There are four simple tasks in this assignment and students need to submit all the four tasks and also host all tasks of assignment 2 on free hosting as you did for assignment 1.

Task 01: PHP Sessions –Explained with a simple login system

You just need to run the following php session related scripts in order to understand the session handling.

1. The first thing we need is a form (form.php) where people can enter their username and password. It could look like this:

```
<html><head><title>Login</title></head>
<body>
<form method="post" action="login.php">
<p>Username: <input type="text" name="username" /></p>
<p>Password: <input type="password" name="password" /></p>
<p><input type="submit" value="Let me in" /></p>
</form></body></html>
```

Fig 01

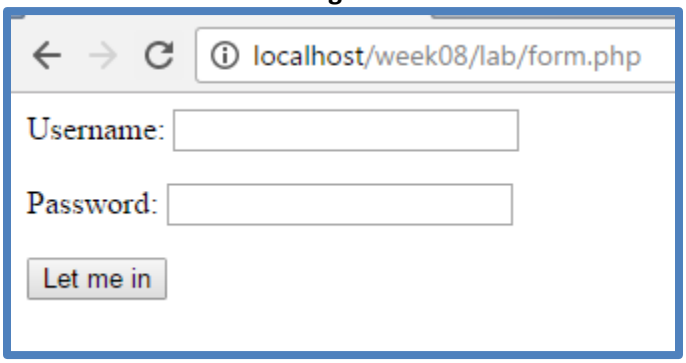


Fig 02

2. Then we create file: login.php. In this file, we check whether the correct username and password that has been entered. If that is the case, we set a session that says that this user is logged in with the correct username and password. [Change the hardcoded values of username to student's first name and password to his/her roll number instead of using "php" as shown in code.]

```
<html><head><title>Login</title></head><body>
<?php
session_start();
// Check if username and password are correct
if ($_POST["username"] == "php" && $_POST["password"] == "php") {
    // If correct, we set the session to YES
    $_SESSION["Login"] = "YES";
    echo "<h1>You are now logged correctly in</h1>";
    echo "<p><a href='document.php'>Link to protected file</a><p/>";
} else {
    // If not correct, we set the session to NO
    $_SESSION["Login"] = "NO";
    echo "<h1>You are NOT logged correctly in </h1>";
    echo "<p><a href='document.php'>Link to protected file</a></p>";
} ?>
</body></html>
```

Fig 03

3. In the protected file (document.php), we want to check whether the user is logged in properly. If this is not the case, the user is sent back to the login form. This is how the protection is made:

```php
<?php
//  Start up your PHP Session
session_start();
//  If the user is not logged in send him/her to the login form
if ($_SESSION["Login"] != "YES") {
    header("Location: form.php"); //Redirect to PHP form again.
}
?>
<html><head><title>Document</title></head><body>
<h1>This document is protected</h1>
<p>You can only see it if you are logged in.</p>
</body></html>
```

Fig 04

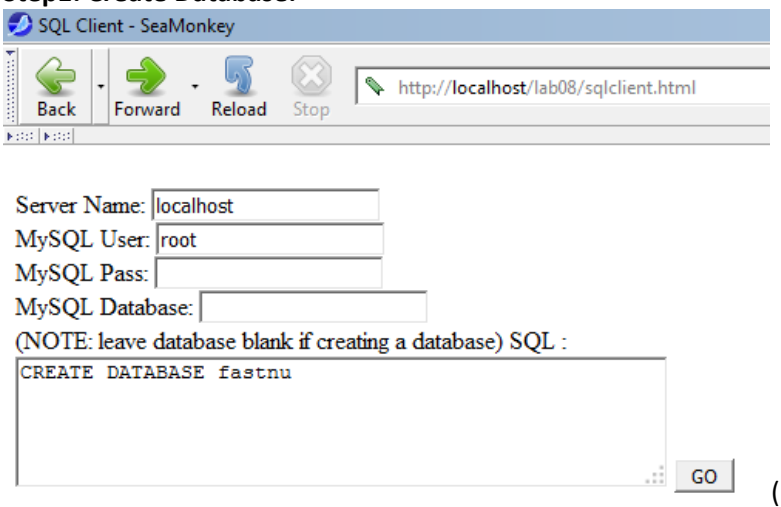
Task 02: PHP / MySQL

Students need to make a web based MySQL client in html5/php. Create form elements in HTML5 "**sqlclient.html**" [figure 5] and post the connection parameters and sql query to "**sqlclient.php**" script (figure 6). [Hint Use mysql_connect() and mysql_query() API in PHP page.] Follow the following screen shots in order to understand it's working. From this web based SQL client you need to be able to:

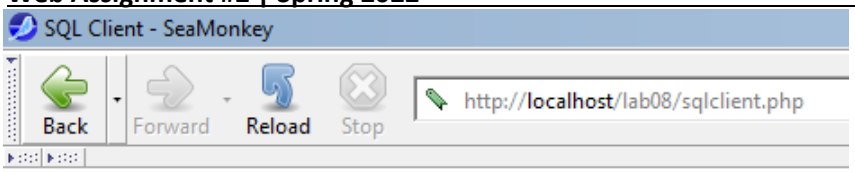
(1) create the database called fastnu
(2) create table ComputerScience
(3) insert records into the ComputerScience table.
(4) Update the record to confirm teacher Ather
(5) Insert email column in ComputerScience table after teacher

Students need to complete this task using **Pale Moon** (https://www.palemoon.org) browser and any one of the following MySQL clients: (1) **HeidiSQL** (https://www.heidisql.com/download.php) client OR (2) Command Line MySQL Client (built-in XAMPP) OR (3) MySQL workbench (https://www.mysql.com/products/workbench), OR (3) phpMyAdmin (comes with **XAMPP**). Just follow the following steps (step1 – step15), capture your own screen shots for each step. **Do not use SQLyog client OR SeaMonkey browser as for this tutorial I have already used its screenshots.**

Step1: Create Database:



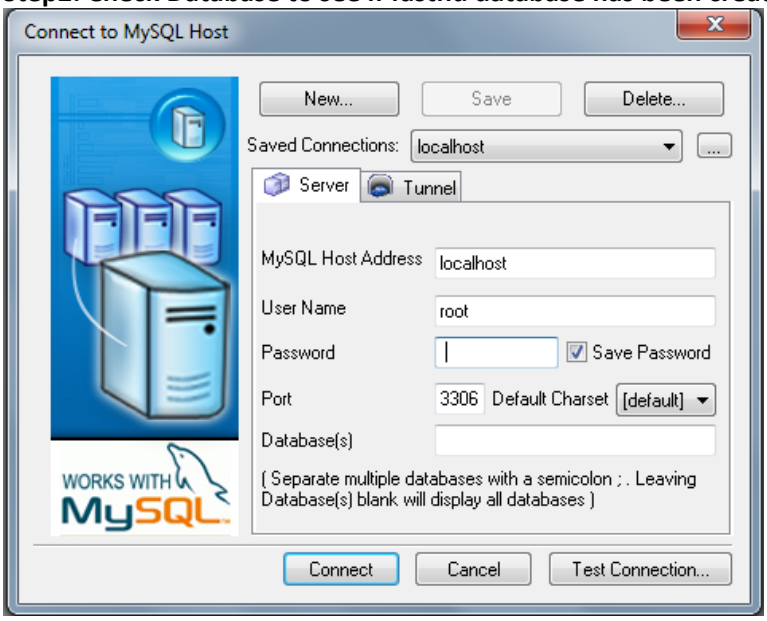
(**Fig 05**)

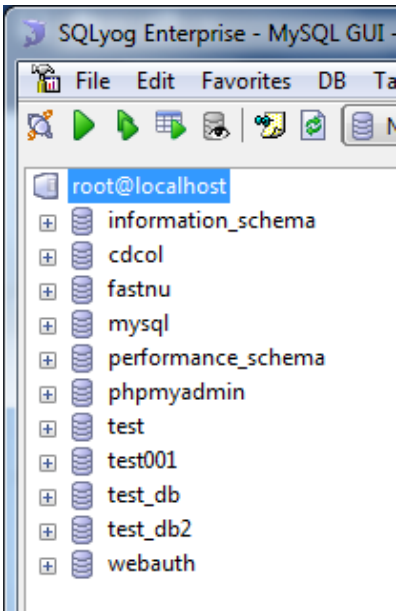


Querry result: 1
SQL :
CREATE DATABASE fastnu
executed! Check via HeidiSQL/SQLyog client to cnfirm.

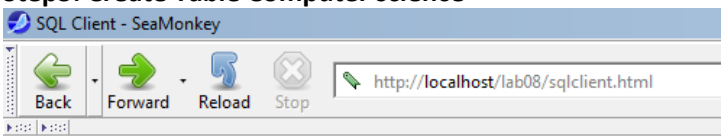
(**Fig 06**)

Step2: Check Database to see if fastnu database has been created:





Step3: Create Table Computer science



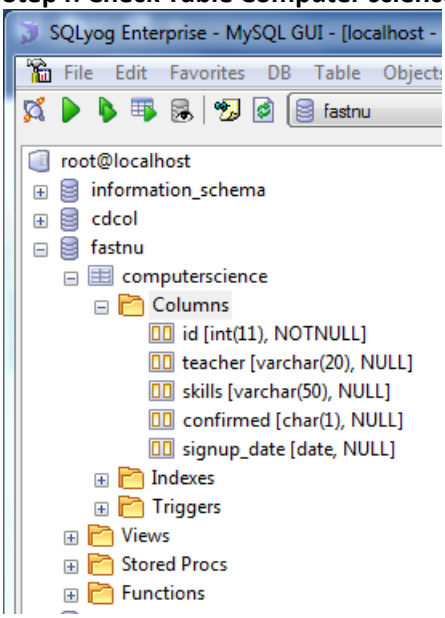
Server Name: localhost
MySQL User: root
MySQL Pass:
MySQL Database: fastnu
(NOTE: leave database blank if creating a database) SQL :

```
CREATE TABLE ComputerScience (id INT NOT NULL
PRIMARY KEY AUTO_INCREMENT,
teacher VARCHAR(20),
skills VARCHAR(50),
confirmed CHAR(1),
signup_date DATE);
```

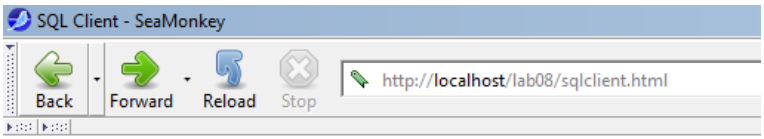
GO

Querry result: 1
SQL :
CREATE TABLE ComputerScience (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT, teacher VARCHAR(20), skills VARCHAR(50), confirmed CHAR(1), signup_date DATE);
executed! Check via HeidiSQL/SQLyog client to cnfirm.

Step4: Check Table Computer science



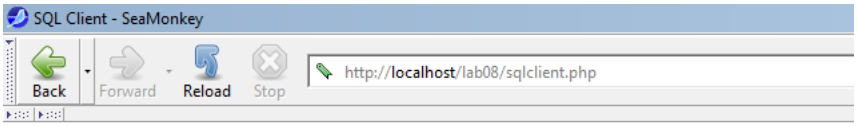
Step5: Insert record 1



Server Name: localhost
MySQL User: root
MySQL Pass:
MySQL Database: fastnu
(NOTE: leave database blank if creating a database) SQL :

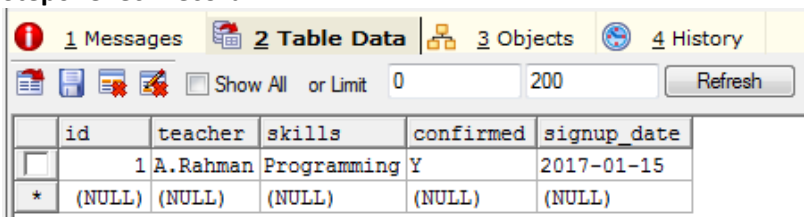
```
INSERT INTO ComputerScience VALUES (NULL,
'A.Rahman', 'Programming','Y', '2017-01-15');
```

GO



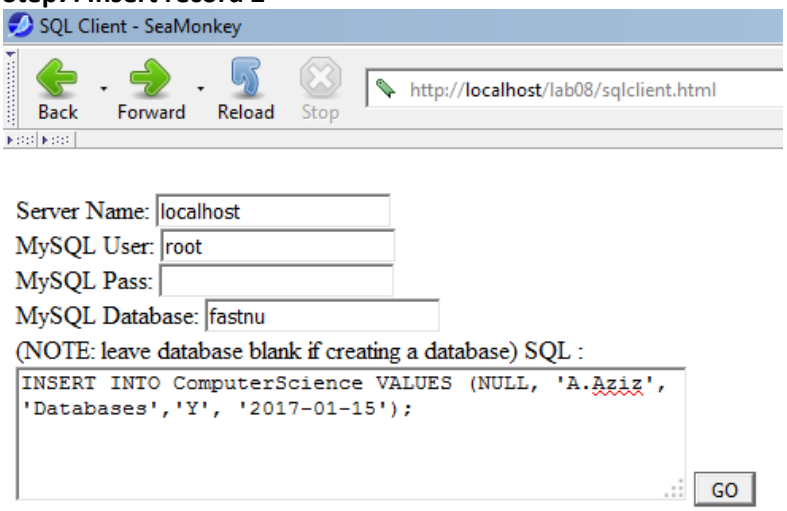
Querry result: 1
SQL :
INSERT INTO ComputerScience VALUES (NULL, 'A.Rahman', 'Programming','Y', '2017-01-15');
executed! Check via HeidiSQL/SQLyog client to cnfirm.

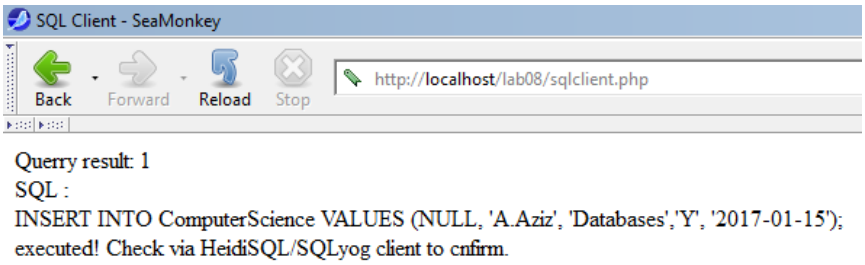
Step6: Check record 1



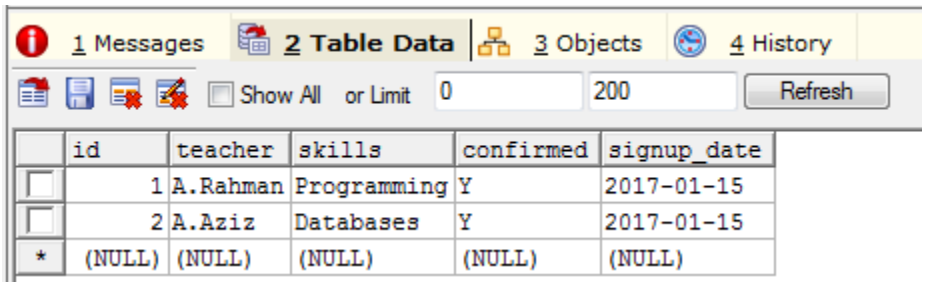
	id	teacher	skills	confirmed	signup_date
	1	A.Rahman	Programming	Y	2017-01-15
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Step7: Insert record 2



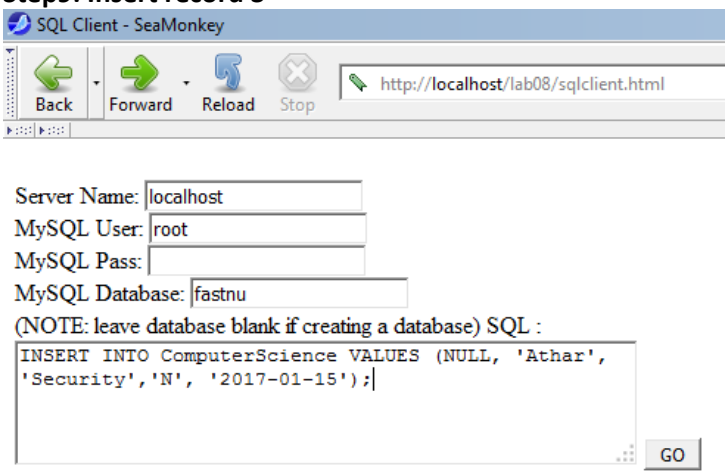


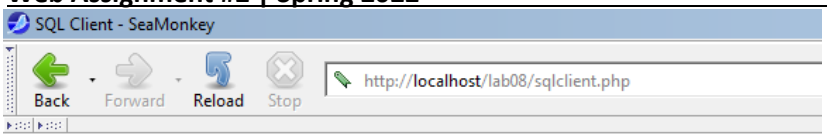
Step8: Check record 2



	id	teacher	skills	confirmed	signup_date
	1	A.Rahman	Programming	Y	2017-01-15
	2	A.Aziz	Databases	Y	2017-01-15
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

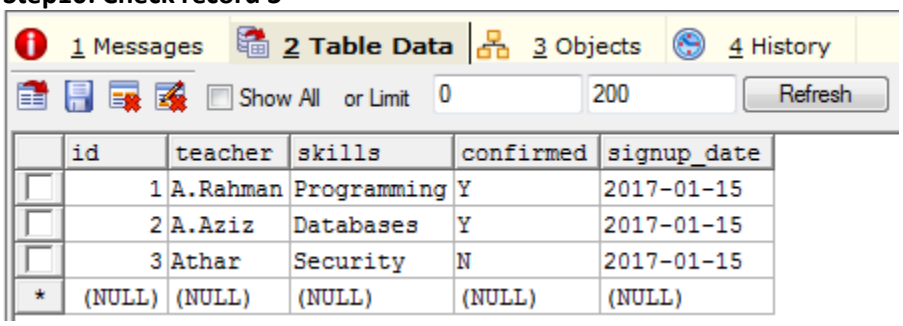
Step9: Insert record 3





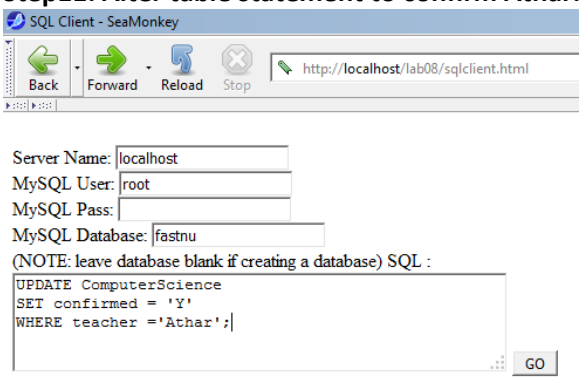
Querry result: 1
SQL :
INSERT INTO ComputerScience VALUES (NULL, 'Athar', 'Security','N', '2017-01-15');
executed! Check via HeidiSQL/SQLyog client to cnfirm.

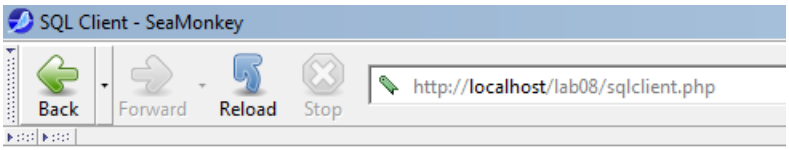
Step10: Check record 3



	id	teacher	skills	confirmed	signup_date
	1	A.Rahman	Programming	Y	2017-01-15
	2	A.Aziz	Databases	Y	2017-01-15
	3	Athar	Security	N	2017-01-15
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

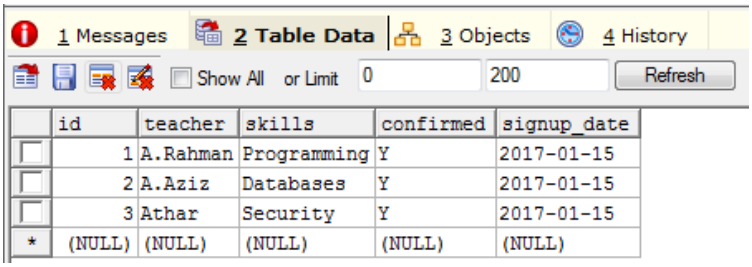
Step11: Alter table statement to confirm Athar.





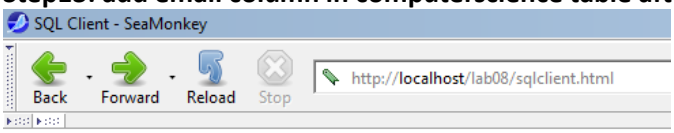
Querry result: 1
SQL :
UPDATE ComputerScience SET confirmed = 'Y' WHERE teacher ='Athar';
executed! Check via HeidiSQL/SQLyog client to cnfirm.

Step12: Check database again to confirm if Athar status has been updated to 'Y'.



	id	teacher	skills	confirmed	signup_date
	1	A.Rahman	Programming	Y	2017-01-15
	2	A.Aziz	Databases	Y	2017-01-15
	3	Athar	Security	Y	2017-01-15
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

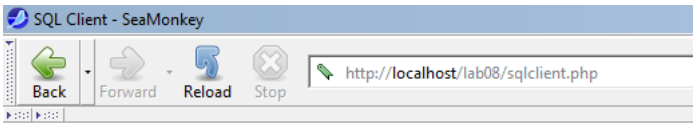
Step13: add email column in computerscience table after teacher:



Server Name: localhost
MySQL User: root
MySQL Pass:
MySQL Database: fastnu
(NOTE: leave database blank if creating a database) SQL :

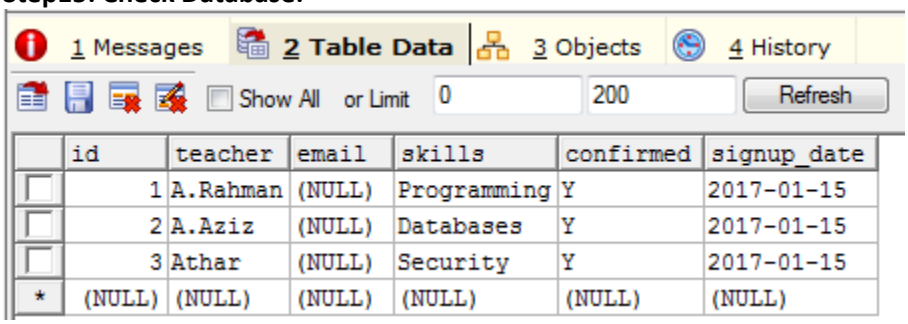
```
ALTER TABLE ComputerScience ADD email VARCHAR(45)
AFTER teacher;
```

GO



Querry result: 1
SQL :
ALTER TABLE ComputerScience ADD email VARCHAR(45) AFTER teacher;
executed! Check via HeidiSQL/SQLyog client to cnfirm.

Step15: Check Database:



	id	teacher	email	skills	confirmed	signup_date
☐	1	A.Rahman	(NULL)	Programming	Y	2017-01-15
☐	2	A.Aziz	(NULL)	Databases	Y	2017-01-15
☐	3	Athar	(NULL)	Security	Y	2017-01-15
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

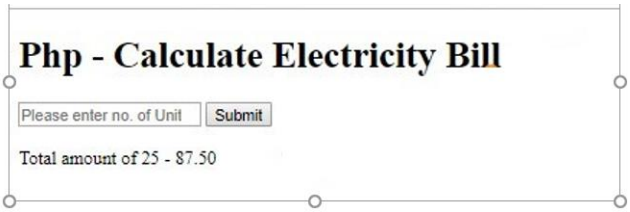
Task 03 : Write a program to calculate Electricity bill in PHP

Description:

You need to write a PHP program to calculate electricity bill using if-else conditions.

Conditions:

- ***For first 50 units – Rs. 3.50/unit***
- ***For next 100 units – Rs. 4.00/unit***
- ***For next 100 units – Rs. 5.20/unit***
- ***For units above 250 – Rs. 6.50/unit***
- ***You can use conditional statements.***



Task 04 : Write a program to calculate Electricity bill in PHP

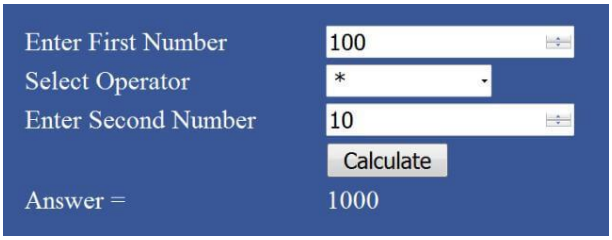
Write a simple calculator program in PHP using switch case

Description:

You need to write a simple calculator program in PHP using switch case.

Operations:

- ***Addition***
- ***Subtraction***
- ***Multiplication***
- ***Division***



Important Note:

1. Tasks in this assignment are very basic and easy to perform. So keep in mind plagiarism will be dealt strictly and no extension will be provided in deadline.
2. Date assigned **22-03-2022**. Last date of submission is **30-03-2022 11 AM.**
3. Assignments will not be accepted after due date.
4. Students are required to submit the assignment **individually.**
5. Plagiarism, if detected, will result in zero marks. (automation tools will be used to detect pleg, so be very careful)
6. Assignment must be submitted via slate / google forms only as announced. Will not be accepted on email.
7. A document report in MS Word format **(in /doc folder)** will be required. Contents of Report
   a. Cover Page of assignment must contain: Student name, Roll no, Date of submission and live **hosting URL**.
   b. Attach electronic snapshot of (this) assignment question after cover page.
   c. Screenshots of all the web pages / outputs etc.
8. Submit all the code of your portal (**in /code folder: /code/task1 /code/task2 /code/task3 /code/task4**).
9. **You also need to host your assignment to online free hosting and provide the url for the hosting of assignment#2 as you did for assignment#1. (do not overwrite your assignment 1 hosting URL instead use a new URL or sub-domain)**
10. Submit the single zip archive (**YourName_RollNo_WebSp22Ass02.zip**) of the assignment files.