6. SUPPORT PROVIDED

a. Tasks in this assignment can be done mainly using the prescribed chapters of the textbook.
b. Try and make use of Google to find solutions to errors. As a PHP-developer you WILL spend hours on Internet forums seeking solutions to problems.
c. Make use of PHP documentation available online (see chapter 2 on how to access it)
d. Participate in *my*Unisa forum
e. E-mail the lecturer for help

7. ASSIGNMENT 3

1. Make use of comments in your code
2. All the tasks and subtasks (excluding code in the iframe) will produce output of some sort to the screen. We first consider the output produced, and then look at the code to see how the output was produced.
3. When a task has subtasks (marked using (a) and (b)), label the subtasks clearly in the output wherever possible.

Task 1: Chapter 13: page name=task1.php **24 marks**

This task consists of two different subtasks.

(a) **[12 marks]**

Code a function (`boolToText()`) that takes two arguments; the first argument takes a boolean value (0 or 1) and the second argument takes the format in which the given boolean value must be displayed. The default value for the second argument is 1.

When the value of the second argument is `1`, the function displays `"False"` for `0` and `"True"` for `1`. When the value of the second argument is `2`, the function displays `"No"` for `0` and `"Yes"` for `1`. When the value of the second argument is `3`, the function displays `"Negative"` for `0` and `"Positive"` for `1`. For any other format, the function displays `0` for `0` and `1` for `1`.

Invoke the function four times as follows:

```
boolToText(1);
boolToText(0, 2);
boolToText(1, 3);
boolToText(0, 5);
```

(b) **[12 marks]**

Write code for a function that accepts any number of arguments, and displays the total number of arguments and the total number of numeral arguments passed to it in each invocation. The function must make use of a variable-length parameter list.

For example, if the function is invoked with the arguments `"Thando", 23, "Busi", 40`, it will display the following:

```
Total number of arguments: 4, total number of numerals in these arguments: 2
```

On the other hand, if the function is invoked with the argument `"Mutsa"`, it will display the following:

```
Total number of arguments: 1, total number of numerals in these arguments: 0
```

Write code to invoke the function twice with arguments:
(1) `"Thando", 23, "Busi", 40`
(2) `"Mutsa"`

Task 2: Chapter 14: page name=task2.php **15 marks**

Write code for a class named `Square` with two properties; one to store the name of the shape (in this case `"Square"`) and one to represent the length of one side of a square. The class must have

- a constructor that accepts one parameter to initialise the length of the square. The constructor also initialises the name of the shape to `"Square"`
- getter methods to return values of properties
- one setter method to change the value of the length of one side of a square
- a method `getArea()` that calculates and returns the area of a square.
  Area of a square is $length\ of\ one\ side \times length\ of\ one\ side$
- a method `getPerimeter()` that calculates and returns the perimeter of a square
  Perimeter of a square is $4 \times length\ of\ one\ side$

Create an object of the `Square` class. Demonstrate how each method in the class is invoked using the created `Square` object and display output for each method invocation, where appropriate.

Task 3: Chapter 14: page name=task3.php **25 marks**

ICT3612 has three assignments. Assignments 1, 2 and 3 contributes 10%, 10% and 80% respectively toward the year mark. For example, if a student obtained 70%, 80% and 50% for assignments 1, 2 and 3 respectively, their year mark will be 55 (70×.1+80 ×.1+50 ×.8).

- Write code for a class named `AssignmentRecord` that satisfies the following:

- It has four private properties for storing a student number and three assignment marks.
- It has three class constants to store the weights of assignment contributions to the year mark. Initialise these constants to values .1,.1 and .8.
- It has a constructor with four parameters, which are used to initialise the properties of the class.
- It has a method that calculates and returns the year mark. Use the properties and constants of the class to calculate the year mark.
- It has a method `__toString()` that returns values of the properties of the class in comma separated values format.
For example, this method should return 123456,70,80,50 for an assignment record of a student (123456) with marks 70%, 80% and 50% for assignments 1, 2 and 3 respectively.

- Write code for a class named `FullRecord`, a subclass of `AssignmentRecord`, that satisfies the

following:

- It has one private property to store the exam mark of a student.
- It has a constructor with five parameters that are used to initialise the properties of this class and `AssignmentRecord`.
- It has a method `__toString()` that returns values of the properties of this class and `AssignmentRecord` in comma separated format of the This method must use the value returned by `__toString()` in `AssignmentRecord`.

For example, this method should return 123456,70,80,50,55 for a student (123456) with marks 70%, 80%, 50%, 55% for assignments 1, 2 and 3, and exam respectively.

- Write code to create a `FullRecord` object. Invoke the method to calculate the year mark and display the year mark for this object. Invoke the `__toString()` and display this string for this object.

Task 4: Chapter 15 : page name=task4.php **15 marks**

This task consists of two different subtasks.

(a) **[10 marks]**

Write a class named `Validate` that can be used to validate user selected usernames and passwords for a system.

The class has two static properties and they both store regular expressions to validate usernames and passwords. The username must only consist of four lowercase letters. The password must be a 6 to 8-digit number.

The class has two static methods. The first method accepts a username and validates it against the regular expression for username and returns true or false. The second method accepts a password and validates it against the regular expression for password and returns true or false.

Invoke the methods in `Validate` twice each, with usernames and passwords that will return true and false in different calls of the relevant methods.

(b) **[5 marks]**

For the given pattern `/^[01]?\d\/[0-3]\d\/\d{4}$/` present one string that will be accepted by the pattern and another string that will not be accepted by the pattern.

Code the pattern and the input strings in `preg_match()`. Display the values returned by `preg_match()` for both valid and invalid input strings.

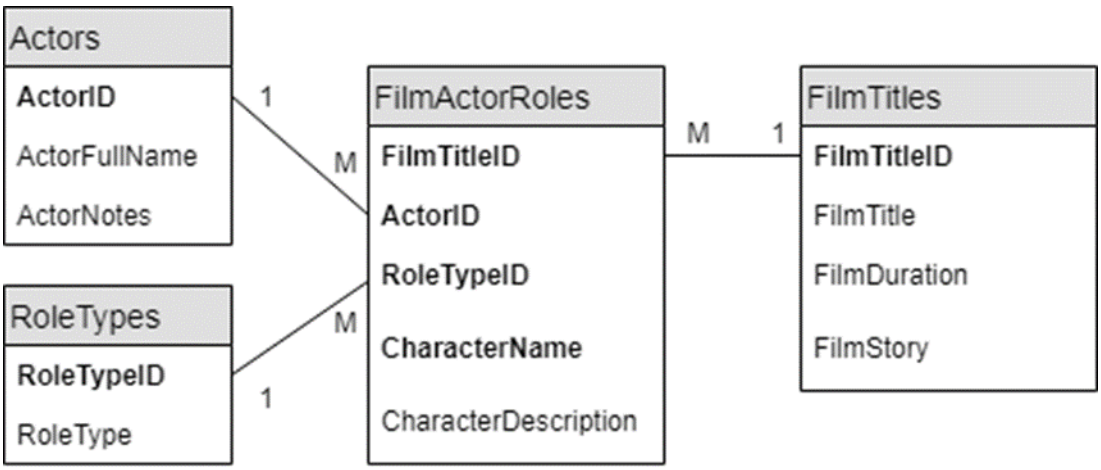
Task 5: Chapter 16: page name=task5.php **10 marks**

Refer to the figure in the Section titled 'How to apply the third normal form' in Chapter 16.

Explain briefly what each table represent (i.e. the entity it represents). List all the primary and foreign keys for each table.

Task 6: Chapter 16, 17, 18, 19: page name=task6.php **25 marks**

Create a database with four tables. The basic structure of the tables and relationships between the tables are illustrated below:



Populate the tables with realistic data, with at least 5 rows of data in each table. Note that the SQL data type for each column is not given in the diagram. So you have to decide on this aspect.

Write PHP code to connect to the database and display the contents of all four database tables in four different HTML tables on the web page.

Below the HTML tables, there must be a form with five radio buttons where the user can choose to view the results of one of the five different SELECT queries at a time. Details of these five SELECT queries are given below:

1) The first SELECT query must make use of the ORDER BY clause
2) The second SELECT query must make use of the LIKE operator
3) The third SELECT query must make use of an inner join that joins two tables
4) The fourth SELECT query must make use of a WHERE clause with the OR logical operator
5) The fifth SELECT query must make use of the aggregate function MAX.

You can decide on the other details of these SELECT queries.

Task 7: Chapter 21: page name=task7.php **12 marks**

List five secure website URLs, the respective certification authorities for the site's digital secure certificate and the date on which the digital secure certificate expires.

Task 8: Chapter 23: page name=task8.php **14 marks**

This task consists of two different subtasks. Both these subtasks involve modification of `task3.php`.

(a) Add a function named `writeToFile()` that accepts an array of `FullRecord` objects. The function iterates through the array and write the string returned by the `__toString()` function for each object into a file named `fullrecords.txt`. Each line of data `fullrecords.txt` represent the string representation of a different object.

Invoke the function with an array of at least five `FullRecord` objects.

Create a hyperlink pointing to the file `fullrecords.txt` and include it on the task page. When the user clicks on the link, the browser can then download it. (7)

(b) Write code to read the file `fullrecords.txt` in order to create an array of `FullRecord` objects. An object of `FullRecord` is created from the information given in each line of the file, and each created object is added to an array. Iterate through the array and invoke `__toString()` on each object. Display the string returned by `__toString()` for each object. (7)

Task 9: Chapter 13 to 23: page name=task9.php **30 marks**

In this task you will provide a solution or part of a solution to a problem for a community of your choice. The solution you are expected to design and code is a web-based database application. The purpose of these tasks is for you to apply your skills to address a real community-based problem.

The term community should be taken in a broader sense and the choice of community is entirely yours. Examples of your communities are student, residential, work and religious communities. The problem you are trying to address could be based on your personal experience in the community or gathered from the members of a community.

Given below is an example of a problem and a possible web-based database solution to address this problem. This is a sample case – do not use this for your assignment, please identify a unique problem relevant to your community of choice.

Problem: Companies/municipalities are expected to consult residents of an area to build/include certain types of infrastructure (examples include cellphone and webcam towers). These consultations happen during scheduled face-to-face meetings with the residents, which are poorly attended. Residents want to have their opinion on the proposed infrastructure noted but feel overlooked because they could not attend face-to-face consultation meetings.

Solution: In addition to the face-to-face consultations, an online database application that allows residents to indicate their support or opposition to the proposed infrastructure development can address this issue to a certain extent.

(a) Explain the community-based problem you are trying to address in this question. Also explain your proposed solution to address the problem. Describe the problem and the solution using at least 5 sentences (see above description of a sample problem and a possible solution). Display this explanation on the task page. **(5)**

(b) The solution must have at least two database tables, user-friendly forms/options to add, modify and delete data from the database table and to view the data in the database table in appropriate formats. The code in the solution must be structured using the Model View Controller (MVC) Pattern. When using MVC pattern your solution will have numerous PHP files. For the purpose of displaying the code in iframe, you can include code in all the files into one txt file for this task. **(25)**