ITECH3224—WWW Technology 2: Practical Assignment

University of Ballarat, Graduate School of ITMS, Semester 2, 2009

Details

Due date
Refer to course description.

Weighting
30% of final grade.

Preface
This is an individual assignment, **not** a group assignment. This version is for students who are enrolled in an undergraduate degree i.e. Bachelors degree.

Description of task
You are required to develop a database driven web application which will enable users to browse information on world countries, cities and languages. The application will allow an administrator to maintain the database contents. The application will also allow registered users to monitor a preferred list of countries.

Your site should be written in PHP (with additional, HTML and JavaScript files) and store and update all persistent data in a MySQL Server database. Where possible, form validation should be done on the client side. This means that JavaScript code should be used to validate user input. Other validation e.g. comparison to stored data will need to be done on the server side.

It is suggested that you download a Web Design Template from http://www.oswd.org/.

Application Requirements

1. Web site to view and update Countries, Cities and Languages of the world.
2. world.sql contains MySQL create database and tables (Country, City, CountryLanguage) including SQL that inserts some data.
3. You will be required to create a user table with userid, name, and password (must use the MySQL password() function), and role (e.g. system admin, registered user). This table will be used to authenticate (check) when a user logs into your web page.
4. You will also be required to create a registered user table with userid and the values of the countries that the user wants to monitor.
5. The main page should show an image, of your choice, to reflect the 'global' nature of the site, e.g. map of world, or country flags, etc.
6. The major component of the main page should be the search functions that enable a user to retrieve a list of countries, cities, and / or languages.
7. The list produced should be a list of links to another pop-up window with the detailed information for the selected link. If a country list is produced for example, selecting a country will display a pop-up window with all the information stored in the Country table.
8. The main page should also have a login component for system administrator and registered users. You do not have to login to the site to use the search functions.
9. If a user logs in and they are known to the system as a system administrator they should have access to screens that enable to add, update or delete the respective Country, City, and Language information. No other user type should see or have access to these functions.

10. If a user logs in and they are known to the system as a registered user then they should be given an extra search option to restrict their lists to include only the countries in their registered list. They should also be given an option to be able to add or delete countries from their registered list.
11. A user should be given the option to be able to become a registered user and be able to enter the countries that form their registered list.
12. At least one list (Country, City, or Language) must be constructed using a dynamic XML document and XML style sheets. The user should be given the option to save the list produced as an XSLT file. This file (not the list on the web page) should contain all the attributes that the user could retrieve by selecting a link on the list. In other words the Country XML file for example, should have all the attributes defined in the Country table.
13. At least one list must be constructed using HTML and either inline or external style sheets (CSS).
14. Search options available on the main screen:
    a. Full or Partial city name **->** City list (<u>City name</u>, Country) **->** City screen (all City attributes). I.e. an entry field should exist with an appropriate label to be able to enter part or all of a city name e.g. 'New York' or just 'york'. Then pressing a search button the application replaces the current window contents with a City list with the fields City name, and Country. Clicking on the city field will bring up a pop-up window containing the City screen with all of the City attributes.
    b. Cities within a country (list box of country names) **->** City list (<u>City name</u>, Country) **->** City screen (all City attributes)
    c. Cities with population < 1,000,000 or >= 1,000,000 (radio button to choose) **->** City list (<u>City name</u>, Country) **->** City screen (all City attributes)
    d. Full or Partial country name **->** Country list (<u>Country</u>, Continent) **->** Country screen (all Country attributes)
    e. Countries within a Continent (list box to choose) **->** Country list (<u>Country</u>, Continent) **->** Country screen (all Country attributes)
    f. Countries with population < 1 mill / 1 mill – 4.9… mill / 5 mill – 9.9… mill / > 10 mill (list box to choose) **->** Country list (<u>Country</u>, Continent) **->** Country screen (all Country attributes)
    g. Official languages **->** Language list (<u>Language</u>, Country) **->** Language screen (all CountryLanguage attributes)
    h. All languages **->** Language list (<u>Language</u>, Country) **->** Language screen (all CountryLanguage attributes)

What to submit

1. A cover page (A4 format) - displaying course code, course name, assignment title and student id with student's signature.
2. A CD-ROM Disk containing your application (e.g. all PHP, JavaScript, XML, Style Sheet, DB SQL file, and Web Design Template files).
3. The CD-ROM Disk should be appropriately labeled containing student id course code.
4. The CD-ROM Disk should be appropriately stored in a plastic or paper CD cover
5. Application files should be stored according to their usage and scope. The application should use subdirectories to differentiate server side files, database files (DDL queries), documentation including installation instructions, and an acknowledgement file for source of external templates and classes etc.

Assignment Marking Sheet

General Requirements	Mark
Cover Page provided with student id and signature	2
CD–ROM Disk contains label with course code and student id	2
Subdirectories to differentiate server side files, database files (DDL queries), documentation including installation instructions, acknowledgement file for source of external templates and classes	6

Database Administration	Mark
Database tables (country, city, language (1 marks), and user (1 marks), preferred 2 marks)) exist and are appropriate for the application (allows for more than one preferred country for registered users)	4
Appropriate privileges granted to username (worldapp), password (worldpass)	2
Test data (particularly registered users and preferred) exists	2
Passwords not stored as clear text	2

General Programming	Mark
DB access contains defined database hostname (localhost), username (worldapp), password (worldpass) and database name (world)	1
Appropriate / consistent coding style with useful comments	2
Appropriate usage of add_slashes and strip_slashes	1
Appropriate validation of input fields (e.g. mandatory fields, valid values).	2
JavaScript validation of input fields	1
PHP code runs without run time errors	3

Functional Requirements	Mark
'Global' image	1
Login	2
Registration	2
Manage country (Add/Edit/Delete)	2
Manage city (Add/Edit/Delete)	2
Manage language (Add/Edit/Delete)	2
Manage preferred list (Add/Delete)	2
Search – 1 mark for each search filter that works correctly	8
Filter searches for registered users	2
List – Each list item links to appropriate pop up window	4
Pop up windows contain complete data (country, city, language)	3