CS3101 Databases Practical 2

Due: 21:00, Monday 10 April 2017 (Monday week 10) **Weighting:** 40%

Aim

The aim of the practical is to promote awareness of database connection.

Requirements

In this practical you are to produce a web-based user interface to your database for the online delivery platform you created in Practical 1. You could for example use HTML, CSS and PHP or Node.js for this assignment. There are of course many other options and if you want to use a different programming language and/or framework, please ask us first. Please note, the user should not have to enter any SQL, you are creating a user interface for the online delivery platform.

Your interface should be accessible via a web browser from your School home directory from e.g.

http://rr71.host.cs.st-andrews.ac.uk/cs3101/

where *rr71* should be replaced with your own username. More details are given in the next page.

Submission

All work must be submitted electronically as a *zip* file via MMS. In case of problems with the MariaDB or PHP installation, please email fixit@cs.st-andrews.ac.uk indicating you have registered for the Databases module.

Kasim Terzic, David Harris-Birtill
6 March 2017

Details

This practical involves producing some code to provide a web-based user interface for the online delivery platform application for which you designed and implemented a database in Practical 1. You do not have to provide a complete user interface, but can concentrate on some core functions as outlined below.

If you decide to use PHP, be aware that instructions for accessing your database using PHP will be provided in lectures and in a separate hand out. You should examine the examples shown in class (lecture slides and hand out) and use the on-line documentation for MySQL functions provided at http://www.php.net/manual/en/ref.mysql.php as well as information on general PHP functions at http://www.php.net/manual/en/.

Your final interface should be accessible by a browser from e.g.http://rr71.host.cs.st-andrews.ac.uk/cs3101/

where *rr71* should be replaced with your own username. If you decide to use node.js, please submit your source tree and provide a README file which describes how to run your server and which packages need to be installed.

If you decide to use something else, check with us first, and ensure that any required technologies are available in the labs and that your solution can be run on the lab machines without trouble.

Basic Requirements

Design and implement a web-based user interface to the online food delivery platform introduced in Practical 1. The interface must connect to your MariaDB database from the last practical and should provide users with at least the following functionally:

- Permit the user to view all restaurants in the region
- Permit the user to view the menu for each restaurant
- Permit the user to search for specific dishes across all restaurants
- Permit the user to place an order containing one menu item only
- Permit the user to search for popular restaurants and dishes

Remember, while the main task is to provide a user-friendly user interface to your database and system, good code design is still important. Regardless of your chosen programming language, try to create a sensible set of functions/methods and possibly classes to provide an API for accessing your database and providing the required functionality.

Extension Elements

Please first build a basic working solution, fulfilling the Basic Requirements listed above, before trying anything listed below. To gain a mark above 16, you should complete one or more of the items below:

- Permit the user to register and/or login to the system (if you are using PHP, *session* variables may help you record when someone has logged in)
- Permit the user to view a list of messages ordered by distance from the user's current location
- Permit the user to view the locations of restaurants on a map (you may find Google Maps or OpenStreetMap can help you with this)
- Provide an administrator login and ability for the administrator to add and remove restaurants
- Estimate delivery time, either by checking past delivery times for relevant couriers, or by estimating it based on the distance

Report

You should write a report including the information listed below. In your report, where appropriate, try to concentrate on *why* you did something rather than merely explaining what you did:

- Instructions for accessing your web-based interface which should be deployed on the School servers
- A short overview summarising the functionality of your application indicating the level of completeness with respect to the requirements listed above
- An accurate summary stating which files or code fragments you have written and any code you have modified from that which was given out in class or which you have sourced from elsewhere
- A short justification of any modifications you made to your original database implementation
- A discussion of your interface design, implementation and testing and any interesting features including the addition of any features beyond the original specification
- Documentation of operations you can perform using your system and their results, including documentation of insertions, deletions, updates and queries which took place as a result of operations, showing database content before and after operations
- Evidence of testing (e.g. using screenshots in an appendix) for the five basic requirements

To be handed in

Upload to MMS a single *zip* file containing

1. All the source files for your interface
2. Your word-processed report as **PDF**

Marking

The submission will be marked on the University's 20 point scale according to the mark descriptors in the Student Handbook at:

https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptors

The following aspects will also be considered:

- Quality of the report – documentation of design, implementation and testing, explaining and justifying design and implementation decisions
- Quality of the design and implementation – designing your code to contain general purpose (and reusable) functions/methods/classes is likely to score higher
- Completeness of the implementation with respect to the requirements

I would remind you to ensure you are following the relevant guidelines on good academic practice as outlined at

https://www.st-andrews.ac.uk/students/rules/academicpractice/

Lateness Penalties

The standard lateness penalties apply to coursework submitted late. As indicated at
https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#lateness-penalties