

EVENTS AND REPORT USING MYSQL & PHP

We are going to have a look on creating events (schedules) in database by doing one practical example and generating the report of database record in the pdf form using **fpdf.php**

1. EVENTS

Lets discuss the events of the database first and find the answers to following questions:

- How Events are created?
- Where can we find events in database?
- How events can be created by sql query?
- What are the types of events?
- How many query can we write in events to perform an action on schedule time?

We will see the functionality of events by considering the following scenario:

Let suppose you want the record of the user registration table to be backup into another table (let's say userbackup table) to reduce the load on the server. This can be made possible by creating an event which automatically transfers the user registration table record into the userbackup table. For this you have to create

- **User/Registration Table** - to register users
- **UserBackup Table** - to store backup data
- **An event** - to schedule a time for transferring or backing up the user registration table data into the userbackup table.

By using **phpmyadmin** database we had created the following table

USER TABLE

Create the **user** table by the following sql query:

```
CREATE TABLE `user` (  
  `id` int(28) NOT NULL AUTO_INCREMENT,  
  `FName` varchar(128) NOT NULL,  
  `MName` varchar(128) NOT NULL,  
  `LName` varchar(128) DEFAULT NULL,  
  `Email` varchar(128) NOT NULL,  
  `Mobile` varchar(128) NOT NULL,  
  `Password` varchar(128) NOT NULL,  
  `DOB` date NOT NULL,  
  `Gender` varchar(128) NOT NULL,  
  `currenttime` datetime NOT NULL DEFAULT current_timestamp(),  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=30 DEFAULT CHARSET=utf8mb4
```

Insert some dummy data into the database or create a UI/Frontend for the registration page for inserting data into the database

alt text

and the resultant table along with the user record will look like this :

alt text

Note : the currenttime field will contain the time at which the user was registered using NOW() function in php;

USERBACKUP TABLE

Create the **user** table by the following sql query:

```
CREATE TABLE `userbackup` (  
  `b_id` int(128) NOT NULL,  
  `fname` varchar(128) NOT NULL,  
  `mname` varchar(128) NOT NULL,  
  `lname` varchar(128) DEFAULT NULL,  
  `email` varchar(128) NOT NULL,  
  `mobile` varchar(128) NOT NULL,  
  `pswd` varchar(128) NOT NULL,  
  `dob` date NOT NULL,  
  `gender` varchar(128) NOT NULL,  
  `currenttime` datetime(6) NOT NULL DEFAULT current_timestamp(6),  
  PRIMARY KEY (`b_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

After running the above sql query a table with null data record will be created as shown below:

alt text

Now our task is to transfer the values from user table into the userbackup table automatically
For this create an event

CREATE EVENT THROUGH SQL QUERY

*On the navbar above in phpmyadmin click on the sql option and write the following query there:

```
CREATE EVENT `userbackup`  
ON SCHEDULE EVERY 2 MINUTE  
STARTS '2021-02-23 20:30:11'  
ON COMPLETION PRESERVE ENABLE  
COMMENT 'Archive table'  
DO  
BEGIN  
INSERT INTO userbackup (b_id, fname, lname, email, mobile, pswd, dob, gender, currenttime)  
SELECT id, FName, LName, Email, Mobile, Password, DOB, Gender, NOW()  
FROM user  
WHERE NOT EXISTS (SELECT b_id, fname, lname, email, mobile, pswd, dob, gender, currenttime FROM userbackup WHERE user.id=userbackup.b_id);  
END
```

Lets understand the above event query step by step:

- **CREATE EVENT userbackup** - This will create an event with name 'userbackup'
- **ON SCHEDULE EVERY 2 MINUTE** - This means that the event will be active or start after every 2 minutes.
- **STARTS '2021-02-23 20:30:11'** - This sets the start timing of the event or in simply means from when you want to start this event.
- **COMMENT 'Archive table'** - This is optional statement to make you remember that for which purpose you had created the event.
- **DO** - This means that which action you want to perform and write the action afterwards DO keyword
- **BEGIN** - When you want to execute multiple sql queries after DO keyword than you should write BEGIN keyword
- **INSERT INTO userbackup(b_id,fname,..)** - It will insert the user table record into the userbackup table
- **SELECT id,FName,.. FROM user** - It will select the data from the user table which we want to put in userbackup table.
- **WHERE NOT EXISTS (SELECT b_id,fname,.. FROM userbackup WHERE user.id=userbackup.b_id)** - It will check whether the user table data already exist in the userbackup table or not .If it is then it will not copy that data returning false to the **WHERE NOT EXISTS** keyword hence no data will be selected and inserted.
- **END** - It shows end to the BEGIN keyword.

Note: The query here is a single sql query but it is a good practice to write the queries inside BEGIN and END keyword.

Note: The **STARTS '2021-02-23 20:30:11'** statement indicate the start of schedule but this query doesn't have the **ENDS** statemnet that means it will be a recursive event which will be automatically generate again and again.

While the events with both STARTS and ENDS are called **ONE TIME** events

After running the above query the userbackup table will fill with all the record of the user table as shown:

alt text

CONCLUSION OF EVENTS

Thats how the events are created in the database to perform an action related to database automatically.

2. REPORT

Let suppose you want the record of the user registration table to be available in the pdf or report form so that you can print it out easily.

This can be made possible by using **FPDF**, which will generate the database record of users in the pdf form.

WHAT IS FPDF?

FPDF is a PHP class which allows to generate PDF files with pure PHP, that is to say without using the PDFlib library. F from FPDF stands for Free: you may use it for any kind of usage and modify it to suit your needs.

GETTING STARTED:

Consider the user table below:

alt text

We want this above record to be generated as a pdf file. To do this perform the following steps:

1. **Create a Frontend:** In this case, we have a facebook login page.

alt text

alt text

In the above UI, we have an option for generating the record of all the user or specific user on the basis of Full name or email address or both.

2. **Code:** The following code will demonstrate that how the fpdf will generate the report.

Install fpdf zip file and extract it into the new folder called **lib** and import the class of pdf.php in the pdf.php file.

```
<?php
require('lib/fpdf.php');
include 'process.php';
```

Make a new connection to the database to get the user data or run sql queries.

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "signboxtask3";
$conn = new mysqli($servername, $username, $password, $dbname);
```

Make a new object of FPDF class with

- First parameter 'P' -Portrait mode.
- Second parameter 'mm' -page size unit in millimeter.
- First parameter 'A4' -Page style sets to A4.

```
$pdf = new FPDF('P', 'mm', 'A4');
```

Call a new page from AddPage() class method

```
$pdf->AddPage();
```

Set a logo image by calling Image() class method with parameters(image name, left, top, size).

```
$pdf->Image('fb.png',83,9,40);
```

Set a Line break to 12

```
$pdf->Ln(12);
```

Set the font, weight, and size of the **USER REPORT** cell.

```
$pdf->SetFont('Arial','B',15);
```

Customize cell setting by parameters(left, top, text, border(set to 0 if no border), newline)

```
$pdf->Cell(190,7,'USER REPORT ',0,1,'C');
$pdf->SetFont('Arial','',10);
$pdf->Cell(190,5,'By: Facebook',0,1,'C');
$pdf->Ln(10);
$pdf->SetFont('Arial','B',10);
$pdf->Cell(14,0,'Dated : ',0,1,'C');
```

Show the current date on report using date() function.

```
$pdf->Cell(52,0,date("Y.m.d"),0,1,'C');
$pdf->Cell(325,0,'Developer : Ayesha Noor Khan',0,1,'C');
$pdf->Cell(10,7,' ',0,1);
$pdf->SetFont('Arial','B',10);
```

From here we are starting to customize the report table.

```
$pdf->SetFillColor(0,128,255);
$pdf->SetTextColor(255);
$pdf->SetDrawColor(255);
$pdf->SetLineWidth(.3);
$pdf->SetFont('', 'B');
$pdf->Ln(7);
```

- **SetFillColor** - background color of cell. It takes rgb value(currently set to lightblue).
- **SetTextColor** - text color of cell. It takes rgb value(currently set to black).
- **SetDrawColor** - table lines color . It takes rgb value(currently set to black).
- **SetLineWidth** - table line width.

Give the text to table head with the last parameter of **TRUE** to activate the background color of **SetFillColor**

```
$pdf->Cell(20,6,'USER ID',1,0,'C',TRUE);
$pdf->Cell(45,6,'NAME',1,0,'C',TRUE);
$pdf->Cell(57,6,'EMAIL',1,0,'C',TRUE);
$pdf->Cell(26,6,'CONTACT',1,0,'C',TRUE);
$pdf->Cell(23,6,'BIRTHDATE',1,0,'C',TRUE);
$pdf->Cell(18,6,'GENDER',1,1,'C',TRUE);
$pdf->SetTextColor(0);
$pdf->SetDrawColor(192,192,192);
$pdf->Ln(3);
$pdf->SetFont('Arial','',10);
```

This will simply check for the server POST request and check whether single user report is required or to get all user's report.

NOTE : If all the users are required it will use the query **SELECT * FROM user** and pass the database value into the cell by the statement **\$pdf->Cell(45,6,\$row['FName'],1,0);**

```

if($_SERVER["REQUEST_METHOD"] == "POST") {
    if(isset($_POST['generateall']))
    {

$detail = mysqli_query($conn, "SELECT * FROM user");
while ($row = mysqli_fetch_array($detail)){
    $pdf->Cell(20,6,$row['id'],1,0);
    $pdf->Cell(45,6,$row['FName'],1,0);
    $pdf->Cell(57,6,$row['Email'],1,0);
    $pdf->Cell(26,6,$row['Mobile'],1,0);
    $pdf->Cell(23,6,$row['DOB'],1,0);
    $pdf->Cell(18,6,$row['Gender'],1,1);
}
}
}

```

If a specific user report is required it will search it by user name or email or both.

```

else if(isset($_POST['generate']))
{

$ueinfo=$_POST['ueinfo'];
$uminfo=$_POST['uminfo'];
if($_POST['uminfo'] != null && $_POST['ueinfo'] == null)
{
$detail2 = mysqli_query($conn, "SELECT * FROM user where FName ='$uminfo' ");
}
else if($_POST['uminfo'] == null && $_POST['ueinfo'] != null)
{
$detail2 = mysqli_query($conn, "SELECT * FROM user where Email ='$ueinfo' ");
}
else if($_POST['uminfo'] != null && $_POST['ueinfo'] != null)
{
$detail2 = mysqli_query($conn, "SELECT * FROM user where Email ='$ueinfo' AND FName ='$uminfo' ");
}
while ($row = mysqli_fetch_array($detail2)){
    $pdf->Cell(20,6,$row['id'],1,0);
    $pdf->Cell(45,6,$row['FName'],1,0);
    $pdf->Cell(57,6,$row['Email'],1,0);
    $pdf->Cell(26,6,$row['Mobile'],1,0);
    $pdf->Cell(23,6,$row['DOB'],1,0);
    $pdf->Cell(18,6,$row['Gender'],1,1);
}
}
}

```

To set the page border, set the **SetDrawColor** to be 0 for black line color and use **Rect(5, 5, 200, 287, 'D');** for page border width and height. The 'D' is used for **A4 page size**.

```

$pdf->SetDrawColor(0);
$pdf->Rect(5, 5, 200, 287, 'D');

```

To clean the already exist data in buffer use **ob_end_clean()** method for no garbage value in the report.

```

ob_end_clean();

```

To show the output of all the above code use **Output()** method of FPDF class.

```

$pdf->Output();
?>

```

CONCLUSION OF EVENTS:

The above code will generate a pdf file report of the users that are registered in the user table.

- [For full report generator code click here \(https://github.com/ayeshank/Signbox/blob/master/Task3/pdf.php\)](https://github.com/ayeshank/Signbox/blob/master/Task3/pdf.php)
- [For full webpage code click here \(https://github.com/ayeshank/Signbox/tree/master/Task3\)](https://github.com/ayeshank/Signbox/tree/master/Task3)