SULTAN QABOOS UNIVERSITY, COLLEGE OF SCIENCE, DEPARTMENT OF COMPUTER SCIENCE
COMP4701: WEB APPLICATION DEVELOPMENT- FALL 2025
PROJECT – PHASE 1 – 100 POINTS

Due date: Tuesday – 27th October 2025 @ 11:59pm

PART A [25 POINTS]: PROJECT SPECIFICATION

1) **Project Information [6 Points]:**
   Provide the project title, objectives, and motivation. Explain why this project was selected, emphasizing its real-world relevance, benefits, and the specific problems it aims to solve. Justify the need for your proposed solution and identify the potential users who will benefit from it.
2) **User Interviews [4 Points]:**
   Identify and interview at least two real users with different roles (e.g., customer, administrator). Summarize the findings and explain how these interviews helped define your project requirements.
3) **Project Requirements [6 Points]:**
   Specify the main system requirements as numbered points. Include both functional requirements (key features and behaviors of your website) and non-functional requirements (usability, performance, and reliability).
4) **Application Design [4 Points]:**
   Provide a clear architectural overview of your web application. Explain the interactions among the front-end, back-end, and database layers. Include a general description of the main website pages, their roles, and how data flows between them.
5) **Database Design [5 Points]:**
   Describe your database structure by identifying entities, attributes, and relationships. Present an ER or EER diagram and describe at least three database tables, including fields and constraints (primary keys, foreign keys, etc.).

PART B [55 POINTS] - IMPLEMENTATION: Develop a fully functional ASP.NET Core web application that integrates a modern front-end framework (such as React, Blazor, Angular, or jQuery) to deliver a dynamic, responsive, and user-friendly experience. Your application should meet the following requirements:

6) **Enumeration [2 Points]:**
   Define and implement at least one enumeration type that is used consistently throughout your web application. This enumeration should represent a meaningful set of constant values and help improve code readability and maintainability.
7) **C# Classes [6 Points]:**
   Create at least three well-structured C# classes representing the three database entities (DB tables). Each class should include appropriate fields, properties, constructors, and methods to encapsulate business logic. The classes will be used to maintain the project information as collections of the three classes in the Web pages below.
8) **Front-End Framework [10 Points]:**
   Use front-end frameworks (e.g., **React**, **Blazor**, **Angular**, and/or **jQuery**) with your ASP.NET Core application. You need to use of the main features in the framework to enhance user interaction, responsiveness, and dynamic content rendering across your application.
9) **Razor Web Pages [12 Points]:**
   Develop at least four Razor Pages using ASP.NET Core integrated with your chosen front-end framework. Each page should support form-based data input and output using various HTML controls (text boxes, dropdown lists, checkboxes, buttons, etc.). The pages must demonstrate data interaction with the C# class collections and support consistent navigation across the application.

10) **Business Logic [5 Points]:**
Implement the business logic layer as described in your project requirements. Use Razor Pages to handle user input, process the received data, and present meaningful results or feedback to users.

11) **Custom Layout [5 Points]:**
Create a unified layout for all web pages in your application using ASP.NET Core layout, style, and front-end frameworks. You need to use different types of layout sections and appropriate menu to all pages.

12) **Error Handling [5 Points]:**
Implement comprehensive error handling throughout your application. Use **C# Exceptions** in your Web Application to capture exceptions, and display user-friendly error messages in the UI. Ensure that the application gracefully handles common issues like invalid inputs, server errors, and missing data.

13) **Bootstrap Styling [5 Points]:**
Apply responsive and modern styling using **Bootstrap** framework to create a visually appealing interface. Customize Bootstrap components to match the design theme of your application, ensuring a cohesive look and feel across all pages. Include styles for forms, buttons, tables, and navigation elements (font, text color, background color, border style and color …).

14) **Form Validation [5 Points]:**
Use form validations with appropriate style and messages for each form and class in your application. Ensure at least four different types of validation (e.g., required fields, data format, string length, range validation) are applied across your forms.

PART C – [20 POINTS] COLLABORATION AND SUBMISSION

15) Report and Video Demo [8 Points]:

- **a.** Prepare a detailed project report that includes a cover page with project details and a table of contents generated using MS Word.
- **b.** **Use video demo** to demonstrating the main functionalities in your application, and how you have address all project requirements as stated above.

16) **Teamwork [5 Points]:**
The project tasks should be distributed evenly among team members. You need to specify the contribution for each team member in a tabular format. Clearly indicate each team member's contribution in a table, specifying tasks and the percentage of work completed by each member.

17) **GitHub Repository and Moodle Submission [7 Points]:**

- **a)** Each team member is required to create a **GitHub** account (https://github.com) to improve the teamwork.
- **b)** Create a **GitHub project** to share your project with all team members and set the repository visibility to **public** so that the teaching assistant can easily access, review, and test the application.
- **c)** You need to upload your report and code in both Moodle and **GitHib** accounts.