CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Student Management System (SMS) is a web-based application developed using PHP and MySQL. It is designed to streamline the management of student-related activities in schools and educational institutions. The SMS provides an efficient and effective way for both students and school authorities to manage their day-to-day tasks and activities.

The SMS consists of various modules that cover different aspects of student management. The Dashboard module provides an overview of important statistics such as the total number of classes, students, class notices, and public notices. This gives administrators a quick snapshot of the school's current status. The Class module allows administrators to manage classes, including the ability to add, update, and delete class details. This ensures that accurate class information is maintained in the system. The Student module enables administrators to manage student data, including adding new students, updating their information, and deleting records if necessary. This module centralizes all student information, such as personal details, contact information, and academic records, making it easier for administrators to access and manage student data. The Notices module allows administrators to create and manage class notices. The Public Notices module allows administrators to create and manage public notices that can be viewed by all students, staff, and parents. The Pages module allows administrators to manage the content of the about us and contact us pages of the administration. This module provides flexibility in updating and customizing the website's static pages. The Search module enables administrators to search for specific students using their student ID. This helps administrators quickly find and retrieve student information when needed.

The Reports module provides administrators with an overview of student registrations within a specified time period. This module helps administrators track enrollment trends and generate reports for analysis and decision-making purposes. Finally, the SMS allows administrators to update their profiles, change passwords, and recover passwords if forgotten. This ensures that administrators have control over their accounts and can maintain the security of the system.

1.2 PROBLEM STATEMENT

Many educational institutions are still burdened with inadequate, outdated student management systems. These systems, often reliant on manual data entry and paper-based records, present numerous challenges that hinder efficiency, accuracy, and overall student experience.

1.3 MOTIVATION

The motivation behind developing this student management system is to address the limitations of the current manual system, improve efficiency and effectiveness, and provide students, parents, and faculty members with easy access to accurate and real-time information for informed decision-making and support.

1. **Inefficiency of the manual system:**

   The current manual system for managing student information is time-consuming and prone to errors. It requires a lot of paperwork and manual data entry, leading to inefficiency in data management.
2. **Lack of real-time information:**

   The manual system lacks the ability to provide instant updates on student information such as attendance, grades, and academic progress. This hinders timely intervention and support for students who may be struggling academically.
3. **Communication challenges:**

   The manual system does not provide effective communication channels between students, parents, and faculty members. This limits the ability to address concerns, share important information, and foster collaboration.
4. **Data analysis limitations:**

   The manual system does not provide the capability to generate comprehensive reports and perform data analysis. This impedes data-driven decision-making and the ability to identify patterns and trends in student performance.

5. **Cost and resource implications:**

   The manual system requires significant human resources, such as administrative staff, to manage and maintain student records. This incurs costs for the school and limits the allocation of resources to other important areas.

6. **Desire for modern features:**

   There is a need for a student management system that incorporates modern features like real-time updates, communication tools, and data analytics. This would enhance the overall student experience, improve administrative efficiency, and enable data-driven decision-making.

OBJECTIVES

1. **User-Friendly Interface:**

   Make the system easy to use for teachers and students, so they can enter and find information without confusion.

2. **Efficient Attendance Tracking:**

   Use technology to automatically keep track of who attends class, making it accurate and saving time for teachers.

3. **Dynamic Administrator Capabilities:**

   Give school administrators tools to easily change and update student information, adapting to the school's needs over time.

4. **Enhanced Security Measures:**

   Protect student information with strong passwords, keeping it safe from people who shouldn't have access.

5. **Streamlined Leave Management:**

   Make it simple for students to request time off and for teachers to approve it, reducing the workload for everyone.

6. **Improved Overall Efficiency:**

   Use the system to save time and resources, allowing the school to focus more on teaching and less on paperwork.

1.4 PROPOSED SOLUTION

1. **Automation of Administrative Tasks:**

   Automated processes for tasks such as attendance tracking, leave management, and report generation, reducing manual workload and errors.

2. **User Adoption and Satisfaction:**

   High user adoption rates and increased satisfaction among administrators, teachers, and students due to the system's user-friendly design and efficiency.

3. **Accurate and Timely Reporting:**

   Quick and accurate generation of various reports, providing administrators with real-time insights into student performance, attendance, and other key metrics.

4. **Enhanced Parental Engagement:**

   Improved communication channels facilitate increased parental involvement, providing parents with timely updates on their child's academic progress and overall well-being.

5. **Adaptability to Changing Needs:**

   A system designed to adapt seamlessly to changes in student details and evolving administrative requirements, ensuring long-term relevance and usefulness.

6. **Streamlined Student Information:**

   A well-organized and easily accessible repository of student information, allowing for efficient management and retrieval of records.

7. **Improved Communication:**

   Enhanced communication channels between teachers, administrators, and students, fostering transparency and timely information dissemination.

ADVANTAGES

1. **Centralized Information:**

   Having all important information in one central location makes it easier for the admin to access and stay updated on the status of the institution.

2. **Efficient Class Management:**

   The ability to easily add, update, or delete classes helps in organizing and managing the various subjects and divisions within the institution.

3. **Streamlined Student Management:**

   Efficiently managing student details ensures that the admin has accurate information and can easily communicate with students or track their progress.

4. **Notice Management**:

   The ability to manage notices ensures that important information is effectively communicated to relevant parties in a timely manner.

5. **Public Notice Management:**

   Managing public notices separately allows for the dissemination of important information to a wider audience or the general public when necessary.

6. **Page Management:**

   Updating and maintaining the "About Us" and "Contact Us" pages ensures that visitors or potential stakeholders have access to up-to-date and accurate information about the institution.

7. **Efficient Search Functionality:**

   Being able to search for students by their student ID saves time and effort in locating specific students and accessing their details.

8. **Reports:**

   The ability to generate reports on student enrollment provides valuable insights and helps in tracking the growth or progress of the institution.

9. **Profile Management:**

   Updating their own profile information helps the admin maintain the integrity and accuracy of their details within the system.

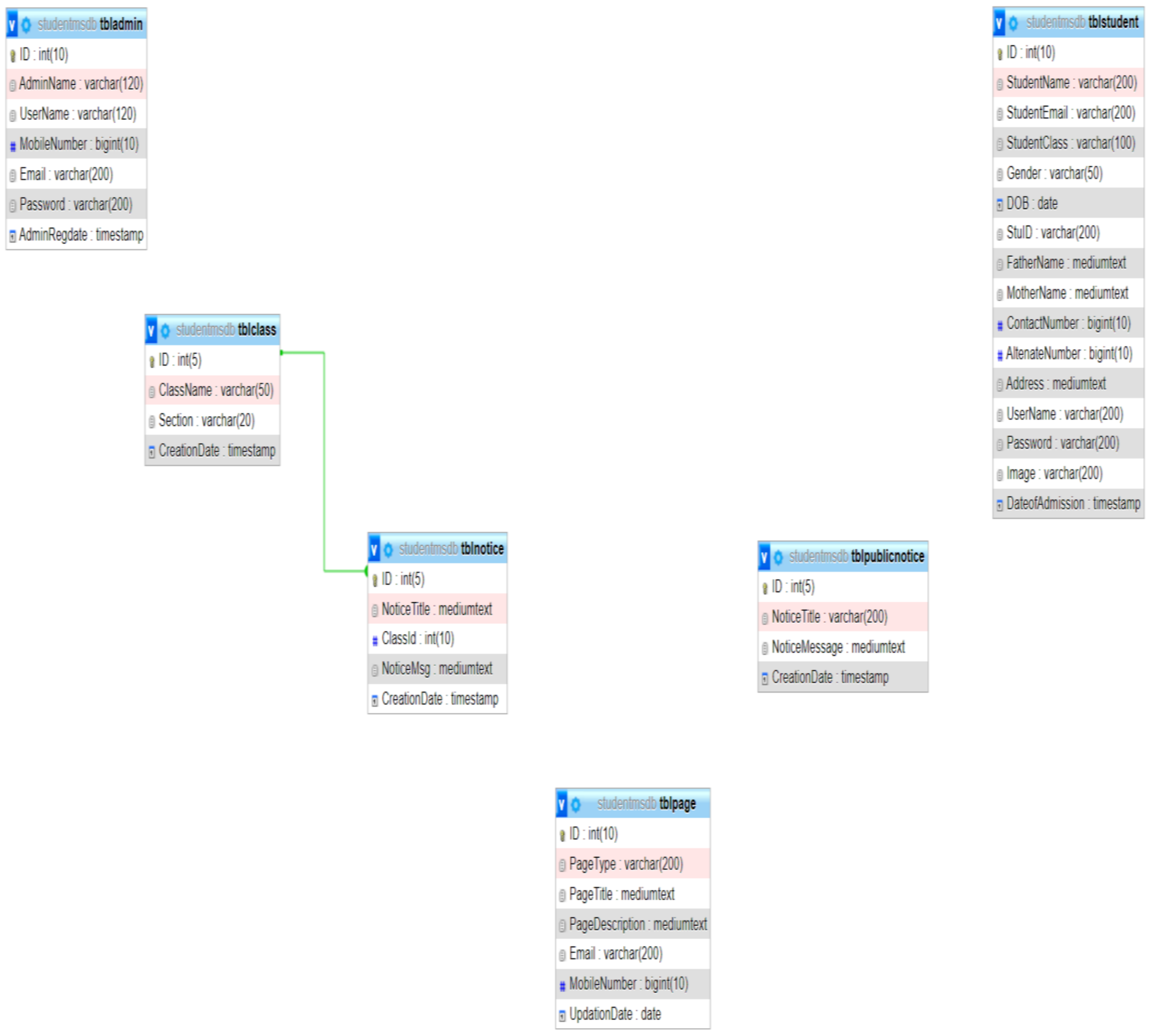
10. **Security and Recovery:**

    The ability to change and recover their password adds an extra layer of security to the admin's account, ensuring that only authorized individuals have access to the system.

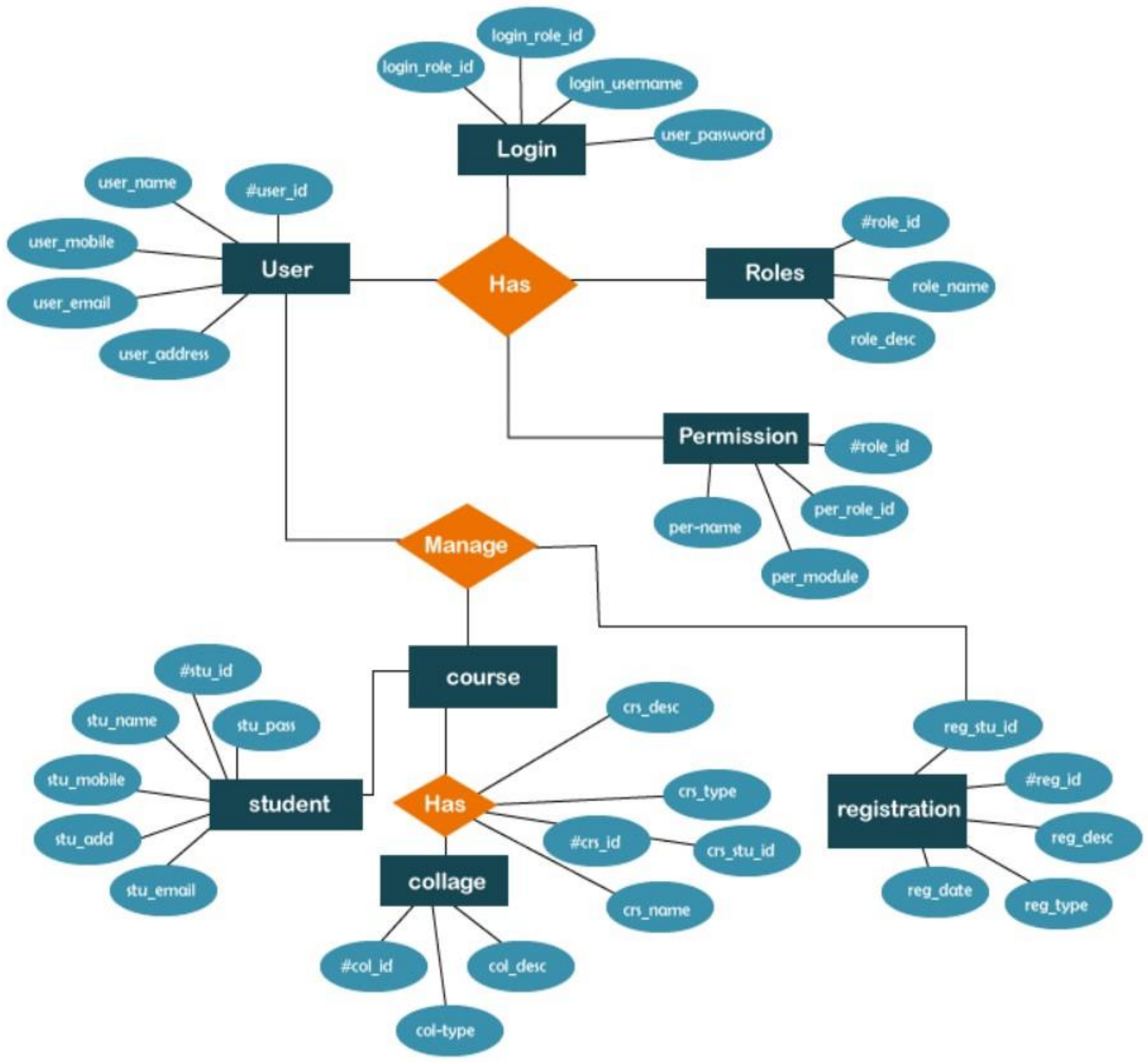
CHAPTER 2

SYSTEM DESIGN

2.1 SCHEMA DIAGRAM



2.2 ER DIAGRAM



CHAPTER 3

IMPLEMENTATION

3.1 LANGUAGE USED FOR IMPLEMENTATION

1. HTML (Hypertext Markup Language):

HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the content and structure of web content. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. It is primarily used for creating the structure and content of web pages. While HTML alone may not be sufficient for implementing a payroll management system, it is often used in conjunction with other languages, such as CSS for styling and JavaScript for interactivity, to build web-based interfaces for such systems. HTML provides the basic building blocks for organizing and presenting information on the web, including forms for inputting data, tables for organizing data, and various elements for structuring content.

In addition to providing the structure and content of web pages, HTML also enables the inclusion of multimedia elements such as images, videos, and audio. This allows for a more dynamic and engaging user experience. With HTML, you can also create links that connect different web pages, enabling users to navigate through a website easily. HTML has evolved over time, with the latest version being HTML5. This version introduced new features and enhancements, including better support for multimedia elements, improved handling of forms, and the ability to create more interactive and responsive web applications.

Fig. 3.1.1 Logo of HTML

2. Cascading Style Sheets (CSS):

CSS is a programming language used to control the presentation and layout of HTML documents. CSS is used to define styles for web pages, including colours, fonts, spacing, and more. It allows developers to separate the structure of a webpage (HTML) from its presentation (CSS), making it easier to maintain and update websites. CSS is a style sheet language used for specifying the presentation and styling of a document written in a markup language such as HTML or XML.

CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS enables the separation of content (HTML) from presentation (styling), making it easier to manage and maintain websites. CSS rules consist of selectors that target HTML elements and declarations that define how those elements should be styled. CSS follows the cascade mechanism, where styles are applied based on specificity and order of declaration, allowing for precise control over styling. CSS treats each HTML element as a box model consisting of content, padding, border, and margin, which can be styled individually. It provides techniques like media queries and flexible layouts to create responsive designs that adapt to different screen sizes and devices.

Fig. 3.1.2 Logo of CSS

3. JavaScript:

JS is a high-level programming language that follows the ECMAScript standard. It was originally designed as a scripting language for websites but became widely adopted as a general-purpose programming language, and is currently the most popular programming language in use. JavaScript is usually found running in a web browser as interactive or automated content, ranging from popup messages and live clocks to large web applications.

JavaScript is also commonly used in server-side programming through platforms like Node.js, or "embedded" in non-JavaScript applications where the base programming language lacks the high-level functionality that JavaScript offers. JavaScript is a versatile programming language commonly used for web development. It allows developers to add interactive elements, animations, and dynamic content to websites. JavaScript can also be used for server-side development (Node.js), mobile app development, and even desktop application development (Electron).

Fig. 3.1.3 Logo of JavaScript

4. PHP (Hypertext Preprocessor):

PHP is a server-side scripting language primarily used for web development. It is commonly embedded into HTML to create dynamic web pages, handle forms, manage databases, and perform other server-side tasks. PHP is open-source and widely supported, making it a popular choice for building websites and web applications. PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code which may be any type of data, such as generated HTML or binary image data would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist that can be employed to orchestrate or facilitate the generation of that response. PHP code can also be directly executed from the command line. Choosing between PHP5.6 and PHP7.x depends on various factors such as the compatibility of your codebase with newer PHP versions, the performance requirements of your application, and the level of support and updates for the PHP version you choose.

Fig. 3.1.4 Logo of PHP

5. MySQL:

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing structured data. It is developed, distributed, and supported by Oracle Corporation. MySQL is known for its reliability, scalability, and ease of use, making it a popular choice for web applications and other data-driven projects. It uses SQL (Structured Query Language) for querying and managing databases.

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems.

Fig. 3.1.5 Logo of MySQL

6. jQuery:

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. jQuery is a powerful library that simplifies the process of working with JavaScript. It allows developers to easily interact with HTML elements, manipulate the structure and content of a webpage, manage events, create animations, and perform Ajax (asynchronous JavaScript and XML) requests.

One of the key features of jQuery is its ability to simplify HTML document traversal and manipulation. With just a few lines of code, developers can easily select specific elements on a webpage and perform actions such as changing their content, styling, or position.jQuery also provides a straightforward way to handle events. Whether it's a click, hover, or keyboard event, developers can easily attach functions to these events and execute specific actions when they occur. This makes it easier to create dynamic and interactive webpages.

Animations are another area where jQuery excels. Developers can animate the properties of HTML elements, such as their size, position, or opacity, with just a few lines of code. This brings life to a webpage and provides a more engaging user experience.Additionally, jQuery simplifies the process of making Ajax requests. With methods like $.ajax(), developers can easily send and receive data from a server without reloading the entire webpage. This enables the creation of more dynamic and responsive web applications.jQuery is also known for its wide browser compatibility. It provides a consistent API that works across different browsers, ensuring that developers can write code that behaves consistently regardless of the user's chosen browser.

Fig. 3.1.5 Logo of jQuery

3.2 PLATFORM USED FOR IMPLEMENTATION

XAMPP:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It consists mainly of Apache HTTP Server, MariaDB database, and interpreters for scripting languages such as PHP and Perl. The name XAMPP stands for Cross-Platform (X), Apache (A), MariaDB/MySQL (M), PHP (P), and Perl (P). It provides an easy way to set up a local server environment for testing and development purposes on Windows, macOS, and Linux systems.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

a. **Cross-Platform:**

   Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.

b. **Apache:**

   It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

c. **MariaDB:**

   Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.

d. **PHP:**

It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for Hypertext Processor.

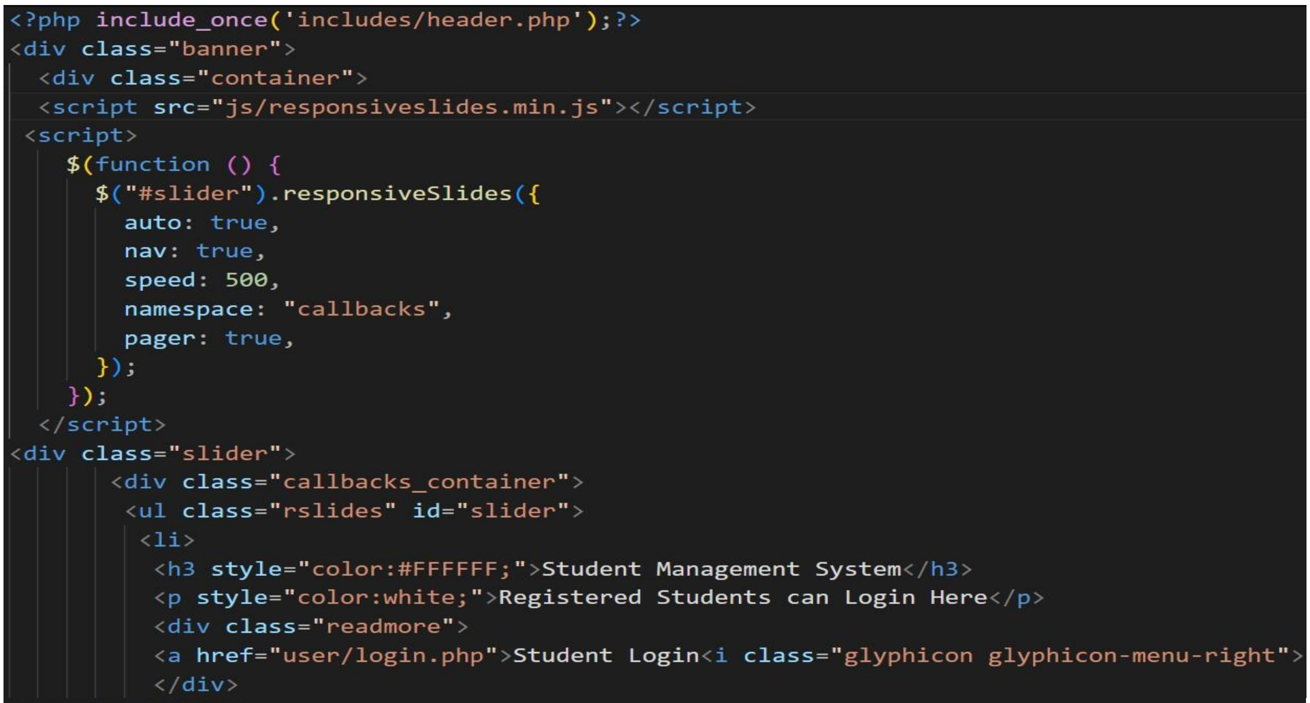
e. Perl:

It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.

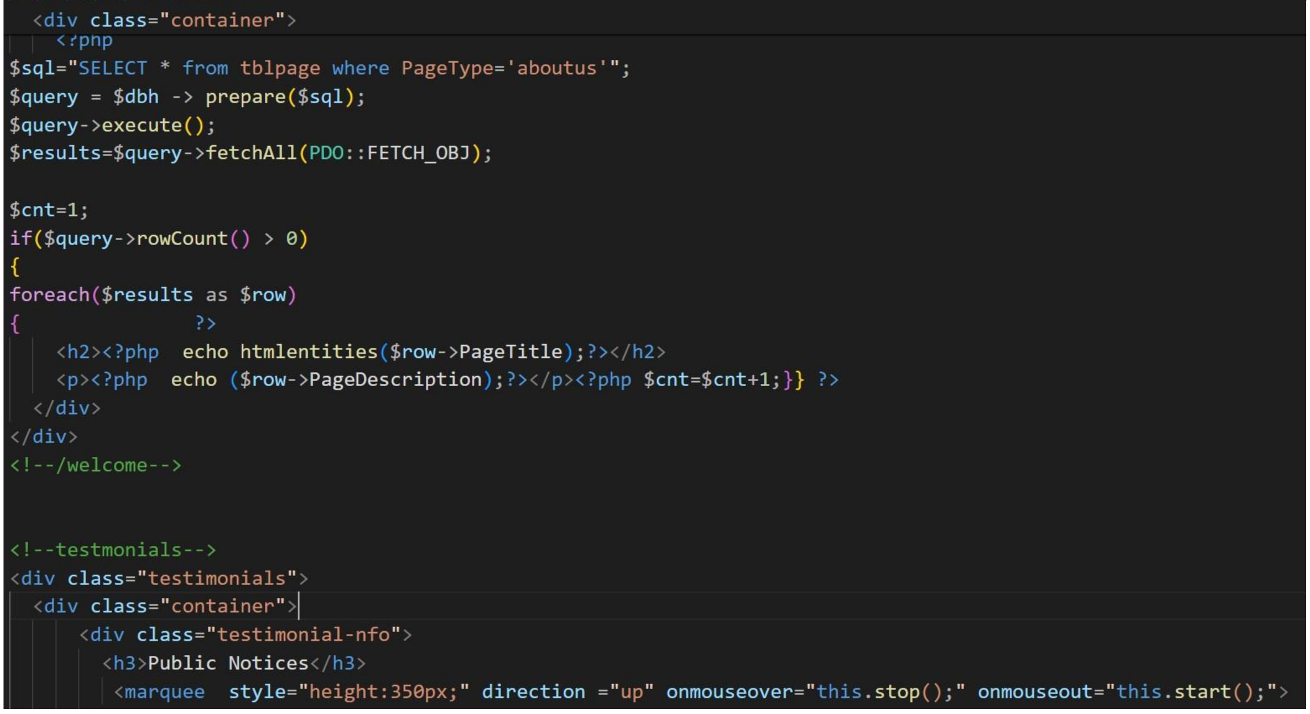
Fig. 3.2 Logo of XAMPP

3.3 SQL COMMANDS AND QUERIES

To View Dashboard:

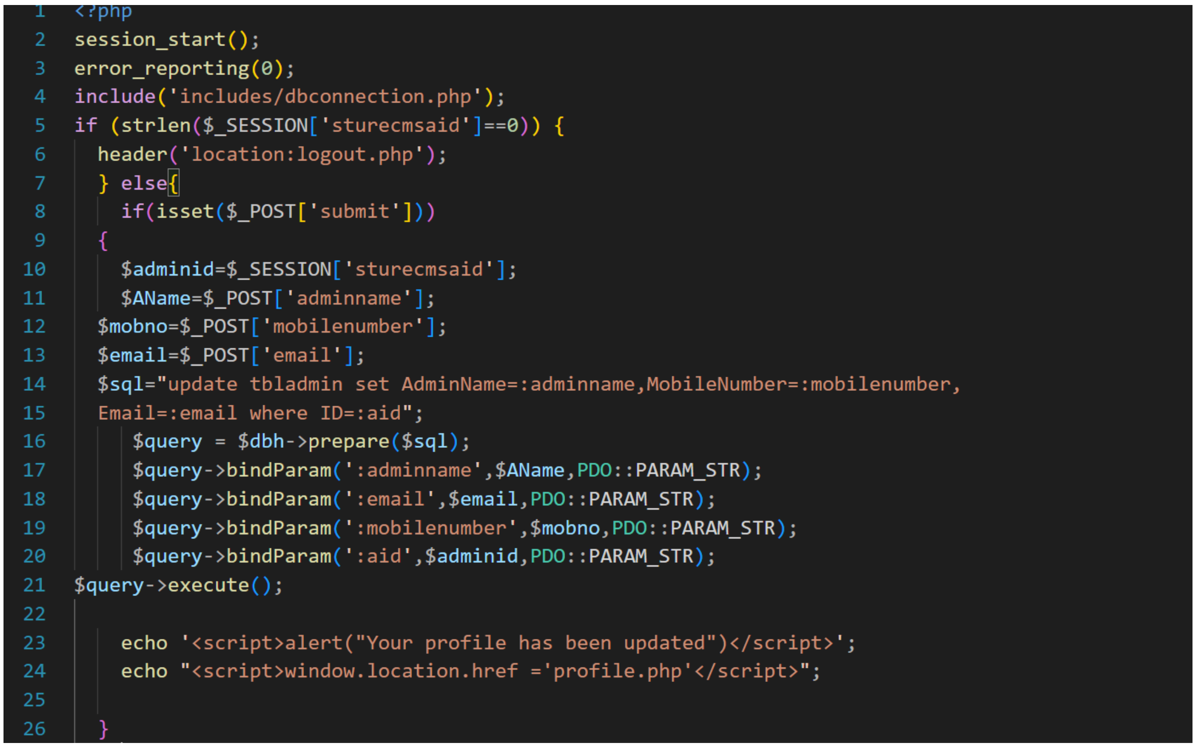


```
<?php include_once('includes/header.php');?>
<div class="banner">
  <div class="container">
  <script src="js/responsiveslides.min.js"></script>
 <script>
    $(function () {
      $("#slider").responsiveSlides({
        auto: true,
        nav: true,
        speed: 500,
        namespace: "callbacks",
        pager: true,
      });
    });
  </script>
<div class="slider">
        <div class="callbacks_container">
         <ul class="rslides" id="slider">
          <li>
           <h3 style="color:#FFFFFF;">Student Management System</h3>
           <p style="color:white;">Registered Students can Login Here</p>
           <div class="readmore">
           <a href="user/login.php">Student Login<i class="glyphicon glyphicon-menu-right">
           </div>
```



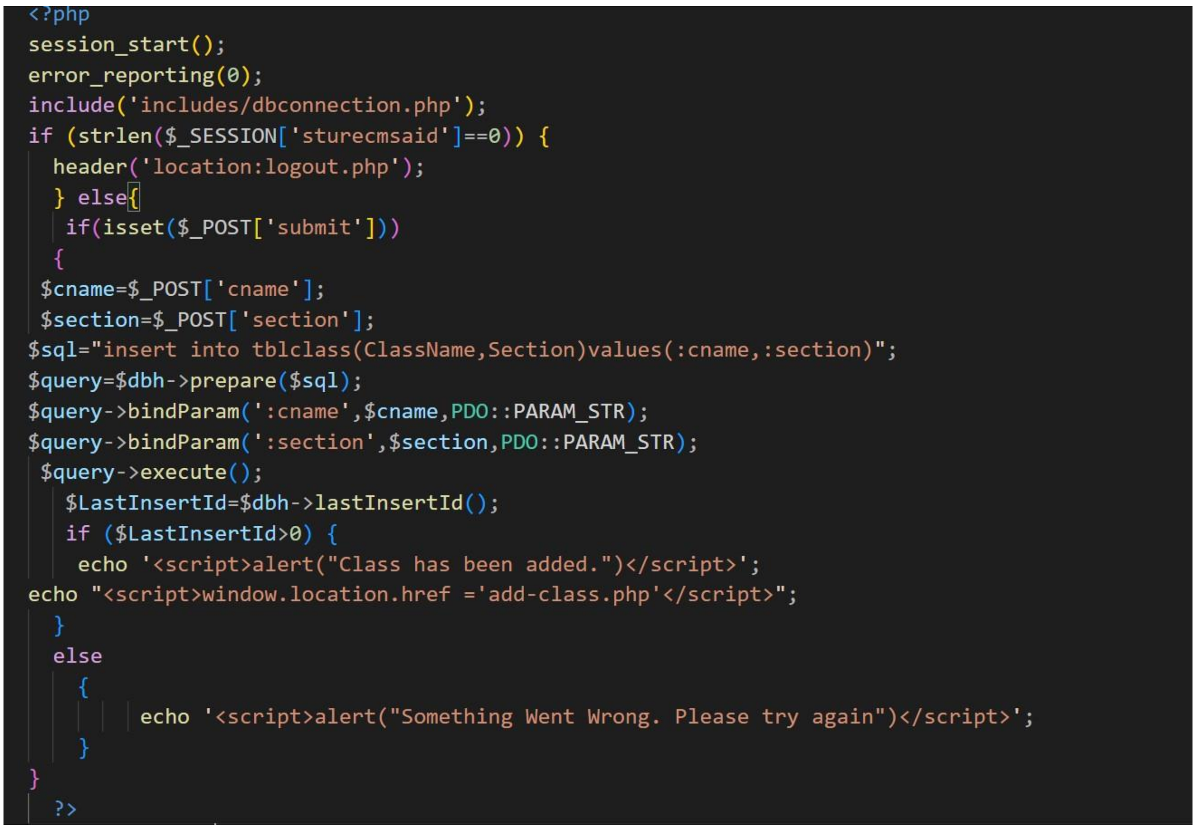
```
  <div class="container">
    <?php
$sql="SELECT * from tblpage where PageType='aboutus'";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);

$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $row)
{               ?>
    <h2><?php  echo htmlentities($row->PageTitle);?></h2>
    <p><?php  echo ($row->PageDescription);?></p><?php $cnt=$cnt+1;}} ?>
  </div>
</div>
<!--/welcome-->


<!--testmonials-->
<div class="testimonials">
  <div class="container">
        <div class="testimonial-nfo">
          <h3>Public Notices</h3>
           <marquee  style="height:350px;" direction ="up" onmouseover="this.stop();" onmouseout="this.start();">
```

To view Admin Profile:



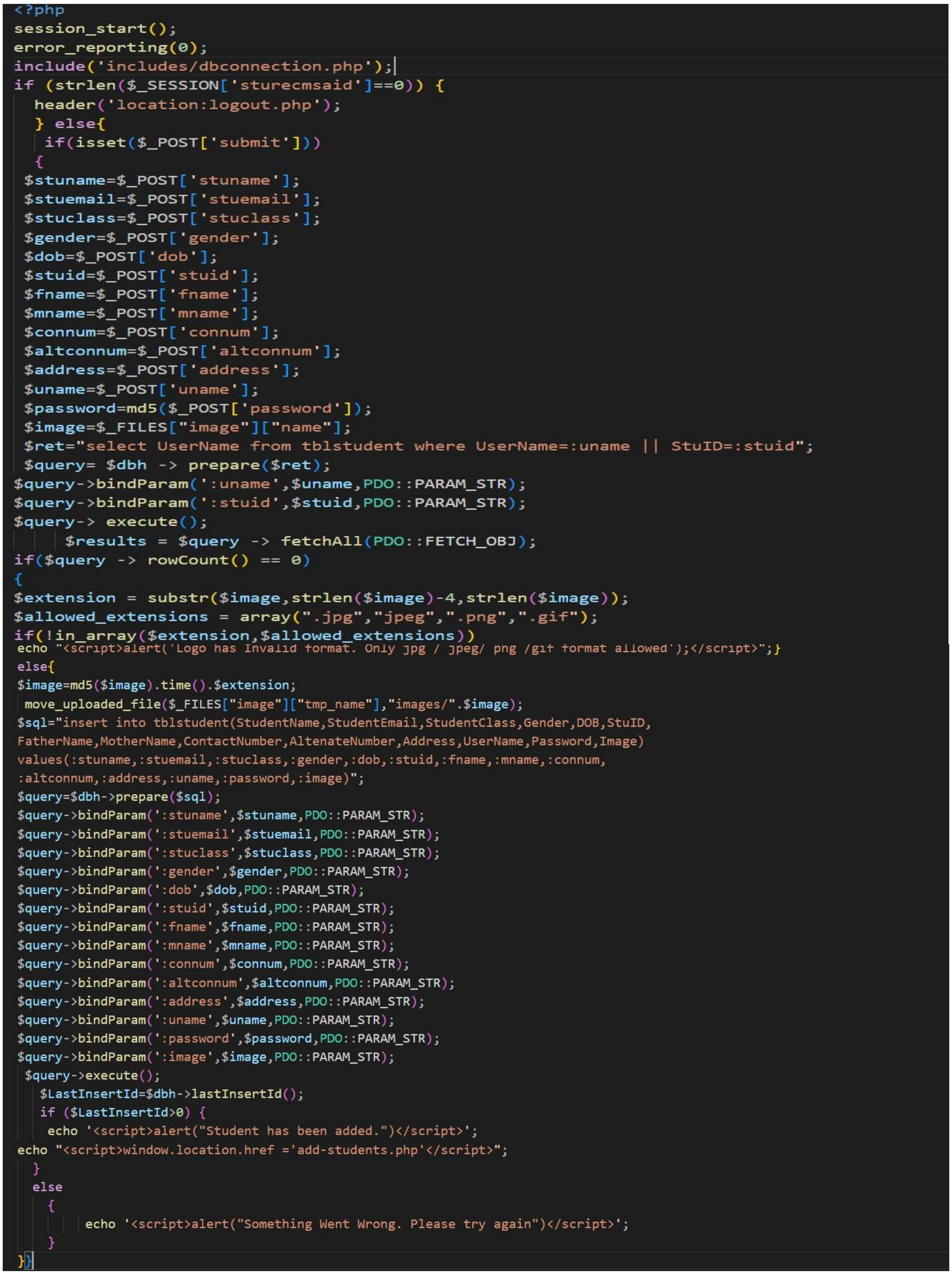
```php
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['sturecmsaid']==0)) {
  header('location:logout.php');
  } else{
    if(isset($_POST['submit']))
  {
    $adminid=$_SESSION['sturecmsaid'];
    $AName=$_POST['adminname'];
  $mobno=$_POST['mobilenumber'];
  $email=$_POST['email'];
  $sql="update tbladmin set AdminName=:adminname,MobileNumber=:mobilenumber,
  Email=:email where ID=:aid";
     $query = $dbh->prepare($sql);
     $query->bindParam(':adminname',$AName,PDO::PARAM_STR);
     $query->bindParam(':email',$email,PDO::PARAM_STR);
     $query->bindParam(':mobilenumber',$mobno,PDO::PARAM_STR);
     $query->bindParam(':aid',$adminid,PDO::PARAM_STR);
$query->execute();

    echo '<script>alert("Your profile has been updated")</script>';
    echo "<script>window.location.href ='profile.php'</script>";

  }
```

To add Class:



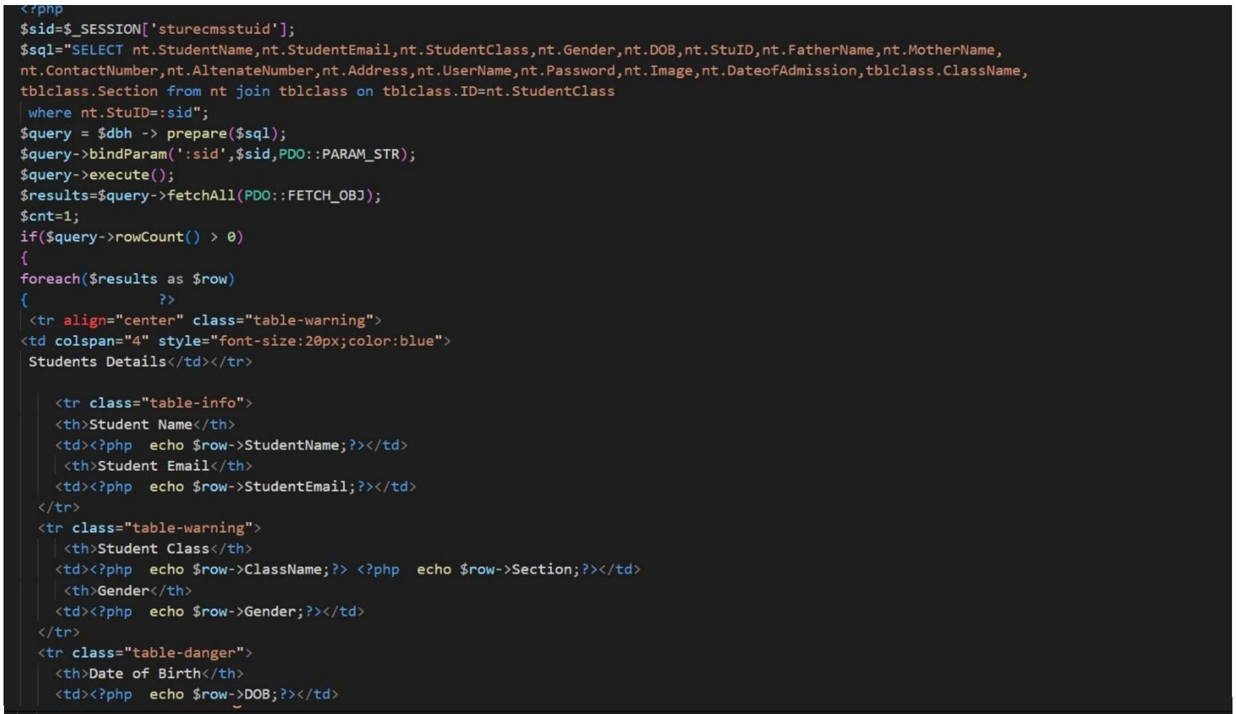
```php
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['sturecmsaid']==0)) {
  header('location:logout.php');
  } else{
   if(isset($_POST['submit']))
  {
 $cname=$_POST['cname'];
 $section=$_POST['section'];
$sql="insert into tblclass(ClassName,Section)values(:cname,:section)";
$query=$dbh->prepare($sql);
$query->bindParam(':cname',$cname,PDO::PARAM_STR);
$query->bindParam(':section',$section,PDO::PARAM_STR);
 $query->execute();
   $LastInsertId=$dbh->lastInsertId();
   if ($LastInsertId>0) {
    echo '<script>alert("Class has been added.")</script>';
echo "<script>window.location.href ='add-class.php'</script>";
  }
  else
    {
         echo '<script>alert("Something Went Wrong. Please try again")</script>';
    }
}
  ?>
```

To add Students:

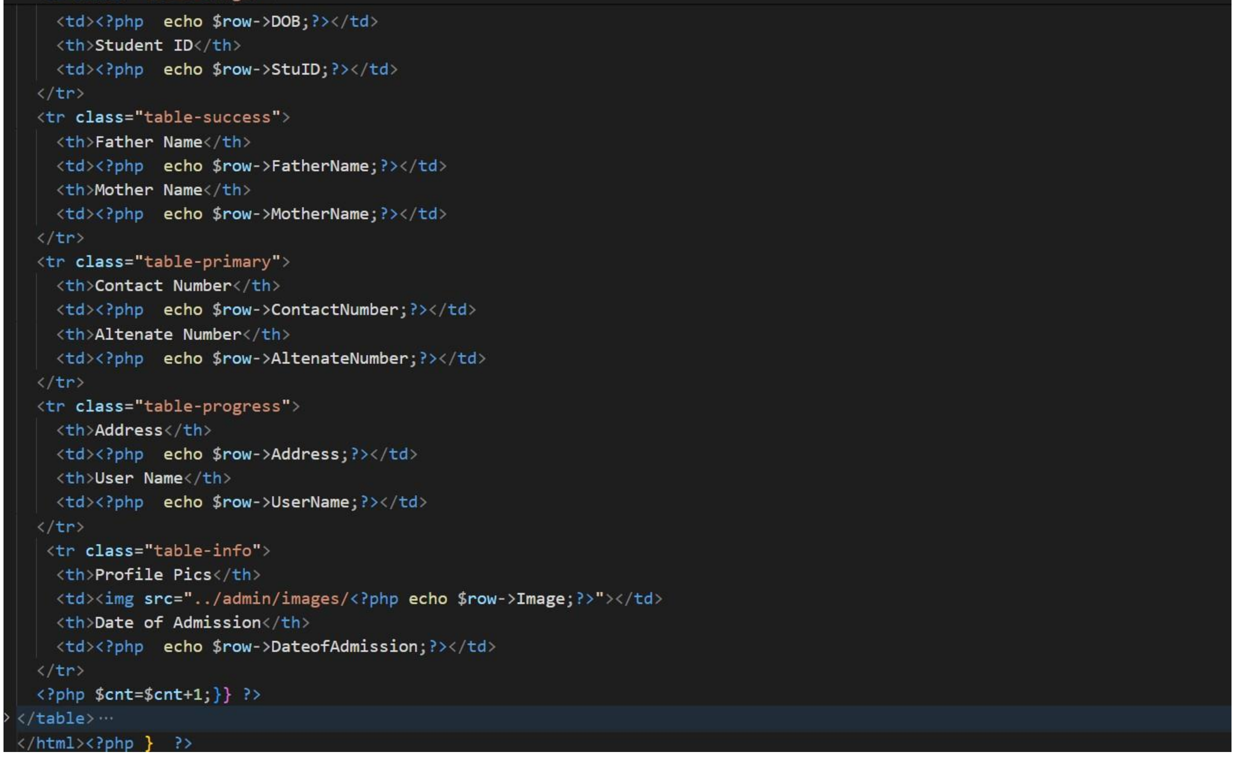


```php
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['sturecmsaid']==0)) {
  header('location:logout.php');
  } else{
   if(isset($_POST['submit']))
  {
 $stuname=$_POST['stuname'];
 $stuemail=$_POST['stuemail'];
 $stuclass=$_POST['stuclass'];
 $gender=$_POST['gender'];
 $dob=$_POST['dob'];
 $stuid=$_POST['stuid'];
 $fname=$_POST['fname'];
 $mname=$_POST['mname'];
 $connum=$_POST['connum'];
 $altconnum=$_POST['altconnum'];
 $address=$_POST['address'];
 $uname=$_POST['uname'];
 $password=md5($_POST['password']);
 $image=$_FILES["image"]["name"];
 $ret="select UserName from tblstudent where UserName=:uname || StuID=:stuid";
 $query= $dbh -> prepare($ret);
$query->bindParam(':uname',$uname,PDO::PARAM_STR);
$query->bindParam(':stuid',$stuid,PDO::PARAM_STR);
$query-> execute();
    $results = $query -> fetchAll(PDO::FETCH_OBJ);
if($query -> rowCount() == 0)
{
$extension = substr($image,strlen($image)-4,strlen($image));
$allowed_extensions = array(".jpg","jpeg",".png",".gif");
if(!in_array($extension,$allowed_extensions))
echo "<script>alert('Logo has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";}
else{
$image=md5($image).time().$extension;
 move_uploaded_file($_FILES["image"]["tmp_name"],"images/".$image);
$sql="insert into tblstudent(StudentName,StudentEmail,StudentClass,Gender,DOB,StuID,
FatherName,MotherName,ContactNumber,AltenateNumber,Address,UserName,Password,Image)
values(:stuname,:stuemail,:stuclass,:gender,:dob,:stuid,:fname,:mname,:connum,
:altconnum,:address,:uname,:password,:image)";
$query=$dbh->prepare($sql);
$query->bindParam(':stuname',$stuname,PDO::PARAM_STR);
$query->bindParam(':stuemail',$stuemail,PDO::PARAM_STR);
$query->bindParam(':stuclass',$stuclass,PDO::PARAM_STR);
$query->bindParam(':gender',$gender,PDO::PARAM_STR);
$query->bindParam(':dob',$dob,PDO::PARAM_STR);
$query->bindParam(':stuid',$stuid,PDO::PARAM_STR);
$query->bindParam(':fname',$fname,PDO::PARAM_STR);
$query->bindParam(':mname',$mname,PDO::PARAM_STR);
$query->bindParam(':connum',$connum,PDO::PARAM_STR);
$query->bindParam(':altconnum',$altconnum,PDO::PARAM_STR);
$query->bindParam(':address',$address,PDO::PARAM_STR);
$query->bindParam(':uname',$uname,PDO::PARAM_STR);
$query->bindParam(':password',$password,PDO::PARAM_STR);
$query->bindParam(':image',$image,PDO::PARAM_STR);
 $query->execute();
   $LastInsertId=$dbh->lastInsertId();
   if ($LastInsertId>0) {
    echo '<script>alert("Student has been added.")</script>';
echo "<script>window.location.href ='add-students.php'</script>";
  }
  else
    {
         echo '<script>alert("Something Went Wrong. Please try again")</script>';
    }
}}
```

To view Student Profile:

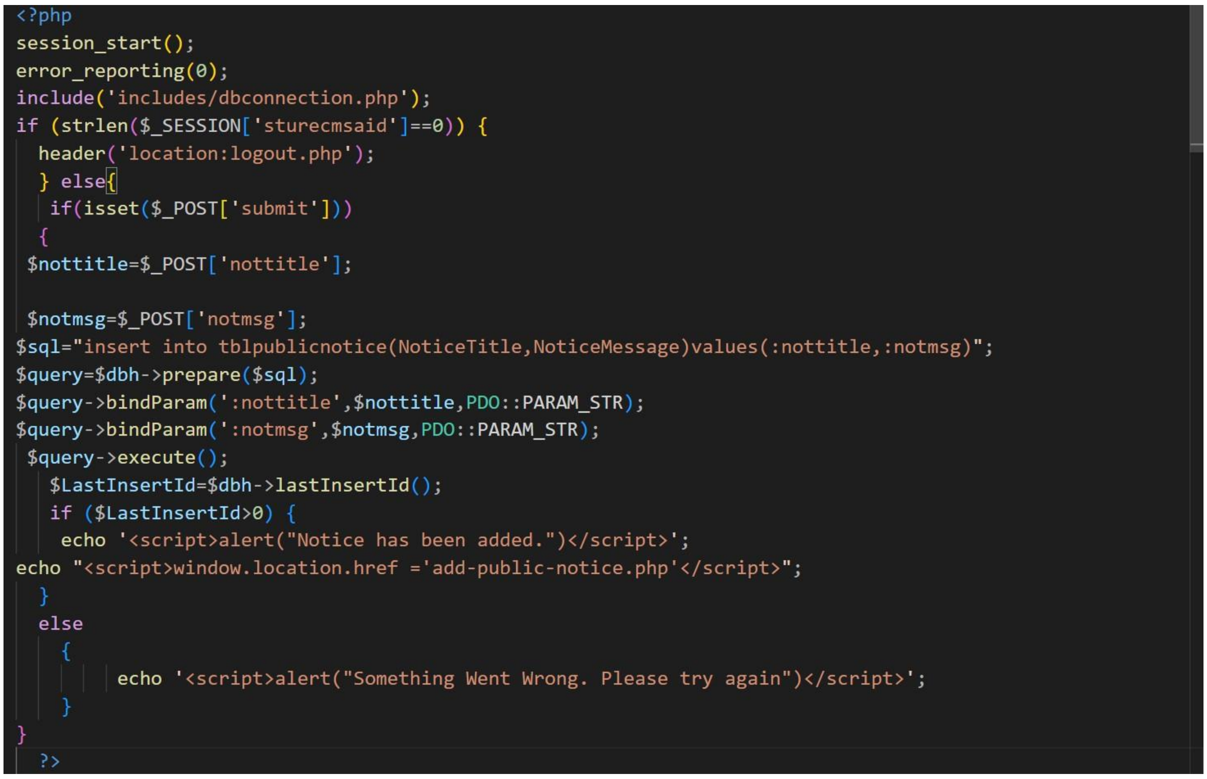


```
<?php
$sid=$_SESSION['sturecmsstuid'];
$sql="SELECT nt.StudentName,nt.StudentEmail,nt.StudentClass,nt.Gender,nt.DOB,nt.StuID,nt.FatherName,nt.MotherName,
nt.ContactNumber,nt.AltenateNumber,nt.Address,nt.UserName,nt.Password,nt.Image,nt.DateofAdmission,tblclass.ClassName,
tblclass.Section from nt join tblclass on tblclass.ID=nt.StudentClass
 where nt.StuID=:sid";
$query = $dbh -> prepare($sql);
$query->bindParam(':sid',$sid,PDO::PARAM_STR);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $row)
{               ?>
 <tr align="center" class="table-warning">
<td colspan="4" style="font-size:20px;color:blue">
 Students Details</td></tr>

    <tr class="table-info">
    <th>Student Name</th>
    <td><?php  echo $row->StudentName;?></td>
     <th>Student Email</th>
    <td><?php  echo $row->StudentEmail;?></td>
  </tr>
  <tr class="table-warning">
     <th>Student Class</th>
    <td><?php  echo $row->ClassName;?> <?php  echo $row->Section;?></td>
     <th>Gender</th>
    <td><?php  echo $row->Gender;?></td>
  </tr>
  <tr class="table-danger">
    <th>Date of Birth</th>
    <td><?php  echo $row->DOB;?></td>
```



```
    <td><?php  echo $row->DOB;?></td>
    <th>Student ID</th>
    <td><?php  echo $row->StuID;?></td>
  </tr>
  <tr class="table-success">
    <th>Father Name</th>
    <td><?php  echo $row->FatherName;?></td>
    <th>Mother Name</th>
    <td><?php  echo $row->MotherName;?></td>
  </tr>
  <tr class="table-primary">
    <th>Contact Number</th>
    <td><?php  echo $row->ContactNumber;?></td>
    <th>Altenate Number</th>
    <td><?php  echo $row->AltenateNumber;?></td>
  </tr>
  <tr class="table-progress">
    <th>Address</th>
    <td><?php  echo $row->Address;?></td>
    <th>User Name</th>
    <td><?php  echo $row->UserName;?></td>
  </tr>
   <tr class="table-info">
    <th>Profile Pics</th>
    <td><img src="../admin/images/<?php echo $row->Image;?>"></td>
    <th>Date of Admission</th>
    <td><?php  echo $row->DateofAdmission;?></td>
  </tr>
  <?php $cnt=$cnt+1;}} ?>
> </table>···
 </html><?php }  ?>
```

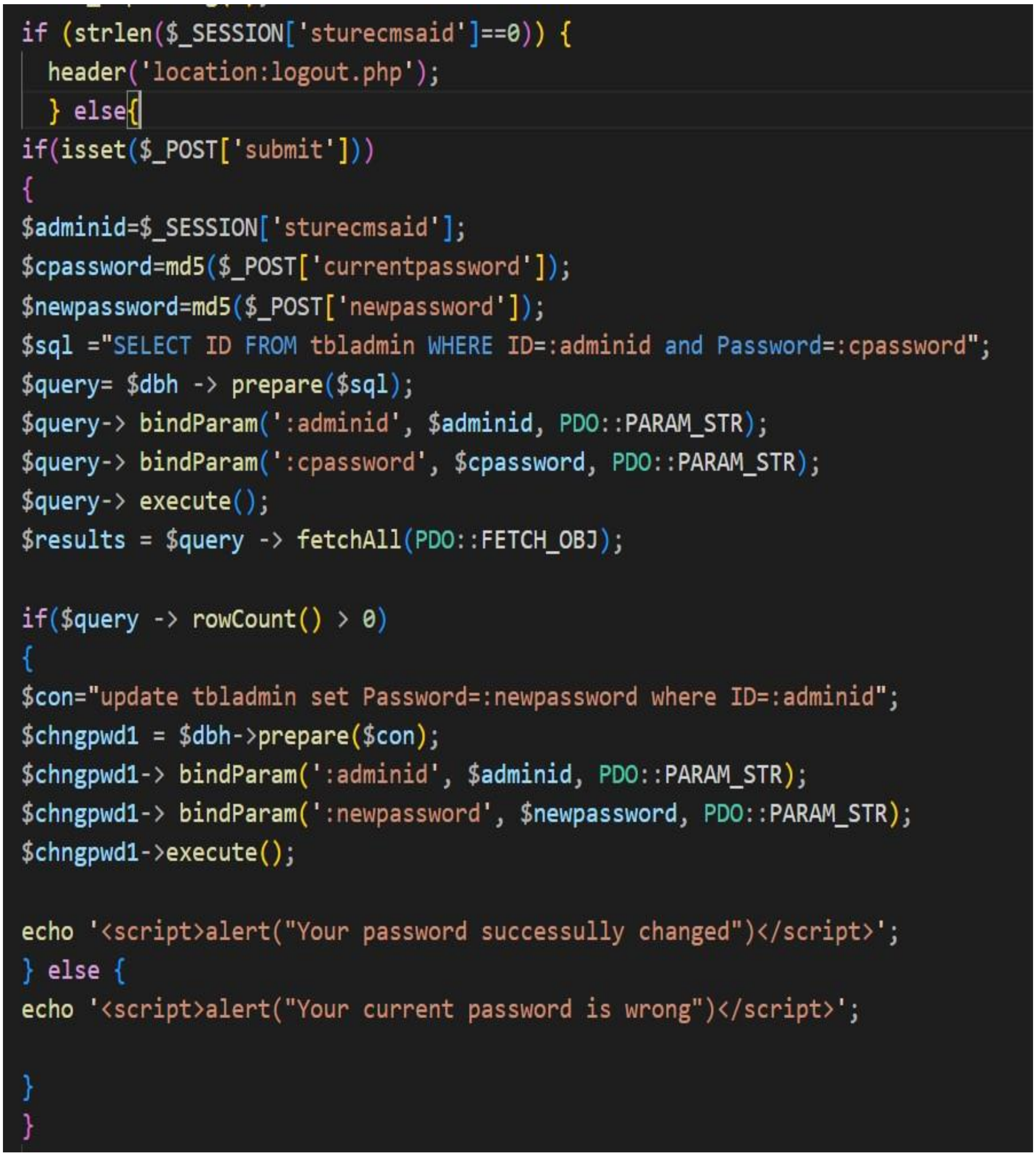
To add Publicnotice:



```php
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['sturecmsaid']==0)) {
  header('location:logout.php');
  } else{
   if(isset($_POST['submit']))
  {
 $nottitle=$_POST['nottitle'];

 $notmsg=$_POST['notmsg'];
$sql="insert into tblpublicnotice(NoticeTitle,NoticeMessage)values(:nottitle,:notmsg)";
$query=$dbh->prepare($sql);
$query->bindParam(':nottitle',$nottitle,PDO::PARAM_STR);
$query->bindParam(':notmsg',$notmsg,PDO::PARAM_STR);
 $query->execute();
   $LastInsertId=$dbh->lastInsertId();
   if ($LastInsertId>0) {
    echo '<script>alert("Notice has been added.")</script>';
echo "<script>window.location.href ='add-public-notice.php'</script>";
  }
  else
    {
         echo '<script>alert("Something Went Wrong. Please try again")</script>';
    }
}
  ?>
```

3.4 OUTPUT TESTING:

Created the individual login for each student and Rectified the bug in the student login form



```
if (strlen($_SESSION['sturecmsaid']==0)) {
  header('location:logout.php');
  } else{
if(isset($_POST['submit']))
{
$adminid=$_SESSION['sturecmsaid'];
$cpassword=md5($_POST['currentpassword']);
$newpassword=md5($_POST['newpassword']);
$sql ="SELECT ID FROM tbladmin WHERE ID=:adminid and Password=:cpassword";
$query= $dbh -> prepare($sql);
$query-> bindParam(':adminid', $adminid, PDO::PARAM_STR);
$query-> bindParam(':cpassword', $cpassword, PDO::PARAM_STR);
$query-> execute();
$results = $query -> fetchAll(PDO::FETCH_OBJ);

if($query -> rowCount() > 0)
{
$con="update tbladmin set Password=:newpassword where ID=:adminid";
$chngpwd1 = $dbh->prepare($con);
$chngpwd1-> bindParam(':adminid', $adminid, PDO::PARAM_STR);
$chngpwd1-> bindParam(':newpassword', $newpassword, PDO::PARAM_STR);
$chngpwd1->execute();

echo '<script>alert("Your password successully changed")</script>';
} else {
echo '<script>alert("Your current password is wrong")</script>';

}
}
```

CHAPTER 4

RESULTS

4.1 SNAPSHOTS

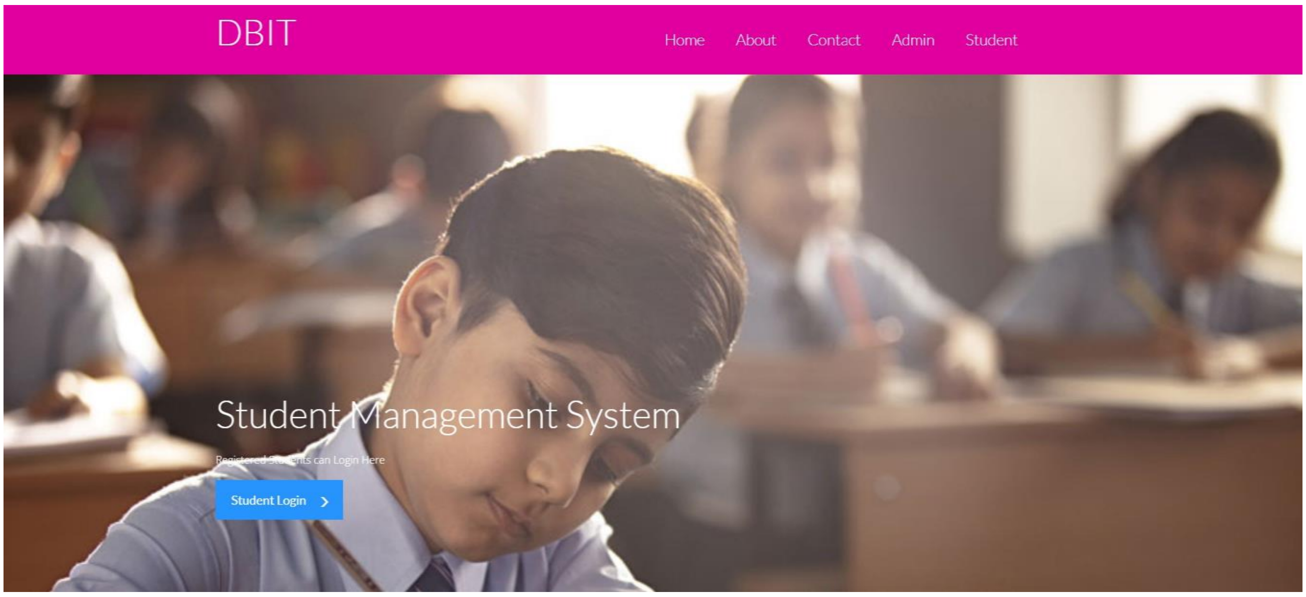
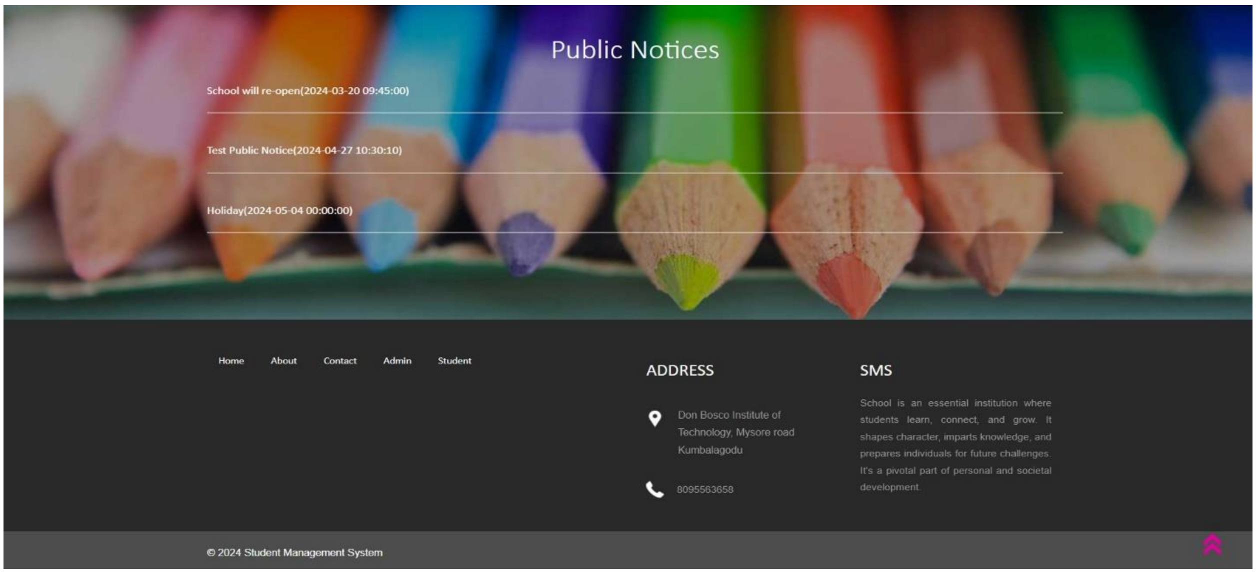


Fig. 4.1 Home Page



School will re-open(2024-03-20 09:45:00)

Test Public Notice(2024-04-27 10:30:10)

Holiday(2024-05-04 00:00:00)

ADDRESS

Don Bosco Institute of Technology, Mysore road Kumbalagodu

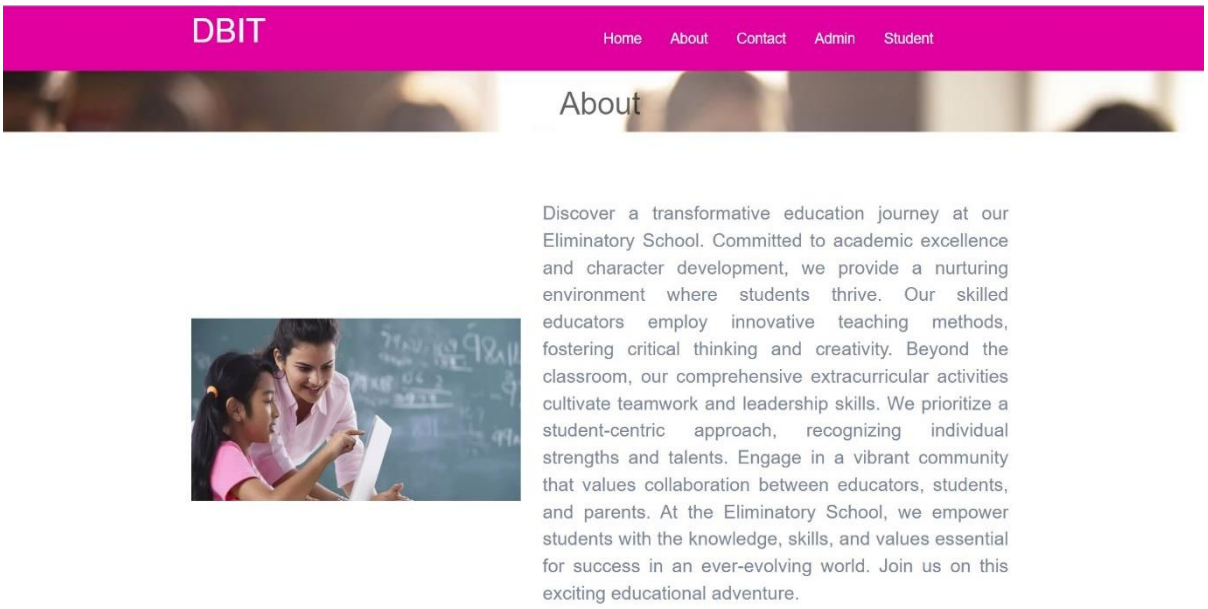
8095563658

SMS

School is an essential institution where students learn, connect, and grow. It shapes character, imparts knowledge, and prepares individuals for future challenges. It's a pivotal part of personal and societal development.

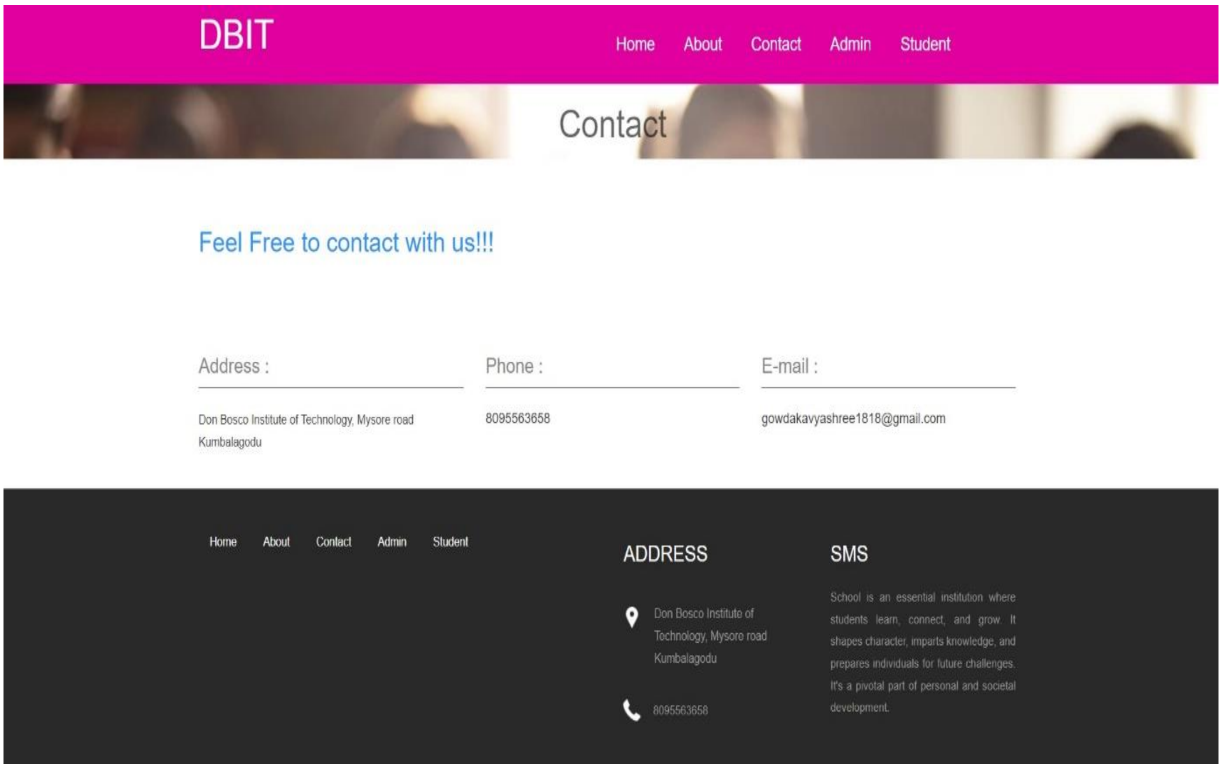
© 2024 Student Management System

Fig. 4.2 Home Page



Discover a transformative education journey at our Eliminatory School. Committed to academic excellence and character development, we provide a nurturing environment where students thrive. Our skilled educators employ innovative teaching methods, fostering critical thinking and creativity. Beyond the classroom, our comprehensive extracurricular activities cultivate teamwork and leadership skills. We prioritize a student-centric approach, recognizing individual strengths and talents. Engage in a vibrant community that values collaboration between educators, students, and parents. At the Eliminatory School, we empower students with the knowledge, skills, and values essential for success in an ever-evolving world. Join us on this exciting educational adventure.

Fig. 4.3 About Page



Feel Free to contact with us!!!

Address :	Phone :	E-mail :
Don Bosco Institute of Technology, Mysore road Kumbalagodu	8095563658	gowdakavyashree1818@gmail.com

ADDRESS

Don Bosco Institute of Technology, Mysore road Kumbalagodu

8095563658

SMS

School is an essential institution where students learn, connect, and grow. It shapes character, imparts knowledge, and prepares individuals for future challenges. It's a pivotal part of personal and societal development.

Fig. 4.4 Contact Page

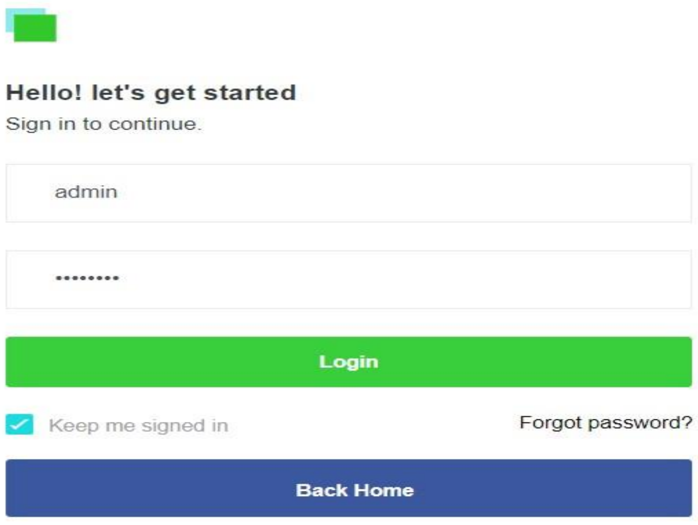


Fig. 4.5 Admin Page

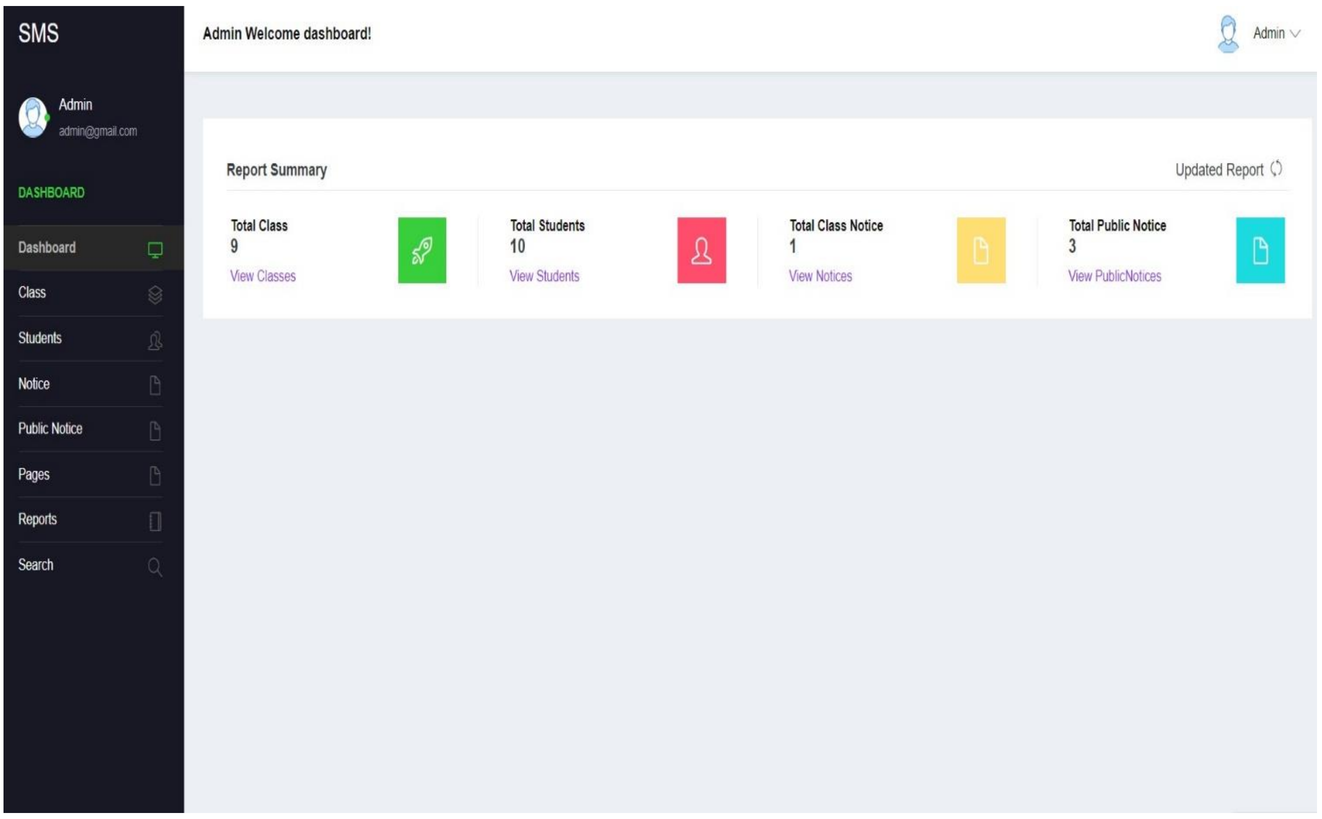
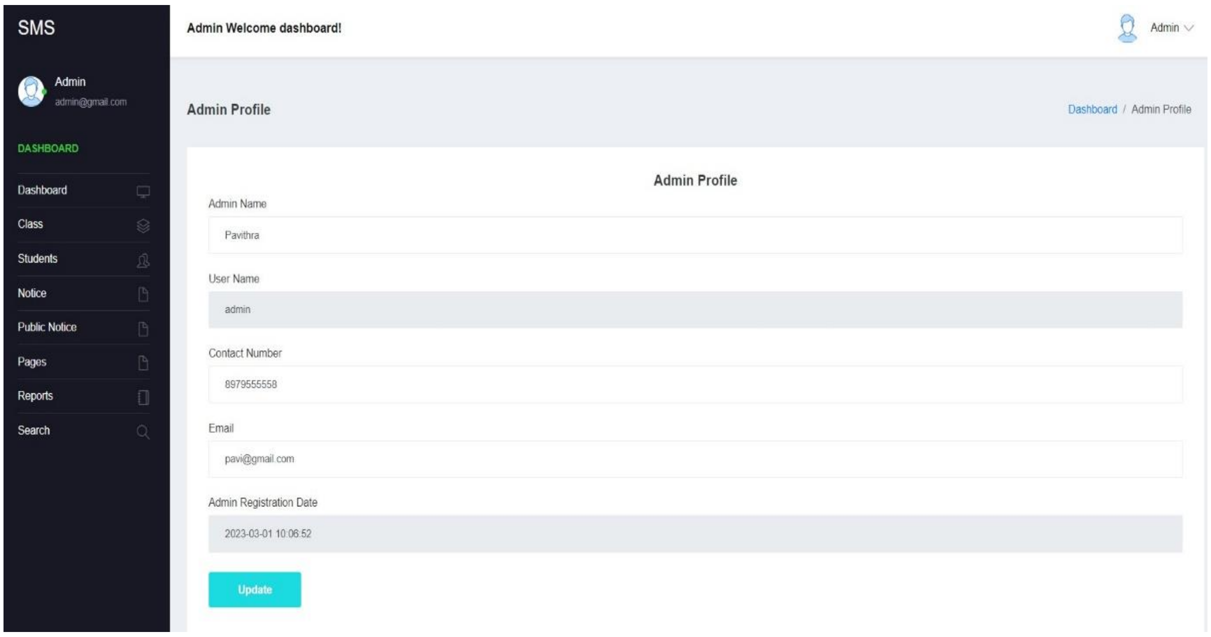


Fig. 4.6 Admin Dashboard



Admin Profile

Admin Name: Pavithra

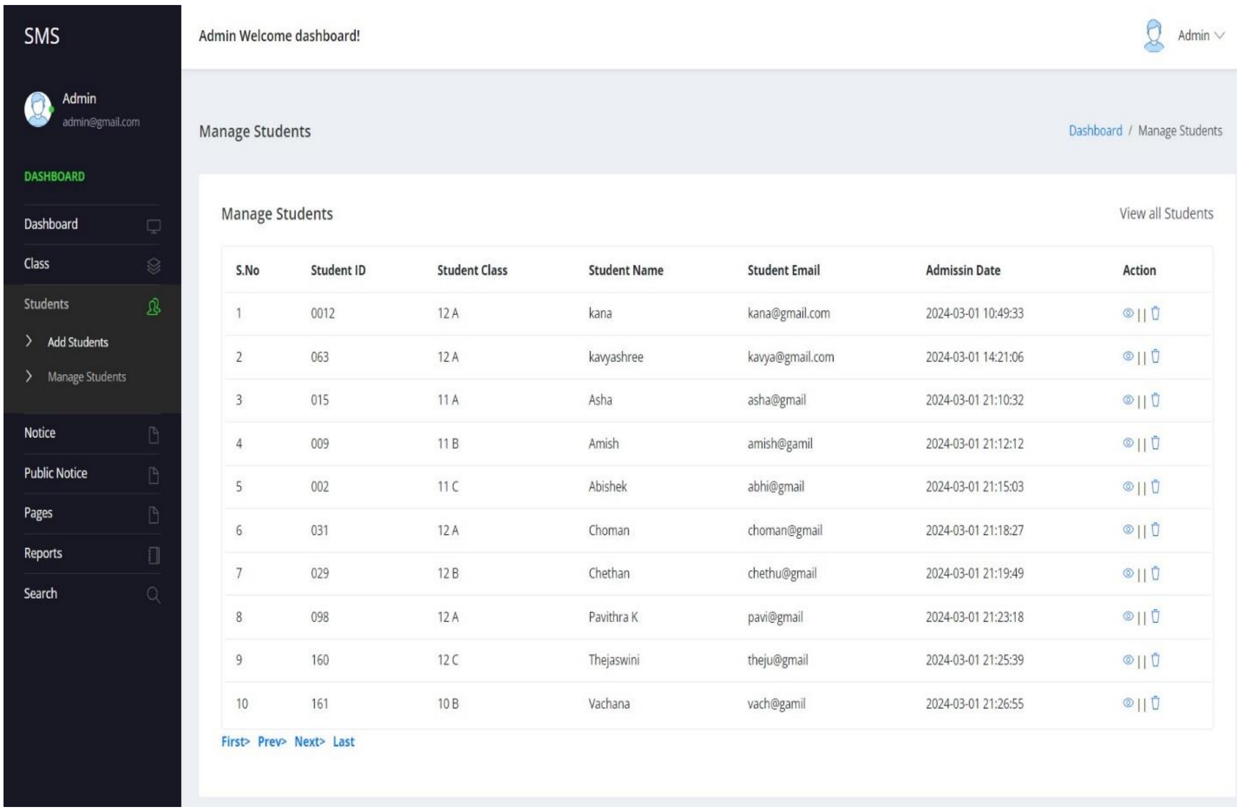
User Name: admin

Contact Number: 8979555558

Email: pavi@gmail.com

Admin Registration Date: 2023-03-01 10:06:52

Fig. 4.7 Admin Profile Page



Manage Students

S.No	Student ID	Student Class	Student Name	Student Email	Admissin Date	Action
1	0012	12 A	kana	kana@gmail.com	2024-03-01 10:49:33	◉ || 🗑
2	063	12 A	kavyashree	kavya@gmail.com	2024-03-01 14:21:06	◉ || 🗑
3	015	11 A	Asha	asha@gmail	2024-03-01 21:10:32	◉ || 🗑
4	009	11 B	Amish	amish@gamil	2024-03-01 21:12:12	◉ || 🗑
5	002	11 C	Abishek	abhi@gmail	2024-03-01 21:15:03	◉ || 🗑
6	031	12 A	Choman	choman@gmail	2024-03-01 21:18:27	◉ || 🗑
7	029	12 B	Chethan	chethu@gmail	2024-03-01 21:19:49	◉ || 🗑
8	098	12 A	Pavithra K	pavi@gmail	2024-03-01 21:23:18	◉ || 🗑
9	160	12 C	Thejaswini	theju@gmail	2024-03-01 21:25:39	◉ || 🗑
10	161	10 B	Vachana	vach@gamil	2024-03-01 21:26:55	◉ || 🗑

Fig. 4.8 Student Manage

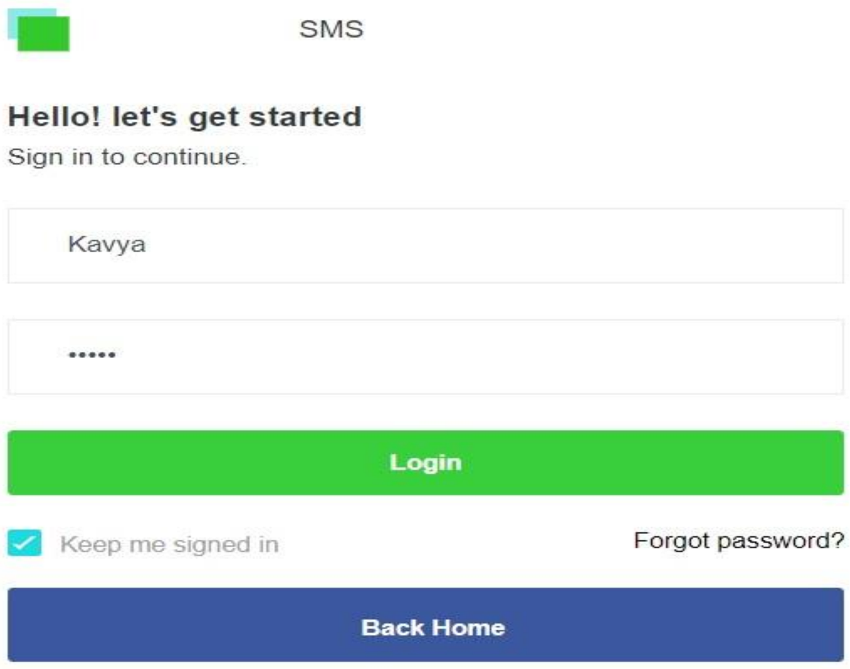


Fig. 4.9 Student Login Page

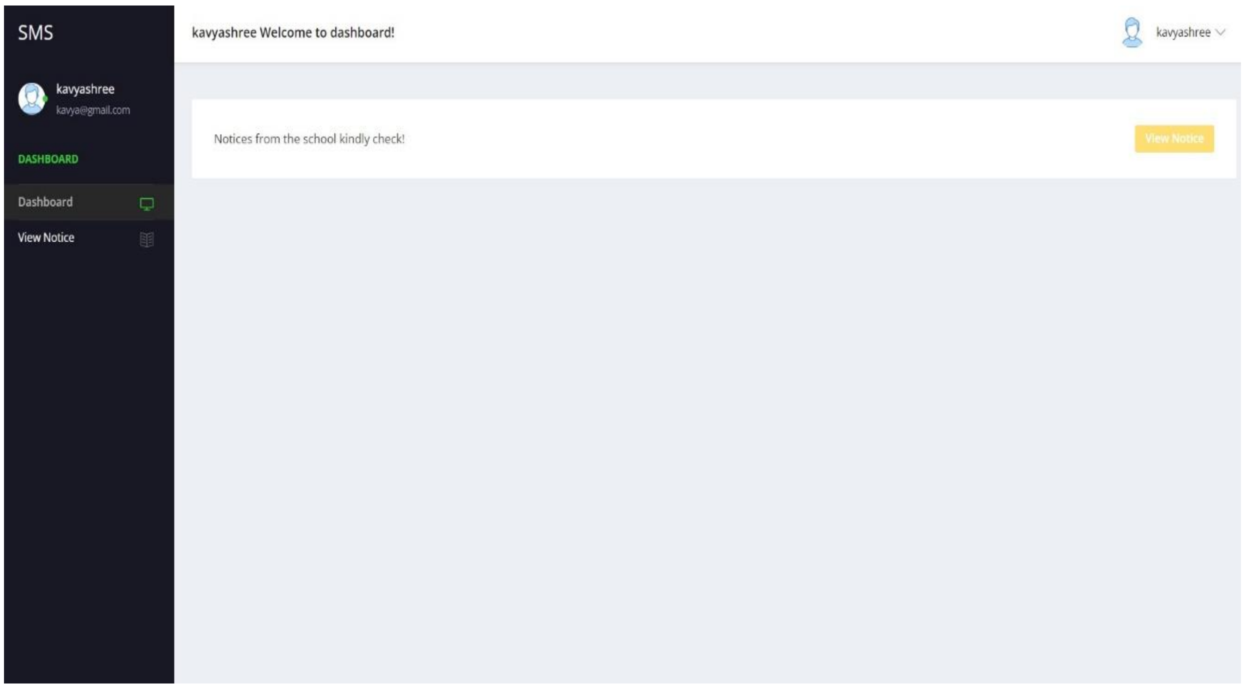
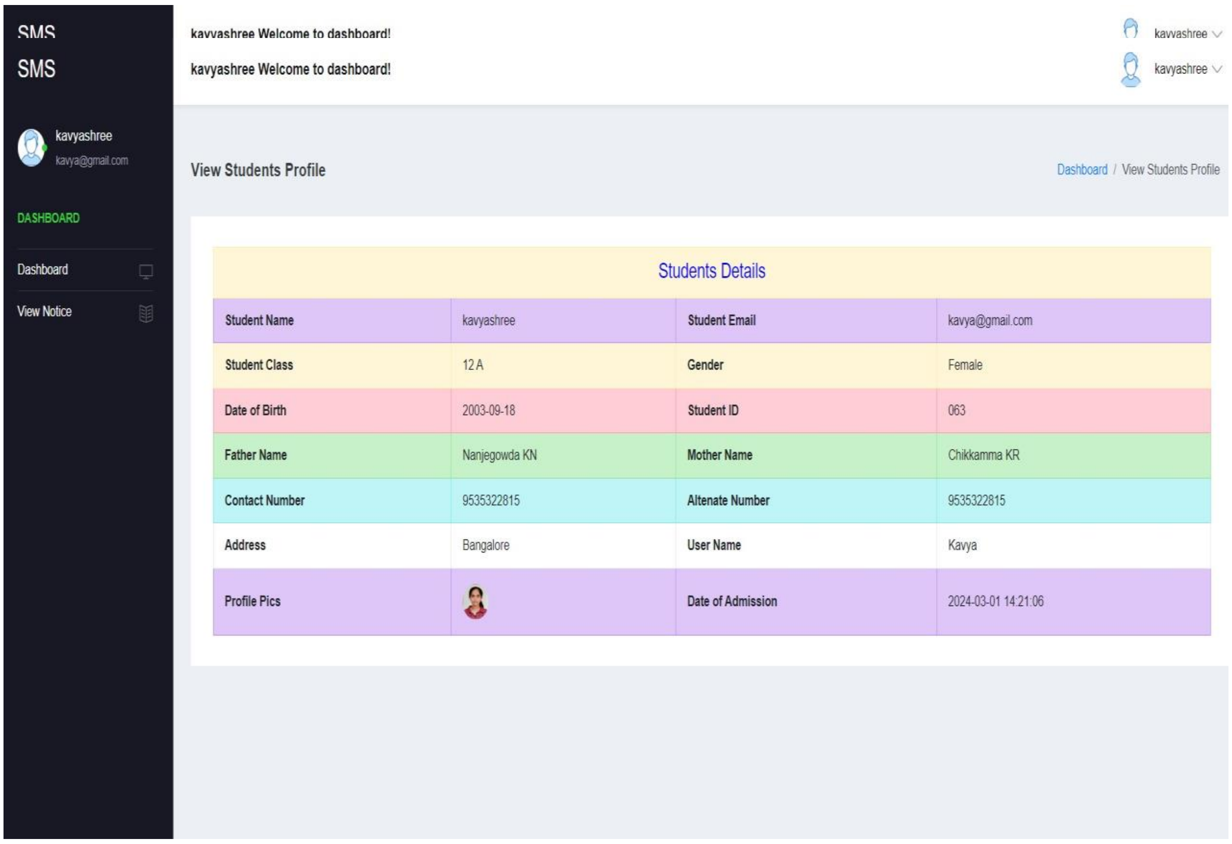


Fig. 4.10 Student Dashboard



SMS

SMS

kavyashree Welcome to dashboard!

kavyashree Welcome to dashboard!

kavyashree

kavyashree

kavyashree

kavya@gmail.com

DASHBOARD

Dashboard

View Notice

View Students Profile

Dashboard / View Students Profile

Students Details

Student Name	kavyashree	Student Email	kavya@gmail.com
Student Class	12 A	Gender	Female
Date of Birth	2003-09-18	Student ID	063
Father Name	Nanjegowda KN	Mother Name	Chikkamma KR
Contact Number	9535322815	Altenate Number	9535322815
Address	Bangalore	User Name	Kavya
Profile Pics		Date of Admission	2024-03-01 14:21:06

Fig. 4.11 Student Profile Page

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

The Student Management System is a web-based application that helps students, as well as school authorities, manage various activities related to students. The current system is manual, time-consuming, and costly. By implementing this online system, administrators can perform tasks efficiently and effectively.

The system includes modules for the admin, who can manage classes, students, notices, public notices, pages, and reports. The admin can also search for students, update their profile, change passwords, and recover passwords.

By using the Student Management System, schools can streamline their processes, improve communication between students and administrators, track academic performance, and enhance parental involvement. Additionally, the system offers centralized data management, security, and analytics capabilities.

Overall, the Student Management System is a valuable tool for educational institutions, providing numerous benefits for both students and administrators. Student Management System can also lead to cost savings for educational institutions. The automation of administrative tasks reduces the need for manual paperwork and eliminates the risk of errors or misplaced data. This can save time and resources for both teachers and administrators, allowing them to focus more on providing quality education.

Furthermore, the online nature of the system enables easy access to information from anywhere at any time. Students, parents, and teachers can access the system remotely, reducing the dependence on physical presence and facilitating communication and collaboration. The system benefits students, parents, teachers, and administrators alike, ultimately contributing to the overall success and growth of the educational institution.

5.2 FUTURE ENHANCEMENT

Some potential future enhancements for the Student Management System mentioned below could include:

- **Notifications and reminders:**
  Implementing a notification system that sends automatic reminders to students and parents about upcoming events, important deadlines, or changes in the schedule can help keep everyone informed and organized.
- **Attendance management:**
  Adding features to track and manage student attendance, such as generating attendance reports, setting up automated alerts for excessive absences, and integrating with biometric or RFID systems for easy and accurate attendance tracking.
- **Examination and grading system:**
  Developing an integrated examination and grading system that allows teachers to create online exams, automatically grade them, and provide instant feedback to students. This can also include features like generating report cards and calculating cumulative grade point averages.
- **Online assignment submission and grading:**
  Implementing a feature where students can submit assignments online and teachers can review and grade them digitally, saving time and paper.
- **Integration with learning management systems (LMS):**
  Integrating the Student Management System with popular LMS platforms like Moodle or Canvas to provide a seamless experience for teachers and students, allowing for centralized access to course content, assignments, grades, and discussions.
- **Financial management:**
  Adding features to manage and track student fees payments, generate invoices, and communicate with parents about payment deadlines and outstanding balances.
- **Analytics and reporting:**
  Adding advanced analytics and reporting capabilities to provide administrators with insights about student performance, attendance trends, and other key metrics. This can help identify areas of improvement and make data-driven decisions.

REFERENCES

1. Student Management System Project using PHP and MySQL, 2022
   https://phpgurukul.com/student-management-system-using-php-and-mysql/
2. Student Management System,2022
   https://www.ijraset.com/research-paper/student-management-system
3. https://journal-dogorangsang.in/no_1_Online_23/81_apr.pdf
4. youtube.com/watch?si=O2WP330l0lqe7e9X&v=pWr3VHAmFFQ&feature=youtu.be