Project 2
CIT1C21 - Application Development Project (40%)
CIA1C06 - Database Application Development (30%)
(System Code Implementation)

Introduction

This assignment aims to assess students' understanding and practical application of web development principles. The objective is to design and implement a functional website by creating both the front-end and back-end components.

Learning outcomes:

By the end of this assignment, you should be able to:

- Implement the frontend of a website using web technologies (HTML, CSS, JavaScript).
- Implement the backend of a website, including server-side logic and database integration.

Notes:

a. This is an individual assignment.
b. Your solution should cover the requirements above.

Using solutions found online or AI generated work from the internet is prohibited.

Report Template:

A **report template**, labeled ‘ADEV DBAV-AY2425 Project 2 Template’, will be provided for you to complete. In the template, you will need to include:

ADEV

- List down all the URLs to the respective pages and provide sample input.
- Provide screenshot(s) of the user interfaces of the main features.
- Provide screenshot(s) of your additional features and explain how they enhance the website.

DBAV

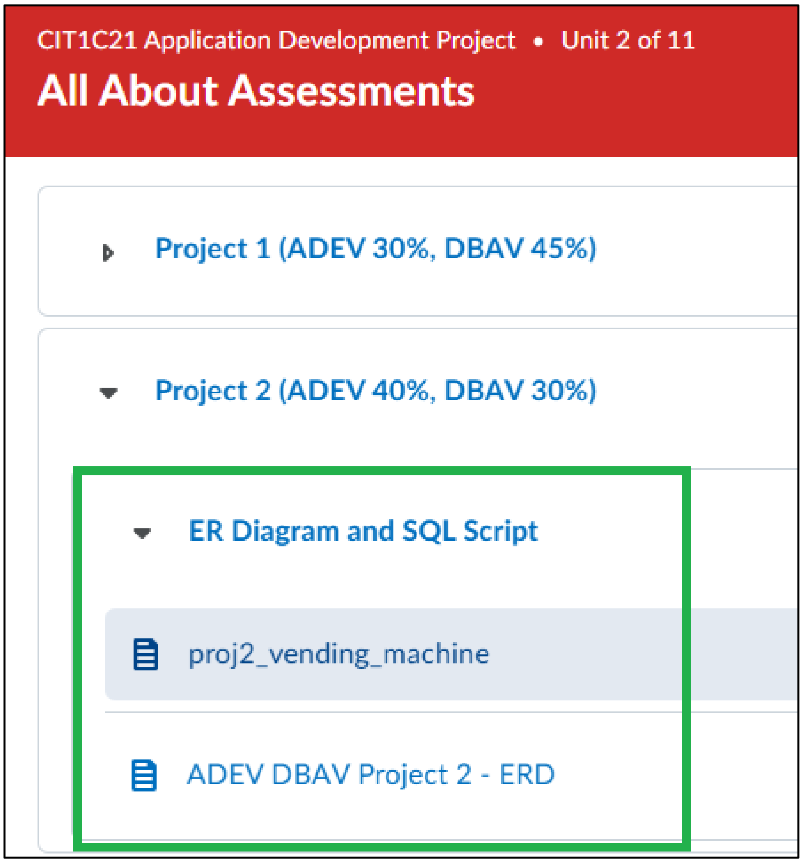
- List all the SQL statements, route pattern, HTTP method and payload (if any) for the REST APIs implemented.
- Explain the error-handling mechanisms that were implemented.

Note: Your tutor will only grade the web pages and API routes listed in the report. Therefore, it is your responsibility to ensure that all the required information is included.

Project Brief

Based on the website proposals for an online website that will enhance students' experience studying in the polytechnic, one of the ideas has been selected. You will be provided with the database through an SQL script and will be required to populate the data on your own.

Note: SQL script and ERD will be available for download in week 9 (9th December), download it from LMS.
(LMS -> ADEV -> Content -> All About Assessments -> Project 2 (ADEV 40%, DBAV 30%) -> ER Diagram and SQL Script)



CIT1C21 Application Development Project • Unit 2 of 11

All About Assessments

Project 1 (ADEV 30%, DBAV 45%)

Project 2 (ADEV 40%, DBAV 30%)

ER Diagram and SQL Script

proj2_vending_machine

ADEV DBAV Project 2 - ERD

Vending Machine

This website provides a comprehensive list of all vending machines located on TP campus, along with details on the items available for purchase and payment methods. Your role is to provide the admin of the website with the functionality to view, create, update, and delete the vending machine items. Your website should allow the following operations:

- **Display all the vending machines**

Create a main page that displays all the vending machines from the database. The following information should be displayed:

- The block the vending machine is located at
- The floor where the vending machine is located
- Payment method(s) available for the vending machine
- A hyperlink or modal that displays more information about the items in the vending machine

- **Display all vending machine items**

Display the items of the selected vending machine in a new page or modal when the user clicks on the hyperlink. The following item information should be displayed:

- Item name
- Item cost
- Item image
- Item availability

- **Add, update and delete vending machine items**

You will implement the functionality to add new vending machine items, update and delete existing items. The following information should be entered when the admin adds a new item:

- Item name
- Item cost
- Item image
- Item availability
- Quantity

ADEV Part 2: System Code Implementation (40%)

Frontend Coding

You are tasked with coding the website's frontend. The website must use the RESTful API that you will implement in the DBAV section of this project. Your design should incorporate essential elements that map directly to the structure outlined in the ER diagram, demonstrating a clear connection between your frontend and database.

You have the freedom to design the web pages in the style of your choice, but you will need to ensure the project requirements are met.

- ### Frontend Implementation (25%)

Implement the code to allow the user to access and utilize the website's features to perform required tasks.

You will need to list down all the URLs to the different pages of the website on the report and provide the sample input values if any.

- ### User Interface (10%)

Implement a user interface that is cohesive, intuitive, easy to navigate and appropriate for the intended user.

Provide screenshot(s) of the implemented user interface for each of the core features in the report.

- **Additional Features (5%)**

You are encouraged to enhance the website with extra features beyond the basic requirements. These additions should improve the functionality, usability, or aesthetics of the site. (Max of 2 additional features)

Apart from code implementation, you must explain how the additional features enhance the website. Provide screenshot(s) and explanation of the additional feature(s) in the report provided.

You have a 150-word limit for the explanation for each additional feature.

DBAV Part 2: System Code Implementation (30%)

You are tasked with coding the backend of your web application to provide RESTful API for your frontend code. The SQL scripts for the database will be provided, however, you are required to populate the tables with meaningful data.

• API Implementation (16%)

Demonstrate API implementations of the SQL queries for all required operations. There should be at least one SQL query each for Create, Update, Retrieve, and Delete operations.

Include all SQL statements, route patterns, HTTP methods, and payloads (if any) in the report.

• Additional Features (4%)

Based on the additional features you have identified and designed, you are tasked to link the frontend implementation to the backend. Provide the RESTful API implementation.

Include all SQL statements, route patterns, HTTP methods, and payloads (if any) in the report.

• Error Handling (10%)

You are tasked to implement 2 (TWO) error-handling mechanisms on your server-side system, explaining how you carried out the implementation and why this is important.

You have a 200-word limit per mechanism.

Submission

You are to submit the report, project, and the SQL scripts. Submit the deliverables into LMS under the Assessment folder by **Week 15, Friday, 24h Jan 2025, at 1700hrs (5 pm).**

Submission instructions:

1. Make a copy of your project.
2. In the copy, remove the node_modules folder.
3. Export SQL scripts of your database to a single self-contained file. (Refer to the following YouTube video: https://www.youtube.com/watch?v=lbrVhjnM5MQ for steps to export your SQL scripts)
4. Put your project, SQL script and the report into the same folder and zip up the folder.
5. You should name your folder according to the following format:

   StudentID-YourName-ADEV-DBAV-AY2425-Oct-Project 2.zip

   e.g.: 1234567D-Teo_Chu_Chu-ADEV-DBAV-AY2425-Oct-Project 2.zip

Penalty for Late Submission

late and < 1 day	: 10% deduction from absolute mark given for the assignment
late >= 1 and < 2 days	: 20% deduction from absolute mark
late >= 2 days	: No marks awarded

Note that "day" includes **non-working days** (Sat, Sun and public holidays).

Project – Grading Criteria

The grading criteria for **ADEV Part 2: System Code Implementation (40%)** will be based on the following:

Criteria	In Context	Performance Level				
		A (>=80%)	**B (>=70%)**	**C (>=60%)**	**D (>=50%)**	**F (<50%)**
		Excellent	**Good**	**Average**	**Pass**	**Below Standard**
Frontend Implementation (25%)	Completeness of features (15%)	All the required website features are fully implemented and functional		Only a portion of the website features are implemented and functional.	A minimal number of the required website features are implemented and functional.	Non-submission or no features are functional.
	Robustness of frontend (10%) (Robustness means to prevent error and remain functional in the event of error)	The website consistently handles requests under typical conditions and recovers well from common errors or abnormal inputs. Mechanisms to ensure valid operations are highly reliable and consistently effective, preventing almost all errors.	The website handles standard operations without crashing but may not recover gracefully from all unexpected conditions. Mechanisms to ensure valid operations are effective in most scenarios, minimizing errors during normal use.	The website handles standard operations without crashing but may not recover gracefully from all unexpected conditions. Mechanisms to ensure valid operations are present but inconsistently applied, leading to occasional errors.	The website frequently crashes or returns errors during normal operations. Mechanisms to ensure valid operations are weak, leading to common errors or unexpected results.	Non-submission or no measures are considered to make the website robust.

User Interface (10%)	Intuitive (5%) - Ease of use	The interaction flow is seamless and user-friendly, allowing users to move effortlessly between sections and complete tasks without confusion.	The interaction flow is generally smooth, though some areas could benefit from clearer organization or labels. Most actions are easy to figure out	The interaction flow works but may require extra effort from users. Some elements are unclear or difficult to locate, reducing the ease of use.	The interaction flow is confusing and disorganized, making it difficult for users to find features or complete tasks efficiently.	The interaction flow is broken and highly confusing with non-functional elements that require constant page transitions and prevent task completion.
	Visual design (5%) - aesthetics - coherence	The design is polished and professional, with a cohesive and harmonious overall presentation. Elements are arranged thoughtfully to create a visually engaging and readable interface.	The design is appealing, with a well-organized structure, though there are minor inconsistencies in presentation.	The design feels basic and inconsistent, with elements that lack harmony and attention to detail, leading to a less engaging appearance.	The design lacks quality, with poorly arranged or mismatched elements that make the interface confusing or unattractive.	The design appears incomplete or unprofessional. Crucial elements missing, making the website unusable.
Additional Features (5%)	Relevance (3%)	The additional features provide exceptional value to the application, greatly enhancing its overall	The additional features provide significant value to the application, enhancing its overall	The additional features provide moderate value to the application, partially enhancing its overall	The additional features provide minimal value to the application and do not enhance its	Non-submission or failure to add any meaningful functionality or user experience improvement.

		functionality and user experience.	functionality and user experience.	functionality or user experience.	overall functionality or user experience.	
	Explanation (2%)	The explanation is clear, thorough, and well-justified, showing strong alignment between the feature and the website's goals or user needs. The rationale demonstrates thoughtful consideration and adds significant value to the website.	The explanation is clear and relevant but may lack depth or a strong connection to user needs. The rationale is reasonable and aligns with the website's purpose.	The explanation is basic, with limited detail or justification for how the feature adds value. The rationale may feel generic or disconnected from user needs.	The explanation is vague or lacks justification for the feature. It does not demonstrate why the feature was added or how it contributes to the website.	Non-submission, no connection or the explanation is incomplete, providing no reasoning for the feature.

The grading criteria for **DBAV Part 2: System Code Implementation (30%)** will be based on the following:

Criteria	**In Context**	**Performance Level**				
		A (>=80%)	**B (>=70%)**	**C (>=60%)**	**D (>=50%)**	**F (<50%)**
		Excellent	**Good**	**Average**	**Pass**	**Below Standard**
API Implementation (16%)	Backend implementation (16%)	All the website's backend requirements are met, with the API consistently handling requests under typical conditions			Part of the website's backend requirements are met, with the API handling some requests under typical conditions	No database operation is present Incorrect SQL Codes, Implementation failed to function, Non-submission of Part 2
		Proper SQL implementation for CRUD operations, including the use of multiple JOIN queries	Proper SQL implementation of CRUD operations, including the use of at least one join query.	Correct SQL implementation of CRUD operations	Partially correct SQL implementation of CRUD operations	
		Populated data contains comprehensive set of meaningful records	Populated data contains good set of meaningful records	Populated data contains good set of meaningful records	Populated data contains very few records	
Additional Features (4%)	Implementation of Additional	Both additional feature's	Both additional feature's backend	At least one of the additional feature's	At least one of the additional feature's	No backend implementation of

	Features from ADEV Part 2 (4%)	backend API implementation is provided and fully functional, successfully handling all requests under any conditions	API implementation is provided and fully functional, handling requests under typical conditions	backend API implementation is provided and fully functional, handling requests under typical conditions	backend API implementation is provided and fully functional, handling requests under typical conditions	additional feature provided.	
Error Handling (10%)	Explanation (5%)	The importance of both error-handling mechanisms are clear and well elaborated, and supported with strong examples	The importance of both error-handling mechanisms are explained with	The importance of both error-handling mechanisms are briefly elaborated with a weak example	The importance of both error-handling mechanisms are briefly elaborated but clear	The importance of both error-handling mechanisms are not relevant or unclear Only one error handling mechanism is mentioned	
	Implementation (5%)	The API recovers well from all types of errors or abnormal inputs, giving clear, actionable error messages for various scenarios	The API recovers well from most errors or abnormal inputs, but not all messages are clear or helpful	The API can only handle either all abnormal inputs or only other errors, but does not provide meaningful error messages when error occurs	The API can handle simple errors but does not provide meaningful error messages	One or zero error handling mechanisms implemented	

		Implementation for each error-handling mechanism is thoroughly explained and shows strong understanding of the code	Implementation for each error-handling mechanism is well explained and shows strong understanding of the code	Implementation for each error-handling mechanism is briefly explained but shows strong understanding of the code	Implementation for each error-handling mechanism is briefly explained and shows some understanding of the code	Explanation for code implementation is poor and shows clear lack of understanding	