|  |  |  |
| --- | --- | --- |
|  | TARC | **BMIT2013 WEB-BASED INTEGRATED SYSTEMS**  **Assignment: Web-Based System Development** |

**ADVICE:** Read the project specifications and requirements **thoroughly**.

|  |
| --- |
| **1.0 General Information** |

**Objective:** Apply the web programming skills that you learned from the course **BMIT2013** to develop a secured data-driven web-based system by using **HTML**, **CSS**, **JavaScript (jQuery)**, **PHP**, **MySQL** 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 7.0**.

**Submission Date:** **WEEK 13 OR 14** = Slide + Web project + Database export.

To be decided by tutor depending on the number of teams.

**Feedback to Student:** **WEEK 13 OR 14**.

|  |
| --- |
| **2.0 Web Technologies and Restrictions** |

|  |
| --- |
| **2.1 Web Technologies** |

* Use **HTML5** and **CSS3** for the frontend.
* Use **jQuery** instead of plain JavaScript whenever possible.
* The project must be built by using **PHP 8.2.12** or above.
* The database must be implemented by using **MySQL**.
* The database programming must be implemented by using **PDO**.

|  |
| --- |
| **2.2 Restrictions** |

The followings are **NOT** allowed:

* Ready-made HTML and CSS layout templates.
* PHP frameworks (e.g. Laravel, Yii and CodeIgniter).
* CSS frameworks (e.g. Bootstrap, Tailwind and Foundation).
* JavaScript frameworks (e.g. React, Angular and Vue).

**[NOTE]** Small external PHP, JavaScript and jQuery libraries are **ALLOWED** to ease your project development (and it is sometimes unavoidable). However, better marks will be awarded if you code it yourself whenever possible.

|  |
| --- |
| **3.0 Web Programming Conventions** |

The **web programming conventions** used in the course must be strictly followed:

* **Reuse** the base PHP and JavaScript libraries created in the practical in your project.
* **Modify** or **extend** the base PHP and JavaScript libraries if necessary.
* Create your own **reusable library functions** if you are doing something repeatedly.
* Use **HTML helpers** to generate input controls, error messages, etc.
* Create your own reusable **HTML helpers** if necessary.
* **Avoid** inline JavaScript for event-handling. Use **jQuery** event-handling instead.
* If you refer to external references, **adapt** the codes to match our conventions.
* You can **reuse the layouts** created in the practical and **enhance** further.
* You can **reuse the demos** created in the practical and **enhance** further.
* **Follow** other programming styles as what we adhere in the practical.

**[NOTE]** This is **NOT** meant to stop you from being creative. Instead, this is to prevent you from copying codes blindly from external references. You can refer to external references, but please **understand** how the codes work, then **modify** and **adapt** the codes to match our conventions. Codes that cannot be explained and do not follow conventions will be penalized.

|  |
| --- |
| **4.0 Specifications and Requirements** |

You are to develop an **ONLINE SHOPING WEB-BASED SYSTEM** for a fictional company, selling any product of your choice. The requirements of the web-based system are:

|  |
| --- |
| **4.1 Overall Requirements** |

* Organize your pages and resources meaningfully into **sub folders**.
* **Server-side input validations** must be implemented with proper error messages.
* **Authorizations** (web security) must be implemented to protect relevant pages.
* The **user interface** should be clean, neat and user-friendly.

**[NOTE]** This is **NOT** a web design course. A simple but user-friendly user interface is sufficient. Put your focus on improving the system modules, functions, features and user experience.

**[NOTE]** Static pages with hardcoded information do not bring credits. Please focus on creating dynamic pages using JavaScript (jQuery) and PHP. Program your pages with database access, so that the information is retrieved from database rather than hardcoded.

|  |
| --- |
| **4.2 Database Implementation** |

Implement your database by using **MySQL**. Create the necessary tables and columns (with proper primary keys, foreign keys, data types, lengths, relationship and other settings). Add more tables and columns to suit your additional modules and functions. Follow the **naming conventions** as used in practical (e.g. lowercase names, underscore to separate words, do not repeat table name in column name, etc.). Think as a programmer.

**[NOTE]** Ensure you pre-insert sufficient **sample data** for project demonstration purpose.

|  |
| --- |
| **4.3 Basic Modules and Functions** |

The following basic modules and functions **MUST BE** implemented. Evenly distribute the tasks among team members:

|  |  |
| --- | --- |
| **Security**   * Roles: Admin + Member * Login + Logout * Password Hashing * Password Reset   **User Profile**   * Profile Update * Password Update * Profile Photo Upload   **Member Maintenance**   * Member Listing + Detail (Admin) * Basic Searching (Admin) * Member Registration * Profile Photo Upload | **Product Maintenance (Admin)**   * Product Listing + Detail * Basic Searching * Product CRUD * Product Photo Upload   **Shopping Cart (Member)**   * Product Listing + Detail * Basic Searching * Shopping Cart * Checkout + Create Order   **[NOTE]** Payment is excluded from basic.  **Order Maintenance**   * Order History + Detail (Member) * Order Listing + Detail (Admin) |

|  |
| --- |
| **4.4 Additional Modules and Functions** |

Through **self-study** and **self-research**, each team member should implement some additional modules and functions to enhance and improve the web-based system (in the aspects of user experience, functionality, performance, security, etc.). For example (but not limited to):

|  |  |
| --- | --- |
| * Category Maintenance + CRUD * Admin Maintenance + CRUD * Order Cancellation (Member) * Order Status Update (Admin) * Payment (Fake - Data Entry Only) * Payment (Real - Using Stripe API) * E-Receipt (Email or PDF) * Shipping Address Handling * Add to Favorites or Wishlist * Discount Voucher Handling * Reward Point Handling * Product Stock Handling * Low-In-Stock Alert * Product Rating + Review * Record Listing (Table View + Photo View) * Product Filtering (by Category) * Product Filtering (by Price Range) * Filtering, Sorting and Paging (Combined) * 1 Product = Multiple Photos * Multiple Photos Upload * Product Photos Sliders (Dynamic) | * Drag-and-Drop Photo Upload * Product Video Integration (YouTube) * Webcam Integration (Capture Photo) * Image Processing (Flip, Rotate, etc.) * Batch Insertion (From Text or CSV Files) * Batch Updating (e.g. Increase * Batch Deletion * User Email Verification (Email) * Captcha Integration (3rd-Party Library) * Temporary Login Blocking (3 Attempts) * Block + Unblock User Account (Admin) * Remember Me (Retain Login Session) * SMS Integration (Security Code, etc.) * Google Maps Integration (Store Location) * Generate + Scan QR Code * Permanent Shopping Cart (for Member) * Remember User Preference (e.g. Theme) * Data Charts (Pie, Column, Bar, etc.) * Top Selling Products (e.g. Top 5) * Real-Time Chat * AJAX Integration |

|  |
| --- |
| **5.0 Slide (ERD and Screenshots)** |

You are required to prepare and submit a **slide** (using Microsoft PowerPoint or Google Slides) that contains the following sections:

* **Cover Page:** System title, program, tutorial group and team members.
* **Entity Relationship Diagram (ERD):** Generate using PHPMyAdmin.
* For each team member:
  + **Sub Cover Page:** Member name, list of modules and functions handled.
  + **Screenshots:** For all pages handled with page titles.

Refer to the plain **slide template** given for the overall structure. You may use different design or theme for the slide.

**[NOTE]** The slide is mainly to record your ERD, modules, functions and screenshots (instead of writing a formal report). It may **NOT** be used during the project demonstration session.

|  |
| --- |
| **6.0 Project Demonstration** |

A project demonstration session will be arranged in **WEEK 13 OR 14**. All modules, functions and features need to be demonstrated during the project demonstration session. **Technical questions** will be asked to determine your understanding. Be sure you understand your codes.

|  |
| --- |
| **7.0 Final Deliverables** |

The following final deliverables are required to be submitted in **WEEK 13 OR 14**:

* Slide (**PDF** format).
* PHP web project folder (**ZIP** format).
* Database export file (**SQL** format).

**[NOTE]** Tutor to decide the submission method (e.g. Google Classroom, Google Drive, etc.).

|  |
| --- |
| **8.0 Student Ethics** |

|  |
| --- |
| **8.1 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. The final project marks **may be different** among team members.

|  |
| --- |
| **8.2 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**.

|  |
| --- |
| **8.3 Late Submission** |

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

|  |  |
| --- | --- |
| **No. of Day Late** | **Deduction** |
| **1** day | **20%** from the project marks earned |
| **2** days | **50%** from the project marks earned |
| **3** days | **100%** from the project marks earned |

|  |
| --- |
| **9.0 Assessment Criteria** |

|  |  |
| --- | --- |
| **Course Learning Outcomes (CLO)** | |
| CLO2 | Build secured data-driven web-based systems by using the practical skills learned. (P3, PLO3) |
| CLO3 | Demonstrate enhancements to web-based systems through self-study and self-research. (A3, PLO9) |

Marks allocated for each sub-component are as follows:

|  |  |  |
| --- | --- | --- |
| **No** | **Criteria** | **Marks** |
| 1. | **Core Implementation (CLO2)**  - User Experience (5%)  - Database Design (5%)  - Web Programming Convention (10%)  - Web Security (10%)  - Input Validations (10%)  - Basic Modules and Functions (30%) | 70% |
| 2. | **Advanced Implementation (CLO3)**  - Additional Modules and Functions (30%) | 30% |
| **TOTAL :** | | **100%** |

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