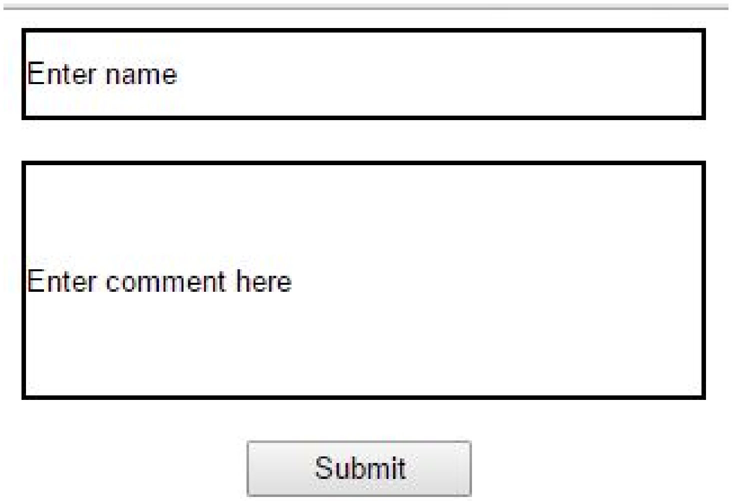
Create a comments section on website using Adobe Dreamweaver

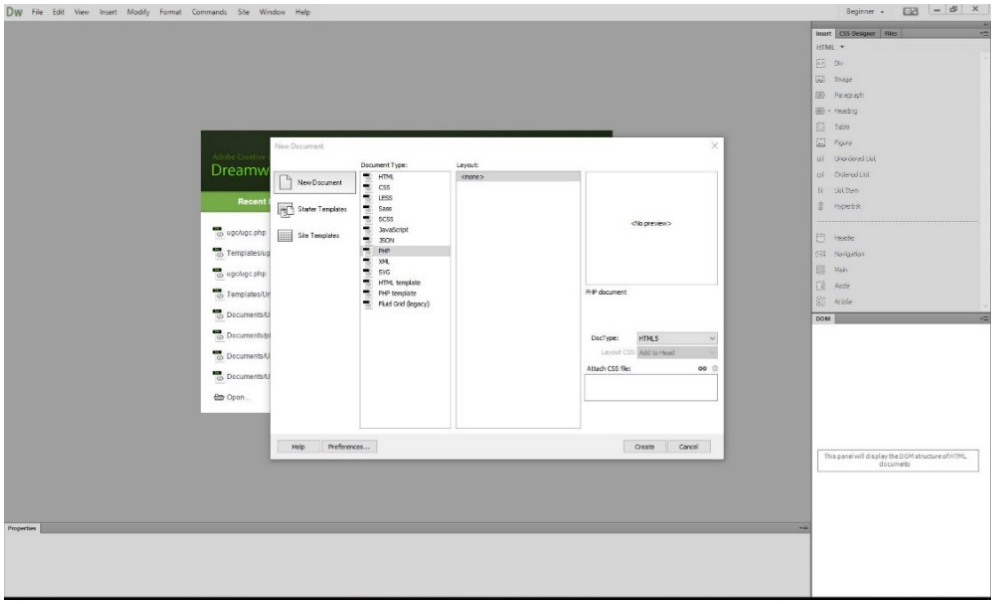
This tutorial explains how to create a web application for comments by users using Adobe Dreamweaver. Before we begin make sure you have set up PHP development environment and have MySQL installed on your computer. You also need to install and configure the Apache XAMPP Web server. Read [Setting up a PHP development environment for Dreamweaver]() for more information on the subject.

Shown below is the image of the app.



Follow the following steps to get started with the task:

1. Launch Adobe Dreamweaver and create a new file by choosing File>New.
2. Select PHP as shown in the image below. This will enable you to write code snippets in PHP in your webpage file.



3. Now we begin writing code

a. First, we begin with client side development. We need two textboxes and one button in our UI.

```
<form method="post" action="ugc.php">

   <input type="text" name="tex1" style="border-style:double; width:300px;
height:35px; background-color:transparent" value="Enter name">

   <br><br>

   <input type="text" name="tex2" style="border-style:double; width:300px;
height:100px;  background-color:transparent" value="Enter comment here">

   <br><br>

   <input type="submit"  style="width:100px; height:25px; margin-left:100px">

   <p id="p"></p>

</form>
```

b. Once you have created the UI, you can proceed to the server side development. We also need to create a database and table in MySQL to store the data provided by the user.
Open MySQL Command Line Client to write MySQL queries.
We create a new database using the following query,

```
create database db;
```

We created a new database by the name ‘db’.

Now we select the database,

```
select db;
```

In this database, we create a new table ‘t2’,

```
CREATE TABLE t2

(

name varchar(100),

comment varchar(300)

);
```

Once you have created a new database and table, we can proceed to writing code in PHP to connect our web app with the database and to perform basic functions like appending to the database and retrieving data from it.

The PHP code snippet is shown below,

```
<?php

  $dbhst='localhost'; // host

  $usnm='root'; // username

  $pass='password'; // password

  $dbname='db'; // database name

  // Create connection

  $conn = new mysqli($dbhst, $usnm, $pass, $dbname);

  // Check connection

  if ($conn->connect_error) { die("Connection failed: " . $conn->connect_error); }

  $name=' '; $comment=' '; // Declare variables to store values entered by user

  if(isset($_POST["tex1"])){ $name = $_POST["tex1"];} // isset() determines if variable is set and is not null

  if(isset($_POST["tex2"])){ $comment = $_POST["tex2"]; }

  $sql="INSERT into t2 values ("."'".$name."','".$comment."')";

  $result = $conn->query($sql);

  $sql2="SELECT * FROM t2"; // MySQL query to fetch data from table t2

  $result = $conn->query($sql2);

  if($result === FALSE) {   die(mysql_error());  }

  $result = $conn->query("SELECT * from t2") or trigger_error($conn->error);

  $row = $result->fetch_array(MYSQL_BOTH);

  while ($row = $result->fetch_array(MYSQL_BOTH)) {

  echo $row[0]; echo '  >>  '; echo $row[1]; echo '<br>'; // Printing values retrieved from db onto the screen

  } $conn->close(); // close connection

  ?>
```

The syntax `new mysqli($dbhst, $usnm, $pass, $dbname)` makes a new connection to the MySQL server.

After we have established a new connection to MySQL server, we insert the name and the comment provided by the user into the table that we created in MySQL.

The INSERT INTO statement is used to insert a new row of data in a table.

```
$sql="INSERT into t2 values (".'"'.$name."','".$comment."')";
```

Using the statement shown above, we insert name and comment entered by the user into the database.

Now to retrieve data from the database, we use the SELECT statement as shown,

```
$sql2="SELECT * FROM t2";
```

The `fetch_array()`function fetches a result as an associative or numeric array.

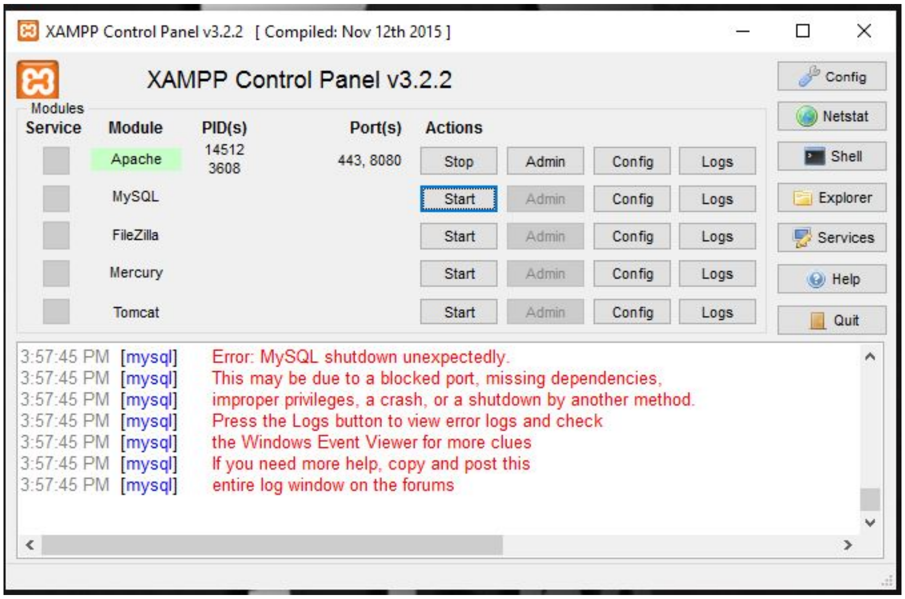
Now, using the `echo` function print the results retrieved from the database on the screen.

In the end, we close the connection established with the MySQL server using the statement

```
$conn->close();
```

Our code has now been completed. Now, we can test run the app on localhost.

4. Start the Apache web server from the XAMPP control panel.


XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	14512 3608	443, 8080	Stop	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
3:57:45 PM  [mysql]     Error: MySQL shutdown unexpectedly.
3:57:45 PM  [mysql]     This may be due to a blocked port, missing dependencies,
3:57:45 PM  [mysql]     improper privileges, a crash, or a shutdown by another method.
3:57:45 PM  [mysql]     Press the Logs button to view error logs and check
3:57:45 PM  [mysql]     the Windows Event Viewer for more clues
3:57:45 PM  [mysql]     If you need more help, copy and post this
3:57:45 PM  [mysql]     entire log window on the forums
```

5. Go to XAMPP installation directory. You will see a folder named ‘htdocs’. Save your PHP file in this folder.
6. Run the file on http://localhost or http://127.0.0.1:8080