BMIT2023 WEB AND MOBILE SYSTEMS ASSIGNMENT

1.0 General Information

Objective: Apply the programming skills that you learnt from the course to develop a web application by using **ASP.NET MVC**, **Entity Framework**, **SQL Server Express** and other relevant web technologies.

Assessment Weight: **80%** of the coursework marks. **56%** of the overall marks.

Team Size: **FOUR (4)** students in a team.
Discuss with tutor if there are less or more students in a team.

Submission Mode: **Softcopy**. Please refer to **Section 6.0**.

Submission Date: **WEEK 7** = Short report + Project + Demonstration.

Feedback to Student: **WEEK 7**.

2.0 Project Architecture and Technologies

- The project must be built by using the recommended **.NET** version.
- The web application must be developed by using **ASP.NET MVC** and **C#**.
- The data layer must be implemented by using **Entity Framework** as the ORM tool.
- The database must be implemented by using **SQL Server Express**.
- Other relevant web technologies (e.g. jQuery, Bootstrap, etc.) can be used as needed.

[NOTE] External .NET, JavaScript and CSS libraries are **ALLOWED** to ease project development. However, better marks will be awarded if you code it yourself whenever possible.

3.0 Project Title and Requirements

PROPOSE YOUR OWN PROJECT TITLE. The project must be able to cover various programming aspects that you learnt from the course. For examples (but not limited to):

[NOTE] E-commerce system for selling and buying products is **NOT ALLOWED**.

- An attendance tracking system for tuition centers, schools or others.
- An e-learning system for tuition centers, schools or others.

- A reservation system for budget hotels, sport facilities or others.
- A society and event management system for schools, universities or others.
- An appointment system for consultation, coaching, pet grooming or other services.
- A ticketing system for interstate bus services, cinemas, events or others.
- A menu and ordering system for in-house dining in restaurants, canteens or others.
- A job recruitment system for posting, searching and application for job vacancies.
- A house or room rental system for house owners and tenants.
- A report and response system for schools, condos, etc. (e.g. reporting broken lamp).
- A management system for a specific type of business e.g. laundry, grocery store, etc.
- OTHER proposed titles.

[NOTE] Get **approval from tutor** for your project title, as certain project titles are less suitable and unable to cover various programming aspects that you need to implement.

The REQUIREMENTS of the project are:

3.1 Core Modules

Identify and implement the relevant **core modules** (and sub-modules) specific to your project title. Different project titles involve different business processes and flows. Your core modules should at least support all the basic and essential functions for the system to work properly in accordance to some common use cases for the project title. Implement appropriate database operations to support the relevant business processes and flows.

[NOTE] Experience with and borrow ideas from similar real-life systems. Simplify the use cases whenever appropriate, so that the project scope is feasible for a student assignment.

For example: A basic e-commerce system may have the following core modules. Please note that each core module can be further divided into the relevant sub-modules and functions.

• Security • Admin Maintenance • Member Maintenance • Category Maintenance • Product Maintenance	• Product Catalog • Shopping Cart • Ordering and Payment • Order Maintenance • Order History

[NOTE] Each student handles around 2-3 core modules (with sub-modules). No upper limit.

3.2 Input Validations

All input data must be properly **validated** by using both **client-side** and **server-side** input validation techniques. It is essential for the system to ensure that all input data are logically valid and the data integrity are always be maintained.

3.3 General System Security

Basic authentication and authorization should be implemented to grant different access privileges to different users based on their roles (e.g. ADMIN, TUTOR, STUDENT, etc.). You should identify the different user roles the system should have, and prevent unauthorized users from accessing protected pages, functions and data.

This includes implementing the SECURITY module to handle login, logout and other security functions to detect user identity and role. All other modules (their pages, functions and data) should be protected accordingly whenever necessary.

3.4 Data Layer

Design and implement a database (by using **SQL Server Express**) to store and maintain the relevant data tables and records required by the system. Design your database carefully based on the core modules as well as any additional features you plan to implement into the system.

Entity Framework must be used to implement the data layer (entity classes and data access classes). **DO NOT** mix SQL statements with your business logics. Make use of the data access methods provided by **Entity Framework** for accessing the database.

Store only the data the system needs. For example: If you do not plan to send postal mails to the users, there is no point to keep their postal addresses. However, you should ensure that the database contains **sufficient sample data** for demonstration purpose.

3.5 Additional Features

Via **self-study** and **self-research**, you should implement some additional features, functions or modules to enhance and improve the system (in the aspects user experience, functionality, performance, security, etc.). The basic and essential core modules ensure the system can at least work properly. Whereas additional features provide added values to the system.

Please ensure that the additional features that you implement are relevant, useful and well-integrated into the project. Some possible examples are listed below (but not limited to):

- Captcha or other anti-bot features (using external libraries or online services)
- Temporary login blocking (after 3 times of failed login attempts)
- Password reset (email with token-based link, etc.)
- User account activation or email verification (email with token-based link, etc.)
- E-receipt or e-ticket (generated in email or PDF format)
- Webcam integration (to capture profile photo, product photo, etc.)
- Advanced image processing (select and crop, flip and rotate, etc.)
- Multiple photos per record (one-to-many relationship between a record and photos)
- Multiple photos upload per form submission
- Batch insert, update and delete (enter or select manually, read from text files, etc.)
- User preferences and customizations (theme, navigation menu order, etc.)
- Google Maps integration (to display store location, mark and incident, etc.)
- QR code and barcode encoding and decoding (scan using webcam)
- Interactive onscreen seat selection (for bus, cinema, etc.)
- Interactive onscreen ingredient selection (for cake, food, etc.)
- 3rd-party payment API integration (Stripe, PayPal, etc.)
- UX improvement (e.g. drag and drop, keyboard shortcut, notification sound, etc.)
- SMS features (e.g. one-time password, notification, reminder, etc.)
- AJAX features (e.g. AJAX-based searching, sorting, paging, etc.)
- Real-time features (e.g. real-time chat, real-time order update, etc.)

- PWA features (e.g. desktop notification, offline access, etc.)
- Onscreen charts for reporting purpose (e.g. pie charts, column charts, etc.)
- Multi-language support for UI labels and texts (globalization and localization)
- OTHER additional features relevant to the project.

[NOTE] Each student handles around 3-4 additional features. No upper limit.

4.0 Short Report

General format of the short report is as follows:

Font	Calibri, 11pt
Line Spacing	Single line spacing (with a blank line between paragraphs)

You are required to prepare and submit a short report that contains the following sections:

4.1 System Modules Outline

Outline the modules, sub-modules, functions and additional features you plan to implement to the system. Highlight the additional features. You are **NOT REQUIRED** to describe the items, but you need to break down and group them logically. Indicate clearly the **person-in-charge (PIC)** for each module, sub-module, function and addition feature.

- Estimated length: 1-2 pages.

[NOTE] Refer to the short report template given for examples.

4.2 Entity Class Diagram

Attach in this section, the entity class diagram of the system. You may make use of the entity class diagram generated by **Entity Framework** in **Visual Studio** (you will learn how to do so).

- Estimated length: 1-2 pages.

4.3 Monetization Models

Propose appropriate monetization models for making your system sustainable (i.e. generating incomes or revenues). Some general software revenue streams are (but not limited to):

• License • Usage-Based • Advertising • In-App Purchases	• Subscription • Freemium • Transaction Fees • Support

[NOTE] Propose around 3-4 monetization models.

Assume that you are the **software provider**. For each of the proposed monetization models, describe how you would charge your clients (the companies who purchase the software) and/or users (the people who use the software). Provide a rough (but logical) estimate of the

total incomes or revenues you will generate from each of the proposed monetization models.

[NOTE] This is merely a proposal. You are **NOT REQUIRED** to implement the said monetization models into your system. For example: Suggesting "In-App Purchases" or "Transaction Fees" does not mean you must implement the relevant charging mechanisms into the system.

- Estimated length: 2-3 pages.

4.4 System Screenshots

Provide the screenshots for **ALL** pages. Group the screenshots **by students, then by modules**. Capture the screenshots at an appropriate zoom level (not too small to be viewed clearly).

5.0 Project Demonstration

A project demonstration session will be arranged, either **FACE-TO-FACE** or **ONLINE**, depending on the latest time schedule and venue availability. All functionalities and features need to be demonstrated during the project demonstration session.

6.0 Final Deliverables

The following final deliverables are required to be submitted in **WEEK 7**:

- A **softcopy** of the short report (**PDF** format).
- A **softcopy** of the .NET solution folder, which contains all the necessary resources such as ASP.NET MVC project, SQL Server Express database file, CSS files, JavaScript files, images, third-party libraries, etc. (**ZIP** format).

  [NOTE] Clean the project and delete the intermediate files and folders before you zip the .NET solution folder. This greatly reduces the project file size.

- A **softcopy** of the READ ME file (**TXT** format). Provide the login credentials for different user accounts of different roles for testing your system.

[NOTE] Tutor to decide the submission method (e.g. Google Classroom, Google Drive, copying files during project demonstration, etc.).

7.0 Student Ethics

7.1 Team Work

This project is based on **team work** and each team member should **contribute equally**. Any member who is irresponsible (e.g. no contribution, or contribute too little) should be reported and will be penalized. Please note some sections are to be assessed as **INDIVIDUAL MARKS**. The final project marks **may be different** among team members.

7.2 Plagiarism

All works submitted must be **original**. You can discuss with your friends. You can gather some ideas from web resources. However, the program must be **your own works**. You can teach your friends how to solve a certain programming problem, but do not program on behalf of them. **Do not share** your programs with others. Those who failed to explain their codes during project demonstration session may be suspected for plagiarism.
The student who copies **AND** the student who provides an opportunity for others to copy his/her programs, will both be penalized. **BOTH PARTIES** will receive **ZERO (0) MARK** for this project. The matter will also be reported to the faculty level for further disciplinary action.

[NOTE] Reuse and **adapt** practical layouts and demos are **ALLOWED**. But more credits will be given if there are obvious **enhancements** and **improvements**.

7.3 Late Submission

Late submission which is not supported by **VALID** and **CONCRETE** reason will be penalized:

No. of Day Late	Deduction
1-3 days	**10%** from the project marks earned
4-7 days	**20%** from the project marks earned
After 7 days	**Reject assessment** and **0%** awarded

8.0 Assessment Criteria

Course Learning Outcomes (CLO)	
CLO2	Build web and mobile systems by using the practical skills learned. (P3, PLO3)
CLO3	Propose appropriate monetization models for web and mobile systems. (A3, PLO10)

Marks allocated for each sub-component are as follows:

No	Criteria	Marks
1.	**Short Report** (CLO3) - System Modules Outline (5%) - Entity Class Diagram (5%) - Monetization Models (10%)	20%
2.	**System Implementation** (CLO2) - General Web Architecture (5%) - Presentation Layer (5%) - Data Layer (10%) - General System Security (10%) **** INDIVIDUAL MARKS **** - Core Modules (30%) - Additional Features (20%)	80%

TOTAL :	**100%**

[NOTE] Refer to **marking rubric** for detailed assessment criteria.