CS2410 Internet Applications and Techniques Coursework

Due date: Wednesday 30th March 2016 midnight

Introduction

This coursework asks you to demonstrate your ability to design and develop a three-tier dynamic database-driven website. It contributes 25% toward the CS2410 mark.

Scenario

The company "Aston animal sanctuary" has approached you to design and implement their new website. They run a sanctuary where animals are looked after before being adopted by members of the public. They are interested in a website that allows them to record and manage adoptions, and where members of the public are able to view animals that need homes. There should be two types of user in your system: staff and normal user.

A staff user should be able to:

- Log into the system and be presented with a home page showing
  a) The animals available for adoption
  b) A list of pending adoption requests
- Add a new animal to the system, these animals are added one at a time and are listed in the database as owned by the staff account and available for adoption
- See a list of all animals in the system and who owns them
- View all adoption requests made by ALL users and whether they were approved or denied
- Approve or deny adoption requests made

A normal user should be able to:

- Register a new account
- Login to their account with the home page showing
  - o A list of their animals
  - o A list of pending adoption requests
- List all the animals available for adoption
- Search for specific animals, by name and age (or as a stretcher, type and breed)
- Make an adoption request
- View all their adoption requests made and whether they were approved or denied

Information stored about the animals in the system should include their name, date of birth, a description about the animal and a photo. Animals are initially owned by the staff account, and so are considered available for adoption (use the available column in the animal table of the database). When an animal is requested for adoption this is changed to not available. The adoption is then either approved by the staff account and so the animal changes owner and continues to be unavailable, or the adoption is rejected and so the status is set once again to available.

In general, some basic animal information can be displayed initially; this then should have a link to a page where you can read all information about the animal. A suggested version would have the animal's picture, age, name listed on a page. This then links to a webpage about the animal where the date of birth and description are also displayed.

When adding an animal a picture should be provided. Pictures should be uploaded to the webserver, with only the link stored in the database. You may want to restrict the file upload size as well as the file type to save space on your H: drives. Both types of user should be able to search for animals by ID, name or age.

All users should be able to log in and out of the system and it should be both secure and follow basic user interface principles.

Your implementation must be in PHP, HTML and MySQL. You are free to choose whether to use AJAX, object-oriented programming PHP, CSS, framework and other relevant technologies or not . You are suggested to use Wampserver or Xampp which includes Apache, PHP, MySQL, etc in one package.

Security

Security is important in all web-based software, so some basic validation is required in forms as well as the following:

- Restrict file upload to only images
- Form validation to ensure only correct entries can be added (e.g. not random strings of characters for dates etc.)
- Ensure that HTML and SQL injection is handled
- Do not allow a user to log in with a normal user account and access staff only pages

Stretchers

This is a list of things that will help you get the maximum number of marks.

- Allow the staff account the option to edit an animal's details
- Store the animal type in the database and allow users to search for animals by type
- Search for animals based on a set of criteria (e.g. "I only want to see animals under six months old" etc.)
- Allow for tables to be sorted based on their heading. (Sort by name, age or type etc.)
- If there is no picture for an animal, what will you display?

Database information

Your database could use the following four tables to represent the information held in the system.

- `User (userID, staff, password)`
- `Animal (animalID, name, dateofbirth, description, photo, available)`
- `Owns (userID, animalID)`
- `AdoptionRequest (adoptionID, userID, animalID, approved)`

The `staff`, `available` and `approved` fields could use Boolean values or other suitable type. The `approved` field could take into account three options: Not yet processed (null), approved (true) or not approved (false).

Submission:

You need to submit your work to the Blackboard as well as to the host server.

For Submission to Blackboard, the following documents are required:

- A copy of your webpage, including all php, html, css, image files and anything else the webpage requires to function
- An exported version of your database from phpmyadmin (SQL file)
- A short report (no more than 3 A4-size pages) about your system covering the following:
  - A short summary of your system.
  - A link to the entry page of your website, normally index.html or index.php
  - Any user names and passwords used to access your system.
    Please note, you should have at least the staff account and two normal users. There should be two animals available for adoption and each of the normal users should have at least one animal of their own.
  - Identify where each part of your system implements specific features, including where sessions were used, where you used get or post information for forms, and your security practices.
  - Any assumptions made by the system should also be stated.

You should put all the above documents in one folder and name this folder using your university username. You need to compress this folder to a <yourusername>.zip file and submit this zip file to the blackboard.

Submission to the university host server should be done before the deadline as well. You should have received an email with the server (<user>.eas-CS2410-1516.aston.ac.uk) information and your login detail. Please reference the host server documents on blackboard on how to use the server.

Your website must be hosted on our remote server (<user>.eas-cs2410-1516.aston.ac.uk). You should test your host server well ahead of the submission deadline.

Notes:

- Without source code submitted through Blackboard, your coursework will not be marked.
- If you do not put your website/database on the host server, your mark will be deducted by **20%.**
- Late submission will incur a penalty of 10% of the available marks for each day.
- Your submission must be your own effort. Copying and sharing work is forbidden. If you are found to have copied or shared part or all of your submission your mark may be reduced or set to zero (or worse).
- The coursework will be checked using the Google **Chrome** browser. You need to make sure all your system work on it.

Assessment Criteria:

Marks for this work will be awarded as follows:

- **Basic Operations: (total of 70%)**
  - **Staff Account functionality: 30%**
    - Staff able to log into account

        - Staff can add an animal
        - Staff can list all animals in their system and all adoption requests, whether pending or not
        - Staff can approve or deny pending adoption requests
    - **Normal User Account functionality: 30%**
        - User can create an account
        - User can log into their account
        - User can see all animals available for adoption
        - User can make an adoption request
        - User can list all their past and current adoption requests
    - **General: 10%**
        - Animal details are able to be viewed when clicked on
        - It is possible to search for specific animals by id, name or age
        - It is possible to log out of the system
        - Good coding practices adhered to, code commented and code layout all considered
- **Security: 10%**
    - Restrict File uploads
    - Restrict webpages to correct account type
    - Sanitise and validate inputs to prevent HTML and SQL injection
- **Stretchers: 10%**
    - Edit animals
    - Store animal type in the database and allow users to search by type of animal
    - Table sorting based on headers
    - Default pictures
- **Report: 10%**