

STUDY PHASE

1. INTRODUCTION

Our project entitled as “Online Internal Mark Management System”. The main objective of the system is to efficiently consolidate the internal mark of students through a computerised system not only saves a lot of time but also gives fast results. This Internal Mark Management System is mainly focused to consolidate and generate score cards of students. The consolidation and generating score cards are a very difficult task in these days. This system provides two logins, one is for the admin (teacher) and the second one is for the student. This type of system will be the solution to the problems discussed above.

An Online Internal Management System will help the teacher to consolidate marks, generate score card and also viewing of complaint and score of students.

1.1 OBJECTIVES

- Studying the existing system to find out its problem.
- Finding the solution to the problem found in the existing system.
- Designing and building a web application of the system

1.2 TECHNOLOGIES MATERIALS AND METHODS

1.2.1 DATABASE TOOLS

MySQL:

MySQL is a fast, easy to use relational database. It is currently the most popular open-source database. It is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications. MySQL is used for many small and big businesses. It is developed, marketed and supported by MySQL AB, a Swedish company. It is written in C and C++.

1.2.2 PROGRAMMING TOOLS

PHP:

- PHP stands for Hypertext Pre-processor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is a server-side scripting language.
- Performance: Script written in PHP executes much faster than those scripts written in other languages such as JSP & ASP.
- Open-Source Software: PHP source code is free available on the web; you can develop all the version of PHP according to your requirement without paying any cost.

- Platform Independent: PHP are available for WINDOWS, MAC, and LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
- Compatibility: PHP is compatible with almost all local servers used today like Apache, IIS etc.
- Embedded: PHP code can be easily embedded within HTML tags and script.

1.3 FEASIBILITY STUDIES

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it. Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering blindly into risky businesses.

Understanding Feasibility Studies

A feasibility study is simply an assessment of the practicality of a proposed plan or project. As the name implies, these studies ask: Is this project feasible? Do we have the people, tools, technology, and resources necessary for this project to succeed? Will the project get us the return on investment (ROI) that we need and expect?

The goals of feasibility studies are as follows:

- To understand thoroughly all aspects of a project, concept, or plan
- To become aware of any potential problems that could occur while implementing the project
- To determine if, after considering all significant factors, the project is viable—that is, worth undertaking.

The Importance of Feasibility Studies

Feasibility studies are important to business development. They can allow a business to address where and how it will operate. They can also identify potential obstacles that may impede its operations and recognize the amount of funding it will need to get the business up and running. Feasibility studies aim for marketing strategies that could help convince investors or banks that investing in a particular project or business is a wise choice.

1.3.1 TECHNICAL FEASIBILITY

Technical feasibility concerns whether a project can meet its performance objectives. We have already seen some examples of this:

- Modelling the queues for the enrolment process checked that the proposed system would cope with its expected load.

- Developing an expert system for programme advice checked that it would be feasible for candidates to make wise enrolment choices over the internet.

The primary aim of a technical feasibility study is to remove uncertainty. Since the present value of a proposed project depends on the probability of its success, it is important to quantify the risk before committing money to it.

For example, some uncertainty still surrounds the automatic timetabling of examinations: Will a computer system be able to fit the examination sessions into the available period, as the existing manual system has proved that it can?

A key problem here is that the input data for the algorithm won't be available until after enrolments are complete. If the timetabling application proves a failure, time and money will have been wasted, the enrolment staff will feel let down, and the reputation and morale of the project team will suffer. It is crucial that automatic timetabling succeeds. Unfortunately, even after the enrolment period, the data will only contain information for new candidates, and even a successful experiment won't guarantee similar success when a set of data for all candidates become available.

The information that is needed for input to the timetabling process is a list of subject clashes: two subjects clash if at least one candidate is enrolled in both of them, which implies that their examinations mustn't be held at the same time. To assume the worst case, that every subject clashes with each other, would lead to a timetable that only allowed one subject per session, which is clearly unrealistic.

A more realistic test could be made by using the fact that subjects can only clash with subjects that are common to the same study programme and the same semester. Each subject might be assumed to clash with all Offered subjects that share any programme with it. This scenario can be tested as soon as the Components table has been populated, preceding the enrolment period. If the timetabling program works well with this set of data, it is certain to succeed in practice.

But what if this test proves unsatisfactory? If the programme proves inadequate, it is well to know as soon as possible. With luck, the subject clashes that were used to schedule the previous semester's examination timetable might still be available. If they are, and a test using these data proves that the timetabling algorithm is adequate, all is well. All these considerations suggest a strategy for reducing the uncertainty about the project's success before too much effort is wasted.

Likewise, in any project, technical feasibility should be assessed before making any serious financial commitment. Unfortunately, in the eagerness to deliver a product on schedule, such experiments may seem like a waste of time — but if omitted, a project runs the risk of being an expensive failure.

1.3.2 ECONOMIC FEASIBILITY

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources

that will be needed to implement the project, and an assessment of its chances of success. It consists of market analysis, economic analysis, technical and strategic analysis.

1.3.3 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, reducibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development require appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phase.

1.4 SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solution is given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

1.4.1 EXISTING SYSTEM

There are systems to manage internal mark. This type of systems will consolidate internal mark of students. But these systems don't give any provision for students to access this system and also, they did not store the marks. The current system does not provide an option to generate score card and also did not allow the user to interact with teachers through this system.

Consolidating marks manually is a time-consuming process and after all these processes there is chance to occur errors. Working with numbers manually is a tough job and it is very much error prone.

1.4.2 PROPOSED SYSTEM

A teacher or admin and student can use this system each user have their own username and password and also the admin and students have different pages. Already registered users can only access this system. There is no form is provided to register into this system.

When the student's login to the system, there is a hyper link is provided to enter the complaints. The complaints that are entered by the students will be evaluated by the admin or teacher through his/her admin page.

The registered admin/teacher can access the system using their username and password. They can input the internal mark of students, view score card, generate score card(pdf), view complaints and upload a large number of students marks using bulk uploading.

1.5 SYSTEM REQUIREMENT SPECIFICATION

- Online Internal Mark Management System has 2 modules – Admin, Student.
- The admin has the superior power over the system.
- Admin can input marks (using form and bulk uploading), generate score card.
- Admin can view score card, view complaints.
- Students can input their complaints through there page.

1.5.1 PURPOSE

- The main objective of the system is to efficiently tabulate internal mark of students through a fully automated system not only saves a lot of time but also gives accurate results.
- This gives facility to communicate students with teachers easily. Students can let teachers to know their problems.
- Teacher/admin can generate score card of students in a single click, it saves a lot of time and reduces chance of error.

1.5.2 SCOPE

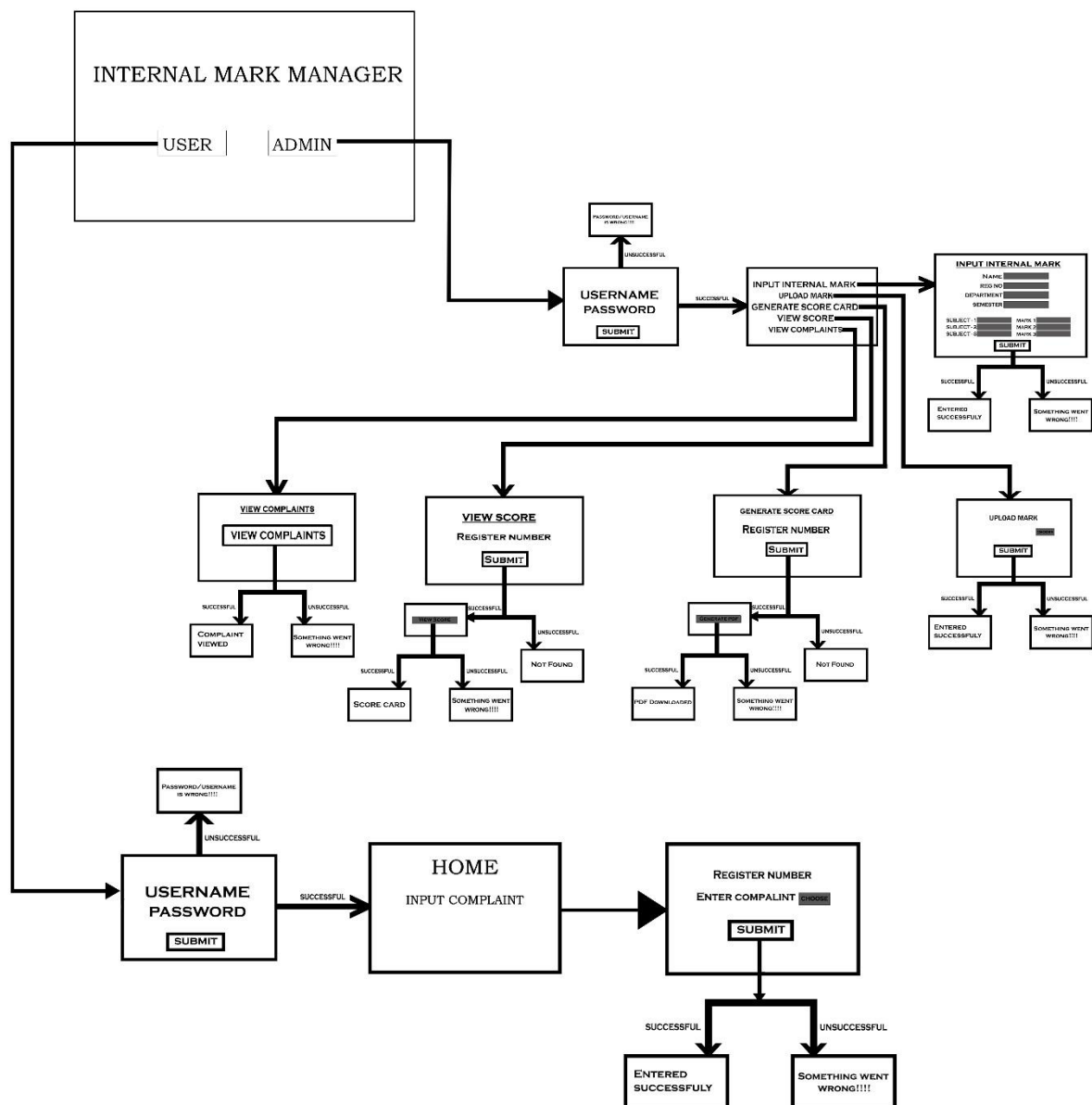
The system will be helpful to both the teachers/admin and students. Any number of admin and teachers are possible and number of semesters is also dynamic so this system will be versatile. Also, students have the provision to enter complaints and the number of students is not fixed.

DESIGN PHASE

2.1 INTRODUCTION TO SCREEN FLOW DIAGRAM

The screen flow diagrams that are used to represent the flow of screens from one to another. This is a structure which is helpful to study the overall design of the system.

2.1.1. SCREEN FLOW DIAGRAM



2.2 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access

easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system.

In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- Accuracy and integrity
- Recovery from failure

A database is an integrated collection of data and provides centralized access to the data. Usually, the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

2.2.1 NORMALIZATION

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design.

A bad database design may lead to certain undesirable things like:

- 1 Repetition of information
- 2 Inability to represent certain information
- 3 Loss of information.

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization's application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs.

Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database. Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus, every normalized relation is in 1NF. Numerous normal forms have been defined. Code defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF. 2NF relations are more desirable than 1NF and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF. Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. I can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So, it is also called “no loss decomposition”.

First Normal Form: A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form: A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form: A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

2.2.2 TABLE STRUCTURE

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS. Data is stored in tables, which is available in the backend. Data can be inserted into table using forms and bulk uploading.

There are mainly 6 tables in our project. They are,

1. login
2. student
3. mark
4. subject
5. studlogin
6. complaint

2.2.2.1 LOGIN

#	Name	Type	Collation	Attributes	Null
1	username (primary key)	Varchar(250)	Latin1_swedish_ci		No
2	password	Varchar(250)	Latin1_swedish_ci		No

2.2.2.2 STUDENT

#	Name	Type	Collation	Attributes	Null
1	reg_no (primary key)	Varchar(300)	Latin1_swedish_ci		No
2	name	Varchar(500)	Latin1_swedish_ci		No
3	mname	Varchar(250)	Latin1_swedish_ci		No
4	lname	Varchar(250)	Latin1_swedish_ci		No
5	department	Varchar(200)	Latin1_swedish_ci		No
6	phone_number	Varchar(200)	Latin1_swedish_ci		No
7	mail_id	Varchar(200)	Latin1_swedish_ci		No

2.2.2.3 MARK

#	Name	Type	Collation	Attributes	Null
1	reg_nom (primary key)	Varchar(250)	Latin1_swedish_ci		No
2	semesterm (primary key)	Varchar(200)	Latin1_swedish_ci		No
3	s1internal1	Varchar(250)	Latin1_swedish_ci		No
4	s1cinternal1	Varchar(250)	Latin1_swedish_ci		No
5	s1semiassi	Varchar(150)	Latin1_swedish_ci		Yes
6	s2internal1	Varchar(250)	Latin1_swedish_ci		No
7	s2cinternal1	Varchar(250)	Latin1_swedish_ci		No
8	s2semiassi	Varchar(150)	Latin1_swedish_ci		Yes
9	s3internal1	Varchar(250)	Latin1_swedish_ci		No
10	s3cinternal1	Varchar(250)	Latin1_swedish_ci		No
11	s3semiassi	Varchar(150)	Latin1_swedish_ci		Yes
12	attendance	Varchar(150)	Latin1_swedish_ci		No

2.2.2.4 SUBJECT

#	Name	Type	Collation	Attributes	Null
1	reg_nos (primary key)	Varchar(200)	Latin1_swedish_ci		No
2	semesters (primary key)	Varchar(200)	Latin1_swedish_ci		No
3	subject1	Varchar(500)	Latin1_swedish_ci		No
4	subject2	Varchar(500)	Latin1_swedish_ci		No
5	subject3	Varchar(500)	Latin1_swedish_ci		No

2.2.2.5 STUDLOGIN

#	Name	Type	Collation	Attributes	Null
1	username (primary key)	Varchar(200)	Latin1_swedish_ci		No
2	password	Varchar(200)	Latin1_swedish_ci		No

2.2.2.6 COMPLAINT

#	Name	Type	Collation	Attributes	Null
1	com_id (primary key)	smallint(6)			No
2	date	date			No
3	reg_no	smallint(6)			Yes
4	complaint	varchar(1000)	Latin1_swedish_ci		Yes

2.3 SYSTEM DESIGN

The system design is the most creative and challenging phase of system development life cycle. It is an approach for the creation of proposed system, in which the logic and details structure of the proposed system is designed, which will help the system coding. The most creative and challenging phase of the system development process is design phase it is a solution, how to approach to the creation of the proposed system. Design is the first step in the development of the engineered product is initiated only after a clear exposition of expected product is available. System Design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system. A number of sub- systems is to be identified which constitute the whole system.

2.3.1 SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system, System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g., the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts

to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships.

The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people.

2.3.2 MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components.

Two types of modular programming techniques.

1. Top down
2. Bottom-up Approaches

TOP-DOWN APPROACH

Top-down programming means I start at the top and work towards down. In this I first consider the entire program and it is then sub divided in to less complex, smaller and easily manageable form until a stage is reach when further break down will serve no useful purpose. That is, to cut all the program in to a number of independent task and then cut task in to smaller sub task and so on. Until they are small enough to code easily. These task and sub task form the basic functions in the program. In top-down programming the programmers write the main () first. That means all the major functions it will need. Later the programmers look at the requirements of each of these.

BOTTOM-UP APPROACH

Bottom-up programming is exactly opposite top-down programming. In this I start at bottom work towards top. That is the programmers write the lower-level function first and then using this lower-level functions they write the next higher level and so on. In these basic elements are specified first, these elements are link together to form large programs and this is repeated until the complete program is formed.

Advantages

- Reusability of code
- Disadvantages
- Less easy to implement compared to top down

DEVELOPMENT PHASE

3.1. SYSTEM ENVIRONMENT

Hardware specification:

Processor: 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz

RAM: 8 GB

Hard Disk: 256 GB

Input Device: Standard Keyboard and Mouse

Software specification:

Operating System: Windows 11

Front End: PHP

Back End: MySQL

3.2. SELECTION OF PROGRAMMING LANGUAGE AND OPERATING SYSTEM

Coding is an important part of programming paradigm for software development. It is mainly used to develop apps, websites and software. The programming language contains project planning, analysis, design, coding, testing and maintenance. All the above requirements are considered the programming language for “Database Development for Spice Compounds”. It is developed using tools such as Python within the Ubuntu 14.04 platform, Flask Framework for python (Web development). And the MySQL is used as the back end.

Windows is a computer operating system based on the Windows NT Kernel distributed by Microsoft. It is a paid operating system for PC. It is very much powerful operating system.

In high-level programming languages, python provide strong priority on readability and efficiency, and is mainly compared to other languages like Java, PHP, or C++.Python can be used to build server-side web applications. It is designed to be highly readable. In web programming python working with multiple types of servers, databases and web frameworks for the development of web applications. Python is also used for the representation of chemical structure with openbabel. Openbabel is a chemical toolbox used to represent chemical data. Flask is a micro-framework and is suitable for small-scale applications.

Flask has contained more features than other frameworks. It consists of features like unit testing and built-in development server that allow to create reliable and efficient web applications.

3.3. CODING

Coding is a list of step-by-step instructions that get computers to do what you want them to do. This step is also called programming phase. The performance of software design starts by using program code with appropriate programming language and developing error free executable programs in efficient manner. Coding is undertaken once the design phase is complete and the design documents have been successfully reviewed. Computer Coding is term used for writing codes & executing it for getting desired output.

In this phase, every module identified and specified in the design document is independently coded and unit tested.

- The input to the coding phase is the design document.
- During the coding phase, various modules identified in the design document are coded according to the respective module specifications. In this phase, each module identified and specified in the design document is independently coded and unit tested.
- A coding standard gives a regular form to the codes written by different engineers.
- It provides sound understanding of the code
- It encourages good programming practice.

INDEX

```
<html>
<head>
<title>Internal mark management</title>
<style>
*{
margin: 0;
padding: 0;
font-family: 'Poppins', sans-serif;
}
.banner-text{
text-align: center;
color: black;
padding-top: 200px;
}
.banner-text h1{
font-size: 90px;
font-family: 'san-serif';
}
.banner-btn{
margin: 70px auto 0;
font-size: 20px;
font-weight: 300;
```



```

}
.banner-btn a{
width: 150px;
text-decoration: none;
display: inline-block;
margin: 0 10px;
padding: 12px 0;
color:black;
border: 0.5px solid black;
position: relative;
z-index: 1;
transition: color 0.5s;
}
.banner-btn a span{
width: 0;
height: 100%;
position: absolute;
top: 0;
left:0;
background: cyan;
z-index: -1;
transition: 0.5s;
}
.banner-btn a:hover span{
width: 100%;
}
.banner-btn a:hover{
color:black;
}
</style>
</head>
<body bgcolor= "dffffe">
<section>
<div class="banner-text">
<h1>Internal Mark Management</h1>
<div class="banner-btn">
<a href="screen2.php"><span></span>Admin</a>
<a href="studentlogin.php"><span></span>User</a>
</div>
</div>
</section>
</body>
</html>

```

SCREEN2

```
<!doctype html>
<?php
session_start();
?>
<html>
<center>
<br><br>
<head>
<h1>LOGIN</h1>
</head>
<body bgcolor="dffffe">
<form action="" method="post">
<table cellspacing="10" cellpadding="10">
<tr>
<td>Username</td>
<td><input type="text" name="user"></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" name="pass"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="sub" value="login"></td>
<td></td>
</tr>
</table>
</form>
</body>
</center>
</html>
<?php
if(isset($_POST["sub"]))
{
$name=$_POST["user"];
$word=$_POST["pass"];
$con=mysqli_connect("localhost","root","","internal_mark");
$sql="select * FROM login WHERE username='$name' and password='$word'";
$res=mysqli_query($con,$sql);
$rescheck=mysqli_num_rows($res);
//if($name="ashin" && $word="pass")
if($rescheck > 0)
{
$_SESSION["uname"]=$name;
header('location:screen3.php');
```

```

}
else
{
echo "username or password is wrong";
}
}
else
{
}
}
?>

```

SCREEN3

```

<html>
<body bgcolor="dffffe">
<center>
<br>
<b><h1>HOME</h1></b>
<br>
</center>
</html>
<?php
session_start();
if(isset($_SESSION["uname"]))
{
echo '<center>';
echo '<a href="screen4.php">Input internal mark</a>';
echo '<br><br>';
echo '<a href="screen6.0.php">Generate score card</a>';
echo '<br><br>';
echo '<a href="screen7.php">View Score</a>';
echo '<br><br>';
echo '<a href="screen8.php">view complaints</a>';
echo '<br><br>';
echo '<a href="excel.php">Upload mark</a>';
echo '<br><br>';
echo '</center>';
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>
<html>
<a href="logout.php">Logout</a>
</body>
</html>

```

SCREEN4

```
<!doctype html>
<?php
session_start();
if(isset($_SESSION["uname"]))
{
?>
<html>
<head>
<title>Input mark</title>
</head>
<body bgcolor="dffffe">
<center>
<br><br>
<h2>INPUT MARK</h2>
<form action="" method="post">
<table cellspacing="10" cellpadding="10">
<tr>
<td>Register number </td>
<td></td>
<td><input type="text" name="re"></td>
</tr>
<tr>
<td>Semester</td>
<td></td>
<td><input type="text" name="sem"></td>
</tr>
<tr>
<td>Subject-1</td>
<td></td>
<td><input type="text" name="sub1"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr1"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr2"></td>
</tr>
<tr>
<td>Assingment<input type="radio" name="assign" value="1"></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seminar<input type="radio" name="assign" value="0"></td>
<td></td>
<td></td>
</tr>
<tr>
```

```

<td>Mark</td>
<td></td>
<td><input type='text' name='semi1'></td>
</tr>
<tr>
<td>Subject-2</td>
<td></td>
<td><input type="text" name="sub2"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr21"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr22"></td>
</tr>
<tr>
<td>Assingment<input type="radio" name="tassign" value="1"></td>
<td></td>
<td>Seminar<input type="radio" name="tassign" value="0"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="semi2"></td>
</tr>
<tr>
<td>Subject-3</td>
<td></td>
<td><input type="text" name="sub3"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr31"></td>
</tr>
<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="intr32"></td>
</tr>
<tr>
<td>Assingment<input type="radio" name="thassign" value="1"></td>
<td></td>
<td>Seminar<input type="radio" name="thassign" value="0"></td>
</tr>

```

```

<tr>
<td>Mark</td>
<td></td>
<td><input type="text" name="semi3"></td>
</tr>
<tr>
<td>Attendance</td>
<td></td>
<td><input type="text" name="atte"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="sub" value="submit"></td>
<td></td>
</tr>
</table>
</form>
</center>
</html>
<?php
if(isset($_POST["sub"]))
{
$reg=$_POST["re"];
$seme=$_POST["sem"];
$subj1=$_POST["sub1"];
$subj2=$_POST["sub2"];
$subj3=$_POST["sub3"];
$passign=$_POST['assign'];
$ptassign=$_POST['tassign'];
$pthassign=$_POST['thassign'];

$pintr1=$_POST["intr1"];
$pintr2=$_POST["intr2"];
$pssemi1=$_POST["semi1"];

$pintr21=$_POST["intr21"];
$pintr22=$_POST["intr22"];
$pssemi2=$_POST["semi2"];

$pintr31=$_POST["intr31"];
$pintr32=$_POST["intr32"];
$pssemi3=$_POST["semi3"];

$patte=$_POST["atte"];

$con5=mysqli_connect("localhost","root","","internal_mark");
$sql5="select reg_no from student where reg_no='$reg'";
$res5=mysqli_query($con5,$sql5);
$rccheck=mysqli_num_rows($res5);

```

```

if($rcheck>0)
{
$con=mysqli_connect("localhost","root","","internal_mark");
$sql="select reg_nom, semesterm FROM mark WHERE reg_nom='$reg' and
semesterm='$seme';";
$res=mysqli_query($con,$sql);
$rescheck=mysqli_num_rows($res);
if($rescheck<1)
{
$sql2="insert into mark values
('$reg','$seme','$pintr1','$pintr2','$psemi1','$pintr21','$pintr22','$psemi2','$pintr31','$pintr32','$
psemi3','$patte');";
$res2=mysqli_query($con,$sql2);
}
else
{
echo "Already inserted into mark table";
}
$con1=mysqli_connect("localhost","root","","internal_mark");
$sql3="select * FROM subject WHERE reg_nos='$reg' and semesters='$seme';";
$res3=mysqli_query($con1,$sql3);
$rescheck2=mysqli_num_rows($res3);
if($rescheck2<1)
{
$sql4="insert into subject values ('$reg','$seme','$subj1','$subj2','$subj3');";
$res4=mysqli_query($con1,$sql4);
}
else
{
echo '<br>';
echo "Already inserted into subject table";
}
}
else
{
echo "student is not registered";
}
if($res4 && $res2)
{
echo '<script type="text/javascript">alert("Entered")</script>';
}
else {
echo '<script type="text/javascript">alert("Not Entered")</script>';
}
}
else
{
}
?>

```

```

<html>
<br>
<a href="screen3.php">Back to home page</a>
<br><br>
<a href="logout.php">Logout</a>
</body>
</html>
<?php
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>

```

SCREEN5

```

<?php
session_start();
if(isset($_SESSION["puname"]))
{
?>
<html>
<body bgcolor="dffffe">
<head>
<br><br>
<center>
<h2>INPUT COMPLAINT</h2>
</head>

<body>
<form action="" method="post">
<table cellspacing="10" cellpadding="10">
<tr>
<td>Register number</td>
<td><input type="text" name="rg"></td>
</tr>
<tr>
<td>Enter complaint</td>
<td><textarea id="text-area" name="com">
</textarea></td>
</tr><tr>
<td></td>
<td><input type="submit" name="sub" value="Submit"></td>
<td></td>
</tr></table>
</form>
</body>
</center>

```



```

</html>
<?php
if(isset($_POST["sub"]))
{
$reg=$_POST["rg"];
$pcom=$_POST["com"];
$dt = date('Y-m-d');
$con=mysqli_connect("localhost","root","","internal_mark");
$sql1="select reg_no from student where reg_no='$reg'";
$res1=mysqli_query($con,$sql1);
$rescheck=mysqli_num_rows($res1);
if($rescheck>0)
{
$sql="insert into complaint(date,reg_no,complaint) values('$dt','$reg','$pcom')";
$res=mysqli_query($con,$sql);
if($res)
{
echo '<script type="text/javascript">alert("entered")</script>';
}
else
{
echo '<script type="text/javascript">alert("not entered")</script>';
}
}
else{
echo '<script type="text/javascript">alert("no student is registered in this register number")</script>';
}
}
else
{
}
?>
<html>
<a href="studhome.php">Back to home page</a>
<br><br>
<a href="logout.php">Logout</a>
</body>
</html>
<?php
}
else{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>

```

SCREEN6.0

```
<?php
session_start();
if(isset($_SESSION["uname"]))
{
?>
<html>
<head>
<title>Pdf generation</title>
<!--<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />-->
</head>
<body bgcolor="dffffe">
<center>
<br><br>
<h2>GENERATE SCORE CARD</h2>
<form action="screen6.1.php" method="POST">
<table cellspacing="10" cellpadding="10">
<tr>
<td>Register number</td><td><input type="text" name="reg"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="sub" class="btn btn-danger" value="submit"></td>
<td></td>
</tr>
</table>
</form>
</center>
</body>
<a href="screen3.php">Back to home</a>
<br>
<br>
<a href="logout.php">Logout</a>
</html>
<?php
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>
```

SCREEN6.1

```
<?php
if(isset($_POST["sub"]))
{
$re=$_POST["reg"];
$con=mysqli_connect("localhost","root","","internal_mark");
$sql1="SELECT * FROM student WHERE reg_no='$re'";

$sql3="SELECT mark.semesterm as sem, mark.s1internal1 as s1internal1,
mark.s1cinternal1 as s1internal2, mark.s2internal1 as s2internal1, mark.s2cinternal1 as
s2internal2,
mark.s3internal1 as s3internal1, mark.s3cinternal1 as s3internal2, mark.s1semiassi as sa1,
mark.s2semiassi as sa2, mark.s3semiassi as sa3, mark.attendance as atten,
subject.reg_nos AS reg_nos,
subject.subject1 as sub1, subject.subject2 as sub2, subject.subject3 as sub3,mark.reg_nom
AS reg_nom
FROM subject,mark WHERE subject.reg_nos=mark.reg_nom AND
subject.semesters=mark.semesterm AND subject.reg_nos='$re'";

$res2=mysqli_query($con,$sql3);
$res=mysqli_query($con,$sql1);
$rescheck=mysqli_num_rows($res);
$rescheck2=mysqli_num_rows($res2);
if($rescheck && $rescheck2>0)
{
require('tcpdf_min/tcpdf.php');
$obj_pdf = new TCPDF('P', PDF_UNIT, PDF_PAGE_FORMAT, true, 'UTF-8', false);
$obj_pdf->SetCreator(PDF_CREATOR);
$obj_pdf->SetTitle("Score Card");
$obj_pdf->SetHeaderData("", "", PDF_HEADER_TITLE, PDF_HEADER_STRING);
$obj_pdf->setHeaderFont(Array(PDF_FONT_NAME_MAIN, "",
PDF_FONT_SIZE_MAIN));
$obj_pdf->setFooterFont(Array(PDF_FONT_NAME_DATA, "",
PDF_FONT_SIZE_DATA));
$obj_pdf->SetDefaultMonospacedFont('helvetica');
$obj_pdf->SetFooterMargin(PDF_MARGIN_FOOTER);
$obj_pdf->SetMargins(PDF_MARGIN_LEFT, '5', PDF_MARGIN_RIGHT);
$obj_pdf->setPrintHeader(false);
$obj_pdf->setPrintFooter(false);
$obj_pdf->SetAutoPageBreak(TRUE, 10);
$obj_pdf->SetFont('helvetica', "", 12);
$obj_pdf->AddPage();
$content = "";
$content .= ' <br>
<h1 align="center">SCORE CARD</h1>
<h3 align="center">Student details</h3>
<table border="1" cellspacing="0" cellpadding="5">
```

```

<tr>
<th width="12%">Reg_no</th>
<th width="20%">Name</th>
<th width="15%">Department</th>
<th width="20%">Phone number</th>
<th width="33%">Mail ID</th>
</tr>';
$content .= fetch_data($re);
$content .= '</table>';
$obj_pdf->writeHTML($content);
$content1 = "";
$content1 .= '
<h3 align="center">Internal mark</h3>
<table border="1" cellspacing="0" cellpadding="5">
<tr>
<th width="15%">Semester</th>
<th width="15%">Subject1</th>
<th width="10%">Mark1</th>
<th width="15%">Subject2</th>
<th width="10%">Mark2</th>
<th width="15%">Subject3</th>
<th width="10%">Mark3</th>
<th width="10%">Grade</th>
</tr>';
$content1 .= fetch_data1($re);
$content1 .= '</table>';
$obj_pdf->writeHTML($content1);
$obj_pdf->Output('sample.pdf', 'I');
}
else{
echo "no student is found in this register number!!!!";?>
<html><br><br><a href="screen6.0.php">Generate another pdf</a></html>
<?php
}
}
function fetch_data($re)
{
$output = "";
$connect = mysqli_connect("localhost", "root", "", "internal_mark");
$sql = "SELECT * FROM student WHERE reg_no='$re'";
$result = mysqli_query($connect, $sql);
while($row = mysqli_fetch_array($result))
{
$name=$row["name"];
$mname=$row["mname"];
$lname=$row["lname"];

if($mname=="")
{

```

```

$naf=$fname.' '.$lname;
}
else
{
$naf=$fname.' '.$mname.' '.$lname;
}
$output .= '<tr>
<td>'.$row["reg_no"].'</td>
<td>'.$naf.'</td>
<td>'.$row["department"].'</td>
<td>'.$row["phone_number"].'</td>
<td>'.$row["mail_id"].'</td>
</tr>';
}
return $output;
}

function fetch_data1($re)
{
$output = "";
$connect2=mysqli_connect("localhost", "root", "", "internal_mark");

$sql2="SELECT mark.semesterm as sem, mark.s1internal1 as s1internal1,
mark.s1cinternal1 as s1internal2, mark.s2internal1 as s2internal1, mark.s2cinternal1 as
s2internal2,
mark.s3internal1 as s3internal1, mark.s3cinternal1 as s3internal2, mark.s1semiassi as sa1,
mark.s2semiassi as sa2, mark.s3semiassi as sa3, mark.attendance as atten,
subject.reg_nos AS reg_nos,
subject.subject1 as sub1, subject.subject2 as sub2, subject.subject3 as sub3,mark.reg_nom
AS reg_nom
FROM subject,mark WHERE subject.reg_nos=mark.reg_nom AND
subject.semesters=mark.semesterm AND subject.reg_nos='$re';";

$result2=mysqli_query($connect2, $sql2);
while($row = mysqli_fetch_array($result2))
{
$s1internal1=$row["s1internal1"];
$s1internal2=$row["s1internal2"];
$s2internal1=$row["s2internal1"];
$s2internal2=$row["s2internal2"];
$s3internal1=$row["s3internal1"];
$s3internal2=$row["s3internal2"];
$sa1=$row["sa1"];
$sa2=$row["sa2"];
$sa3=$row["sa3"];
$atten=$row["atten"];

$mark1=((($s1internal1)/16)+((($s1internal2)/16)+($sa1)+($atten);
$mark2=((($s2internal1)/16)+((($s2internal2)/16)+($sa2)+($atten);

```

```
$mark3=((($s3internal1)/16)+((($s3internal2)/16)+($sa3)+($atten));
$avg=($mark1+$mark2+$mark3)/3;
```

```
if($avg>=18)
{
$gr='a';
}
elseif($avg>=16)
{
$gr='b';
}
elseif($avg>=14)
{
$gr='c';
}
else {
$gr='d';
}
```

```
$output .= '<tr>
<td>'.$row["sem"].'</td>
<td>'.$row["sub1"].'</td>
<td>'.$mark1.'</td>
<td>'.$row["sub2"].'</td>
<td>'.$mark2.'</td>
<td>'.$row["sub3"].'</td>
<td>'.$mark3.'</td>
<td>'.$gr.'</td>
```

```
</tr>';
}
return $output;
}
?>
```

SCREEN7

```
<?php
session_start();
if(isset($_SESSION["uname"]))
{
?>
<html>
<head>
<title>Show score</title>
</head>
<body bgcolor="dffffe">
<center>
<br><br>
```

```

<h2>VIEW SCORE</h2>
<form action="" method="post">
<table cellpadding="10" cellspacing="10">
<tr>
<td>Register number</td>
<td><input type="text" name="rg"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="sub" value="submit"></td>
<td></td>
</tr>
</table>
</form>
<?php
if(isset($_POST["sub"]))
{
$reg=$_POST["rg"];
$con=mysqli_connect("localhost","root","","internal_mark");
$sql="SELECT mark.semesterm as sem, mark.s1internal1 as s1internal1, mark.s1cinternal1
as s1internal2, mark.s2internal1 as s2internal1, mark.s2cinternal1 as s2internal2,
mark.s3internal1 as s3internal1, mark.s3cinternal1 as s3internal2, mark.s1semiassi as sa1,
mark.s2semiassi as sa2, mark.s3semiassi as sa3, mark.attendance as atten,
subject.reg_nos AS reg_nos,
subject.subject1 as sub1, subject.subject2 as sub2, subject.subject3 as sub3,mark.reg_nom
AS reg_nom
FROM subject,mark WHERE subject.reg_nos=mark.reg_nom AND
subject.semesters=mark.semesterm AND subject.reg_nos='$reg';";
$sql1="select * from student where reg_no='$reg';";
$res=mysqli_query($con,$sql);
$res1=mysqli_query($con,$sql1);
$rescheck=mysqli_num_rows($res);
$rescheck1=mysqli_num_rows($res1);
?>
<table border="1" cellpadding="5" cellspacing="0">
<tr>
<th width="20%">Register Number</th>
<th width="20%">Name</th>
<th width="17%">Department</th>
<th width="18%">Phone number</th>
<th width="25%">Mail_id</th>
</tr>
<?php
if($rescheck1>0)
{
while($row=mysqli_fetch_array($res1))
{
$name=$row["name"];
$mname=$row["mname"];

```

```

$name=$row["lname"];

if($mname=="")
{
$naf=$fname.' '.$lname;
}
else
{
$naf=$fname.' '.$mname.' '.$lname;
}
?>
<tr>
<td><?php echo $row["reg_no"];?></td>
<td><?php echo $naf;?></td>
<td><?php echo $row["department"];?></td>
<td><?php echo $row["phone_number"];?></td>
<td><?php echo $row["mail_id"];?></td>
</tr>
<?php
}
}
else
{
echo "no student found";
}
?>
</table>
<br>
<br>
<table border="1" cellspacing="0" cellpadding="5">
<tr>
<th width="15%">Semester</th>
<th width="15%">Subject1</th>
<th width="10%">Mark1</th>
<th width="15%">Subject2</th>
<th width="10%">Mark2</th>
<th width="15%">Subject3</th>
<th width="10%">Mark3</th>
<th width="10%">Grade</th>
</tr>
<?php
if($rescheck>0)
{
while($row=mysqli_fetch_array($res))
{
$s1internal1=$row["s1internal1"];
$s1internal2=$row["s1internal2"];
$s2internal1=$row["s2internal1"];
$s2internal2=$row["s2internal2"];

```



```

$s3internal1=$row["s3internal1"];
$s3internal2=$row["s3internal2"];
$sa1=$row["sa1"];
$sa2=$row["sa2"];
$sa3=$row["sa3"];
$atten=$row["atten"];
$mark1=((($s1internal1)/16)+((($s1internal2)/16)+($sa1)+($atten);
$mark2=((($s2internal1)/16)+((($s2internal2)/16)+($sa2)+($atten);
$mark3=((($s3internal1)/16)+((($s3internal2)/16)+($sa3)+($atten);

$avg=($mark1+$mark2+$mark3)/3;

if($avg>=18)
{
$gr='a';
}
elseif($avg>=16)
{
$gr='b';
}
elseif($avg>=14)
{
$gr='c';
}
else {
$gr='d';
}
?>
<tr>
<td><?php echo $row["sem"];?></td>
<td><?php echo $row["sub1"];?></td>
<td><?php echo $mark1;?></td>
<td><?php echo $row["sub2"];?></td>
<td><?php echo $mark2;?></td>
<td><?php echo $row["sub3"];?></td>
<td><?php echo $mark3;?></td>
<td><?php echo $gr;?></td>
</tr>
<?php
}
}
else
{
echo '<script type="text/javascript">alert("No Student Found in this Register
Number")</script>';
}
?>
</table>
<?php

```

```

    }
    else
    {
    }
    ?>
</center>
<a href="screen3.php">Back to home page</a>
<br>
<a href="logout.php">logout</a>
</body>
</html>
<?php
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>

```

SCREEN8

```

<?php
session_start();
if(isset($_SESSION["uname"]))
{
?>
<!DOCTYPE html>
<html>
<head>
<title>Complaints</title>
</head>
<body bgcolor="dffffe">
<center>
<br><br>
<h2>COMPLAINTS</h2>
<table border="1" cellspacing="0" cellpadding="5">
<tr>
<td>Register Number</td>
<td>Complaint</td>
<td>Delete</td>
</tr>
<?php
include "database.php";
$records = mysqli_query($con,"select * from complaint");
while($data = mysqli_fetch_array($records))
{
?>
<tr>
<td><?php echo $data['reg_no']; ?></td>

```

```

<td><?php echo $data['complaint']; ?></td>
<td><a href="delete.php?id=<?php echo $data['com_id']; ?>">Delete</a></td>
</tr>
<?php
}
?>
</table>
</center>
<a href="screen3.php">Go to home</a>
<br><br>
<a href="logout.php">Logout</a>
</body>
</html>
<?php
}
else {
echo '<script type="text/javascript">alert("Access denied")</script>';
}

```

DELETE

```

<?php
include "database.php";
$id= $_GET['id'];
$del = mysqli_query($con,"delete from complaint where com_id = '$id'");
if($del)
{
header("location:screen8.php");
}
else
{
echo "Error deleting record";
}
?>

```

EXCEL

```

<?php
session_start();
if(isset($_SESSION["uname"]))
{
include 'database.php';
?>
<html>
<head>
<title>Upload file</title>
</head>
<body bgcolor="dffffe">
<center>

```

```

<br><br>
<table cellspacing="10" cellpadding="10">
<form method="post" action="file-upload.php" enctype="multipart/form-data">
<tr>
<td></td>
<td><h2>UPLOAD FILE</h2></td>
<td></td>
</tr>
<tr>
<td>Choose File</td>
<td><input type="file" name="uploadfile"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="submit"></td>
<td></td>
</tr>
</table>
</form>
</center>
<a href="screen3.php">Go to home page</a>
<br><br>
<a href="logout.php">Logout</a>
</body>
</html>
<?php
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>

```

FILEUPLOAD

```

<?php
session_start();
if(isset($_SESSION["uname"]))
{
?>
<html>
<body bgcolor="dffffe">
</body>
<?php
include 'database.php';

$uploadfile=$_FILES['uploadfile']['tmp_name'];

require 'PHPExcel/Classes/PHPExcel.php';

```

```

require_once 'PHPExcel/Classes/PHPExcel/IOFactory.php';
$objExcel=PHPExcel_IOFactory::load($uploadfile);
foreach($objExcel->getWorksheetIterator() as $worksheet)
{
    $highestrow=$worksheet->getHighestRow();

    for($row=0;$row<=$highestrow;$row++)
    {
        $reg_no=$worksheet->getCellByColumnAndRow(0,$row)->getValue();
        $sem=$worksheet->getCellByColumnAndRow(7,$row)->getValue();

        $sub1=$worksheet->getCellByColumnAndRow(8,$row)->getValue();
        $mark11=$worksheet->getCellByColumnAndRow(9,$row)->getValue();
        $mark12=$worksheet->getCellByColumnAndRow(10,$row)->getValue();
        $assisemi1=$worksheet->getCellByColumnAndRow(11,$row)->getValue();

        $sub2=$worksheet->getCellByColumnAndRow(12,$row)->getValue();
        $mark21=$worksheet->getCellByColumnAndRow(13,$row)->getValue();
        $mark22=$worksheet->getCellByColumnAndRow(14,$row)->getValue();
        $assisemi2=$worksheet->getCellByColumnAndRow(15,$row)->getValue();
        $sub3=$worksheet->getCellByColumnAndRow(16,$row)->getValue();
        $mark31=$worksheet->getCellByColumnAndRow(17,$row)->getValue();
        $mark32=$worksheet->getCellByColumnAndRow(18,$row)->getValue();
        $assisemi3=$worksheet->getCellByColumnAndRow(19,$row)->getValue();
        $atten=$worksheet->getCellByColumnAndRow(20,$row)->getValue();

        $sql="SELECT * FROM student WHERE reg_no='$reg_no'";
        $res=mysqli_query($con,$sql);
        $rescheck=mysqli_num_rows($res);

        $sql1="SELECT * FROM mark WHERE reg_nom='$reg_no' && semesterm='$sem'";
        $res1=mysqli_query($con,$sql1);
        $rescheck1=mysqli_num_rows($res1);

        $sql2="SELECT * FROM subject WHERE reg_nos='$reg_no' && semesters='$sem'";
        $res2=mysqli_query($con,$sql2);
        $rescheck2=mysqli_num_rows($res2);

        if($reg_no!="")
        {
            if($rescheck)
            {
                if($rescheck1<1)
                {
                    $insertsub="INSERT INTO `subject`
                    ( `reg_nos`,`semesters`,`subject1`,`subject2`,`subject3`) VALUES
                    ('$reg_no','$sem','$sub1','$sub2','$sub3')";
                    $insertsubqry=mysqli_query($con,$insertsub);
                    echo "Entered to subject table <br />";
                }
            }
        }
    }
}

```

```

    }
    else
    {
        echo "already entered to subject table <br />";
    }
    if($rescheck2<1)
    {
        $insertmark="INSERT INTO mark VALUES
        ('$reg_no','$sem','$mark11','$mark12','$assisemi1','$mark21','$mark22','$assisemi2','$mark3
        1','$mark32','$assisemi3','$atten')";
        $insertmarkres=mysqli_query($con,$insertmark);
        echo "Entered to mark table <br />";
    }
    else
    {
        echo "already entered to mark table <br />";
    }
}
else
{
    echo $reg_no, " not reg <br />";
}
}
}
}
?>
<html>
<br>
<br>
<a href="excel.php">Upload another file</a>
<br>
<br>
<a href="screen3.php">Go to home page</a>
</html>
<?php
}
else {
    echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>

```

DATABASE

```

<?php
$con=mysqli_connect('localhost','root','','internal_mark');
if(mysqli_connect_errno())
{
    echo 'Failed to connect to database'.mysqli_connect_error();
}??>

```

STUDENTLOGIN

```
<?php
session_start();
?>
<html>
<body bgcolor="dffffe">
<center>
<br><br>
<h1>STUDENT LOGIN</h1>
<form action="" method="post">
<table cellspacing="10" cellpadding="10">
<tr>
<td>Username</td>
<td><input type="text" name="stus"></td>
</tr>
<tr>
<td>Password</td>
<td><input type="password" name="stpww"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="sub" value="login"></td>
<td></td>
</tr>
</table>
</form>
</center>
</body>
</html>
<?php
if(isset($_POST["sub"]))
{
    $pstud=$_POST["stus"];
    $pstpw=$_POST["stpww"];
    $con=mysqli_connect("localhost","root","","internal_mark");
    $sql="select * from studlogin where username='$pstud' and password='$pstpw'";
    $res=mysqli_query($con,$sql);
    $rescheck=mysqli_num_rows($res);
    if($rescheck>0)
    {
        $_SESSION["pname"]=$pstud;
        header('location:studhome.php');
    }
    else
    {
        echo '<script type="text/javascript">alert("Username or password is wrong")</script>';
    }
}
```

```
else
{
}
?>
```

STUDHOME

```
<?php
session_start();
if(isset($_SESSION["puname"]))
{
?>
<html>
<head>
<title>Home</title>
</head>
<body bgcolor="dffffe">
<center>
<br>
<b><h1>HOME</h1></b>
<br>
<a href="screen5.php">Input complaints</a>
</center>
<a href="logout.php">Logout</a>
</body>
</html>
<?php
}
else
{
echo '<script type="text/javascript">alert("Access denied")</script>';
}
?>
```

LOGOUT

```
<?php
session_start();
session_unset();
session_destroy();
header('location:index.php');
?>
```


TESTING AND IMPLEMENTATION

4.1. TESTING

It is the process of evaluating a system or its components with the intent to find that whether it satisfies the related requirements. In computer hardware and software development, testing is used at key checkpoints in its done with executing the software overall process to identify whether objectives are being met. The process or method of finding errors in a software application or program so that the application functions according to the end user's requirement is called software testing.

Software testing is used to assess the feature of a software item. Testing process ensures the quality of the product. Testing is the process that should be done during the development of software. In other words, software testing is a verification and validation process.

4.1.1 BLACK BOX TESTING

Black box testing is also called functional testing. It is a software testing method and is used to test the software without knowing the internal structure of code or program. Internal system design is not considered in this type of testing. This type of testing is mainly focus on the software requirements and specifications. In the proposed work black box testing is used for the following requirements:

- Login
- Inputting internal mark
- Generate score card
- View score card
- View complaints

ADVANTAGES OF BLACK BOX TESTING

- The designer and the tester are independent of each other because testing is unbalanced.
- The tester does not need knowledge of any specific programming languages like java, C++...
- Test is done from the point of view of the user.
- Code access not required

DISADVANTAGES OF BLACK BOX TESTING

- The test can be redundant if the software designer has already run a test case
- Test cases are difficult to design, without having clear functional specifications

- It is difficult to identify all possible inputs in limited testing time. So, writing test cases are difficult to design.

4.1.2. WHITE BOX TESTING

White box testing is also known as structural testing and glass box testing. This testing is related to the knowledge of the internal logic of an application's code. It is also called open box testing. Internal software and code working must be known for this type of testing. Tests are related to the coverage of code statements, branches, paths, conditions. In proposed work white box testing is used for the following requirements:

- Programming code
- Database Accessing

ADVANTAGES OF WHITE BOX TESTING

- The testing of the software no need to wait for the GUI
- It helps in optimizing the code
- Beneficent side-effects
- Provide stability and usability of the test cases.
- It helps in removing the extra lines of code, which can bring in hidden defects.
- Is done with executing the software

DISADVANTAGES OF WHITE BOX TESTING

- Expensive
- It takes more time for the tester to develop the test cases.
- Test cases are a waste if changes in the implementation code are done frequently

4.1.3. UNIT TESTING

Unit is the smallest testable part of software. Unit testing is used to validate that individual units of source code are working properly. In object-oriented programming, the smallest unit is a method and it contain a base/super class, abstract class or derived/child class but in procedural programming language a unit may be an individual program, function, procedure, etc., while the main advantage of the unit testing is used to improve the quality of code and save the tester's time and effort.

4.1.4. SYSTEM TESTING

System testing is the testing and is used to ensure that by putting the software in different environments it still works. It is done with executing the software system testing the application is working correctly from the point of view of a user. The main purpose of this system testing is to evaluate the system's compliance with the specified requirements. Whole system is tested as per the requirements. Black-box type testing that is related to overall requirements specifications, covers all combined parts of a system.

4.1.5. USER ACCEPTANCE TESTING

Acceptance testing is to ensure that the delivered product meets the expectations of the user. It is belonging to the class of black box testing. The goal of acceptance testing is to verify that the software is done with executing the software ready and can be used by the end-users to perform those functions and tasks for the software development. This type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

4.1.6. VALIDATION TESTING

Validation is the process of evaluating the final software product that correctly identifies the customer expectations and requirements. Validation Testing checks that the product correctly meets the customer needs. This testing can also be defined as to demonstrate that the product fulfils its intended use when deployed on suitable environment. This testing is done with executing the software.

4.2 TEST CASES

Test Case Id: TC1

Test used: Black Box Testing/White Box Testing

Correct Data:

Username: admin

Password: pass

Function: Admin login

SL NO	STEP	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Enter username only and click submit button	Username: admin	Please enter username and password correctly	Please enter username and password correctly	Success
2	Enter incorrect password or username	Username: admin Password: pa	Wrong username or password	Wrong username or password	Success
3	Enter password and username correctly	Username: admin Password: pass	Valid username and password go to home page	Valid username and password go to home page.	Success

Test Case Id: TC2

Test used: Black Box Testing/White Box Testing

Correct Data:

Username: 400

Password: 400

Function: User login

SL NO	STEP	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Enter username only and click submit button	Username: 400	Please enter username and password correctly	Please enter username and password correctly	Success
2	Enter incorrect password or username	Username: 400 Password: pass	Wrong username or password	Wrong username or password	Success
3	Enter password and username correctly	Username: 400 Password: 400	Valid username and password go to home page	Valid username and password go to home page.	Success

Test Case Id: TC3

Test used: Black Box Testing/White Box Testing

Correct Data:

Register number: 400

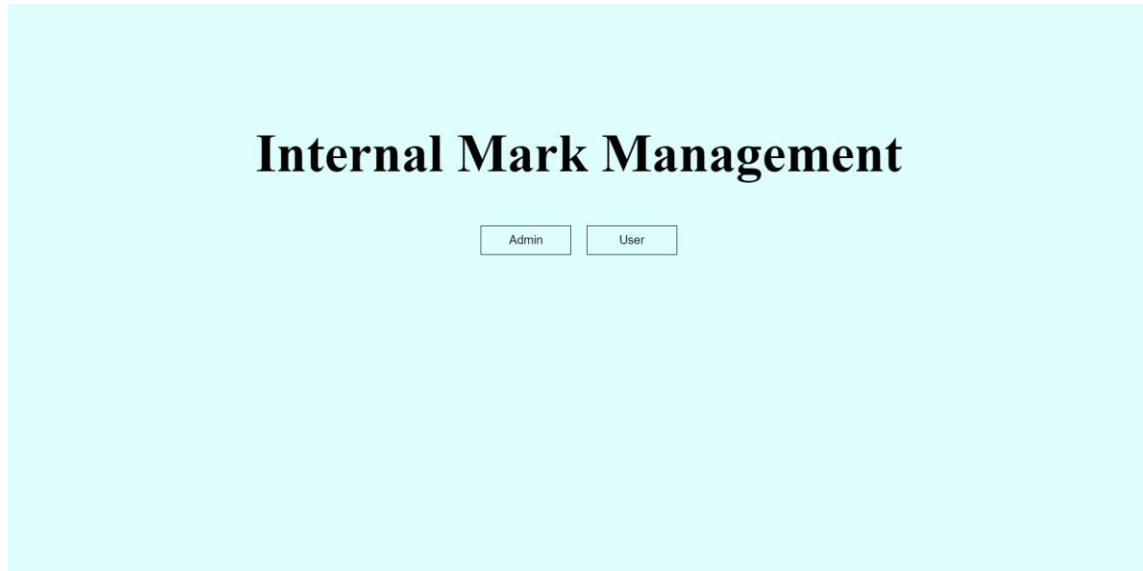
Function: View score of students

SL NO	STEP	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	Click submit button without entering register number	Register number:	Please enter register number	Please enter register number	Success
2	Entered incorrect register number	Register number: 100	Wrong register number	Wrong register number	Success
3	Entered register number correctly	Register number: 400	Valid Register number and show score	Valid Register number and show score	Success

SCREENSHOTS

5.1. FORM DESIGN

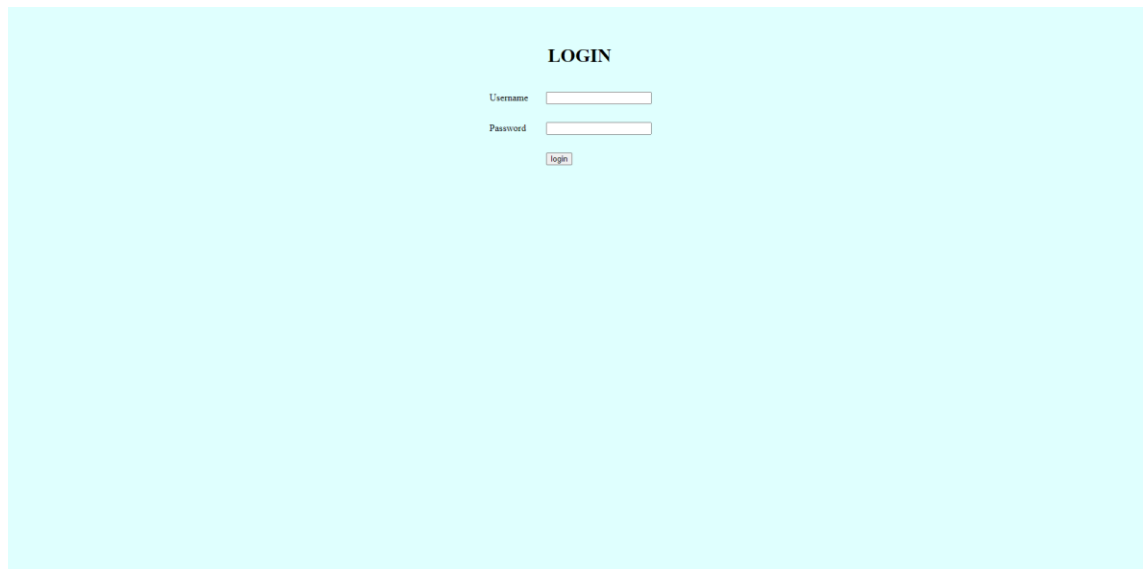
5.1.1. INDEX



A screenshot of a web application interface for 'Internal Mark Management'. The title is centered at the top in a large, bold, black serif font. Below the title are two rectangular buttons, one labeled 'Admin' and one labeled 'User', positioned side-by-side. The entire interface is set against a solid light blue background.

Internal Mark Management

5.1.2. SCREEN2



A screenshot of a web application interface for a login screen. The title 'LOGIN' is centered at the top in a bold, black, all-caps sans-serif font. Below the title are two text input fields. The first field is preceded by the label 'Username' and the second by 'Password', both in a small, black, all-caps sans-serif font. Below the password field is a small rectangular button labeled 'login' in a red, all-caps sans-serif font. The entire interface is set against a solid light blue background.

LOGIN

Username

Password

5.1.3. SCREEN3

[Logout](#)

HOME

[Input internal mark](#)
[Generate score card](#)
[View Score](#)
[view complaints](#)
[Upload mark](#)

5.1.4. SCREEN4

Back to home page

Logout

INPUT MARK

Register number

Semester

Subject-1

Mark

Mark

Assignment ○

Semester ○

Mark

Subject-2

Mark

Mark

Assignment ○

Semester ○

Mark

Subject-3

Mark

Mark

Assignment ○

Semester ○

Mark

Attendance

submit

5.1.5. SCREEN6.0

[Back to home](#)
[Logout](#)

GENERATE SCORE CARD

Register number

submit

5.1.6. SCREEN7

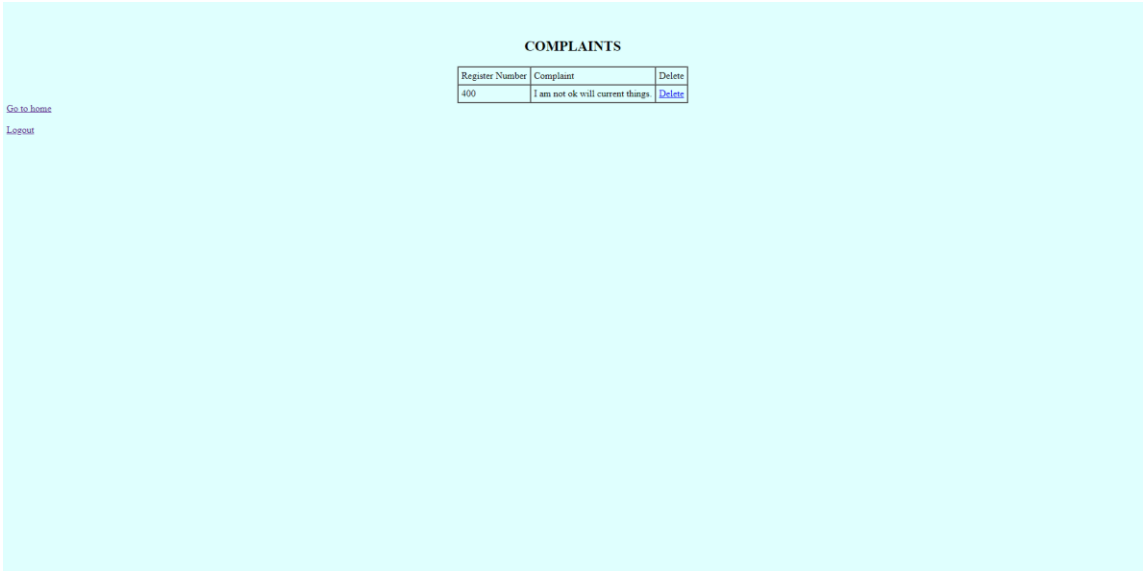
[Back to home page](#)
[logout](#)

VIEW SCORE

Register number

submit

5.1.7. SCREEN8



5.1.8. EXCEL



5.1.9. STUDENTLOGIN

STUDENT LOGIN

Username

Password

login

5.1.10. STUDHOME

HOME

Logout

[Input complaints](#)

5.1.11. SCREEN5

[Back to home page](#)
[Logout](#)

INPUT COMPLAINT

Register number

Enter complaint

Submit

CONCLUSION AND BIBILOGRAPHY

6.0. CONCLUSION

The project was successfully completed within the time span allotted. All the modules are tested separately and put together to form the main system. Finally, the modules are tested with real data and it worked successfully. Thus, the system has fulfilled the entire objective defined. Our goal of developing this “Online Internal Mark Management System” has come to a good result without many defects.

Our project “Online Internal Mark Management System” is very efficient, with the proposed system. Compare to the current manual system, the implementation will reduce manual work.

6.0.1. ADVANTAGES OF PROJECT

1. Minimise human errors.
2. Facilitate accessibility of students to access with teachers.
3. Increased reliability.
4. Saves lot of time.

6.0.2. LIMITATIONS OF PROJECT

In our project no provision for students to view there marks.

6.1. BIBLIOGRAPHY

- ELIAS M AWARD, SYSTEM ANALYSIS AND DESIGN, TATA MCGRAW HILL PUBLICATIONS, 1986
- NAVEEN PRAKASH, PRINCIPAL OF DATABASE MANAGEMENT SYSTEM, TATA MCGRAW HILL PUBLICATIONS, 1991
- PHP MOTHER SITE @www.mysql.com
- PHP MANUAL @<http://php.net/manual/en/>.