

INTRODUCTION

1. INTRODUCTION

1.1 Problem Statement:

This project is aimed at developing an online application for the Training and Placement Cell of the college. The system is an online application that can be accessed throughout the organization and outside as well with proper login credentials provided. This system can be used as an application for the Training and Placement Office of the college to manage the student information with regard to placements. The Feedback module comprises of two sub modules namely 1. Feedback collected, 2. Feedback enquired. Students logging in should be able to upload their information in the form of a Curriculum Vitae.

1.2 Objective of Project :

Feedback is one of the most pivotal and necessary element for an organization to function properly and to improvise. It's role in an educational organization is even important. Hence, to help the preceding batches of students getting into professional organizations, the feedback from previously passed out students working in corporations on real time projects not only provides the students a pre-requisite to have a jump start when they step out of the college. This fills up the gap that exists between the unexposed students to reality and also provides an effective and one stop database for the entire college.

1.3 Scope of Project:

- A well-built database for Alumni, students and faculty alike.
- An effective and portable platform for giving feedback.
- A one stop destination for students seeking guidance for joining a specific organization.
- Expanding the career options of divergent thinking student community.
- Helping students to realize and tap their true potential.

1.4 Existing system:

- Filling of forms by students

Here a form is given to students in which he/she has to fill with some details such as his name, roll number, contact details, percentages(from first year to till date),

Intermediate particulars(name of institution, place, year of pass, percentage), SSC particulars(name of institution, place, year of pass, percentage).

- Preparing excel sheet

From the data collected through filled-forms and green book, excel sheets are prepared.

These excel sheets are used to prepare a list of students who full-fill the requirements of a company visiting the campus and these students are eligible to attend the campus placement.

1.5 Proposed system:

In the proposed system the user need not do all the hectic work. he will be provided with an interface with which he can easily get his work done.

The following are the facilities that are provided by the system to the user.

- Feedback Module

The module will provide a gateway for the enthusiastic Alumni interested, to given their view on the existing work culture and needs also termed feedback.

- Administration Module

The module will provide a user interface for adding student, updating student details, deleting feedback responses, view all students, view all feedbacks.

1.6 Product features :

- Complete automation is possible in this sector, which is against the main disadvantage namely time consuming.
- Can maintain student details who have been studying in the college.
- Any kind of lists based on students profile can be retrieved with in less time.
- Results are uploaded directly from net so that no errors exist in calculating percentages.
- Effective and good means of communication can be facilitated as we have included mailing module in the proposed system.
- User can also register student and can view and delete there profile.

SYSTEM REQUIREMENTS

2.1 REQUIREMENTS

2.1.1 Requirement Analysis

We are overcoming the difficulty of student details which were manual in the current system and here we generate detailed information about the students which will save our time to inform each and every batch and section and student profile is maintained.

2.1.2 Functional Requirements:

A student should be able to login to the system through the first page of the application, and mention his required user name and he should get his details which he can view and update it. An administrator can login into his account and he will update the student information.

2.1.3 Non Functional Requirements:

Usability

This section includes all of those requirements that effect usability.

- We get the response within seconds.
- The software must have a simple, user-friendly interface so customers can save time and confusion.
- As the project is made using php, it has fast loading time then the website made using any other language.

Reliability

- The system is more reliable because of the qualities that are inherited from the chosen platform php. The code built by using php is more reliable

Supportability

- The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform. This application is being developed using xampp, hence it is extremely portable.

Implementation

- The system is implemented in web environment. The apache tomcat is used as the web server and windows Xp/vista/7 is used as the platform.

Interface

- The user interface is based on the web browser. The application is developed using JS and HTML.
- The Interface design is aimed at a flexible front-end communication to provide the user with clear information in navigating a user-friendly interface is planned.

2.2 Software Requirements :

| | |
|-----------------|---|
| IDE | : Vs code or Sublime Text |
| Softwares | : Google Chrome or any other web browsers |
| Database server | : Xampp (Localhost) |

2.3 Hardware Requirements:

| | | |
|-----------|---|-------------------|
| Processor | : | Intel p4 or later |
| RAM | : | 512 MB or More |
| Hard Disk | : | 40 GB or more |

CODING AND IMPLEMENTATION

3.1 SYSTEM DESIGN

Definition :

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

Design of the system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution to “how to” approach to the creation of a new system. This important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.

ER Modeling:

The schemas for the database application can be displayed by means of graphical notation known as Entity Relationship diagram.

The ER model describes data as entities, relationships and attributes.

Entities and attributes :

An entity may be an object with a physical existence (for e.g. A particular person, car or employee) or it may be an object with a conceptual existence (for e.g. a company, a job, or a university course)

Each entity has attributes i.e. the particular properties that describe it. The attribute values that describe each entity become a major part of the data store in the database.

Relationships between entities :

Whenever an attribute of one entity type refers to another entity type, a relationship exists. In the initial design of entity types, relationships are typically captured in the form of attributes. As the design is refined these attributes get converted into relationships between entity types.

In the ER diagrams the emphasis is on representing the schemas rather than the instances. This is more useful in the database design because a database schema changes rarely, whereas contents of the entity sets change frequently.

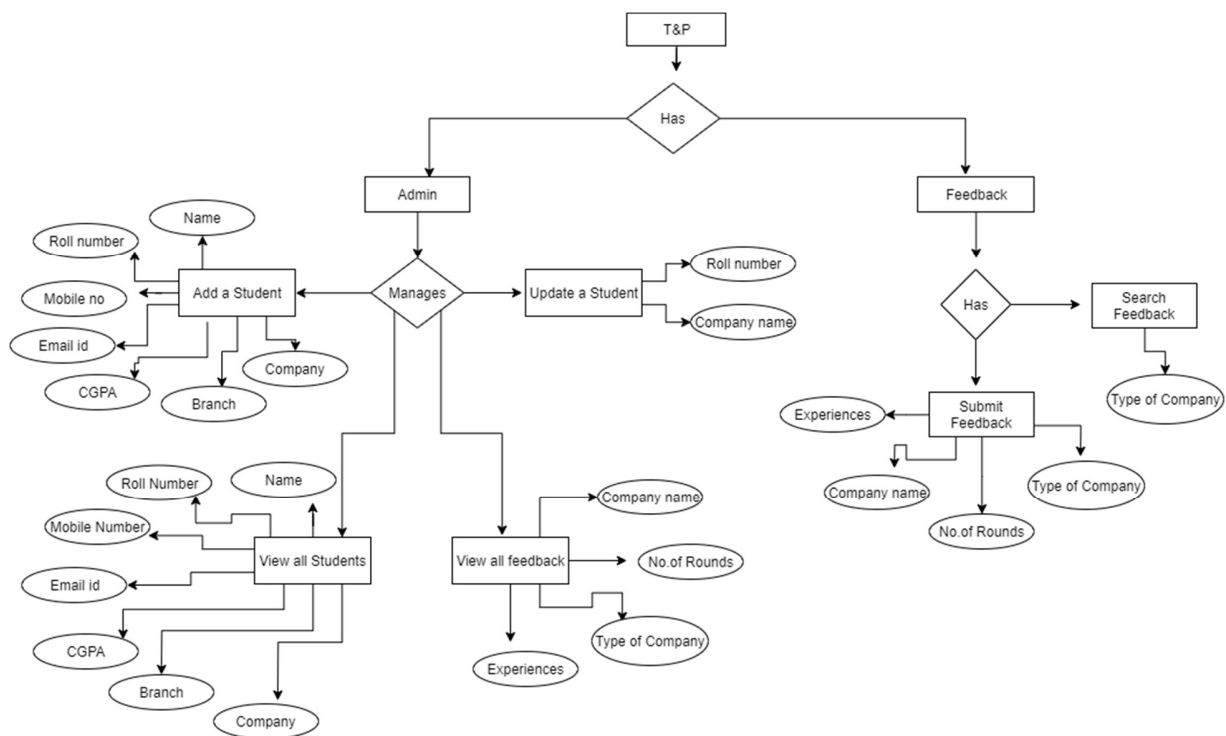


FIGURE 3.1.1 : ER Diagram

3.2 DESCRIPTION OF FRONT END

HYPER TEXT MARKUP LANGUAGE:

The Hyper Text Markup language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML version 4.0. HTML 4.0 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 3.2.

Why to use HTML:

Web site is a collection of pages, publications and documentation that reside on web server. While these page publication and a document as a formatted in any single format you should use HTML for home page and all primary pages and the site. This will enable the millions of web users it considered first formatting any new material you plan to publish on the web. HTML documents are platform independent, meaning that they don't conform to any standard if they are created properly you can move home page to any server platform or you can access them with any compliant www browser.

1. **<HTML>...</HTML>** - All HTML files start and end with the tag pair.
2. **<HEAD>...</HEAD>** - All HTML have a pair of "HEAD" tags that indicate what the title and other attributes of the page are going to be.
3. **<TITLE>...</TITLE>** - this tag indicates what the title of the HTML file is going to be on the BROWSER window title.
4. **<BODY>...</BODY>** - this tag pair is to logically separate the HTML file into the header and the body. Usually the header contains information regarding the html where as the body contains information that the HTML file must actually contain.
5. **The HTML template must look like.**
`<!DOCTYPE HTML PUBLIC "THIS IS AN EXAMPLE">`
`<HTML>`

<HEAD>

<TITLE> YOUR TITLE GOES HERE</TITLE>

</HEAD>

</HTML>

6. **<P>...</P>** - This tag pair used to indicate the paragraph. Any text that needs to be separated into a paragraph must be put in within a paragraph tag.

7. **...** - This tag pair is used to indicate the text within tag pair must be in bold letters.

8. **<I>...</I>** - This tag pair is used to indicate the text within the tag pair must be in italic letters.

9. **** - This tag is used to embed images in the HTML pages. The SRC attribute is used to locate the file name under a directory, the ALT attribute is used to indicate the TOOLTIP message that must appear, and HEIGHT and WIDTH indicate the height and the width of the images that is being shown on the HTML pages.

10. **<H1 ALIGN="CENTER">...</H1>** - This pair of tags is used to indicate that the text must be main title for the HTML page. The ALIGN attribute can be used to set the alignment to "center" or "left" or "right"

11. **<H1>Heading1</H1>**

H2>Heading2</H2>

<H3>Heading3</H3> - This set of tags will show the Headings in smaller fonts as the heading increases.

12. **ALIGN** – The align attribute can be used for headings as well. For <P>...</P> tags also, the ALIGN attribute can be used.

13. **
** - Used to insert a carriage return in the HTML file. The attribute to be used for this is the CLEAR attribute.

14. **<CENTER>...</CENTER>** - To center the entire block of text this tags are used.

15. **<A>...** - Anchor Tags. These tags are used linking namely hyper linking.

Example:

<A HREF=<http://www.ibm.com>>Visit IBM Web Pages

16. Images Basics: Image Tag is used to embed images in the html document. The general syntax is

17. <TABLE>...</TABLE> - This is used to specify the table type of layout in the HTML document.

<TABLE BORDER="1">

<TR>

<TH>Car</TH>

<TH>Company</TH>

</TR>

<TR>

<TH>Concorde</TH>

<TH>Chrysler</TH>

</TR>

</TABLE>

18. Fonts: The ... tag is used to specify text in a particular font.

Example:

 this is a line of text with size 10

Syntax :...

3.3 DESCRIPTION OF SERVER LANGUAGE

Features of PHP:

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document rather than calling an external file to process data. The code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor to Microsoft's Active Server Pages (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Web servers. Software that uses PHP includes Joomla, WordPress, MyBB, and Drupal. PHP was originally created by Rasmus Lerdorf in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the formal reference to the PHP language. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term *PHP*. While PHP originally stood for "Personal Home Page", it is now said to stand for "PHP: Hypertext Preprocessor", a recursive acronym.

History:

Licensing

PHP is free software released under the PHP License, which insists that:

```
Products derived from this software may not be called
"PHP", nor
may "PHP" appear in their name, without prior written
permission
from group@php.net. You may indicate that your software
works in
conjunction with PHP by saying "Foo for PHP" instead of
calling
it "PHP Foo" or "phpfoo"
```

This restriction on use of the name *PHP* makes it incompatible with the GNU General Public License (GPL).

Usage:

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on Web sites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most Web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic Web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a Web server to a client, such as Microsoft's ASP.NET, Sun Microsystems' Java Server Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, Yii Framework, and Zend Framework, offering features similar to other web application frameworks.

The LAMP architecture has become popular in the Web industry as a way of deploying Web applications. PHP is commonly used as the *P* in this bundle alongside Linux, Apache and MySQL, although the *P* may also refer to Python or Perl or some mix of the three. Similar packages are also available for Windows and OS X, then called WAMP and MAMP, with the first letter standing for the respective operating system.

As of April 2007, over 20 million Internet domains had Web services hosted on servers with PHP installed and mod_php was recorded as the most popular Apache HTTP Server module. PHP is used as the server-side programming language on 75% of all Web sites. Web content management systems written in PHP include Media Wiki, Joomla, eZ .

Syntax:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>PHP Test</title>
  </head>
  <body>
    <?php
      echo 'Hello World';
    ?>
```

```
</body>
</html>
```

Hello world program in PHP code embedded in HTML code

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described in PHP code). The most common delimiters are `<?php` to open and `?>` to close PHP sections. `<script language="php">` and `</script>` delimiters are also available, as are the shortened forms `<?` or `<?='` (which is used to echo back a string or variable) and `?>` as well as ASP-style short forms `<%or <%=` and `%>`. While short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, `<?php` and `?>`, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Data types:

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted (") and heredoc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: `/* */` marks block and inline comments; `//` as well as `#` are used for one-line comments. The echo statement is one of several facilities PHP provides to output text, e.g., to a Web browser.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. *if* conditions, *for* and *while* loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signed integer equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is *NULL*. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with

both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax.

The Standard PHP Library (SPL) attempts to solve standard problems and implements efficient data access interfaces and classes.

Functions:

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming, although it does support multi process programming on POSIX systems.

Additional functions can be defined by a developer:

```
function myFunction() { //declares a function, this is named
myFunction
    return 'John Doe'; //returns the value 'John Doe'
}

echo 'My name is ' . myFunction() . '!'; //outputs the text and
the return variable of the                //myFunction, the
function is also called
//the result of the output will be 'My name is John Doe!'
```

Objects:

Basic object-oriented programming functionality was added in PHP 3 and improved in PHP 4. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like value types. The drawback of this method was that the whole object was copied when a variable was assigned or passed as a parameter to a method. In the new approach, objects are referenced by handle, and not by value. PHP 5 introduced private and protected member variables and methods, along with abstract classes, final classes, abstract methods, and final methods. It also introduced a standard way of declaring constructors and destructors, similar to that of other object-oriented languages such as C++, and a standard exception handling model. Furthermore, PHP 5 added interfaces and allowed for multiple interfaces to be implemented. There are special interfaces that allow objects to interact with the runtime system. Objects implementing Array Access can be used with array syntax and objects implementing Iterator or Iterator Aggregate can be used with the foreach language construct. There is no virtual table feature in the engine, so static variables are bound with a name instead of a reference at compile time


```
class Person {
    public $firstName;
    public $lastName;

    public function __construct($firstName, $lastName = '') {
        //Optional parameter
        $this->firstName = $firstName;
        $this->lastName = $lastName;
    }

    public function greet() {
        return "Hello, my name is " . $this->firstName . " " .
        $this->lastName . ".";
    }

    public static function staticGreet($firstName, $lastName) {
        return "Hello, my name is " . $firstName . " " .
        $lastName . ".";
    }
}

$he = new Person('John', 'Smith');
$she = new Person('Sally', 'Davis');
$other = new Person('Joe');

echo $he->greet(); // prints "Hello, my name is John Smith."
echo '<br />';
echo $she->greet(); // prints "Hello, my name is Sally Davis."
echo '<br />';
echo $other->greet(); // prints "Hello, my name is Joe ."
echo '<br />';
echo Person::staticGreet('Jane', 'Doe'); // prints "Hello, my
name is Jane Doe."
```

Visibility of properties and methods

The visibility of PHP properties and methods refers to visibility in PHP. It is defined using the keywords public, private, and protected. The default is public, if only var is used; var is a synonym for public. Items declared public can be accessed everywhere. protected limits access to inherited classes (and to the class that defines the item). private limits visibility only to the class that defines the item. Objects of the same type have access to each other's private and protected members even though they are not the same instance. PHP's member visibility features have sometimes been described as "highly useful." However, they have also sometimes been described as "at best irrelevant and at worst positively harmful."

Speed optimization:

PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the Web page is accessed, PHP scripts can also be deployed in executable format using a PHP compiler.

Code optimizers aim to enhance the performance of the compiled code by reducing its size, merging redundant instructions and making other changes that can reduce the execution time. With PHP, there are often opportunities for code optimization. An example of a code optimizer is the Accelerator PHP extension.

Another approach for reducing compilation overhead for PHP servers is using an opcode cache. Opcode caches work by caching the compiled form of a PHP script (opcodes) in shared memory to avoid the overhead of parsing and compiling the code every time the script runs. An opcode cache, APC, is planned to be built into an upcoming release of PHP.

Opcode caching and code optimization can be combined for best efficiency, as the modifications do not depend on each other (they happen in distinct stages of the compilation).

XAMPP:

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP's name is an acronym for:

- X (to be read as "cross", meaning cross-platform)
- Apache HTTP Server
- MySQL
- PHP
- Perl

The program is released under the terms of the GNU General Public License and acts as a free web server capable of serving dynamic pages. XAMPP is available for Microsoft Windows, Linux, Solaris, and Mac OS X, and is mainly used for web development projects.

Requirements and features:

XAMPP requires only one zip, tar, 7z, or exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache/MySQL/PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin.

Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.

It is offered in both a full, standard version and a smaller version.

Components:

XAMPP 1.7.7, including

- Apache 2.2.21
- MySQL 5.5.16
- PHP 5.3.8
- phpMyAdmin 3.4.5
- FileZilla FTP Server 0.9.39
- Tomcat 7.0

3.4 CODING

Main.html :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      body
      {
        background-image: url("main.jpg");
        background-color: #cccccc;
      }
      p.a
      {
        font-family: "Times New Roman", Times, serif;
        font-size: 100px;
        text-align: center;
        color: #FFE4C4;
      }
      p.b
      {
        font-family:"Times New Roman" , Times, serif;
        font-size: 80px;
        text-align: center;
        color: #FFE4C4;
      }
      p.c
      {
        font-family: sans-serif, Times, serif;
        font-size: 25px;
```

```
text-align: center;
color: white;
}
</style>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Main Page</title>
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" >
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">
<link rel="stylesheet" href="form.css">
</head>
<body >
<div class="container">
<div class="imagebg">
<p class="a">GPREC</p>
<p class="b">Training & Placements cell</p>
<p class="c">The Training and Placement Cell is headed by Dr.B.Veerabhadra Reddy,
Professor of Mechanical Engineering. Apart from a full-time Addl. Training and Placement
Officer, a committee, with one faculty member and one student member from each department,
helps the cell function effectively by coordinating its activities. The cell is fully equipped to
churn out industry ready candidates and is actively liaisoning with industries of repute and keeps
the students informed of various job opportunities.</p>
</div>
<div class="row " style="margin-top: 50px">
```

```
<div class="col-md-6 col-md-offset-3 form-container">
  <div class="row">
    <div class="col-sm-12 form-group">
      <a href="button.php"><button type="submit" class="btn btn-lg btn-warning
btn-block">FEEDBACK SYSTEM</button></a>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12 form-group">
      <a href="ADMIN/adminlogin.php"><button type="submit" class="btn btn-
lg btn-warning btn-block" >ADMIN</button></a>
    </div>
  </div>
</div>
</div>
</div>
</body>
</html>
```

Button.php :

```
<!DOCTYPE html>
<html>
<title>FEEDBACK MODULE</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<style>
.w3-button {width:250px;}
body
{
  background-image: url("main.jpg");
```

```
        background-color: #cccccc;
    }
    p.a
    {
        font-family: "Times New Roman", Times, serif;
        font-size: 100px;
        text-align: center;
        color: #FFE4C4;
    }
    p.b
    {
        font-family:"Times New Roman" , Times, serif;
        font-size: 35px;
        text-align: center;
        color: white;
    }
    p.c
    {
        font-family: sans-serif, Times, serif;
        font-size: 25px;
        text-align: center;
        color: white;
    }
</style>
<body>
    <p class="a">Welcome to Feedback Module</p>
    <p class="b">Hello GPRECIAN, kindly submit your feedback about your last interview
and give your valuable suggestions to juniors for preparing themselves to crack the interviews,
mention areas to focus to get placed in any reputed company. And also those who are preparing
for interviews refer your seniors experiences and seek their suggestions which will help you
more in preparing for particular company.</p>
```

```
<center><div class="w3-container">
  <a href="tnpfeedback\feedbacksubmit.php"><button class="w3-button w3-light-
grey">SUBMIT A FEEDBACK</button></a>
  <a href="student1\tnpsearch.php"><p><button class="w3-button w3-light-grey">SEARCH
FOR FEEDBACK</button></p></a>
  <a href="resume.php"><p><button class="w3-button w3-light-grey">UPLOAD
RESUME</button></p></a>
</div>
<P>
  <form action="redirect.php" method='POST'>
    <CENTER><input type="submit" name="send" value="BACK TO MAIN"
style="font-size:18px; " /></CENTER>
  </form></p>
</P>
</center>
</body>
</html>
```

Redirect.php :

```
<?php
session_start();
if(isset($_POST['send']))
{
    session_destroy();
    echo '<script>window.location.href="../main/main.php"</script>';
}
?>
```

Mainconnect.php :

```
<?php
$servername='localhost';
```



```
$username='root';
$password="";
$db="tnpfeedback";
$con= new mysqli($servername,$username,$password,$db);

$name = $_POST['search'];
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$result = mysqli_query($con, "SELECT * FROM feedback WHERE companyname
LIKE '%{$name}%'");

while ($row = mysqli_fetch_array($result))
{
    echo $row['companyname'] . " " . $row['rounds'] . " " . $row['screening'] . " " .
$row['details'];
    echo "<br>";
}

mysqli_close($con);
?>
```

Feedbacksubmit.php :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Feedback Form</title>
    <!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >

<!-- Optional theme -->

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" >

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">

<link rel="stylesheet" href="form.css">

</head>

<body >

<div class="container">

<div class="imagebg"></div>

<div class="row " style="margin-top: 50px">

<div class="col-md-6 col-md-offset-3 form-container">

<h2>Feedback Form</h2>

<p> Please provide your feedback below: </p>

<form role="form" method="post" id="reused_form"
action="feedbackconnect.php">

<div class="row">

<div class="col-sm-12 form-group">

<p>

<label>Company Name :</label>

<input type="text" class="form-control" id="name" name="name"
required>

</p>

</div>

</div>

<div class="row">

<div class="col-sm-12 form-group">

<p>

<label>Total no.of Rounds :</label>
```

```
<input type="text" class="form-control" id="name" name="rounds"
required>

</p>
</div>
</div>
<div class="row">
  <div class="col-sm-12 form-group">
    <label>Type of Company :</label>
    <p>
      <label class="radio-inline">
        <input type="radio" name="experience" id="radio_experience"
value="Service" >
          Service Based
        </label>
        <label class="radio-inline">
          <input type="radio" name="experience" id="radio_experience"
value="Product" >
            Product Based
        </label>
      </p>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12 form-group">
      <label for="comments"> Share your interview experience and suggestions to
juniors:</label>
      <textarea class="form-control" type="textarea" name="comments"
id="comments" placeholder="Your Comments" maxlength="6000" rows="7"></textarea>
    </div>
  </div>
  <div class="row">
```

```
<div class="col-sm-12 form-group">
    <button type="submit" class="btn btn-lg btn-warning btn-block" >SUBMIT
    FEEDBACK</button>
</div>
</div>
</form>
<form action="redirect.php" method='POST'>
    <CENTER><input type="submit" name="send" value="BACK TO MAIN"
    style="font-size:18px; " /></CENTER>
</form></p>
<div id="success_message" style="width:100%; height:100%; display:none; ">
<h3>Posted your feedback successfully!</h3> </div>
<div id="error_message" style="width:100%; height:100%; display:none; ">
<h3>Error</h3> Sorry there was an error sending your form. </div>
</div>
</div>
</div>
</body>
</html>
```

Feedbackconenct.php :

```
<?php
//xampp server details
$servername = "localhost";
$username = "root";
$password = "";
//create connection
$conn = new mysqli($servername, $username, $password) or die("Not connected");
if($conn)
{
```

```
        echo "connected";
        echo "<br>";
    }
    if(!mysqli_select_db($conn,'tnpfeedback'))
    {
        echo "Database not found";
        echo "<br>";
    }
    //reading data through html form
    $com_name=$_POST['name'];
    $round=$_POST['rounds'];
    $screening = $_POST['experience'];
    $details=$_POST['comments'];
    //inserting data into database
    $sql="INSERT INTO feedback(companyname,rounds,type,details)
    values('$com_name','$round','$screening','$details')";
    if(!mysqli_query($conn,$sql))
    {
        $message = "Your feedback is not recorder";
        echo "<script type='text/javascript'>alert('$message');</script>";
        echo "<br>";
    }
    else
    {
        $message = "Thanks for submitting feedback";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
    header("Location:gohome.php");
?>
```

Gohome.php :

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

    body

    {

        background-image: url(main1.jpg);

        background-size: cover;

        text-align: center;

    }

.button {

    border-radius: 4px;

    background-color: #f4511e;

    border: none;

    color: #FFFFFF;

    text-align: center;

    font-size: 15px;

    padding: 15px;

    width: 300px;

    transition: all 0.5s;

    cursor: pointer;

    margin: 5px;

    object-position: center;

}

.button span {

    cursor: pointer;
```

```
display: inline-block;
position: relative;
transition: 0.5s;
}
.button span:after {
content: '\00bb';
position: absolute;
opacity: 0;
top: 0;
right: -20px;
transition: 0.5s;
}
.button:hover span {
padding-right: 25px;
}
.button:hover span:after {
opacity: 1;
right: 0;
}
</style>
</head>
<body>
<h2>Thanks for submitting the feedback.</h2>
<form action="redirect.php" method="POST">
<a href=""><button class="button" name="send"><span>GO HOME</span></button></a>
</form>
</body>
</html>
```

Tnpsearch.php :

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      body {
        background-image: url("main2.jpg"); /* The image used */
        background-color: #cccccc; /* Used if the image is unavailable */
        height: 500px; /* You must set a specified height */
        background-position: center; /* Center the image */
        background-repeat: no-repeat; /* Do not repeat the image */
        background-size: cover; /* Resize the background image to cover the entire container */
      }
      h2
      {
        color:red;
      }
      table,th,td
      {
        border:1px solid black;
      }
      th
      {
        text-align: center;
      }
    </style>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```



```
<title>Search for company feedback</title>
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" >
</head>
<body >
  <div class="container">
    <div class=""></div></p>
    <form action="" method="POST">
      <div class="row">
        <div class="row " style="margin-top: 50px">
          <div class="col-md-6 col-md-offset-3 form-container">
            <h2>Feedback Search</h2>
            <form action="searchconnect.php" method="POST">
              <div class="row">
                <div class="col-sm-12 form-group">
                  <p>
                    <label>Search with company Name :</label>
                    <label class="radio-inline">
                      <input type="text" class="form-control" id="search" name="search"
required>
                    </label>
                  </p>
                </div>
              </div>
            </div>
          <div class="row">
            <div class="col-sm-12 form-group">
              <button type="submit" name="vamsi" class="btn btn-lg btn-warning btn-
block">SUBMIT</button>
            </div>
          </div>
        </div>
      </div>
    </form>
  </div>
</body>
</html>
```

```
</div>
</form>
<table>
  <tr>
    <th>Company Name</th>
    <th>No.of rounds include in recruitment</th>
    <th>Type of Company</th>
    <th>Feedback Report</th>
  </tr>
  <?php
    $connection=mysqli_connect("localhost","root","");
    $db=mysqli_select_db($connection,"tnpfeedback");
    if(isset($_POST['vamsi']))
    {
      $comname=$_POST['search'];
      $query="SELECT * FROM feedback where companyname='$comname'";
      $quesry_run=mysqli_query($connection,$query);
      while($row=mysqli_fetch_array($quesry_run))
      {
        ?>
        <tr>
          <td> <?php echo $row['companyname']; ?></td>
          <td> <?php echo $row['rounds']; ?></td>
          <td> <?php echo $row['type']; ?></td>
          <td> <?php echo $row['details']; ?></td>
        </tr>
      <?php
    }
  }
?>
```

```
</table>

</div>

</div>

</div>

<p>
    <form action="redirect1.php" method='POST'>
        <CENTER><input type="submit" name="send" value="GO BACK"
style="font-size:18px; " /></CENTER>
    </form></p>

    <p>
        <form action="redirect2.php" method='POST'>
            <CENTER><input type="submit" name="send" value="BACK TO MAIN"
style="font-size:18px; " /></CENTER>
        </form></p>

</body>
</html>
```

Admin login.php :

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body
        {
            background: #ffffff url("bg.jpg") no-repeat center;
            background-size: cover;
            background-position: center;
        }
        h1.a
```

```
        {
            font-size: 60px;
            color: Crimson;
            text-align: center;
        }
h1.b
{
    font-size: 45px;
    color: SlateBlue;
    text-align: center;
}
h1.c
{
    font-size: 40px;
    color:Sienna ;
    text-align: center;
}
#fl
{
    font-size: 25px;
    border-width: 1px;
    padding: 50px 1px 20px 10px;
}
.fl_label
{
    white-space: nowrap;
}
</style>
<title>ADMIN LOGIN</title>
</head>
```

```
<body background="bg.jpeg">
    <h1 class='a'>GPREC</h1>
    <h1 class='b'>TRAINING & PLACEMENT CLUB</h1>
    <h1 class='c'>ADMIN LOGIN</h1>
    <center> <div>
    <form name="f1" method="post" action="loginlast.php" id="f1">
        <table>
            <tr>
                <td class="f1_label">USER NAME :</td>
                <td><input type="text" name="user" value="" />
                </td>
            </tr>
            <tr>
                <td class="f1_label">PASSWORD :</td>
                <td><input type="password" name="pass" value="" />
                </td>
            </tr>
            <tr>
                <td>
                    <CENTER><input type="submit" name="ADD" value="LOGIN" style="font-
size:18px; " /></CENTER>
                </td>
            </tr>
        </table>
    </form>
    <p>
        <form action="redirect2.php" method='POST'>
            <CENTER><input type="submit" name="send" value="BACK TO MAIN"
style="font-size:18px; " /></CENTER>
        </form></p>
```

```
</div></center>
</body>
</html>
```

Admin connect .php :

```
<?php
$conn = mysqli_connect("localhost", "root", "", "adminlogin") or die ("Failed to connect");
$username = $password = "";
$username = $_POST['user'];
$password = $_POST['pass'];
$sql = "SELECT * FROM login WHERE username='$username' AND password='$password'";
$result = mysqli_query($conn, $sql);
if(mysqli_num_rows($result) > 0)
{
    while($row = mysqli_fetch_assoc($result))
    {
        $user = $row["username"];
        $pass = $row["password"];
        session_start();
        $_SESSION['user'] = $user;
        $_SESSION['pass'] = $pass;

        header("location: front.html");
    }
}
else
{
    echo "Invalid username or password";
}
?>
```

Add.html :

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body
    {
      background: #ffffff url("bg.jpg") no-repeat center;
      background-size: cover;
      background-position: center;
    }
    h1.a
    {
      font-size: 60px;
      color: Crimson;
      text-align: center;
    }
    h1.b
    {
      font-size: 45px;
      color: SlateBlue;
      text-align: center;
    }
    h1.c
    {
      font-size: 40px;
      color: Sienna ;
      text-align: center;
    }
  }
```

```
#f1
{
    font-size: 25px;
    border-width: 1px;
    padding: 50px 1px 20px 10px;
}
.fl_label
{
    white-space: nowrap;
}

</style>
<title>ADD STUDENT</title>
</head>
<body background="bg.jpeg">
    <h1 class='a'>GPREC</h1>
    <h1 class='b'>TRAINING & PLACEMENT CLUB</h1>
    <h1 class='c'>ADD STUDENT</h1>
    <center> <div>
    <form name="f1" method="post" action="addconnect.php" id="f1">
        <table>
            <tr>
                <td class="fl_label">NAME :</td>
                <td><input type="text" name="name" value="" required/>
                </td>
            </tr>
            <tr>
                <td class="fl_label">ROLL NUMBER :</td>
                <td><input type="text" name="roll" value="" required/>
                </td>
            </tr>
        </table>
    </div>
    </body>
```



```
<tr>
  <td class="f1_label">BRANCH :</td>
  <td>
    <select name="branch" required>
      <option name="branch">CSE</option>
      <option name="branch">ECE</option>
      <option name="branch">MECH</option>
      <option name="branch">EEE</option>
      <option name="branch">CIVIL</option>
    </select>
  </td>
</tr>
<tr>
  <td class="f1_label">MAIL ID :</td>
  <td><input type="text" name="mail" value="" required/>
</td>
</tr>
<tr>
  <td class="f1_label">PHONE NUMBER :</td>
  <td><input type="text" name="mobile" value="" />
</td>
</tr>
<tr>
  <td class="f1_label">CGPA :</td>
  <td><input type="text" name="cgpa" value="" required/>
</td>
</tr>
<tr>
  <td>
```

```
<CENTER><input type="submit" name="ADD" value="ADD STUDENT"
style="font-size:18px; " /></CENTER>

</td>

</tr>

</table>

</form>

<p>

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

<CENTER><input type="submit" name="send" value="GO BACK" style="font-
size:18px; " /></CENTER>

</form></p>

</div></center>

</body>

</html>
```

Addconnect.php :

```
<?php
//xampp server details
$servername = "localhost";
$username = "root";
$password = "";
//create connection
$conn = new mysqli($servername, $username, $password) or die("Not connected");
if($conn)
{
    echo "connected";
}
if(!mysqli_select_db($conn,'ADMIN'))
{
    echo "no databse fund";
}
```

```
}  
//reading data through html form  
$name=$_POST['name'];  
$roll=$_POST['roll'];  
$branch=$_POST['branch'];  
$mail=$_POST['mail'];  
$mobile=$_POST['mobile'];  
$cgpa=$_POST['cgpa'];  
$company=$_POST['company'];  
//inserting data into database  
$sql="INSERT INTO  
data(NAME,ROLL_NUMBER,BRANCH,MAIL_ID,PHONE_NUMBER,CGPA,COMPANY_  
NAME) values('$name','$roll','$branch','$mail','$mobile','$cgpa','$company')";  
if(!mysqli_query($conn,$sql))  
{  
    $message = "STUDENT IS NOT ADDED TO DATABASE";  
    echo "<script type='text/javascript'>alert('$message');</script>";  
    echo "<br>";  
}  
else  
{  
    $message = "1 STUDENT ADDED TO DATABASE";  
    echo "<script type='text/javascript'>alert('$message');</script>";  
}  
header("Location:gohome.html");  
?>
```

Update.html :

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body
    {
      background: #ffffff url("bg.jpg");
      background-size: cover;
      background-position: center;
    }
    h1.a
    {
      font-size: 60px;
      color: Crimson;
      text-align: center;
    }
    h1.b
    {
      font-size: 45px;
      color: SlateBlue;
      text-align: center;
    }
    h1.c
    {
      font-size: 40px;
      color: Sienna ;
      text-align: center;
    }
  }
```

```
#f1
{
    border-width: 1px;
    font-size: 25px;
    padding: 50px 1px 20px 10px;
}
.fl_label
{
    white-space: nowrap;
}
td.a
{
    text-align: center;
}
</style>
<title>UPDATE STUDENT</title>
</head>
<body>
    <h1 class='a'>GPREC</h1>
    <h1 class='b'>TRAINING & PLACEMENT CLUB</h1>
    <h1 class='c'>UPDATE STUDENT DETAILS</h1>
    <center>
        <form name="f1" method="post" action="updateconnect.php" id="f1">
            <table>
                <tr>
                    <td class="fl_label">ROLL NUMBER :</td>
                    <td><input type="text" name="roll" value="" />
                </td>
            </tr>
        </table>
    </center>
</body>
```

```
<tr>
    <td class="f1_label">COMPANY NAME :</td>
    <td><input type="text" name="company" value="" />
</td>
</tr>
<tr>
    <td class='a'>
        <input type="submit" name="update" value="UPDATE" style="font-
size:18px; " />
    </td>
</tr>
</table>
</form>
<p>
    <form action="redirect.php" method="POST">
        <CENTER><input type="submit" name="send" value="GO BACK" style="font-
size:18px; " /></CENTER>
    </form></p>
</center>
</body>
</html>
```

Updateconnect.php :

```
<?php
//xampp server details
$servername = "localhost";
$username = "root";
$password = "";
//create connection
$conn = new mysqli($servername, $username, $password) or die("Not connected");
```

```
if(!mysqli_select_db($conn,'ADMIN'))
{
    echo "no databse fund";
}
//reading data through html form
$roll=$_POST['roll'];
$company=$_POST['company'];
//inserting data into database
$sql="UPDATE data set COMPANY_NAME= '$company' WHERE
ROLL_NUMBER='$roll'";
if(!mysqli_query($conn,$sql))
{
    $message = "DATA IS NOT UPDATED";
    echo "<script type='text/javascript'>alert('$message');</script>";
    echo "<br>";
}
else
{
    $message = "1 STUDENT DATA UPDATE SUCCESSFULL";
    echo "<script type='text/javascript'>alert('$message');</script>";
}
header("Location:gohome.html");
?>
```

Delete.html :

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body
        {
```

```
        background: #ffffff url("bg.jpg");
        background-size: cover;
        background-position: center;
    }
    h1.a
    {
        font-size: 60px;
        color: Crimson;
        text-align: center;
    }
    h1.b
    {
        font-size: 45px;
        color: SlateBlue;
        text-align: center;
    }
    h1.c
    {
        font-size: 40px;
        color:Sienna ;
        text-align: center;
    }
    #fl
    {
        border-width: 1px;
        font-size: 25px;
        padding: 50px 1px 20px 10px;
    }
    .fl_label
    {
```



```
        white-space: nowrap;
    }
    td.a
    {
        text-align: center;
    }
</style>
<title>UPDATE STUDENT</title>
</head>
<body>
    <h1 class='a'>GPREC</h1>
    <h1 class='b'>TRAINING & PLACEMENT CLUB</h1>
    <h1 class='c'>UPDATE STUDENT DETAILS</h1>
    <center>
    <form name="f1" method="post" action="deleteconnect.php" id="f1">
        <table>
            <tr>
                <td class="f1_label">RESPONSE ID :</td>
                <td><input type="text" name="roll" value="" />
            </td>
            </tr>
            <tr>
                <td class='a'>
                    <input type="submit" name="update" value="DELETE" style="font-size:18px;
" />
                </td>
            </tr>
        </table>
    </form>
    <p>
```

```
<form action="redirect.php" method="POST">
    <CENTER><input type="submit" name="send" value="GO BACK" style="font-
size:18px; " /></CENTER>
</form></p>
</center>
</body>
</html>
```

Deleteconnect.php :

```
<?php
//xampp server details
$servername = "localhost";
$username = "root";
$password = "";
//create connection
$conn = new mysqli($servername, $username, $password) or die("Not connected");
if(!mysqli_select_db($conn,'tnpfeedback'))
{
    echo "no databse fund";
}
//reading data through html form
$roll=$_POST['roll'];
//inserting data into database
$sql="DELETE FROM feedback WHERE id='$roll'";
if(!mysqli_query($conn,$sql))
{
    $message = "DATA IS NOT UPDATED";
    echo "<script type='text/javascript'>alert('$message');</script>";
    echo "<br>";
}
```

```
else
{
    $message = "1 STUDENT DATA UPDATE SUCCESSFULL";
    echo "<script type='text/javascript'>alert('$message');</script>";
}
header("Location:gohome.html");
?>
```

Printfeedback.php :

```
<!DOCTYPE html>
<html lang = "en-US">
<head>
<meta charset = "UTF-8">
<title>contact.php</title>
<style type = "text/css">
    table, th, td {border: 1px solid black};
</style>
</head>
<body bgcolor="cyan">
<p>
<?php
    try {
        $con= new PDO('mysql:host=localhost;dbname=tnpfeedback', "root", "");
        $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $query = "SELECT * FROM feedback";
        //first pass just gets the column names
        print "<table> ";
        $result = $con->query($query);
        //return only the first row (we only need field names)
```

```
$row = $result->fetch(PDO::FETCH_ASSOC);
print " <tr> ";
foreach ($row as $field => $value){
    print " <th>$field</th> ";
} // end foreach
print " </tr> ";
//second query gets the data
$data = $con->query($query);
$data->setFetchMode(PDO::FETCH_ASSOC);
foreach($data as $row){
    print " <tr> ";
    foreach ($row as $name=>$value){
        print " <td>$value</td> ";
    } // end field loop
    print " </tr> ";
} // end record loop
print "</table> ";
} catch(PDOException $e) {
    echo 'ERROR: ' . $e->getMessage();
} // end try
?>
</p>
<form action="redirect.php" method="POST">
    <input type="submit" name="send" value="Go Home">
</form>
</body>
</html>
```

Printstudetn.php :

```
<!DOCTYPE html>
<html lang = "en-US">
<head>
<meta charset = "UTF-8">
<title>contact.php</title>
<style type = "text/css">
    table, th, td {border: 1px solid black};
</style>
</head>
<body bgcolor="cyan">
<p>
<?php
    try {
        $con= new PDO('mysql:host=localhost;dbname=ADMIN', "root", "");
        $con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $query = "SELECT * FROM data";
        //first pass just gets the column names
        print "<table> ";
        $result = $con->query($query);
        //return only the first row (we only need field names)
        $row = $result->fetch(PDO::FETCH_ASSOC);
        print " <tr> ";
        foreach ($row as $field => $value){
            print " <th>$field</th> ";
        } // end foreach
        print " </tr> ";
        //second query gets the data
        $data = $con->query($query);
```

```
$data->setFetchMode(PDO::FETCH_ASSOC);
foreach($data as $row){
    print " <tr> ";
    foreach ($row as $name=>$value){
        print " <td>$value</td> ";
    } // end field loop
    print " </tr> ";
} // end record loop
print "</table> ";
} catch(PDOException $e) {
    echo 'ERROR: ' . $e->getMessage();
} // end try
?>
</p>
<center><form action="redirect.php" method="POST">
    <input type="submit" name="send" value="Go Home">
</form></center>
</body>
</html>
```

Gohome.html :

```
<!DOCTYPE html>
<html>
<head>
    <title>thanks</title>
</head>
<body>
    <form method="POST" action="redirect.php">
        <input type="submit" value="Go Back" name="send">
    </form>
```

```
</body>
```

```
</html>
```

Redirect.php :

```
<?php
session_start();
if(isset($_POST['send']))
{
    session_destroy();
    echo '<script>>window.location.href="../ADMIN/front.html"</script>';
}
?>
```

Redirect1.php :

```
<?php
session_start();
if(isset($_POST['send']))
{
    session_destroy();
    echo '<script>>window.location.href="../ADMIN/adminlogin.php"</script>';
}
?>
```

Redirect2.php :

```
<?php
session_start();
if(isset($_POST['send']))
{
    session_destroy();
    echo '<script>>window.location.href="../main.php"</script>';
}
?>
```

3.5 TESTING

Introduction :

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

Software testing may be viewed as a sub-field of Software Quality Assurance but typically exists independently (and there may be no SQA areas in some companies). In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the code or deliver faster.

Regardless of the methods used or level of formality involved the desired result of testing is a level of confidence in the software so that the organization is confident that the software has an acceptable defect rate. What constitutes an acceptable defect rate depends on the nature of the software. An arcade video game designed to simulate flying an airplane would presumably have a much higher tolerance for defects than software used to control an actual airliner.

A problem with software testing is that the number of defects in a software product can be very large, and the number of configurations of the product larger still. Bugs that occur infrequently are difficult to find in testing. A rule of thumb is that a system that is expected to function without faults for a certain length of time must have already been tested for at least that length of time. This has severe consequences for projects to write long-lived reliable software.

A common practice of software testing is that it is performed by an independent group of testers after the functionality is developed but before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays. Another practice is to start software testing at the same moment the project starts and it is a continuous process until the project finishes.

Another common practice is for test suites to be developed during technical support escalation procedures. Such tests are then maintained in regression testing suites to ensure that future updates to the software don't repeat any of the known mistakes.

Definition :

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behavior of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following routine procedure. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are operations the tester attempts to execute with the product, and the product answers with its behavior in reaction to the probing of the tester[citation needed]. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product—putting the product through its paces. Some of the common quality attributes include capability, reliability, efficiency, portability, maintainability, compatibility and usability. A good test is

sometimes described as one which reveals an error; however, more recent thinking suggests that a good test is one which reveals information of interest to someone who matters within the project community.

3.5.1 Testing Methodologies

Black Box Testing :

It is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

White Box Testing :

It is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.

3.5.2 Typed of Testing :

- Regression Testing.
- Re-Testing.
- Static Testing.
- Dynamic Testing.
- Alpha Testing.
- Beta Testing.
- Monkey Testing
- Compatibility Testing.
- Installation Testing.

🌱 > **Regression Testing:** It is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.

‡ **Re-Testing:** It is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.

‡ **Static Testing:** It is the testing, which is performed on an application when it is not been executed.

Ex: GUI, Document Testing

‡ **Dynamic Testing:** It is the testing which is performed on an application when it is being executed.

Ex: Functional testing.

‡ **Alpha Testing:** It is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.

‡ **Beta-Testing:** It is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.

‡ **Monkey Testing:** It is the process in which abnormal operations, beyond capacity operations are done on the application to check the stability of it in spite of the users abnormal behavior.

‡ **Compatibility testing:** It is the testing process in which usually the products are tested on the environments with different combinations of databases (application servers, browsersetc.) In order to check how far the product is compatible with all these environments platform combination.

‡ **Installation Testing:** It is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

SNAPSHOTS OF PROJECT

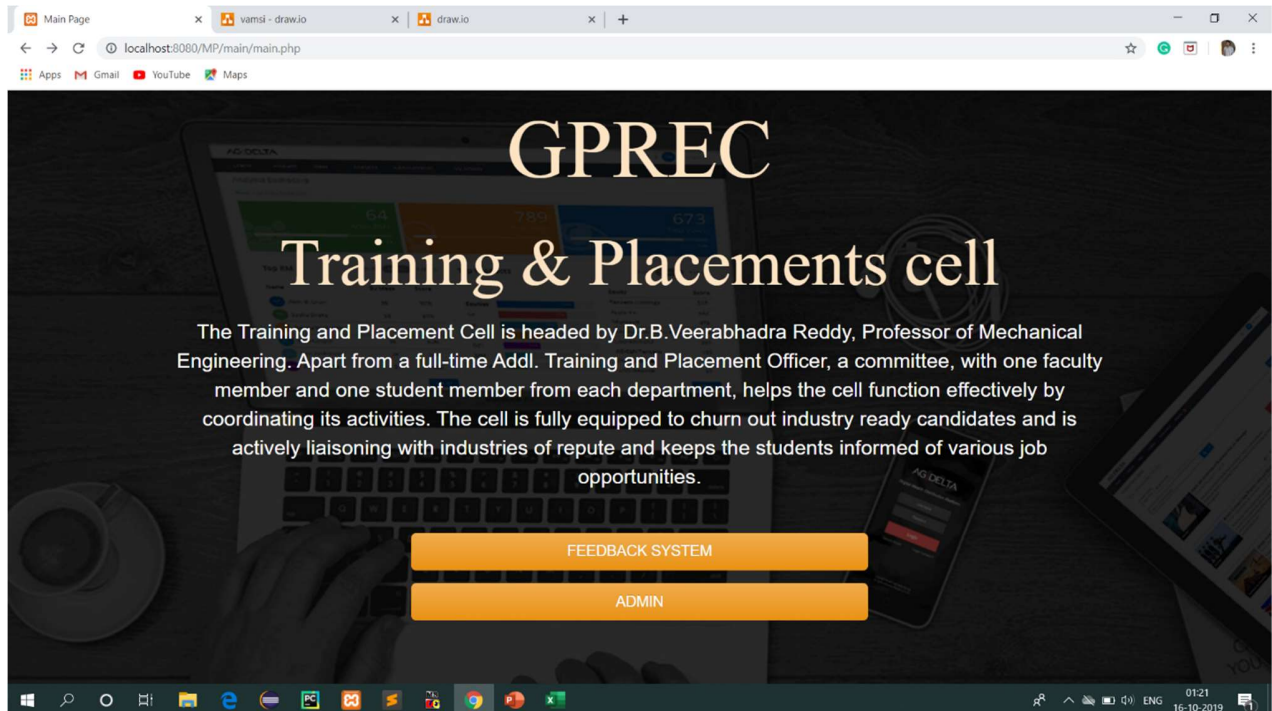
**FIGURE 3.6.1 : HOMEPAGE**

Figure 3.6.1 Shows the homepage of the web application for Training and Placement Cell

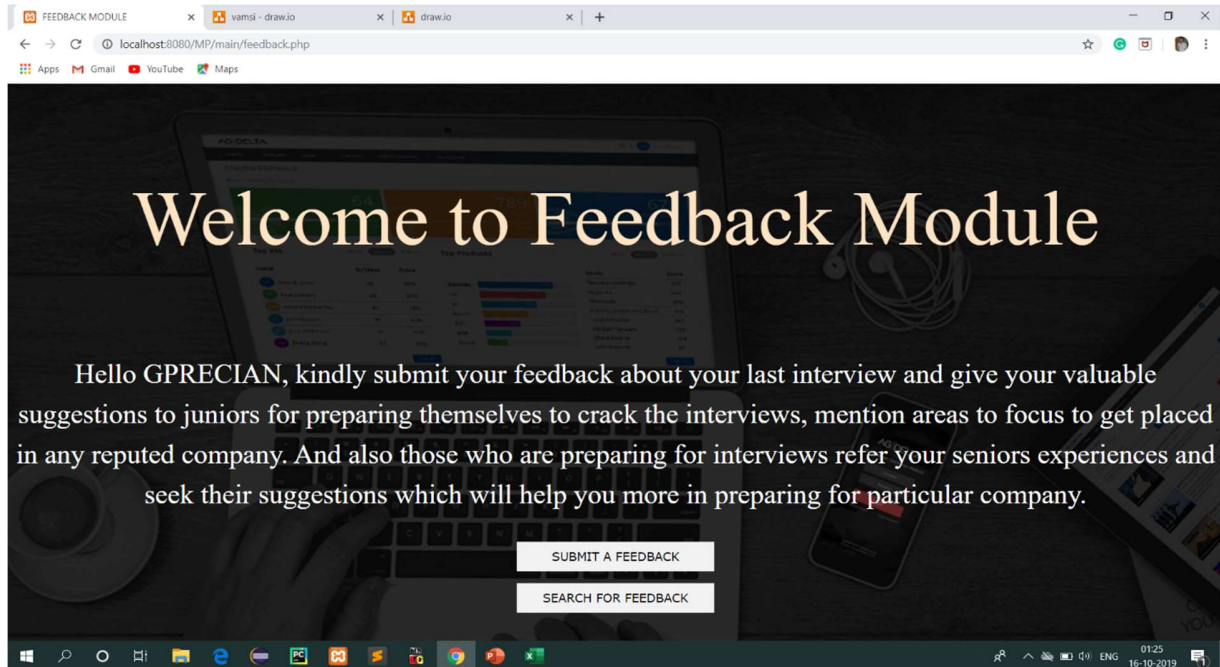
**FIGURE 3.6.2 : FEEDBACK MODULE PAGE**

Figure 3.6.2 Shows the home page of feedback module in this user need to choose their option it may either submitting the submit or searching feedback

Feedback Form

Please provide your feedback below:

Company Name :

Total no. of Rounds :

Type of Company :
☐ Service Based ☐ Product Based

Share your interview experience and suggestions to juniors:
 Your Comments

SUBMIT FEEDBACK

FIGURE 3.6.3 : FEEDBACK SUBMIT PAGE

Figure 3.6.3 Shows the feedback submit page which Alumni or placed student submit their interview feedback

Feedback Search

Search with company Name :

SUBMIT

| Company Name | No. of rounds include in recruitment | Type of Company | Feedback Report |
|--------------|--------------------------------------|-----------------|-----------------|
| | | | |

GO BACK

BACK TO MAIN

FIGURE 3.6.4 : FEEDBACK SEARCH PAGE

Figure 3.6.4 Shows the information about the particular company on user or students choice



FIGURE 3.6.5 : ADMIN LOGIN PAGE

Figure 3.6.5 To access admin operations it needs login credentials

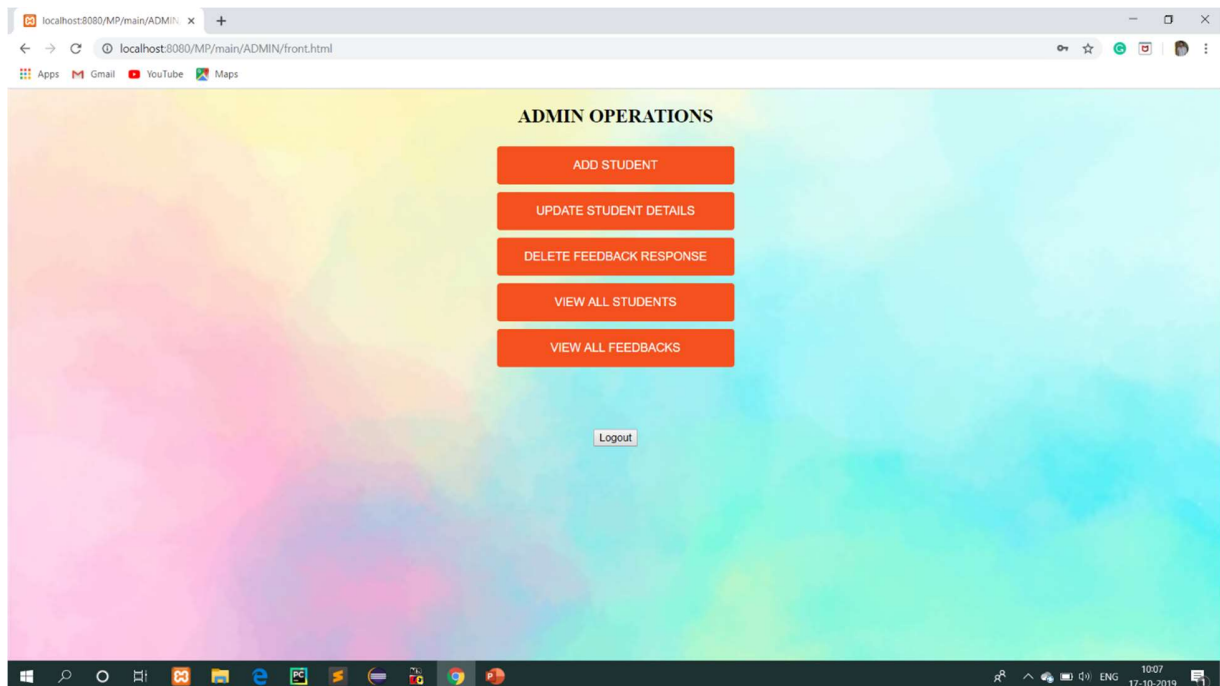


FIGURE 3.6.6 : ADMIN OPERATIONS PAGE

Figure 3.6.6 It displays all the operations of admin.

ADD STUDENT

localhost:8080 / 127.0.0.1 / tnpgf

localhost:8080/MP/main/ADMIN/add.html

GPREC

TRAINING & PLACEMENT CLUB

ADD STUDENT

NAME :

ROLL NUMBER :

BRANCH :

MAIL ID :

PHONE NUMBER :

CGPA :

ADD STUDENT

GO BACK

FIGURE 3.6.7 : ADDING STUDENT PAGE

Figure 3.6.7 Shows the student registrations which is submitted by admin

UPDATE STUDENT

localhost:8080 / 127.0.0.1 / tnpgf

localhost:8080/MP/main/ADMIN/update.html

GPREC

TRAINING & PLACEMENT CLUB

UPDATE STUDENT DETAILS

ROLL NUMBER :

COMPANY NAME :

UPDATE

GO BACK

FIGURE 3.6.8 : UPDATE STUDENT DETAILS

Figure 3.6.8 Shows the student details update form like if particular student got placed in any company it needs to be updated



FIGURE 3.6.9 : DELETE FEEDBACK RESPONSE

Figure 3.6.9 This page contains information about the deleting feedback responses due to the inappropriate content in feedback



FIGURE 3.6.10 : VIEW ALL STUDENT DETAILS

Figure 3.6.10 Shows the all students details which were stored in database

| id | companyname | rounds | type | details |
|----|-------------|--------|---------|---|
| 29 | CTS | 3 | Service | You can find it by seeing the previous amcat paper that was almost similar and in case they allowed to take test at home then have a good network wifi would be best. |
| 30 | EZE | 6 | Service | It includes total of six rounds: 1. Written Test (Coding), 2. GD 3. TR-1 4. Tr-2 5. MR 6. HR Prepare Data structures well for TR rounds. And in MR round they will check your problem-solving skills for certain problem. And HR is common but be genuine. |
| 31 | TCS | 3 | Service | Once you clear the written, you almost had your job in the pocket. Basic C questions will be asked in TR and HR is all about relocation, night shift for our batch and "if" you are asked to sign and don't hesitate to sign it. It won't be a problem signing. |
| 32 | WIPRO | 4 | Service | For me, In TR they asked about my project, internship and they checked about my c language knowledge. They asked me to write any program of my own and asked to explain it. In HR they just asked me whether i was ok with relocation, night shifts. Thats it. |
| 33 | EPAM | 6 | Service | Coding questions are really tough. |
| 34 | MINDTREE | 4 | Service | "TR: Questions based on your resume. In my interview they asked about c, Java, program using structures. And some basic questions in core subjects and project based questions. HR: Mainly they check your communication, confidence, and they way you express your opinion. In my interview they asked to describe recently watched movie and how was your day? what are your hobbies? Aim, short term goals and long term goals. Basic HR questions." |
| 35 | MPHASIS | 4 | Service | In Technical, they asked about java python as I mentioned in my resume. All questions are asked just to test your basic knowledge. Oops concepts, program in Java and python. In HR they asked about my SD and basic ques. |

[Go Home](#)

FIGURE 3.6.11 : VIEW ALL FEEDBACKS

Figure 3.6.11 Displays all the feedback responses given by students and Alumni

CONCLUSION

4.1 CONCLUSION

- ✓ Feedback, more specifically from apt Alumni undeniably is one of the most important element in meticulously planning any improvisation of an organization.
- ✓ As aimed, our project creates an effective, wide range and one stop platform for Alumni working in corporations to give their valued feedback and also provides a trove of impeccable database for the administrator to enable the aspiring students to contemplate and plan their career options. Also, the idea was to substantiate personal time investments which might bring out memorable and long lasting outputs in near future.

BIBLIOGRAPHY

BOOKS

- 1.** Developing PHP Applications for IBM Data Servers
- 2.** Programming PHP
- 3.** PHP 5 Power Programming
- 4.** Practical PHP Programming
- 5.** HTML Black Book

WRITERS

Whei-Jen Chen, Holger Kirstein, Daniel Krook,
Kiran H Nair, Piotr Pietrzak

Rasmus Lerdorf, Kevin Tatroe

Andi Gutmans, Stig Bakken,
Derick Rethans

Paul Hudson

Steven Holzner

REFERENCE LINKS

1. www.w3schools.com/php/
2. www.php.net/
3. en.wikipedia.org/wiki/PHP
4. www.planet-php.net/
5. <https://www.w3schools.com/html/>
6. https://www.w3schools.com/css/css_intro.asp
7. <https://www.apachefriends.org/index.html>
8. <https://www.sublimetext.com>
9. <https://stackoverflow.com>