|  |  |  |
| --- | --- | --- |
|  | **FACULTY OF COMPUTING, ENGINEERING and SCIENCE** | Final mark awarded:\_\_\_\_\_ |

**Assessment Cover Sheet and Feedback Form 2015/16**

**for students on University of South Wales regulations only**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  IS2S521 | Module Title:  DB Development | | Module Lecturer:  Gaylor Boobyer |
| Assessment Title and Tasks:  Web Database Application Development | | | Assessment No. 2 |
| No. of pages submitted in total including this page:  N/A | | | Word Count of submission  (if applicable): N/A |
| Date Set:  19-02-16 | | Submission Date:  22-04-16 | Return Date:  20-05-16 |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Centre on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Centre to be consulted).  Work should be submitted as detailed in your student handbook. You are responsible for checking the method of submission. | |
| **You are required to acknowledge that you have read the above statements by writing your student number (s) in the box:** | Student Number(s):  14568811 |

|  |  |  |
| --- | --- | --- |
| **Section B : Marking and Assessment** | | |
| This assignment will be marked out of 100%  This assignment contributes to 25% of the total module marks.  All coursework marks gained in this module are combined to form an overall element of assessment. In any coursework referral, the overall element will be capped at 40%. | |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Learning Outcomes** |  | |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <http://icis.glam.ac.uk>):  This assignment addresses the following learning outcome(s) of the module:  LO1. To analyse the requirements for an information system and produce a conceptual/logical design that satisfies those requirements.  LO2. To use a given design to develop a database system using current technologies. |  | |
| **Marking Scheme** | **Marks Available** | **Marks Awarded** |
| 1. Database creation | 5 |  |
| 1. Conceptual data model | 10 |  |
| 1. Data capture pages and website navigation | 25 |  |
| 1. File maintenance and login functionality | 30 |  |
| 1. Customer history page | 15 |  |
| 1. Design Improvements | 15 |  |
| **Total** | **100** |  |

|  |
| --- |
| **Section C : Marker’s Feedback** |
| **See Assessment Criteria and Feedback below.** |
| **All marks are subject to confirmation by the Board of Examiners** |

|  |
| --- |
| **Assessment Task:**  **Tasks**  You are required to demonstrate your understanding of web database development by using PHP/MySQL to complete a number of tasks. **You are not allowed to use a WYSIWYG editor.** Doing so mayresult in your submission not reaching the standard required for a pass.    **Task 1**  Create a database using the supplied script. The database will be used to record information about electricity customers.  **Task 2**  Produce a conceptual data model (ER diagram) which represents the structure and relationships of the database. The model should conform to the diagramming standards taught in the earlier part of the module. If in doubt please refer to the notation given on the inside cover of *Database Systems* by Connolly and Begg.  **Task 3**  Use HTML/PHP to create:   1. data entry forms for each of the tables in the database; 2. web pages that display the data in the tables; and 3. appropriate website navigation.   **Task 4**  Use HTML/PHP to create web pages to:   1. delete rows from at least one table in the database; 2. update data in at least one table in the database; and 3. provide normal login behaviour.   **Task 5**  Use HTML/PHP to create a web page to display a list of past and current customers for each property. The format of the list is given below. Please note its following features:   1. The list should be sorted in ascending order of street, property no and occupancy start date. 2. Column headings are different to the ones in the database. 3. Dates are output in the format dd-Mmm-yy.     **Task 6**  Improve the design of the site. This could include adding security features to protect your site from unauthorized access.  **Deliverables**  1. A web database application uploaded to your student space.  2. A zip file uploaded to Blackboard that contains:   1. a document with: 2. the link to the application’s home page; 3. the usernames and passwords needed to gain entry to the application and to the database via phpMyAdmin; and 4. the conceptual data model asked for in Task 2. 5. a folder containing your HTML and PHP source code; and 6. a text file containing the SQL needed to recreate your database.   Just To Clarify – your work is to be uploaded to your student space and not handed-in on a memory stick/USB drive.  **SQL to Create the Database**  DROP TABLE IF EXISTS Occupancy;  DROP TABLE IF EXISTS Property;  DROP TABLE IF EXISTS Customer;  #  # The Property table  #  CREATE TABLE Property (  propertyID int(6) NOT NULL,  propertyNo char(3) NOT NULL,  propertyStreet char(30) NOT Null,  CONSTRAINT Property\_propertyId\_pk PRIMARY KEY (propertyId) );  #  INSERT INTO Property (propertyID, PropertyNo, PropertyStreet) VALUES  (1, 5, 'Keynsham Place'),  (2, 30, 'Westminster Close'),  (3, 84, 'York Road'),  (4, 8, 'Warwick Crescent'),  (5, 42, 'Paddington Green'),  (6, 30, 'Victoria Road'),  (7, 123, 'Bristol Street'),  (8, 23, 'Cardiff Court'),  (9, 17, 'Brecon Road');  #  # The Customer table  #  CREATE TABLE Customer(  customerID int(6) NOT NULL,  firstName char(15) NOT NULL,  lastName char(25) NOT NULL,  CONSTRAINT Customer\_customerId PRIMARY KEY (customerId));  #  INSERT INTO Customer (customerID, firstName, lastname) VALUES  (1, 'Evan', 'Evans'),  (2, 'Jane', 'Rees'),  (3, 'Michael','Williams'),  (4, 'Rebecca','Evans'),  (5, 'Richard','Jones'),  (6, 'Harry', 'Simpson'),  (7, 'John', 'James'),  (8, 'Brenda', 'Watson'),  (9, 'Jill', 'Charles'),  (10, 'Helen', 'Potts'),  (11, 'Jane', 'Potts'),  (12, 'Geoff', 'Carter'),  (13, 'Helen', 'Rees'),  (14, 'Brian', 'Glover'),  (15, 'Dean', 'Richards'),  (16, 'Mora', 'Moon');  #  # The Occupancy table  #  CREATE TABLE occupancy(  occupancyID int(6) NOT NULL auto\_increment,  propertyID int(6) NOT NULL,  customerID int(6) NOT NULL,  occupancyStart date NOT NULL,  occupancyEnd date,  CONSTRAINT Occupancy\_occupancyID\_pk PRIMARY KEY (occupancyID),  CONSTRAINT Occupancy\_propertyID\_fk FOREIGN KEY (propertyID) REFERENCES  Property (propertyID) ON DELETE CASCADE ON UPDATE CASCADE,  CONSTRAINT Occupancy\_customerID\_fk FOREIGN KEY (customerID) REFERENCES  Customer (customerID) ON DELETE CASCADE ON UPDATE CASCADE);  #  INSERT INTO Occupancy (propertyID, customerID, occupancyStart, occupancyEnd) VALUES  ( 1, 1, '2000-03-01', '2006-02-21'),  ( 2, 2, '2000-03-01', '2010-02-16'),  ( 3, 3, '2000-03-01', ''),  ( 4, 4, '2000-03-01', '2002-11-08'),  ( 5, 5, '2000-03-01', '2010-08-25'),  ( 6, 6, '2000-03-01', ''),  ( 7, 7, '2000-03-01', '2011-04-11'),  ( 8, 8, '2000-03-01', '2007-04-11'),  ( 9, 9, '2000-03-01', ''),  ( 4, 10, '2002-11-09', ''),  ( 1, 11, '2006-02-22', '2007-08-31'),  ( 8, 1, '2007-04-12', '2011-03-23'),  ( 1, 12, '2007-09-01', '2009-06-10'),  ( 1, 4, '2009-06-11', ''),  ( 2, 13, '2010-02-17', ''),  ( 5, 14, '2010-08-26', ''),  ( 8, 15, '2011-03-24', ''),  ( 7, 16, '2011-04-12', ''); |

# Marking Scheme

1. Database creation (/5)

Award **5 marks** if there are three tables present that conform to the table definitions supplied in the script.

2. Conceptual data model (/10)

Award **10 marks** for a perfect model. The cardinality of the relationships should be correct. Award **5-7 marks** for a structurally-correct model with some minor notation errors. Award up to **4 marks** for a model with major notational and/or structural errors.

3. Data capture pages and website navigation (25)

Award **3 marks** for each data capture page and display page that functions correctly. There should be two pages for each of the three tables (**18 marks in total**). Award up to **7 marks** for the user interface design. A design closely based on the Birthday app would attract up to **3 marks**. A design at this level might include changes of colours but little else.

4. File maintenance and login functionality (/30)

Award **7 marks** for evidence of at least one correct deletion page. Award **7 marks** for evidence of at least one correct update page. Award up to **10 marks** for a working login page. Award up to **6 marks** for a working user-registration facility.

5. Customer history page (/15)

Award **5 marks** for a page that displays customer id and property id but not customer name and property address. Award **7 marks** if either customer name is displayed instead of customer id or property address is displayed instead of property id. Award **10 marks** if both customer name and property address are displayed i.e. the SQL SELECT statement includes the appropriate joins. Award up to **5 marks** for formatting.

6. Design improvements (/15)

Award up to **10 marks** for security features e.g. redirection of unauthorised users back to the login page. Award a further **5 marks** for other improvements.

1. University Academic Integrity Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from Campus Advice Shops [↑](#footnote-ref-2)