**CHAPTER 1**

**INTRODUCTION**

**1.1AIM**

The scope of the project is to make easy interaction to the user by the system. The basic idea is to send result to the users’mail id and PDA device. The aim is to build a webservice and to avoid server jam while loading exam results, this makes the user comfortable to receive and view at any locations.

* 1. **PROBLEM STATEMENT**

Pervasive results aims at the basic display of results at any locations as enterd by the registered users. People looking the results at same time may cause a problem of server jamming. So our project aims to eliminate this hindrance, by providing results to the registered mail id and respective PDAs. Additionally, the staff has an direct interaction with student in case of displaying results.

* 1. **DESCRIPTION**

The pervasive results means that results can be obtained anywhere and at any time. At first the necessary is the registered login for the user. After registered user the following is displayed

* Name of the student
* Semester
* Subjects
* Mail address

After user selecting these respective options; the details of student are stored in a database. After this the PHP coding enables us to deliver the results to the respective mail id of the student.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 EXISTING SYSTEM**

* In the existing system, the results can only be viewed by client.
* Client gives request to the server and the server gives response. It is based on request-response system.
* The overview of the proposed system is, to avoid the data traffic
* and direct response of the server to the client.
* The server sends the data to the registered clients.
* The server can be a real server which needs required amount of registration to send the data to respective clients

**2.1.1 DISADVANTAGES**

* Server can respond only certain clients and only upto certain extent.
* Sending messages to respective PDAs causes cost of money.
* Server Jamming

**2.2 PROPOSED SYSTEM**

* Referring the existing system, some major enhancements can be made in the proposed system.
* The registered user can only enter the results and other people should register for entering the results.
* After that the name and marks are entered followed by mailid.
* These marks are sent to the mail Id of the respective students.

**2.2.1 ADVANTAGES**

* Getting results at any location.
* Authentication is given for the user who enter the marks.
* Marks for any semesters can be entered.
* It can apply this for all departments.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 FEASIBILITY STUDY**

The main objective of feasibility study is to test the Technical, operational and economical for adding new modules and debugging old running system. All system is feasible for adding new modules and debugging old running system.

The feasibility apects are

* Technical feasibility
* Economic feasibility
* Operational feasibility

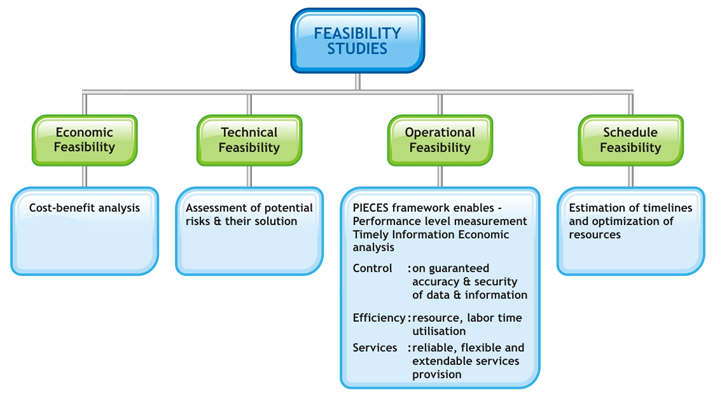
**3.1.1 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. The only technical requirement needed for this project is personal computers. The requirements here are very modest because the system here supports all kinds of operating systems in windows.

So, this system is technically feasible because it needs only personal computers with basic configuration

**3.1.2 ECONOMIC FEASIBILITY**

Economic feasibility is the most frequently used method for evaluating the effectiveness of the proposed system.More commonly known as cost-benefit analysis, the procedure that costs for a proposed system and weights them against the tangible and intangible benefits of the system. The system is cost effective because no cost is required for the execution. The system is economically feasible because the users can easily use this system without any kinds of confusion.

****

**Fig. 3.1 Feasibility Analysis**

**3.1.3 OPERATIONAL FEASIBILITY:**

The aspect of the study is to check the level of response from the server to client. This includes the process of training the user to use the system efficiently. This system will not threaten the user instead it is friendly in its operation. All the user needs to have is an Internet connection provided to his personal computers. The system is operationally feasible because the user can execute by just feeding the details of student.

**3.2 HARDWARE SPECIFICATION**

* System    :   Core I3 processor
* Hard Disk  :   100 GB
* Ram    :   2 GB
* Graphics : 1 GB Radeon

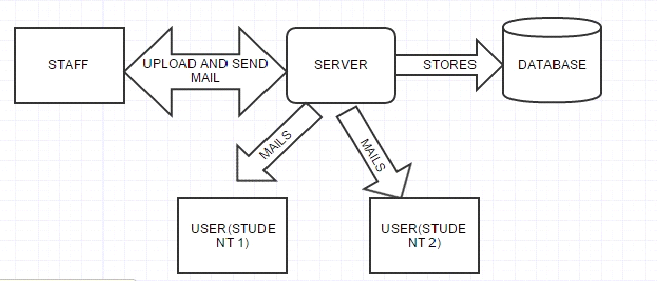
**3.3 SOFTWARE SPECIFICATION**

* Operating system   : Windows 8 pro 64-bit
* Coding Language  : PHP,MYSQL
* Front End Tool : HTML,CSS
* back End Tool : PHP,MYSQL

**CHAPTER 4**

**DETAILED DESIGN**

**4.1 SYSTEM ARCHITECTURE**



**Fig 4.1: System Architecture**

**4.1.1 MODULES**

**HTML**

* **HTML** is the process of creating the web pages.Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the pageWeb browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material.
* The web page is created in such a way that user can register the login id and fill the required student details.

**SQL** is the special purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

* The Data Definition Language (DDL) manages table and index structure. ALTER modifies the structure of an existing object in various ways, for example, adding a column to an existing table or a constraint. TRUNCATE deletes all data from a table in a very fast way, deleting the data inside the table and not the table itself. It usually implies a subsequent COMMIT operation, i.e., it cannot be rolled back (data is not written to the logs for rollback later, unlike DELETE).
* **SQL** is used in order make the respective datas to get stored in the database.

**OBTAINING DATA FROM DATABASE:**

* **SQL**  makes the details to get stored in a database.The values stored remains in the database without any change.
* After this the marks has to be mailed to the respective students.This email validation is done using php language.

**PHP:**

* **PHP** is a server-side scripting language designed for web development but also used as a general-purpose programming language. It is a popular general-purpose scripting language that is especially suited to web development. PHP code can be simply mixed with HTML code, or it can be used in combination with various template engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page
* **php** is used inorder to make the server to send output to many clients as required.

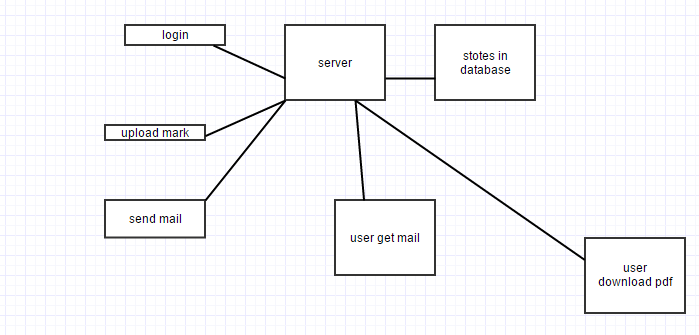
**SMTP**

* Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission.Although electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying.In this module,smtp is used inorder to make the server to send email validation to many clients as required.

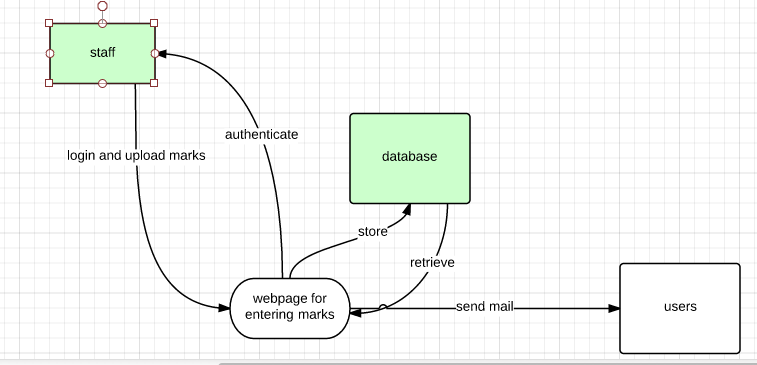
**4.2 DATA FLOW DIAGRAM**

**4.2.1 LEVEL 0**

Staff login in to the webserver and upload the student marks .These marks are stored in to the database while sending mail the marks are retrieved from the database and sent to the user mail.

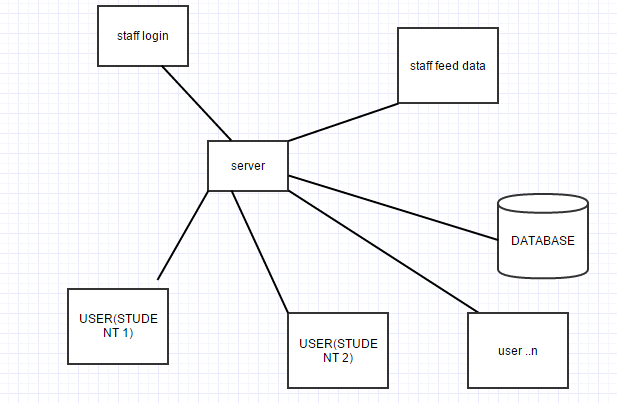
****

**Fig 4.2.1: Level 0**

**4.2.2 LEVEL 1:** 

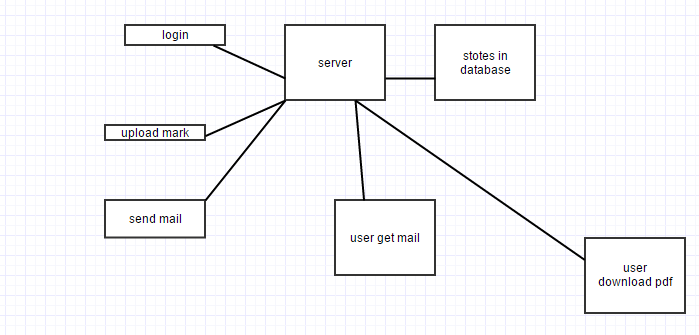
**Fig 4.2.2: Level 1**

**4.2.3 LEVEL 2:**

****

**Fig 4.2.3: Level 2**

**4.2.4 LEVEL 3:**

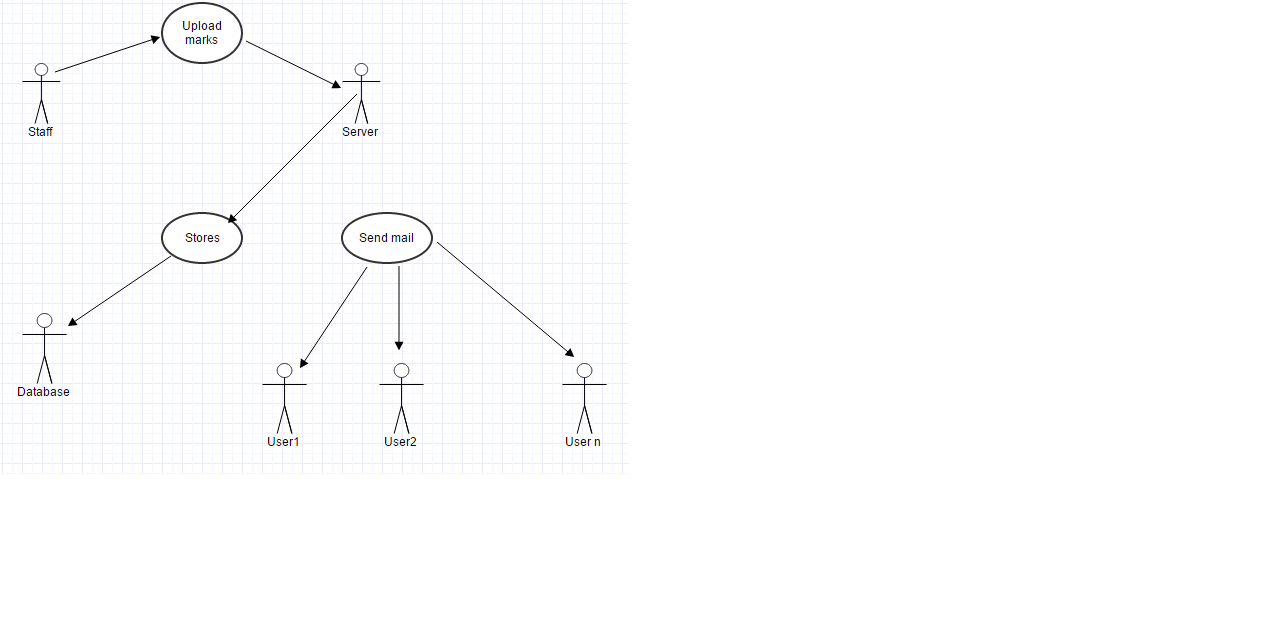
****

**Fig 4.2.4: Level 3**

**4.3 UML DIAGRAMS**

The Unified Modelling Language (UML) is a general purpose modelling language in the field of software engineering. The basic level provides a set of graphic notation techniques to create visual methods of object-oriented software-intensive systems. Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects.

**4.3.1 USECASE DIAGRAM**

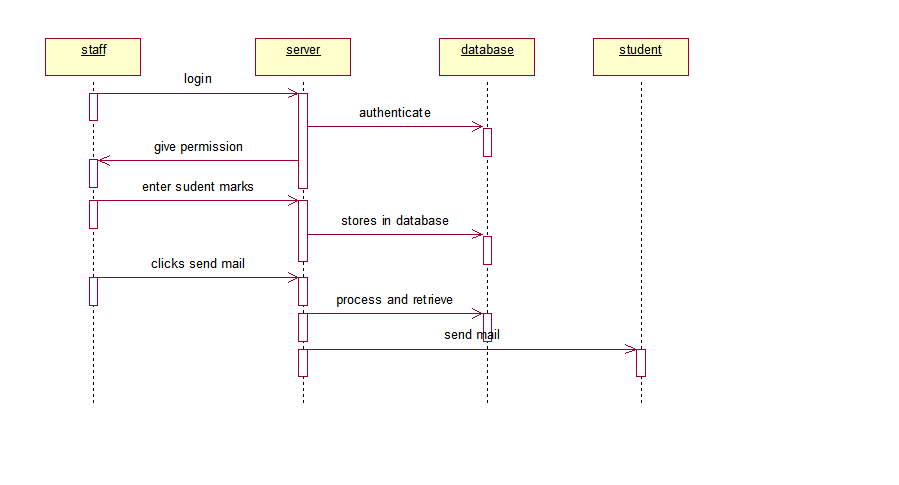
Usecase describe the interaction between one or more actors and the system itself, represented as a sequence of simple steps that take part in a sequence of activities in a dialog with the system to achieve goal.****

**Fig 4.3.1: Use Case Diagram**

There are three actors who are user, gps and web server necessary to run the application. The various functions of these actors like storing, entering, updating and retrieving the details are represented through use cases.

**4.3.2 SEQUENCE DIAGRAM**

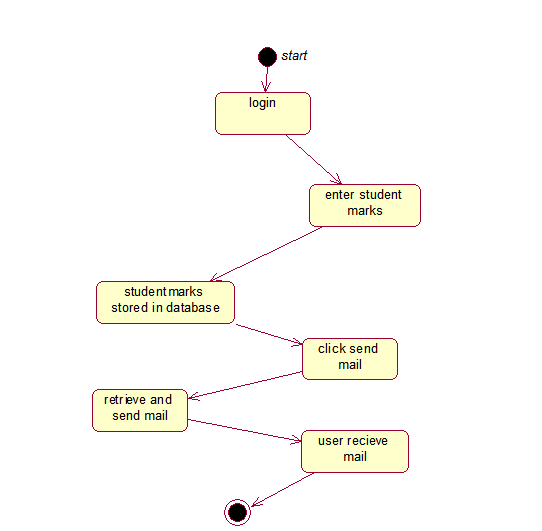
A Sequence diagram shows, as parallel vertical lines different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The actions such as entering the details, storing and so on are expressed in a sequential order through vertical lines. The exchange of data between the actors(user, system and database) are represented by a set of horizontal lines. ****

**Fig 4.3.2: Sequence Diagram**

**4.3.3 ACTIVITY DIAGRAM**

Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

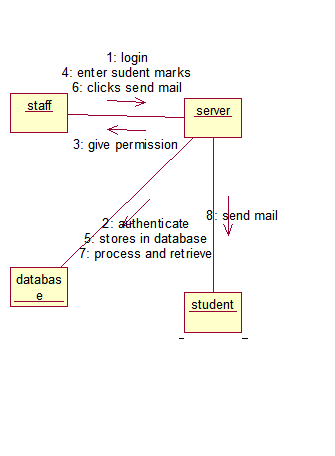


**Fig 4.3.3: Activity Diagram**

The set of activities that takes place among the actors is represented through activity diagram. It has a start and stop state which indicates the user to start and stop the activity respectively. The sequence of activities are represented through arrow marks.

**4.3.4 COLLABORATION DIAGRAM**

A Collaboration diagram is easily represented by modelling objects in a system and representing the association between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of arrows.

****

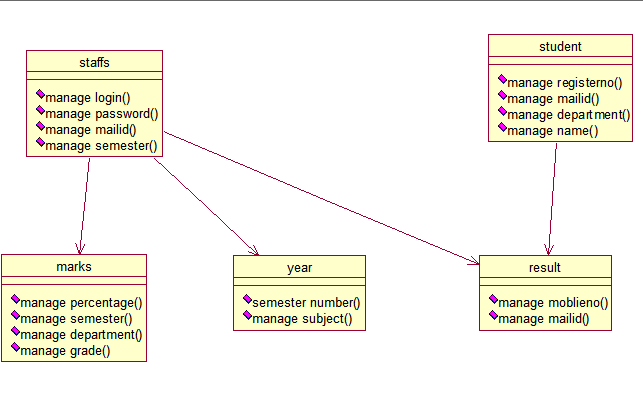
**Fig 4.3.4: Collaboration Diagram**

According to the sequence diagram drawn, an automatic collaboration diagram would be generated which again indicates the relationship among the actors.

**4.3.5 CLASS DIAGRAM:**

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modelling the domain-specific data structure to detailed design of the target system. With the share model facilities,

we can reuse your class model in the interaction diagram for modelling the detailed design of the dynamic behaviour. The Form Diagram allows you to generate diagram automatically with user-defined scope.



**Fig 4.3.4: Class Diagram**

Class diagram describes the attributes and operations of different actors.

Eg: For actor user the attributes are name, gender, source and destination and the operations are entering the details and payment.

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

**5.1 IMPLEMENTATION**

This project deals with server response to respective clients. The user has to register the login id and password. After registering the student name, finally mail id and type of exam is specified. Now 'submit' button is clicked. After clicking the submit button the semesters are displayed .

The semester is clicked and respective subjects for semesters are obtained. Now marks are entered for each subjects and 'send' button is clicked. After clicking this button the datas are stored in a database. The stored data are now sent to the respective mail id of the various students.

**5.2 TESTING**

Testing is an important phase that focuses on an empirical investigation in which the results describe the quality of the system. It cannot confirm system functions properly under all conditions but can establish that it fails under specific conditions. The prime purpose of testing is to guarantee that system successfully built and tested in the development phase meets all the requirements and design parameters.

**5.2.1 UNIT TESTING**

**MAIN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | User clicks in the 'send' button | Result is displayed | Result is displayed | Pass |
| 2 | User clicks in the ‘send' button without Internet | Error message should be displayed | Error message is displayed | Pass |

Table 5.1 Main

**SUBMIT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Authorised User clicks on the submit button | semester should be displayed | semester should be displayed | Pass |
| 2 | Unauthorized User clicks on the submit button | Error message should be displayed | Error message is displayed | Pass |

Table 5.2 Submit

**SEND**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Authorised User clicks on the send button | Results are displayed to mailid | Results are displayed to mailid | Pass |
| 2 | Unauthorised User clicks on the send button without internet | Error message should be displayed | Error message is displayed | Pass |

Table 5.3 Category

**5.2.2 INTEGRATION TESTING:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Results display is successful | Should be able to view the results | Results can be viewed | Pass |
| 2 | Results display is unsuccessful | An error message is displayed | An error message is displayed | Pass |

Table 5.4 Integration testing

**5.2.3 FUNCTIONAL TESTING:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Required details are sent from the server | Results should be stored and displayed | Results are stored and displayed | Pass |
| 2 | Device is not connected to the internet and details cannot be sent | An error message should be displayed | Error message is displayed | Pass |

Table 5.5 Functional Testing

**5.2.4 ACCEPTANCE TESTING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
| 1 | Current mailid is valid | Results should be displayed | Result is displayed | Pass |
| 2 | Current mailid is not valid | An error message should be displayed | Error message is displayed | Pass |

Table 5.6 Acceptance Testing

**5.3 TEST PLAN**

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

**5.4 TEST ANALYSIS**

In this phase of testing, the requirements for software testing are analysed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

**5.5 RESULT**

The project is tested and found to function as expected with no errors. This project gives the way of sending the results to the student at any location .

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

Many students are not aware of their attendance percentage and so this causes a trouble for them in future. This project helps the students in knowing their attendance percentage for every report period. They can also know their marks of every assessment in their respective mail ID . The PDF file is given to every student mail id so that they can download the file and assess themselves by comparing their assessment marks.

**6.2 FUTURE ENHANCEMENT**

In future, we would like to add features to enhance our project . We would like to add more subjects and display results for all departments. Also, we are planning to provide options for the user incase of updating the result at any format they want. We would like to improvise our project by enabling the results to be sent to respective PDAs. We also aims at avoiding the data traffic in a real server.

**APPENDIX-A**

**SAMPLE SOURCE CODE**

**HTML CODING FOR WEB PAGE:**

<html><head><title>ENGINEERING RESULT</title>

<style type="text/css">

body

{

background-image:url('academic.jpg');

background-repeat:no-repeat;

background-position:center;

}

</style></head>

<body><center>

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

NAME:<br>

<input type="text" name="name">

<br>

REG.NUMBER:<br>

<input type="text" name="registernumber"><br>

EMAIL ID:<br>

<input type="email" name="usrtel"><br>

EXAMINATION:<br>

<select name="EXAMINATION">

<option value="CIA 1">CIA1</option>

<option value="CIA 2">CIA2</option>

<option value="CIA3">CIA3</option>

<optionvalue="MODELEXAMINATION">MODEL EXAMINATION</option></select><br>

DEPT:<br>

<select name="DEPT">

<option value="CSE">CSE</option><option value="ECE">ECE</option>

<option value="MECH">MECH</option>

<option value="EEE">EEE</option>

</select>

<br><br>

<button name="register" value="register">Register</button>

</form>

</style></center>

</body>

</html>

**HTML CODING FOR SEMESTER**

<html><head><title>ENGINEERING RESULT</title><style type="text/css">body

{

background-image:url('academic.jpg');

background-repeat:no-repeat;

background-position:center;

}#form\_area{

margin-left:70%;

margin-top:5%;

padding:20px;

border:solid medium;

width:auto;

height:auto;

}</style>

</head><body>

<div id="form\_area">

<p>choose the sem to update:</p><br><br>

<a href="sem1.html">semester 1</a>

<a href="sem2.html">semester 2</a>

<a href="sem3.html">semester 3</a>

<a href="sem4.html">semester 4</a>

<a href="sem5.html">semester 5</a>

<a href="sem6.html">semester 6</a>

<a href="sem7.html">semester 7</a>

<a href="sem8.html">semester 8</a>

</style></center>

</div>

</body>

</html>

**SEMESTER-1:**

<html><head>

<title>ENGINEERING RESULT</title>

<style type="text/css">

</style>

</head></body>

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

tehnicalenglish 1:<br>

<input type="num" name="tehnicalenglish1">

<br>

engineering physics 1:<br>

<input type="num" name="physics1">

<br>

engineering chemistry 1:<br>

<input type="num" name="chemistry1">

<br>

mathematics1:<br>

<input type="num" name="m1">

<br>

engineering graphics:<br>

<input type="num" name="eg">

<br>foc:<br>

<input type="num" name="foc">

<br>

aatendance percentage:<br>

<input type="num" name="attendancepercentage1">

<br><br>

<input type="submit" value="Submit">

<br>

</form>

</style></center>

</body>

</html>

**SEMESTER-2:**

<html><head>

<title>ENGINEERING RESULT</title>

<style type="text/css">body

</style>

</head>

</body>

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

TECHNICAL ENGLISH 2:<br>

<input type="num" name="tehnicalenglish1">

<br>

engineering physics 2:<br>

<input type="num" name="physics1">

<br>engineering chemistry 2:<br>

<input type="num" name="chemistry1">

<br>mathematics2:<br>

<input type="num" name="m1">

<br>EDC:<br>

<input type="num" name="eg">

<br>BCM:<br>

<input type="num" name="foc">

<br>aatendance percentage:<br>

<input type="num" name="attendancepercentage1">

<br><br>

<input type="submit" value="Submit">

<br></form>

</style></center>

</body>

</html>

**SEMESTER-3:**

<html>

<head>

<title>ENGINEERING RESULT</title>

<style type="text/css">

body

{

background-image:url('academic.jpg');

background-repeat:no-repeat;

background-position:center;

}

</style>

</head>

</body>

<form>

oops:<br>

<input type="num" name="tehnicalenglish 1">

<br>

data structure:<br>

<input type="num" name="tehnicalenglish 1">

<br>

tpde:<br>

<input type="num" name="tehnicalenglish 1">

<br>

dpsd:<br>

<input type="num" name="tehnicalenglish 1">

<br>

adc:<br>

<input type="num" name="tehnicalenglish 1">

<br>

foc:<br>

<input type="num" name="tehnicalenglish 1">

<br>

aatendance percentage:<br>

<input type="num" name="attendance percentage">

<br><br>

<input type="submit" value="Submit">

<br>

</form>

</style></center>

</body>

</html>

**SEMESTER-4**

<html>

<head>

<title>ENGINEERING RESULT</title>

<style type="text/css">

body

{

background-image:url('academic.jpg');

background-repeat:no-repeat;

background-position:center;

}

</style>

</head>

</body>

<form>

pqt:<br><input type="num" name="tehnicalenglish 1">

<br>mpc:<br>

<input type="num" name="tehnicalenglish 1">

<br>daa:<br>

<input type="num" name="tehnicalenglish 1">

<br>dbms:<br>

<input type="num" name="tehnicalenglish 1">

<br>coa:<br>

<input type="num" name="tehnicalenglish 1">

<br>foc:<br>

<input type="num" name="tehnicalenglish 1">

<br>aatendance percentage:<br>

<input type="num" name="attendance percentage">

<br><br>

<input type="submit" value="Submit">

<br></form>

</style></center>

</body>

</html>

**EMAIL VALIDATION IN PHP:**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "marks";

// Create connection

$conn = mysqli\_connect($servername, $username, $password,$dbname);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

//echo "Connected successfully";

$emailId = $\_POST["userid"];

$confirmemailId = $\_POST["password"];

$sql="INSERT INTO login(username,password)

VALUES('".$emailId."','".$confirmemailId."')";

if (mysqli\_query($conn, $sql)) {

//header("Location: bank\_details.html");

//echo <a href="index.php">Go back to the main page</a>;

echo 'Successfully Registered';

//echo '<a href="index.html">Go back to the main page</a>';

}

else {

//echo "Error: " . $sql . "<br>" . mysqli\_error($conn);

}

?>

<?phpsession\_start();

$regn=[];

array\_push($regn,$\_SESSION['registernumber']);

require\_once('driverFunctions.php');

foreach($regn as $each)

{

$data = getDetailsFromDB($each);

generateContent($each,$data);

//var\_dump($data);

sendMail($each,$data['email']);

}

?>

**GENERATE PDF FILE:**

functiongetPDFContent($html\_string){

$dompdf = new DOMPDF();

$dompdf->load\_html($html\_string);

$dompdf->render();

//$dompdf->stream("sample.pdf");

$output = $dompdf->output();

return $output;

}

functiongetDetailsFromDB($regno)

{

$host="localhost";

$user="root";

$pass="";

$db\_name="marks";

mysql\_connect($host, $user, $pass);

mysql\_select\_db($db\_name);

$sql="SELECT \* from `test\_marks` WHERE regno=$regno";

$result=mysql\_query($sql) or die($sql."<br/><br/>".mysql\_error());

while($row=mysql\_fetch\_assoc($result))

return $row;

}

functiongenerateContent($regno,$data)

{

$html\_string = getHTMLContent($data);

file\_put\_contents($regno.".html",$html\_string);

$pdf\_string = getPDFContent($html\_string);

file\_put\_contents($regno.".pdf", $pdf\_string);

}

functionsendMail($regno,$email)

{

$mandrill = new Mandrill('AY2Lux1oJMfrNlhZIGumGg'); // this is my API key. Kindly change it to yours after registering

$contents=file\_get\_contents($regno.".html");

$attachment = file\_get\_contents($regno.".pdf");

$attachment\_encoded = base64\_encode($attachment);

$message = array(

'subject' => "Semester Marksheet",

'html' => $contents, // or just use 'html' to support HTMlmarkup

'from\_email' => 'examcell@gmail.com',

'from\_name' => 'Examination Cell',

'to' => array(

array( // add more sub-arrays for additional recipients

'email' => $email,//$row['email'],

),

),

"attachments" => array(

array(

'type' => "application/pdf",

'content' => $attachment\_encoded,

'name' => $regno.".pdf"

)),

/\* Other API parameters (e.g., 'preserve\_recipients => FALSE', 'track\_opens => TRUE',

'track\_clicks' => TRUE) go here \*/

);

$result1 = $mandrill->messages->send($message);

var\_dump($result1);

}

?>

**APPENDIX-B**

**SCREEN SHOTS**

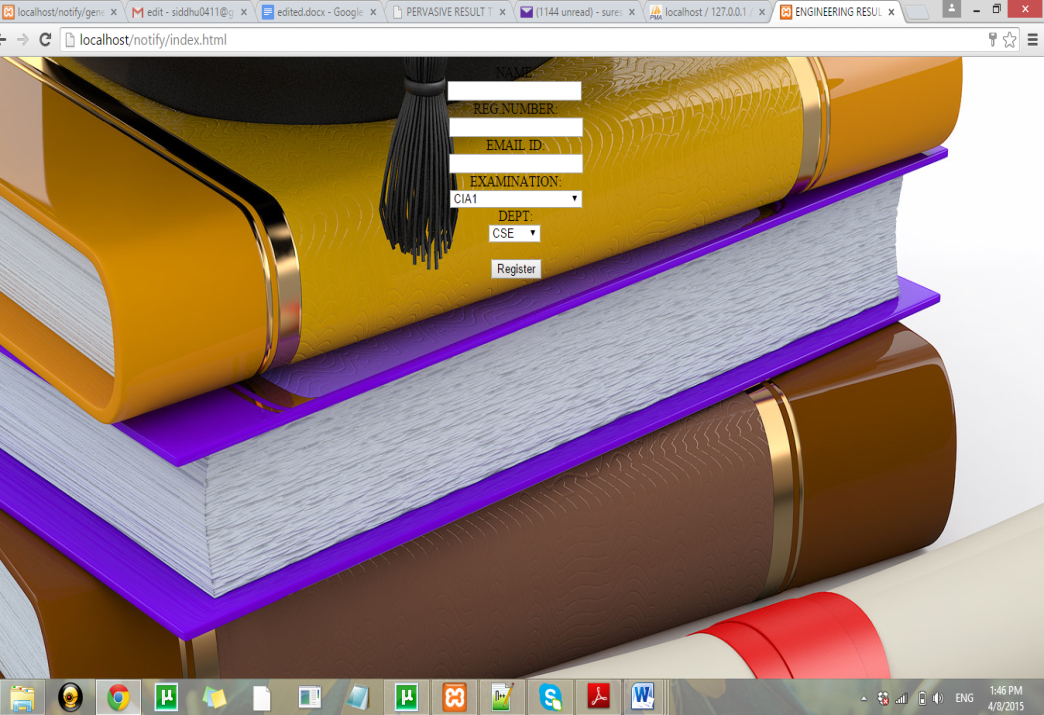
**LOGIN PAGE**

Here the staff use their username and password and login



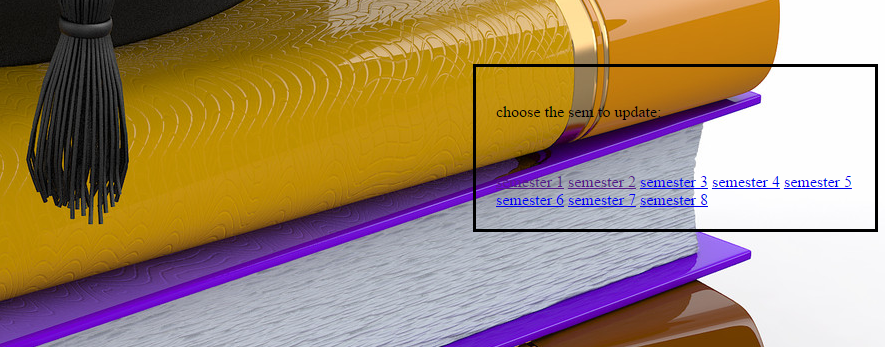
**DATA ENTRY FORM 1:**

Here the staff enter the name of the student ,mailID,semester,dept and clicks REGISTER



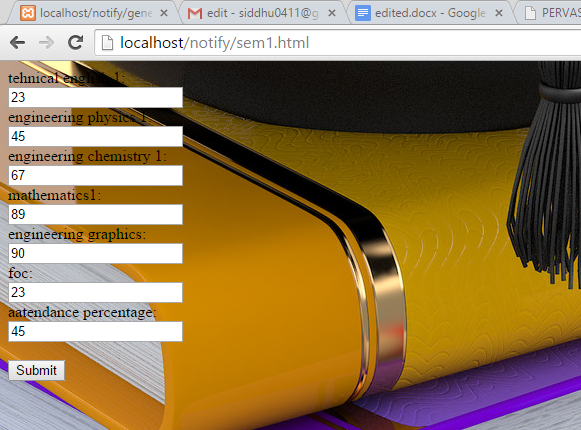
**CHOOSING SEMESTER**

Here the staff choose the semester

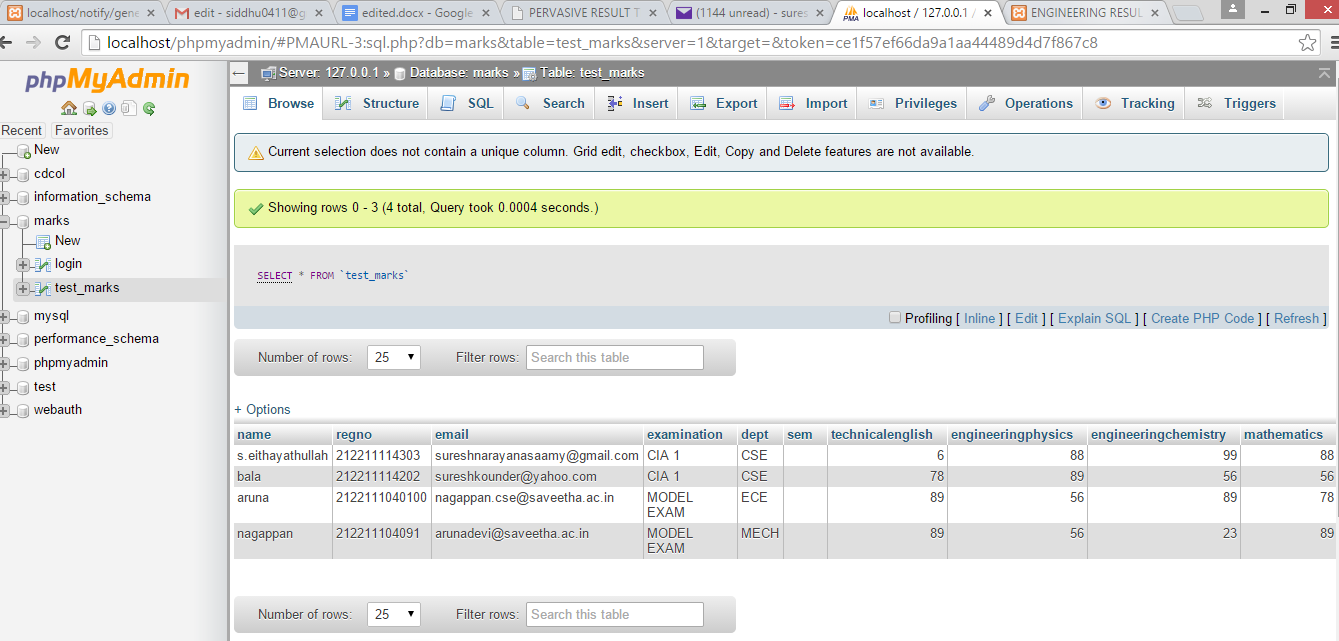


**MARK ENTRY FORM**

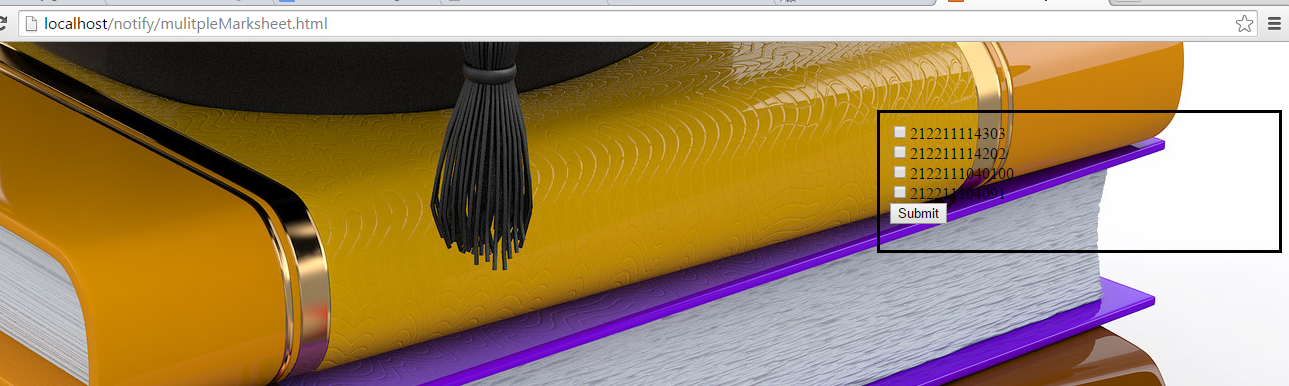
Here the staff enter the marks



**DATABASE**

The entered data are stored in the SQL database

**EMAIL SENDING PAGE**

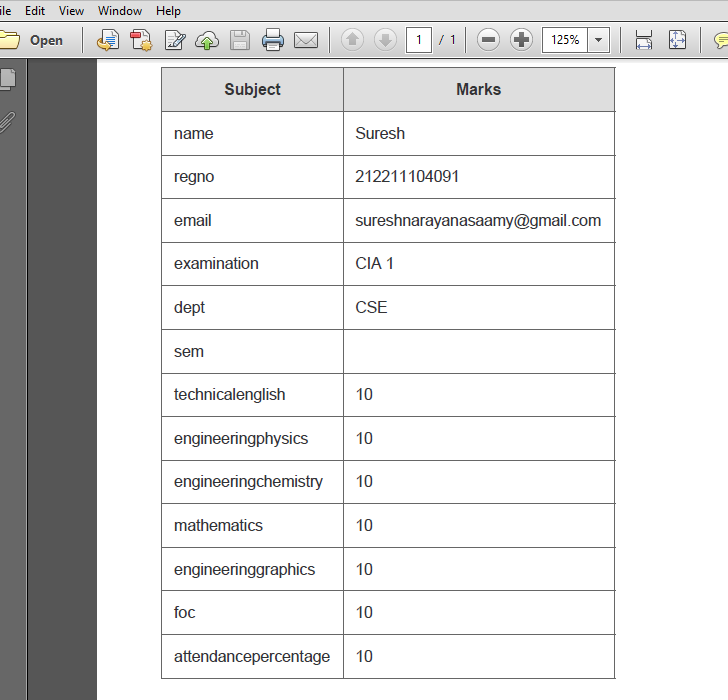
Here the entered registered number in the database are retrieved and shown here in checkbox type , the staff can simply click the checkbox for whom the email has to be sent and click send.****

**RECEIVED EMAIL FROM EXAM CELL**

This is the received mail from the examcenter

****

**PDF ATTACHMENT FILE**



**REFERENCES**

1. SubrataSaha, AtulSajjanhar, Shang Gao, Robert Dew, Ying Zhao, 10th IEEE International Conference on Computer and Information Technology, pp. 698-702, 2010.
2. Web services, http://en.wikipedia.org/wiki/Web\_service (accessed 30 July 2011).
3. C.-J. Lin and Mandrill,http://mandrillapp.com/php
4. R. R. Larson, Geographic Information Retrieval and Spatial Browsing, Smith, L., Gluck, M. (eds.) University of Illinois GIS and Libraries: Patrons, Maps and Spatial Information, pp. 81–124, 1996.
5. T. Joachims, Text categorization with support vector machines. Technical report, LS VIII Number 23, University of Dortmund, 1997.
6. C.-W. Hsu and C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks, Vol. 13, pp. 415– 425, March 2002.
7. Giles M. Foody, and Ajay Mathur, A Relative Evaluation of Multiclass Image Classification by Support Vector Machines, IEEE Transactions on Geoscience and Remote Sensing, Vol. 42, No. 6, June 2004.
8. J. Cervantes, X. Li and W. Yu, Support Vector Classification for Large Data Sets by Reducing Training Data with Change of Classes, IEEE International Conference on Systems, Man and Cybernetics, 2008.
9. Jiawei Han and MichelineKamber, Datamining Concepts and Techniques, Second Edition, Morgan Kaufmann Publishers, November 2005.
10. G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing & Management, Volume 24, Issue 5, pp. 513–523, 1988.
11. Y. Yang and X. Liu, An re-examination of text categorization, 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, pp. 42-49, 1999.
12. Q. R. Zhang and L. Zhang, S. B. Dong and J. H. Tan, Document Indexing in Text Categorization, International Conference on Machine Learning and Cybernetics, 2005.
13. K. Chen and C. Zong, A New Weighting Algorithm for Linear Classifier, International Conference on Natural Language Processing and Knowledge Engineering, 2003.