form) **[30 Marks]**	• The Main Menu is not implemented, or it does not work at all.	• The Main Menu is implemented, but there are notable bugs affecting user interaction.	• The Main Menu is well-implemented with minor issues that do not significantly impact functionality.	• The Main Menu is flawlessly implemented with organised options, and all features work perfectly without any errors.	
	0 - 8 Mark	**9 - 16 Marks**	**17 – 20 Marks**	**21 - 30 Marks**	

Marking Criteria	Does not meet the required standard	Meets the required standard	Partially exceeds the required standard	Greatly exceeds the required standard	Feedback
Local Events and Announcements Page (Windows Form): Technical Requirements	• Stacks, queues, or priority queues are not implemented or do not work correctly.	• Stacks, queues, or priority queues are implemented, but there are significant problems affecting functionality.	• Stacks, queues, or priority queues are well-implemented but may have minor issues.	• The implementation effectively utilises stacks, queues, or priority queues for managing event-related data structures.	
Stacks, Queues, Priority Queues **[15 Marks]**	**0 – 4 Marks**	**5 - 10 Marks**	**11 - 14 Marks**	**15 Marks**	
Local Events and Announcements Page (Windows Form): Technical Requirements	• Hash tables, dictionaries, or sorted dictionaries are not implemented or do not work correctly.	• Hash tables, dictionaries, or sorted dictionaries are implemented, but there are notable issues.	• The use of hash tables, dictionaries, or sorted dictionaries is good but may have minor inefficiencies.	• Hash tables, dictionaries, or sorted dictionaries are seamlessly integrated for organising and retrieving event information.	
Hash Tables, Dictionaries, Sorted Dictionaries [15 Marks]	**0 – 4 Marks**	**5 - 10 Marks**	**11 - 14 Marks**	**15 Marks**	

PART 2 -Task 1					
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Local Events and Announcements Page (Windows Form): **Technical Requirements:** **Sets** **[10 Marks]**	• Sets are not implemented or do not work correctly. **0 – 3 Marks**	• Sets are implemented, but there are notable problems. **4 - 6 Marks**	• The use of sets is good but may have minor issues affecting efficiency. **7 - 8 Marks**	• Sets are effectively incorporated to handle unique categories or dates efficiently. **9 - 10 Marks**	
	Does not meet the required standard	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Additional Requirements: **Search patterns, Smart Recommendations** **[30 Marks]**	• The recommendation feature is not implemented or does not work correctly.	• The recommendation feature is implemented, but there are significant problems affecting the accuracy of suggestions or presentation.	• The recommendation feature is well-implemented but may have minor issues in analysing user preferences or presenting recommendations.	• The recommendation feature is seamlessly integrated, analysing user search patterns and preferences. • An appropriate algorithm or data structure is used to suggest related or	

				recommended events. • Recommendations are presented in a user-friendly manner within the application.	
	0 – 9 Marks	**10 - 15 Marks**	**16 - 20 Marks**	**21 - 30 Marks**	
PART 2 TOTAL					**/100**
Notes to Students:					

POE PART 3 -Task 1					
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Basic Trees, Binary Trees, Binary Search Trees, AVL Trees, Red-Black Trees: **Implementation Effectiveness**	• The tree structures are not implemented or do not work correctly, leading to significant issues in the organisation and retrieval of service request information.	• The implementation of tree structures is present but has notable bugs affecting the organisation and retrieval of service	• The tree structures are well-implemented, with minor issues that do not significantly impact functionality. • There may be a few areas for improvement but overall, a solid and	• The implementation of these tree structures is exceptional, providing an efficient organisation and retrieval mechanism for	

[20 Marks]		request information. • There might be areas that need attention to enhance efficiency.	effective implementation.	service request information. • It demonstrates a flawless integration, addressing potential issues effectively.	
	0 – 5 Marks	**6 - 10 Marks**	**11 – 15 Marks**	**16 - 20 Marks**	

Heaps, Graphs, Graph Traversal, Minimum Spanning Tree: Structures Utilisation [30 Marks]	Does not meet the required standard	Meets the required standard	Partially exceeds the required standard	Greatly exceeds the required standard	Feedback
	• These structures are not utilised or do not work correctly, resulting in significant issues in managing complex relationships and optimising the display of service request status.	• The utilisation of these structures is present, but there are notable issues affecting performance. • The implementation may lack some key elements for efficient management of complex relationships.	• These structures are well-utilised but may have minor issues affecting efficiency. • While the implementation is good, there might be some opportunities to enhance the utilisation of these structures for optimal performance.	• Heaps, graphs, graph traversal, and minimum spanning tree structures are seamlessly integrated, effectively managing complex relationships and optimising the display of service request status. • The implementation demonstrates a deep understanding of their role and efficient utilisation.	
	0 – 9 Marks	**10 - 15 Marks**	**16 - 20 Marks**	**21 - 30 Marks**	

Implementation Report: **Readme File Quality** **[10 Marks]**	• No readme file is submitted, leaving users without essential guidance on compiling, running, and using the software.	• The readme file contains very little useful information, making it challenging to understand how to use the software. • It significantly impacts the user's ability to interact with the application.	• The readme file contains sufficient information but may lack completeness or detail in some areas. • While functional, there might be some areas where additional clarity could improve user guidance.	• The readme file is detailed and clear, providing comprehensive instructions for compiling, running, and using the program. • It is well-organised and easily understandable, contributing to a seamless user experience.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	

POE PART 3 -Task 1					
	Does not meet the required standard	**Meets the required standard**	**Partially exceeds the required standard**	**Greatly exceeds the required standard**	**Feedback**
Implementation Report: **Data Structure Explanation** **[10 Marks]**	• No explanations are provided for the implemented data structures, leaving users without insights into the fundamental components of the application's efficiency.	• Explanations are limited, lacking depth and coherence. Examples are unclear or non-existent. • The understanding of the role of each data structure in enhancing efficiency is not effectively communicated.	• Explanations are present but may lack depth or completeness. Some examples may be missing or unclear. • While providing insights, there is room for improvement in conveying the full impact of each data structure on application efficiency.	• In-depth explanations are provided for each implemented data structure, detailing its role and contribution to the efficiency of the "Service Request Status" feature, with relevant examples. • The explanations are clear, detailed, and effectively communicate the significance of each data structure.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	

POE PART 3 -Task 1					
Project Completion Report: **Project Overview** **[10 Marks]**	• No project overview is provided, depriving users of essential insights into the challenges faced and solutions implemented during the project.	• The project overview is limited, lacking detail, and insights into challenges and solutions. • Users are left with a less comprehensive understanding of the project's journey and problem-solving approaches.	• The project overview is present but may lack detail or insights. Challenges and solutions are briefly mentioned, leaving some aspects of the project's completion not fully explored.	• A comprehensive report details the completion of the entire project, providing insights into challenges faced during the implementation of Task 1, 2 and 3 and how they were overcome. • The overview effectively communicates the project's journey, challenges, and solutions.	
	0 – 10 Marks	**11 - 20 Marks**	**21 -26 Marks**	**27 - 30 Marks**	
POE PART 3 -Task 1					
Project Completion Report: **Key Learnings** **[5 Marks]**	• No key learnings are provided, leaving users without insights into the valuable skills and knowledge gained during the project.	• Key learnings are mentioned but lack detail or specificity. • The discussion provides only a surface-level understanding of the	• Some insights into key learnings are provided but lack depth or clarity. • The discussion could benefit from further elaboration on specific skills,	• Significant insights into key learnings acquired throughout the project, including new skills,	

		learning outcomes from the project.	approaches, or techniques learned.	problem-solving approaches, and programming techniques. • The discussion reflects a deep understanding of the learning process during the project.	
	0 Mark	**1 - 2 Marks**	**3 - 4 Marks**	**5 Marks**	
POE PART 3 -Task 1					
Technology Recommendations: **Suggestions** **[5 Marks]**	• No technology recommendations are provided, missing an opportunity to enhance the application's capabilities.	• Recommendations are limited and lack clear justifications. • The suggested technologies may not provide substantial contributions to the application's functionality or performance.	• Recommendations are present but may lack clarity or justification. • The suggested technologies could benefit from more explicit ties to potential benefits and compatibility with the project.	• Additional technologies or tools are suggested to enhance the functionality or performance of the Municipal Services Application, with clear justifications based on potential benefits and	

				compatibility with the project. • The recommendations are insightful and directly contribute to the application's enhancement.	
	0 Mark	**1 - 2 Marks**	**3 - 4 Marks**	**5 Marks**	
POE PART 3 -Task 1					
Technology Recommendations: **Justification** **[5 Marks]**	• No justifications are provided for the technology recommendations, leaving users without insights into why these technologies are suggested.	• Justifications are unclear or not directly tied to project benefits. • The discussion does not effectively convey the rationale behind the technology recommendations.	• Justifications are present but may lack clarity or may not be directly tied to project benefits. • While providing some rationale, there is room for improvement in clearly connecting each recommendation to the project's needs.	• The justifications for technology recommendations are clear and directly tied to potential benefits and compatibility with the project. • The discussion effectively communicates why each technology is a valuable addition.	
	0 Mark	**1 - 2 Marks**	**3 - 4 Marks**	**5 Marks**	

POE PART 3 -Task 1					
Updates Based on Feedback: **Incorporation of Feedback** **[5 Marks]**	• No file listing updates based on feedback is submitted, missing the opportunity to showcase the application's iterative improvement process.	• The file listing updates is limited, and there is minimal evidence of substantial changes made based on feedback. • The updates may not fully address the provided feedback.	• A file listing updates is provided, but there might be areas where the incorporation of feedback could be more detailed or explicit. • The updates contribute to improvements but may lack thorough documentation.	• A detailed file listing the updates made based on feedback from the lecturer is submitted. • The updates reflect a proactive approach to refining and enhancing the application, addressing feedback effectively.	
	0 – 3 Marks	**4 - 6 Marks**	**7 - 8 Marks**	**9 - 10 Marks**	
PART 1 TOTAL					**/100**
Notes to Students:					