CSC680 Database Web Interface

**Final Exam**

|  |  |
| --- | --- |
| **First Name** | Jungbae |
| **Last Name** | Eriel |
| **ID#** | 041396462 |
| **Email Address** | Jungbae.t.eriel@gmail.com |

Table of Contents

[How to submit 1](#_Toc85882688)

[P1 – Pet Adoption System (40 Points) 2](#_Toc85882689)

[Part 1 - Database (30 Points) 2](#_Toc85882690)

[Part 2 - PHP/HTML Programming (10 Points) 4](#_Toc85882691)

[P2 Cities (35 Points) 4](#_Toc85882692)

[Part 1 – Arrays - City Distances (12 points) 4](#_Toc85882693)

[Part 2 - Central Point (12 Points) 5](#_Toc85882694)

[Part 3 – Add UI (11 Points) 5](#_Toc85882695)

[P3 – Forms - Certificate Generator Tool (25 Points) 6](#_Toc85882696)

# How to submit

After filling all the parts in this file, please follow the following steps.

1. Add your name and ID to the first page.
2. Save the file in the original format (Docx or Doc)

(please do not convert to other file formats e.g. PDF, ZIP, RAR, …).

1. Rename the file as

*YOUR First Name– YOUR Last Name–ID.docx*

**Example:** John-Smith-234566435.docx

1. Upload and submit your file (only via Blackboard).

**Note:** The final run results of each problem (screenshot or a copy of the results) should be added to the run result section “orange boxes”. Displaying the final run results accounts for 15% of the points given to each part.

# P1 – Pet Adoption System (40 Points)

## Part 1 - Database (30 Points)

This question requires PhpMyAdmin to create the **pet\_adoption** database. Follow the below instructions to create this database. Make sure to include the database scripts (CREATE, INSERT, etc.) with your PHP project. **They are also to be graded along with your PHP/HTML code.**

The database must be named as **pet\_adoption**. The database will have these tables:

**PET\_TYPES:**

* ID [INT] – Primary key and auto-generated
* NAME [VARCHAR(50) / NOT NULL] – Type of a pet.

**Notes:** Currently, there are only two supported pet types (**Dog** and **Cat**). You need to manually insert these pet types using the INSERT statements. Despite of having two supported pet types, make sure that no duplicates of pet types is allowed.

**PET\_BREEDS:**

* ID [INT] – Primary key and auto-generated
* NAME [VARCHAR(50) / NOT NULL] – Breed of a pet.
* IS\_DOG\_BREED [TINYINT / NOT NULL] – If set to 1, this is the dog breed. Otherwise, it’s the cat breed.

**Notes:**

* You need to manually insert the pet breeds using the INSERT statements. Make sure that there are no duplicates of dog breeds and/or cat breeds. In other words, create a unique key index on both columns (**NAME** and **IS\_DOG\_BREED**).
* Using the INSERT statements to insert the following cat breeds:
  + **Persian**
  + **Sanvannah**
  + **Japanese Bobtail**
  + **Himalayan**
* Using the INSERT statements to insert the following dog breeds:
  + **Boxer**
  + **Dalmatian**
  + **Golden Retriever**
  + **Shih Tzu**

**PETS:**

* ID [INT] – Primary key and auto-generated
* NAME [VARCHAR(50) / NOT NULL] – Name of a pet to adopted.
* BREED [INT / NOT NULL] – Foreign key to the pet\_breeds table.
* TYPE [INT / NOT NULL] – Foreign key to the pet\_types table.
* ARRIVAL\_DATE [DATETIME / NOT NULL] – Date when the pet arrives to the adoption center.
* OWNER\_FIRST\_NAME [VARCHAR(50) / NULL] – First name of the new owner.
* OWNER\_LAST\_NAME [VARCHAR(50) / NULL] – Last name of the new owner.
* OWNER\_MIDDLE\_NAME [VARCHAR(50) / NULL] – Middle name of the new owner.
* ADOPTION\_DATE [DATETIME / NULL] – Date when the pet is adopted.

**Notes:**

* Enable the delete and update cascading for foreign keys for **BREED** and **TYPE**.
* Insert the pets for adoption using the INSERT statements with the information below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pet\_name | Pet\_type | Pet\_breed | Arrival\_date | Owner\_full\_name | Adoption\_date |
| Fleur | Cat | Himalayan | 2021-03-15 21:01:57 |  | NULL |
| Dolly | Cat | Japanese Bobtail | 2021-03-15 21:01:57 |  | NULL |
| Cozette | Cat | Persian | 2021-03-15 21:01:57 |  | NULL |
| Delilah | Cat | Sanvannah | 2021-03-15 21:01:57 |  | NULL |
| Harold | Dog | Boxer | 2021-03-15 21:01:57 |  | NULL |
| Leo | Dog | Dalmatian | 2021-03-15 21:01:57 |  | NULL |
| Sapphire | Dog | Golden Retriever | 2021-03-15 21:01:57 |  | NULL |
| Yasmin | Dog | Shih Tzu | 2021-03-15 21:01:57 |  | NULL |

* The below SELECT statement is used to generate the above data table. You are welcomed to use it as is or to modify it to suite your needs.

SELECT p.name AS pet\_name,

t.name AS pet\_type,

b.name AS pet\_breed,

p.arrival\_date,

CONCAT(IFNULL(p.owner\_first\_name, ''),

IFNULL(p.owner\_last\_name, ''),

IFNULL(p.owner\_middle\_name, '')) AS owner\_full\_name,

p.adoption\_date

FROM `pets` p

INNER JOIN `pet\_types` t ON t.id = p.type

INNER JOIN `pet\_breeds` b ON b.id = p.breed

ORDER BY t.name, b.name;

Insert your SQL code for creation of database in the following box.

|  |
| --- |
| **Your SQL code for creating the database** |
| **CREATE TABLE PET\_TYPES(**  **ID INT AUTO\_INCREMENT PRIMARY KEY,**  **NAME VARCHAR(50) NOT NULL);**  **CREATE TABLE PET\_BREEDS(**  **ID INT AUTO\_INCREMENT PRIMARY KEY,**  **NAME VARCHAR(50) NOT NULL,**  **IS\_DOG\_BREED TINYINT NOT NULL);**  **CREATE TABLE PETS(**  **ID INT AUTO\_INCREMENT PRIMARY KEY,**  **NAME VARCHAR(50) NOT NULL,**  **BREED INT NOT NULL,**  **TYPE INT NOT NULL,**  **ARRIVAL\_DATE DATETIME NOT NULL,**  **OWNER\_FIRST\_NAME VARCHAR(50) NULL,**  **OWNER\_LAST\_NAME VARCHAR(50) NULL,**  **OWNER\_MIDDLE\_NAME VARCHAR(50) NULL,**  **ADOPTION\_DATE DATETIME NULL**  **);**  **INSERT INTO `PET\_TYPES` (`ID`, `NAME`) VALUES (NULL, 'DOG');**  **INSERT INTO `PET\_TYPES` (`ID`, `NAME`) VALUES (NULL, 'CAT');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'PERSIAN', '2');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'SAVANNAH', '2');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'JAPANESE BOBTAIL', '2');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'HIMALAYAN', '2');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'BOXER', '1');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'DALMATIAN', '1');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'GOLDEN RETRIEVER', '1');**  **INSERT INTO `PET\_BREEDS` (`ID`, `NAME`, `IS\_DOG\_BREED`) VALUES (NULL, 'SHIH ZHU', '1');**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'FLEUR', '4', '2', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'DOLLY', '3', '2', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'COZETTE', '4', '1', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'DELILAH', '4', '2', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'HAROLD', '5', '1', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'LEO', '6', '1', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'SAPPHIRE', '7', '1', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **INSERT INTO `PETS` (`ID`, `NAME`, `BREED`, `TYPE`, `ARRIVAL\_DATE`, `OWNER\_FIRST\_NAME`, `OWNER\_LAST\_NAME`, `OWNER\_MIDDLE\_NAME`, `ADOPTION\_DATE`) VALUES (NULL, 'SHI TZU', '8', '1', '2021-03-15 21:01:57.000000', NULL, NULL, NULL, NULL);**  **ALTER TABLE `PETS` ADD FOREIGN KEY (`BREED`) REFERENCES `PET\_BREEDS`(`ID`) ON DELETE CASCADE ON UPDATE CASCADE;**  **ALTER TABLE `PETS` ADD FOREIGN KEY (`TYPE`) REFERENCES `PET\_TYPES`(`ID`) ON DELETE CASCADE ON UPDATE CASCADE;**    **Graphical user interface, application  Description automatically generated**  **Also please add screenshots of the database tables in the phpMyAdmin environment here.** |

## Part 2 - PHP/HTML Programming (10 Points)

* Implement the home page (**index.php**) to display the above data table with two additional columns: **Edit** and **Delete**.
* The **Edit** column will allow the user to enter the new owner name and the adoption date.
* The **Delete** column will allow the user to remove the selected pet from the pet adoption center.

|  |
| --- |
| **Your PHP Code** |
| **-- Copy and paste your code here** |

Run the code and insert the result in the following box.

|  |
| --- |
| **The run result** |
| **Copy and paste the result here (e.g. the screen shot of the result you get by running the code)**  **Please add the result (screenshot) of data table as well as a sample case for deleting and editing one of the pets in the PETS table (result of clicking on the delete and edit buttons).** |

# P2 Cities (35 Points)

## Part 1 – Arrays - City Distances (12 points)

Consider the following X and Y values for the locations of 5 cities.

**City# X Y**

----------------------------------------

1| 2.5 5

2| 5.1 3

3| 1 9

4| 5.4 54

5| 5.5 2.1

Write a PHP program that calculates the distance among all cities and display it as a 5 \* 5 table. The element at row = i and col = j will show the distance between cities i and j. For example, row 2 and col 3 will show distance between city 2 and city 3.

Note: Distance between two cities is defined as Euclidian distance. For example, the distance between city 2 and city 3 will be:

D = sqrt ( (x2-x3)^2 + (y2-y3)^2 ) = sqrt ( (5.1-1)^2 + (3-9)^2 ) = 7.267

|  |
| --- |
| **Your PHP Code** |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name ="viewport" content ="width=device-width, initial-scale=1.0">  <title>Home</title>  <link rel="stylesheet" href="styles.css">  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">  </head>  <body>  <div class="container">  <table class="table">  <thead>  <tr>  <th scope="col">Distance From City1</th>  <th scope="col">Distance From City2</th>  <th scope="col">Distance From City3</th>  <th scope="col">Distance From City4</th>  <th scope="col">Distance From City5</th>  </tr>  </thead>  <tbody>  <?php  $cities = array(  array(2.5,5),  array(5.1,3),  array(1,9),  array(5.4,5.4),  array(5.5,2.1),  );  $distance = array();  //print\_r($cities);  //print\_r($city1);  for ($i = 0; $i < COUNT($cities); $i++) {  echo "<tr>";  for ($j = 0; $j < COUNT($cities); $j++) {  $x1 = $cities[$i][0];  $x2 = $cities[$j][0];  $y1 = $cities[$i][1];  $y2 = $cities[$j][1];  $ans = sqrt( pow(($x1-$x2),2) + pow(($y1-$y2),2) );  array\_push($distance, $ans);  echo "<td>".$ans."</td>";  }  echo "</tr>";  }  ?>  </tbody>  </table>    </div>    </body>  </html> |

Run the code and insert the result in the following box.

|  |
| --- |
| **The run result** |
| **Graphical user interface, application  Description automatically generated** |

## Part 2 - Central Point (12 Points)

Lets assume we want to build a central big hospital/medical research in one of these cities and want to find the city that can be the best choice for all the other cities to access it. Given a set of cities, the central city is the city that has the shortest total distance to all other cities. Write a PHP program that finds the central city and its total distance to all other cities for the 5 cities in part 1.

**Hint:** This is not much of additional work (perhaps just another 10 lines of code). You just need to find the total (sum) distance from each city to all other cities and the city with the shortest total distance to all other cities.

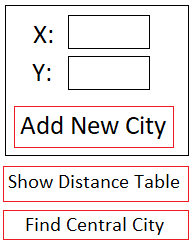
|  |
| --- |
| **Your PHP Code** |
| **-- C**<!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name ="viewport" content ="width=device-width, initial-scale=1.0">  <title>Home</title>  <link rel="stylesheet" href="styles.css">  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">  </head>  <body>  <div class="container">  <table class="table">  <thead>  <tr>  <th scope="col">City</th>  <th scope="col">Total Distance</th>  </tr>  </thead>  <tbody>  <?php  $cities = array(  array(2.5,5),  array(5.1,3),  array(1,9),  array(5.4,5.4),  array(5.5,2.1),  );  $distance = array();  $sum = 0;  $min = PHP\_INT\_MAX;  $city = 0;  for ($i = 0; $i < COUNT($cities); $i++) {  echo "<tr>";  $sum = 0;  for ($j = 0; $j < COUNT($cities); $j++) {  $x1 = $cities[$i][0];  $x2 = $cities[$j][0];  $y1 = $cities[$i][1];  $y2 = $cities[$j][1];  $ans = sqrt( pow(($x1-$x2),2) + pow(($y1-$y2),2) );  //array\_push($distance, $ans);  $sum += $ans;  //echo "<td>".$ans."</td>";    }  if ($sum < $min) {  $min = $sum;  $city = $i+1;  }    $cityNumber = $i+1;  echo "<td>City ".$cityNumber."</td>";  echo "<td>".$sum."</td>";  echo "</tr>";  if($cityNumber == COUNT($cities)) {  echo "The answer is city ".$city;  }  }    ?>  </tbody>  </table>    </div>    </body>  </html> |

Run the code and insert the result in the following box.

|  |
| --- |
| **The run result** |
|  |

## Part 3 – Add UI (11 Points)

Modify the code in part 1 and 2 and add the following UI (Form) to the code.



**Image:** this image shows the form to be created. We have two input fields X and Y and three buttons: 1) “Add New City”, 2) “Show Distance Table”, 3) “Find Central City”

After running the code, by entering X and Y and clicking on “Add New City” we add a new city. This process continues until all the cities are entered. Then we click on “Show Distance Table” to see the distance table (result of part 1). Clicking on the “Find Central City” will show the central point (result of part 2). Test your code with the list of cities given in part 1 (entered one by one) and show the result.

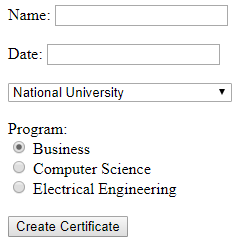
|  |
| --- |
| **Your PHP Code** |
| **-- Copy and paste your code here** |

Run the code and insert the result in the following box.

|  |
| --- |
| **The run result** |
| **Copy and paste the result here (e.g. the screen shot of the result you get by running the code)** |

# P3 – Forms - Certificate Generator Tool (25 Points)

In this problem we are going to design a certificate generator tool. For this problem, the design of the certificate (both HTML and CSS files) are given in the **“Problem 3”** folder. You just need do add the following form to the page and then link it to the certificate fields using PHP code. By filling in the form and clicking on the “Create Certificate” the completion certificate is generated.



**Image:** This image shows the certificate form. We have two text inputs: 1) Name of the student, 2) Date. We have also a dropdown list of three university names 1) National University, 2) San Diego State University, 3) University of California San Diego. Finally we have a radio option of three programs to be selected: 1) Business, 2) Computer Science, 3) Electrical Engineering. At the end we have “Create Certificate” Button.

**Note:** No design work is needed for this problem. All the files (HTML, CSS and images) are given. Just the above form should be added to the page and then the fields should be linked to the certificate fields, so that when we click on “Create Certificate” button, then the certificate with info entered by the user is created.

If you need more details about this problem and expected output, please watch the following 3-minute YouTube Video (the video has no voice. Please CTRL+Click on the link or simply copy and paste it in the address bar of your web browser).

YouTube Video: <https://www.youtube.com/watch?v=7otsJLMJ6GU>

|  |
| --- |
| **Your PHP Code** |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>Online Certificate Generator</title>  <meta charset="utf-8">  <link rel="stylesheet" href="css/style.css">  <meta name="viewport" content="width=device-width, initial-scale=1">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name ="viewport" content ="width=device-width, initial-scale=1.0">  <title>Form</title>  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">  </head>    <body>  <div class="container"></div>  <main>    <div id="main-content">    <?php  $selected = $\_POST['option'];  $image = "images/National-University.png";  if ($selected == "UCSD") {  $image = "images/University-of-California-San-Diego.png";  } else if ($selected == "SDSU") {  $image = "images/San-Diego-State-University.png";  }  echo "<img src=".$image." width='110px' id='logo-image'></img>";  ?>      <p id="university-name"><?php echo $\_POST['option']?></p>    <p id="certificate-of-completion">Certificate of Completion</p>  <p id="acknowledges">THIS ACKNOWLEDGES THAT</p>  <p id="student-name"><?php echo $\_POST['name']?></p>  <p id="successfully-ompleted">HAS SUCCESSFULLY COMPLETED</p>  <p id="program-name"><?php echo $\_POST['radio']?></p>  <p id="date">Date: <?php echo $\_POST['date']?></p>  </div>    <div id="form">  <!-- Add your form here -->  <form action="" method="post">  <div class="form-group">  <label for="exampleInputEmail1">Name</label>  <input type="name" class="form-control" name="name">  </div>  <div class="form-group">  <label for="exampleInputPassword1" >Date</label>  <input type="date" class="form-control" name="date">  </div>    <div class="form-check">  <input class="form-check-input" type="radio" name="radio" value="Business" checked="checked">  <label class="form-check-label" for="business">  Business  </label>  </div>  <div class="form-check">  <input class="form-check-input" type="radio" name="radio" value="Computer Science">  <label class="form-check-label" for="computerScience">  Computer Science  </label>  </div>  <div class="form-check">  <input class="form-check-input" type="radio" name="radio" value="Electrical Engineering">  <label class="form-check-label" for="electricalEngeering">  Electrical Engineering  </label>  </div>    <select class="form-select form-select-lg mb-3" aria-label=".form-select-lg example" name="option">  <option value="National University" name="option">National University</option>  <option value="SDSU" name="option">San Diego State University</option>  <option value="UCSD" name="option">University of California San Diego</option>  </select>    <button type="submit" class="btn btn-primary">Create Certificate</button>    </form>  </div>      </main>      <footer>  <hr>  <p>Design assignment. For educational use.</p>  <hr>  </footer>  </div>  </body>  </html> |

Run the code and create a certificate for Name: John Smith, Date: 10/20/2020 University: San Diego State University, Program: Computer Science and insert the result in the following box.

|  |
| --- |
| **The run result** |
| **Graphical user interface, text, application, email  Description automatically generated** |

**The end**