RICHARD JOHNSON
ANGLICAN COLLEGE
93 Hyatts Road Oakhurst NSW 2761 | 02 9677 2455

Assessment Notification

Year: 12 Subject: Software Engineering Task 3

Task title	**Software Engineering Project**		
Date issued	Tuesday, 20 May 2025	Weighting	30%
Due date	Friday, 27 June 2025	Total marks	50

Outcomes:

- **SE-12-01** justifies methods used to plan, develop and engineer software solutions
- **SE-12-07** designs, develops and implements safe and secure programming solutions
- **SE-12-06** justifies the selection and use of tools and resources to design, develop, manage and evaluate software
- **SE-12-09** applies methods to manage and document the development of a software project

Context:

In this task, you will develop a web-based software engineering project as part of the Software Engineering Project unit. This project allows you to demonstrate mastery of subject knowledge and skills over a focused period. Your project should focus on delivering a Minimum Viable Product (MVP) that clearly addresses the core identified need. A well-executed, smaller-scope project is preferred over an overly ambitious, incomplete one.

You will integrate content from the focus areas of Secure Software Architecture, Programming for the Web, and, optionally, Software Automation to create a functional, secure, and optimised web application that addresses an identified need or opportunity, using Python as the programming language. Throughout the process, you will develop project management, collaboration, and communication skills—highly valued in the industry. Your work will be documented in a project folio, providing evidence of your contribution and development journey.

Description of the task:

PART A: Project Technical Report (12 marks)

Provide a technical report (800–1000 words) written for a software engineering team to reference key information about your work. The report must demonstrate that your Python-based web application addresses an identified need or opportunity and that all stages were conducted with due diligence. Use plain English with technical detail and accuracy. Structure the report with the following headings and subheadings (not included in the word count) and include these deliverables:

• A1. Identifying and Defining (6 marks, 400–500 words):

- **Problem Statement and Feasibility Assessment:** Outline the need or opportunity (e.g., "This project provides a simple tool for students to track homework completion"). Assess scheduling feasibility within the 10-week timeline and resource feasibility (e.g., "Uses free tools like Flask, SQLite, GitHub, PythonAnywhere").

- **Requirements List and Technical Specifications**: List key functional (MVP features) and non-functional requirements (e.g., "User registration, task entry/view, secure login, responsive UI"). Define core data structures, data types (e.g., "SQLite tables with user/task data"), and boundaries (e.g., "Web application only, no mobile app").

• A2. Research and Planning (4 marks, 300–400 words):

- **Planning Justification:** Justify the planning method chosen and how it suits the 10-week timeframe (e.g., "Modified Agile with weekly sprints for rapid feedback").

- **Communication Summary:** Describe how you involved the client/stakeholder (even if simulated), enabled feedback, and managed scope (e.g., "Two documented feedback sessions, tracked changes in log book").

A3. Testing and Evaluating (2 marks, 200–300 words):

- **Security Measures:** Explain the key security measures implemented (e.g., "Used bcrypt for passwords, sanitised user inputs").

- **Testing Methods:** Describe the main testing approaches (e.g., "Manual testing of core features, basic unit tests for critical functions using black-box, white-box, grey-box testing methods.").

PART B: Project Development Record (12 marks)

Provide a completed project development record under the heading "Project Documentation" with these deliverables:

• B1. Project Documentation (6 marks):

- **B1A. Log Book (2 marks):** Include detailed daily or weekly entries outlining specific progress, challenges encountered, bugs fixed, design decisions (and rationale), resources used (including any AI assistance), and reflections (minimum 10 substantial entries). This is critical evidence of your process.

- **B1B. Client Correspondence (2 marks):** Provide evidence (e.g., emails, meeting notes) of minimum 2 well-documented interactions showing substantive feedback and your response/action.

- **B1C. Gantt Chart (2 marks):** Include a Gantt chart outlining key phases, milestones, and deadlines within the 10-week timeframe.

B2. Modelling Tools (6 marks):
Demonstrate use of at least **THREE** modelling tools relevant to your project, chosen from the list below. Briefly justify your choices.

- **Options A (Choose one - 2 marks):**
  - Data Flow Diagram (DFD)
  - Class Diagram
- **Options B (Choose one - 2 marks):**
  - Structure Chart
  - Decision Tree/Table (if relevant to logic)
  - Flowchart
  - Storyboard
  - Wiring Diagram (for mechatronic projects)
- **Options C (Choose one additional, distinct tool - 2 marks):**
  - Another choice from Options A or Options B (must be different from your first selections)

PART C: Complete Software Solution (20 marks)

Develop and document a web-based software solution under the heading "Producing and Implementing" with these deliverables:

• C1. Development and Deployment (14 marks):

- **C1A. Core MVP Functionality (5 marks):**
  Evaluates if all key MVP features are fully implemented and operate correctly. This includes demonstrating accurate Python logic and appropriate data handling to meet the application's main purpose as defined in your requirements.
- **C1B. Web Framework & Backend Integration (5 marks):**
  Assesses the effective use of the web framework (e.g., Flask) for crucial web operations like URL routing, handling user requests, and dynamic page generation (templating). If a database is used, this includes its proper integration for data storage and retrieval.
- **C1C. Code Quality & Readability (4 marks):**
  Judges the professionalism of your code. This means assessing adherence to coding style guidelines (PEP 8), the clarity and usefulness of comments, logical code organisation, and overall ease of understanding and maintainability.

• C2. Security and deployment (4 marks):

- **Security Features:** Document and include at least two significant, well-implemented security features relevant to your application (e.g., using bcrypt for passwords, implementing input validation/sanitisation).

- **GitHub Repository:** Submit via your own GitHub repository with a commit history showing incremental development and meaningful commit messages (minimum 10 commits). This is critical evidence of your process. Include requirements.txt and README.md files.

- **Deployed Solution:** Deploy the application on a platform like PythonAnywhere or Netlify (or similar approved service) with a live URL and clear access/login instructions provided.

- **C3. UI/UX Description (2 marks, 150–250 words):**
    - UI/UX Design and Considerations: Briefly detail the UI design approach and key components (e.g., “Simple interface using Bootstrap, includes navigation, forms, and data display”). Briefly explain key UX considerations (e.g., “Clear navigation, straightforward workflow for core task”).

PART D: Project Presentation (6 marks)

- **D1. Video or Poster Submission (4 marks):**
    - Provide a 4-5 minute video or 2 A3-sized posters summarising your project. Include:
        - Problem/Opportunity: Briefly describe the need addressed.
        - Target Audience: Identify the primary users.
        - Tech Stack (Tools/Technologies): List key tools used (Python, HTML, CSS, js, etc.).
        - Process Timeline: Summarise key stages/dates from your gantt chart.
        - Key Features: Highlight two or three main features of your MVP.
        - Screenshots, UI mock-ups and modelling tools
        - Include appropriate visuals and graphics
        - Do not include inappropriate imagery, copyrighted music or visuals

- **D2. Digital and Printed Presentation (2 marks):**

Your document should have spelling and grammar check and be presented with a table of contents, header, footer, headings, subheadings, screenshots, tables and appropriate visuals. This must be uploaded as a single Word document and as a printed and bound document.

Marking criteria: Refer to the marking scaffold below. All written work must be the student's original creation. Failure to cite sources, including AI-generated content, will be considered academic malpractice and may result in penalties such as reduced marks, a zero grade, or further disciplinary action per the school's academic integrity policy.

Feedback provided: Throughout the unit, you will receive either written or verbal feedback. During discussions, the teacher will highlight your strengths and areas that require further development.

Submission Guidelines:

1. Submit in Canvas, a **single Word document** containing all required written tasks, a **zip file** containing all relevant Python code and project files. **Include a link your GitHub private repo in your Word document on the first page.**
2. Submit all code in your GitHub private repo, the zip file uploaded to Canvas.
3. **Any use of AI must be clearly and explicitly acknowledged and cited in the Log Book and Bibliography.**

Faculty Leader approval:

Students are reminded of the rules and requirements relating to completion, submission and absences for assessment tasks.	 _____ **Faculty Leader approval**

Year 12 Software Engineering: Software Engineering Project – Task 3 Marking Scaffold

CRITERIA	WORK ASPECT	RATINGS					
SE-12-01 justifies methods used to plan, develop and engineer software solutions **SE-12-06** justifies the selection and use of tools and resources to design, develop, manage and evaluate software **SE-12-09** applies methods to manage and document the development of a software project	**PART A: Project Technical Report (12 marks)**						
	A1. Identifying and Defining (6 marks)	**6** **EXTENSIVE** All four elements (Problem, Feasibility, Requirements, Specs) are exceptionally well-defined, specific, deeply interconnected, and demonstrate sophisticated understanding. The identified need is significant, feasibility is meticulously analysed, requirements are exhaustive and testable, and technical specifications are highly precise. Overall quality is outstanding.	**5** **THOROUGH** All four elements are clearly identified, well-explained, specific, and logically connected. The need is clear, feasibility is well-analysed, requirements are complete (MVP & non-functional), and technical specifications are clearly defined. Strong overall quality, though minor areas might benefit from further refinement.	**4** **SOUND** Most or all four elements are present and cover main points, but some aspects may be more general or less precise than Thorough. The problem is stated, feasibility considered, most key requirements listed (though some clarity/testability may vary), and basic technical specifications outlined. A competent understanding is evident, but greater depth or specificity is needed in some areas.	**3** **BASIC** Some elements are present but may be vaguely defined, lack detail, or show inconsistencies. The problem may be unclear, feasibility analysis minimal, requirements incomplete or unfocused (not clearly MVP), and technical specifications rudimentary. A foundational attempt, but significant gaps or weaknesses exist across several elements.	**2** **ELEMENTARY** One or more core elements are poorly defined, significantly incomplete, or largely missing. The problem is obscure, feasibility weak, requirements minimal/unclear, and technical specifications absent or unintelligible. Provides very limited evidence of project definition or understanding.	**1-0** **MINIMAL / NON-SUBMSSION** Non-attempt, or only cursory mention of one or two elements with little substance, clarity, or relevance. Essentially fails to define a software project.
	A2. Research and Planning (4 marks)		**4** **EXTENSIVE** The chosen planning method is clearly explained with specific	**3** **THOROUGH** The planning method is explained with good reasons for its suitability. The	**2** **SOUND** The planning method is identified and some reasons for its choice are	**1** **BASIC / ELEMENTARY** The planning method is named but justification is	**0** **NON-SUBMSSION** Not attempted

			reasons demonstrating its suitability for the 10-week project and how it guided project execution. The communication summary details at least two specific client/stakeholder interactions, showing what feedback was received and what actions were taken, supported by clear evidence. Scope changes are explicitly mentioned and tracked.	communication summary details at least two client/stakeholder interactions with feedback and actions taken, supported by evidence. Scope management is evident.	given. The communication summary describes client interactions and feedback. Evidence is present. Scope management might be generally described.	weak or missing. The communication summary is brief, lists interactions without much detail on feedback/action, or evidence is minimal/absent.	
	A3. Testing and Evaluating (2 marks)				**2**	**1**	**0**
					EXTENSIVE/ COMPLETE At least two specific security measures implemented in the project are clearly explained with rationale. At least two distinct testing methods used are described with specific examples of	**SOUND/PARTIAL** One or two security measures are mentioned, but the explanation of rationale or implementation is brief or general. One or two testing methods are named, but the description of their	**MINIMAL / NON-SUBMSSION** Not attempted, or explanations/descriptions are vague, names only, or missing.

					their application within the project.	application to the project is general or lacks specific examples.	
SE-12-01 justifies methods used to plan, develop and engineer software solutions **SE-12-09** applies methods to manage and document the development of a software project	**PART B: Project Development Record (12 marks)**						
	B1A. Log Book (2 marks)				**EXTENSIVE/ COMPLETE** The Log Book (well over 10 entries) consistently provides detailed, dated records of specific progress, significant challenges with solutions, well-reasoned design choices, comprehensively cited resources (including full AI details), and insightful reflections, forming a rich development narrative.	**SOUND/PARTIAL** The Log Book (at least 10 entries) covers progress, some challenges, decisions, and resources, with an attempt at AI citation and reflections. However, entries may lack consistent depth or detail in some areas, making the overall narrative adequate but less rich.	**MINIMAL/ NON-SUBMISSION** The Log Book (well under 10 entries, or entries are superficial) provides little meaningful insight into the development process, with key elements like challenges, decisions, detailed resource use (especially AI), or reflections largely absent or perfunctory. OR Not attempted.
	B1B. Client Correspondence (2 marks)				**EXTENSIVE/ COMPLETE** Clear, verifiable evidence (e.g., detailed emails, comprehensive meeting notes) of 2+ substantive interactions meticulously details	**SOUND/PARTIAL** Evidence of 2 interactions shows general feedback was sought and some action taken. However, the documentation may lack specific detail on feedback	**MINIMAL/ NON-SUBMISSION** Fewer than 2 interactions are documented, or evidence is minimal/superficial, lacks clarity on feedback, or fails to show resulting actions

					specific client feedback, the student's considered response, and clear, rational actions taken, demonstrating effective communication.	content, the student's rationale, or the explicitness of resulting actions for one or both interactions.	or consideration of stakeholder input. OR Not attempted.
	B1C. Gantt Chart (2 marks)				**EXTENSIVE/ COMPLETE** The detailed, accurate, and professionally presented Gantt chart clearly outlines all key project phases, specific tasks, milestones, and realistic timelines within the 10-week scope, ideally tracking actual progress against the plan effectively.	**SOUND/PARTIAL** The Gantt chart outlines main project phases, major tasks, and deadlines. It is generally clear but may lack some task detail, miss minor milestones, have less precise timelines, or a less refined presentation; progress tracking may be basic.	**MINIMAL/ NON-SUBMISSION** The Gantt chart is very basic, highly incomplete, poorly constructed, difficult to understand, or largely irrelevant to the project's scope or timeframe; key elements are absent. OR Not attempted.
	B2. Modelling Tools (6 marks)	**6**	**5**	**4**	**3**	**2**	**1-0**
		EXTENSIVE Three or more distinct modelling tools are produced accurately, are easy to understand, and clearly represent	**THOROUGH** Three or more modelling tools are produced accurately and clearly represent aspects of the project. The justification	**SOUND** Three modelling tools are presented and are mostly accurate, representing project aspects. Justification for choices is present. One tool or	**BASIC** One to three modelling tools are presented, but one or more may have noticeable inaccuracies or are difficult to understand.	**ELEMENTARY** One or two modelling tools attempted, with significant errors, or they don't clearly represent the project. Justification is weak or missing.	**MINIMAL/ NON-SUBMISSION** Non-submission or minimal attempt at one modelling tool, or tools are irrelevant/ unintelligible. Justification largely absent.

		specific aspects of the project's data, structure, or logic. The justification clearly explains why each specific tool was chosen for the particular aspect of the project it models.	explains why each tool was chosen and is relevant.	justification might be less clear or detailed.	Justification is brief or generic.		
SE-12-06 justifies the selection and use of tools and resources to design, develop, manage and evaluate software **SE-12-07** designs, develops and implements safe and secure programming solutions	**PART C: Complete Software Solution (20 marks)**						
	C1A. Core MVP Functionality (5 marks)	**5** **EXTENSIVE** All listed MVP features are fully working without errors. The Python code for these features is logical, efficient, and directly achieves the application's stated main purpose. Data is handled correctly for all operations.	**4** **THOROUGH** All listed MVP features are implemented and work correctly, with only very minor, non-critical bugs. Python code is logical and handles data well for the app's purpose.	**3** **SOUND** Most listed MVP features work, though some may have noticeable bugs or limitations that don't prevent core use. Python logic is generally sound; data handling is adequate.	**2** **BASIC** Several MVP features are partially working or have significant bugs. Python logic has clear errors; data handling is inconsistent or incorrect..	**1** **ELEMENTARY** Very few MVP features work, or core logic is fundamentally broken. The application struggles to perform its main functions.	**0** **MINIMAL/ NON-SUBMISSION** Not attempted or code is non-functional.
	C1B. Web Framework & Backend Integration (5 marks)	**5** **EXTENSIVE** The web framework is used skilfully for all required web operations: URL routes are logical and work, user inputs are	**4** **THOROUGH** The web framework is used effectively for routing, request handling, and templating. Database integration (if	**3** **SOUND** The web framework is used for most web operations, though some might be simplistic or have minor issues. Database integration (if used) is functional	**2** **BASIC** Web framework use is basic; routing, request handling, or templating have noticeable problems or are incomplete. Database integration	**1** **ELEMENTARY** Minimal or incorrect use of web framework features. Backend integration is largely non-functional or absent.	**0** **NON-SUBMISSION** Not attempted or no framework use evident.

		correctly processed, and web pages are dynamically generated with data passed correctly. If a database is used, it is connected and all data storage/retrieval operations work flawlessly.	used) is correct and supports data operations well.	but might be inefficient or have minor errors.	(if used) is flawed or very limited.		
	C1C. Code Quality & Readability (4 marks)		**4** **EXTENSIVE** Python code consistently follows PEP 8 style guidelines. It is well-organised into logical functions/sections. Comments are frequent, clear, and explain the purpose or rationale of code sections. The code is very easy for someone else to read and understand.	**3** **THOROUGH** Python code largely follows PEP 8. It is organised. Comments are present and helpful. The code is generally easy to read.	**2** **SOUND** Python code shows some attempt at PEP 8 and organisation. Some comments are present. The code is understandable with some effort.	**1** **BASIC / ELEMENTARY** Python code has many PEP 8 violations, is poorly organised, or lacks useful comments. The code is difficult to read.	**0** **NON-SUBMSSION** Not attempted or code is disorganised and uncommented.
	C2. Security and deployment (4 marks):		**EXTENSIVE** Two or more significant security features are	**THOROUGH** Two significant security features are well-implemented	**SOUND** At least one significant security feature is	**BASIC / ELEMENTARY** One security feature attempted	**NON-SUBMSSION** Not attempted, or security features are absent, GitHub

			robustly implemented, function correctly, and are clearly documented with rationale. GitHub repository demonstrates an exemplary commit history (well over 10 meaningful, incremental commits), including a comprehensive README.md and accurate requirements.txt. The solution is flawlessly deployed with a live URL and very clear access instructions	and documented. GitHub repository has a strong commit history (10+ meaningful, incremental commits), with good README.md and requirements.txt. The solution is deployed effectively with a live URL and clear access instructions.	implemented and documented. GitHub repository meets minimum commit requirements with generally meaningful messages and necessary files. The solution is deployed with a live URL, *though access instructions OR stability might have minor issues*	but poorly implemented. GitHub repository has major omissions. Deployed solution is problematic *OR not attempted*	repository not used or empty, and no deployment attempted.
	C3. UI/UX Description (2 marks)				**EXTENSIVE/ COMPLETE** The description (150-250 words) clearly and comprehensively details the UI design approach, key UI components, and provides insightful, specific UX considerations	**SOUND/PARTIAL** The description addresses most aspects of UI/UX but may be brief in some areas (e.g., UX considerations less detailed or generic) or slightly outside the word count. Key UI components are listed, but the	**MINIMAL / NON-SUBMSSION** Not attempted, or the description is significantly below word count, vague, largely irrelevant, or fails to address key aspects of UI design or UX considerations.

					directly relevant to the project's functionality and target audience. The writing is concise and articulate.	rationale for design or UX choices could be clearer or more specific.	
SE-12-01 justifies methods used to plan, develop and engineer software solutions **SE-12-09** applies methods to manage and document the development of a software project	**PART D: Project Presentation (6 marks)**						
	D1. Video or Poster Submission (4 marks)		**ENTENSIVE** The video (4-5 mins) or posters (2 x A3) are highly engaging, clear, concise, and professionally produced. All required content elements (Problem, Audience, Tech Stack, Timeline, Key Features, Visuals of project/models) are thoroughly covered, well-explained, and effectively illustrated. Visuals are excellent and significantly enhance understanding. Adheres perfectly to all constraints (e.g., no	**THOROUGH** The video or posters are clear, well-organized, and effectively communicate the project. All required content elements are present and adequately covered with good explanations/illustrations. Visuals are good and support understanding. Adheres to all constraints. Timing/size is appropriate.	**SOUND** The video or posters cover most required content elements, though some may be brief, less clear, or superficially explained. Visuals are adequate but may not always enhance understanding significantly. May have minor deviations from constraints or ideal timing/size.	**BASIC / ELEMENTARY** The video or posters are present but miss several key content elements, are unclear, poorly organized, or lack sufficient detail. Visuals are sparse, low quality, or not effective. May have notable issues with constraints, timing/size, or overall presentation quality.	**NON-SUBMSSION** Not attempted, or submission is fundamentally flawed, missing most content, or disregards major constraints.

			copyrighted material).				
	D2. Digital and Printed Presentation (2 marks)				**EXTENSIVE/ COMPLETE** The complete document (digital Word file and printed/bound copy) is professionally presented with flawless spelling and grammar. All specified formatting requirements (Table of Contents, header, footer, headings, subheadings) are expertly and consistently implemented. Screenshots, tables, and other visuals are high quality, appropriately integrated, and clearly labelled, enhancing the document.	**SOUND/PARTIAL** The document is generally well-presented but may contain minor spelling/grammar errors or slight inconsistencies in formatting. Most specified presentation elements are present and largely correct. Visuals are included and generally appropriate. Submitted in both required formats.	**MINIMAL / NON-SUBMSSION** Not attempted, or the document has significant and frequent errors in spelling/grammar, lacks multiple key presentation elements, is poorly structured, or is not submitted in the required formats. Visuals are absent or poorly integrated.