

INFS3202/7202 Individual Project

Overview

1. The individual project is intended to evaluate your proficiency in designing and developing WIS, with an emphasis on the technology aspect of the course.
2. In general, this project includes a **project proposal (5 marks)** and **implementation (40 marks)** that will be assessed in **THREE** milestone check-up sessions based on academic merit.
3. **THREE** project topics in **Table 1** have been provided for you to choose from, and you will need to select one and only one to work on independently throughout the semester. Any other project proposed by the student needs to be approved by the teaching team.
4. The proposal needs to be submitted online in Blackboard and will be tested against the Turnitin system for a plagiarism test. For details of the proposal, please refer to **Appendix I**.
5. The general technical requirements are listed in **Table 2**, please make sure you carefully read them through and feel free to ask questions to your tutor.
6. The features that require to be implemented are categorised into two groups: ungraded and graded features.
 - a. The ungraded features include all the core features that have been covered and practised in the practical sessions. They are the core of your web project, but they will be NOT graded.
 - b. According to the workload and complexity, the graded features are divided into three difficulty levels: basic, intermediate, and advanced features. They are assessed by different marks.
 - c. In each code check-up session (milestone 2 or milestone 3), only 20-mark features will be assessed and recorded. Any feature(s) that result in exceeding 20 marks will not be evaluated in this check-up session.
 - d. Similar features will be assessed as one feature.
 - e. More specific implementation details can be found in **Appendix II**.
7. The entire project will be assessed at **THREE** milestone check-up sessions. More details of the project's timeline can be found in **Table 3**.
8. All the submissions, including the proposal and source code, will be uploaded online to the Blackboard system. Any extension request must be made at least **TWO** business days before the due date via mySI-net. For details about late submission, please refer to the items in Sec 5.3 in ECP.

Table 1: Three pre-defined projects and their respective features

Projects	Description	Desired Features
<p>Online learning system</p> <p><i>e.g.: learn.uq.edu.au</i> <i>or</i> <i>www.coursera.org</i></p>	<p>An online learning platform is a web-based virtual learning environment and learning management system that allows the educator to upload learning content and students/learner to view and download content.</p>	<ol style="list-style-type: none"> 1. Upload/Download video/pdf files. 2. Write descriptions and add tags. 3. Add comments. 4. Like the video/pdf files. 5. Add to collection. 6. Shares video/pdf files. 7. Pay for the courses. 8. Search for course (with filtering). 9. Enrol a course (notification mechanism)
<p>Online discussion forum</p> <p><i>e.g.: edstem.org/au or piazza.com</i></p>	<p>An online discussion platform is a learning management system that allows students to ask questions in a forum-type format. Instructors can moderate the discussion, along with endorsing accurate answers.</p>	<ol style="list-style-type: none"> 1. Post a question (in a particular category) 2. Rank answers 3. Search for questions 4. Reply and Rating answer 5. Recommendation (e.g., by most liked/viewed answer) 6. Bookmark/pin a post 7. Send Message/ Message Box
<p>Scientific data exploration system</p> <p><i>e.g.: rawgraphs.io</i></p>	<p>A scientific data exploration system is a web-based platform designed to assist researchers in exploring and analyzing large, complex data sets in various scientific domains. Taking medical data as an example, such a system allows users to generate graphic representation of literature medical data.</p>	<ol style="list-style-type: none"> 1. Search literatures based on keywords/ratings 2. Conditional search, e.g., year, venue, journal 3. Visualize search results using graphical visualization, e.g., word cloud 4. Export visualization results to PDF file 5. Donation to support the website, e.g., PayPal 6. Add comments/bookmark a paper 7. Share paper to other users through internal mail 8. Subscribe author

Table 2: Technical requirements for the project.

Tech Requirements	Description	Details
Use client-side technologies to develop the front end of your web application.	Client-side technologies are software technologies that run on the client-side or user-end of a web application. These technologies are executed by a web browser and are responsible for displaying and manipulating web content in a real-time manner. Client-side technologies are important in web development because they provide a rich and interactive user experience and examples include HTML, CSS, JavaScript, jQuery, Angular, React, AJAX, etc.	You are free to utilize any client-side technologies of your choice for developing the front-end web pages. It is important to note that this item is not graded, but exceptional front-end design will be taken into consideration as a merit factor in the student competition.
Use server-side technologies to develop the back end of your web application.	Server-side technologies are software technologies that run on the server-side or back-end of a web application. These technologies are responsible for processing requests from the client-side, generating dynamic content, and communicating with databases and other resources. Server-side technologies are important in web development because they enable the processing of data and information in a secure and efficient manner. Such technologies include but are not limited to PHP, JavaScript, Node.js, Python, CodeIgniter, MySQL/MariaDB, MongoDB, Redis, and more.	While you have the freedom to use any server-side technologies for back-end development, it is recommended that you use PHP and CodeIgniter as the server-side scripting language and framework. It should be noted that other solutions may receive limited support from the teaching team. More specific requirements can be found in Appendix II .
Adopt a design pattern in your project development.	An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context. Architectural patterns are similar to software design pattern but have a broader scope. Representative patterns include Model-view-controller pattern (MVC), three-tiered pattern, etc.	It is recommended that you use the MVC pattern for your project implementation, but you have the option to incorporate other design patterns, such as the three-tiered pattern. <u>Please note that failure to identify any pattern in your project will result in a zero mark.</u>
Deploy the web in cloud (UQCloud Zone, GCP, AWS, etc.)	UQcloud is the University of Queensland's cloud computing platform that provides a range of services to support teaching, research, and administration activities. UQzone is a part of the UQcloud platform that provides students with access to various	It is strongly recommended that you utilize your UQCloud zone to deploy your web project, although deployment on other popular public cloud platforms, such as AWS, GCP, or Azure, is

	online services, including learning management systems, student email, and online resources.	also acceptable. <u>It is important to note that developing solely in a local environment without deploying in the cloud is not permissible and will result in a grade of zero.</u>
Implement desirable features w.r.t. the selected project (the last column in Table 1). (in Appendix I) Implement basic and advanced features (in Appendix II)	Server-side scripting is a programming technique that enables the server to produce dynamic web pages by executing scripts on the server side. In this technique, the client sends a request to the server, and the server responds by generating and sending the requested web page to the client. The server-side script can access data from various sources, such as databases, file systems, or other servers, to generate the output. Common server-side scripting languages include PHP, Java, Python, and JavaScript. The functions implemented by server-side scripting on the server side include the following types: handling user requests, generating dynamic content, interacting with databases, handling authentication and authorization, providing APIs, etc. Generally speaking, server-side scripting plays a critical role in enabling dynamic web content, data storage, and access control on the server-side.	Beside the core features (ungraded), your project will have features that can be grouped into three levels: basic, intermediate, and advanced. The basic, intermediate, and advanced features will be assessed according to the marking criteria in Appendix II . You must demonstrate the implemented features twice, in Milestone 2 and Milestone 3 checkups, with each demonstration being capped at 20 points. You have the freedom to combine features whose points add up to 20 in each checkup.

Assignment Project Overview
 WeChat: cestbon688525@163.com
 Email: accoder – over 5250+ Help

Table 3. Project's Timeline

Milestone	Deadline	Description
First checkup	The deadline for submitting the proposal is 4 pm on Friday 24 March 2023, which falls in Week 5. The check-up session will start in Week 6.	It is important to note that the proposal should be implemented in the following milestones and deployed on UQ Zone. Failure to adhere to this may result in a lower grade. <u>Any major differences between the proposal and the implementation may also result in a lower grade.</u> The details of the proposal and the marking criteria can be found in Appendix I .
Second checkup	The deadline for submitting the source code for Milestone II is 4 pm on Friday 21 April 2023, which falls in Week 9. The check-up session will start in Week 10.	You are required to implement features with a total of 20 points. It should be noted that any feature(s) that result in exceeding 20 points will not be evaluated in this checkup.
Third checkup	The deadline for submitting the source code for Milestone III is 4 pm on Friday 12 May 2023, which falls in Week 12. The check-up session will start in Week 13.	For the last checkup, you must implement an additional set of features that are worth a total of 20 points. However, it is important to note that any features exceeding 20 points will not be evaluated in this checkup. These extra features will still be considered in the student competition based on their levels of difficulty and originality.

Assignment Project Quiz Exam Essay Help
 WeChat: cestbon6880
 Email: accoder_overseas@163.com

Appendix I. Individual Project Proposal

Proposal (5 marks)

A project proposal is a pre-project document that presents a brief and comprehensive summary of the plan for a particular project. It should provide an outline of the project's goals, objectives, timeline, and chosen technologies. For this assessment, you are required to write a proposal that **must NOT exceed two A4 pages** (excluding tables, images, and references) and it should include the following sections:

- **Introduction:** You are required to provide a comprehensive explanation of the main purpose of your project. **(0.5 marks)**
- **Objectives:** You are required to plan and design the key functional components of your project. You need to outline the desirable features (e.g., interactions with users, certain online business transactions, or special interactive visual effects) that you will implement. You may use the example feature table in the project specifications as a reference. **(2 marks)**
- **UI/UX Design:** You should include a clear depiction of the main flows and functions of your project. You can use wireframes or mockups to demonstrate the main pages, which can be created by hand or using any wireframe tools. **(2 marks)**
- **Timeline:** You are required to outline the major deliverables for each milestone. **(0.5 marks)**

Submission:

Your proposal must be submitted as a **PDF** on Blackboard by **4 pm on Friday 24 March 2023 in week 5**.

Marking Criteria:

The proposal will be evaluated based on the following criteria:

- **Completeness:** The proposal must provide enough information about the type of application that will be developed, the technology to be used, and the business function to be implemented.
- **Soundness:** The proposal should explain the reasoning behind the design components such as front-end elements, project structure, visual effects, interactive functions, and scripting language techniques.
- **Adaptiveness:** Your proposal must detail the deployment environment for the project and address the technical challenges associated with implementing the application.

Appendix II. Project Implementation Details

Implementation (40 marks)

The implementation phase of a project involves the actual development and creation of the proposed system based on the submitted project proposal. This phase includes coding, testing, and integrating different components of the system. The implementation is an important phase in project development as it brings the project from the planning stage to the actual development stage. The implementation phase in this course comprises two milestone check-up sessions, namely Milestone 2 and Milestone 3, where each session is worth a maximum of 20 marks. If the combined features exceed 20 marks, the score will be capped at 20. It is important to note that a failure to demonstrate the implemented features in Milestone 2 will result in a score of zero.

Technical Requirements:

- It is expected that you complete the assignment independently. The use of generative AI tools like ChatGPT during any implementation phase is prohibited. The use of popular libraries is permitted, provided that you have a thorough understanding of the functionality of each feature implemented in your project. It is crucial to acknowledge the tutor and reference any libraries used in the source code's comment section. If you cannot explain the code to the tutors, you may lose marks.
- You must deploy your project to a remote server (e.g., UQCloud zone, GCP, or AWS) to get it marked.
- Your project must follow a design architectural pattern, such as MVC or 3-tier architecture, which includes communication between front-end, back-end, and database components.
- It is mandatory to implement all core features listed in **Table 4** in addition to the graded features in Milestone 2 check-up session. However, please note that the core features will not be graded since they are mainly covered and practised in practical sessions.
- The graded features are categorized into basic (2 marks each), intermediate (3 marks each), and advanced (4 marks each) levels, and you have the flexibility to choose the combination of features to achieve a total of 20 marks in Milestone 2 and another 20 marks in Milestone 3.
- Each Milestone check-up session assesses the implementation for up to 20 points, so it is important to select features strategically. You cannot accumulate all the 40-point features in the final assessment milestone. If you fail to demonstrate the 20-point features in Milestone 2, your score in Milestone 3 will be capped at 20 points.
- Note that each feature will only be marked once. For instance, the registration function for both admin and users will be considered as one feature due to their similarities in implementation.

Table 4. Feature groups

Core features (compulsory but ungraded)	
1. Login	<ul style="list-style-type: none">• Server-side validation is required (e.g., informing the user if the username is not registered in the database or if the password is incorrect).• Each page should reflect the user's login status (e.g., the profile/logout button should appear only when the user has logged in).• Some content should only be visible to logged-in users.• Choose an appropriate HTTP request method (e.g., GET, POST, PUT, etc.) for client-server communication.
2. Registration	<ul style="list-style-type: none">• Username and email address should be checked to be unique.• Password strength should be checked.• The registered user data should be saved in the database.• Choose an appropriate HTTP request method (e.g., GET, POST, PUT, etc.) for client-server communication.
3. User Profile	<ul style="list-style-type: none">• Create a user profile interface that is only accessible when the user has logged in.• All the user's information should be fetched from the database.
4. Cookies	<ul style="list-style-type: none">• Use cookies in at least one of the features you implemented
5. AJAX	<ul style="list-style-type: none">• Use AJAX in at least one of the features you implemented.
Basic features (2 marks each)	
	<ul style="list-style-type: none">● Remember me: retain the user's login details even after the session expired.● Maintain scroll position of large HTML page when client return● Continuously loading data when scrolling● Favorites or rating● Search box autocompletion● Adding course (e.g. events, goods, pictures)

- Writing comments/reviews
- User profile updating (e.g., email, phone number)
- Display the user's current location on a map
- Image processing (e.g., resize, compress and add watermarks to images)
- Third-party API integration (e.g., sending an SMS to verify a phone number using SMS API)
- Basic file uploading: (e.g., profile image, food pictures, thumbnails)
- Using drag and drop to choose files for uploading (prerequisite: basic file uploading)
- Multiple files uploading at the same time (prerequisite: basic file uploading)
- Web Security (e.g., captcha, password encryption, data sanitizing)
- Online Payment Integration (e.g., PayPal, Stripe)
- Image and PDF Manipulation (e.g., sending receipt)
- **Others at the same difficulty level**

Intermediate features (3 marks each)

- Item Searching (e.g., images, meals or items, depending on the topic)
 - Each resulting search item should contain a link that redirects to a detailed page
- Email Verification
 - After registration, an email should be sent to the user for email verification.
 - This feature could be implemented using a random token or a verification code.
 - The user should be able to see the verification status, i.e. email verified or not.
- Forgot Password
 - Take account of security concerns.
 - This feature could be implemented using reset tokens or secret questions
- Shopping Cart/Wishlist/Calendar
 - The information should be stored in the database
 - Users should be able to add and remove items from the cart/wish list/calendar.
- **Others at the same difficulty level**

Advanced features (4 marks each)

- Data Mining (e.g., association rule mining)
- Management Dashboard (statistical information)
- Push Notification (e.g., when the item is sold, when the meal is delivered)
- Recommendation Algorithm (statistic based)
- Computer Vision Algorithm (e.g., detecting objects in images)
- Real-time Chatbot (self-implemented)
- **Others at the same difficulty level**

----- THE END -----