COA123 - Web Programming

Individual Coursework (80% of the module mark)

Dr. Hossein Nevisi (h.nevisi@lboro.ac.uk)

The following tasks are built around creating an online website to explore data related to sports, specifically track and road cycling at the London Olympics. You will be working with a given read-only database containing three tables (country, cyclist, and event), which will be used for the majority of the tasks.

There are four tasks in total; the first three being tightly specified and the fourth one allowing you room to design your own solution. Please read the individual task details and also the "Further Details" section that follows before starting your programming.

For Tasks 1, 2 and 3 you are provided with html pages (bmi.html, athletes.html, details.html) containing forms which will provide the input to your php pages. You are to ensure that your solutions work with these test pages. You should not modify these html files; all your coding should go into your php files.

Task 1 (8%)

Write a php script (**bmi.php**) to produce as output an html table of BMI (Body Mass Index) values, based on the input of 4 values as follows:

‘min_weight’, ‘max_weight’ (input in kg)
‘min_height’, ‘max_height’ (input in cm)

The finished table will have rows headed by ‘min_weight’ to ‘max_weight’ (incremented in steps of 5), and columns headed by ‘min_height’ to ‘max_height’ (incremented in steps of 5). You may assume the input values are multiples of 5 and you should produce output to 3 decimal places.

The formula you should use to calculate the BMI is:

$$BMI = \frac{weight\ (kg)}{height\ (m)^2}$$

For example, given min_height=140, max_height=160, min_weight=60, max_weight = 75 in the bmi.html, the answer from your bmi.php should be something like

Height → ↓ **Weight**	**140**	**145**	**150**	**155**	**160**
60	30.612	28.537	…		
65	33.163	30.916	…		
70	…	…	etc.		
75					

Task 2 (15%)

Write a php script (**athletes.php**) to accept as input an ISO country code 'country_id' and name 'part_name' and produce as output an html table listing the name and the number of participated events of those cyclists from 'country_id' whose name column (from the cyclist table) contains the string 'part_name' (for example the input could be "GBR" for 'country_id' and "vic" for 'part_name'). Your search can ignore the case of the input strings.

Task 3 (12%)

Write a php script (**details.php**) to echo a JSON data structure containing the name of each cyclist born between the given inputs 'date_1' and 'date_2' (inclusive), together with their country's name, sorted descending by their DOB. Note: this script should echo only the JSON data structure (in the correct syntax).

Task 4 (65%)

Design your own webpage (must start from **compare.php**), which allows a user to input two country iso_id values and outputs a comparison of the two countries. As a minimum, the page should output the details of how many medals (gold, silver, bronze, and total) each country won, together with a list of names of their cyclists. This would potentially obtain up to **30%**.

To have a more comprehensive comparison of those two countries, you are expected to demonstrate an interactive way to allow the user to choose a ranking criterion (from this list: gold medals, number of cyclists, and average age of cyclists when they took part in the London 2012) and see the ranking* of those two countries amongst all the countries. This would potentially obtain up to **20%**.

You can be creative about how you compare and show the comparison of the countries.

You should consider using "AJAX requests or Fetch API" as well as "JSON" to make your webpage more interactive and allow your webpage to be updated asynchronously (**15%** available for effective use of them).

* E.g., a country will rank 1 on gold medals, if it has the highest number of gold medals amongst all the countries. The country with the next highest number of gold medals will ranks 2, and so on. If, for example, two countries have the same number of gold medals, they should be ranked the same, and the next countries should be shifted accordingly. This ranking system works the same for number of cyclists. For average age of cyclists, the country with the lowest average age will rank 1.

Further Details

- **Database**

You are provided with a (read only) MySQL database on sci-mysql server. You can inspect the database content using phpMyAdmin https://sci-mysql.lboro.ac.uk/ :

- Server name: sci-mysql
- DB name: coa123cdb
- Username: coa123cycle
- Password: bgt87awx!@2FD

It contains three tables as follows:

- country
  iso_id (ISO country code)
  gdp (recent Gross Domestic Product of the country)

population (recent population of the country)
country_name
gold (number of golds received in London 2012 Olympic games)
silver
bronze
total (total number of medals)

- cyclist

  cyclist_id
  iso_id
  name
  height (height in cm)
  weight (weight in kg)
  gender (M or F)
  dob (stored as a MySQL Date in format yyyy--mm--dd)
  sport (type of cycling events the athlete does)

- event (the events the athlete participated)

  record_id
  cyclist_id
  event_name

Note: some data may be missing or dummy. The data originally came from the Datablog of the Guardian Newspaper, which we kindly acknowledge. You should only use this data (from the above MySQL database) to complete the tasks.

➢ Coding

You are permitted to use jQuery and Chart.js JavaScript libraries as well as the Bootstrap framework, and the code that was part of the COA123 lectures and labs. All other coding must be your own. Please note that you must NOT use any absolute directory in the code.

➢ Submission

All coursework files (including the given html files for Tasks 1-3) **must be uploaded to sci-project.lboro.ac.uk** within the coursework folder "yourstudentidolympics" (**all lowercase letters**). For example, if your Student ID is F031234, then the name of your coursework folder on sci-project must be "f031234olympics". All must be in lowercase including the letter in your student id. This folder must be directly under in your "web" directory. You will receive email alert if you do not have this CW folder when the deadline of CW is close.

For the CW submission, all php files of your script and start html files for each task must be at the root level of your coursework folder. If you are going to use other types of files (such as js, css, and images), you need to insert them at the root level as well; therefore, you must not create any sub-folder in your coursework folder.

No CW folder or incorrect file naming = Zero marks

Immediately after the deadline, a script will be run which will lock you out of your coursework directory.

If you have any problem with your account on the sci-project server, please contact Mr. James Skevington it.services@lboro.ac.uk .

➢ Marking criteria and feedback

For all tasks you can obtain marks for:

- obtaining the correct results for queries
- attention to detail (e.g., validation of user input)
- design of interface (e.g., clarity, informative, easy to use, especially for Task 4)

For Task 4, you are encouraged to be creative to achieve good usability, functionality and layout. You will also obtain marks by paying attention to details in designing your own user interface; for example, users' input validation, user-friendly interface, responsive design, informative result/display, and page layout (e.g., clarity, consistency and easy-to-use). You might consider using copyright free graphic images. Your webpage does not need any design consistency with the html pages provided for previous tasks, and you don't have to list the result in an HTML table in Task 4.

For more details, look into the marking scheme document.

Coursework will be marked using Chrome on Windows.

➢ Plagiarism

The coursework is individual and therefore should of course be your own work. Failure to do so would leave you open to prosecution for Academic Misconduct.

If you have any questions, please ask during your lab practical sessions, or email Dr Hossein Nevisi (h.nevisi@lboro.ac.uk).

-- END --