**Department of Information and Technology**

**(HKIIT)**

Higher Diploma in Software Engineering (IT114105)

Module Name : Internet & Multimedia Applications Development

Module Code : ITP4523M

Submission Deadline : Phase 1: 12th teaching week of Semester 2

Phase 2: 7th teaching week of Semester 3

Hand in Methods : *To be announced by the lecturer*

This Group Project : **40% of total module marks** (*also it is part of EA components)*

**The result of EA will not be counted if you do not meet the minimum 70% attendance requirement (if any) governed by the general academic regulations of your programme/course unless approval of the campus principal has been granted.**

1. Objectives

In this project, students are asked to:

* build a web application which provides different functions for *Customer* and *Staff.*
* apply software development skills to develop a website which is user-friendly, interactive, robust and easy to maintain.
* apply the knowledge that you learned in this module to solve the tasks. (i.e. HTML, CSS, JavaScript, PHP, SQL commands and Python)

1. A simplified procedure to show how the web application will be used

There are *two user roles* for the management System:

* 1. *A Customer* can make the orders and retrieve the order records.
  2. *A Staff* can manage products, material items and update the order records.

1. Driving Question

You are the CEO of Smile & Sunshine Toy Co. Ltd. Briefly explain how a centralized management system could help the company to promote sales and increase profit.

|  |
| --- |
|  |

1. Design for *Customer* (Interface Design: 15 marks; Function: 30 marks)

|  |  |
| --- | --- |
|  | Done By |

* 1. Make the orders

Managed to make orders for the products supplied by our company.

View product information

Allow the user to easily select products available from various products (display products when stock quantity is greater than zero).

\* Sort the list of products *wisely* to facilitate smart purchasing

Required information for creating an orderas below:

1. Customer ID
2. Customer Name
3. Customer Password
4. Customer Telephone
5. Customer Address
6. Company Name

\* *Customer ID* should be generated automatically by the system as Primary Key

\* Update the *stock item quantity* after an order is created

* 1. View the order records

Managed to view any order records created by the customers. The system should display the following necessary information for each order record.

Required information for order record page:

* 1. Order ID
  2. Order Date
  3. Product ID
  4. Order Quantity
  5. Order Cost
  6. Customer ID
  7. Order Delivery Date
  8. Order Status

Function requirement:

\* List the orders in ascending or descending order by at least TWO columns selectable by the user

* 1. Update Customer profile’s information

Managed to update the customer profile’s information.

**Only allow the use to update the following information:**

1. Password
2. Contact Number
3. Customer Address
   1. Delete Order record

Delete an order record from the *Orders* table and related records in *Material* table. Update the *material Quantity item* after an order is deleted

Function requirement:

* A confirmation message should be displayed to let the user decide whether the selected order should be deleted or not.

\* The order can only be deleted two days before the receipts delivery date.

1. Python Plug-in: Price Converter (Function: 10 marks)

Develop a simple Python flask application running on **port 8080** to determine currency conversion based on the *order total amount* (the default order total amount is US dollars). The **RESTful API** should accept the HTTP GET request and send the *required response* (in JSON format) from the Python program.

|  |  |
| --- | --- |
| **URL Request** | **/cost\_convert/*<amount>*/*<currency>*/*<rate>*** |
| **Input Parameter** | ***<amount>*** Numeric value, indicating the total amount of original price in **US dollars**. (must be a *positive number*) ***<currency>*** Must enter **"EUR"** or **"HKD"** or **"JPY"**.  ***<rate>***Exchange Rate (must be a *positive number*). |
| **Response in JSON** | { "result":"accepted", "converted\_amount":\_\_\_\_\_\_\_  }  *or*  { "result":"rejected", "reason":\_\_\_\_\_\_\_  } |

The following table shows the meaning of each **Exchange Rate** given the **Currency**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Currency** | **JPY** | **EUR** | **HKD** |
| **Exchange Rate** | **110** | **0.82** | **7.8** |
| **Meaning** | 1 USD = 110 JPY | 1 USD = 0.82 EUR | 1 USD = 7.8 HKD |

Below are some example requests and the corresponding responses :

|  |
| --- |
| **HTTP GET request**  **Response in JSON format** |
| <http://127.0.0.1:8080/cost>\_convert/200/CHF/110  { "result":"rejected", "reason":"Error : Currency must be 'HKD' or 'EUR' or 'JPY'" } |
| <http://127.0.0.1:8080/cost>\_convert/100/HKD/-7.8  { "result":"rejected", "reason":"Rate must be a positive number" } |
| <http://127.0.0.1:8080/cost>\_convert/-10/JPY/110  { "result":"rejected", "reason":"Amount must be a positive number" } |
| <http://127.0.0.1:8080/cost>\_convert/100/EUR/0.82  { "result":"accepted", "converted\_amount":82 } |
| <http://127.0.0.1:8080/cost>\_convert/100/HKD/7.8  { "result":"accepted", "converted\_amount":780 } |

1. Design for *Staff* (Interface Design: 15 marks; Function: 30 marks)

|  |  |
| --- | --- |
|  | Done By |

* 1. Insert products’ information

Managed to insert the product with the following information

Required Item information:

1. Product ID
2. Product Name
3. Product Description
4. Product Image
5. Single Product cost
6. Material ID
7. Material Quantity

\**Product ID* should be generate automatically by the system as Primary Key.

\* A product may consist of one or multiple materials.

* 1. Insert Materials’ information

Managed to insert the Material with the following information

Required Item information:

1. Material ID
2. Material Name
3. Physical Quantity
4. Reserved Quantity
5. Unit
6. Re-order level

\**Material ID* should be generated automatically by the system as Primary Key.

\**Reserved Quantity* should be equal or greater than *Physical Quantity.*

* 1. Update order records and related material record

Managed to update any order. The Staff should be able to:

View any orders and materials’ information.

Update the status of the selected order to "accepted" or "rejected".

Modify the order quantity of any orders.

Update the Reserved Quantity of related materials used in any orders.

Required information for managing an order as below:

1. Order ID
2. Order Date
3. Product ID
4. Product Image
5. Order Quantity
6. Total Order Amount
7. Customer’s Contact Name
8. Customer’s Contact Number
9. Delivery Address
10. Delivery Date
11. Order Status
12. Material Name
13. Used materials in the order
14. Material Physical Quantity
15. Material Available Quantity
16. Unit of the material
17. Material Low Stock Alert
    1. Generate report

Managed to view the statistics which can help the Staff to wisely manage materials.

Required information for report page:

1. Order ID
2. Product Name
3. Product Image
4. Total number for each order item
5. Total sales amount ($) for each order item \*

\*total sales amount = price \* order quantity

* 1. Delete product

Delete a product from the product list.

Function requirement:

* A confirmation message should be displayed to let the user decide whether the selected product should be deleted or not.

\*A product can be deleted only when the product has no existing related orders.

1. Form your project group

Each student needs to form a project group, **the maximum number of students in each group is 2.** Strongly recommend you to form a group to complete this project as you can benefit from sharing skills/codes amongst your members, and you can learn to plan, coordinate, and integrate work done by each member. Study carefully the given ERD and table structures before you start the implementation.

1. Additional requirements of your project
   1. Your web site should only use PHP as the server-side programming language (i.e. not ASP, ASP.NET, JSP, servlet etc.), however, you may use JavaScript and CSS for specific purposes. The database server used must be mySQL (version 5.0 or above).
   2. In your PHP code, you must ensure to use the following *parameter values* for the following mySQL database functions :  
       $conn = *mysqli\_connect*($hostname, $username, $password, $database);  
      set to the values below in a PHP script which is *shared by* the web pages :  
       $hostname = "**127.0.0.1**";  
       $database = "**projectDB**";  
       $username = "**root**";  
       $password = "";
2. Items to submit (Phase 1) (30% of total project marks)

Submit all UI design using CSS and HTML. Submission deadline will be announced by the

lecturer.

1. Items to submit (Phase 2) (70% of total project marks)
   1. A ***CD-ROM*** or ***DVD-ROM*** which stores a ***softcopy of all files*** for the whole web site. All files must be stored in non-compressed format (no .zip or .rar files please!)
   2. provide a SQL script file ***CreateProjectDB.sql*** to let the lecturer to re-create the database and test data
   3. for the SQL script file ***CreateProjectDB.sql***, it must contain *CREATE TABLE* commands to setup the database tables in **projectDB** database. Include necessary *INSERT* statements to add additional sample records you want to provide. The following is a sample SQL script :

drop database IF EXISTS **projectDB**;

create database **projectDB** character set utf8;

use **projectDB**;

You must specify the **InnoDB** engine for a database table :

ENGINE = **InnoDB**

Full explanation of different ***mySQL database engines*** :

<http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html>

drop table IF EXISTS **Users**;

Create table **Users** (

userName Varchar(30) NOT NULL,

userPswd Varchar(10),

Primary Key (userName)) **ENGINE = InnoDB**;

INSERT INTO **Users** (userName, userPswd) VALUES

('admin1', 'secret1'),

('admin2', 'secret2');

* 1. a ***demonstration*** of your completed web site should be recorded by   
     a *30-day free-trial software Camtasia Studi*o 8  
     (<http://discover.techsmith.com/try-camtasia/clkn/https/www.techsmith.com/download/camtasia/>).   
     You should save different parts of your demonstration into different **.mp4** files. In a *Word* document named **video\_list.docx**, briefly describe the main content of each demo video file you have created. The video files will facilitate the lecturer to have in-depth evaluation of your web application. Here are some online tutorials for **Camtasia Studio 8** <http://www.techsmith.com/tutorial-camtasia-current.html> :  
     ***Getting Started: 1 - Record Full Screen*** :   
     <http://www.techsmith.com/tutorial-camtasia-record-full-screen.html>   
     ***Produce and Share an MP4 Video*** *:*   
     <http://www.techsmith.com/tutorial-camtasia-produce-and-share-mp4-video.html>

1. Assessment criteria of your project
   1. The functions implemented can perform correctly in *general* and *special* situations
   2. *Enough detail* of database records and extensive *data validation*
   3. Techniques used to promote *code reusability* (e.g. share common PHP/JavaScript/CSS files amongst different web pages) and *standardize the user-interface* of the web pages
   4. Coding style (e.g. indentation, meaningful variable names, modularity by user-defined functions etc.) and meaningful *comment* is added to program codes
   5. *Creativity* to enhance implemented functions so that they become easy to use, more interactive to the users or can handle some problems in real life situation
   6. Screen design and overall *quality of the integration* of different functions in the web site
2. A guideline for web development

It is a step-by-step approach I suggested for inexperienced web developers to develop the web site easily:

* decide what information to be displayed and design a number of web pages in HTML code (not PHP code at this stage) to display the information
* think about the site structure by creating different sub-folders to store files of different purposes (e.g. **images** folder to store image files, **style** folder to store CSS files, **Connections** folder to store files which define the settings for database connection) and design the linkages between the pages. You can easily view the site structure using DW8's site map view
* create HTML web pages (do not add JavaScript so soon) and design the layout with HTML codes and CSS rules. It is a good practice to check your .html files can pass the XHTML validation after completing a .html file
* when using CSS, it is preferred to create *external CSS files* (stylesheets) which can be reused in other web pages, so that other pages can have consistent formatting
* use DW CS6's template features which can help you to create a new page with a standard layout and also it provides common editable regions for web pages created from the same template.
* define frameset(s) and navigation bar or menu to link up different pages
* add JavaScript code to produce more interactive behaviors (such as validate data in the form, highlight a table row with different background color when the mouse moves over a table row). It is preferred to use *external JavaScript file* which will be reused in other web pages
* replace hyperlink text with image / button to beautify the links. Dreamweaver can help you to create nice Flash buttons easily
* finally, it comes to the hardest work, that is to convert some of the HTML codes into PHP codes in order to generate dynamic contents from data extracted from database, cookie and PHP pre-defined arrays ($\_POST, $\_GET, $\_COOKIE, $\_SESSION, $\_FILES, $\_SERVER etc.)

1. Penalty for plagiarism

* Each student needs to submit his/her own work. *Plagiarism* (抄襲) will be treated seriously.
* All group projects that have been found involved wholly or partly in plagiarism (no matter these projects are from the original authors or from the plagiarists) will score ZERO marks. Furthermore, disciplinary action will be followed.

**Late submission will receive ZERO marks**