Student Admission System

Table of Contents

[Abstract 3](#_Toc51082913)

[Introduction 4](#_Toc51082914)

[System Specification 4](#_Toc51082915)

[Design Methodology 5](#_Toc51082916)

[DATABASE DESIGN 6](#_Toc51082917)

[TABLES 7](#_Toc51082918)

[Work Flow 10](#_Toc51082919)

[Coding 11](#_Toc51082920)

[CODE EFFICIENCY 12](#_Toc51082921)

[TECHNOLOGY SPECIFICATIONS 13](#_Toc51082922)

[IMPLEMENTATION AND MAINTENANCE 15](#_Toc51082923)

[TESTING 16](#_Toc51082924)

[FUTURE SCOPE 19](#_Toc51082925)

[CONCLUSION 19](#_Toc51082926)

[BIBLIOGRAPHY 20](#_Toc51082927)

[APPENDICES 21](#_Toc51082928)

# Abstract

Traditional college admission is a hectic process, which involves students visiting off-site campus, taking application, filling it and then submission is another hectic story. 0n the day of admission, the flow of candidates is very high and it requires both manual processing and record keeping at the same time that makes the process lengthy and difficult to keep track of the admission status of a candidate in multiple departments. At present admission process is done manually with pen and paper which is very inefficient and utilizes much efforts and time. This college admission management system helps to make the admission process much easier and helps in maintaining database in an efficient way. In this system college admin can add the college details and the stream details. College can create the cut off list for the current year and the students are expected to register on the website and apply for the desired stream. College can register admissions of new students and also remove the students who denies the admission. College can make 3 list of cut off and the final list of students can be viewed. There are 2 modules in this system namely admin (college) and user (students). Admin can add college details and cut off list, they can select students, register and remove students, and can also view the final list of students. Users (Students) can login and register to the respective streams where the admin (college) select the students based on their criteria that were followed by the college.

# Introduction

**Student Admission System** is aimed at developing an online admission application for a college. This system is an online system that can be accessed throughout the organization and outside as well with proper login provided. Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to admit and monitor the whole process. When a user log in to the system. He would only view details of the student. He can't perform any changes .The system has two modules. They are

* User
* Administrator

Students logging is to apply for the course by filling an application form provided by online. College principal/administrator logging in may also access/search information put up by the students.

# System Specification

1. **Hardware Configuration:**

* System :  Multimedia PC
* Processor :  Pentium 4 or above
* Memory :  512MB RAM
* Hard  Disk :  80GB or above
* Keyboard :  104 standards
* Monitor : SVGA
* Modem : Dial up/Broadband

1. **Software Configuration:**

* Front end              : PHP, HTML, CSS, JavaScript
* Back end : MYSQL
* Operating System : Windows ,LINUX

# Design Methodology

System design is the solution to the creation of a new system. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design.

During logical design phase the analyst describes inputs (sources), out puts (destinations), databases (data sores) and procedures (data flows) all in a format that meats the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design.

The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

# DATABASE DESIGN

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. DBMS allow data to be protected and organized separately from other resources. Database is an integrated collection of data. The most significant form of data as seen by the programmers is data as stored on the direct access storage devices. This is the difference between logical and physical data.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives: -

* Data integration.
* Data integrity.
* Data independence.

The proposed system stores the information relevant for processing in the MS SQL SERVER database. This database contains tables, where each table corresponds to one particular type of information. Each piece of information in table is called a field or column. A table also contains records, which is a set of fields. All records in a table have the same set of fields with different information. There are primary key fields that uniquely identify a record in a table. There are also fields that contain primary key from another table called foreign keys.

# TABLES

**1. PERSONAL DETAILS**

|  |  |
| --- | --- |
| **FIELDNAME** | **DATATYPE** |
| regid | Varchar() |
| name | Varchar() |
| father | Varchar() |
| age | Integer |
| gender | Varchar() |
| religion | Varchar() |
| caste | Varchar() |
| day | Integer |
| Month | Integer |
| year | Integer |

**2. LOGIN**

|  |  |
| --- | --- |
| **FIELDNAME** | **DATATYPE** |
| Regid | Varchar() |
| Password | Varchar() |
| Type | Varchar() |
| Flag | Integer |

**3.  EXAM**

|  |  |
| --- | --- |
| **FIELDNAME** | **DATATYPE** |
| Regid | Varchar() |
| Eqxam | Varchar() |
| Rollno | Integer |
| Passyear | Integer |
| Mark | Integer |
| Total | Integer |
| Percent | Integer |

**4. COURSE**

|  |  |
| --- | --- |
| **FIELDNAME** | **DATATYPE** |
| regid | Varchar() |
| course | Varchar() |

**5. ADDRESS**

|  |  |
| --- | --- |
| **FIELDNAME** | **DATATYPE** |
| regid | Varchar() |
| house | Varchar() |
| post | Varchar() |
| street | Varchar() |
| city | Varchar() |
| pin | Integer |
| Email-id | Varchar() |

# Work Flow

1. **Dataflow Diagram**

Level 0

Username + Password

Level 1

New Student Details

Old Student Details

Username + Password

Student details

Level 2

# Coding

A code is an ordered collection symbols to provide unique identification of data. Codes can be used by people who do not with data processing; the following are characters of a good code generation. Characteristics of good coding are

* Uniqueness
* Meaningfulness
* Stability
* Uniform Size and Format
* Simplicity
* Conciseness
* Versatility

             The goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design. The coding phase affects both testing and maintenance profoundly.   As we saw earlier, the time spent in coding is a small percentage of the total software cost , while testing and maintenance consume the major percentage. Thus it should be clear the goal during coding should not be to reduce the implementation cost, but the goal should be to reduce the cost of later phases, even if it means that the cost of this phase has to increase. In other words, the goal during this phase is not to simplify the job of the programmer. Rather the goal should be to simplify the job of the tester and the maintainer.

# CODE EFFICIENCY

**1.CORRECTNESS**

The tester and the maintainer are not the programmers itself. Therefore it is very important to produce correct and simple code. The code for the proposed system is correct and simple. Anyone who knows HTML and PHP can understand the code very easily.

**2. RELIABILITY**

The program must be reliable, that is, it must be execute reliably in a variety of systems.  The code for proposed system is consistent.

**3. ROBUSTNESS**

The robustness of the code is associated with memory management and exception handling. Since he proposed system is developed with HTML this feature is not ensured profoundly.

**4. DEVELOPMENT EFFICIENCY**

The PHP ensures development efficiency by providing compatibility with almost all web servers used today. Extending PHP is pretty easy. It is a full-fledged development environment consisting of a several megabytes of source code. PHP scripts are browser neutral, the script cannot be captured via a eb browser’s ’View Source’ capabilities.

**5. EXECUTION EFFICIENCY**

PHP is easy to learn and runs efficiently and quickly on any compatible web server. The PHP is a parsed language. The parser which execute PHP produce the desired result as HTML code. The code written for the proposed system consists of many PHP files which execute on the web server and return output as HTML to the browser.

# TECHNOLOGY SPECIFICATIONS

**1. PHP (Hyper text Preprocessor)**

PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

**2. My SQL**

My SQL query () sends a unique query (multiple queries are not supported) to the currently active database on the server that's associated with the specified *link\_identifier.* The query string should not end with a semicolon. If the link identifier is not specified, the last link opened by My SQL connection is assumed. If no such link is found, it will try to create one as if [mysql\_connect ()](about:blank) was called with no arguments. If by chance no connection is found or established, an E\_WARNING level warning is generated.

**OPTIMIZATION OF CODE**

Code optimization aims at improving execution efficiency of a program. This is achieved in  two ways:

* Redundancies in a program are eliminated.
* Computations in a program are rearranged or rewritten to make it execute efficiently.
* The optimization must not change the meaning of a program.

The ‘Online Admission system ‘ optimizes the code by using the optimization techniques such as dead code elimination and frequency reduction. Thus improves the execution efficiency.

**VALIDATION CHECKS**

Validation means observing the behavior of the system. The verification and validation means that will ensure that the output of a phase is consistent with its input and that the output of the phase is consistent with the overall requirements of the system. The ‘College Alumni’ system performed validation by verifying the output of each phase. This is done to ensure that it is consistent with the required output. If not we apply certain mechanisms for repairing and thereby achieved the requirement.

# IMPLEMENTATION AND MAINTENANCE

Implementation is an activity that is contained throughout the development phase. It is a process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successful system should be delivered and users should have confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation.

**1. IMPLEMENTATION PLANS**

The major activities in implementation plan are cost estimation, schedule and milestone determination, project staffing, quality control plans, and controlling and monitoring plans The implementation plan involves the following:

* Testing to confirm effectiveness.
* Detection and correction of errors.

The system has two modules. They are

* User
* Administrator

Students logging is to apply for the course by filling an applycation form provided by online. College principal/administrator logging in may also access/search information put up by the students.

**2. CHANGEOVER METHOD**

If the implementation is changed from one data structure to another, the internals of the object need to be changed, that is the data definitions and the implementation of the operations. From the outside, the directory object can continue to be used in the same manner as before, because its interface is not changed.

# TESTING

Software testing is a critical element of the software development cycle. The testing is essential for ensuring the Quality of the software developed and represents the ultimate view of specification, design and code generation. Software testing is defined as the process by which one detects the defects in the software. Testing is a set of activities that work towards the integration of entire computer based system.

A good test case is one that has a high probability of finding an as-yet undiscovered error. A successful test is one such uncovers or finds such errors. If testing is conducted successfully, it will uncover errors in the software. It also demonstrates that software functions are being performed according to specifications and also behavioral and performance requirements are satisfied. For this, test plans have to be prepared. The implementation of a computer system requires that test data has to be prepared and that all the elements in the system are tested in a planned and efficient manner. Nothing is complete without testing, as it is vital success of the system.

**1.TESTING OBJECTIVES**

There are several rules that can serve as testing objectives. They are:

* Testing is process of executing a program and finding a bug.

A good test case is one that has a high probability of finding an undiscovered.

* A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met.

**2. UNIT TESTING**

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

This is the first level of testing. In this, codes are written such that from one module, we can move on to the next module according to the choice we enter.

**3. SYSTEM TESTING**

In this, the entire system  was tested as a whole with all forms, code, modules and class modules.System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions. System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

**4. INTEGRATION TESTING**

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed.

Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time, conducting test to uncover errors associated with the interface. In the testing the programs are contructed and tested in the small segments.

**5. WHITE BOX TESTING**

White-box testing is concerned with testing the implementation of the program.

The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program. White box testing is also called structuraltesting.

To test the structure of a program, structural testing aims to achieve test cases that will force the desired coverage of different structures. Various criteria have been proposed for this.

There are three different approaches to structural testing: control flow-based testing, data flow-based testing, and mutation testing.

**6. BLACK BOX TESTING**

In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the internals of the module or the program are not considered for selection of test cases.

In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. This form of testing is also called functional or behavioural testing.

The most obvious functional testing procedure is exhaustive testing. One criterion for generating test cases is to generate them randomly. There are no formal rules for designing test cases for functional testing. In fact, there are no precise criteria for selecting test cases.

# FUTURE SCOPE

The future scope of this project is very broad

Few of them are:

* This can be implemented in less time for proper admission process
* This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
* The user had not need to travel a  long distance for the admission and his/her time is also saved as a result of this automated system

# CONCLUSION

This system, being the first We have created in PHP, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself.

Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three

must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first.

Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability.

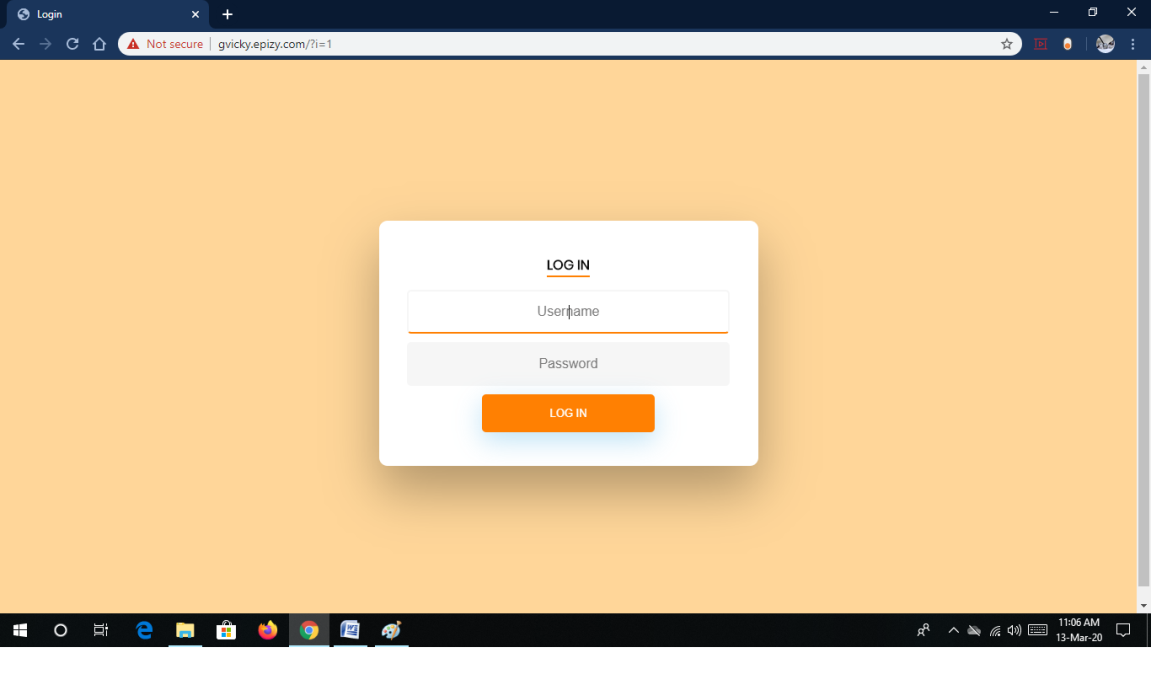
Overall, the system performs well, and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

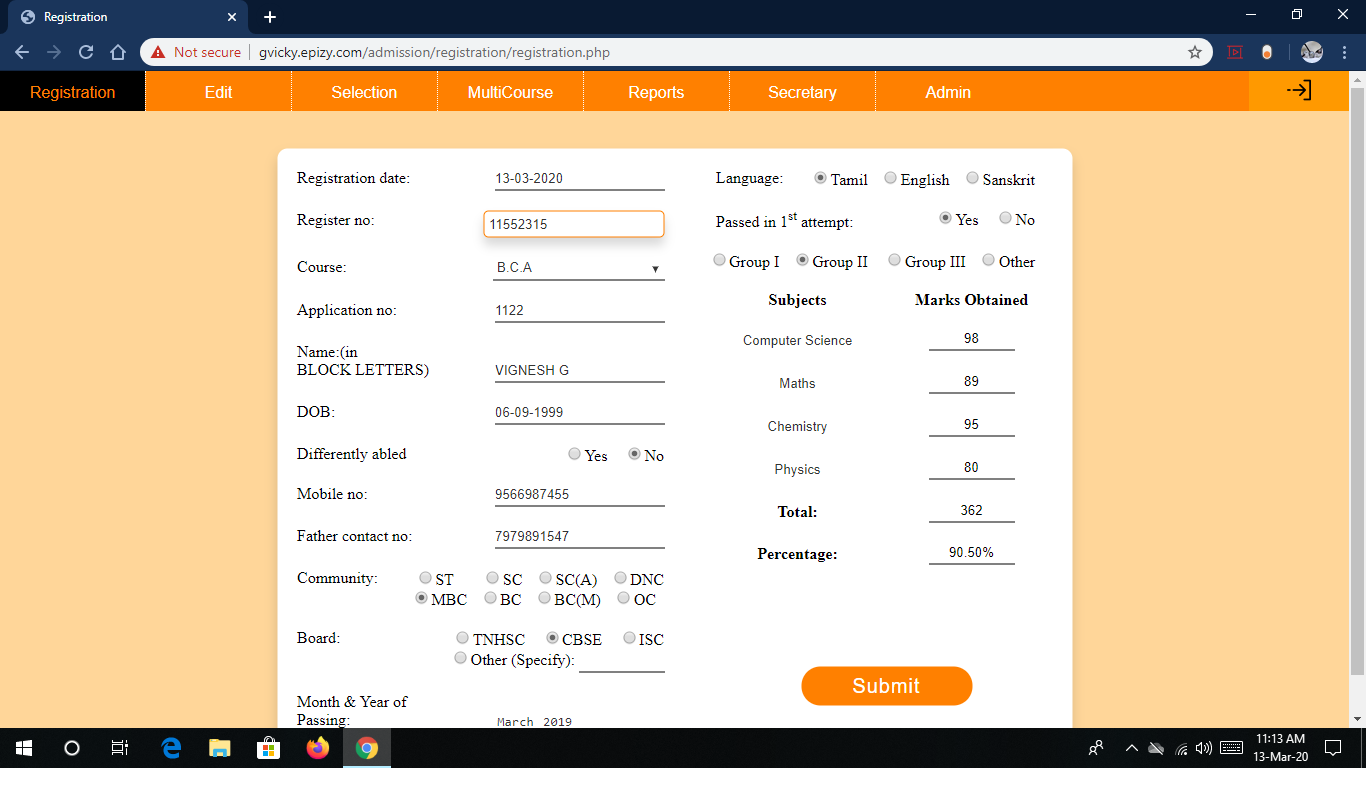
# BIBLIOGRAPHY

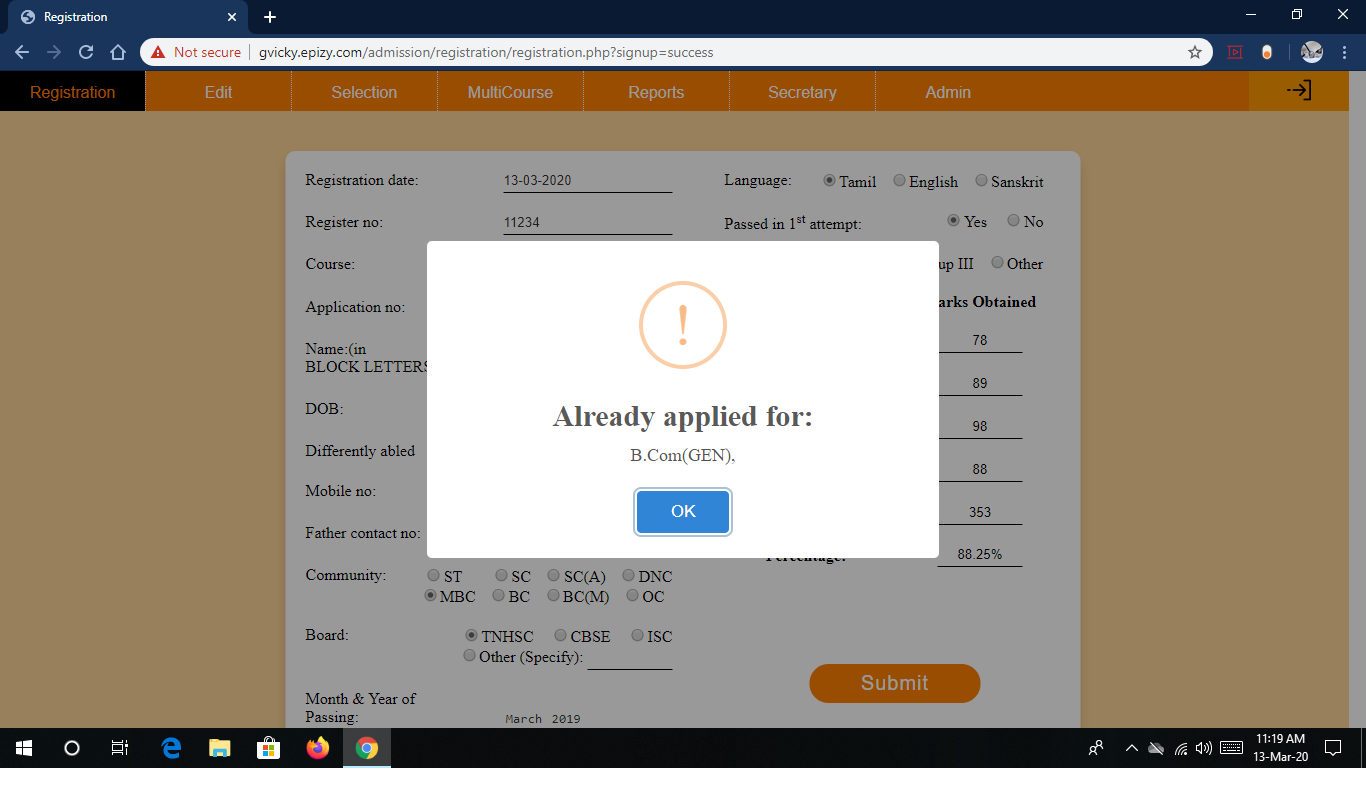
* An Integrated approach to Software Engineering,Narosa publishers, Third edition by Pankaj Jalote.
* The PHP Manual-www.php.net.
* PHP Manual-www.w3school.com.
* Secure PHP Development-Building 50 Practical Applications by Mohammed J Kabir,Wiley Publishing,inc.
* PHP for Absolute Beginners by Jason Lengstorf

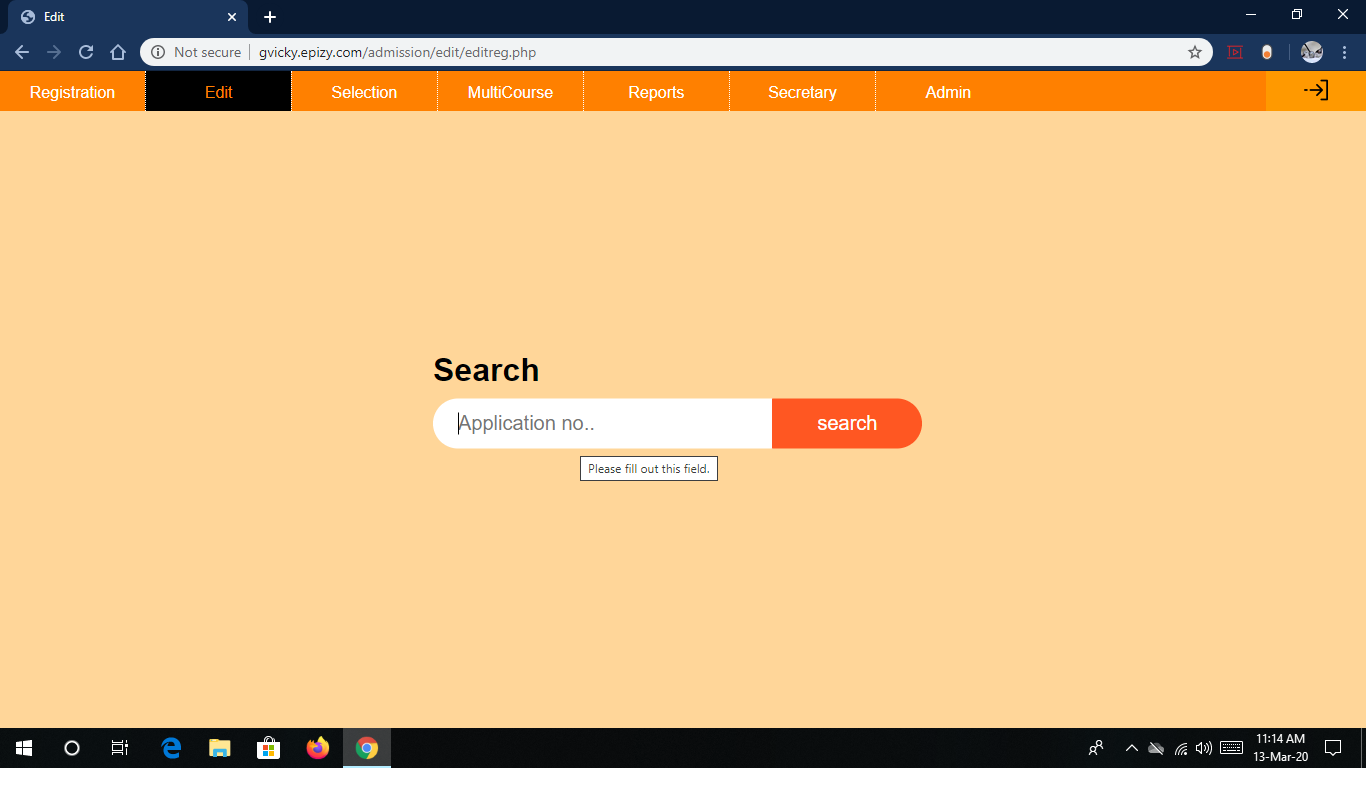
# APPENDICES

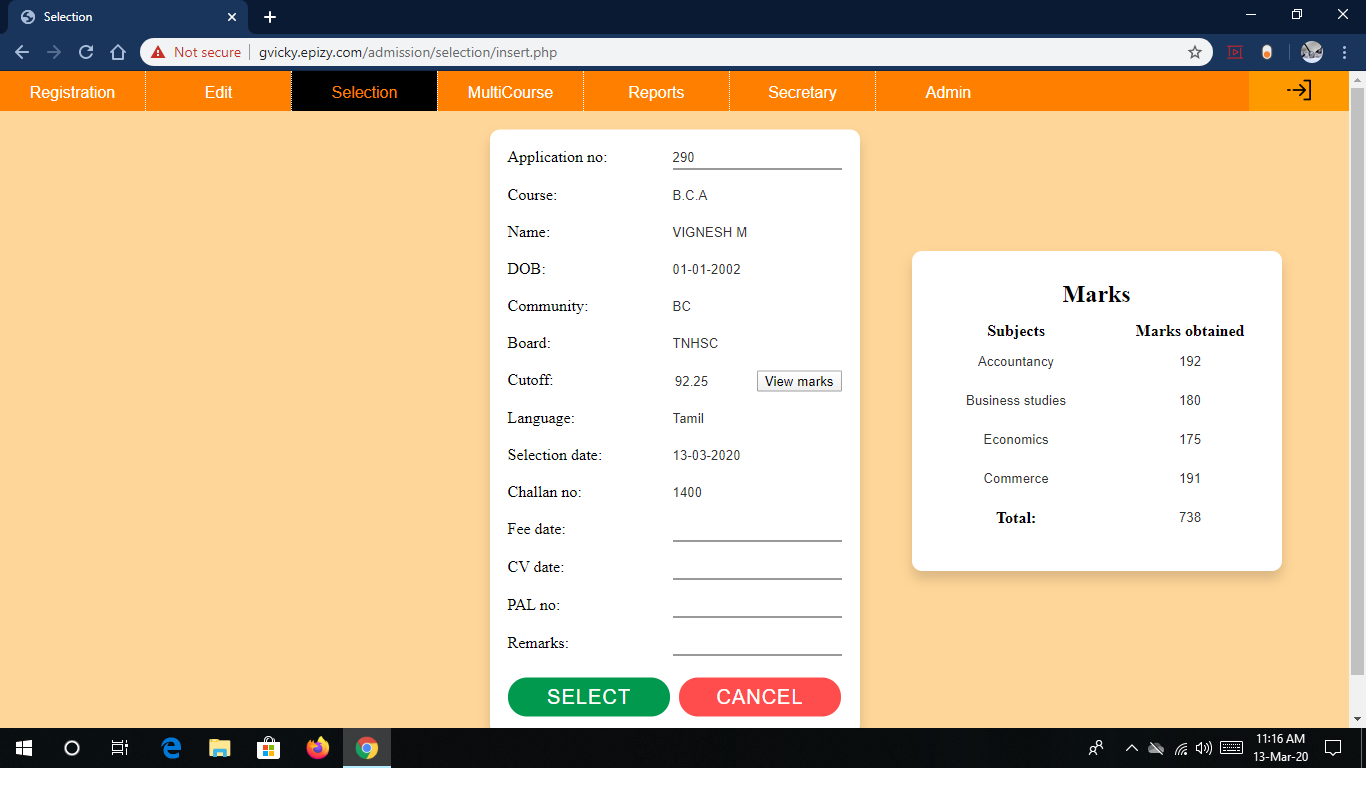
**SCREENSHOTS**

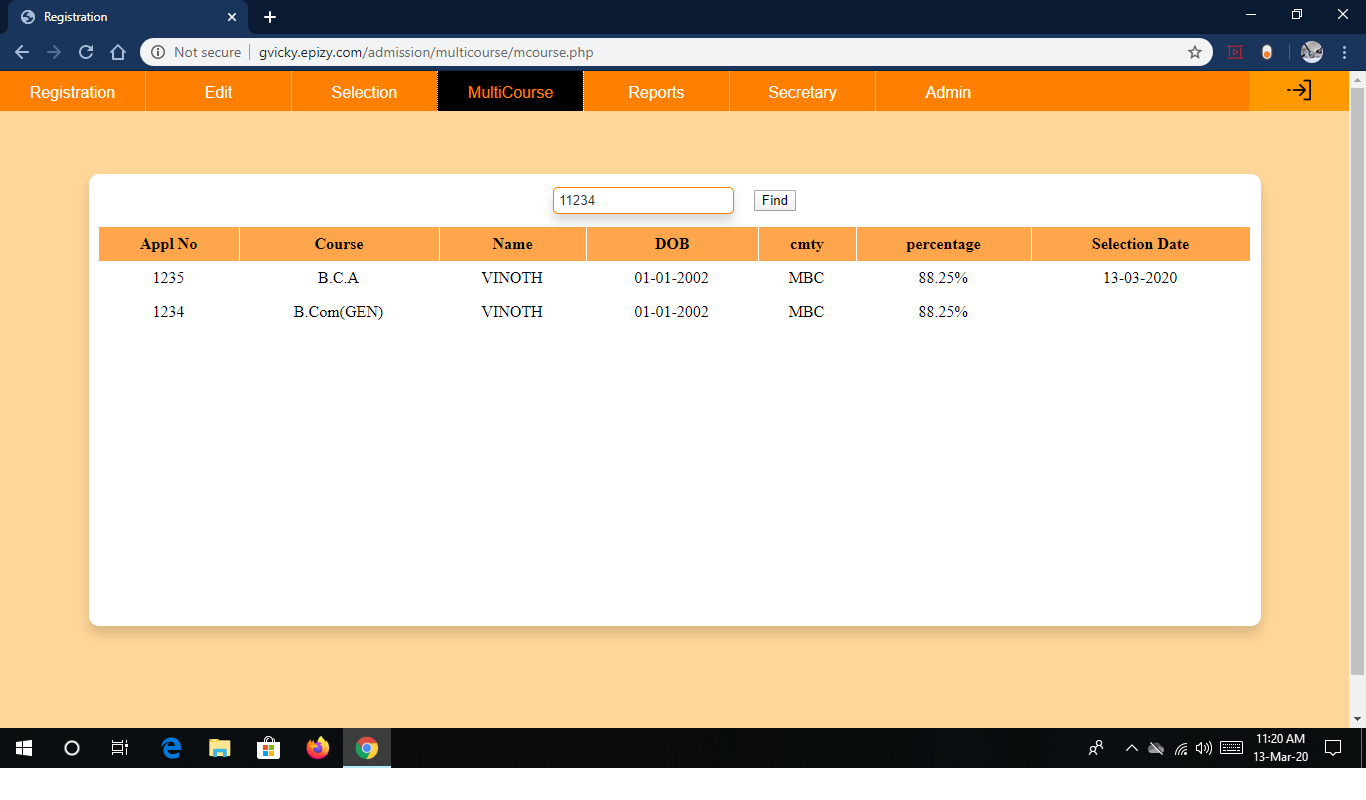


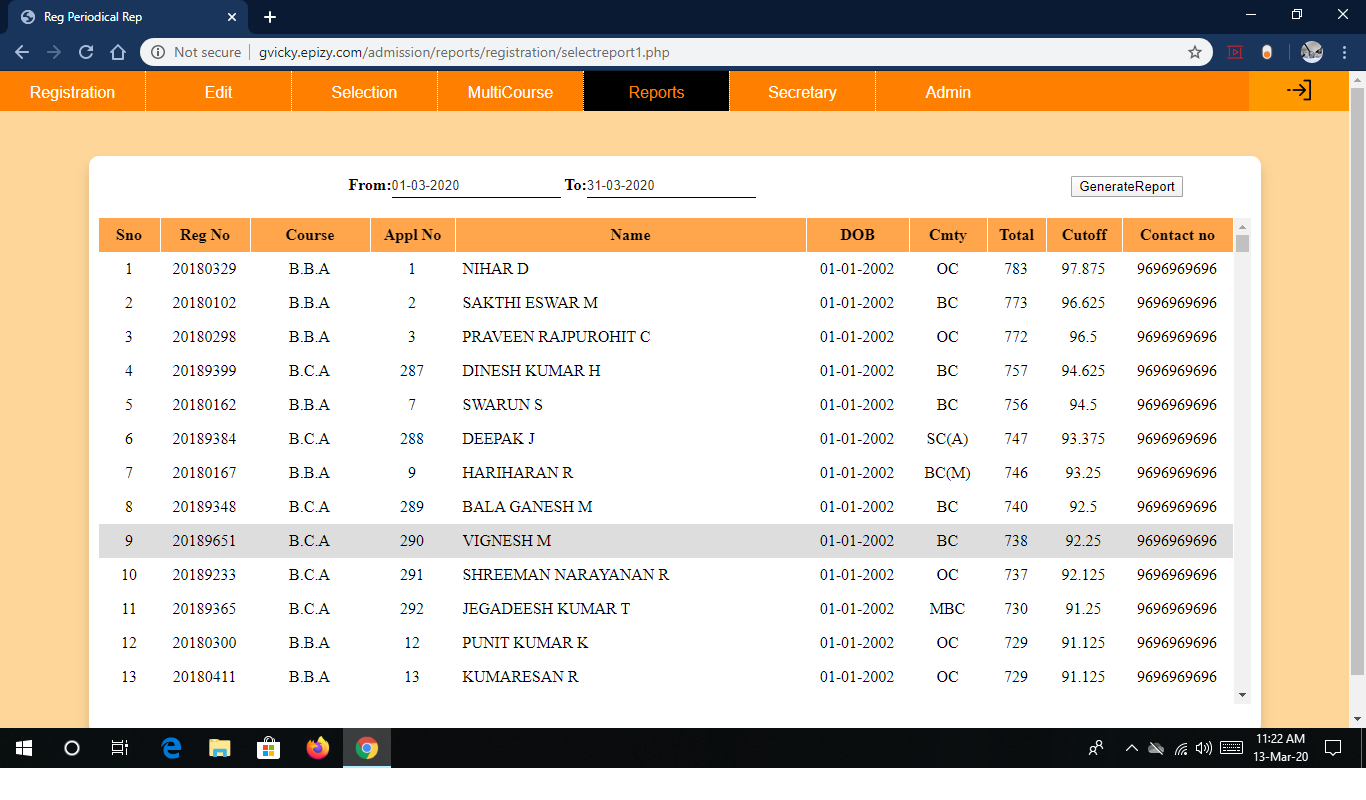


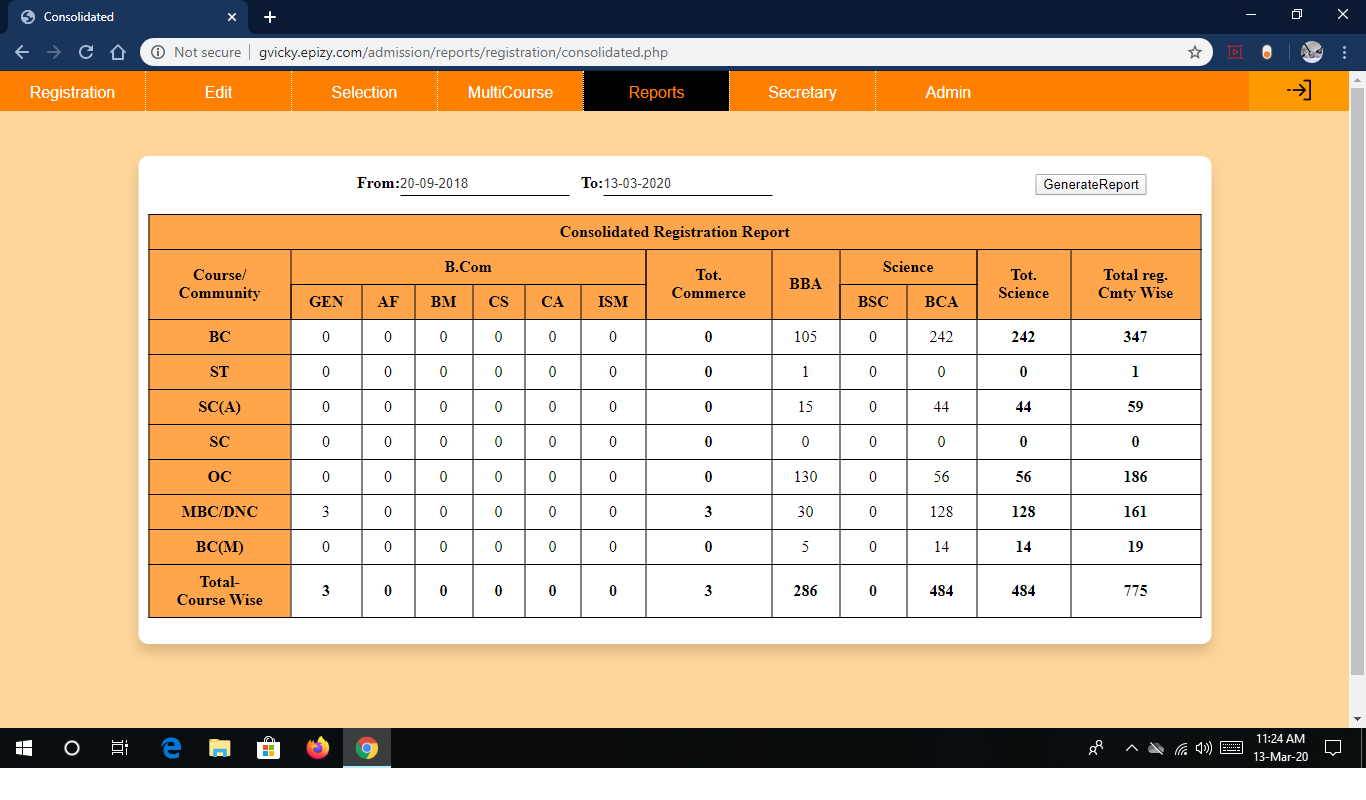


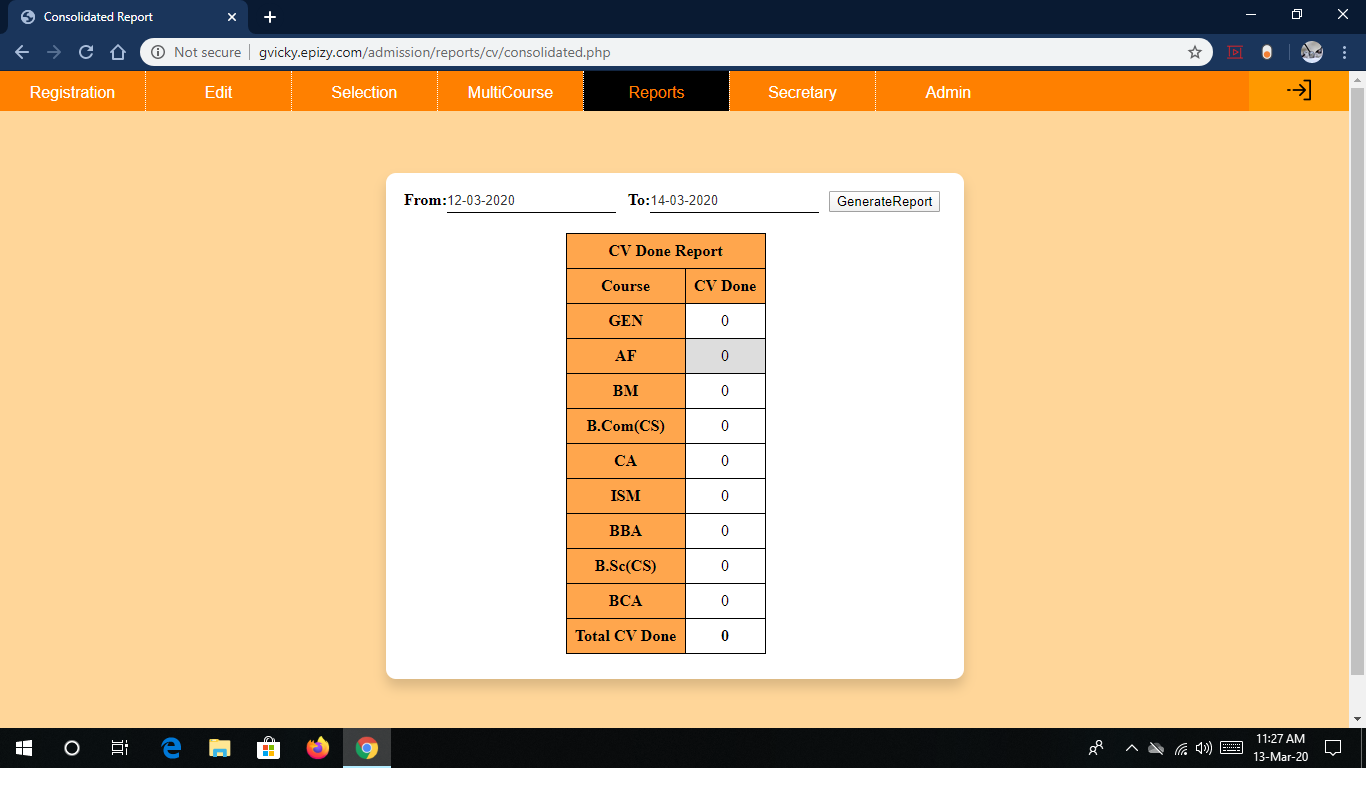


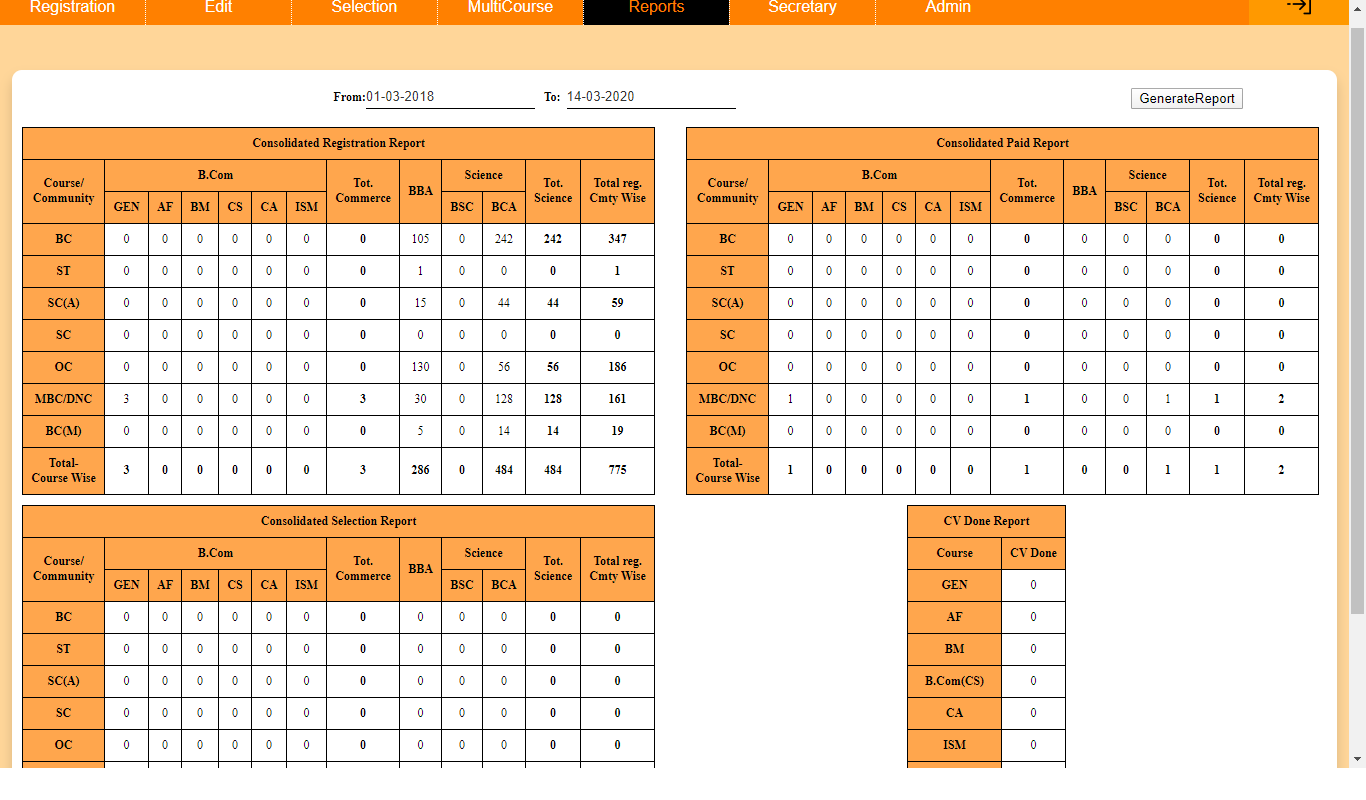


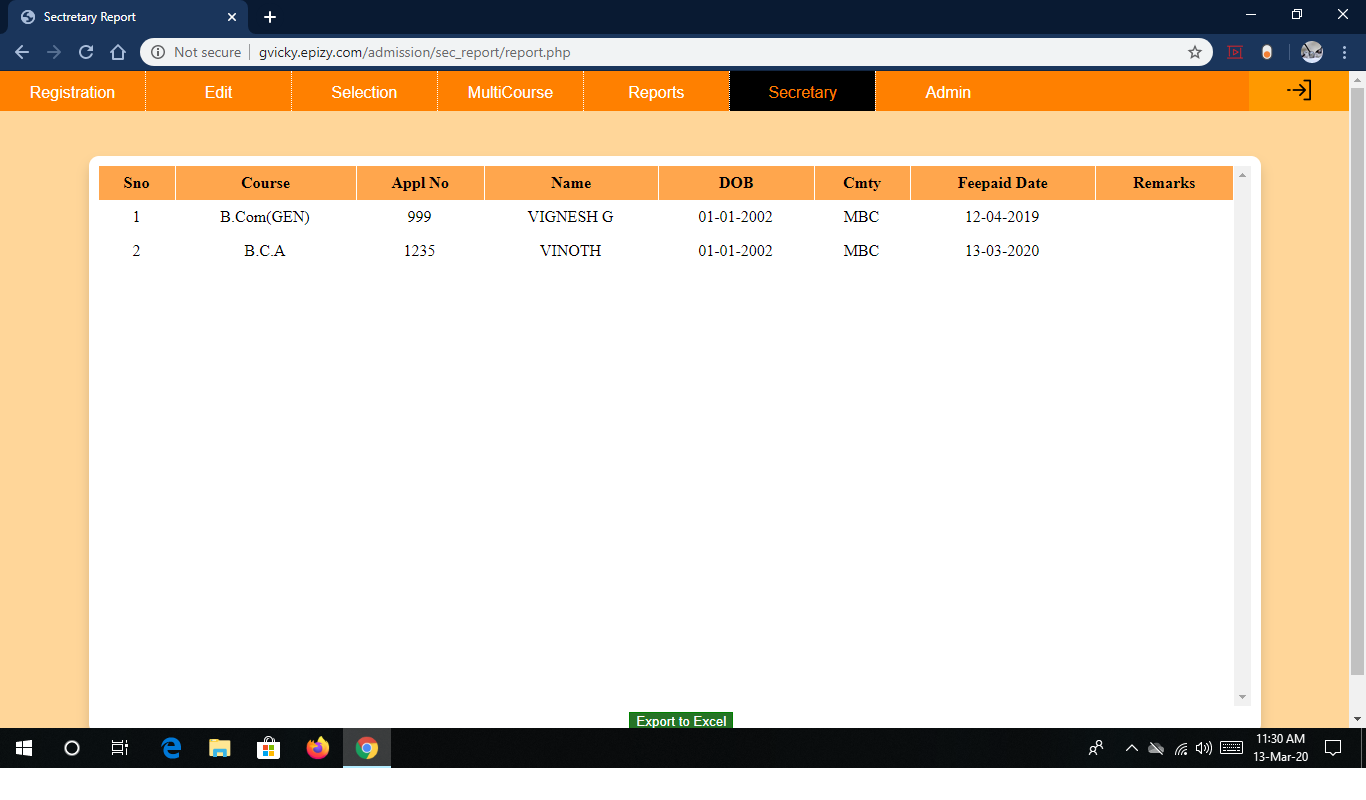














**Code**

*Filename: Registration.php*

<?php

session\_start();

if(!isset($\_SESSION['user'])){

header("Location: ../../index.php");

exit();

}

include("../database/dbconnection.php");

$conn=new mysqli($servername,$username,$password,$db);

if($conn->connect\_error){

die("Connection error:".$conn->connect\_error);

exit();

}

$sql="SELECT \* FROM `master`";

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

if($result->num\_rows>0){

while($row=$result->fetch\_assoc()){

$tmax = $row["tmax"];

$tmin = $row["tmin"];

$cmax = $row["cmax"];

$cmin = $row["cmin"];

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang='en' dir='ltr'>

<head>

<link rel='stylesheet' type='text/css' href='regstyle.css' />

<script src="../sweetalert2/sweetalert2.all.min.js"></script>

<link rel="stylesheet" href="../jquery/jquery-ui-1.12.1/jquery-ui.css" />

<script type="text/javascript" src="../jquery/jquery-ui-1.12.1/external/jquery/jquery.js"></script>

<script type="text/javascript" src="../jquery/jquery-ui-1.12.1/jquery-ui.js"></script>

<meta charset='UTF-8' />

<title>Registration</title>

</head>

<body onload="setdate();">

<?php

if($\_SESSION['user'] == "user"){

echo '<div class="main">

<ul class="mainnav">

<li class="active"><a href="registration.php">Registration</a></li>

<li><a href="../edit/editreg.php">Edit</a></li>

<li class="right"><a href="../logout/logout.php"><img src="../logout.png"></a></li>

</ul>

<br style="clear: both;">

</div>';

}elseif($\_SESSION['user'] == "admin"){

echo '<div class="main">

<ul class="mainnav">

<li class="active"><a href="registration.php">Registration</a></li>

<li><a href="../edit/editreg.php">Edit</a></li>

<li><a href="../selection/insert.php">Selection</a></li>

<li><a href="../multicourse/mcourse.php">MultiCourse</a></li>

<li class="hassubs"><a href="javascript:void(0)">Reports</a>

</ul>

</li>

<li class="subs hassubs"><a href="javascript:void(0)">Selection</a>

<ul class="dropdown">

<li class="subs sub2"><a href="../reports/selection/report1.php">Branch</a></li>

<li class="subs sub2"><a href="../reports/selection/report2.php">Branch\_Community</a></li>

<li class="subs sub2"><a href="../reports/selection/report3.php">Branch&ensp;Mark</a></li>

<li class="subs sub2"><a href="../reports/selection/consolidated.php">Consolidated</a></li>

</ul>

</li>

<li class="subs hassubs"><a href="javascript:void(0)">Fee&ensp;Paid</a>

<ul class="dropdown">

</li>

<li class="hassubs"><a href="javascript:void (0)">Secretary</a>

<ul class="dropdown">

<li class="subs"><a href="../sec\_report/report.php">Report</a></li>

<li class="subs"><a href="../sec\_report/dailyrep.php">Today&ensp;Rep</a></li>

</ul>

</li>

<li><a href="../admin/page.php">Admin</a></li>

<li class="right"><a href="../logout/logout.php"><img src="../logout.png"></a></li>

</ul>

<br style="clear: both;">

</div>';

}elseif($\_SESSION['user'] == "incharge"){

header("Location: ../selection/insert.php");

}elseif($\_SESSION["user"] == "secretary"){

header("Location: ../sec\_report/report.php");

}elseif($\_SESSION["user"] == "cvuser"){

header("Location: ../reports/cv/cvrep.php");

}

?>

<div>

<script type="text/javascript">

var $zoho = $zoho || {};

$zoho.salesiq = $zoho.salesiq || {

widgetcode: "d2c83bbfa695ee1f84536a6d6bf3a0f396112f5499033fd70c6d36f88d1e01af1a2010ab7b6727677d37b27582c0e9c4",

values: {},

ready: function() {}

};

var d = document;

s = d.createElement("script");

s.type = "text/javascript";

s.id = "zsiqscript";

s.defer = true;

s.src = "https://salesiq.zoho.com/widget";

t = d.getElementsByTagName("script")[0];

t.parentNode.insertBefore(s, t);

d.write("<div id='zsiqwidget'></div>");

</script>

</div>

<div class='container'>

<form class='myform' autocomplete='off' onsubmit="return validate()" action='regphp.php' method='post' name='myform' id="myform">

<div class='box1'>

<table class='containertable'>

<tr>

<td>

<label for='regdate'>Registration date:</label>

<div class='fields'>

<input type="text" id="mydate" name="regdate" required>

</div>

</td>

</tr>

<tr>

<td><label for='regno'>Register no:</label>

<div class='fields'>

<input type='text' maxlength="9" name='regno' required onblur="find()" autofocus>

</div>

</td>

</tr>

<tr>

<td> <label for='course'>Course:</label>

<div class='fields'>

<select name="course" required>

<option value="B.Com(GEN)">B.Com(GEN)</option>

<option value="B.Com(CS)">B.Com(CS)</option>

<option value="B.Com(A&F)">B.Com(A&amp;F)</option>

<option value="B.Com(BM)">B.Com(BM)</option>

<option value="B.Com(CA)">B.Com(CA)</option>

<option value="B.Com(ISM)">B.Com(ISM)</option>

<option value="B.B.A">B.B.A</option>

<option value="B.Sc.,(CS)">B.Sc.,(CS)</option>

<option value="B.C.A">B.C.A</option>

</select>

</div>

</td>

</tr>

<tr>

<td>

<label for='applno'>Application no:</label>

<div class='fields'>

<input type='text' name='applno' maxlength="4" required id="applno">

</div>

</td>

</tr>

<tr>

<td>

<label for='regname'>Name:(in<br /> BLOCK LETTERS)</label>

<div class='fields'>

<input type='text' name='regname' required placeholder="(eg: VIGNESH G S)">

</div>

</td>

</tr>

<tr>

<td>

<label for='dob'>DOB:</label>

<div class='fields'>

<input type="text" value="01-01-2002" name="dob" required>

</div>

</td>

</tr>

<tr>

<td>

<label for='disability'>Differently abled</label>

<div class='fields'>

<input type='radio' name='disability' value='yes' required>Yes &ensp;

<input type='radio' name='disability' value='no' checked>No

</div>

</td>

</tr>

<tr>

<td>

<label for='mobileno'>Mobile no:</label>

<div class='fields'>

<input type='text' name='mobileno' required maxlength='10'>

</div>

</td>

</tr>

<tr>

<td>

<label for='fatherno'>Father contact no:</label>

<div class='fields'>

<input type='text' name='fatherno' maxlength='10'>

</div>

</td>

</tr>

<tr>

<td>

<label for='community'>Community:</label>

<div class='fields'>

<input type='radio' name='community' id='community' value='ST'>ST&ensp;&ensp;&ensp;

<input type='radio' name='community' id='community' value='SC'>SC&ensp;

<input type='radio' name='community' id='community' required value='SC(A)'>SC(A)&ensp;

<input type='radio' name='community' id='community' value='DNC'>DNC

</div>

<br>

<div class='fields'>

<input type='radio' name='community' id='community' value='MBC'>MBC&ensp;

<input type='radio' name='community' id='community' value='BC'>BC&ensp;

<input type='radio' name='community' id='community' value='BC(M)'>BC(M)&ensp;

<input type='radio' name='community' id='community' value='OC' checked>OC&ensp;

</div>

</td>

</tr>

<tr>

<td>

<label for='board'>Board:</label>

<div class='fields'>

<input type='radio' name='board' value='TNHSC' onclick='setboarddetail()' required checked>TNHSC &ensp;

<input type='radio' name='board' value='CBSE' onclick='setboarddetail()'>CBSE &ensp;

<input type='radio' name='board' value='ISC' onclick='setboarddetail()'>ISC

</div>

<br>

<div class='fields'>

<input type='radio' name='board' value='other' onclick='setboarddetail()'>Other (Specify): <input type='text' name='boarddetail' size='8'>

</div>

</td>

</tr>

<tr>

<td>

<label for='yearofpassing'>Month &amp; Year of<br> Passing:</label>

<div class='fields'>

<input type='month' name='yearofpassing' value="2019-03" required>

</div>

</td>

</tr>

</table>

</div>

<div class='box2'>

<table>

<tr>

<td>

<label for="language">&ensp;&ensp;&ensp;&ensp;Language:</label>

<div class="fields">

<input type="radio" name="language" value="Tamil" required checked>Tamil&ensp;

<input type="radio" name="language" value="English" required>English&ensp;

<input type="radio" name="language" value="Sanskrit" required>Sanskrit&ensp;&ensp;

</div>

</td>

</tr>

<tr>

<td>

<label for='attempt'>&ensp;&ensp;&ensp;&ensp;Passed in 1<sup>st</sup> attempt:</label>

<div class='fields'>

<input type='radio' name='attempt' value='yes' checked>Yes &ensp;

<input type='radio' name='attempt' value='no' required>No&ensp;&ensp;

</div>

</td>

</tr>

<tr>

<td>

<div class='column2'>

<div class='group'>

&ensp;&ensp;&ensp;<input type='radio' name='group' value='group1' required checked onclick='groupSelection()'>Group I&ensp;

<input type='radio' name='group' value='group2' onclick='groupSelection()'>Group II

&ensp;

<input type='radio' name='group' value='group3' onclick='groupSelection()'>Group III&ensp;

<input type='radio' name='group' value='other' onclick='groupSelection()'>Other

</div>

<div class='marktable'>

<table>

<tr>

<th>

Subjects

</th>

<th>

Marks Obtained

</th>

</tr>

<tr>

<th>

<input type='text' name='subject1' id="subject1" value='Biology' readonly>

<input type="text" list="sub1" name="sub1" hidden autocomplete="on">

<datalist id="sub1">

<option value="Biology">

<option value="Maths">

<option value="Chemistry">

<option value="Computer Science">

<option value="Physics">

<option value="Accountancy">

<option value="Commerce">

<option value="Business Studies">

<option value="Economics">

</datalist>

</th>

<td>

<input type='text' name='mark1' size='8' maxlength='3' min="0" max="100" required onblur="gettotal(); getpercentage();">

</td>

</tr>

<tr>

<th>

<input type='text' name='subject2' value='Maths' readonly>

<input type="text" list="sub2" name="sub2" hidden autocomplete="on">

<datalist id="sub2">

<option value="Biology">

<option value="Maths">

<option value="Chemistry">

<option value="Computer Science">

<option value="Physics">

<option value="Accountancy">

<option value="Commerce">

<option value="Business Studies">

<option value="Economics">

</datalist>

</th>

<td>

<input type='text' name='mark2' size='8' maxlength='3' min="0" max="100" required onblur="gettotal(); getpercentage();">

</td>

</tr>

<tr>

<th>

<input type='text' name='subject3' value='Chemistry' readonly>

<input type="text" list="sub3" name="sub3" hidden autocomplete="on">

<datalist id="sub3">

<option value="Biology">

<option value="Maths">

<option value="Chemistry">

<option value="Computer Science">

<option value="Physics">

<option value="Accountancy">

<option value="Commerce">

<option value="Business Studies">

<option value="Economics">

</datalist>

</th>

<td>

<input type='text' name='mark3' size='8' maxlength='3' min="0" max="100" required onblur="gettotal(); getpercentage();">

</td>

</tr>

<tr>

<th>

<input type='text' name='subject4' value='Physics' readonly>

<input type="text" list="sub4" name="sub4" hidden autocomplete="on">

<datalist id="sub4">

<option value="Biology">

<option value="Maths">

<option value="Chemistry">

<option value="Computer Science">

<option value="Physics">

<option value="Accountancy">

<option value="Commerce">

<option value="Business Studies">

<option value="Economics">

</datalist>

</th>

<td>

<input type='text' name='mark4' size='8' maxlength='3' min="0" max="100" required onblur="gettotal(); getpercentage();">

</td>

</tr>

<tr>

<th>

<label for="total">Total:</label>

</th>

<td>

<input type='text' name='total' size='8' onclick="gettotal(); getpercentage();">

</td>

</tr>

<tr>

<th>

<label for="percentage">Percentage:</label>

</th>

<td>

<input type='text' name='percentage' size='8' onclick="gettotal(); getpercentage();">

</td>

</tr>

</table>

</div>

</div>

</td>

</tr>

</table>

<span id="success"></span>

</div>

<div class='submitbtn'>

<input type='submit' name='submit' value='Submit' class='submit'>

</div>

</form>

</div>

<script>

function setdate() {

var today = new Date();

var dd = today.getDate();

var mm = today.getMonth() + 1;

var yy = today.getFullYear();

if (dd < 10) {

dd = '0' + dd;

}

if (mm < 10) {

mm = '0' + mm;

}

today = dd + "-" + mm + "-" + yy;

document.forms["myform"]["regdate"].value = today;

}

var tmax = <?php echo $tmax;?>;

var tmin = <?php echo $tmin;?>;

var cmax = <?php echo $cmax;?>;

var cmin = <?php echo $cmin;?>;

function validate() {

var regname = document.forms["myform"]["regname"].value;

var regno = document.forms["myform"]["regno"].value;

var applno = document.forms["myform"]["applno"].value;

var sub3 = document.forms["myform"]["sub3"].value;

var sub4 = document.forms["myform"]["sub4"].value;

var numberpattern = /[^0-9]/;

var namepattern = /[^A-Z]$/i;

if (namepattern.test(regname)) {

callalert("Enter the Name in BLOCK LETTERS");

callfocus("regname");

return false;

}

if (/[^A-Z]$/i.test(boarddetail)) {

callalert("Enter the Board detail");

callfocus("boarddetail");

return false;

}

if (numberpattern.test(regno)) {

callalert("Enter the Register no.");

callfocus("regno");

return false;

}

if (numberpattern.test(fatherno)) {

callalert("Enter the father contact number");

callfocus("fatherno");

return false;

}

if (numberpattern.test(mark1)) {

callalert("Enter the mark in subject 1");

callfocus("mark1");

return false;

}

if (numberpattern.test(mark2)) {

callalert("Enter the mark mark in subject 2");

callfocus("mark2");

return false;

}

if (board != "other") {

if (board == "TNHSC") {

if ((mark1 > 0 && mark1 < tmin) || mark1 > tmax) {

callalert("Enter Mark between " + tmin + " - " + tmax);

callfocus("mark1");

return false;

}

if ((mark2 > 0 && mark2 < tmin) || mark2 > tmax) {

callalert("Enter Mark between " + tmin + " - " + tmax);

callfocus("mark2");

return false;

} }

if (board != "TNHSC") {

if ((mark1 > 0 && mark1 < cmin) || mark1 > cmax) {

callalert("Enter Mark between " + cmin + " - " + cmax);

callfocus("mark1");

return false;

}

if ((mark2 > 0 && mark2 < cmin) || mark2 > cmax) {

callalert("Enter Mark between " + cmin + " - " + cmax);

callfocus("mark2");

return false;

}

if ((mark3 > 0 && mark3 < cmin) || mark3 > cmax) {

callalert("Enter Mark between " + cmin + " - " + cmax);

callfocus("mark3");

return false;

}

}

}

if (/[^A-Z]$/i.test(sub1)) {

callalert("Enter the subject1");

callfocus("sub1");

return false;

}

if (/[^A-Z]$/i.test(sub2)) {

callalert("Enter the subject2");

callfocus("sub2");

return false;

}

}

function callalert(message) {

swal(message);

}

function callfocus(element) {

document.forms["myform"][element].focus();

document.forms["myform"][element].value = "";

}

function groupSelection() {

var group = document.forms["myform"]["group"].value;

if (group == "group1") {

document.forms["myform"]["sub1"].removeAttribute("required");

document.forms["myform"]["subject1"].setAttribute("readonly", "");

document.forms["myform"]["subject2"].setAttribute("readonly", "");

document.forms["myform"]["subject3"].setAttribute("readonly", "");

document.forms["myform"]["subject4"].setAttribute("readonly", "");

}

if (group == "group3") {

document.forms["myform"]["sub1"].removeAttribute("required");

document.forms["myform"]["sub2"].removeAttribute("required");

document.forms["myform"]["subject3"].setAttribute("readonly", "");

document.forms["myform"]["subject4"].setAttribute("readonly", "");

}

if (group == "other") {;

document.forms['myform']['sub2'].hidden = false;

document.forms['myform']['sub2'].style.borderBottom = "1px solid #333333";

document.forms["myform"]["sub2"].setAttribute("required", "");

}

}

function setboarddetail() {

if (document.forms["myform"]["board"].value == "other") {

callfocus("boarddetail");

document.forms["myform"]["boarddetail"].setAttribute("required", "");

} else {

document.forms["myform"]["boarddetail"].removeAttribute("required");

document.forms["myform"]["boarddetail"].value = "";

}

if (document.forms["myform"]["board"].value != "other") {

document.forms["myform"]["boarddetail"].removeAttribute("required");

document.forms["myform"]["boarddetail"].value = "";

}

}

function gettotal() {

var numberpattern = /[^0-9]/;

var total = 0;

if (numberpattern.test(mark1)) {

callalert("Enter the mark in subject 1");

callfocus("mark1");

}

if (numberpattern.test(mark2)) {

callalert("Enter the mark in subject 2");

callfocus("mark2");

}

if (numberpattern.test(mark3)) {

callalert("Enter the mark in subject 3");

callfocus("mark3");

}

if (numberpattern.test(mark4)) {

callalert("Enter the mark in subject 4");

callfocus("mark4");

}

} else {

mark3 = parseInt(mark3);

}

if (mark4 == "") {

mark4 = 0;

} else {

mark4 = parseInt(mark4);

}

total = mark1 + mark2 + mark3 + mark4;

if (isNaN(total)) {

total = "";

}

document.forms["myform"]["total"].value = total;

if (total == 0) {

document.forms["myform"]["percentage"].value = "0%";

}

}

function getpercentage() {

var total = 0;

var per = 0;

var x = 0;

var tper = 0;

var cper = 0;

var numberpattern = /[^0-9]/;

if (numberpattern.test(mark1)) {

callalert("Enter the mark in subject 1");

callfocus("mark1");

}

if (mark1 == "" || mark1 == 0) {

mark1 = 0;

} else {

mark1 = parseInt(mark1);

x++;

}

if (mark2 == "" || mark2 == 0) {

mark2 = 0;

} else {

mark2 = parseInt(mark2);

x++;

}

if (mark3 == "" || mark3 == 0) {

mark3 = 0;

} else {

mark3 = parseInt(mark3);

x++;

}

if (mark4 == "" || mark4 == 0) {

mark4 = 0;

} else {

mark4 = parseInt(mark4);

x++;

}

total = mark1 + mark2 + mark3 + mark4;

if (tmax == 200) {

tper = 8;

} else {

tper = 4;

}

if (document.forms["myform"]["board"].value == "TNHSC") {

per = total / tper;

} else {

per = total / x;

}

if (isNaN(per)) {

per = "";

}

document.forms["myform"]["percentage"].value = per.toFixed(2) + "%";

}

function find() {

var regno = document.forms["myform"]["regno"].value;

dbparam = JSON.stringify(regno);

var xmlhttp = new XMLHttpRequest();

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

myobj = JSON.parse(this.responseText);

if (myobj.name != "") {

setdate();

document.forms["myform"]["regno"].value = myobj.regno;

document.forms["myform"]["community"].value = myobj.community;

if (myobj.board != "TNHSC" && myobj.board != "CBSE" && myobj.board != "ISC") {

document.forms["myform"]["board"].value = "other";

document.forms["myform"]["boarddetail"].value = myobj.board;

} else {

document.forms["myform"]["board"].value = myobj.board;

}

document.forms["myform"]["yearofpassing"].value = myobj.monthofpassing;

document.forms["myform"]["language"].value = myobj.language;

document.forms["myform"]["attempt"].value = myobj.passedattempt;

document.forms["myform"]["group"].value = myobj.XIIgroup;

if (myobj.XIIgroup == "other") {

document.forms["myform"]["subject1"].hidden = true;

document.forms['myform']['sub4'].hidden = false;

document.forms["myform"]["sub4"].value = myobj.subject4;

} else {

document.forms["myform"]["subject4"].value = myobj.subject4;

}

document.forms["myform"]["mark1"].value = myobj.mark1;

document.forms["myform"]["total"].value = myobj.total;

document.forms["myform"]["percentage"].value = myobj.percentage;

if (myobj.courses.indexOf(",") !== -1) {

Swal.fire(

'Already applied for:',

myobj.courses,

'warning'

};

xmlhttp.open("POST", "fetchrecord.php", true);

xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xmlhttp.send("x=" + dbparam);

}

$(document).ready(function() {

$("input[name='regdate'],input[name='dob']").datepicker({

yearRange: "1998:2022",

showAnim: "fold”

dateFormat: "dd-mm-yy"

});

});

</script>

<?php

if(!isset($\_GET['signup'])){

exit();

}else{

$signup=$\_GET['signup'];

if($signup=="success"){

echo '<script>

$("#success").fadeIn().html("Successfully submitted");

setTimeout(function(){

$("#success").fadeOut("slow");

}, 2000);

document.forms["myform"]["regno"].focus();

</script>';

}elseif(\strpos($signup,'sqlerror') !== false){

$sqlerrordetail=$\_GET['sqlerrordetail'];

echo "<script>Swal({

type: 'error',

title: 'Oops...',

text: 'Record already exists!',

footer: '<a href>Why do I have this issue?</a>'

})

</script>";

}elseif(\strpos($signup,'dberror') !== false){

$dberrordetail=$\_GET['dberrordetail'];

echo "<script>Swal({

type: 'error',

title: 'Oops...',

text: 'Unable to submit form ',

footer: '<a href>Why do I have this issue?</a>'

})

</script>";

}

}

?>

</body>

</html>