**PHP Projects: Create an Opinion Poll Application**

In this PHP project**,**we are going to create an opinion poll application.

The opinion poll will consist of 3 major components;

**Front controller** – this is the index page that will determine the HTML code to be loaded. This will ensure that our application has a single entry point. This will give us more control over the application.

**Business Logic** – this will contain the PHP code for interacting with the database. This will allow us to separate the business logic from the presentation making our application easy to maintain

**Views** – this will contain the HTML code. We will have two pages namely;

* opinion.html.php – this will contain the HTML code with the question and options
* results.html.php – this will contain the HTML code that displays the opinion poll results

Assumptions made

The opinion poll will ask the question –

What is your favourite[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)Library?

Answers wold be

* JQuery
* MooTools
* YUI Library
* Glow

Here are the steps to create the application –

**Step 1) Database Connectivity**

This section assumes knowledge of MySQL and how to administer it, if you are not familiar with these MySQL, check our [SQL tutorials](https://www.guru99.com/sql.html) section.

Our application will have one table only with 3 fields namely;

* id – auto generate number as the primary key
* choice – the number representing a presidential candidate
* ts – the timestamp for the vote

The script below creates our js\_libraries table.

<?php

CREATE TABLE `js\_libraries` (

  `id` int(11) NOT NULL AUTO\_INCREMENT,

  `choice` tinyint(4) NOT NULL DEFAULT '0',

  `ts` timestamp NULL DEFAULT NULL,

  PRIMARY KEY (`id`)

);

?>

**Step 2) Coding our application**

Let’s now create our business logic layer that will handle the database connectivity. *'opinion\_poll\_model.php'*

<?php

class Opinion\_poll\_model {

 private $db\_handle; private $host = 'localhost'; private $db = 'opinion\_poll';private $uid = 'root'; private $pwd = 'melody';

    public function \_\_construct() {

        $this->db\_handle = mysqli\_connect($this->host, $this->uid, $this->pwd); //connect to MySQL server

        if (!$this->db\_handle) die("Unable to connect to MySQL: " . mysqli\_error());

        if (!mysqli\_select\_db($this->db\_handle,$this->db)) die("Unable to select database: " . mysqli\_error());

    }

    private function execute\_query($sql\_stmt) {

        $result = mysqli\_query($db\_handle,$sql\_stmt); //execute SQL statement

        return !$result ? FALSE : TRUE;

    }

    public function select($sql\_stmt) {

        $result = mysqli\_query($db\_handle,$sql\_stmt);

        if (!$result) die("Database access failed: " . mysqli\_error());

        $rows = mysqli\_num\_rows($result);

        $data = array();

        if ($rows) {

            while ($row = mysqli\_fetch\_array($result)) {

                $data = $row;

            }

        }

        return $data;

    }

    public function insert($sql\_stmt) {

        return $this->execute\_query($sql\_stmt);

    }

    public function \_\_destruct(){

        mysqli\_close($this->db\_handle);

    }

}

?>

  HERE,

* “public function \_\_construct()” is the class constructor method that is used to establish the database connection
* “public function execute\_query(…)” is the method for executing queries such as insert, update and delete
* “public function select” is the method for retrieving data from the database and returning a numeric array.
* “public function insert(…)” is the insert method that calls the execute\_query method.
* “public function \_\_destruct()” is the class destructor that closes the database connection.

Let’s now create the front controller *index.php*

<?php

require 'opinion\_poll\_model.php';

$model = new Opinion\_poll\_model();

if (count($\_POST) == 1) {

    echo "<script>alert('You did not vote!');</script>";

}

if (count($\_POST) > 1) {

    $ts = date("Y-m-d H:i:s");

    $option = $\_POST['vote'][0];

    $sql\_stmt = "INSERT INTO js\_libraries (`choice`,`ts`) VALUES ($option,'$ts')";

    $model->insert($sql\_stmt);

    $sql\_stmt = "SELECT COUNT(choice) choices\_count FROM js\_libraries;";

    $choices\_count = $model->select($sql\_stmt);

    $libraries = array("", "JQuery", "MooTools", "YUI Library", "Glow");

    $table\_rows = '';

    for ($i = 1; $i < 5; $i++) {

        $sql\_stmt = "SELECT COUNT(choice) choices\_count FROM js\_libraries WHERE choice = $i;";

        $result = $model->select($sql\_stmt);

        $table\_rows .= "<tr><td>" . $ libraries [$i] . " Got:</td><td><b>" . $result[0] . "</b> votes</td></tr>";

    }

    require 'results.html.php';

    exit;

}

require 'opinion.html.php';

?>

  HERE,

* “require 'opinion\_poll\_model.php';” loads the business logic class
* “$model = new Opinion\_poll\_model();” creates an instance of the business logic class
* “if (count($\_POST) == 1)…” performs the data validation and uses JavaScript to display a message box if not candidate has been voted for.
* “if (count($\_POST) > 1)…” checks if a vote has been selected by counting the number of items in the $\_POST array. If no item has been select, the $\_POST will only contain the submit item. If a candidate has been chosen, the $\_POST array will two elements, the submit and vote item. This code is also used to insert a new vote record and then display the results page
* “exit;” is used to terminate the script execution after the results have been displayed so that the opinion poll form is not displayed.
* “require 'opinion.html.php';” displays the opinion poll form if nothing has been selected.

Let’s now create the views. *opinion.html.php*

<html>

    <head>

        <title>JavaScript Libraries - Opinion Poll</title>

    </head>

    <body>

        <h2>JavaScript Libraries - Opinion Poll</h2>

        <p><b>What is your favorite JavaScript?</b></p>

        <form method="POST" action="index.php">

            <p> <input type="radio" name="vote" value="1" />JQuery

                <br /><input type="radio" name="vote" value="2" />MooToolsl

                <br /><input type="radio" name="vote" value="3" />YUI Library

                <br /><input type="radio" name="vote" value="4" />Glow

            </p>

            <p><input type="submit" name="submitbutton" value="OK" /></p>

        </form>

    </body>

</html>

*results.html.php*

<html>

    <head>

        <title>JavaScript Libraries Poll Results</title>

    </head>

    <body>

        <h2>Opinion Poll Results</h2>

        <p><b>What is your favorite JavaScript Library?</b></p>

        <p><b><?php echo $choices\_count[0]; ?></b> people have thus far taken part in this poll:</p>

        <p>

<table>

            <?php echo($table\_rows); ?>

        </table>

</body>

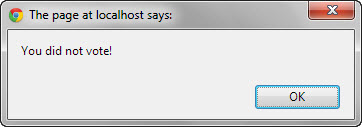
</html>

**Step 3) Testing our application**

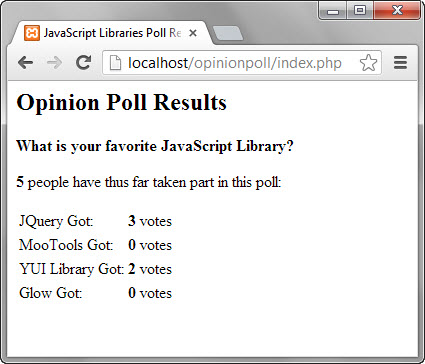
Assuming you have saved the files in opinionpoll folder, browse to the URL <http://localhost/opinionpoll/>

[](https://www.guru99.com/images/2013/04/opinion_poll_home.jpg)

If you click on Ok button without selecting a JS library, you will get the following message box.

[](https://www.guru99.com/images/2013/04/validation_msg.jpg)

Select a JS library then click on OK button. You will get the results page similar to the one shown below.

[](https://www.guru99.com/images/2013/04/opinion_poll_results.jpg)

**Summary**

* Dividing your application into business logic, front controller view layers  is a good application design practice
* JavaScript is useful for performing client side validation
* It is a good programming practice to use file.html.php for files that contain both html and php codes
* The opinion poll application demonstrates how the knowledge learnt in the previous lessons can be put together to developing a working application with a database back end.