MANCHESTER METROPOLITAN UNIVERSITY

DEPARTMENT OF COMPUTING & MATHEMATICS

ASSIGNMENT COVER SHEET

COURSE: CDTN

UNIT: 6G5Z1003 Web and Mobile Development

LECTURER: Paul Marsden/Kristopher Welsh/John Darby/Nick Whittaker

ASSIGNMENT: 1

ASSIGNMENT TYPE: Individual

ISSUE DATE:

HAND-IN DATE: 11.59 pm 18th December 2015

Marks released: Saturday 16th January 2016

PROCEDURE FOR HANDING IN WORK:

Moodle submission. See assignment specification.

PENALTIES FOR LATE HAND-IN:

See Regulations for Undergraduate Programmes of Study

EXCEPTIONAL FACTORS AFFECTING YOUR

PERFORMANCE/ABSCENCE:

See Regulations for Undergraduate Programmes of Study

PLAGIARISM

Students found guilty of cheating, plagiarising (copying) or seeking to gain an unfair advantage will face severe penalties. See the Faculty Student Handbook for further information.

ASSESSMENT CRITERIA:

See section at the end of the brief.

LEARNING OUTCOMES COVERED: 1,2,3,4

Scenario

CongoCorp is a retailer specialising in music recordings and as such has a small database of the current titles it stocks. Congo currently operates out of a shop front in Newton Heath but now wishes to have a web presence where it can display its wares and take customer orders. You have been charged with the job of providing a 'proof of concept' website that dynamically generates web pages in response to user requests. You are provided with a script congo.sql, this is to be used to create and populate a set of database tables that contain data relating to music recordings and the tracks therein. You are to create a dynamic website which uses these database tables as the basis for the content on certain dynamic pages of the site.

The tables have the following structure:

```
Music_Recordings (
  recording_id int(11),
  artist_name varchar(75),
  title varchar(75),
  category varchar(75),
  image_name varchar(75),
  num_tracks int(11),
  price float,
  stock_count int(11))
```

```
Music_Categories (
  id int(11) default NULL,
  name varchar(75) default
NULL
)

Music_Tracks (
  id int(11),
  recording_id int(11),
  title varchar(75),
  duration int(11)
)
```

The assignment is broken down into grade bands, complete each band before the credit for further grade bands is accessed. You are to complete the assignment using Java.

Refer to the formative assessment relating to week 4 as this will help you to complete tasks 1 and 2.

Preparatory Task

1) Download assignment.zip from Moodle and unzip it.
2) Edit and run the script congo.sql, such that it creates the necessary tables in your mudfoot/home Mysql area. (please refer to the appropriate guides on Moodle to install eclipse, java, and mysql on your machine)
3) Verify that the tables have been successfully created and populated. (Use MySql at terminal window)

Task 1 (40%-50%)

1) Create a dynamic web project in eclipse, it is to be called Congo_*your_student_id.*
2) Create a servlet, `MusicListing.java`, that generates a web page showing the details of all music recordings (albums), the results should be tabulated with column headings.
3) Create a html page, index.html, this should have a hyperlink or button that takes you to 2) this should be accessed using the relative URL, `albums`,
4) Create another html page, category.html, this page provides a drop down menu from which it is possible to display a table of album details based on selecting a particular musical category.
5) All pages displayed should contain a link back to the index page.

Task 2 (50%-60%)

1) Complete task 1
2) Create and use a cascading style sheet used throughout the site that displays a suitable logo on each page in the same position.
3) Create price.html, a page providing a drop down menu from which it is possible to display a table of albums based on selecting a particular price range.
4) Create artist.html that provides a text entry search for the albums of a particular artist, should the search not return any albums then a suitable error page is to be displayed.
5) All pages that display the results of various searches should now display a hyper link from the album title which goes to another page where the details of the album and a full track listing are displayed.
6) All pages should display a link back to the index page.

Task 3(60%-70%)

1) Extend the application, allowing albums to be added to a shopping cart from whichever page displays album details. Once the shopping is completed, the user will be able to amend their order by deleting items from the cart. The interim cart page should display; the title, artist and price of albums: and the running total of the order
2) The checkout page is to display the total price and details of the order and provide a form for delivery and contact details to be entered (payment details are not required). All details are to be completed before submission of the order. On submission the details of the order are to be entered into a suitable set of database tables. (A print screen of the table of orders contents will suffice here).

Task 4 (70%-80%)

Implement a logging in system with users distinguished as either admin or customer. Those not logged in can access the searches but cannot buy anything. The system will have a registration screen and the details stored in a suitable database table(s). Allocate customers a unique customer number. When logged in, as well as the functionality in the previous tasks, there is to be the ability to check their current orders. Admin users will be able to get details of all current orders. Admin has the functionality to amend the stock database by deleting/adding albums.

Other tasks (80%+)

Demonstrate use the MVC and DAO design patterns by implementing JavaBeans, jsp and the use of response.redirect. Implement any other features such as the use of AJAX to provide update/gossip information etc.

Add functionality to displaying pages that allows the listings to be ordered by price or title or artist, etc. Provide any other functionality as desired.

Warning: Using MVC/DAO etc. will require substantial rewrite of application if this has not been used from the start. Please ensure that you have a backup of the code if you are going to re-factor/re-write it.

Please Note: Use of jsp and patterns at lower grade bands will draw further marks and may allow for grade boundaries to be crossed.

What you need to hand in.

1. You are to submit a single zip file to Moodle. The filename is the form:
   surname _studentnumber.zip e.g. Marsden_ 99999999.zip

   The zip will contain:

   a. An exported zip file of your entire Eclipse project, *Congo_your_student_id.zip*.

A document, *surname _studentnumber .pdf* , with your details (ID, name, course and email) as the

header. The first line will indicate the grade you feel you have achieved. There should follow a short report (no more than 1250 words) justifying your grade, using code snippets only to indicate how your application works. The report will also contain screenshots of your working application as evidence, as well as MySQL output to indicate the nature of any new tables created. Evidence should be clear, shots indicating the shopping cart process for example. **Java files need to be fully commented and include your name and id number at the top of each and every file.**

Demonstrations:

A random sample of projects will be demonstrated to Nick, John and I prior to final release of the marks.

Paul Marsden

p.marsden@mmu.ac.uk