College of Art, Technology and Environment
ACADEMIC YEAR 2025/26

Assessment Brief

Unit Details

Unit Code	UWE213
Unit Title	Web Data Applications
Unit Leaders	Professor Jun Hong, Khoa Phung
Unit Tutors	Professor Jun Hong, Khoa Phung
Year	2025 – 2026
Task Name	Group Project
Total Number of Assessments for This Unit	2
Weighting	60%
Task Description	Group Project and Individual Reflection Report

Dates

Date issued to students	21st October 2025
Marks and Feedback	As soon as possible
Submission Date	28th November 2025
Submission Place	UWE Blackboard
Submission Time	Before midnight on 28th November 2025 (17:00 UK time). *This assessment is NOT eligible for late submission.*

Section 1: Overview of Assessment

This assessment assesses the following learning outcomes:

1. Effectively use web standards for the retrieval and representation of data to derive meaningful and useful structure (form) and information (content) from a variety of web services.
2. Using relational databases and JSON as a grounding, understand the emergence and uses made of very-large-scale NoSQL key-value pair databases. Analyse through specific examples the isomorphic nature of these databases, how schemas are and can be applied, and how these databases are queried, updated, replicated, and maintained.
3. Understand the value of data in an organisational and wider societal context. Appreciate the "context of use" and when and how this data needs to be authenticated, authorised, validated, mined, shared, secured, and maintained.

The assessment is worth ***60%*** of the total mark.

This assessment is a group project, requiring you to work in a group of ***2 to 3 (maximum) students***. You need to be in the same group as in Data Schemas and Advanced Data Modelling units.

The primary objective of this assessment is to design and develop a web application based on the case study provided in Section 4.

As a group, you are required to implement **FR1, FR2, and AT LEAST ONE more functional requirement** from the functional requirements list in Section 4. *Note that each functional requirement group consists of multiple related requirements and thus, you will need to implement them all.*

Additionally, you will individually reflect on your development experience, roles, problem-solving strategies, group collaboration, and learning outcomes. This reflective process will help you to gain insights into your strengths and areas for growth, contributing to your ongoing professional development.

Your group will need to interact with the client and stakeholders (your tutors) as part of completing your assessment, simulating real-world client interactions. The details of the assessment are described further in Section 3.

You are encouraged to refer to the Marking Criteria in Section 5 to understand the expectations and grading standards. This reference will guide your efforts and help you aim for excellent performance in this assessment.

Section 2: Deliverables (Final Submission)

1. Codebase (*Group* – Folder)

- *Front-end Folder (named **front-end**):*
  - **index.html**: Main HTML file.
  - **CSS files**: Styling files.
  - **JavaScript files**: Client-side scripting files.
  - **Media assets**: Any images or media used.

- *Back-end Folder (named **back-end**):*
  - **Server file (e.g., index.js)**: Main server logic.
  - **Database schema**: Schema files for chosen database (SQLite or MongoDB).
  - **Configuration files**: Any necessary configuration files.

- *README File:* Instructions on how to run and/or test the application.

2. Video Demo

- A **video demo (MP4, up to 5 minutes)** of your developed website, including a brief overview of your project (e.g., the additional FR you have chosen in addition to FR1 and FR2), a walkthrough of your website highlighting its key features and how it meets the requirements.

3. Reflection Report (*Individual* – Word/PDF)

- Discussion of your choices, roles, problem-solving strategies, and group collaboration experience.
- Reflection on your feelings and learning outcomes.
- Suggestions for improvements if given a second chance.
- More detail in Part 5. Word limit: 500.

Please ensure that ALL elements are compressed into a single zip file for final submission.

Section 3: Specifications

Part 1: Front-end Implementation (45%)

Develop the front end of the web application using HTML, CSS, and JavaScript.

Part 2: Back-end Implementation (45%)

Develop the back end of the web application to support the front-end functionality and manage data operations. You should use ExpressJS for API development.

Part 3: (10%)

A video demonstration of your developed website.

Part 4: Reflection Report (Individual Part) (10%)

Reflect on your web application development experience and the lessons learned during the project.

Word limit: 500 words.

Section 4: Case Study & Requirements

Student Bay is a new digital marketplace concept designed specifically for university students to buy and sell second-hand items with each other. The platform will help students save money by purchasing used textbooks, electronics, furniture, and other essentials from fellow students.

The platform will allow students to advertise and swap everything from course textbooks and laptops to mini-fridges and bicycles.

Your team has been chosen to develop Student Bay from the ground up. Here are the key areas you must focus on when developing the platform.

- **Item Discovery:** Users need efficient ways to find specific items, particularly textbooks for their courses. The search functionality should be intuitive and comprehensive.
- **Item Management:** The system requires smooth processes for tracking item availability and collecting feedback where possible.
- **Trust and Safety:** Building confidence between users is essential. This includes university email verification, optional student/staff ID validation, profile completeness indicators, and visible rating histories from previous buyers and sellers.

Functional Requirements

FR1: User Profiles and Verification
Members create profiles with multiple verification options. University email addresses (e.g., .edu domains) serve as the primary verification method, with email confirmation required.

Profiles display trust indicators including verification status (verifies / unverified), join date, successful sales count, and average rating. A profile completeness score shows as a percentage based on completed fields such as photo, bio, university, and/or year of study.

The system supports three membership tiers:

- Public Members: View-only access, no verification required.
- Student Members: Full buying and selling privileges, verified via university email.
- Affiliated Members: University staff with full buying and selling privileges, verified via institutional staff email.

FR2: Item Listings
Users can post items with up to five photos, detailed descriptions, condition ratings (New to Poor), and asking prices.

Each listing displays a view counter to help sellers gauge interest in their items.

Each category includes specific fields relevant to that item type:

Category	Specific Fields
Textbooks	ISBN, Course Code, Module Name, Edition, Author
Electronics	Brand, Model Number, Warranty Status, Original Purchase Date, Accessories Included
Furniture	Item Type, Dimensions, Material, Assembly Required, Condition Details
Clothing	Size, Brand, Material, Colour, Gender
Sports Equipment	Brand, Size/Dimensions, Sport Type, Condition Details

FR3: Search and Discovery
Users can search items using keywords, category filters (Textbooks, Electronics, Furniture, Clothing, Sports Equipment), price ranges, and condition. Textbook searches specifically support course codes, module names, and ISBN lookups.

Search results show essential information at a glance: thumbnail images, prices, seller ratings, posting dates, and view counts.

FR4: Watch Lists and Notifications
Users can save items to a personal watch list for tracking. The system sends email notifications when watched items reduce in price, change status, or when new items matching saved search criteria appear.

FR5: Item Status Management
Sellers manage three item states: Available, Reserved (pending collection), and Sold.

When marking an item as sold, sellers can either select the buyer from interested registered users or indicate the buyer was not a system user. This distinction determines whether the rating process can be initiated.

The system maintains complete records of all listings, purchases, and status changes with associated dates.

FR6: Ratings and Reviews
When a seller completes a sale with a registered buyer, the system enables a review process. The buyer can provide a star rating (1-5) and optional written feedback (maximum 1000 characters). Reviews become permanent additions to seller profiles.

If the buyer is not a registered user, the seller simply marks the item as sold without triggering the review process.

Each buyer can review a seller only once per completed sale, preventing duplicate reviews for the same item sold.

Section 5: Marking Criteria

A holistic approach will be taken in marking the submission.

The individual mark of each member in the group is reflective of their contribution to both the group-based parts and their individual deliverables. For adjusting individual marks, meeting minutes, presentation reviews, individual contribution towards group work, participating in lectorial activities will be considered.

Please note that normal rules about plagiarism and collusion apply as usual. Plagiarism in group deliverables will affect the overall group deliverables. Individual plagiarism by an individual in a section will affect the overall individual and group contribution. Zero marks could be awarded for non-engagement in group work.

Criteria	< 39 (Fail)	40 – 49 (3rd)	50 – 59 (2:2)	60 – 69 (2:1)	70+ (1st)	80+ (High 1st)
Front-end Implementation: HTML & CSS **[20%]**	The HTML & CSS is poorly implemented, lacks structure, and does not match the design. The page is not responsive, and the user interface is not user-friendly.	The HTML & CSS matches the design but lacks some structure and usability. The page is somewhat responsive, and the user interface is usable but not intuitive.	The HTML & CSS matches the design and has satisfactory structure and usability. The page is responsive, and the user interface is intuitive.	The HTML & CSS matches the design, has good structure, and is user-friendly. The page is responsive across multiple devices and the user interface enhances the user experience.	The HTML & CSS matches the design, has excellent structure, and is very user-friendly. The page is responsive across all devices and the user interface significantly enhances the user experience.	The HTML & CSS matches the design, has outstanding structure, and is extremely user-friendly. The page is highly responsive across all devices and the user interface significantly enhances the user experience and offers innovative features.

Front-end Implementation: JavaScript **[25%]**	The JavaScript is poorly implemented, lacks interactivity, and does not match the design. It does not include the use of DOM manipulation, event handling, or form validation.	The JavaScript matches the design but lacks some interactivity and usability. It includes some use of DOM manipulation, event handling, and form validation, but these features are not implemented effectively.	The JavaScript matches the design and has satisfactory interactivity and usability. It includes effective use of DOM manipulation, event handling, and form validation.	The JavaScript matches the design, has good interactivity, and is user-friendly. It includes effective and efficient use of DOM manipulation, event handling, form validation, and enhances the user experience.	The JavaScript matches the design, has excellent interactivity, and is very user-friendly. It includes effective and efficient use of DOM manipulation, event handling, and form validation, significantly enhances the user experience and demonstrates a deep understanding of JavaScript.	The JavaScript matches the design, has outstanding interactivity, and is extremely user-friendly. It includes effective and efficient use of DOM manipulation, event handling, and form validation, significantly enhances the user experience, demonstrates a deep understanding of JavaScript, and includes innovative features.

Back-end Implementation: Node.js & Express **[25%]**	The Node.js & Express is poorly implemented, lacks functionality, and does not support the front-end well. It does not include the use of routing, middleware, form data handling, or session cookies.	The Node.js & Express supports the front-end but lacks some functionality. It includes some use of routing, middleware, form data handling, or session cookies, but these features are not implemented effectively.	The Node.js & Express supports the front-end and has satisfactory functionality. It includes effective use of routing, middleware, form data handling, or session cookies.	The Node.js & Express supports the front-end and has good functionality with minor bugs. It includes effective and efficient use of routing, middleware, form data handling, or session cookies, and enhances the user experience.	The Node.js & Express supports the front-end and has excellent functionality with hardly any bugs. It includes effective and efficient use of routing, middleware, form data handling, or session cookies, significantly enhances the user experience and demonstrates a deep understanding of Node.js & Express.	The Node.js & Express supports the front-end and has outstanding functionality with no bugs. It includes effective and efficient use of routing, middleware, form data handling, or session cookies, significantly enhances the user experience, demonstrates a deep understanding of Node.js & Express, and includes innovative features.

Back-end Implementation: Database **[10%]**	The database lacks structure and does not support the application's functionality. No justification for the choice between SQL (SQLite) or NoSQL (MongoDB) is provided. CRUD operations are not included, and there is no evidence of understanding database manipulation.	The database has some structure but lacks some functionality. A basic justification for the choice between SQL (SQLite) or NoSQL (MongoDB) is provided. Some CRUD operations are implemented, but they are not effective.	The database has satisfactory structure and supports the application's functionality. A clear justification for the choice between SQL (SQLite) or NoSQL (MongoDB) is provided. CRUD operations are effectively used, and the code includes basic documentation.	The database has good structure, supports the application's functionality, and shows understanding of SQL (SQLite) or NoSQL (MongoDB) with a well-explained justification. CRUD operations are effectively and efficiently implemented, and the code documentation enhances the user experience.	The database has excellent structure, supports the application's functionality, and shows deep understanding of SQL (SQLite) or NoSQL (MongoDB) with strong justification. CRUD operations are implemented in an innovative way, and the code documentation significantly enhances the user experience and demonstrates deep understanding.	The database has outstanding structure, supports the application's functionality, and shows innovative use and deep understanding of SQL (SQLite) or NoSQL (MongoDB) with comprehensive justification. CRUD operations are implemented with exceptional efficiency, and the code documentation offers innovative features and significantly enhances the user experience.

Video Demo **[10%]**	The video demo is poorly structured, exceeds time limit significantly, has poor audio/visual quality, and fails to demonstrate key functionality. The project overview is unclear or missing, and the walkthrough does not effectively showcase the application's features.	The video demo has basic structure and stays mostly within time limit, but has some audio/visual quality issues. The project overview is present but lacks clarity, and the walkthrough covers some features but misses important functionality or requirements.	The video demo is well-structured, stays within time limit, and has acceptable audio/visual quality. The project overview clearly explains the chosen FR, and the walkthrough adequately demonstrates key features and requirements.	The video demo is well-organised, stays within time limit, and has good audio/visual quality with clear narration. The project overview is comprehensive, and the walkthrough effectively demonstrates all key features and how they meet the requirements.	The video demo is excellently structured, utilises time efficiently, and has high-quality audio/visual with engaging narration. The project overview is thorough and insightful, and the walkthrough comprehensively demonstrates all features with clear explanations of technical implementation.	The video demo is well-produced, maximises time efficiency, and has professional-quality audio/visual with compelling narration. The project overview shows deep understanding and innovation, and the walkthrough provides an outstanding demonstration with insightful technical explanations and innovative presentation techniques.

Individual Reflection **[10%]**	The reflection lacks depth, fails to discuss personal roles, problem-solving strategies, group collaboration, learning outcomes, and provides no insights into improvements.	The reflection discusses roles and some strategies but lacks detail and clarity. There is minimal discussion of collaboration and learning outcomes.	The reflection adequately discusses roles, strategies, collaboration, and learning outcomes but may lack personal insights and specific examples.	The reflection is well-written, discussing roles, strategies, collaboration, and learning outcomes in detail, with some personal insights and examples. Suggestions for improvements are clear.	The reflection is excellently written, providing deep insights into roles, strategies, collaboration, and learning outcomes. It includes personal examples and well-considered suggestions for improvements.	The reflection provides exceptional insights and innovative thoughts on roles, strategies, collaboration, and learning outcomes. It includes detailed examples and innovative suggestions for improvements.

Section 6: Appendices

6.1. Assessment Offences

How do I avoid an Assessment Offence on this module?
Use the support above if you feel unable to submit your own work for this module.

UWE Bristol's UWE's Assessment Offences Policy requires that you submit work that is entirely your own and reflects your own learning. It is important to:

- Ensure that you reference all sources used, using the UWE Harvard system. Use the guidance available on UWE's Study Skills referencing pages.
- Avoid copying and pasting any work into this assessment, including your own previous assessments, work from other students or internet sources.
- Develop your own style, arguments and wording. Avoid copying and changing individual words but keeping essentially the same sentences and/or structures from other sources.
- Never give your work to others who may copy it.
- If you are doing an individual assessment, develop your own work and preparation. Do not allow anyone to make amendments to your work (including proof-readers, who may highlight issues but not edit the work).
- When submitting your work, you will be required to confirm that the work is your own. Text-matching software and other methods are routinely used to check submissions against other submissions to the university and internet sources. Details of what constitutes plagiarism and how to avoid it can be found on UWE's Study Skills pages about avoiding plagiarism.

6.2. Use of Generative AI (ChatGPT or similar)

	Generative AI must not be used in this assessment for writing code and reflective report.
	You can only use Generative AI in this assessment for assisting you with technical issues and checking spelling and grammar. If this has been done, it must be declared and acknowledged.