UNIT – V

ESTABLISHING A DATABASE CONNECTION

&

WORKING WITH DATABASE

5.1 Overview of Database

- Database is a collection of organized information, so that it can be easily accessed, managed and updated.

- In Relational Database Management System, every database is consist of one or more tables inside it.

- Thus you can say database is collection of one or more tables which are related with one another.

- A table in database used to store records. The records in the table are organized in the form or rows. Each row in the table represents particular record.

- A record that is contained in the table is collection of one or more columns. Each column in the record represents field of specific type in the database table.

5.2 Introduction to MYSQL

- MySQL is open source relation database management system.
- It is easy to use database management system for small and big business applications.
- MySQL is developed by a Swedish company named MySQL AB. It was first released in January 1998.

5.2 Introduction to MYSQL

- **Advantages:**

- It is an open source database system, so you do not need to purchase license version of it.

- It is based on Structured Query Language (SQL).

- It can run on almost all platforms such as Linux, Unix and Windows.

- It can work with many programming languages such as PHP, C, C++, JAVA, PERL etc.

- It supports large database.

5.2 Introduction to MYSQL

- **Data types (Field Types) in MySQL:**

MySQL Field Type	Description	Example
`char(length)`	Any character can be in this field, but the field will have a fixed length.	Customer's State field always has two characters.
`varchar(length)`	Any character can be in this field, and the data can vary in length from 0 to 255 characters. Maximum length of field is denoted in parentheses.	Customer's Address field has letters and numbers and varies in length.
`int(length)`	Numeric field that stores integers that can range from -2147483648 to +2147483647, but can be limited with the `length` parameter. The `length` parameter limits the number of digits that can be shown, not the value. Mathematical functions can be performed on data in this field.	Quantity of a product on hand.

5.2 Introduction to MYSQL

- **Data types (Field Types) in MySQL:**

`int(length) unsigned`	Numeric field that stores positive integers (and zero) up to 4294967295. The `length` parameter limits the number of digits that can be displayed. Mathematical functions can be performed on data in this field.	Customer ID (if entirely numerical).
`text`	Any character can be in this field, and the maximum size of the data is 65536 characters.	Comments field that allows longer text to be stored, without limiting field to 255 characters.
`decimal(length,dec)`	Numeric field that can store decimals. The `length` parameter limits the number of digits that can be displayed, and the `dec` parameter limits the number of decimal places that can be stored. For example, a price field that would store prices up to 999.99 would be defined as `decimal(5,2)`.	Prices.
`enum("option1", "option2",...)`	Allows only certain values to be stored in this field, such as "true" and "false," or a list of states. 65535 different options are allowed.	Gender field for your users will have a value either "male" or "female."
`date`	Stores a date as yyyy-mm-dd.	Date of order, a birthday, or the date a user joined as a registered user.
`time`	Stores time as hh:mm:ss.	Time a news article was added to the Web site.
`datetime`	Multipurpose field that stores date and time as yyyy-mm-dd hh:mm:ss.	Last date and time a user visited your Web page.

5.2 Introduction to MYSQL

- **Integration of PHP with MySQL:**
- It is possible to execute various commands of MySQL from PHP application.
- PHP provides various built in functions that allow you to use MySQL commands from PHP page. Thus you can integrate PHP with MySQL.
- Following are the frequently used PHP functions that allows the execution of MySQL commands:
  - **mysql_connect** (“*servername*", "*user*", "*pass*"): Connects to the MySQL server.
  - **mysql_select_db**("*database name*"): Equivalent to the MySQL command USE; makes the selected database the active one.
  - **mysql_query**("*query*“,”connection”): Used to send any type of MySQL command to the server.

5.2 Introduction to MYSQL

- **Integration of PHP with MySQL:**

  - **mysql_num_rows**("*results variable from query*"): Used to return number of records available in the results of a database query.
  - **mysql_fetch_rows**("*results variable from query*"): Used to return a row of the entire results of a database query.
  - **mysql_fetch_array**("*results variable from query*"): Used to return several rows of the entire results of a database query.
  - **mysql_close**("connection"): close the connection with the MySQL server.
  - **mysql_error**(): Shows the error message that has been returned directly from the MySQL server.

5.3 Creating Database using PHPMyAdmin & Console

Covered in Laboratory

5.4 Connecting with Database , Creating & executing queries using *mysql_query()*

- **5.4.1 Connecting with Database**

- Before you start working with MySQL database server, you first have to establish connection with MySQL server.

- PHP allows you to connect with MySQL server using mysql_connect() function.

- **Syntax:** mysql_connect (“*servername*", "*username*", "*password*")

5.4 Connecting with Database , Creating & executing queries using *mysql_query()*

- **5.4.1 Connecting with Database**
  - **Servername:** Indicates name of the MySQL server with which you want to establish connection.
  - **username:** Indicates name of the user using which you can logs on to MySQL server.
  - **password:** Indicates password of the user using which you can logs on to MySQL server.

- This function returns a boolean value TRUE or FALSE.
- If connection with MySQL server is established successfully, then it returns TRUE otherwise returns FALSE.

5.4 Connecting with Database , Creating & executing queries using *mysql_query()*

- **5.4.1 Connecting with Database**

- **Example:**

```
$con = mysql_connect("localhost", "root");

if(! con)
{
        die(mysql_error());
}
```

5.4 Connecting with Database , Creating & executing queries using *mysql_query()*

- **5.4.2 Creating & executing queries using *mysql_query()***

- mysql_query() function allows you to specify and execute SQL (MySQL commands) on MySQL server.

- **Syntax:** mysql_query ("*query*“,”*connection_name*”)

  - **query:** Indicates the SQL to be executed.

  - **Connection_name:** Indicates the name of the connection that is established using mysql_connect() function.

5.4 Connecting with Database , Creating & executing queries using *mysql_query()*

- **5.4.2 Creating & executing queries using *mysql_query()***
- **Example:**

```
$con = mysql_connect("localhost", "root");
$db = mysql_selectdb("KDP");

$res = mysql_query("create table Student(Enrollment integer, Name
        varchar(20), Semester integer, Department varchar(20))" , $con);

if(res)
{
    echo "Table Created Successfully.";
}
else
{
    echo "Error while executing query.";

}
```

5.5 Creating table, inserting data in to table through HTML Forms

- **5.5.1 creating table**

```
<?php

    $con = mysql_connect("localhost", "root");
    $db = mysql_selectdb("KDP");

        $res = mysql_query("create table Student(Enrollment integer, Name
                varchar(20), Semester integer, Department varchar(20))" , $con);

        if(res)
        {
                echo "Table Created Successfully.";
        }
        else
        {
        echo "Error while executing query.";
        }
?>
```

5.5 Creating table, inserting data in to table through HTML Forms

- **5.5.2 inserting data in to table through HTML Forms**
- **From. php:**

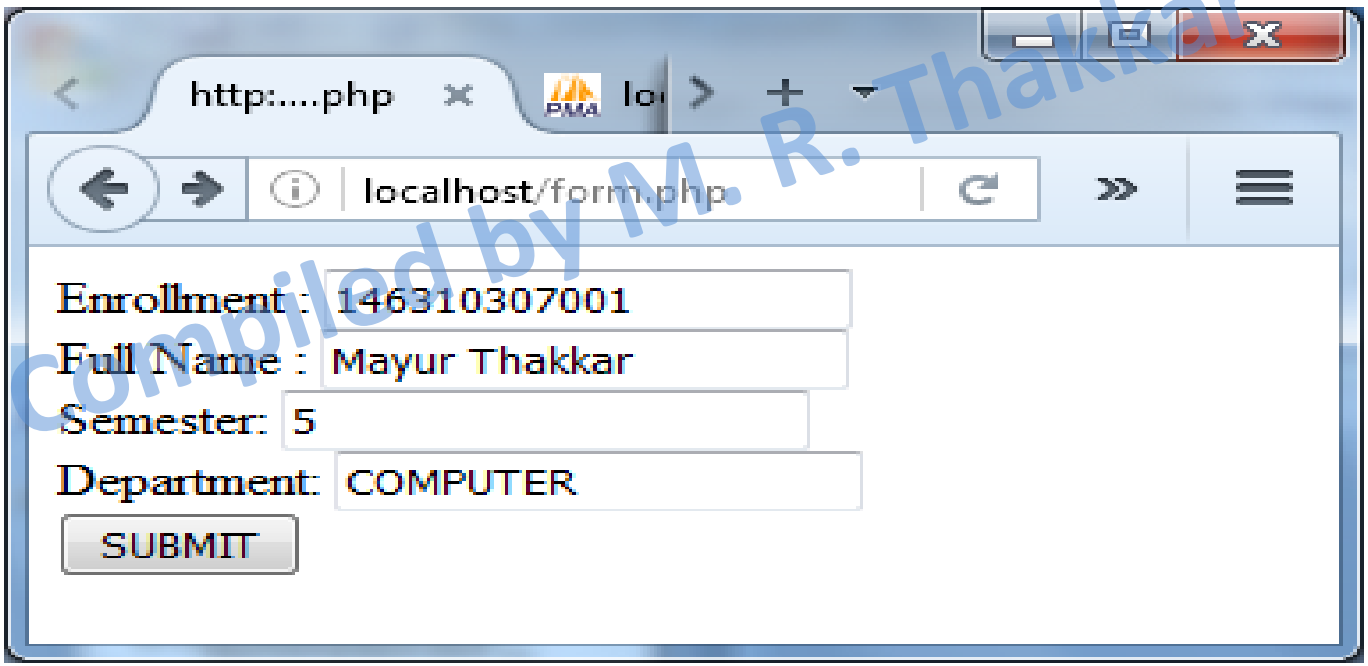
```
<form action = "addrecord.php" method="POST">

    Enrollment : <input type = "text" name = "enroll" >
    <BR/>
    Full Name : <input type = "text" name = "fname" >
    <BR/>
    Semester: <input type = "text" name = "sem" >
    <BR/>
    Department: <input type = "text" name = "dep" >
    <BR/>
    <input type = "Submit" name = "submit" value = "SUBMIT" >

</form>
```

5.5 Creating table, inserting data in to table through HTML Forms

- **5.5.2 inserting data in to table through HTML Forms**
- **From. php:**



5.5 Creating table, inserting data in to table through HTML Forms

- **5.5.2 inserting data in to table through HTML Forms**
- **Addrecord.php:**

```
<?php

    $con = mysql_connect("localhost", "root");
    $db = mysql_selectdb("KDP");

    $enroll = $_POST["enroll"];
    $name = $_POST["fname"];
    $sem = $_POST["sem"];
    $dep = $_POST["dep"];

    $res = mysql_query("Insert into  Student values ($enroll , $name , $sem , $dep)",$con);

    mysql_close($con);

    echo "Record inserted Successfully.“;
?>
```

5.6 Retrieving data from Table, using *mysql_numrows(), Printing the output using PHP and HTML*

- **displayrecord.php:**

```
<TABLE   border = “1”>
<tr>
        <th> Enrollment </th> <th> Name </th> <th> Semester </th> <th> Department </th>
 </tr>

    <?php
        $con = mysql_connect("localhost", "root");
        $db = mysql_selectdb("KDP");

        $res = mysql_query(“Select *  from  Student", $con);
```

5.6 Retrieving data from Table, using *mysql_numrows(), Printing the output using PHP and HTML*

- **displayrecord.php:**

```
if(mysql_num_rows($res)>0)
 {
        while ( $row = mysql_fetch_array($res))
        {
                echo  "<tr>   <td> $row[Enrollment] </td> <td>  $row[Name]
               </td> <td> $row[Semester] </td> <td> $row[Department] </td>
                </tr>";
        }
echo "</TABLE>";

mysql_close($con);

?>
```

****