Autumn 2023: Web Application Assignment

Due by 7:00pm on Friday 2nd June 2023 **Assessment Weight: 30%**

A. Requirements

a) ALL instructions given in this document MUST be followed to be **eligible** for full marks for the Web Application Assignment. This document has eight (8) pages in total including four (4) Appendices.
b) This assignment is **NOT** a group assignment; collusion, plagiarism, cheating of any kind is not acceptable. As part of your submission, you MUST certify that all work submitted is your own. If you cannot honestly certify that the work is your own, then do not submit the assignment. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).
c) All assignment submissions will be checked for academic misconduct using the MOSS program from Stanford University.
d) Design the web pages with ease of navigation and operation, attractiveness, and accessibility in mind. Images other than those provided in the assignment zip file, if any, may also be used in the assignment.
e) Your code must guard against SQL injection and Cross Site Scripting attacks. That is, sanitise user input.
f) All assignment files are to be uploaded in the **project** folder in your TWA web site on the TWA server as follows:
   - php and html files in the `project` folder
   - css files in the `project/css` folder
   - images in the `project/images` folder
   - javascript in the `project/javascript` folder.

   Note: Compressed archive files (eg, zip, tar etc) are not acceptable and will not count toward submission requirements

g) Complete the full submission process before the due date and time. See section D for details of the submission process.
h) All styling and page layout must be achieved using CSS. The use of Bootstrap or other frameworks is not permitted.
i) jQuery or similar are not permitted.

For the problem definition described in section B you must
j) include your authorship details at the top of **each file** in coded comments.
k) **reference** all sources that you used for inspiration of your solution as per Section C of this document.
l) ensure that your web application renders correctly in Chrome and runs correctly from the TWA web server.

B. Web Application Assignment Details

B(i) - Background information and description

The aim of this project is to develop a Pet Rescue website for an organisation called `Annie's Animal Adoptions` that finds homes for unwanted pets such as dogs, cats, birds, horses, and rabbits. The details for each pet that is available for adoption are available in a MySQL database named `petrescue`. The Pet Rescue website connects to this database to dynamically create web pages that allow users (potential pet adopters) to view the pets available for adoption and submit adoption applications. Additionally, the website will allow the administrators of `Annie's Animal Adoptions` to manage pet listings, applications, and adoptions. The website will use PHP for server-side scripting and MySQLi for database interaction.

B(ii) – Functional Requirements

Your Web Application **must**

- be coded using HTML 5, CSS, JavaScript, and PHP as necessary. Note: any files described below that require any type of server-side functionality must be PHP files to achieve the functionality,
- Use MySQLi to utilise the `petrescue` database,
- provide easy-to-use navigation for all user types as described in the following page descriptions,
- be designed for a 1366-pixel wide screen and be accessible,
- provide the following page content and functionality as described. Note: php pages other than those listed below may be needed for a full solution.

Index Page
The index page should include appropriate headings, logo, images, and description/overview[1] of `Annie's Animal Adoptions`, along with links to the Pet Listings, Register, and Login pages.

Access Allowed:
- All user types.

Pet Listings Page
The pet listings page should display summary information about the pets available for adoption. For each available pet display their photo, name, suburb, and state. The page should also include a search function that allows users to filter pets by species and gender. The user can click on any pet in this listing page to obtain more details about the pet. Doing so will open the Pet Details page for the selected pet.

Access Allowed:
- All user types.

Pet Details Page
The Pet Details page provides detailed information about **a selected** pet, including its photo, age, gender, breed, brief description, location[2], and adoption fee. Additionally, the page will feature a form that allows **registered users** to submit an adoption application for the pet.

Adoption applications can only be submitted by registered users who are currently logged in to the site. If a user is not already registered, they must first register for an account and then log in before they can submit an adoption application. The adoption application form requires users to submit their full name[3], contact details[4], and a statement indicating why they should be considered as an appropriate owner for the pet. By completing this form in full and accurately, potential adopters can increase their chances of being considered for adoption of their desired pet.

Access Allowed:
- All user types.
- Only registered users can submit an adoption application.

Validation:
- **Client-side JavaScript** validation for mandatory fields[5]. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and cannot submit. If no errors are detected, the form is submitted to the server where server-side validation is undertaken before the data is updated in the `petrescue` database.
- **Server-side PHP** validation to check data types and sizes of input as necessary, and to guard against SQL injection and XSS attack. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and the database is not updated. If no errors are detected the data is updated in the database.

Manage Pet Listings
Administrators of Annie's Animal Adoptions should be able to log in to the website and manage pet listings, including adding new pets, updating pet information, and removing adopted pets from the listing.

To add a new pet to the `petrescue` database the administrator will need to enter the following data items about the animal into an appropriate form: name, species, breed, age, gender[6], description, image filename[7], suburb, state[8], adoption fee.

Th functionality required in this implementation of the site is to **insert new pets only**. Updating and removing pets is not required in this version of the site.

---

[1] A text file, `index_Text.txt` has been provided in the project zip file that includes suitable text that can be used in the index page to describe the organisation.
[2] Suburb and state
[3] first name and surname: These details should be automatically completed from the logged in user information available in the database.
[4] mobile phone number and email address: These details should be automatically completed from the logged in user information available in the database.
[5] all fields on the adoption application are mandatory.
[6] Check the data dictionary for the allowed values for gender.
[7] The administrator enters the filename for an image that is stored in the images folder.
[8] Check the data dictionary for the allowed values for state.

Access Allowed:

- Administrators only.

Validation:

- **Server-side PHP** validation for mandatory fields[9], check data types and sizes of input as necessary, and to guard against SQL injection and XSS attack. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and the database is not updated. If no errors are detected the data is updated in the database.

Manage Adoption Applications

Administrators of Annie's Animal Adoptions should be able to log in to the website and manage adoption applications, including viewing, and approving or rejecting applications. This page will need an appropriate interface that allows the administrator to choose an application to view from all the pending applications and then decide whether the application is approved, rejected, or should remain as pending. The decision made by the administrator regarding the application needs to be updated in the `petrescue` database accordingly[10].

Access Allowed:

- Administrators only.

Validation:

- **Server-side PHP** validation to check data types and sizes of input as necessary, and to guard against SQL injection and XSS attack. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and the database is not updated. If no errors are detected the data is updated in the database.

Registration Page

To apply for animal adoption, users are required to register with the site. Registration involves completing a form on the registration page, where users must provide the following information: first name, last name, email address, mobile number, and password. These details will be securely stored in the `petrescue` database. It is essential that all fields in the registration form are completed by the user, and the provided email address must be unique to ensure a successful registration process. Since passwords should never be stored in plain text, the provided password must be hashed using the sha256 algorithm before being stored in the database.

Access Allowed:

- All user types.

Validation:

- **Client-side JavaScript** validation for mandatory fields[11]. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and cannot submit. If no errors are detected, the form is submitted to the server where server-side validation is undertaken before the data is updated in the `petrescue` database.
- **Server-side PHP** validation to check data types and sizes of input as necessary, and to guard against SQL injection and XSS attack. If any errors are detected, the form will display appropriate error messages using in-page DOM notifications and the database is not updated. If no errors are detected the data is updated in the database.

User Authentication and Authorisation

There are three types of users for the website: potential pet adopters, registered adopters, and administrators.

Potential pet adopters do not need to register or login to use the site. However, they will not be able to access any pages that are designated for administrators only nor will they be able to apply to adopt a pet. If a user tries to access an administrator-only page without logging in as an administrator, they will be automatically redirected to the site's index page.

As described in the preceding pages, certain parts of the site are to be accessed only by either logged-in registered adopters or logged-in administrators. The same login form will be used by both registered adopters and administrators.

To log in, registered adopters or site administrators must provide their user credentials in a suitable form, which will be authenticated against the user details stored in the `petrescue` database. The login process must use a postback form. Upon

[9] all fields, except breed, on the add pet form are mandatory.
[10] The outcome of the adoption application may need to be updated in both the **adoptions** and the **pets** table depending upon the actual outcome. Don't try to do this in one query.
[11] all fields, except mobile number, on the registration form are mandatory.

successful login a registered adopter is to be redirected to the **pet listings** page. Upon successful authentication a site administrator is to be redirected to the **manage pet listings** page.

Logged-in users, of any type, must be able to log off the system when they are finished using it from any page of the site. Successful logoff should redirect to the index page.

B(iii) – Database Description

1. Tables in the ***petrescue*** database are described in the **Data Dictionary in appendix 1**.

2. You have been provided with your own copy of the ***petrescue*** database on the TWA server. To access this database, you need to use a username and password. Details on how to connect to your copy of the database are in **appendix 2**.

3. The tables within the ***petrescue*** database have already been populated with some data. Use the `allTables.php` script to view the data (make sure you use the connection information as indicated above). This script is found in the project zip file.

4. A list of usernames and [plain text, decrypted] passwords can be found in **User Credentials in appendix 3**.

B(iv) – HTML, CSS, JavaScript, PHP, and image files

CSS: All page styling is to be achieved using CSS. Create a CSS file called `Annies_Master.css`. Add your CSS rules to this file. You may create additional CSS files if you wish. All CSS files should be uploaded to the `project/css` folder of your TWA website.

JavaScript: Some functionality in the web site will need to be achieved using JavaScript (eg, the client-side validation). Create a JS file called `Annies_Script.js`. Add your JavaScript to this file. You may create additional JavaScript files if you wish. The JavaScript files should be uploaded to the `project/javascript` folder of your TWA website.

PHP and HTML: Most pages listed in section B(ii) will need to be PHP files to achieve the required server-side functionality. Additional PHP and HTML files may also be created as part of your solution if desired and **if doing so does not contradict/negate the stated page requirements** (for example, the login process must be achieved using a single php file that utilises postback). PHP and HTML files should be uploaded to the `project` folder of your TWA website.

Images: A set of image files have been provided in the project zip file. These image files should be uploaded to the `project/images` folder of your TWA website. The dimension of each image is 500 x 500 pixels. These files have specific names that have been used in the petrescue database for specific pets. They are to be used when displaying the profile information for the pet. You may add other images to the `project/images` folder and the database if you wish.

C. Referencing

Referencing must follow the guidelines given in Section 2.5.3 of the TWA Learning Guide.

D. Submission Instructions

To submit your Assignment, you must do the following by the due date and time specified on page 1 of this document.

1. Upload all assignment files in the **project** folder in your TWA web site on the TWA server as follows:
    a. php and html files in the `project` folder
    b. css files in the `project/css` folder
    c. images in the `project/images` folder
    d. javascript in the `project/javascript` folder.
2. Run the submission script located at

    `http://twaaut.cdms.westernsydney.edu.au/submit/submit.asp`

    As part of the submission, you will be prompted for your TWA website username and password. You will then be asked to read the WSU policy on misconduct and certify that work submitted by you is your own work. This action will be logged in a database for future reference and is deemed to be evidence that you claim that your work is original. Next, you will need to select from a drop-down list the Assessment you are submitting, eg, Assignment 1, and click the *Submit Assessment*

button. The web page will then display a listing of the files you have submitted along with a receipt number. You should print or screenshot this page for proof of submission.

E. Marking Criteria and Standards

The marking criteria and standards for the Web Application Assignment are published in Section 2.5.3 of the Learning guide and will be used to assess your assignment submission according to the specific weightings identified in the table below:

CRITERIA	WEIGHT
CODE FUNCTIONALITY/CORRECTNESS	60%
WEB PAGE DESIGN	25%
FORM DESIGN	10%
CODE READABILITY	5%

Appendices

Appendix 1 – petrescue Data Dictionary

The `petrescue` database consists of 3 tables. Each table is described below. Primary and foreign keys are also indicated. A description of how to connect to your copy of the database is given in Appendix 2. Some notes on inserting and updating the tables are provided in Appendix 4.

Table Name: users

This table provides details about users that can access the system including their login credentials (email and password) and their personal details.

Note: passwords are encrypted using the sha256 algorithm. For testing purposes, a copy of the plain text passwords is given in Appendix 3.

Your database credentials have **Select** and **Insert** privileges for this table.

Column	Type	Null	Default	Comments
user_id	int	No		This is an auto incrementing number to uniquely identify a table row. **You do not insert this number into the database it is determined automatically.** Unique identifier for a user. *Primary Key*
first_name	varchar(50)	No		user's first name
last_name	varchar(50)	No		user's last name
email	varchar(100)	No		Email address. This is used as the user's username when logging in. It must be unique for all users.
mobile	varchar(10)	Yes	NULL	Mobile number for user
password	varchar(100)	No		user's password. This is used for authentication. The value stored in this field is encrypted using the sha256 algorithm.
is_admin	tinyint(1)	No	0	Flag to indicate if the user is an administrator. 1 for Yes, 0 for No.
date_registered	timestamp	Yes	Current date time	Date that the user registered on the web site. Auto generated by default. Will only have a value for non-administrators.

Table Name: adoptions

This table stores data about adoption applications submitted by registered adopters to adopt a pet.

Your database credentials have **Select, Insert,** and **Update** privileges for this table.

Column	Type	Null	Default	Comments
application_id	int	No		This is an auto incrementing number to uniquely identify an adoption application. **You do not insert this number into the database it is determined automatically.** *Primary Key*
pet_id	int	No		Unique identifier of the pet on the adoption application. *Foreign Key*
user_id	int	No		Unique identifier of user that submitted the adoption application for the pet. *Foreign Key*
application_date	timestamp	Yes	Current date time	The date the adoption application was submitted. Auto generated by default.
application_status	enum('Pending', 'Approved', 'Rejected')	No	Pending	Status of the adoption application.
application_notes	text	Yes	NULL	Application information provided by the applicant describing why they would be a suitable new owner. The administrator uses this information to help them decide if the applicant is an appropriate new owner for this animal.
adoption_date	timestamp	Yes	NULL	The date the adoption was approved. Will only have a value when `application_status` is 'Approved'

Table Name: pets

This table gives details of pets for adoption.

Your database credentials have **Select, Insert** and **Update** privileges for this table.

Column	Type	Null	Default	Comments
pet_id	int	No		This is an auto incrementing number to uniquely identify a pet. **You do not insert this number into the database it is determined automatically.** Unique identifier for a pet. *Primary Key*
name	varchar(50)	No		Name of animal
species	varchar(50)	No		Species of animal
breed	varchar(50)	Yes	*NULL*	Breed of animal (if known)
age	int	No		Age of animal
gender	enum('Male', 'Female', 'Unknown')	No		Gender of animal
description	text	No		Brief description of the animal
image_path	varchar(100)	No		Path information to display image of animal
status	enum('Available', 'Adopted', 'Adoption Pending')	No	Available	Status of the animal.
suburb	varchar(50)	No		The suburb in which the pet is currently located

state	enum('NSW', 'VIC', 'ACT', 'QLD')	No		The state in which the pet is currently located.
fee	decimal(10,0)	No		The fee to adopt the pet.
date_added	timestamp	Yes	Current date time	The date upon which the animal was added to the database. Auto generated by default.

Appendix 2 – Connecting to your Pet Rescue Database

You have your own copy of the ***petrescue*** database. To access this database, you have to use a MySQL username and password. The following generic connection information can be used to connect to your ***petrescue*** database from your php scripts:

Database name: ***petrescue###***
Username: twa###
Password: twa###**XX**
Server: localhost

where **###** is your twa site **number**, and **XX** refers to the first two characters of your twa site password.

For example, if your TWA site is twa**999**, and your password is **ab**cd7890, then the following would be your connection information:
Database name: ***petrescue999***
Username: twa**999**
Password: twa**999ab**
Server: localhost

Hence, to connect to the ***petrescue999*** database from your php script you would require code similar to the following:

```
$dbConn = new mysqli('localhost', 'twa999', 'twa999ab', 'petrescue999');
if ($dbConn->connect_error) {
    die('Connection error (' . $dbConn->connect_errno . ')'
    . $dbConn->connect_error);
}
```

Note: The tables within the database have already been populated with sample data. Use the `allTables.php` script to view the data (make sure you use the connection information as indicated above in the script).

Appendix 3 – User Credentials for Pet Rescue web application

User Credentials for Annie's Animal Adoption system.

The passwords stored in the **password** field of the **users** table are encrypted using the sha256 algorithm. Below are the plain text passwords for these users.

email	Plain text **password**
Annie@AAA.org	adopt
Pip@AAA.org	catsRBest
steve@gmail.com	mypetRescuepassword
rose@gmail.com	roseLovesDogs
bob@yahoo.com	secure
kate@gmail.com	xbox

Appendix 4 – Notes about inserting and updating records in your Pet Rescue database

- When inserting into a table do not supply the primary key value since the RDBMS will automatically generate this when the insert takes place.

- When inserting into a table, if a field has an automatic default value do not supply your own value for this field unless you want it to be different to the default value.

  For example, when inserting into the `users` table do not supply the `is_admin` field value since it will default to 0 which is the required value for a registered adopter. Only administrators have the value of 1, which is never assigned via the registration form.

- When updating the `adoptions` table, the only fields that will need updating are the `application_status` field and possibly the `adoption_date` field[12].

- When updating the `pets` table, the only field that will need updating is the `status` field.

[12] See the comments column for this field in the data dictionary in Appendix 1 to identify when this field should be updated.