

UNIT-2

PHP and MYSQL Programming

1. Create a form containing two input fields (Name, Email ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.
2. Write a PHP script for that creates a database named "DB-1" in MySQL.
3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.
4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.
5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.
6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.
7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.

Unit -2 PHP and MYSQL Programming

❖ Introduction:

In this unit we will discuss how to use advance PHP programming. Also will discuss inbuilt functions. In this unit will discuss MySQL Database connectivity functions and MySQL query functions. You have already studied about MySQL database in early semester and you are aware about how to create database and tables in MySQL. In this unit will discuss summary of database and tables, different types of data type available in MySQL. Now a day a newer version of MySQL is used with name of MySQLi where I means improved version of MySQL. MySQLi is a relational database driver used in the PHP scripting language to provide an interface with MySQL databases. We will discuss CRUD operations using sample programming.

2.1 How to Create Database in MySQL:

For creating data base in MySQL we use following steps.

Step 1: First we will start XAMP control panel, if you have already started then you directly write localhost in browser and click on phpMyAdmin.

Step 2: It will show you phpMyAdmin under this MySQL database will open.

Step 3: To create new database press on new link or databases in menu link. It will show you create database page inside this page enter database in textbox and press create button it will create new database.no need to change collection of database.

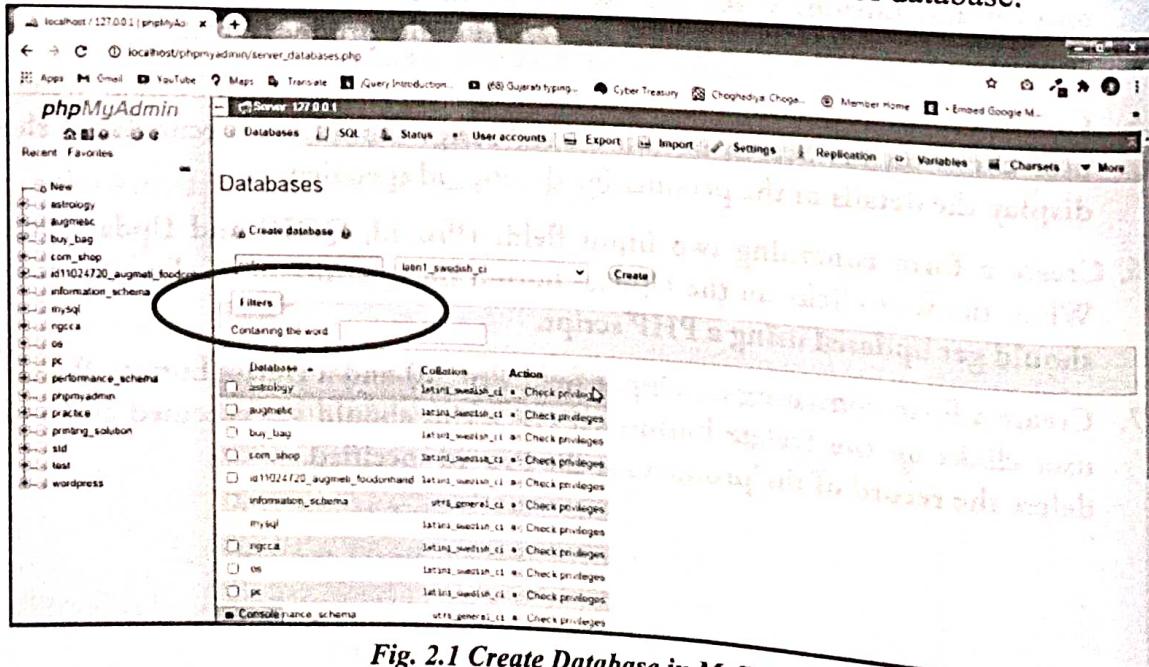


Fig. 2.1 Create Database in MySQL.

After this you will find your database in database list.

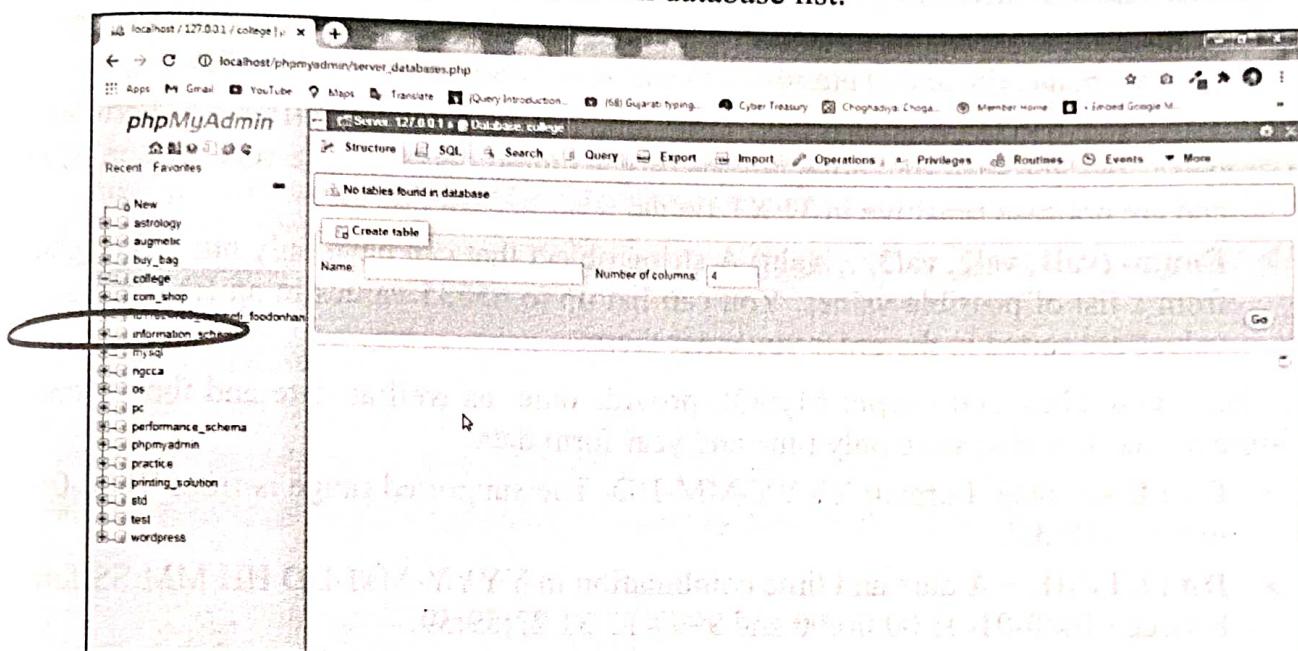


Fig. 2.2 Database Created in MySQL.

Before creating table we first learn about different data types in MySQL. This data types are allow storing your data in tables. We can use suitable data type to store a data. In below list I have defined all the data types which are most frequently used in real world.

2.2 Different Data types in Database:

2.2.1 Numeric Data type: In this data type we define different numeric types including int, samllint, tinyint, bigint , float ,double.

- **Tinyint** size(-128 to 127)
- **Smallint** size(-32768 to 32767)
- **Int** size(-2147483648 to 2147483647) is a medium integer
- **Bigint** size(9223372036854775808 to 9223372036854775807) large integer
- **Float** (size, decimal) where size define total size of float number and decimal define decimal number place. Decimal precision can go to 24 places.
- **Double** (size, decimal) where size define total size of float number and decimal define decimal number place. Decimal precision can go to 53 places.

2.2.2 String Data type: In database most data are store in string format.String can hold anything from plain text to binary data such as images or files. Strings can be compared and searched based on pattern matching by using the LIKE operator, regular expression, and full-text search.

- **CHAR-** size(1 to 255) A fixed-length string characters.

- **VARCHAR**- size(1 to 255) – A variable-length string characters.
- **Blob**- size(65535) Is used to store binary large object. It also store binary large data such as image, document etc.
- **Text**- size(65535) it is also hold large amounts of data. The difference between the two is that the sorts and comparisons on the stored data are case sensitive on BLOBs and are not case sensitive in TEXT fields.
- **Enum**- (val1, val2, val3, ... valn) A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. The values are sorted in the order you enter them.

2.2.3 Date and Time Data type: MySQL provide date as well as date and time format to store a data. It is also store only time and year form data.

- **DATE** - A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'.
- **DATETIME** – A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.
- **TIME** – Stores the time in a HH:MM:SS format.
- **YEAR** - Stores a year in a 2-digit or a 4-digit format.

2.3 How to create table in MySQL.

To create a table in MySQL database first you have to select a database in which you want to create a tables. After that you have to follow bellow steps.

Step1: when you select database you will find create table below this label you will find textbox.it will show you following screen.

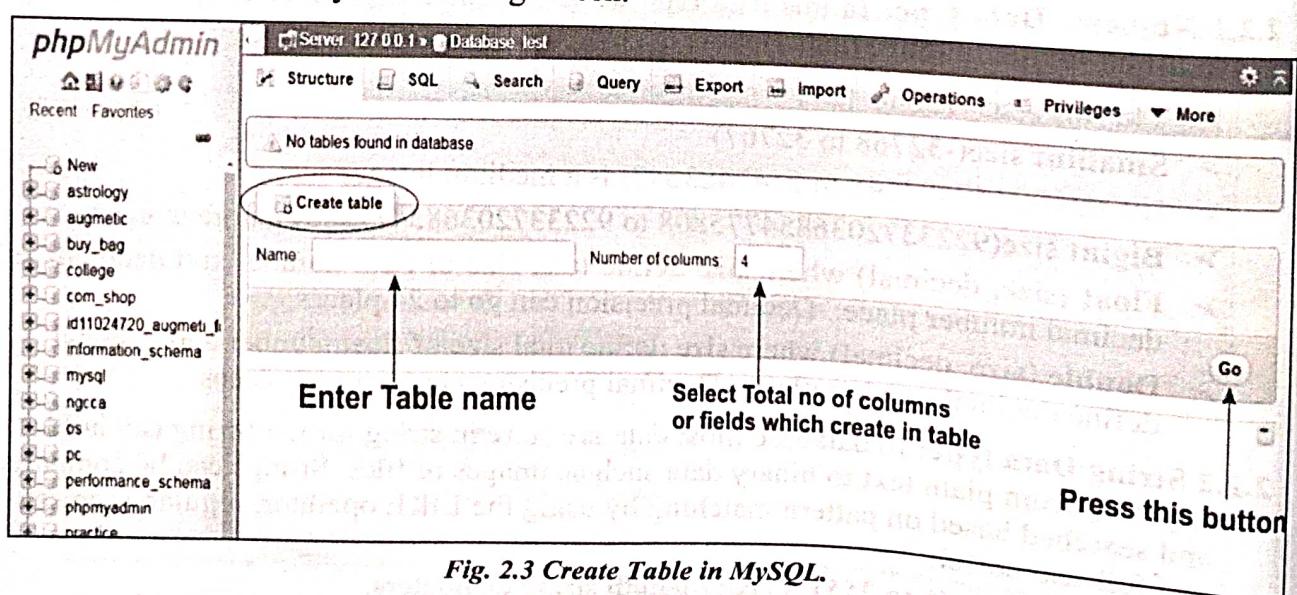


Fig. 2.3 Create Table in MySQL.

Step2: inside this textbox you have to write table name for example I want to create table admin then you will write admin.

Step3: After giving table name you have to select number of columns, if in your table there are 3 columns then you have to select 3. After giving name and selecting number of columns press /click on go button at right side.it will create table inside database it will show you following screen.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	AJ
	INT		None					
	INT		None					
	INT		None					
	INT		None					
	INT		None					
	INT		None					
	INT		None					
	INT		None					

Fig. 2.4 Table Attributes in MySQL.

In above screen we find different headers to design our table .These headers are used to mention table field name, size, data type, primary-key and auto-increment constrains.

Name: This field is used to mention field name, here we define any user define name of fields, like Stud_Id, Full_Name, Address etc. make sure while you give a name of field at that time not mention space between two words, but use underscore sign. Also give an meaning full name which will help you to identify proper field name.

Type: This field is used to define data type of record which we will store inside the table. It has same data types like we used in programming language and also use in oracle and other database. In above topic I have mention each data type with its capacity in size to store number of characters or digits. For example I have mention Full_Name field and name will store in alphabets then we select varchar data type to store a full name

Size Length/Values: This field is used to define a size or length of data which we store in data base. IT defines how many? Number of digits or characters or text we can store in single field. For example I have mention Full_Name field then we assume that full name will be up to 100 character length not more then, so we will mention 100 as a size. Make sure size is matter, while you prepare a database put only required size so on database server it will not load while you work with query string.

Default: This field is used to set default value in field. Suppose there is some specific value is enter in record then at that time we select As define and then will enter that

value inside default header. If it is none then it will store data while user will enter. If you want to allow a field to store null record then we will select NULL as a default value.

Null: It is in form of check box. While we want to allow a field to add null data, then at that time we will check the null in that particular field only. For example we want to set Pan_card_number field in student record but every student don't have pan card then we will select null in that field so it will allow null pan card number in record.

Index: This is used to set an index like primary key, unique, index etc. while we set any field as an primary key then we will select primary in index for that particular field only. For example we want to set Student_Id as a primary key then we will set index as a primary key in that field.

AI Auto Increment: This is in form of check box. If we want to set any field record will enter auto increment value then we will check this. Most probably it will work with primary key fields record where we set primary key and also the record of primary key will auto increment.

Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
Student_ID	int(5)			No	None		AUTO_INCREMENT	<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Fullname	varchar(50)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Address	text	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
City_Id	int(4)			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Pincode	int(6)			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Email	varchar(100)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Mobile	varchar(14)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
DOB	date			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
Gender	varchar(6)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More

Fig. 2.5 Table with all attributes in MySQL.

After this you will press on save button then it will create table in your database.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Student_ID	int(5)			No	None		AUTO_INCREMENT	<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
2	Fullname	varchar(50)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
3	Address	text	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
4	City_Id	int(4)			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
5	Pincode	int(6)			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
6	Email	varchar(100)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
7	Mobile	varchar(14)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
8	DOB	date			No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More
9	Gender	varchar(6)	latin1_swedish_ci		No	None			<input type="checkbox"/> Change <input checked="" type="radio"/> Drop <input type="checkbox"/> More

Fig. 2.6 Table Created in MySQL.

• Foreign Key Concept in MySQL:

For foreign key fields there is no foreign key mention as any index. For that we have to give the same name of primary key in to the foreign key field. For example in Student Table I want to add City_Id as a foreign key then I will give same name from primary key field inside the City Table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	City_Id	int(4)	latin1_swedish_ci		No	None		AUTO_INCREMENT	
2	City_Name	varchar(100)	latin1_swedish_ci		No	None			

Fig. 2.7 City Table With Primary-key City_Id.

- In above City Table City_Id is Primary key:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Student_ID	int(5)	latin1_swedish_ci		No	None		AUTO_INCREMENT	
2	Fullname	varchar(50)	latin1_swedish_ci		No	None			
3	Address	text	latin1_swedish_ci		No	None			
4	City_Id	int(4)	latin1_swedish_ci		No	None			
5	Pincode	int(6)	latin1_swedish_ci		No	None			
6	Email	varchar(100)	latin1_swedish_ci		No	None			
7	Mobile	varchar(14)	latin1_swedish_ci		No	None			
8	DOB	date	latin1_swedish_ci		No	None			
9	Gender	varchar(6)	latin1_swedish_ci		No	None			

Fig. 2.8 Student Table With Foreign-key City_Id.

2.4 Different PHP Functions related to MySQL Database.

2.4.1 Connecting Database with PHP program.

In PHP program while we want to send any request to the MySQL database or receive any response from MySQL database at that time we need to connect database with PHP program. After establish a connection with MySQL database we can access Database and their table using query in PHP program. Now a days MySQL functions are used with new version MySQLi, so we will study all new function of MySQLi.

mysqli_connect(): this function is used to establish connection with database. There are four parameters are pass as an arguments in this function all are mention with value as under. This function is initialize with variable because while connection function will execute successfully then, it will return 1 other wise 0 so we can check the return value for identifying whether connection function work properly or not.

```
mysqli_connect("hostname", "database username", "database password",
"database name");
```

hostname: it is hosting server name where you host your web application and there is MySQL database. In our local machine we install XAMPP,WAMP or LAMP at that time we enter hostname as a “**localhost**”.

Database username: in Mysql Database username is define at the time of creating database on hosting server. So we will define that database username which we have created. In our local machine while we install XAMPP,WAMP or LAMP at same time database is also install and default database username set as a “**root**” in ALL Apache platforms.

Database password: in Mysql Database Password is define at the time of creating database on hosting server. So we will define that database password which we have created. In our local machine while we install XAMPP,WAMP or LAMP at same time database is also install and default database password set as a empty so will define “” in ALL Apache platforms.

Database name: in Mysql Database name is define at the time of creating database on hosting server. So we will define that database name which we have created. In our local machine while we install XAMPP,WAMP or LAMP we create our own database at the time of creating database what ever the name we assign that name is given as database name for example we have create university database then name of database is “**university**”, same you can assing your own database name which we want to connect with PHP program.

mysqli_close(“connection variable name”): this function is used to close previously open database connection. We pass connection variable name as an argument of this function. I have mention this function in example.

mysqli_connect_error(): This function will pass error message from the last connection error, if any. I have explain this function in example. We can also define our own error message by checking the return value of connection variable, if it is return 0 then connection is not establish something wrong in mysqli_connect() function. I have mention in method 2 of connection.php file.

connection.php //method 1

```
<?php
$con=mysqli_connect("localhost","root","","university");
If(!$con)
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
    exit();
}
```

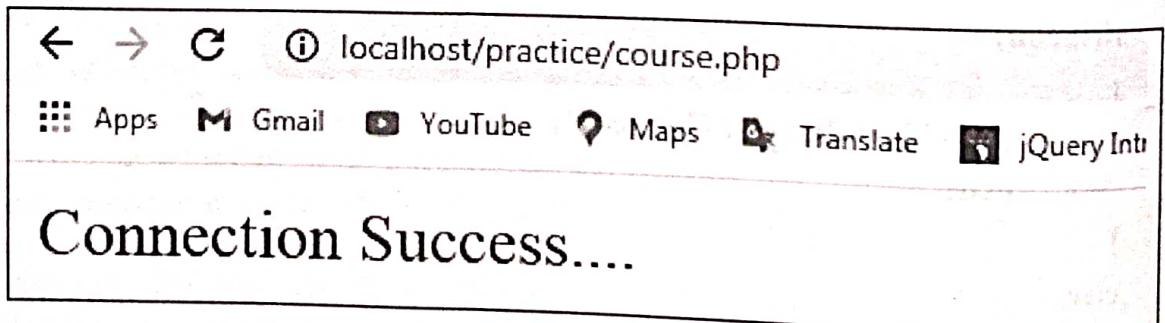
```
    }
else
{
    Echo "Connection Success....";
    exit;
}
mysqli_close($con);
?>
connection.php //method 2
<?php
// we can also mention this function in different way like. Initialize all the
parameter with variable and then pass that variable in function argument.
$host="localhost";
$dbuser="root";
$dbpassword="";
$dbname="university";
$con=mysqli_connect($host,$dbuser,$dbpassword,$dbname);

if(!$con)
{
    echo "Database not Connect";
    exit;
}
else
{
    Echo "Connection Success....";
    exit;
}
mysqli_close($con);
?>
connection.php // method 3
<?php
// we can also use IP address a host name
```

```

$host="127.0.0.1";
$dbuser="root";
$dbpassword="";
$dbname="university";
$con=mysqli_connect($host,$dbuser,$dbpassword,$dbname);
if(!$con)
{
    echo "Database not Connect";
    exit;
}
else
{
    Echo "Connection Success....";
    exit;
}
mysqli_close($con);
?>

```

*Fig. 2.9 Database Connection.*

Make sure while you work with PHP programs which use connection function repeatedly at that time we will make our connection file separately, with extension of PHP. Then we will include that connection file inside our PHP programs where our database connection is established and we can do request and respond on that page. It is more easy way to use connection function repeatedly in all PHP programs where database connection function is required.

2.4.2 Execution of Query in PHP Program.

In PHP program we can work with different database query strings like create table, insert record in table, select or search or retrieve data from table, update record in table or delete a record from table. For different operations we define query strings inside the

MySQL query function. There are two parameters are pass as an argument, first one is use connection variable as an object of database connection, which we have define in connection.php file. Second argument is query string which will pass as for request to the database and will execute the query. These two arguments are compulsory to define. Following function is used in PHP program where we want to execute a query. Before using this function we will add connection.php file or connection function with all argument then and then only query function will execute.

```

mysqli_query($con,"query string");
<?php
    $con=mysqli_connect("localhost","root","","university");
    if (!$con)
    {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
        exit();
    }
    $query= mysqli_query($con,"select * from course");
    if($query)
    {
        echo "Query Execute Successfully";
    }
    mysqli_close($con);
?>

```

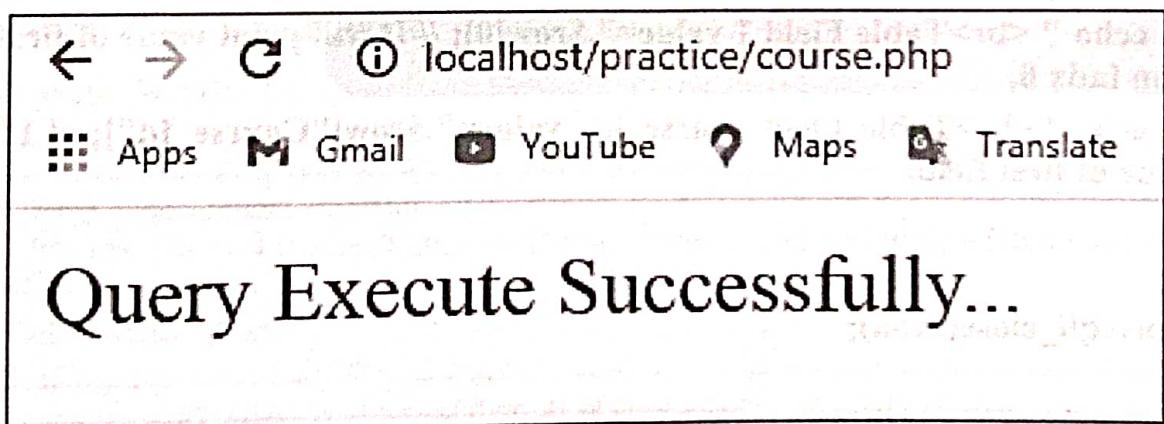


Fig. 2.10 Query Execution using mysql query function.

2.4.3 Fetch or retrieve records after execution of Query in PHP Program.

In PHP program while we want to retrieve or fetch records form database for that we used different fetch functions. This function will work after executing query function, so

`mysqli_query()` function is compulsory to execute. We use `select` query for retrieving or fetching data inside the query function. There are 4 fetch functions are use this functions are inbuilt function of PHP.

- **`mysqli_fetch_array()`:** In this function we will pass query variable which return the query result as an argument. This function is used to fetch record in index array, associative array or both. In bellow example `$row[0]` is mention as an index of table first field. `$row["Course_Id"]` is mention as a key of table first field.

Syntax: `mysqli_fetch_array(Query variable);`

`print_r()`: I have used this function to print result of `mysqli_fetch_array()` function. It will print whole array of single record. it will index array, associative array or both.

```
<?php
```

```
    echo "mysqli_fetch_array() function demo <br>";
```

```
    $con=mysqli_connect("localhost","root","","university");
```

```
    if(!$con)
```

```
{
```

```
    echo "Failed to connect to MySQL: ". mysqli_connect_error();
```

```
    exit();
```

```
}
```

```
    //echo "select * from course";
```

```
    $query= mysqli_query($con,"select * from course");
```

```
    $row=mysqli_fetch_array($query);
```

```
    print_r($row);
```

echo "
Table Field 1 value=". \$row[0]; // It will print value of first field as it is on indx 0.

echo "
Table Field course_id value=". \$row["Course_Id"]; // It will print value of first field.

```
    mysqli_close($con);
```

```
?>
```

mysli_fetch_array() function demo

Array ([0] => 1 [Course_Id] => 1 [1] => B.com [CourseName] => B.com [2] => Fulltime [CourseType] => Fulltime [3] => 3 [Duration] => 3)

Table Field 1 value=1

Table Field course_id value=1

Fig. 2.11 fetch record from database using `mysqli_fetch_array`.

- **mysqli_fetch_assoc()**: In this function we will pass query variable which return the query result as an argument. This function is used to fetch record in, Associative array only, means this function will show result with table field name as a Key of array. It will use field name as a key of array. In bellow example \$row["Course_Id"] is mention as a key of table first field. \$row["CourseName"] is mention as a key of table Second field.

Syntax: **mysqli_fetch_assoc(Query variable);**

<?php

echo "mysqli_fetch_assoc() function demo
";

\$con=mysqli_connect("localhost","root","","university");

if(!\$con)

{

echo "Failed to connect to MySQL: " . mysqli_connect_error();

exit();

}

//echo "select * from course";

\$query= mysqli_query(\$con,"select * from course");

\$row=mysqli_fetch_assoc(\$query);

print_r(\$row);

echo "
Table Field 1 value=". \$row["Course_Id"]; // It will print value of first field as it is on index 0.

echo "
Table Field 2 value=". \$row["CourseName"]; // It will print value of Second field as it is on index 1.

mysqli_close(\$con);

?>

mysqli_fetch_assoc() function demo

Array ([Course_Id] => 1 [CourseName] => B.com [CourseType] => Fulltime [Duration] => 3)

Table Field 1 value=1

Table Field 2 value=B.com

Fig. 2.12 fetch record from database using mysqli_fetch_assoc.

- **mysqli_fetch_row()**: In this function we will pass query variable which return the query result. This function is used to fetch record in, index array only, means this function will show result of table in order of index 0,1,2 ..n in array. In bellow example

\$row[0] is mention as an index of table first field. \$row[1] is mention as an index of table Second field and so on.

Syntax: mysqli_fetch_row(Query variable);

```
<?php
```

```
echo "mysqli_fetch_assoc() function demo <br>";
```

```
$con=mysqli_connect("localhost","root","","","university");
```

```
if(!$con)
```

```
{
```

```
echo "Failed to connect to MySQL: " . mysqli_connect_error();
```

```
exit();
```

```
}
```

```
//echo "select * from course";
```

```
$query= mysqli_query($con,"select * from course");
```

```
$row=mysqli_fetch_row($query);
```

```
print_r($row);
```

```
echo "<br>Table Field 1 value=". $row[0]; // It will print value of first field as it is  
on index 0.
```

```
echo "<br>Table Field 2 value=". $row[1]; // It will print value of Second field as  
it is on index 1.
```

```
mysqli_close($con);
```

```
?>
```

```
mysqli_fetch_assoc() function demo
```

```
Array ([0] => 1 [1] => B.com [2] => Fulltime [3] => 3 )
```

```
Table Field 1 value=1
```

```
Table Field 2 value=B.com
```

Fig. 2.13 fetch record from database using mysqli_fetch_row.

- **mysqli_num_rows():** This function is used to get total number of record fetch after executing a query. If your query will execute then it will get 5 records then it will print 5 as a value. It is also used to check total number of records available in table. In this function we will pass query variable as an argument of function.

Syntax: mysqli_num_rows(query variable);

```
<?php
```

```

$con=mysqli_connect("localhost","root","","university");
if(!$con)
{
echo "Failed to connect to MySQL: ". mysqli_connect_error();
exit();
}

$query= mysqli_query($con,"select * from course");

$row=mysqli_num_rows($query);
echo "Total Number of Records in Course Table=$row";
mysqli_close($con);
?>

```

As we show in result I got 5 as a value and also there are 5 records in course table.

	Course_Id	CourseName	CourseType	Duration
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	1	B.com	Fulltime	3
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	2	BA	Fulltime	3
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	3	BCA	Fulltime	3
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	4	BBA	Fulltime	3
<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	5	PGDCA	Fulltime	2

Fig. 2.14 Total number of records mysqli_num_rows.

2.5 PHP CRUD Operations:

In PHP CRUD Operations we work with Insert a new record, update record, Delete record. Also we will learn how to write a query in PHP with some rules. In CRUD Operation first step is to create a Database and Tables. Second step is to insert a record in

Database table using PHP program, in which we use some MySQL functions to send a request query and get response from database.

2.5.1 Insert Operation:

For insert new record in table we follow some steps.

Step 1: make an HTML form with action page and method to post data or insert a data in table.

Step 2: after making form we will make one action page where we will receive all the form data which are posted and after that we will use PHP global to receive posted data inside the action page.

Step 3: after receiving posted data we will use connection file to establish connection with database.

Step 4: after establishing a connection we will write insert query, for insert a record in table and then will use MySQL Query function to execute our insert query.

Here are some syntax rules to follow:

- The SQL query must be quoted in PHP.
- String values inside the SQL query must be quoted.
- Numeric values must not be quoted.
- The word NULL must not be quoted.

We will use following query to insert new record.

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...);
```

Note: if any auto increment fields (like "id" column) or TIMESTAMP with default update of current_timestamp (like the "reg_date" column), it is no need to be specified in the SQL query; MySQL will automatically add the value.

Step 5: after insert query will execute we will redirect to the next page where we can check our newly inserted record.

For Example:

std_reg.php

This file is used to draw a form where user will fill all the details and that detail will send for insert in database table.

```
<?php
include("conn.php");
$action="Add";
?>
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<script>
function can0
{
    window.location="index.php";
}
</script>
</head>
<body>
<form action="std_query.php" method="post">
<input type="hidden" name="Action" value="<?php echo $action;?>">
<input type="hidden" name="sid" value="">
<table>
    <tr>
        <td><label>Fullname</label></td>
        <td><input type="text" name="Fullname" value="" required></td>
    </tr>
    <tr>
        <td><label>Email</label></td>
        <td><input type="email" name="Email" value="" required></td>
    </tr>
    <tr>
        <td><label>Mobile</label></td>
        <td><input type="tel" name="Mobile" value="" required></td>
    </tr>
    <tr>
        <td><label>Password</label></td>
        <td><input type="password" name="Password" value="" required></td>
    </tr>
```

```

<tr>
    <td><label>State</label></td>
    <td><select name="State">
        <option value="">Please Select</option>
        <?php
            $state="select * from state order by State_Name ASC";
            $qry=mysqli_query($con,$state);
            while($res=mysqli_fetch_array($qry))
            {
                <option value="<?php echo $res["State_Id"];?>"> <?php echo
                    $res["State_Name"];?>
            }
        <?php
    </td>
</tr>
<tr>
    <td colspan="3" align="center">
        <input type="submit" name="submit" value="Submit">
        <input type="reset" name="reset" value="reset">
        <button name="cancle" value="cancle" onClick="can0();>Cancle</button>
    </td>
</tr>
</table>
</form>
</body>
</html>

```



Untitled Document X localhost

localhost/std/std_reg.php

Apps Gmail YouTube Maps

Fullname

Email

Mobile

Password

State

Fig. 2.15 Add student record in database.

- **std_query.php**

This file is used for writing a query. Here in this file I will write all Insert update and Delete query for Student record. After submitting a form this file will check the action, if action is "Add" then it will go for insert operation and add new record in student table.

```
<?php
    include("conn.php");
    $Action=$_POST["Action"];
    $sid=$_POST["sid"];
    $Fullname=$_POST["Fullname"];
    $Email=$_POST["Email"];
    $Mobile=$_POST["Mobile"];
    $Password=$_POST["Password"];
    $stid=$_POST["state"];
    if($Action=="Add")
    {
        $ins_qry="insert into student (Student_Id, Fullname, Email, Mobile, Password, State_Id) values ('', '$Fullname', '$Email', '$Mobile', '$Password', '$stid')";
        $res=mysqli_query($con,$ins_qry);
        if($res)
        {
            echo "RECORD INSERTED SUCCESSFULL ";
        }
        else
    }
```

```

    {
        echo "RECORD NOT INSERTED TRY AGAIN";
    }
}

<?php

```

	Student_Id	Fullname	Email	Mobile	Password	State_Id
<input type="checkbox"/>	1	Jigar K Patel	jigar@gmail.com	1111122222	12341234	2
<input type="checkbox"/>	8	Jigar Patel	jigar123@gmail.com	1254545747852	j123	3
<input type="checkbox"/>	13	Ghanshyam	ghanshyam@gamil.com	12312312312	123456	3
<input type="checkbox"/>	15	Milin Kanajiya	milin123@rediffmail.com	8899007762	123123	5
<input type="checkbox"/>	16	Dhrupal	dhrupal123@ymail.com	1233456780	123123	4
<input type="checkbox"/>	18	Umesh Patel	umesh.patel@yahoo.co.in	09427607080	umesh123	3
<input type="checkbox"/>	21	Ashok Patel	ashok787@hotmail.com	9898989898	ashok@123	5

Fig. 2.16 insert new record in database using insert query.

After adding a new record in Database table we can view a record on our web page for that we will make one more page student_record.php. This page is used to print a record in table format. In this page we will write select query to fetch all the records from table.

student_record.php

```

<?php
    include("conn.php");
    session_start();
    if(isset($_SESSION["Username"]))
    {
        $username=$_SESSION["Username"];
    }
    $selqry="select *from student";
    $res=mysqli_query($con,$selqry);
    $msg=$_GET["msg"];
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

```

```

<body>
<table border="1">
  <tr>
    <th>Student Id</th>
    <th>Fullname</th>
    <th>Email</th>
    <th>Mobile</th>
    <th>State</th>
    <th>Edit</th>
    <th>Delete</th>
  </tr>
  while($row=mysqli_fetch_array($res))
  {
    ?>
    <tr>
      <td><?php echo $row["Student_Id"];?></td>
      <td><?php echo $row["Fullname"];?></td>
      <td><?php echo $row["Email"];?></td>
      <td><?php echo $row["Mobile"];?></td>
      <td><?php
        $st="Select State_Name from state where
        State_Id='".$row["State_Id"]."';"
        $str=mysqli_query($con,$st);
        $strow=mysqli_fetch_array($str);
        ?>
        <td><?php echo $strow["State_Name"];?></td>
        <td><a href="std_reg.php?Action=Edit&sid=<?php echo
        $row["Student_Id"];?>">Edit</a>
        </td>
        <td><a href="std_query.php?Action=Delete&sid=<?php echo
        $row["Student_Id"];?>">Delete</a></td>
      </tr>
    <?php
  }
}
  
```

```
?>
</table>
</body>
</html>
```

Student Id	Fullname	Email	Mobile	State	Edit	Delete
1	Jigar K Patel	jigar@gmail.com	1111122222	MP	Edit	Delete
8	Jigar Patel	jigar123@gmail.com	1254545747852	MH	Edit	Delete
13	Ghanshyam	ghanshyam@gamil.com	12312312312	MH	Edit	Delete
15	Milin Kanojiya	milin123@rediffmail.com	8899007762	HR	Edit	Delete
16	Dhrupal	dhrupal123@yahoo.com	1233456780	RJ	Edit	Delete
18	Umesh Patel	umesh.patel@yahoo.co.in	09427607080	MH	Edit	Delete
21	Ashok Patel	ashok787@hotmail.com	9898989898	HR	Edit	Delete

Fig. 2.17 fetches student's record from student table using mysqli_fetch_array.

In above table you can see the link of Edit a record, when we click on this link it will open that record for edit. We will use same registration page for edit a record. On that link I have make an URL to redirect student id and action to edit a record of that particular student.

localhost/std/std_reg.php?Action=Edit&sid=21

In above URL page will redirect to std_reg.php with Action variable and Sid variable also will send values of that variable. For Update a record we follow following steps.

Step 1: we will get the value of Action and sid in std_reg.php page.here we are not making a new page of update record, we use same page which we have used for insert or register a new record.

Step 2: After getting value of action and sid we will use connection file and also will write select query for getting a record of that only student who's sid will match.

Step 3: After getting all the records of that sid we will set all value inside the form with value attribute.

Step 4: User can see the form with value which is already posted. Now user can change the value with new value and submit the form.

Step 5 : form value will submitted on action page which we have already created for insert new record. On that action page will get all the value for update/edit record.

Step 6: in action page we have set one if condition for checking an action. If action is edit or update then and only then it will work on update query and update new record with old record.

We will follow same rules for writing an update query.

- The SQL query must be quoted in PHP.
- String values inside the SQL query must be quoted.
- Numeric values must not be quoted.
- The word NULL must not be quoted.

We will use following query to insert new record.

**UPDATE table_name set column1=value1, column2=value2, column3=value 3....
columnN=value n where fieldname= match value.**

Step 7: After update query will execute we will redirect to the next page where we can check our updated record of that particular student.

std_reg.php

```
<?php
include("conn.php");
session_start();
// action and sid will come from link which is set on student_record.php file.

$action=$_GET["Action"];
$sid=$_GET["sid"];
//if action is Edit then will come inside the if condition otherwise It will go in to else
part.

if($action=="Edit")
{
    $std_qry="select * from student where Student_Id=$sid";
    $res_std=mysqli_query($con,$std_qry);
    $std_row=mysqli_fetch_array($res_std);
    $Fullname=$std_row["Fullname"];
    $Email=$std_row["Email"];
    $Mobile=$std_row["Mobile"];
    $State_Id=$std_row["State_Id"];
}
else
{
    $action="Insert"; //new record insert
}
```

```

}

?>

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<script>
function can()
{
    window.location="index.php";
}
</script>
</head>
<body>
<form action="std_manage.php" method="post">??"&action=std_manage&id=1
<input type="hidden" name="action" value="<?php echo $action;?>">
<input type="hidden" name="sid" value="<?php echo $sid;?>">
<table>
<tr>
<td><label>Fullname</label></td>
<td><input type="text" name="Fullname" value="<?php echo $Fullname;?>" required></td>
</tr>
<tr>
<td><label>Email</label></td>
<td><input type="email" name="Email" value="<?php echo $Email;?>" required></td>
</tr>
<tr>
<td><label>Mobile</label></td>
<td><input type="tel" name="Mobile" value="<?php echo $Mobile;?>" required></td>

```

```

</tr>
<?php if($action=="Insert") {
?
<tr>
<td><label>Password</label></td>
<td><input type="password" name="Password" value="" required></td>
</tr>
<?php
}
?>
<tr>
<td><label>State</label></td>
<td><select name="State">
<option value="">Please Select</option>
<?php
$state="select * from state order by State_Name ASC";
$qry=mysqli_query($con,$state); //mysqli_query("connection variable, Query var");
while($res=mysqli_fetch_array($qry))
{
    //echo $res["State_Name"]."<br>";
?>
<option value=<?php echo $res["State_Id"];?>> <?php
if($res["State_Id"]==$State_Id){echo "selected"; }else{ echo "";}?>><?php echo
$res["State_Name"];?></option>
<?php
}
?>
</select></td>
</tr>
<tr>
<td colspan="3" align="center"><input type="submit" name="submit" value="Submit">
<input type="reset" name="reset" value="reset">

```

```

<button name="cancel" value="cancel" onClick="can0();>Cancel</button>
</td>
</tr>
</table>
</form>
</body>
</html>

```

Fullname | Ashok Patel

Email/Username | ashok787@hotmail.com

Mobile | 9898989898

password |

State | HR

submit | reset | Cancel

Fullname | Ashok Kanubhai Patel

Email/Username | ashok787@hotmail.com

Mobile | 9898989898

password |

State | RJ

submit | reset | Cancel

Fig. 2.18 Edit student record name and state.

std_query.php

```

<?php
    include("conn.php");
    $Action=$_POST["Action"];
    $sid=$_POST["sid"];
    $Fullname=$_POST["Fullname"];
    $Email=$_POST["Email"];
    $Mobile=$_POST["Mobile"];
    $Password=$_POST["Password"];
    $stid=$_POST["state"];

    //Add INSERT NEW RECORD
    if($Action=="Add")
    {
        $ins_qry="insert into student
        (Student_Id,Fullname,Email,Mobile,Password,State_Id)
        values('','$Fullname','$Email','$Mobile','$Password','$stid')";
        $res=mysqli_query($con,$ins_qry);
    }

```

```

if($res)
{
    echo " RECORD INSERTED SUCCESSFULL ";
    //header("location:std_record.php?msg=RECORD INSERTED SUCCESSFULL");
}
else
{
    echo "RECORD NOT INSERTED PLEASE TRY AGAIN";
    //header("location:std_record.php?msg= RECORD NOT INSERTED PLEASE
TRY AGAIN ");
}

//UPDATE RECORD
if($Action=="Edit")
{
    $std_edit="update student set
    Fullname='$Fullname',Email='$Email',Mobile='$Mobile',Password='$Password',St
    ate_Id='$stid' where Student_Id='$sid'";
    $result=mysqli_query($con,$std_edit);
    if($result)
    {
        header("location:std_record.php?msg=RECORD updated");
    }
    else
    {
        header("location:std_record.php?msg=RECORD not update");
    }
}
}

```

After updating a record in Database table we can view a record on our web page for that we will make one more page student_record.php. This page is used to print a record in table format. In this page we will write select query to fetch all the records from table.

student_record.php

```
<?php
include("conn.php");
```

```

session_start();
if(isset($_SESSION["Username"]))
{
    $username=$_SESSION["Username"];
}
$selqry="select * from student";
$res=mysqli_query($con,$selqry);
$msg=$_GET["msg"];
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<table border="1">
<tr>
<th>Student Id</th>
<th>Fullname</th>
<th>Email</th>
<th>Mobile</th>
<th>State</th>
<th>Edit</th>
<th>Delete</th>
</tr>
while($row=mysqli_fetch_array($res))
{
    ?>
    <tr>
        <td><?php echo $row["Student_Id"];?></td>
        <td><?php echo $row["Fullname"];?></td>
        <td><?php echo $row["Email"];?></td>
        <td><?php echo $row["Mobile"];?></td>
}

```

```

<?php
    $st="Select State_Name from state where
State_Id='".$row["State_Id"]."";
    $str=mysqli_query($con,$st);
    $strow=mysqli_fetch_array($str);
?
<td><?php echo $strow["State_Name"];?></td>
<td><a href="std_reg.php?Action=Edit&sid=<?php echo
$row["Student_Id"];?>">Edit</a>
</td>
<td><a href="std_query.php?Action=Delete&sid=<?php echo
$row["Student_Id"];?>">Delete</a></td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

Student Id	Fullname	Email	Mobile	State	Edit	Delete
1	Jigar K Patel	jigar@gmail.com	1111122222	MP	Edit	Delete
8	Jigar Patel	jigar123@gmail.com	1254545747852	MH	Edit	Delete
13	Ghanshyam	ghanshyam@gamil.com	12312312312	MH	Edit	Delete
15	Milin Kanajiya	milin123@rediffmail.com	8899007762	HR	Edit	Delete
16	Dhrupal	dhrupal123@ymali.com	1233456780	RJ	Edit	Delete
18	Umesh Patel	umesh.patel@yahoo.co.in	09427607080	MH	Edit	Delete
21	Ashok Kanubhai Patel	ashok787@hotmail.com	9898989898	RJ	Edit	Delete

Fig. 2.19 After Edit student record name and state fetch record from student table using mysqli_fetch_array.

In above table you can see the link of Delete a record, when we click on this link it will delete that record. On that link I have make an URL to redirect student id and action to delete a record of that particular student.

localhost/std/std_query.php?Action=Delete&sid=21

In above URL page will redirect to std_query.php with Action variable and sid variable also will send values of that variables. For Delete a record we follow following steps.

CC-310 Web Applications Development – II Practical

Step 1: We will first get the value of Action and sid in std_query.php page.

Step 2: After getting a value of action and sid will check the condition if action is Delete then will work on Delete query.

Delete from table_name where fieldname=value;

Step 3: If Delete Query will work successfully then we will check the record is deleted or not then we will redirect on student_record.php.

Std_query.php

```
<?php
```

```
//DELETE RECORD
```

```
echo SAction=S_GET["Action"];
```

```
echo Ssid=S_GET["sid"];
```

```
if(SAction=="Delete")
```

```
{
```

```
Sdel_qry="delete from student where Student_Id='Ssid'";
```

```
Sres_del=mysqli_query($con,Sdel_qry);
```

```
if(Sres_del)
```

```
{
```

```
header("location:std_record.php?msg=RECORD DELETED SUCCESSFULLY");
```

```
}
```

```
else
```

```
{
```

```
header("location:std_record.php?msg=RECORD NOT DELETED");
```

```
}
```

```
}
```

```
?>
```

Student Id	Fullname	Email	Mobile	State	Edit	Delete
1	Jigar K Patel	jigar@gmail.com	1111122222	MP	Edit	Delete
8	Jigar Patel	jigar123@gmail.com	1254545747852	MH	Edit	Delete
13	Ghanshyam	ghanshyam@gamil.com	12312312312	MH	Edit	Delete
15	Milin Kanajiya	milin123@rediffmail.com	8899007762	HR	Edit	Delete
16	Dhrupal	dhrupal123@ymali.com	1233456780	RJ	Edit	Delete
18	Umesh Patel	umesh.patel@yahoo.co.in	09427607080	MH	Edit	Delete

Fig. 2.20 Delete Record from database .

Unit -2 Practical

1. Create a form containing two input fields (Name, Email_ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.

Prg1.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<form action="prg1_submit.php" method="post">
<!--we can set method get also-->
<table>
<tr>
    <td><label for="name">Name</label></td>
    <td><input type="text" name="fullname" ></td>
</tr>
<tr>
    <td><label for="email">Email</label></td>
    <td><input type="email" name="email" ></td>
</tr>
<tr>
    <td colspan="2"><input type="submit" name="Submit" value="Submit" >
    <input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
```

```
</body>
</html>
```

Fig. 2.21 Submit form data on another page using post method.

prg1_submit.php

```
<?php
$fullname=$_POST["fullname"];
$email=$_POST["email"];
//if form method is get then we use$_GET["fullname"]; to receive data from form.
echo "Form Submit following Data<br>";
echo "Fullname=". $fullname . "<br>";
echo "Email=$email<br>";
?>
```

Fig. 2.22 Data posted from form and received by action page.

2. Write a PHP script for that creates a database named "DB-1" in MySQL.

Note: for working with database file first we have to establish a connection between PHP and mysql. For that we use connection function. I have here use one external php file connection.php which I will include in my program.

connection.php

```
<?php
    error_reporting("E_ALL");// this will not show any warning msg.
    $host="localhost";
    $user="root";
    $pass="";
    $con=mysqli_connect($host,$user,$pass);
    if($con)
    {
        echo "Connection Established";
    }
    else
    {
        echo "Connection not Established";
    }
?>
```

Prg2.php

```
<?php
    include ("connection.php");
    $db_create="create database db1";
    $qry=mysqli_query($con,$db_create);
    if($qry)
    {
        echo "Database Created Successfully";
    }
    else
    {
        echo "Database not Created.";
    }
?>
```

?>

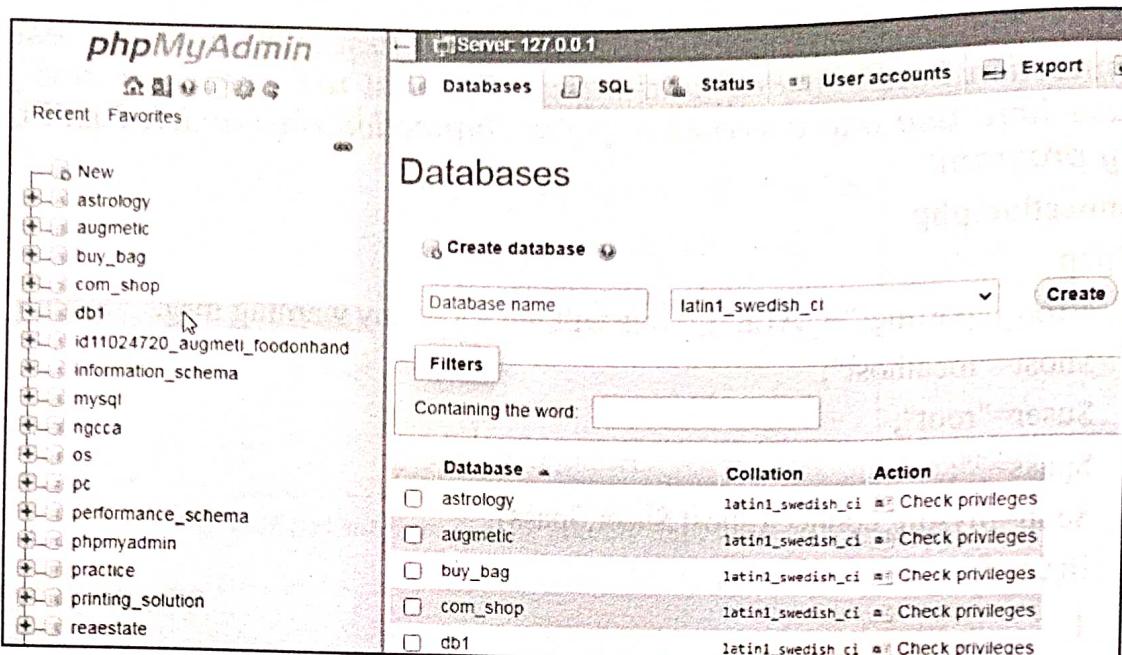


Fig. 2.23 Create Database using create database query.

3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.

Note: for creating a table within database, first we have to establish a connection between PHP and mysql database. For that we use connection function.i have here use one external php file connection.php which I will include in my program.

connection.php

```
<?php
error_reporting("E_ALL");// this will not show any warning msg.
$host="localhost";
$user="root";
$pass="";
$dbname="db1";
$con=mysqli_connect($host,$user,$pass,$dbname);
if($con)
{
    echo "Connection Established<br>";
}
```

```

else
{
    echo "Connection not Established<br>";
}
?>

Prg2.php
<?php
include ("uconnection.php");
$db_create="CREATE TABLE product
(Pro_Id INT(3) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
Pro_name VARCHAR(100) NOT NULL,
Pro_price FLOAT(7,2) NOT NULL,
QOH INT(5)NOT NULL)";
$qry=mysqli_query($con,$db_create);
if($qry)
{
    echo "PRODUCT TABLE CREATED SUCCESSFULLY";
}
else
{
    echo "PRODUCT TABLE NOT CREATED.";
}
?>

```

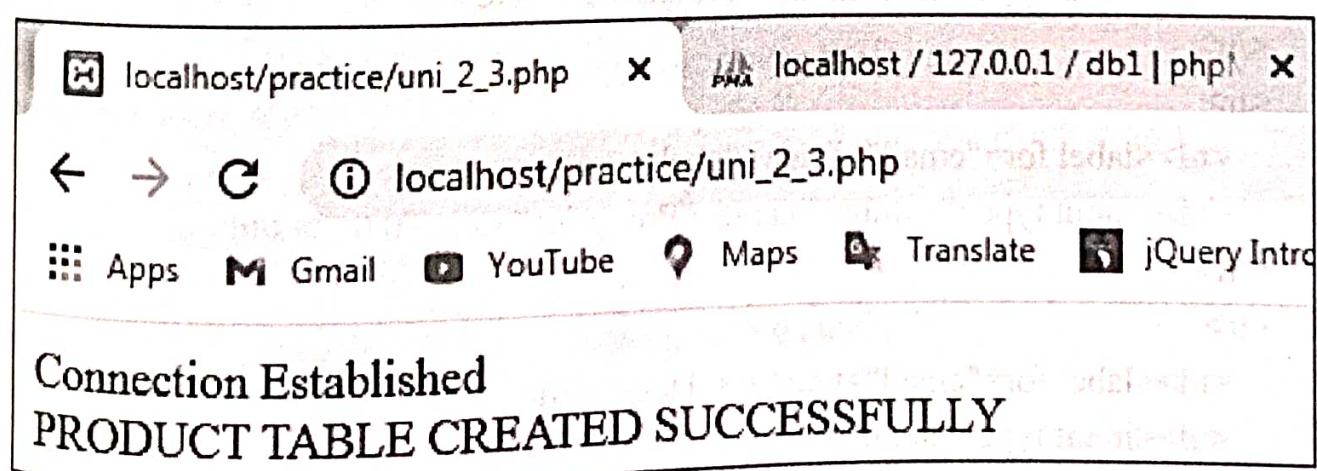


Fig. 2.24 Create Table using create table query.

4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.

Prg3.php

```
<?php
if(isset($_GET["msg"]))
{
    echo $_GET["msg"];
}
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="product_query.php" method="post">
<table>
<caption><b>Add Product</b></caption>
<tr>
<td><label for="name">Productname</label></td>
<td><input type="text" name="Pro_name" ></td>
</tr>
<tr>
<td><label for="email">Price</label></td>
<td><input type="number" name="Pro_price" step="0.01"></td>
</tr>
<tr>
<td><label for="email">Quality</label></td>
<td><input type="number" name="QOH" ></td>
</tr>
<tr>
```

```

<td colspan="2"><input type="submit" name="Submit" value="Submit" >
<input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>

```

Add Product	
Productname	USB Head Phone
Price	1260.50
Quality	70
<input type="button" value="Submit"/> <input type="button" value="reset"/>	

Fig. 2.25 Insert Product in product table using add product form .

➤ **product_query.php**

```

<?php
    include("uni_2_connection.php");
    $productname=$_POST["Pro_name"];
    $price=$_POST["Pro_price"];
    $qoh=$_POST["QOH"];
    if(!empty($productname) && !empty($price) && !empty($qoh))
    {
        $ins_qry="insert into product (Pro_name,Pro_price,QOH) values
        ('$productname','$price','$qoh')";
        $res=mysqli_query($con,$ins_qry);
        if($res)
        {
            header("location:product_record.php");
        }
    }
    else
    {

```

```
header("location:uni_2_4.php?msg=Please Try Again");
```

```
}
```

```
?>
```

➤ **Product_record.php**

```
<?php
```

```
include("uni_2_connection.php");
```

```
$sel_qry="select * from product";
```

```
$res=mysqli_query($con,$sel_qry);
```

```
?>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
<h3><a href="uni_2_4.php">+Add New Record</a></h3>
```

```
<table border="1">
```

```
    <tr bgcolor="#6699FF">
```

```
        <th>Product Id</th>
```

```
        <th>Product Name</th>
```

```
        <th>Product Price</th>
```

```
        <th>Product Quantity</th>
```

```
        <th>Edit</th>
```

```
    </tr>
```

```
<?php
```

```
    while($row=mysqli_fetch_array($res))
```

```
{
```

```
?>
```

```
    <tr bgcolor="#FFCC33">
```

```
        <td><?php echo $row["Pro_Id"];?></td>
```

```
        <td><?php echo $row["Pro_name"];?></td>
```

```
        <td><?php echo $row["Pro_price"];?></td>
```

```
        <td><?php echo $row["QOH"];?></td>
```

```

<td><?php echo "<a href=Prg6.php?pid=$row[Pro_Id] style=text-decoration:none>Edit</a>"?></td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

Product Id	Product Name	Product Price	Product Quantity
1	Key Board	325.85	100
2	Mouse	245.75	200
3	Pen drive 8GB	420.75	150
4	USB Head Phone	1260.50	70

Fig. 2.26 Product Added in Product table and fetch record from product table .

5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.

Prg5.php

```

<?php
include("uni_2_connection.php");
if(!empty($_POST["search_product"]))
{
    $search=$_POST["search_product"];
    $selqry="select * from product where Pro_Id=$search";
    $res=mysqli_query($con,$selqry);
}
else

```

```

{
$sel_qry="select * from product";
$res=mysqli_query($con,$sel_qry);
}
?>
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<div style="float:left; width:150px;"><h3><a href="uni_2_4.php">+Add New Record</a></h3></div>
<div style="float:right; width:150px; margin-right:100px;">
<form action=<?php $_SERVER["PHP_SELF"];"> method="post">
<table>
<tr>
<td>
<select name="search_product">
<option value="">Please Select</option>
<?php
$sel_qry_search="select * from product";
$res_search=mysqli_query($con,$sel_qry_search);
while($row_search=mysqli_fetch_array($res_search))
{
?>
<option value=<?php echo $row_search["Pro_Id"];?>>
<?php echo $row_search["Pro_name"];?>
</option>
<?php
}
?>

```

```

        </select>
    </td>
    <td>
        <input type="submit" name="submit" value="Search"></td>
    </tr>
</table>
</form>
</div>
<div style="margin-top:50px; margin-left:200px; float:left;">
<table border="1">
<caption><b>Product Detail</b></caption>
    <tr bgcolor="#6699FF">
        <th>Product Id</th>
        <th>Product Name</th>
        <th>Product Price</th>
        <th>Product Quantity</th>
    </tr>
    <?php
        while($row=mysqli_fetch_array($res))
        {
            ?>
    <tr bgcolor="#FFCC33">
        <td><?php echo $row["Pro_Id"];?></td>
        <td><?php echo $row["Pro_name"];?></td>
        <td><?php echo $row["Pro_price"];?></td>
        <td><?php echo $row["QOH"];?></td>
    </tr>
    <?php
        }
    ?>
</table>
</div>
</body>
</html>

```

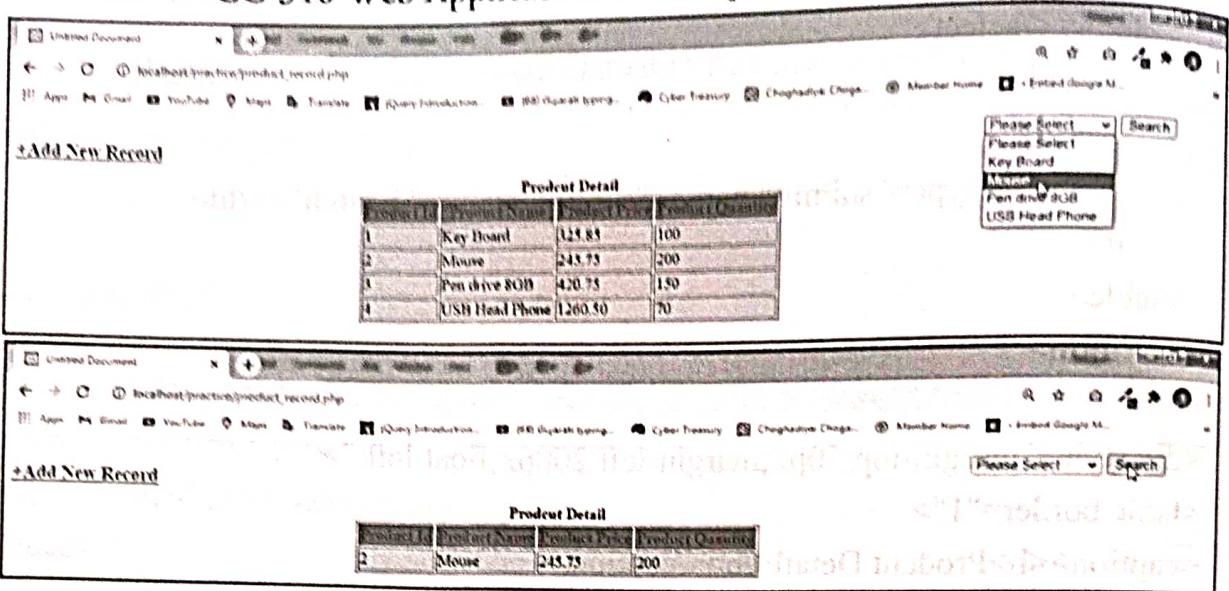


Fig. 2.27 Search Record from product table giving product id.

- 6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.**

> **Prg6.php**

```
<?php
include("uni_2_connection.php");
$pro_id=$_GET["pid"];//this id is come from product_record.php page.
```

```
$sel_product="select * from product where Pro_Id=$pro_id";
```

```
$res_product=mysqli_query($con,$sel_product);
```

```
$row_product=mysqli_fetch_array($res_product);
```

```
$pid=$row_product["Pro_Id"];
```

```
$qty=$row_product["QOH"];
```

```
if(isset($_GET["msg"]))
```

```
{
```

```
echo $_GET["msg"];
```

```
}
```

```
?>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```

<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<form action="product_update.php" method="post">
    <table>
        <caption>Product Update</caption>
        <tr>
            <td><label for="product">Product Name</label></td>
            <td>
                <select name="Pro_Id">
                    <option value="">Please Select</option>
                    <?php
                        $sel_product="select * from product order by Pro_name ASC";
                        $qry=mysqli_query($con,$sel_product);
                        while($row_search=mysqli_fetch_array($qry))
                        {
                            ?>
                    <option value="<?php echo $row_search["Pro_Id"];?>" <?php
                        if($row_search["Pro_Id"]==$pid){echo "selected"; } else{ echo "";} ?>>
                        <?php echo $row_search["Pro_name"];?>
                    </option>
                    <?php
                }
                ?>
                </select>
            </td>
        </tr>
        <tr>
            <td><label for="quantity">Quantity</label></td>
            <td><input type="number" name="QOH" value="<?php echo $qty;?>"></td>
        </tr>
    </table>
</form>

```

```

<tr>
    <td colspan="2"><input type="submit" name="Submit" value="Submit">
    <input type="reset" name="reset" value="reset"></td>
</tr>
</table>
</form>
</body>
</html>

```

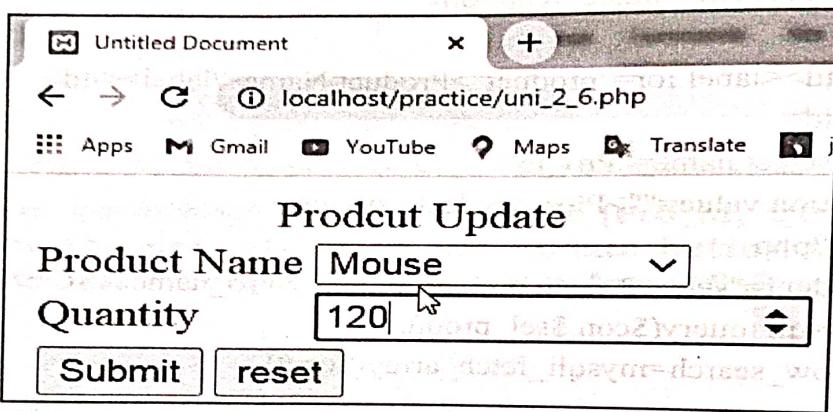


Fig. 2.28 Edit Product using product update form.

➤ **Product_update.php**

```

<?php
    include("uni_2_connection.php");
    $Pro_Id=$_POST["Pro_Id"];
    echo $qoh=$_POST["QOH"];
    if(!empty($qoh))
    {
        $update_qry="update product set QOH=$qoh where Pro_Id=$Pro_Id";
        $update_res=mysqli_query($con,$update_qry);
        if($update_res)
        {
            header("location:product_record.php");
        }
    }
}

```

```

else
{
    header("location:prg6.php?msg=Please Try Again");
}
?>

```

Product Detail				
Product Id	Product Name	Product Price	Product Quantity	Edit
1	Key Board	325.85	100	Edit
2	Mouse	245.75	120	Edit
3	Pen drive 8GB	420.75	150	Edit
4	USB Head Phone	1260.50	500	Edit

Fig. 2.29 Fetch Edited Record from database

- 7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.**

Product_record.php

```

<?php
include("uni_2_connection.php");
if(!empty($_POST["search_product"]))
{
    $search=$_POST["search_product"];
    $selqry="select * from product where Pro_Id=$search";
    $res=mysqli_query($con,$selqry);
}
else
{
    $selqry="select * from product";
    $res=mysqli_query($con,$selqry);
}
?>

<!doctype html>
<html>
<head>

```

```

<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<div style="float:left; width:150px;"><h3><a href="uni_2_4.php">+Add New
Record</a></h3></div>
<div style="float:right; width:150px; margin-right:100px;">
<form action="php $_SERVER['PHP_SELF']; ?" method="post">
<table>
<tr>
<td>
<select name="search_product">
<option value="">Please Select</option>
<?php
$sel_qry_search="select * from product";
$res_search=mysqli_query($con,$sel_qry_search);
while($row_search=mysqli_fetch_array($res_search))
{
    <option value="<?php echo
$row_search['Pro_Id']; ?>">
        <?php echo $row_search['Pro_name']; ?>
    </option>
    <?php
}
?>
</select>
</td>
<td>
<input type="submit" name="submit" value="Search"></td>
</tr>

```

```

</table>
</form>
</div>
<div style="margin-top:50px; margin-left:200px; float:left;">

<table border="1">
<caption><b>Product Detail</b></caption>
<tr bgcolor="#6699FF">
<th>Product Id</th>
<th>Product Name</th>
<th>Product Price</th>
<th>Product Quantity</th>
<th>Edit</th>
<th>Delete</th>
</tr>
<?php
while($row=mysqli_fetch_array($res))
{
?>
<tr bgcolor="#FFCC33">
<td><?php echo $row["Pro_Id"];?></td>
<td><?php echo $row["Pro_name"];?></td>
<td><?php echo $row["Pro_price"];?></td>
<td><?php echo $row["QOH"];?></td>
<td><?php echo "<a href=uni_2_6.php?pid=$row[Pro_Id] style=text-decoration:none>Edit</a>"?></td>
<td><?php echo "<a href=uni_2_7.php?pid=$row[Pro_Id] style=text-decoration:none>Delete</a>"?></td>
</tr>
<?php
}
?>
</table>
</div>

```

</body>

</html>

Product Detail					
Product ID	Product Name	Product Price	Product Quantity	Edit	Delete
1	Key Board	325.85	100	Edit	Delete
2	Mouse	245.75	120	Edit	Delete
3	Pen drive 8GB	420.75	150	Edit	Delete

Fig. 2.30 Delete Product

➤ Prg7.php

<?php

include("uni_2_connection.php");

echo \$Pro_Id=\$_GET["pid"];// this id is come from product_record.php page.

if(!empty(\$Pro_Id))

{

echo \$delete_qry="delete from product where Pro_Id=\$Pro_Id";

\$delete_res=mysqli_query(\$con,\$delete_qry);

if(\$delete_res)

{

header("location:product_record.php");

}

}

else

{

header("location:uni_2_7.php?msg=Please Try Again");

}

?>

Product Detail					
Product ID	Product Name	Product Price	Product Quantity	Edit	Delete
1	Key Board	325.85	100	Edit	Delete
2	Mouse	245.75	120	Edit	Delete

Fig. 2.31 Deleted Record from database.