



Detailed Assessment Brief

Assessment 3 Full-Stack Web Application with Responsible AI Integration

Unit Code/Description	ICT203 Web Application Development
Course/Subject	BIT
Semester	1
Unit Learning Outcomes Addressed	1, 2, 3
Assessment Objective	Three-member student group to develop Full-Stack Web Application with Responsible AI Integration
Assessment Title/Type	Full-Stack Web Application with Responsible AI Integration
Word Count (if Applicable)	
Due Date	End of Week 11
Weighting	30
Instructions to Students	<p>1. Industry Scenario (Authentic Brief) You are a small software delivery team contracted by a client (choose one):</p> <ul style="list-style-type: none">• Community services provider (appointments + case notes)• Small retail business (catalogue + orders)• Local events organiser (events + ticket requests)• University club/association (membership + announcements) <p>The client requires a secure, responsive, database-driven web application with clear delivery milestones, documentation, and a deployable build.</p> <p>2. Project Goal Design, build, test, and deploy a full-stack web application that demonstrates:</p> <ul style="list-style-type: none">• Professional software engineering workflow (planning → build → test → deploy)• Secure development practices• Team collaboration and role accountability• Responsible AI augmentation (AI used as an assistant, not a replacement) <p>3. Technology Baseline (Recommended, adaptable) Choose one stack: Option A: PHP + MySQL (Foundations-aligned)</p>

	<ul style="list-style-type: none"> • Front-end: HTML/CSS/JS • Backend: PHP 8+ (MVC-lite structure acceptable) • DB: MySQL + PDO • Auth: sessions + password hashing <p>Option B: Node.js + Express + MySQL (Modern industry variant)</p> <ul style="list-style-type: none"> • Front-end: HTML/CSS/JS (or lightweight templating) • Backend: Node.js + Express (REST API) • DB: MySQL (or PostgreSQL if available) <p>Constraints:</p> <ul style="list-style-type: none"> • Frameworks permitted but must be justified (Bootstrap allowed; React optional only if the course expects it). • REST APIs can be used • Must use GitHub with branches and pull requests. <p>4. Minimum Functional Requirements (Must-Have)</p> <p>4.1 Core Features</p> <p>Your application must include:</p> <ol style="list-style-type: none"> 1. User authentication & role-based access <ul style="list-style-type: none"> ○ Roles: Admin + Standard User (minimum) ○ Secure password storage (bcrypt / password_hash) 2. CRUD module (at least 2 entities) <ul style="list-style-type: none"> ○ Example entities: Users, Products, Orders, Events, Bookings, Posts, Appointments 3. Search + filter + pagination on a primary dataset 4. Data validation <ul style="list-style-type: none"> ○ Client-side validation (JS) ○ Server-side validation (backend) 5. Responsive UI <ul style="list-style-type: none"> ○ Mobile-first, accessible, professional layout 6. Auditability <ul style="list-style-type: none"> ○ Activity log or “created/updated by + timestamps” for key records <p>4.2 Functional Requirements</p> <ul style="list-style-type: none"> • Security baseline: input sanitisation, prepared statements, access control checks • Usability: consistent navigation, error messages, form feedback • Reliability: graceful handling of invalid inputs, empty states, and system errors • Maintainability: code structure, naming conventions, README, installation guide <p>5. Required AI Integration (Responsible and Value-Adding)</p> <p>Students must incorporate AI in one of these approved ways (minimum one, maximum two to keep scope realistic):</p> <p>AI Option 1: “Smart Assistant” (Low-risk, high value)</p>
--	---

	<ul style="list-style-type: none"> • A “Help” assistant that answers questions about the system using a curated FAQ/knowledge base stored as text/JSON. • Implementation can be: <ul style="list-style-type: none"> ○ Rule-based + optional LLM summarisation ○ Or local curated responses (no external calls) <p>AI Option 2: Text Summarisation / Drafting Workflow (Business use case)</p> <ul style="list-style-type: none"> • Example: summarise an incident report, case note, or customer request into a standard template • Must include: <ul style="list-style-type: none"> ○ User review/edit before saving ○ Clear disclaimer: “AI-generated content requires human review” <p>AI Option 3: Intelligent Search (Semantic-ish)</p> <ul style="list-style-type: none"> • Tagging or keyword extraction for records (e.g., auto-suggest categories) • Must log what AI suggested and what user accepted <p>AI Option 4: UI/UX Assistant Feature</p> <ul style="list-style-type: none"> • Generate alternative titles/descriptions for content (events posts, product descriptions) • Must be editable before publishing <p>AI Safety Requirements (Mandatory):</p> <ul style="list-style-type: none"> • AI outputs must be reviewed by a human user before being stored or published. • No sensitive personal data is to be sent to public AI tools (simulate data if needed). • Include an AI Use Statement in the README and report.
Format/Structure	<p>. Assessment Structure and Timing (Recommended)</p> <p>Milestone 1 (Week 4–5): Proposal + Design (10–15%)</p> <ul style="list-style-type: none"> • Proposal + ERD + architecture + backlog <p>Milestone 2 (Week 7–8): MVP Demo (20–25%)</p> <ul style="list-style-type: none"> • Auth + one entity CRUD + basic UI + DB integration <p>Milestone 3 (Week 11–12): Final Delivery</p> <ul style="list-style-type: none"> • Full features + AI feature + security + testing + deployment
Referencing Style	
Submission Guidelines for the Assessment	<p>6. Deliverables (What to Submit)</p> <p>Deliverable A — Project Repository (Group)</p> <ul style="list-style-type: none"> • GitHub repo with: <ul style="list-style-type: none"> ○ README.md (setup, run, credentials for demo accounts, deployment link) ○ /docs folder (required documents below) ○ meaningful commit history and PRs <p>Deliverable B — Working Application</p> <ul style="list-style-type: none"> • Exported htdocs from XAMPP

	<ul style="list-style-type: none"> Database schema included (SQL dump or migration scripts) <p>Deliverable C — Documentation Pack (/docs)</p> <ol style="list-style-type: none"> Project Proposal (2–3 pages) <ul style="list-style-type: none"> problem statement, target users, features, tech stack, roles, milestones System Design <ul style="list-style-type: none"> Architecture diagram (client–server–DB) ERD (database schema) API endpoints list (if applicable) Security & Risk Register (1–2 pages) <ul style="list-style-type: none"> Top 5 risks + mitigation (incl. AI risks) Test Evidence <ul style="list-style-type: none"> At least 10 test cases (mix of positive/negative) Evidence screenshots or logs AI Governance Appendix <ul style="list-style-type: none"> AI feature chosen, data handling, prompt approach, human-in-the-loop mechanism AI limitations and mitigations <p>Deliverable D — Team Presentation + Demo (10–12 minutes)</p> <ul style="list-style-type: none"> 6–8 slides maximum + live demo Each member speaks (minimum 1–2 minutes each) <p>7. Team Roles (3–4 students)</p> <p>Each team must assign roles (can overlap but must be explicit):</p> <ul style="list-style-type: none"> Team Lead / Scrum Master (workflow, coordination, backlog) Front-End Lead (UI, accessibility, responsiveness) Back-End Lead (API/server logic, security) Data/QA Lead (DB design, testing, documentation)
Late Submission Policy for Assessment	Without an approved extension or mitigating circumstances, assessment items submitted after the due date will be penalised at a rate of 10% of the total mark allocated to the assessment item per day (or part day) late. Any assessment items submitted more than ten (10) days after the due date will receive zero marks. Extensions due to mitigating circumstances are at the discretion of the Lecturer for the unit and must be granted in writing.
Academic Integrity and AI Use Requirements	This assessment allows full integration of and the use of Generative AI. Follow the instructions for responsive AI use

Marking Rubric for Assessment 3(Total Marks 30%)

Assessment Criteria	High Distinction (HD)	Distinction (DI)	Credit (CR)	Pass (PS)	Fail (FL)	Marks
1. Functional Completeness & Business Requirements	All required features fully implemented, robust, and aligned to realistic business workflows. Handles edge cases and errors professionally.	Most features implemented with minor gaps or edge-case issues. Business logic mostly sound.	Core features implemented but workflows are limited or partially complete.	Basic functionality present; several required features missing or unreliable.	Major features missing or system largely non-functional.	8
2. System Architecture & Database Design	Clear, well-structured architecture. ERD fully normalised and aligned to system logic. Code separation demonstrates strong design thinking.	Appropriate architecture with minor design or normalisation issues.	Functional but simplistic architecture. Limited modularity or scalability.	Minimal architectural planning. Database works but lacks clarity.	Poor or missing architecture and/or incorrect database design.	4
3. Front-End Quality, UX & Accessibility	Responsive, polished UI with excellent UX. Accessibility principles clearly applied (semantic HTML, labels, contrast, keyboard support).	Responsive and usable UI with minor UX or accessibility issues.	Basic responsiveness and usability. Limited accessibility awareness.	Functional but visually basic UI. Poor responsiveness or accessibility.	Unusable or inaccessible interface. No responsiveness.	4
4. Security & Data Handling	Secure coding practices consistently applied (validation, sanitisation, prepared statements, access control, password hashing). Risks well mitigated.	Good security practices with minor omissions.	Some security awareness evident but incomplete implementation.	Minimal security applied. Significant vulnerabilities remain.	Security ignored or critically flawed.	4

5. AI Feature Implementation & Responsible Use	AI feature is meaningful, well-integrated, and adds clear value. Human-in-the-loop review implemented. AI risks identified and mitigated. Excellent governance documentation.	AI feature adds value and is responsibly implemented with user oversight.	AI feature present but limited in impact or integration. Governance addressed superficially.	Minimal AI use with limited justification or control.	AI missing, inappropriate, or used irresponsibly.	4
6. Testing, Quality Assurance & Documentation	Comprehensive testing (positive & negative cases) with clear evidence. Professional documentation.	Adequate testing and documentation with minor gaps.	Basic testing evidence provided.	Minimal or poorly documented testing.	No meaningful testing or documentation.	2
7. Individual Contribution & Professional Practice	Clear, substantial contribution evidenced via commits, pull requests, documentation, and leadership. Demonstrates professional teamwork.	Strong contribution with clear evidence.	Adequate contribution but uneven participation.	Limited contribution; relies heavily on others.	Minimal or no meaningful contribution.	4