A

Project Report

On

“Exam Hub”

BCA SEM V PROJECT

BHAKTA KAVI NARSINH MEHTA UNIVERSITY

Submitted in partial fulfillment for the BCA 5th Semester

SUBMITTED BY

Odedara Hitesh (03052000379)

Dr. Subhash College of Computer Science Junagadh

Certificate

Roll No._____ Enrolment No._____

This is to certify that the Project / practical work was satisfactorily carried out and hence submitted this report is the Bonafede work of

Mr. _____

Student of Bachelor of Computer Application Sem ___ in the

_____ Laboratory of Dr. Subhash College of Computer Science Junagadh during the academic year 2022-23.

_____ _____

Internal Guide **HOD**

_____

Principal

ACKNOWLEDGEMENT

I take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

I am privileged to express our sense of gratitude to our respected **Project guide Prof._____** unparalleled knowledge, moral fiber, and judgment along with his know-how, was an immense support in completing the project.

I am also grateful to **Dr. Vimal Parmar,** the **Principal,** and all other faculties of BCA for the brainwave and encouragement given. I take this opportunity also to thank our friends and contemporaries for their cooperation and compliance.

And lastly, I would like to pay my sincere regards to my parents and to our near and dear ones whose moral support and guidance helped to complete and submit the project well before the due date.

Hitesh Odedara (03052000379)

Bachelor of Computer Application

Dr.Subhash College of Computer Science

– Junagadh

_____

Principal

DECLARATION

I _____ student of Dr. Subhash College of Computer Science in Bachelor of Computer Application Sem 5 declares that this project is not submitted anywhere else in the market or in any other form of use.

I also assure you that no copy of this project is shared with any other student and this project is developed by us with the guidance of our faculties.

Hitesh Odedara

TABLE OF CONTENTS

PROJECT PROFILE

1.1 Student Profile
1.2 Project Definition
1.3 System Specification

Project Profile		
	Project Title	Exam Hub
	Project Type	Web Application
	Tools & Technology	VS Code Editor, Bootstrap, JS Modes
	Front – End	HTML, CSS, JS,PHP
	Back – End	MySQL
	Team Member	1

1.1 Student Profile

Name	**:**	Odedara Hitesh
Enrolment No.	**:**	03052000379
Mobile No:		+91 9913262131
Email Id.	**:**	Superstars0986@gmail.com
Guided By	**:**	Prof. Dr. Subhash College of Computer Science, Junagadh (362001)

1.2 Project Definition: -

- Useful for educational institutes to prepare for an exam and save the time that will take to check the paper and prepare mark sheets. it will help the institute to test students and develop their skills.to allow the department to create tests and answers.
- An online examination system is the complete agenda of an online-based test that has got multiple features and functionalities. The online examination system uses online exam software through which the tests are created, conducted, and also evaluated. This type of examination system has got multiple benefits few of them are that it eliminates the dependency on paper for the question and answer sheets, and eliminates any sort of manual workload which is too much in the case of an offline test.
- In the Online Examination System, the entire steps involved in a paper-based test right from the beginning to the end are digitized with the use of online assessment software. Let's read how each step is automated with the application of the online exam platform.
- It is used by Coaching, institutes, colleges, universities, government, and corporate to conduct online assessments with ease.

1.2 System Specification:

A high-level requirements specification is required. The purpose of the requirements analysis is to identify requirements for the proposed system. The emphasis is on the discovery of user requirements. Each requirement (or problem) must be defined and documented in the requirements catalog. Each requirement is recorded in the requirements catalog on a requirements catalog form. A copy of the form is in the appendix section of the standards manual.

The form should be completed as follows:

- Project/System – **Exam Hub**
- Analyst – **Hitesh Odedara.**
- Version - a version number is assigned to a requirement. The initial version of a requirement is number one. However, the requirement may need to be updated in the course of the development of the project, so then the requirements catalog entry will be replaced with the updated version of the
- requirement and the updated version number will reflect this change.

- Status - the status of the requirement will be either ongoing or complete. When the status is ongoing the status box will be empty, and when the status is complete the status box will contain a tick
- Page - page numbering will be maintained within the catalog.
- Source/Origin - is the originator of the requirement; the person with the responsibility for negotiation about the requirement.
- Requirement Number - a unique requirement number is assigned to each requirement. A requirement number is an incremental number starting with e.
- Priority - a priority is assigned to the requirement. The priority given is agreed upon between the originator and the analyst. There are three priority levels:

high (H), medium (M), and low (L). **Exam Hub**

- Functional Requirements and Non-Functional Requirements - see next page.

Human Resource Management and Performance Evaluation System. • Related Documents – a reference to any related documents, eg: us documentation, data flow diagrams.

- Proposed Solution - any possible solution or general comments.

➢SOFTWARE SPECIFICATION

- Operating System: Windows 22000.978/64- bit
- Front End : HTML & CSS and PHP
- Database: MySQL
- Server: Apache

➢HARDEWARE SPECIFICATION

- Processor: x64 compatible processor with 2.60 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or grater

TOOLS / PLATFORM

2.1 Front End Tools 2.2 Back End Tools

2.1 Front End Tools

- **PHP**

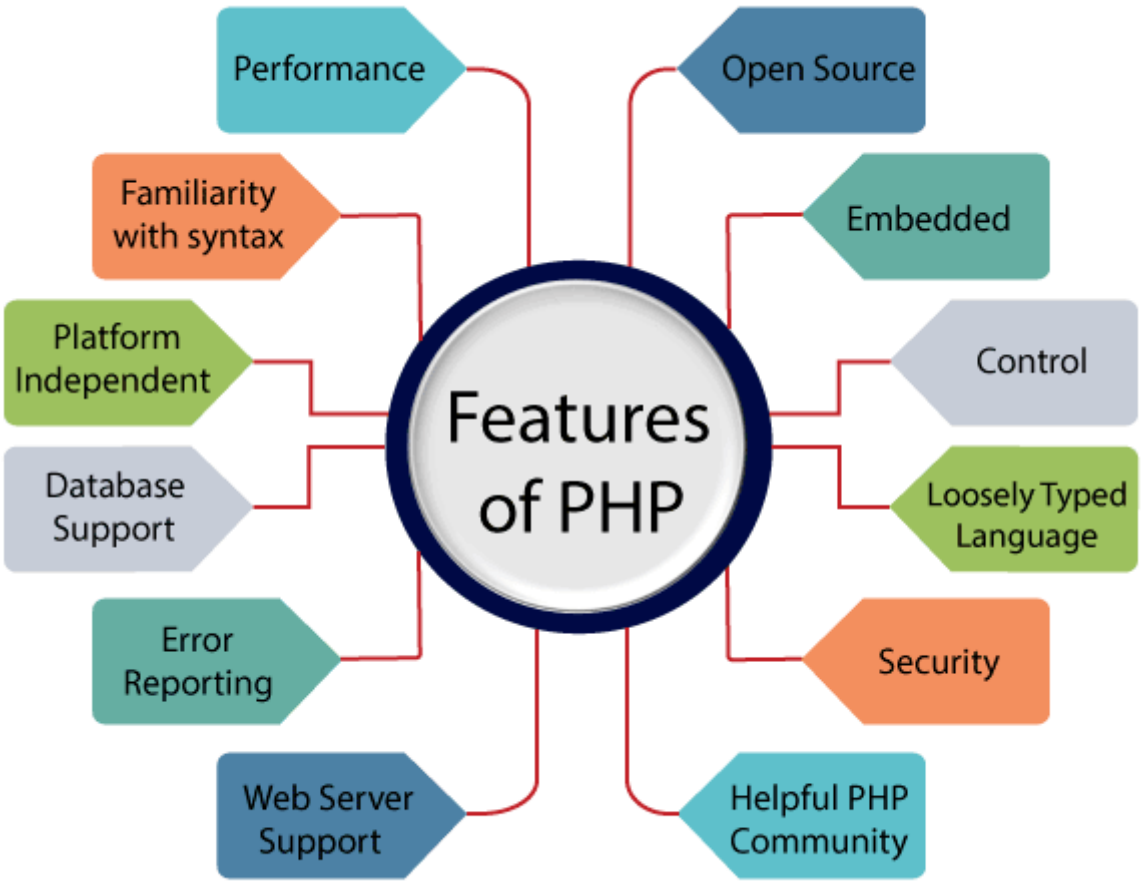
✞ **WHAT IS PHP :**

- PHP was developed by Rasmus Lerdorf . It is freeware.
- It is also a weakly typed, freeform language.
- PHP has since evolved into a powerful server-side markup language with a syntax that resembles a mix between Perl and C.
- PHP is a server-side scripting language designed specifically for the Web.

- Within an HTML page, we can embed PHP code that will be executed each time the page is visited.
- HTML generates the web page with static text and images.
- However the need evolved for dynamic web-based applications, mostly involving database usage. These dynamic usages are fascinated by PHP.
- Other tasks that PHP is especially good at are database access, disk access, networking, and text manipulation.
- PHP is an excellent alternative to such similar programming solutions as Microsoft's proprietary scripting engine ASP and Allaire's rather expensive ColdFusion.
- As mentioned before, PHP is a cross-platform language.

FEATURES OF PHP :

- The main feature of PHP is; it is an open source scripting language so you can free download this and use it.
- PHP is a server-side scripting language. It is an open-source scripting language.
- It is widely used all over the world. It is faster than another scripting language. Some important features of PHP are given below;


Features of PHP
Performance
Familiarity with syntax
Platform Independent
Database Support
Error Reporting
Web Server Support
Open Source
Embedded
Control
Loosely Typed Language
Security
Helpful PHP Community

History of PHP :

- PHP was conceived sometime in the fall of 1994 by Rasmus Lerdorf.
- Early non-released versions were used on his home page to keep track of who was looking at his online resume.
- The first version used by others was available sometime in early 1995 and was known as the Personal Home Page Tools.
- It consisted of a very simplistic parser engine that only understood a few special macros and several utilities that were in common use on home pages back then.
- A guestbook, a counter, and some other stuff. The parser was rewritten in mid-1995 and named PHP/FI Version 2.
- The FI came from another package Rasmus had written which interpreted HTML form data. He combined the Personal Home Page tools scripts with the Form Interpreter and added MySQL support and PHP/FI was born. PHP/FI grew at an amazing pace and people started contributing code to it.
- It is difficult to give any hard statistics, but it is estimated that by late 1996 PHP/FI was in use on at least 15,000 websites around the world
- By mid-1997 this number had grown to over 50,000. Mid-1997 also saw a change in the development of PHP.
- It changed from being Rasmus' pet project that a handful of people had contributed to, to being a much more organized team effort.

- The parser was rewritten from scratch by Zeev Suraski and Andi Gutmans and this new parser formed the basis for PHP Version 3.
- A lot of the utility code from PHP/FI was ported over to PHP3 and a lot of it was completely rewritten.
- Today (end-1999) either PHP/FI or PHP3 ships with several commercial products such as C2's Stronghold web server and RedHat Linux.
- A conservative estimate based on an extrapolation from numbers provided by NetCraft (see also Netcraft Web Server Survey) would be that PHP is in use on over 1,000,000 sites around the world.
- To put that in perspective, that is more sites than run Netscape's flagship Enterprise server on the Internet.
- Also as of this writing, work is underway on the next generation of PHP, which will utilize the powerful Zend scripting engine to deliver higher performance, and will also support running under web servers other than Apache as a native server module.

.

- **HTML**

✞ WHAT IS HTML

- **HTML** stands for **Hyper Text Markup Language**, which is the most widely used language on the Web to develop web pages.
- **HTML** was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995.
- HTML 4.01 was a major version of HTML and it was published in late 1999.
- Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

- Originally, **HTML** was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

✞ Features of HTML

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by HTML anchor tag), so it enhances the interested browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, Macintosh, etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes them more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

- **CSS**

✞ WHAT IS CSS

- **Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML, or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.
- Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braillebased
- tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

- The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.
- The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

2.2 Back End Tool:

• MySQL

MySQL is currently the most popular database management system software used for managing relational databases. It is open-source database software, which is supported by Oracle Company. It is a fast, scalable, and easy-to-use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is ***My Ess Que Ell***. *However, you can pronounce it your way.* Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, macOS, etc. with C, C++, and Java languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables. ○ It allows us to update the table indexes automatically.
- It uses many SQL queries and combines useful information from multiple tables for the end-users.

History of MySQL:

The project of MySQL was started in 1979 when MySQL's inventor **Michael Widenius** developed an in-house database tool called **UNIREG** for managing databases. After that, UNIREG has been rewritten in several different languages and extended to handle big databases. After some time, Michael Widenius contacted **David Hughes**, the author of MySQL, to see if Hughes would be interested in connecting MySQL to UNIREG's B+ ISAM handler to provide indexing to MySQL. That's the way MySQL came into existence.

ANALYSIS

3.1 About Existing System

3.2 Advantages of the proposed system

3.3 Use a case diagram

3.4 DFD (Module-wise)

3.1 Existing System:

- Google Forms is a survey administration software included as part of the free, web-based Google Docs Editors suite offered by Google. The service also includes Google Docs, Google Sheets, Google Slides, Google Drawings, Google Sites, and Google Keep. Google Forms is only available as a web application. The app allows users to create and edit surveys online while collaborating with other users in real-time. The collected information can be automatically entered into a spreadsheet.

- The Google Forms service has undergone several updates over the years. Features include, but are not limited to, menu search, shuffle of questions for randomized order, limiting responses to once per person, shorter URLs, custom themes,[2] automatically generating answer suggestions when creating forms,[3] and an "Upload file" option for users answering questions that require them to share content or files from their computer or Google Drive.

- In October 2014, Google introduced add-ons for Google Forms that enable third-party developers to add new features to surveys,[4] while in July 2017, Google updated Forms to add several new features. "Intelligent response validation" is capable of detecting text input in form fields to identify what is written and ask the user to correct the information if wrongly input. Depending on file-sharing settings in Google Drive, users can request file uploads from individuals outside multi-option answers in a table.

3.2 Advantages and Disadvantages

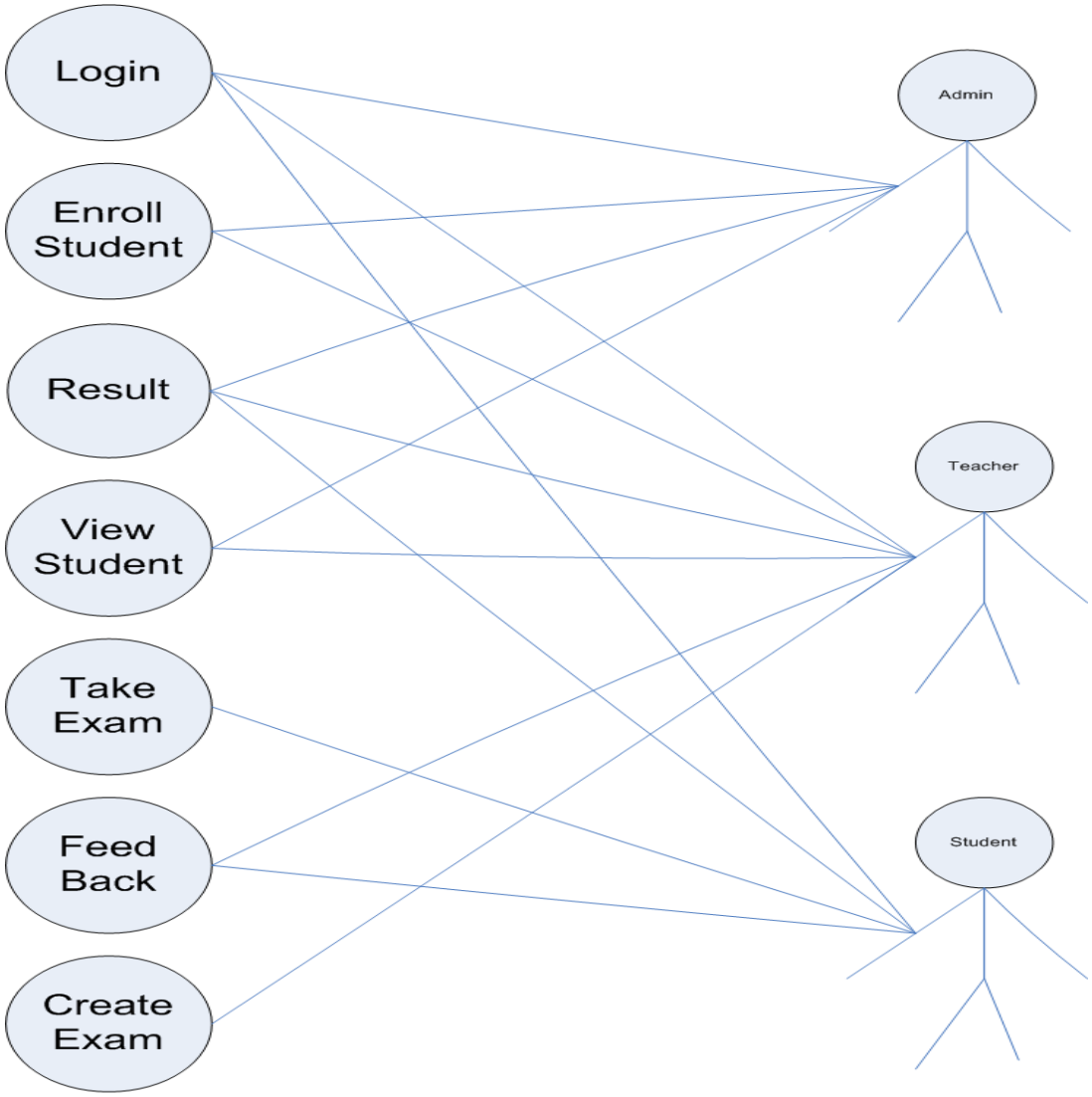
Advantages:

- Provide Fast Paper Creation.
- Easy Interface.
- One Teacher Can Create Multiple Paper.
- Admin Can Manage All Users.
- Student Can Show Result Instantly

Disadvantages:

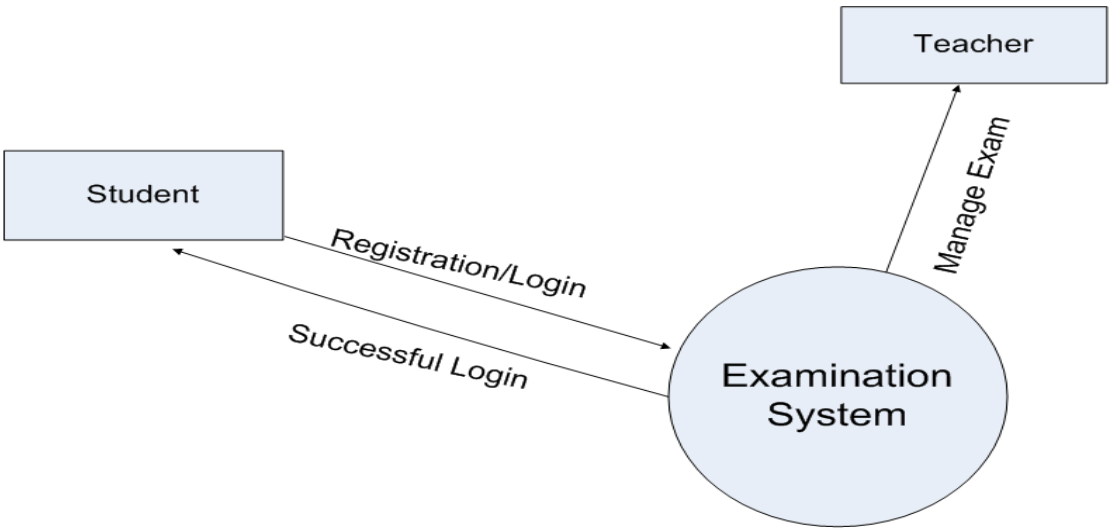
- Only OMR Paper Creation Available.
- One Student Can Fill Paper Many Times.

3.3 Use Case Diagram

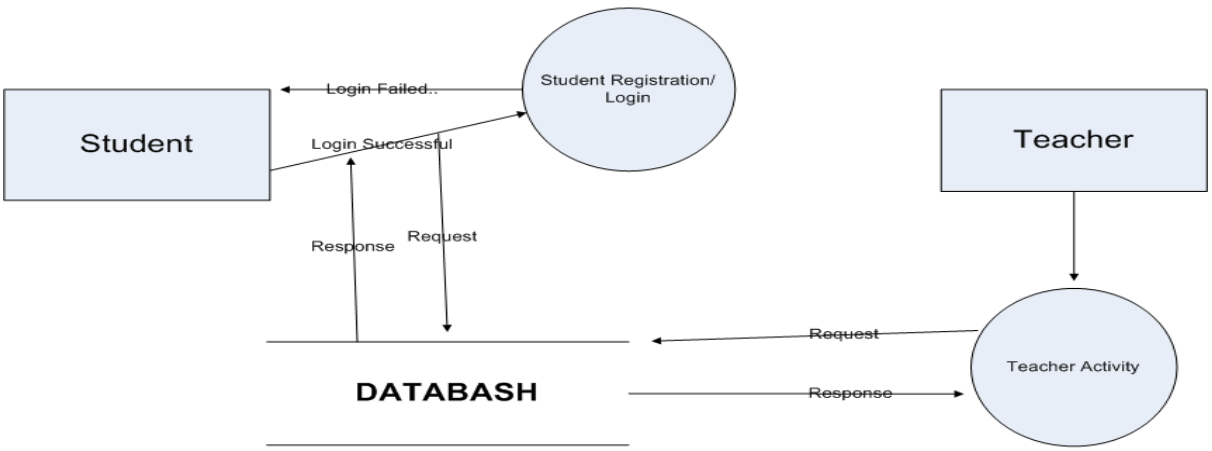


Use Case Diagram

3.5 Data Flow Diagram:



0 Level (DFD)



Level 1 (DFD)

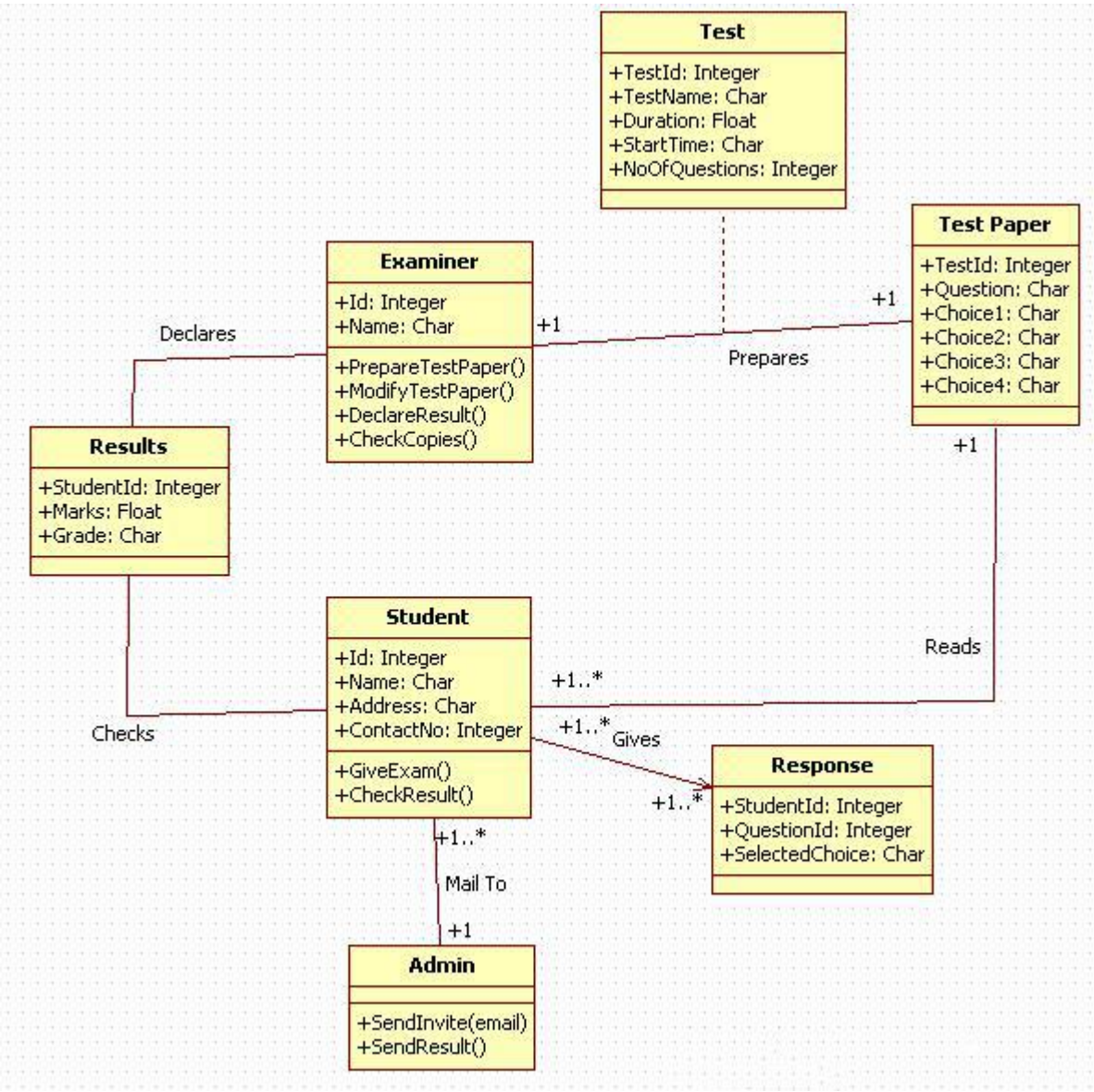
DESIGN

4.1 Class Diagram

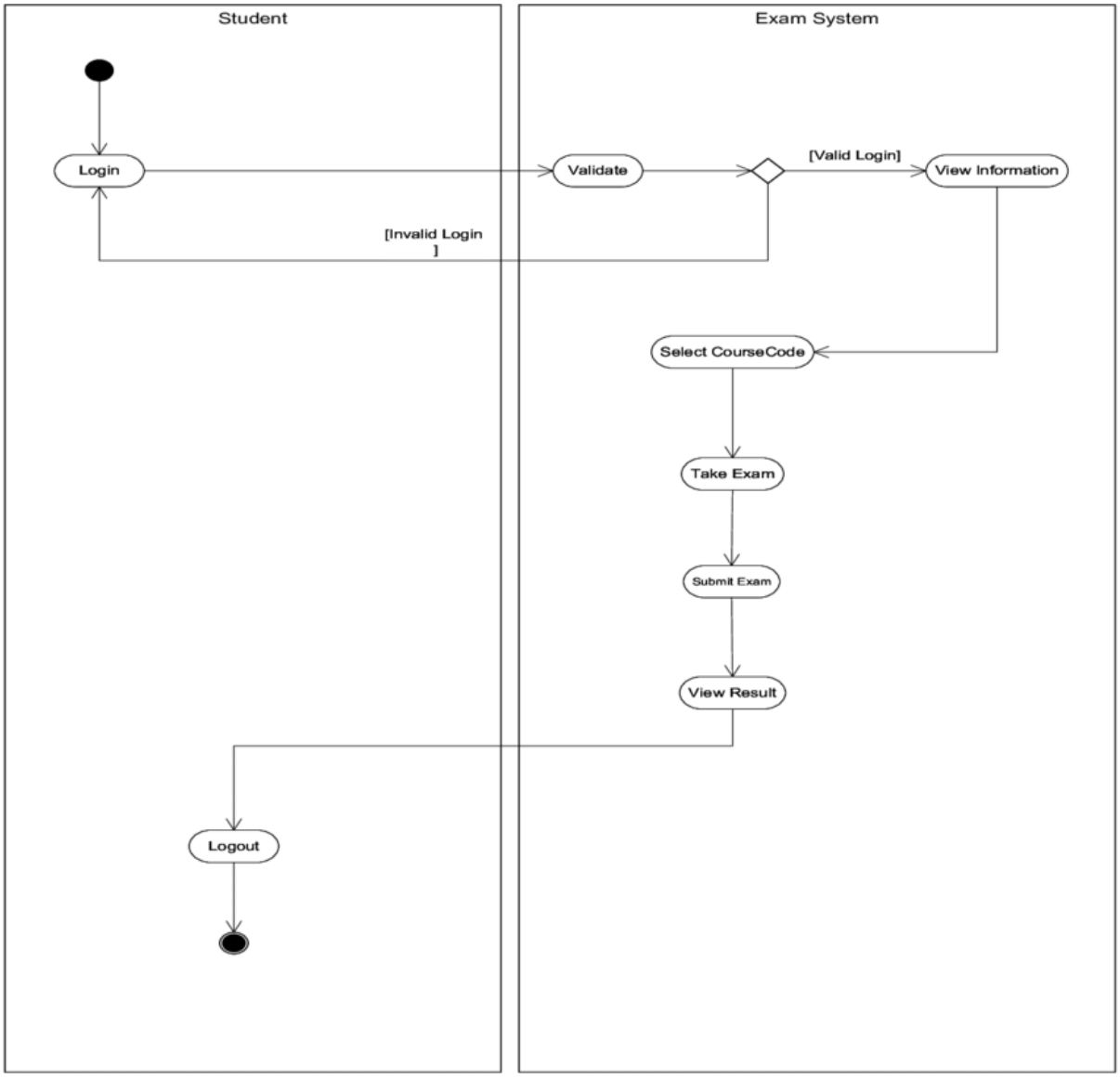
4.2 Activity Diagram

4.3 Data Dictionary

4.1 Class Diagram:



4.2 Activity Diagram:



4.3 Data Dictionary:

admin_info Table:

Primary Key: id

Filed_Name	Data_type	Length	Description
id	int	100	For store admin id
admin_name	varchar	50	For store admin name
admin_email	varchar	50	For store admin email
admin_password	Varchar	50	For store admin password

teacher_info Table:

Primary Key: id

Filed_Name	Data_type	Length	Description
id	int	100	For store teacher id
teacher_name	varchar	50	For store teacher name
teacher_email	varchar	50	For store teacher email
teacher_password	Varchar	50	For store teacher password

student_info Table:

Primary Key: id

Filed_Name	Data_type	Length	Description
id	int	100	For store student id
student_name	varchar	50	For store student name
student_email	varchar	50	For store student email
student_password	Varchar	50	For store student password

feedback Table:

Primary Key: id

Filed_Name	Data_type	Length	Description
id	int	100	For store feedback id
user_name	varchar	50	For store user name
user_email	varchar	50	For store user email
user_mobile	int	50	For store user number
user_message	varchar	100	For store user message

pepar_data Table:

Primary Key:id

Filed_Name	Data_type	Length	Description
id	int	100	For pepar result id
teacher_name	varchar	50	For store teacher name
pepar_name	varchar	50	For store pepar name
pepar_table	Varchar	50	For store pepar table name

result_data Table:

Primary Key:id

Filed_Name	Data_type	Length	Description
id	int	100	For store result id
teacher_name	varchar	50	For store teacher name
result_name	varchar	50	For store result name
result_table	Varchar	50	For store result table name

5.1 Sample Code

5.2 Screen Layout

5.3 Testing

5.1 Sample Code:

Signup System code:

```
//main signup code…
if(isset($_POST['signup_btn']))
{
   $username=$_POST['username'];
   $email=$_POST['email'];
   $password=$_POST['password'];
   $cpassword=$_POST['cpassword'];

   if($password === $cpassword)
   {
      $admin_query="SELECT * FROM admin_info WHERE admin_name='$username' AND admin_password='$password'";
      $admin_query_run=mysqli_query($conn,$admin_query);

      $teacher_query="SELECT * FROM teacher_info WHERE teacher_name='$username' AND teacher_password='$password'";
      $teacher_query_run=mysqli_query($conn,$teacher_query);
```

```
    $student_query="SELECT * FROM student_info WHERE
student_name='$username' AND student_password='$password'";
    $student_query_run=mysqli_query($conn,$student_query);

    if(mysqli_fetch_array($admin_query_run))
    {
      $_SESSION['status']="User Available in Database...";
      $_SESSION['status_code']="error";
      echo
"<script>window.location.replace('http://localhost/project/main_signuppage.php')</
script>";
    }
    elseif(mysqli_fetch_array($teacher_query_run))
    {
      $_SESSION['status']="User Available in Database...";
      $_SESSION['status_code']="error";
      echo
"<script>window.location.replace('http://localhost/project/main_signuppage.php')</
script>";
    }
    elseif(mysqli_fetch_array($student_query_run))
```

```
    {
        $_SESSION['status']="User Available in Database...";
        $_SESSION['status_code']="error";
        echo
"<script>window.location.replace('http://localhost/project/main_signuppage.php')</
script>";
    }
    else
    {
        $query="INSERT INTO student_info
(student_name,student_email,student_password) VALUES
('$username','$email','$password')";
        $run_query=mysqli_query($conn,$query);
        if($run_query)
        {
            $_SESSION['status']="Sign up Successfully...";
            $_SESSION['status_code']="success";
            echo
"<script>window.location.replace('http://localhost/project/main_loginpage.php')</s
cript>";
        }
        else
```

```
        {
            $_SESSION['status']="Email is Taken by Another Person...";
            $_SESSION['status_code']="error";
            echo
"<script>window.location.replace('http://localhost/project/main_signuppage.php')</
script>";
        }
      }
   }
}
```

Login System code:

```
//main login code
if(isset($_POST['login_btn']))
{
```

```
$username=$_POST['username'];
$password=$_POST['password'];

$admin_query="SELECT * FROM admin_info WHERE admin_name='$username' AND admin_password='$password'";
$admin_query_run=mysqli_query($conn,$admin_query);

$teacher_query="SELECT * FROM teacher_info WHERE teacher_name='$username' AND teacher_password='$password'";
$teacher_query_run=mysqli_query($conn,$teacher_query);

$student_query="SELECT * FROM student_info WHERE student_name='$username' AND student_password='$password'";
$student_query_run=mysqli_query($conn,$student_query);

if(mysqli_fetch_array($admin_query_run))
{
  $_SESSION['admin_name']= $username;
  header("Location:admin/dashboard.php");
}
elseif(mysqli_fetch_array($teacher_query_run))
```

```
    {
        $_SESSION['teacher_name']=$username;
        header("Location:teacher/dashboard.php");
    }
    elseif(mysqli_fetch_array($student_query_run))
    {
        $_SESSION['student_name']=$username;
        header("Location:student/dashboard.php");
    } else{
            $_SESSION['status']="Please Sign up and Try Again";
            $_SESSION['status_code']="error";
            echo
"<script>window.location.replace('http://localhost/project/main_loginpage.php')</s
cript>";
    }
}
```

Paper Creation Code:

```
// new pepar add in database
if(isset($_POST['save_pepar_data']))
{

    // random number genretor
    $randomnum=rand(0,100);

    // store session data
    $teachername=$_SESSION['teachername'];
    $title=$_SESSION['pepar_title'];

    unset($_SESSION['teachername']);
    unset($_SESSION['pepar_title']);

    // marge name
    $tb_name=$title.$randomnum;
   // replaceing space in string
   $new_name=str_replace(" ","",$tb_name);
   $new_tb_name=str_replace(".","",$new_name);
```

```
$result_tb_name=$new_tb_name."_result";
$creator_query="CREATE table $result_tb_name(id INT NOT NULL AUTO_INCREMENT,student_name VARCHAR(50),marks int(50),total_marks int(50),pa_fa varchar(50),PRIMARY KEY(ID))";
$result_run_query=mysqli_query($conn,$creator_query);

if(!$result_run_query)
{
    $_SESSION['status'] = "Result table is not Created...";
    $_SESSION['status_code'] = "error";
    header('Location: show_all_pepar.php');
}
$result_query="insert into result_data(teacher_name,result_name,result_table) values ('$teachername','$title','$result_tb_name')";
$run_query=mysqli_query($conn,$result_query);

if(!$run_query)
{
    $_SESSION['status'] = "Data Not Insert in Paper Data Table... ";
    $_SESSION['status_code'] = "error";
```

```
        header('Location: show_all_pepar.php');
    }
     //insert creator pepar data in database
    $creator_query="insert into pepar_data(teacher_name,pepar_title,pepar_table)
values ('$teachername','$title','$new_tb_name')";
    $run_query=mysqli_query($conn,$creator_query);

    if(!$run_query)
    {
        $_SESSION['status'] = "Data Not Insert in Paper Data Table... ";
        $_SESSION['status_code'] = "error";
        header('Location: show_all_pepar.php');
    }

     // create table for a pepar
    $new_tb_query="CREATE table $new_tb_name(id INT NOT NULL
AUTO_INCREMENT,question VARCHAR(200),a VARCHAR(50), b VARCHAR(50), c
VARCHAR(50), d VARCHAR(50), ans VARCHAR(50),PRIMARY KEY(ID))";

     // execute query
    $create_query=mysqli_query($conn,$new_tb_query);
```

```
If (!$create_query) {
   $_SESSION['status'] = "Table Are Not Created... ";
   $_SESSION['status_code'] = "error";
   header('Location: show_all_pepar.php');
} else {
   $question = $_POST['question'];
   $a = $_POST['a'];
   $b = $_POST['b'];
   $c = $_POST['c'];
   $d = $_POST['d'];
   $ans = $_POST['ans'];

   foreach($question as $index => $questions)
   {
      $s_question = $questions;
      $s_a = $a[$index];
      $s_b = $b[$index];
      $s_c = $c[$index];
      $s_d = $d[$index];
      $s_ans = $ans[$index];
```

```
            // insert pepar data in created table
        $query = "INSERT INTO $new_tb_name(question,a,b,c,d,ans) VALUES
('$s_question','$s_a','$s_b','$s_c','$s_d','$s_ans')";
        $query_run = mysqli_query($conn, $query);
      }
      if($query_run)
      {
        $_SESSION['status'] = "Pepar is Created...";
        $_SESSION['status_code'] = "success";
        header('Location: show_all_pepar.php');
      } else{
        $_SESSION['status'] = "Pepar Questions Not Inserted... ";
        $_SESSION['status_code'] = "error";
        header('Location: show_all_pepar.php');
      }
   }
}
```

Paper Checking code:

```
// pepar checking Logic
if(isset($_POST['Submit_pepar_data']))
{

   $ans=$_POST['ans'];
   $pepar_name=$_SESSION['pepar_name'];
   unset($_SESSION['pepar_name']);

   $teacher=$_SESSION['teachername'];
   unset($_SESSION['teachername']);

   $student=$_SESSION['student_name'];



   // for a find test table name
   $query = "SELECT * FROM pepar_data WHERE id='$pepar_name'";
   $query_run = mysqli_query($conn, $query);
   $row=mysqli_fetch_array($query_run);
```

```
$tb_title=$row[2];
$tb_name=$row[3];


// for a total number of question count
$query="SELECT id FROM $tb_name ORDER BY id ";
$run_query =mysqli_query($conn,$query);
if(!$run_query)
{
   echo '<script> alert("not working query")</script>';
}
$pepar_row = mysqli_num_rows($run_query);


// for fetch data  teacher ans
$query = "SELECT * FROM $tb_name";
$query_run2 = mysqli_query($conn, $query);

while($row = mysqli_fetch_assoc($query_run2))
{
```

```
   $ansewer= $row['ans'];
   $ansvar[]=$ansewer;

}

//for fetch student ans
$mark=0;
foreach($ans as $index => $ans_save)
   {

      $s_ans =$ans_save;
      $tb_ans=$ansvar[$index];

      // for a checking module
      if($s_ans===$tb_ans)
      {
         $mark=$mark+1;
      }
```

```
}

// for a pass or fall
$haf_mark=$pepar_row/2;
if($mark>=$haf_mark)
{
   $pa_fa = "PASS";
}
else
{
   $pa_fa="FALL";
}

$name=$tb_name."_result";

$teacher_name=str_replace(" ","",$teacher);
$new_teacher_name=str_replace(".","",$teacher_name);

$new_name=str_replace(" ","",$tb_name);
$new_tb_name=str_replace(".","",$new_name);
```

```
    // inserting data into result table
    $result_query="INSERT into $name (student_name,marks,total_marks,pa_fa)
values('$student','$mark','$pepar_row','$pa_fa')";
    $run_result=mysqli_query($conn,$result_query);

    if($run_result)
    {
       $_SESSION['status'] = "Your Pepar is Submited... Show Result...";
       $_SESSION['status_code'] = "success";
       header('Location: all_pepars.php');
    }
    else
    {
       $_SESSION['status'] = "some error apendes....";
       $_SESSION['status_code'] = "error";
       header('Location: fill_pepar.php');
    }
}
```

Profile Updete Code:

```
//for a update student profile


if(isset($_POST['profile_editbtn']))
{
   $name=$_SESSION['student_name'];
   $username=$_POST['edit_username'];
   $email=$_POST['edit_email'];
   $password=$_POST['edit_password'];
   $cpassword=$_POST['edit_cpassword'];


   if($cpassword === $password)
   {
      $query="update student_info set
student_name='$username',student_email='$email',student_password='$password'
where student_name='$name'";
      unset($_SESSION['student_name']);
      $_SESSION['student_name']= $username;
      $query_run=mysqli_query($conn,$query);
      if($query_run)
```

```
    {
        $_SESSION['status'] = "Profile Updated...";
        $_SESSION['status_code'] = "success";
        header('Location: dashboard.php');
    }
  }
  else
    {
        $_SESSION['status'] = "Profile Not Updated...";
        $_SESSION['status_code'] = "error";
        header('Location: dashboard.php');
    }
}
```

Feedback Sending Code:

```
//for a sent feedback
  if(isset($_POST['SubmitButton']))
  {
    $username=$_POST['name'];
    $email=$_POST['email'];
```

```
    $number=$_POST['mobile'];
    $message=$_POST['message'];

    $query="INSERT into
feedback(user_name,user_email,user_mobile,user_message)
values('$username','$email','$number','$message')";
    $run_query = mysqli_query($conn,$query);
    if($run_query)
    {
      $_SESSION['status'] = "Feedback sent Successfully...";
      $_SESSION['status_code'] = "success";
      header('Location: dashboard.php');
    }
    else
    {
      $_SESSION['status'] = "Feedback Not sent Successfully...";
      $_SESSION['status_code'] = "error";
      header('Location: dashboard.php');
    }

  }
```

Logout system code:

```
//for a logout system logic
if(isset($_POST["admin_logout"]))
{

   unset($_SESSION['admin_name']);
   header("location: main_loginpage.php");
}
if(isset($_POST["teacher_logout"]))
{

   unset($_SESSION['teacher_name']);
   header("location: main_loginpage.php");
}

if(isset($_POST["student_logout"]))
{
  unset($_SESSION['student_name']);
   header("location: main_loginpage.php");
```

}

Profile Modal code:

```
<!-- profile modal -->
<div class="modal fade" id="profileModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel2"
   aria-hidden="true">
   <div class="modal-dialog" role="document">
      <div class="modal-content">
         <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel2">Update Your Profile?</h5>
         </div>
         <?php
         $name = $_SESSION['student_name'];
         $query="select * from student_info where student_name='$name'";
         $query_run = mysqli_query($conn,$query);
         foreach($query_run as $row)
         {
            ?>

         <div class="modal-body">
```

```
<div class="form-group">
  <form method="POST" action="student_code.php">
    <input type="hidden" name="edit_id" class="form-control"
value="<?php echo $row['id']?>"
      placeholder="id">
</div>
<div class="form-group">
  <label> Username </label>
  <input type="text" name="edit_username" class="form-control"
    value="<?php echo $row['student_name']?>" placeholder="Enter
Username" required>
</div>
<div class="form-group">
  <label>Email</label>
  <input type="email" name="edit_email" class="form-control
checking_email"
    value="<?php echo $row['student_email']?>" placeholder="Enter
Email" required>
  <small class="error_email" style="color: red;"></small>
</div>
<div class="form-group">
  <label>Password</label>
```

```
            <input type="text" name="edit_password" class="form-control"
              value="<?php echo $row['student_password']?>" placeholder="Enter
Password" required>
            </div>
            <div class="form-group">
              <label>Confirm Password</label>
              <input type="password" name="edit_cpassword" class="form-control"
placeholder="Confirm Password"
                required>
            </div>  </div>
          <div class="modal-footer">
            <button class="btn btn-secondary" type="button" data-
dismiss="modal">Cancel</button>
            <div class="modal-footer">
              <button type="submit" name="profile_editbtn" class="btn btn-
primary">Update</button>
            </div>
            <?php }?>
            </form>
          </div>
        </div></div></div>
```

Feedback Modal Code:

```
<!-- Feedback Modal-->
<div class="modal fade" id="feedbackModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
  aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Give a Feedback...</h5>
        <button class="close" type="button" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true"></span>
        </button>
      </div>
      <div class="modal-body">
        <form class="user" method="POST" action="student_code.php">
          <div class="form-group row mb-3">
            <div class="col-sm-12 mb-3 mb-sm-0">
              <input type="text" class="form-control" id="name" name="name" placeholder="Enter Name"
                required>
```

```
            </div>
          </div>
          <div class="form-group mb-3">
            <input type="email" class="form-control" id="email" name="email"
placeholder="Email Address"
              required>
          </div>
          <div class="form-group">
            <input type="tel" class="form-control " id="mobile" name="mobile"
placeholder="Mobile Number ">
          </div>
          <div class="form-group mb-3">
            <textArea type="message" class="form-control " id="message"
name="message"
              Placeholder="Enter Feedbcak Message" required></textArea>
          </div>
          <button type="submit" id="SubmitButton" name="SubmitButton"
            class="btn btn-primary btn-user btn-block">
            Submit Feedback
          </button>
        </form>
```

```
        </div>
      </div>
    </div>
</div>
```

Logout Modal Code:

```
<!-- Logout Modal-->
<div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
   aria-hidden="true">
   <div class="modal-dialog" role="document">
      <div class="modal-content">
         <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>
            <button class="close" type="button" data-dismiss="modal" aria-label="Close">
               <span aria-hidden="true"></span>
            </button>
         </div>
         <div class="modal-body">
```

```
            Select "Logout" below if you are ready to end your current session.
          </div>
          <div class="modal-footer">
            <button class="btn btn-secondary" type="button" data-
dismiss="modal">Cancel</button>
            <form action="../logout.php" method="post">
              <button type="submit" class="btn btn-primary" name="student_logout">
                <i class="fas fa-sign-out-alt fa-sm fa-fw mr-2 text-gray-400"></i>
                Logout
              </button>
            </form>
          </div>
        </div>
      </div>
</div>
```

Custom Error message JS Code:

```
<!-- for custom error on js alert -->
   <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
   <?php
      @session_start();
      if(isset($_SESSION["status"]) && $_SESSION["status"]!='')
      {?>
    <script>
      swal({
         title:"<?php echo $_SESSION['status'];?>",
         icon:"<?php echo $_SESSION['status_code'];?>",
         button:"Ok"
      });
    </script>
    <?php
    unset($_SESSION['status']);
    unset($_SESSION['status_code']);
      }
    ?>
```

Dashboard Card Code:

```
<!-- Total Student Card -->
      <div class="col-xl-3 col-md-6 mb-4">
        <div class="card border-left-primary shadow h-100 py-2">
          <div class="card-body">
            <div class="row no-gutters align-items-center">
              <div class="col mr-2">
                <div class="text-xs font-weight-bold text-primary text-uppercase mb-
1">
                  Students</div>
                <?php
                      $query="SELECT id FROM student_info ORDER BY id ";
                      $run_query =mysqli_query($conn,$query);
                      if(!$run_query){
                         echo '<script> alert("not working query")</script>';
                      }
                      $student_row = mysqli_num_rows($run_query);
                  ?>
                <div class="h5 mb-0 font-weight-bold text-gray-800"><?php echo
$student_row;?></div>
              </div>
```

```
            <div class="col-auto">
              <a href="student_registration.php">
                <i class="bi bi-people fa-2x text-gray-300"></i>
              </a>
            </div>
          </div>
        </div>
      </div>
    </div>
```

Footer Code:

```
<!-- Footer -->
<footer class="sticky-footer bg-white">
  <div class="container my-auto">
    <div class="copyright text-center my-auto">
      <span>Copyright By Hitesh Odedara </span>
    </div>
  </div>
</footer>
```

Security Logic Code:

```
//security logic
session_start();
   if(!$_SESSION['student_name'])
   {
      header("location: ../main_loginpage.php");
   }
```

404 Error Page Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple 404 Error Page</title>
  <link rel="stylesheet" href="404style.css">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css">

<style>
  .page_404 {
  padding: 40px 0;
  background: #fff;
  font-family: "Arvo", serif;
 }
 .page_404 img {
  width: 100%;
 }
 .four_zero_four_bg {
  background-image: url(https://cdn.dribbble.com/users/285475/screenshots/2083086/dribbble_1.gif);
  height: 400px;
  background-position: center;
 }

 .four_zero_four_bg h1 {
  font-size: 80px;
```

```
}

.four_zero_four_bg h3 {
  font-size: 80px;
}
.link_404 {
  color: #fff !important;
  padding: 10px 20px;
  background: #244bc0;
  margin: 20px 0;
  display: inline-block;
}
.contant_box_404 {
  margin-top: -50px;
}
</style>

</head>
<body>
 <section class="page_404">
```

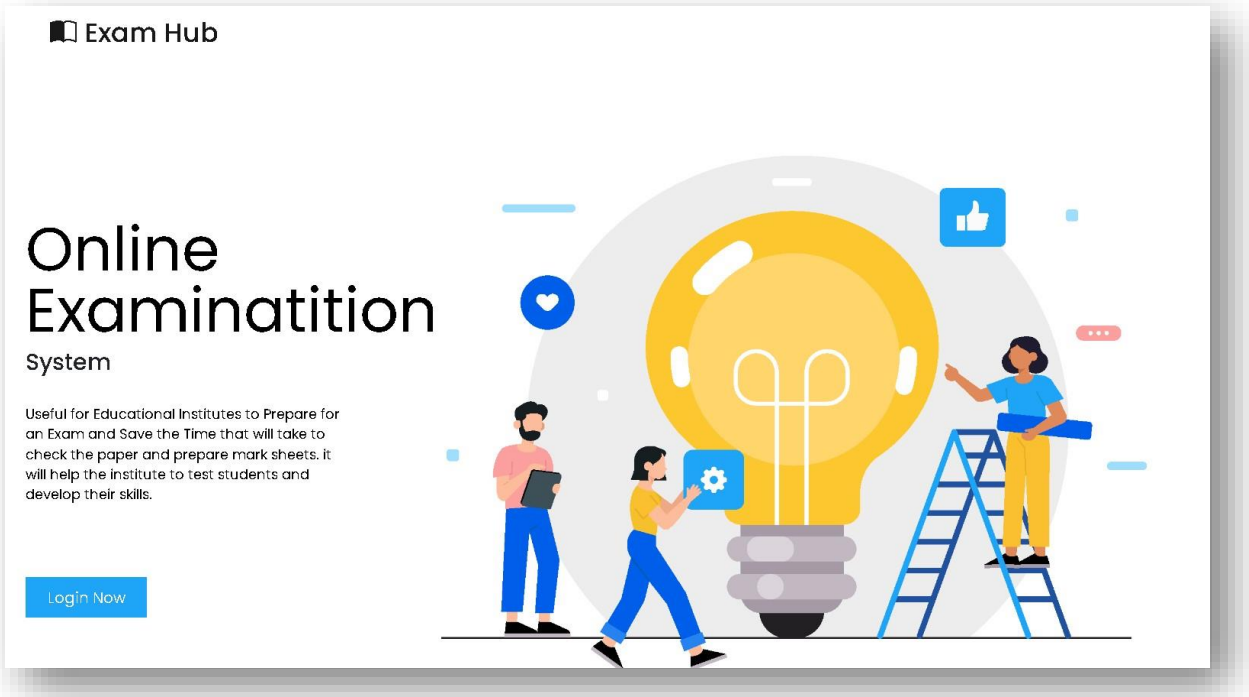
```
<div class="container">
 <div class="row">
  <div class="col-sm-12 ">
   <div class="col-sm-10 col-sm-offset-1  text-center">
    <div class="four_zero_four_bg">
     <h1 class="text-center ">404</h1>
    </div>
    <div class="contant_box_404">
     <h3 class="h2">
      Look like you're lost
     </h3>
     <p>the page you are looking for not avaible!</p>
     <h1>Go Back</h1>
     <!-- <a href="" class="link_404">Go to Home</a> -->
    </div>
   </div></div></div></div>
 </section>
</body>
</html>
```

DataBase Connection String Code:

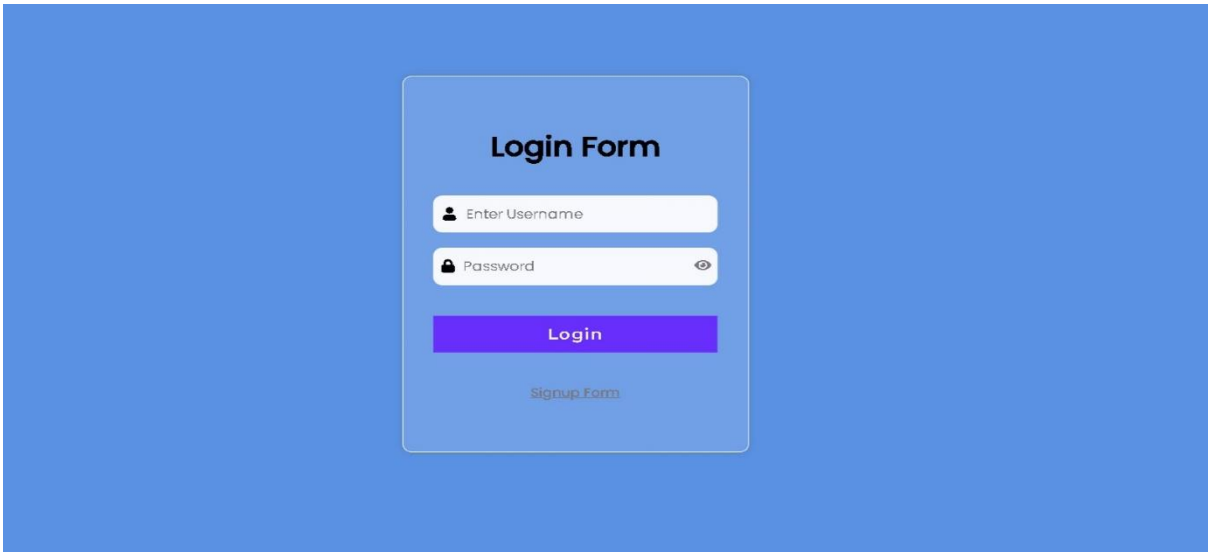
```
<?php
$server_name = "localhost";
$db_username = "root";
$db_password = "";
$db_name = "project";
$conn = mysqli_connect($server_name,$db_username,$db_password,$db_name);
?>
```

5.2 Screen Layout :

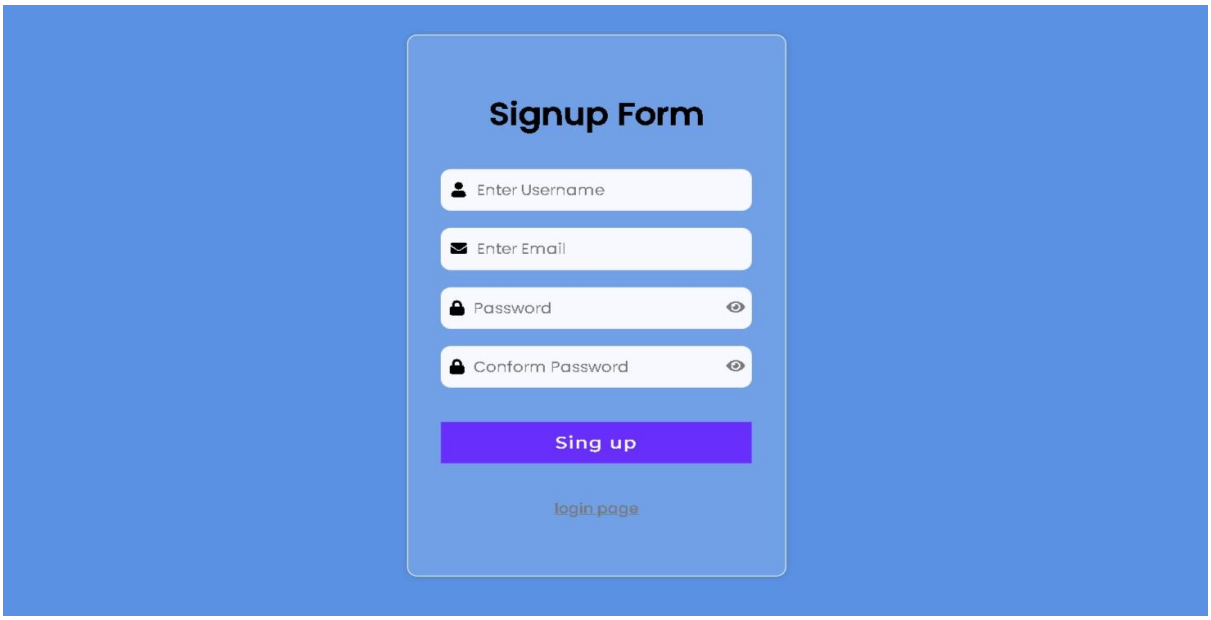
Intro page:



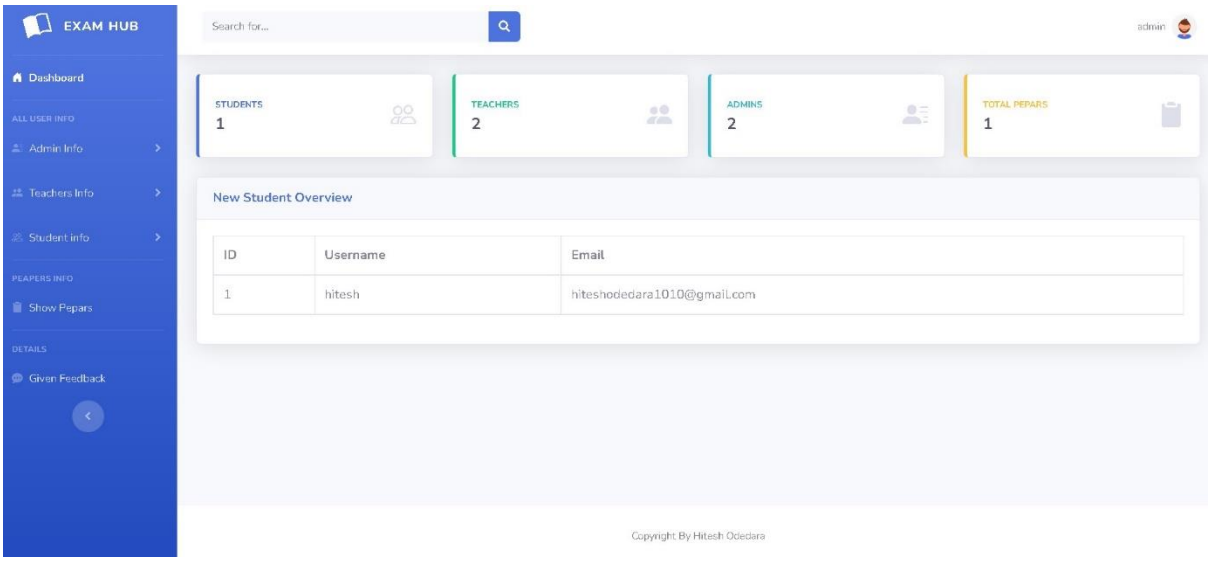
Login Page:

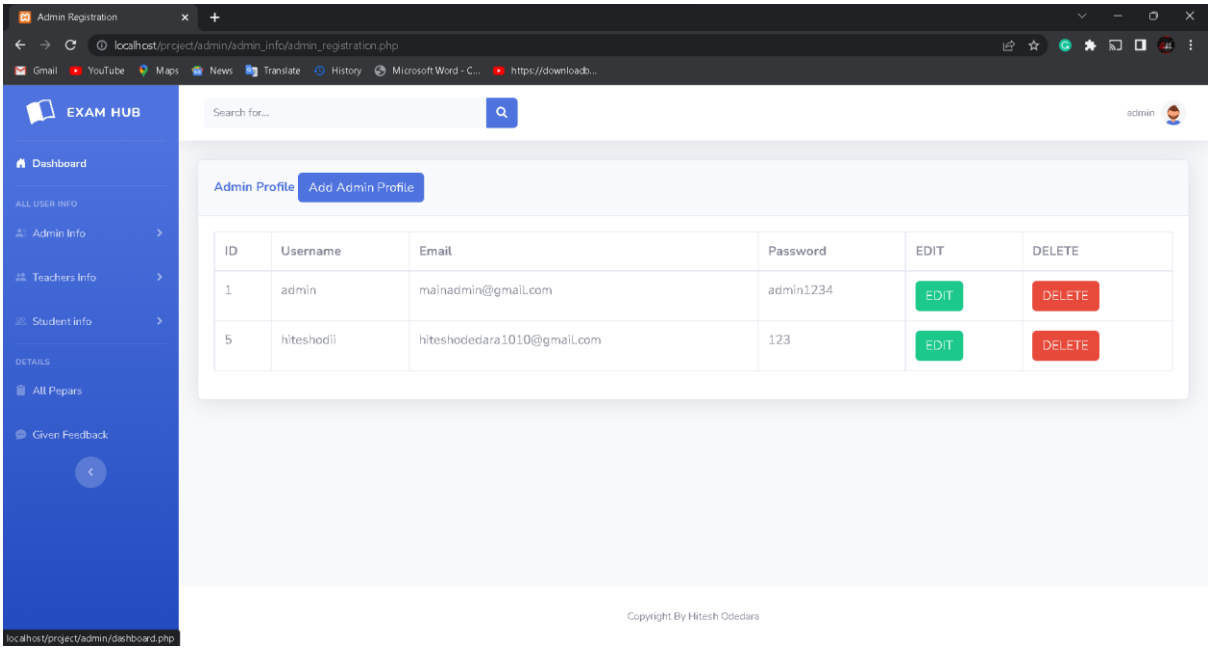


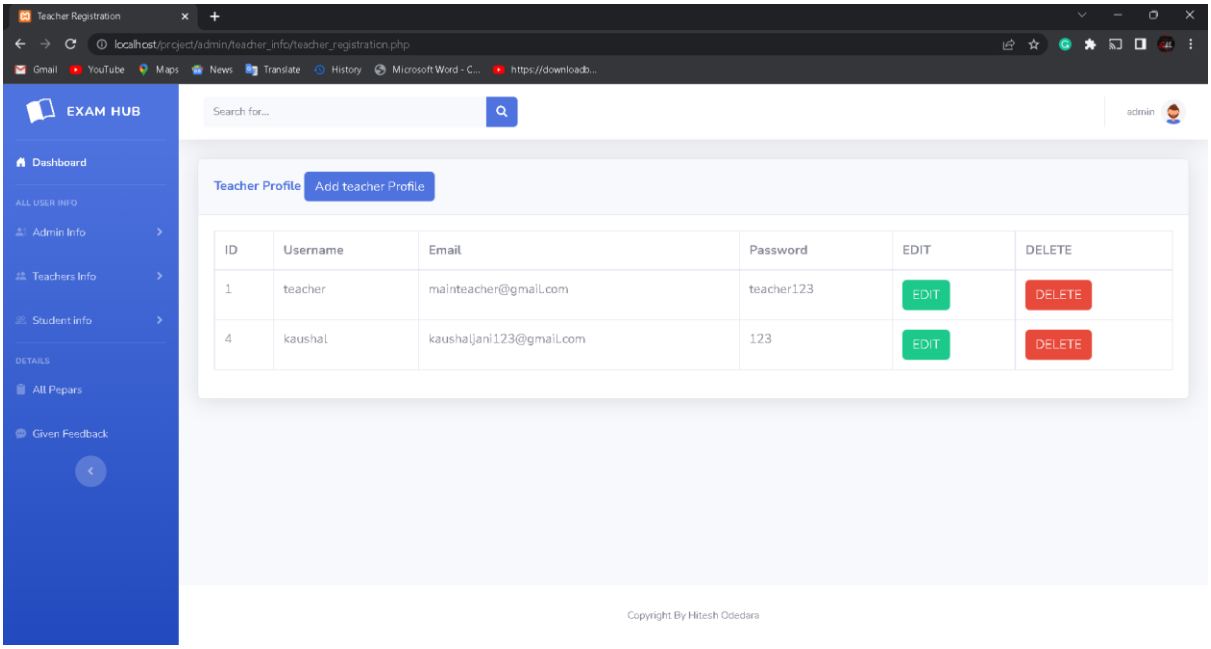
Signup Page:

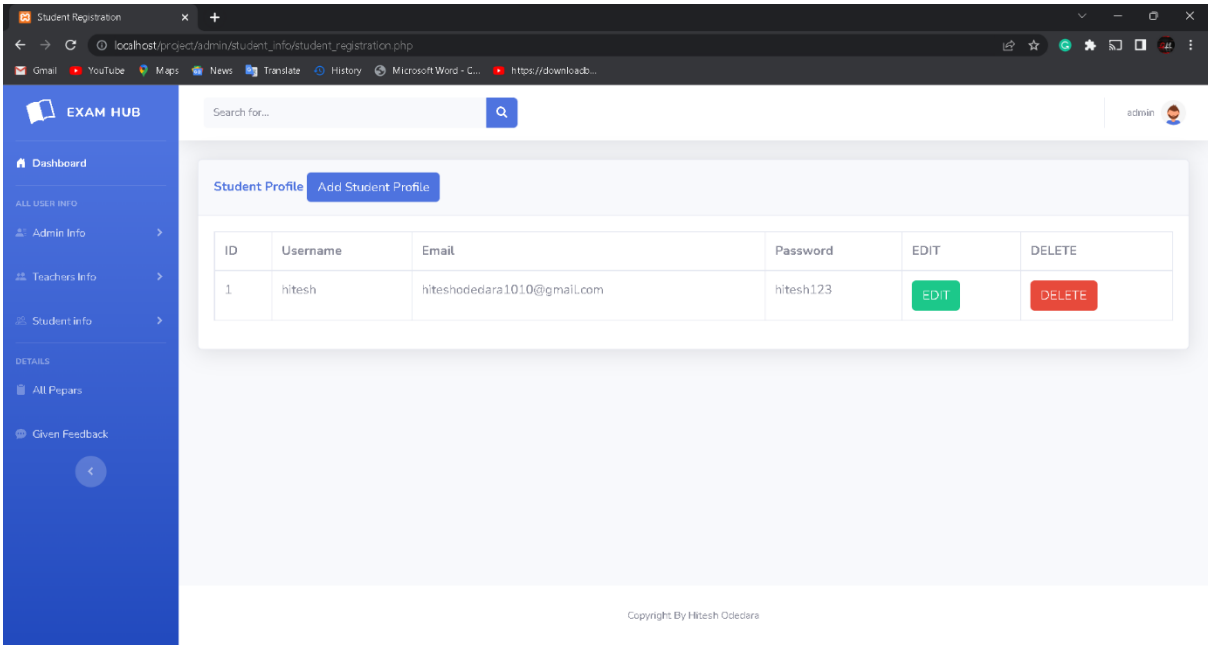


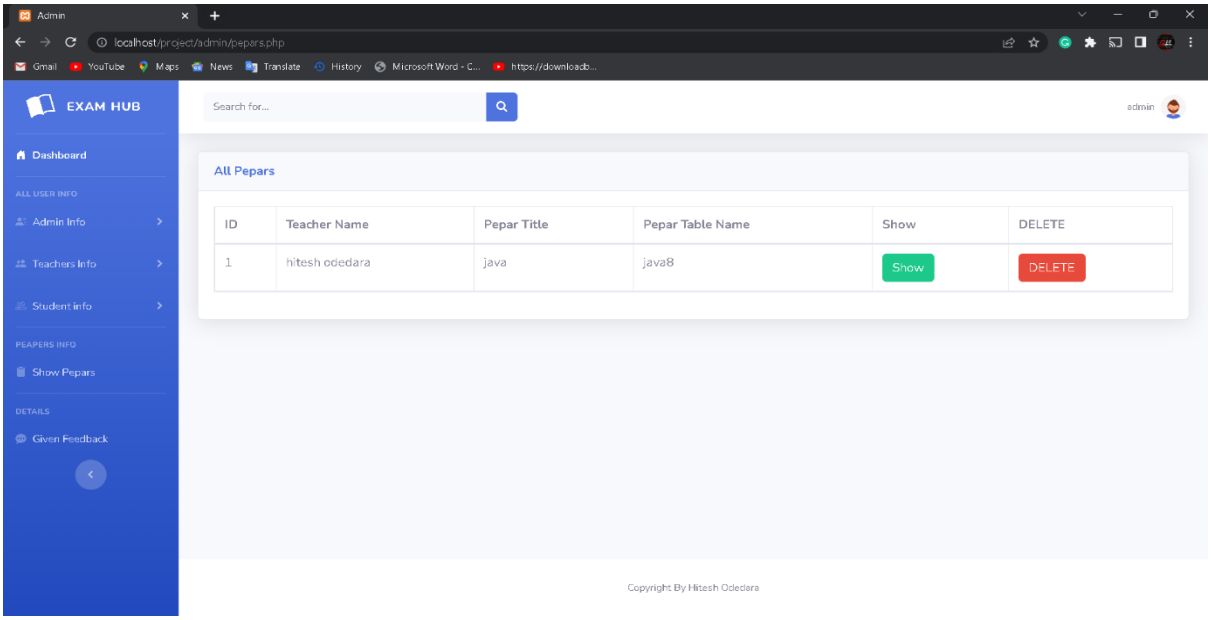
Admin Pannal Pages:

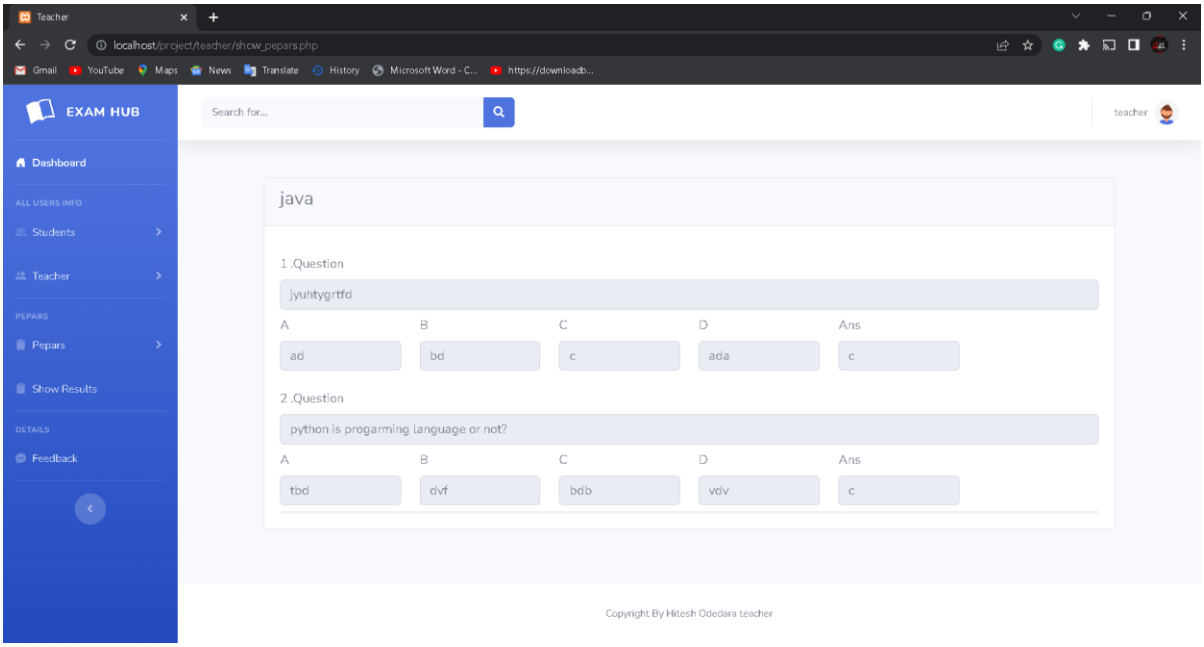




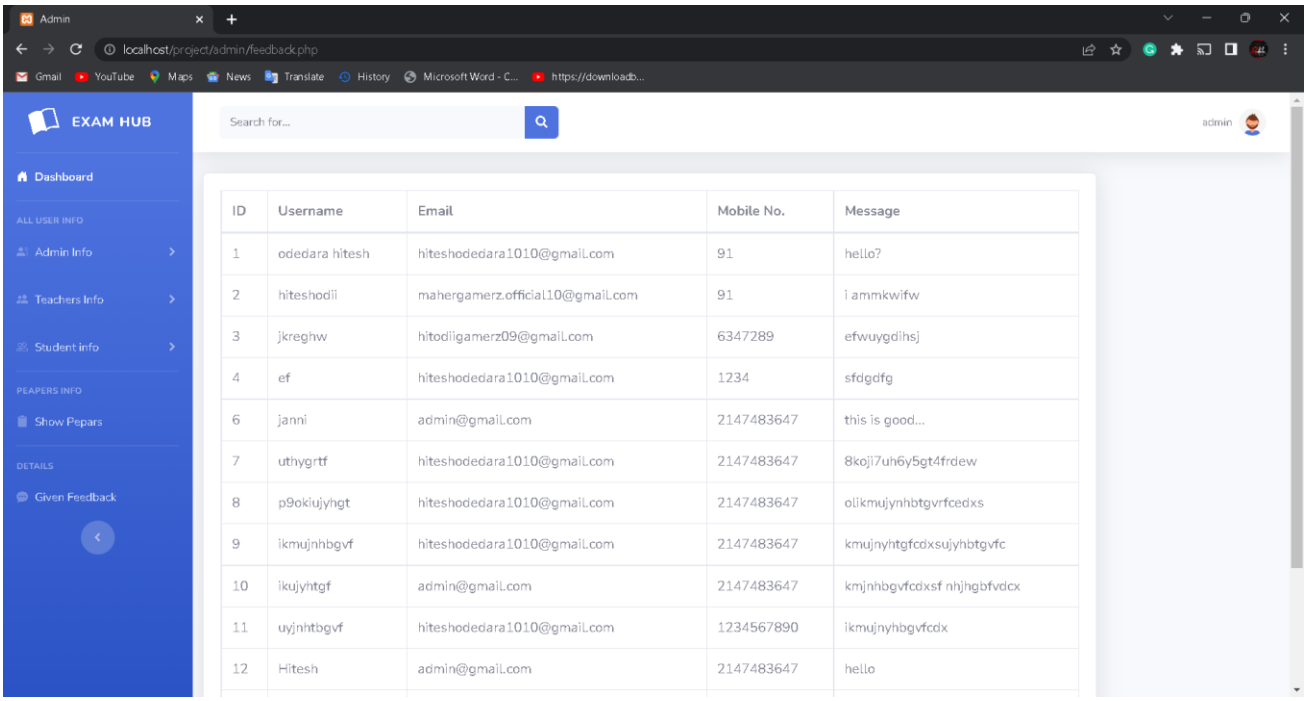

Teacher Registration
localhost/project/admin/teacher_info/teacher_registration.php
EXAM HUB
Search for...
admin
Dashboard
ALL USER INFO
Admin Info
Teachers Info
Student info
DETAILS
All Pepars
Given Feedback
Teacher Profile
Add teacher Profile
ID
Username
Email
Password
EDIT
DELETE
1
teacher
mainteacher@gmail.com
teacher123
EDIT
DELETE
4
kaushal
kaushaljani123@gmail.com
123
EDIT
DELETE
Copyright By Hitesh Odedara


Student Registration
localhost/project/admin/student_info/student_registration.php
EXAM HUB
Search for...
admin
Dashboard
ALL USER INFO
Admin Info
Teachers Info
Student info
DETAILS
All Pepars
Given Feedback
Student Profile
Add Student Profile
ID
Username
Email
Password
EDIT
DELETE
1
hitesh
hiteshodedara1010@gmail.com
hitesh123
EDIT
DELETE
Copyright By Hitesh Odedara

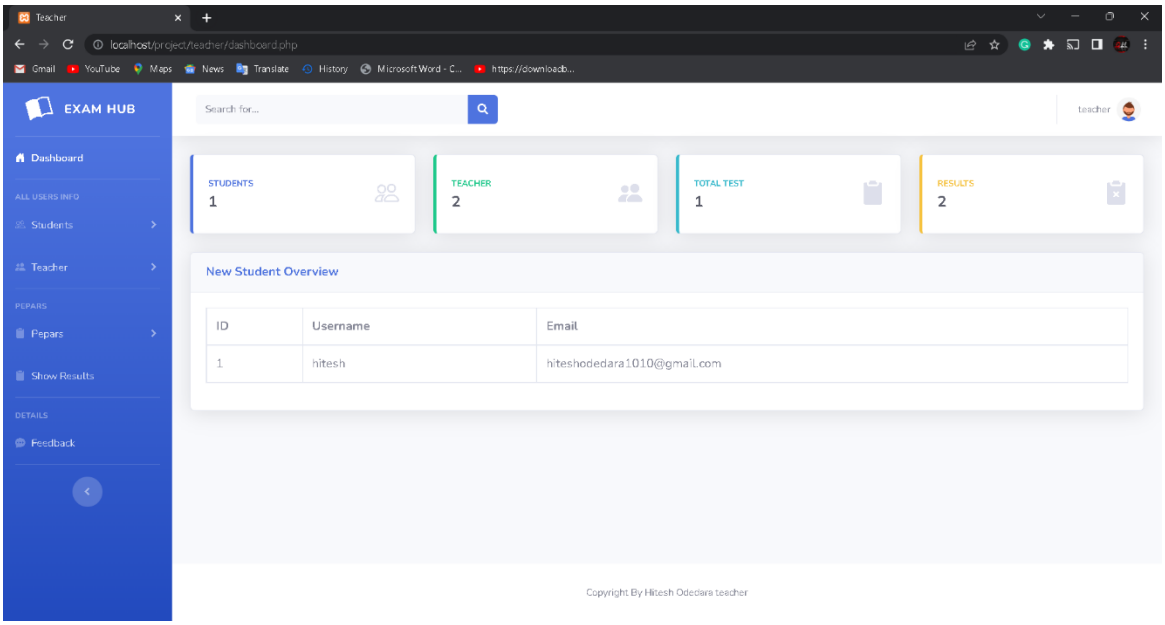

Admin
localhost/project/admin/pepars.php
EXAM HUB
Search for...
admin
Dashboard
ALL USER INFO
Admin Info
Teachers Info
Student info
PEAPERS INFO
Show Pepars
DETAILS
Given Feedback
All Pepars
ID
Teacher Name
Pepar Title
Pepar Table Name
Show
DELETE
1
hitesh odedara
java
java8
Show
DELETE
Copyright By Hitesh Odedara

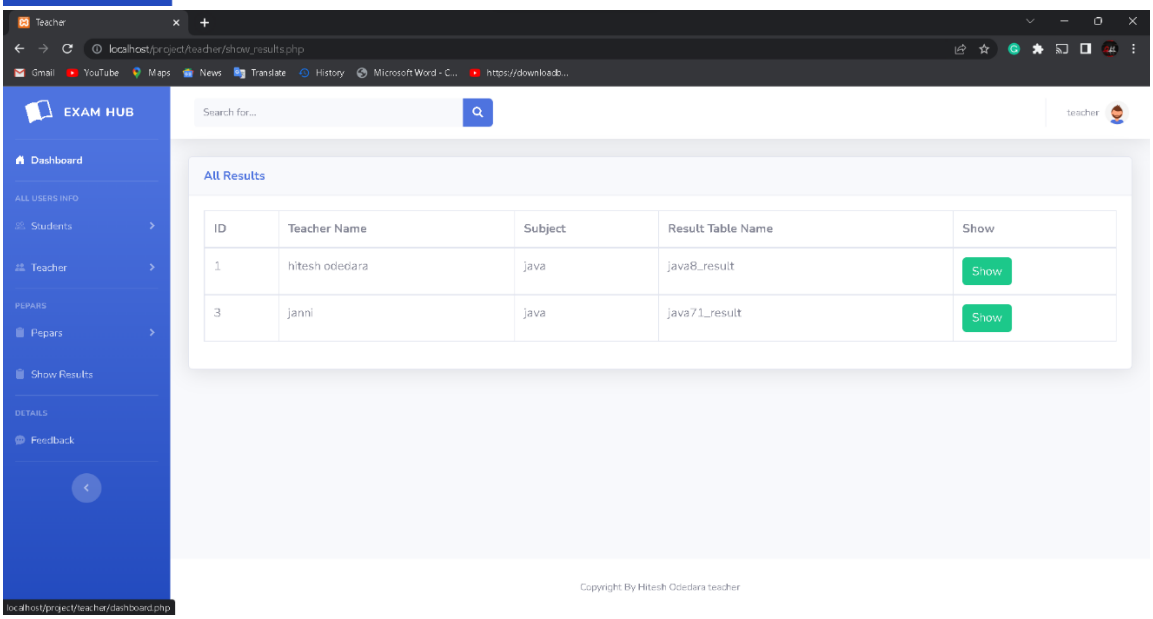

Teacher
localhost/project/teacher/show_pepars.php
EXAM HUB
Search for...
teacher
Dashboard
ALL USERS INFO
Students
Teacher
PEPARS
Pepars
Show Results
DETAILS
Feedback
java
1.Question
jyuhtygrtfd
A
B
C
D
Ans
ad
bd
c
ada
c
2.Question
python is progarming language or not?
A
B
C
D
Ans
tbd
dvf
bdb
vdv
c
Copyright By Hitesh Odedara teacher

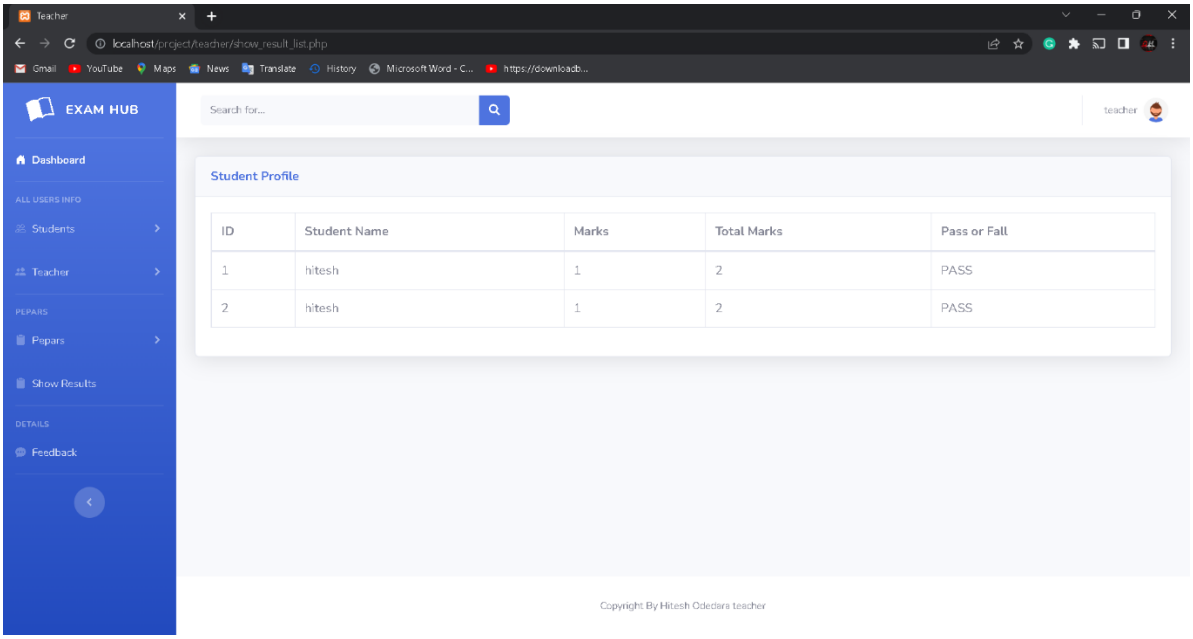
Given Feedbacks Page:

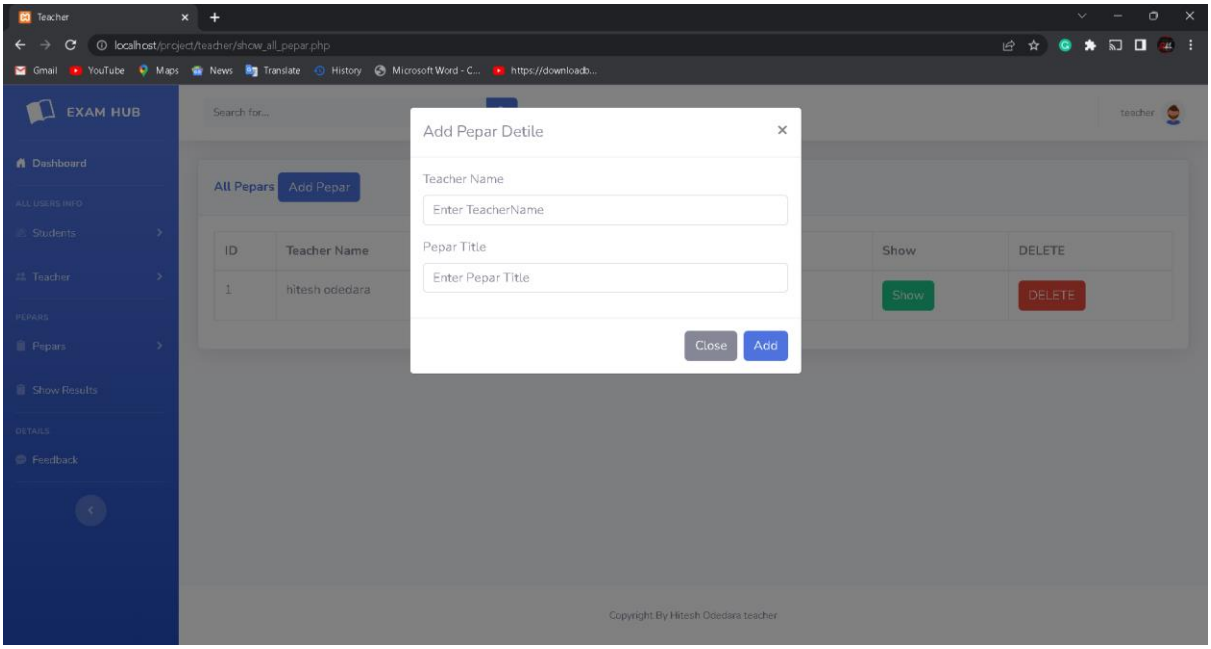


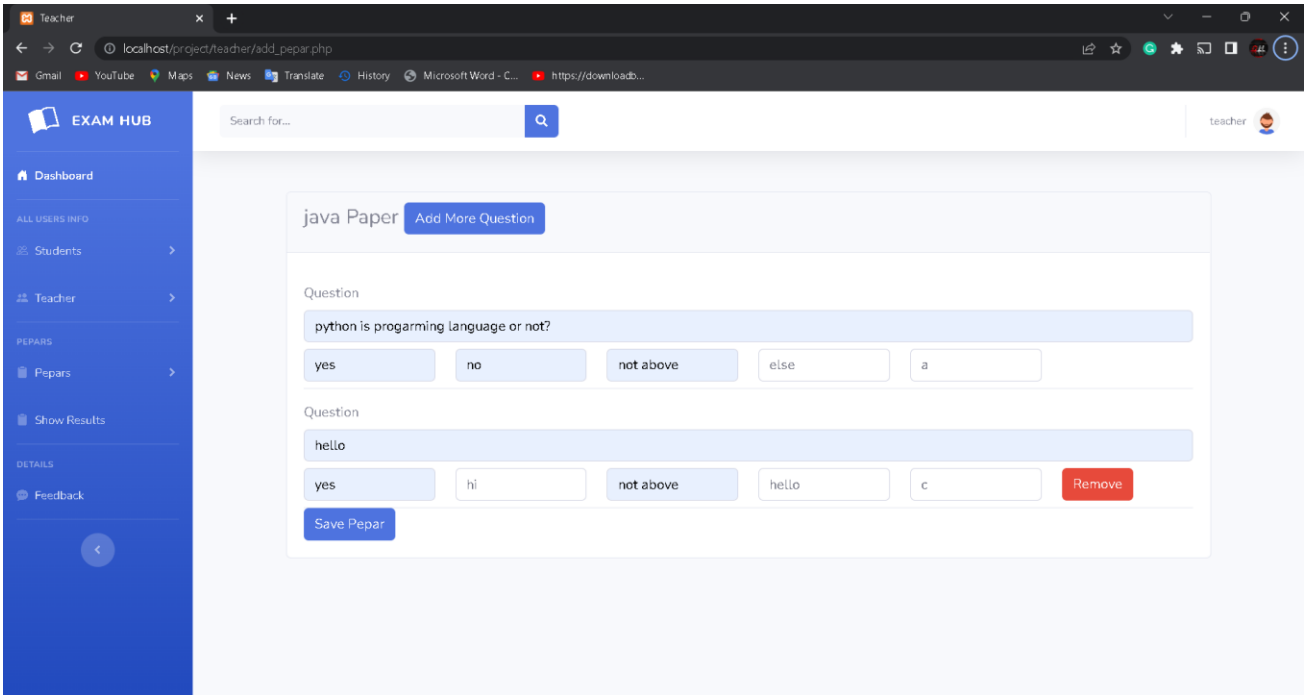
Teacher Pannal Pages:



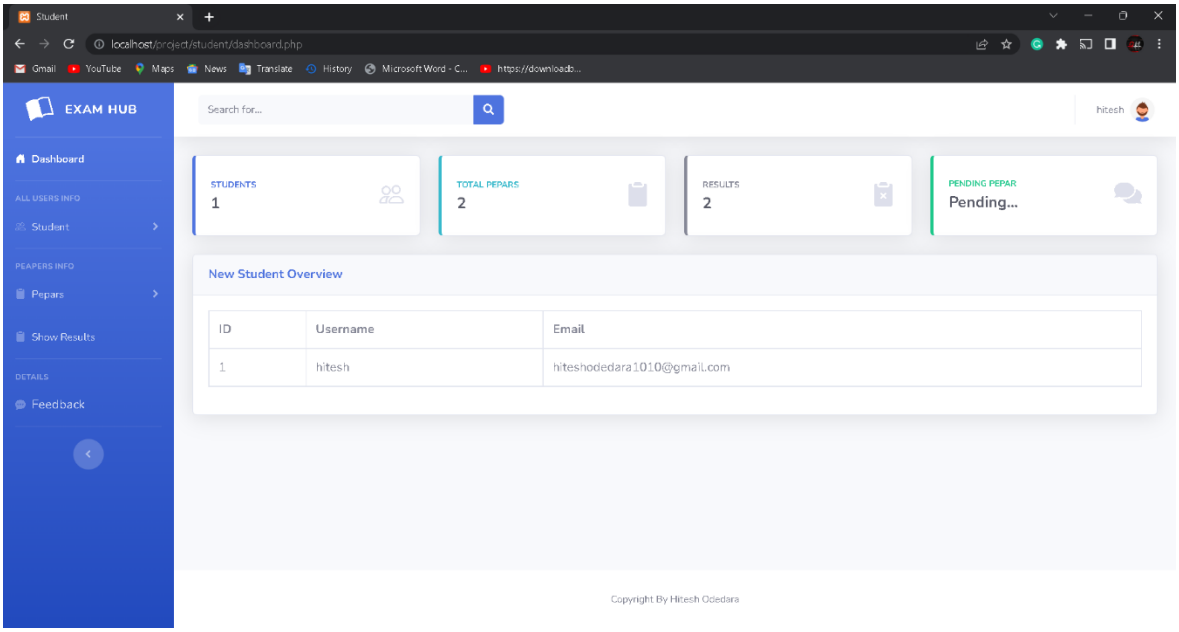


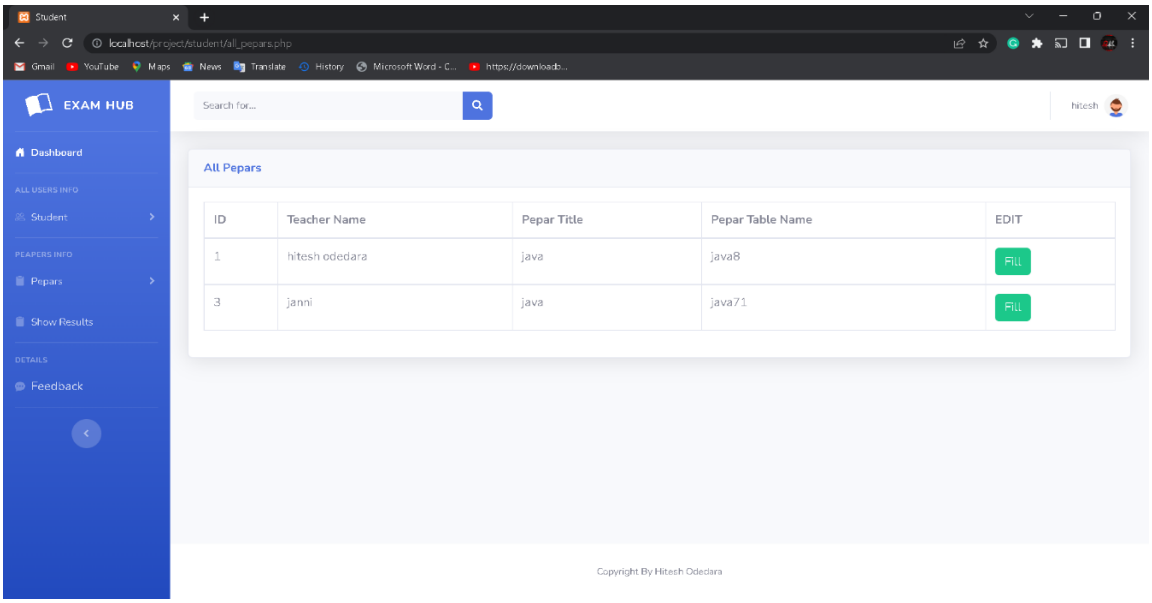

Teacher
localhost/project/teacher/show_result_list.php
EXAM HUB
Search for...
teacher
Dashboard
ALL USERS INFO
Students
Teacher
PEPARS
Pepars
Show Results
DETAILS
Feedback
Student Profile
ID
Student Name
Marks
Total Marks
Pass or Fall
1
hitesh
1
2
PASS
2
hitesh
1
2
PASS
Copyright By Hitesh Odedara teacher

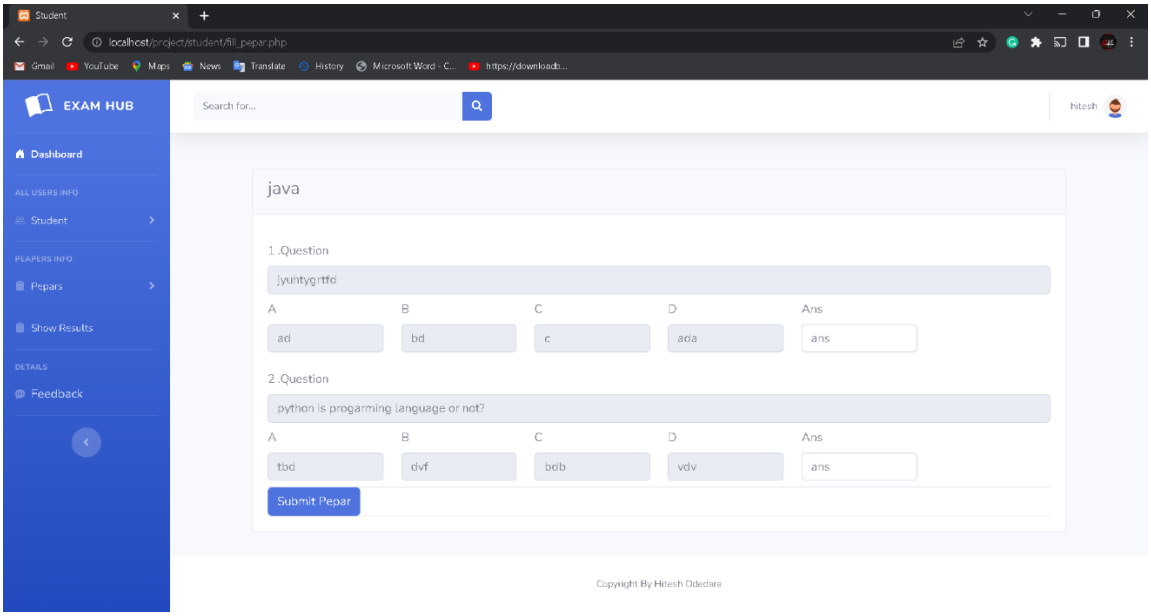

Teacher
localhost/project/teacher/show_all_pepar.php
EXAM HUB
Search for...
teacher
Dashboard
All Pepars
Add Pepar
ID
Teacher Name
Show
DELETE
1
hitesh odedara
Show
DELETE
Add Pepar Detile
Teacher Name
Enter TeacherName
Pepar Title
Enter Pepar Title
Close
Add
Copyright By Hitesh Odedara teacher


Teacher
localhost/project/teacher/add_pepar.php
Gmail
YouTube
Maps
News
Translate
History
Microsoft Word - C...
https://downloadb...
EXAM HUB
Search for...
teacher
Dashboard
ALL USERS INFO
Students
Teacher
PEPARS
Pepars
Show Results
DETAILS
Feedback
java Paper
Add More Question
Question
python is progarming language or not?
yes
no
not above
else
a
Question
hello
yes
hi
not above
hello
c
Remove
Save Pepar

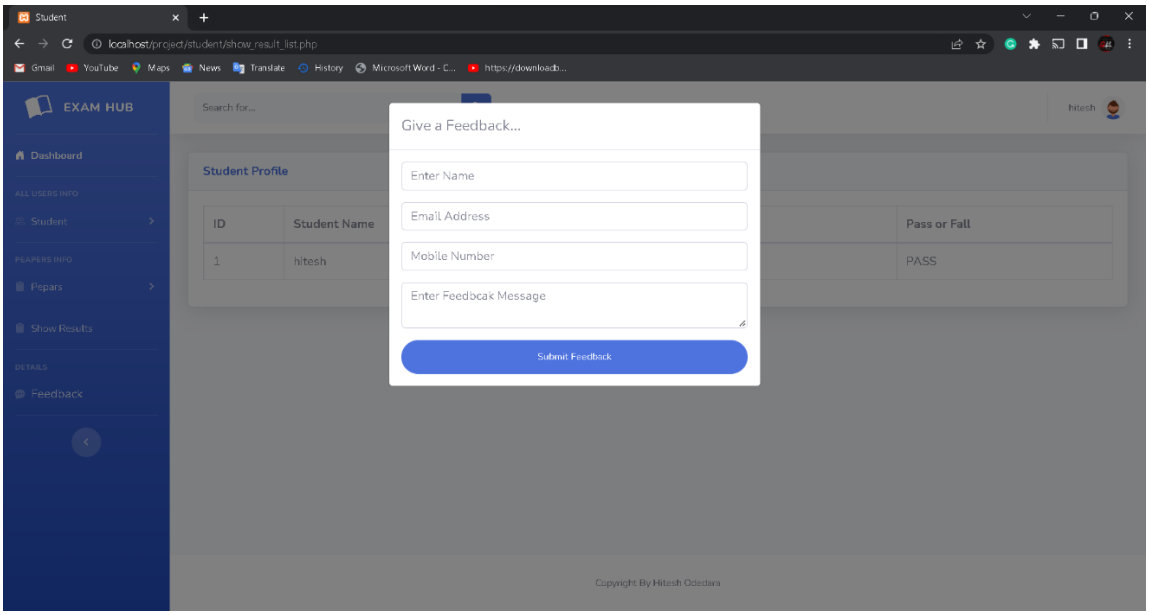
Student Pannal Pages:



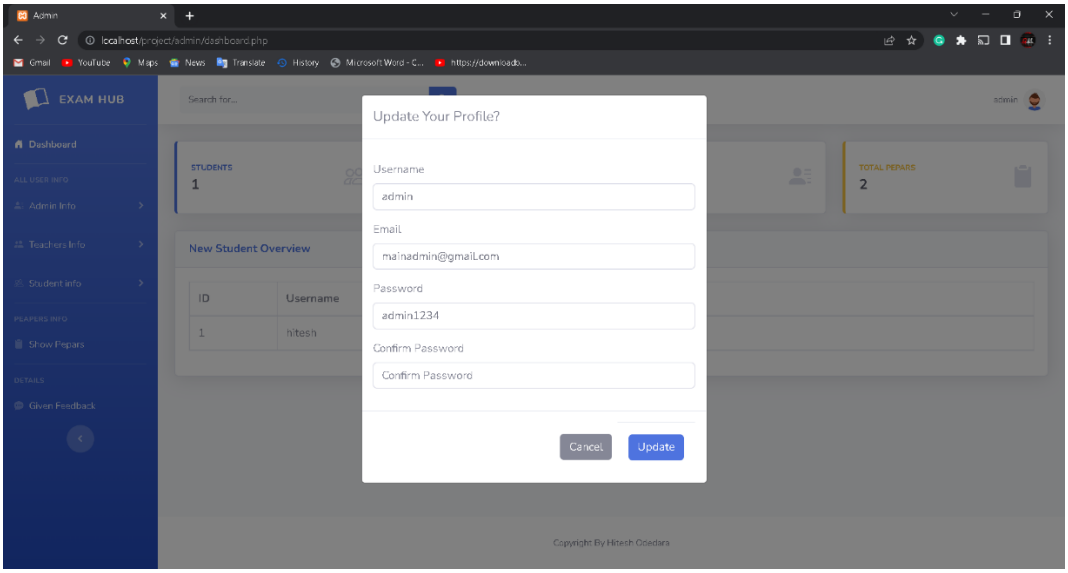




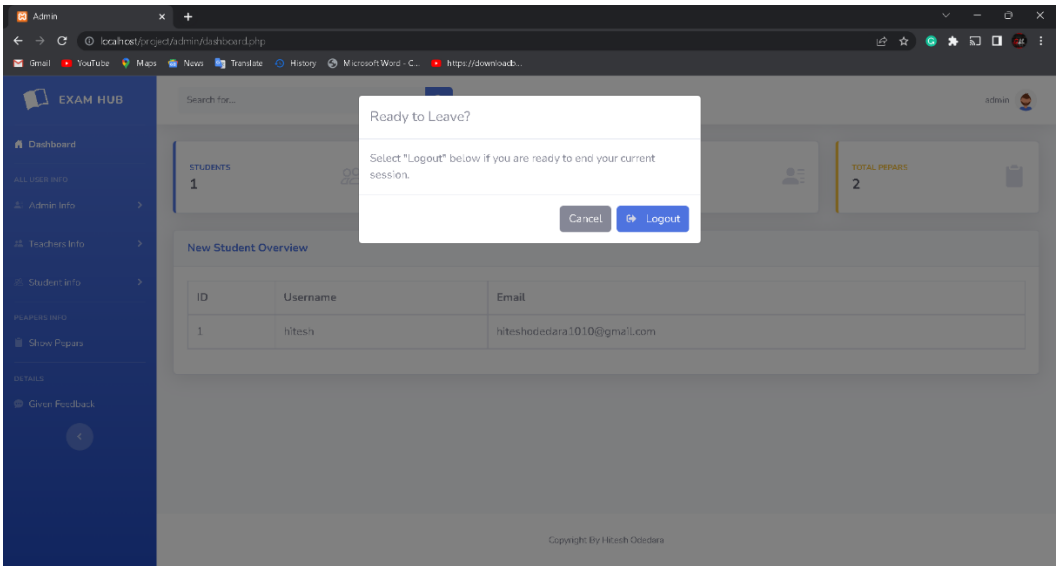
Feedback Model:



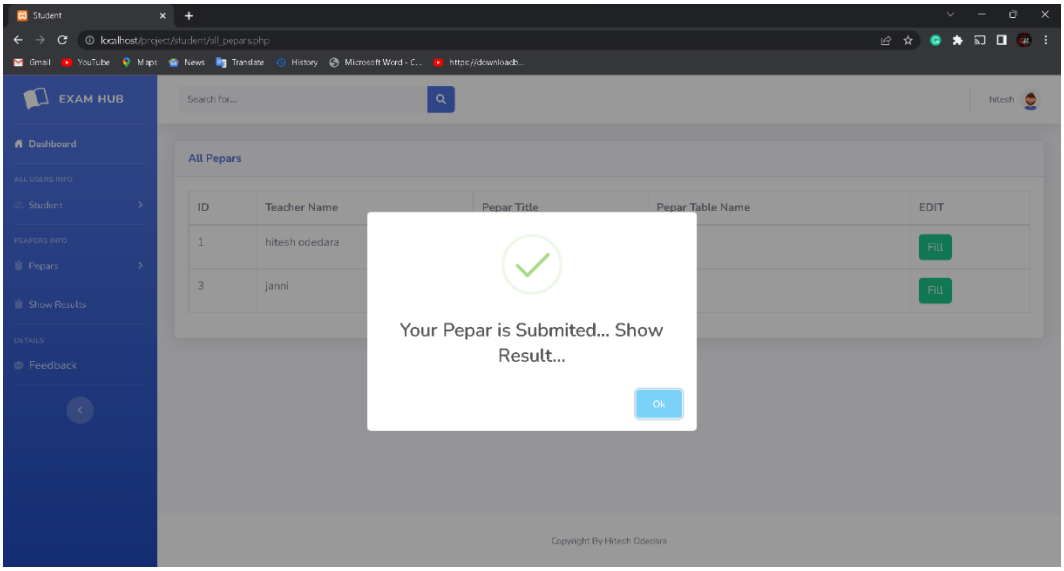
Profile Modal:



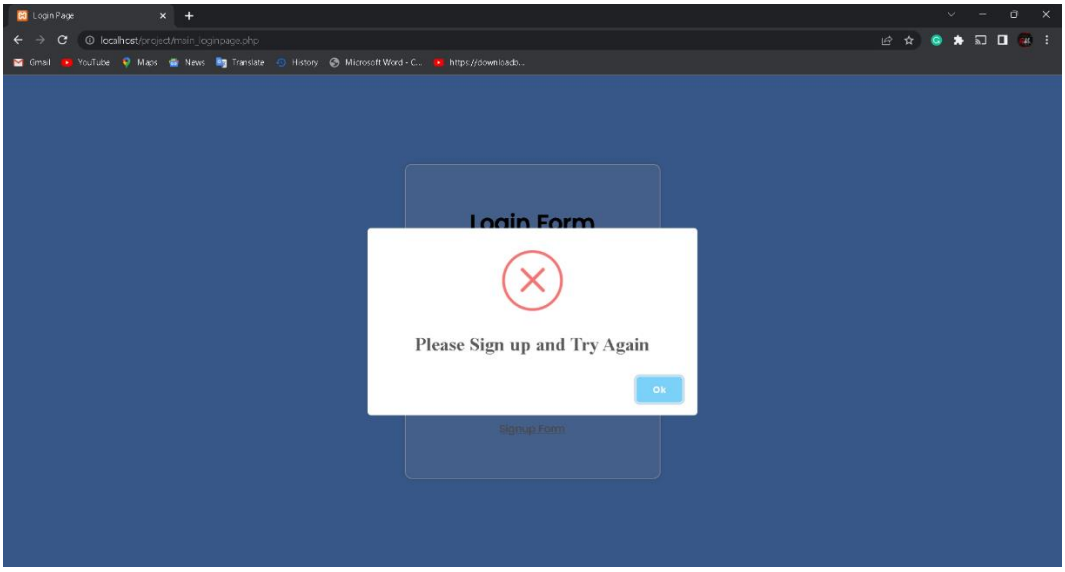
Logout Modal:



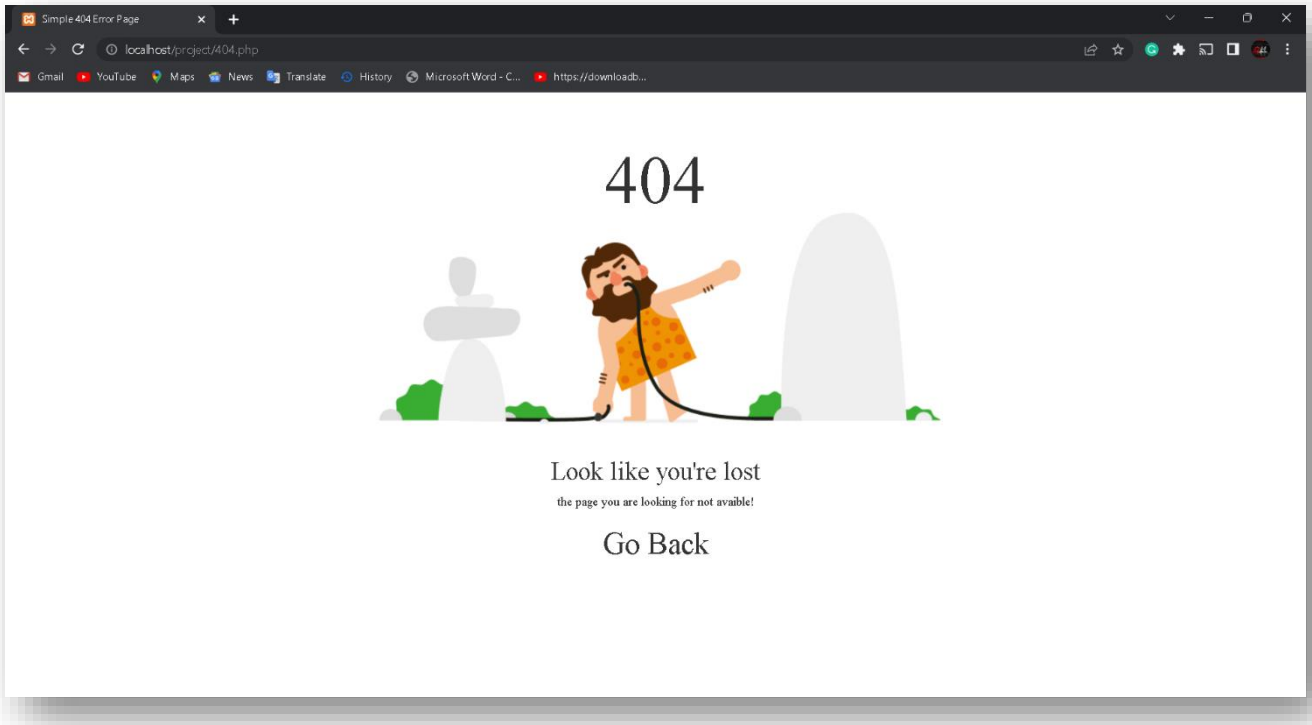
Custom Success Message:



Custom Error Message:



404 Error Page:



Testing and Implementation

5.3 Testing:

Testing is the process of evaluating a system or application, to check whether the application meets all requirements of the client and to detect the errors.

Generally testing can be classified into static testing and dynamic testing.

again, Dynamic Testing is classified into two types: Structural Testing (or) white box, Functional Testing (or) Black Box testing.

Static Testing:

Verification activities fall into the category of Static Testing. Static testing refers to testing something that's not running. It is examining and reviewing it. i.e., to check whether the work done meets the standards of the organization. Reviews, Inspections and Walk-through are static testing methodologies.

Example: The specification is a document and not an executing program.

When we read it to find out the issues, it is considered as static testing.

1.Dynamic Testing:

Dynamic Testing involves working with the software, giving input values and checking if the output is as expected. These are the Validation activities. Unit Tests, Integration Tests, System Tests and Acceptance Tests are few of the Dynamic Testing methodologies.

Techniques used are determined by type of testing that must be conducted.

- Functional ("black box") testing.
- Structural (usually called "white box") testing

Black box testing involves looking at the specifications and does not require examining the code of a program. Tests that examine the observable behaviour of software as evidenced by its outputs without referencing to internal functions is black box testing. It is not based on any knowledge of internal design or code and tests are based on requirements and functionality. Nowadays there are automatic code generation tools and code re-use becomes more prevalent, analysis of source code itself becomes less important and functional tests become more important.

Black box testing is easy as it is based on system's response for a given user's input. If we know the business functionality of the product, we can do black box testing. On the other hand, white box testing requires programming knowledge to know the internals of the code. Also it is time consuming. So only a developer can become a white box tester.

1. Unit Testing:

Soon after the program is corrected for syntax errors, the program has to be checked for logical errors, at the unit level. Programs may have simple user inputs or outputs thru screens or reports. The inputs must be validated for their format, data type, boundary conditions etc. Also, the elementary functionality of the program must be verified. In most of the cases, the programmers themselves will do this unit testing.

If there are any problems while doing this testing, the programmers will Be bug the program and the bugs will be fixed and will be tested again

for the bug fix. It can be tested independently. Unit testing is also termed as component testing.

For example, if we need to test a java class and method, even without any screen, the programmer can test the same. Thus, any piece of code, that can be independently executed, can be independently tested as well. This is termed as unit testing.

3. Integration Testing

When all the individual program units are tested in the unit testing phase and all units are clear of any known bugs, the interfaces between those modules will be tested, toestablish that they communicate to each other property via the specified APIs and thusthey can be integrated into an application. The integration test may be performed by the independent testers or by the development team members

4. **System Testing:**

   After all the interfaces are tested between multiple modules, the whole set of software is tested to establish that all modules work together correctly as an application or system or package. This is again performed by independent testes. The testers will have to do the system testing through they are end users of the application. System testing includes special testing methods like performance testing, interoperability testing, stability testing, etc.

5. **Acceptance Testing**

   After the software product is tested by the SDU in 3 different stages (unit, integration and system), the client will test it, in their place, in a near-realtime or simulated environment.

Implementation

This Phase will provide users with the documentation and training required to use the system effectively. Data Conversion will only occur once, but user documentation will be required. Deployment of the product will be carried out, on the hardware that is going to be used in production (on live systems). Deployment itself requires careful planning. Once the product is deployed, initial data will be populated, user training will happen.

Release to Production and warranty Period

When the clients to the acceptance testing and finds no problems, then they will accept the software and now they have to start using the software in their real office. It may be a real bank for a banking application. During the acceptance testing also, there may be some bug fixed done to the software. Before the software is put into production (real time environment), the SDU must release the latest versions of the software, which has no known bugs to the client.

This may be in a CD or tape or an Internet download. The clients will then install the latest software in their production system and will start using. This is called go-live process.

In the same way every product has a guarantee period software also has a warranty period. Normally it will be a 60-day or a 90-day warranty period for most kinds of software. Depending upon the complexity and size of the software the warranty period may be extended or shortened. During this warranty period, in case of any problems, the SDU has the responsibility to rectify the problem, at no charges. Again, it depends upon the contract, whether to charge for the fix or not and may vary from industry to industry.

Strategic Approach Of Software Testing

Testing is a set of activities that can be planned in advance and conducted systematically. For this reason, a template for software testing--a set of steps into which we can place specific test case design techniques and testing methods--should be defined for the software process.

A number of software testing strategies have been proposed in the literature. All provide the software developer with a template for testing and all have the following generic characteristics:

- Testing begins at the component level and works 'outward' toward the integration of the entire computer-based system

- Different testing techniques are appropriate at different points in time(unit testing, integration testing, system testing, acceptance testing unit testing).

- Testing is conducted by the developer of the software and (for large projects) an independent test group.

- Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code

segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when dead-line pressure begins to rise, progress must be measurable and problems must surface as early as possible.

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design-- the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

Unit Test Considerations

The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (basis paths) through the control structure are exercised to ensure that all statements in a module have been executed at least once. And finally, all error handling paths are tested.Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. In addition, local data structures should be exercised and the local impact on global data should be ascertained (if possible) during unit testing.Selective testing *of* execution paths is an essential task during the unit test. Test cases should be designed to uncover errors due to erroneous computations, incorrect comparisons, and improper control flow. Basis path and loop testing are effective techniques for uncovering a broad array of path errors.

Among the more common errors in computation are:

- Misunderstood or incorrect arithmetic precedence,
- mixed mode operations,
- incorrect initialization
- precision inaccuracy

- Incorrect symbolic representation of an expression.

Test cases should uncover errors such as

- Comparison of different data types,
- incorrect logical operators or precedence,
- expectation of equality when precision error makes equality unlikely,
- incorrect comparison of variables,
- improper or no existent loop termination,
- failure to exit when divergent iteration is encountered
- Improperly modified loop variables.

Test Cases

A **Test case**, in software engineering, is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a *test oracle*. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software

program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as *test scripts*, particularly when written - when they are usually collected into test suites.

We must design tests that have the highest likelihood of finding the most errors with a minimum amount of time & effort.

Software requirements are tested using 'Black Box' test case design techniques.

Black-Box tests are used to demonstrate that:

- Software functions are operational.
- Input is properly accepted.
- Output is correctly produced.
- Integrity of external information (database) is maintained.

White-Box tests can be designed only after a component-level design (or source code) exists. The logical details of the program must be available. The status of the program

is examined at various points to determine if the expected status corresponds with the actual status.

Internal program logic is tested using ‘White Box’ test case design techniques.

Using White-Box testing methods we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

LIMITATIONS AND FUTURE ENHANCEMENT

Limitation:

- Only OMR Paper Creation Available.
- More Security.
- Admin Can Manage Papers.

Future Enhancement:

- True False Paper Creation Available.
- More Security.
- Admin Can Manage Papers.
- Bug fixes and performance.
- More attractive.

BIBLIOGRAPHY

Book Used:

- PHP For Dummies
- PHP Beginners Guide by McGraw-Hill Publication
- JavaScript By McGraw-Hill Publication

Reference Sites:

1. https://www.google.com/
2. YouTube Channel name: https://youtu.be/IqrBr-KT5qk
3. https://getbootstrap.com/docs/5.2/getting-started/introduction/
4. https://stackoverflow.com/
5.