

PHP

CSI2132

Introduction

- PHP is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML
- It has also evolved to include a command-line Interface capability and can be used in standalone graphical applications.
- It can be deployed on most web servers and also as a standalone shell on almost every operating system

Installation

- We will concentrate on installing PHP on a web server
- TWO options
 - Install Apache Webserver and Install PHP.
Configure Apache for PHP
 - Install WAMP: Which is a preconfigured for PHP, Apache and MySQL
- Note that Both Apache and PHP are open source

Installation – Option 1

- Download Apache From the following Path
<http://httpd.apache.org/download.cgi#apache22>
- Download PHP from the following path
<http://windows.php.net/download/>
- Install Apache
- Install PHP

Installing WAMP

- Download WAMP from <http://www.wampserver.com/en/>
- Double click on Wampserver2.2 and follow the wizard to install a preconfigured PHP on an Apache Server
- After Installation check on the quick launch bar to see if WAMP Server is running.
- If WAMP Server is running the icon changes colour to green

Configuring WAMP for Postgres DB

- Copy **libpq.dll** from **wamp\bin\php\php5.3.5** to **wamp\bin\apache\Apache2.2.17\bin**
- Restart the Wamp Server
- Click on Wamp ICON from the quick launch bar to open context menu
- Select PHP->PHP Extensions-> php_pgsq

Alternatively

Open php.ini file and uncomment
extensions=php_pgsq

PHP Web App

- PHP code is embedded into HTML webpage using `<?php ?>` tags
- Assuming we want to create a single page, which has a text box and command button such that when the user enters his/her name in the text box and click on the command button. A text showing “Greetings + User name” is displayed

First PHP Website

- Available IDE
 - ZEND Studio
 - NetBeans
 - Notepad++

First PHP Website

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4
5     <title>My Greetings Website</title>
6   </head>
7   <body>
8     <form id="testform" name="testform" method="post" action="" >
9       <p> <label for="name">Enter your name:</label>
10      <input name="iname" type="text" id="iname"/>
11     </p>
12     <p><input type="submit" name="bgreet" value="Greet"/>
13   </form>
14
15 </body>
16 </html>
```

First PHP Website

Check if the button is clicked if it is then display greetings

```
<?php
if(array_key_exists('iname', $_POST))
{
    echo "Hello Welcome " . $_POST['iname'] ;
}
?>
</body>
```

My First PHP Website

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>My Greetings Website</title>
  </head>
  <body>
    <form id="testform" name="testform" method="post" action="" >
      <p> <label for="name">Enter your name:</label>
        <input name="iname" type="text" id="iname"/>
      </p>
      <p><input type="submit" name="bgreet" value="Greet"/>
    </form>
    <?php
    if(array_key_exists('iname', $_POST))
    {
      echo "Hello Welcome " . $_POST['iname'] ;
    }
    ?>
  </body>
</html>
```

Publishing the website

- Save the file as greeting.php
- Create your website folder in c:\WAMP\www\ folder
- Copy greeting.php to the website folder, that was just created
- Open browser and type <http://localhost/foldername/greeting.php>

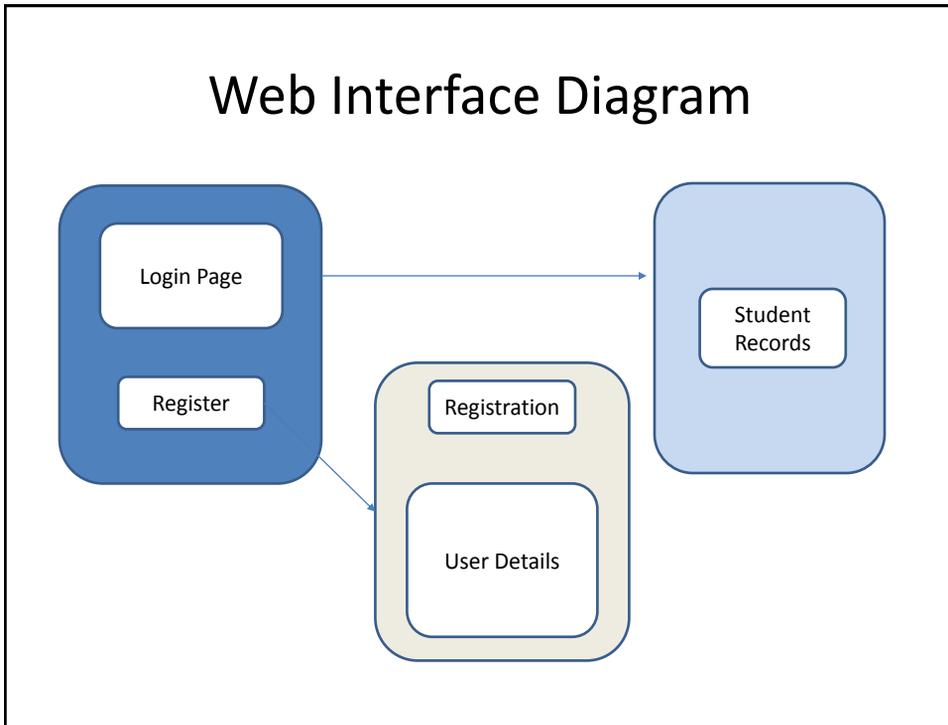
Database Driven Website

- The following are the objects required to communicate with a postgres database
- Pg_connect() – Connecting to a database
- Pg_query()- excuting query objects
- Pg_execute()-Executing query objects
- Pg_prepare() – preparing query statement to prevent SQL injection
- Pg_num_row()- counting number of row retrieved

Sample Web Site Design

- Student Records Database
- We will design a simple three page interface website
- Requirements
 - Each Student Must Register with student detail information including access password
 - Login using student number and password
 - View Student Grades
 - The index page is the login page

Web Interface Diagram



Database information

Run the following script using PGAdmin

```
CREATE SCHEMA php_project AUTHORIZATION username ;  
Set Search_path='php_project';
```

```
CREATE TABLE Student  
(  
    Student_NUM varchar(10) PRIMARY KEY,  
    LAST_NAME VARCHAR(15),  
    FIRST_NAME VARCHAR(15),  
    STUDENT_PASS VARCHAR(15),  
    STREET VARCHAR(30),  
    CITY VARCHAR(15),  
    GENDER VARCHAR(6),  
    EMAIL VARCHAR(20)  
);
```

```

CREATE TABLE COURSES
(
    COURSE_NUM varchar(7) PRIMARY KEY,
    COURSE VARCHAR(30)
);

CREATE TABLE GRADES
(
    Student_NUM varchar(10) ,
    Course_Num varchar(7),
    Year integer,
    Sec varchar(15),
    Grade Char(2),
    PRIMARY KEY (Student_num, Course_num)
);

```

```

INSERT INTO COURSES VALUES ('CSI1034','Intro to Programing');
INSERT INTO COURSES VALUES ('CSI1035','Software Usability');
INSERT INTO COURSES VALUES ('CSI2134','Databases I');
INSERT INTO COURSES VALUES ('CSI2045','Data Structures');
INSERT INTO COURSES VALUES ('CSI3056','Formal Languages');
INSERT INTO COURSES VALUES ('CSI3040','Intro to Computers');

```

```

INSERT INTO GRADES VALUES
('#','CSI1034',1,'FALL','A');

```

```

INSERT INTO GRADES VALUES
('#','CSI1035',1,'FALL','B+');

```

```

INSERT INTO GRADES VALUES
('#','CSI2045',1,'FALL','A+');

```

```

INSERT INTO GRADES VALUES
• ('#','CSI2134',2,'WINTER','A')

```

Replace # with your student Number or an arbitrary student number

Login Page

- Create an HTML page with label student number and an input box for student number
- Label password and an input box for password

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style1.css"/>
    <title>Student Database</title>
  </head>
  <body>
    <div id="header">USER LOGIN FORM</div>
    <form method="POST" action="">
      <p>Student #: <input type="text" name="studentnum" id="studentnum"/></p>
      <p>Password: <input type="password" name="userPassword" id="userPassword" /></p>
      <p><input type="submit" value="Login" name="login" id="login" /></p>
    </form>
    <a href="register.php">Register</a>
  </body>
</html>

```

Login Page

- Session object: this is used to store objects across pages.
- To start a session in PHP we use `session_start()` method
- To store values in Session object we use `$_SESSION[]` associative array
- GET and POST: these are used for transferring form data for processing and across pages
- POST is more secure than GET since get makes the form data visible
- PHP stores “posted” values using associative array `$_POST[]` and `$_GET[]` associative array for GET functions

Login Page PHP Code

- Here What we want to do is to validate the login information entered by the user
- Redirect to Records Page if record exist
- Store User information into session object so it can be used across pages
- Use POST object to handle login information

- Under the `</head>` tag add the following PHP code

```
<?php
session_start();
//Check if the login button was clicked and the login value is in the POST array object
if(array_key_exists('login',$_POST))
{
    //RETRIEVE THE USERS STUDENT NUMBER AND PASSWORD FROM THE LOGIN FORM
    $studentnum=$_POST['studentnum'];
    $password=$_POST['userPassword'];

    //GET DATABASE CONNECTION STRING
    $conn_string="host=localhost port=5432 dbname=dantw005 user=postgres password=nicedan";

    //CONNECT TO DATABASE
    $dbconn=pg_connect($conn_string) or die('Connection failed');

    //QUERY DATABASE TO SEE IF USER EXIST
    //USE PARAMETERS TO AVOID SQL INJECTION
    $query="SELECT * FROM php_project.Student WHERE Student_NUM=$1 AND STUDENT_PASS=$2";

    //PREPARE THE STATEMENT TO AVOID SQL INJECTION
    $stmt=pg_prepare($dbconn,"ps",$query);
    $result=pg_execute($dbconn,"ps",array($studentnum,$password));
```

- Continue with the following code
- Header("location: url") – is for automatically redirecting to another page

```

if(!$result){
    die("Error in SQL query:" .pg_last_error());
}
//CHECK ROW COUNT IF ROW COUNT IS GREATER THAN 0 RECORD EXIST
$row_count= pg_num_rows($result);
if($row_count>0)
{
    //KEEP USER INFORMATION ACROSS PAGES AND REDIRECT TO RECORDS PAGE
    $_SESSION['studentnum']=$studentnum;
    header("location: http://localhost/Tutorial/records.php");
    exit;
}
echo "Data Successfully Entered ". "<a href='index.php'>login now</a>";

//free memory
pg_free_result($result);
//close connection
pg_close($dbconn);
}
?>
<body>

<div id="header">USER LOGIN FORM</div>
<form method="POST" action="">

```

Register Page

- The idea of the register page is to see how to insert data into a database using PHP
- Information to save include student number, last name, first name, password, confirm password, street, city and gender

Register Page

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style1.css"/>
    <title>Student Database</title>
  </head>
  <body>
    <div id="header">USER REGISTRATION FORM</div>
    <form id="testform" name="testform" method="post" action="" >
      <p> <label for="istudentnum">Student #:</label>
        <input name="istudentnum" type="text" id="repno"/>
      </p>
      <p> <label for="ilastname">Last name:</label>
        <input name="ilastname" type="text" id="ilastname"/>
      </p>
      <p> <label for="ifirstname">First name:</label>
        <input name="ifirstname" type="text" id="ifirstname"/>
      </p>
      <p> <label for="ipassword">Password:</label>
        <input name="ipassword" type="password" id="ipassword"/>
      </p>
      <p> <label for="iconfpass">Confirm password:</label>

```

Register Page

```

      <input name="iconfpass" type="password" id="iconfpass"/>
    </p>
    <p>
      <label for="istreet">Street:</label>
      <input name="istreet" type="text" id="istreet"/>
    </p>
    <p>
      <label for="icity" id="formLabel">City:</label>
      <input name="icity" type="text" id="icity"/>
    </p>
    <p>
      <label for="igender">gender:</label>
      <select name="igender">
        <option value="male">male</option>
        <option value="female">female</option>
      </select>
    </p>
    <p>
      <label for="iemail">Email:</label>
      <input name="iemail" id="iemail" type="text"/>
    </p>
    <p>
      <input type="submit" name="save" value="Register"/>
    </p>
  </form>
</body>
</html>

```

PHP CODE FOR REGISTER PAGE

```

</head>
<?php
if(array_key_exists('save',$_POST))
{
    $studentnum=$_POST['istudentnum'];
    $lastname=$_POST['ilastname'];
    $firstname=$_POST['ifirstname'];
    $password=$_POST['ipassword'];
    $street=$_POST['istreet'];
    $city=$_POST['icity'];
    $gender=$_POST['igender'];
    $email=$_POST['iemail'];
    $conn_string="host=localhost port=5432 dbname=dantw005 user=postgres password=nicedan";
    $dbconn=pg_connect($conn_string) or die('Connection failed');
    $query="INSERT INTO php_project.student(Student_num,last_name,
    First_name,Student_Pass,Street,City,Gender,Email) VALUES ('$studentnum','$lastname',
    '$firstname','$password','$street','$city','$gender','$email')";
    $result=pg_query($dbconn,$query);
    if(!$result){
        die("Error in SQL query:" .pg_last_error());
    }
    echo "Data Successfully Entered ". "<a href='index.php'>login now</a>";
    //free memory
    pg_free_result($result);
    //close connection
    pg_close($dbconn);
}
?>
</body>

```

Records Page

- The records page contains academic records of all students
- When a registered user logs in, the records of the user is displayed
- Sessions are used to prevent unauthorized users from viewing records

Records Page

- Create html page with a table and show dummy records

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="css/style1.css"/>
    <title>Student Database</title>
  </head>
  <body>
    <div id="header">Student Record Details</div>
    <table>
      <tr>
        <th>Course</th>
        <th>Year</th>
        <th>Session</th>
        <th>Grade</th>
      </tr>
      <tr>
        <td>600000</td>
        <td> 3</td>
        <td>Fall</td>
        <td>F</td>
      </tr>
    </table>
    <br/>
    <br/><a href="update_profile.php?studentnum=<?php $studentnum; ?>">Update Profile</a>
  </body>
</html>
```

Add PHP Code to Records Page

```
<?php
session_start();
//RETRIEVE STUDENT NUMBER FROM SESSION OBJECT
if(!isset($_SESSION['studentnum'])){
    echo "Please ".<a href='Login.php'>Login</a>";
    exit;
}
$dbh=pg_connect("host=localhost port=5432 dbname=dantw005 user=postgres password=ni");
if (!$dbh) {
    die("Error in connection: " . pg_last_error());
}
$studentnum=$_SESSION['studentnum'];
// QUERY TO SELECT FROM THREE TABLES
$sql = "SELECT c.COURSE,g.YEAR,g.SEC,
        g.GRADE FROM php_project.Student s,php_project.Grades g,php_project.Course:
WHERE s.Student_Num=g.Student_Num AND g.Course_Num=c.Course_Num
AND s.Student_Num=$1";
$stmt = pg_prepare($dbh,"ps", $sql);
$result=pg_execute($dbh,"ps",array($studentnum));
if (!$result) {
    die("Error in SQL query: " . pg_last_error());
}
//free memory
pg_free_result($result);
//close connection
pg_close($dbconn);
?>
```

rep length:1735 lines:59 Ln:19 Col:45 Sel:0 Dos\Windows ANSI INS

Continue....

- Modify the table to show actual student records from resultset

```
</tr>
</tr>
<!-- LOOP THROUGH THE RECORDSET AND DISPLAY ALL RECORDS IN A TABLE-->
<?php while($row=pg_fetch_array($result)) { ?>
  <tr>
    <td><?php echo $row[0]; ?></td>
    <td> <?php echo $row[1]; ?></td>
    <td><?php echo $row[2]; ?></td>
    <td><?php echo $row[3]; ?></td>
  </tr>
<?php }?>
</table>
<br/>
<br/><a href="update_profile.php?studentnum=<?php $studentnum; ?>">Update Profile</a>
/html>
```

THANK YOU