1. Introduction

1.1 Overview

This Web Application provides facility to examination Form Postal worldwide. It saves time as it allows number of students/candidates to submit the exam form at same time and make payment of exam form online, so there is no need to wait in the long queue. It is evaluated and generated by the server. Administrator has privileges to create, modify and delete the test papers and its particular questions. User can login and fill the exam form and submit with online payment his specific card, also payment receipt can be email It eliminates the use of items like pen, paper, etc.

1.2 Problem Statement

Every student has to feel the burden of standing in a long queue, waiting all day just to submit their Exam form, Online exam form portal System has reduced that burden to 0%. Since the traditional have many drawbacks such as time consuming, Difficulty of analyzing the test manually, more observers are required to fill exam form of many students. Limitation of number of students can fill examination form at a time. Now students can submit their form online from their home or cyber cafe without standing or waiting in a long queue.

1.3 Objective

- To design and Develop web application according to the need of student.
- To provide proper solution to the student according to their problems and enquiries
- To generate reports based on all the queries and problems which gives brief details about particular problems and status.
- To save the wastage of time of manual method of checking student and submit exam form.

1.4 Thesis Organization

Chapter 1 consists of the introduction about why examination exam form system is required. Chapter 2 presents the software requirement specification document. Chapter 3 discusses work done and technologies adopted for the work. The description regarding every step involved in proposed work is explained in this chapter. Chapter 4 presents system screenshots and coding. Chapter 5 consists of result and discussion. Chapter 6 gives summary and conclusion of project and Chapter 7 consist of websites referred for this project.

2. Software Requirement Specification

2.1 Software Requirement

- Back end: PHP
- Database: PHPMyadmin, MySQL database.
- Front End : HTML, CSS, Bootstrap, JavaScript, jQuery.
- H/W requirement: Processor: 3 mhz , Ram: 4 gb , HDD: 500gb.

2.2 Literature Survey

Sr No.	Website / Software	Area	Topics covered
1.	www.W3schools.com	Web Development	HTML,CSS, . jQuery.
2.	www.php.w3cschool.com	Coding	PHP
3.	www.instamojo.com	Payment	Payment gateway
4	www.tutorialspoint.com	Web Development	Bootstrap
5	www.javascript.w3cschool.com	Web Development	Javascript

Table 2.2 Literature Survey

3. Work done

3.1 Introduction

3.1.1 Software (XAMP Server, Notepad++)

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.
It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well.
XAMPP has four primary components. These are:

1. **Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.
2. **MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world's most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress.
3. **PHP:** PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.
4. **Perl**: Perl is a high-level, dynamic programming language used extensively in network programming, system admin, etc. Although less popular for web development purposes, Perl has a lot of niche applications.

Different versions of XAMPP may have additional components such as phpMyAdmin, OpenSSL, etc. to create full-fledged web servers.

How to Install XAMPP?

Head over to XAMPP for Windows homepage on ApacheFriends.org. Here, we will find multiple versions of XAMPP.

we will install **XAMPP Windows** which includes:

Component	Description	Version
Apache	Default web server application	2.4.4
MySQL	Database management system	5.5.32
PHP	Server-side, general purpose programming language	5.4.19
phpMyAdmin	Admin tool for working with MySQL	4.0.4
OpenSSL	An open-source implementation of two popular security protocols – SSL and TSL	0.9.8
XAMPP Control Panel	A simple control panel for working with different XAMPP components	3.2.1
Webalizer	An analytics tool that generates user logs and usage metrics.	2.23
MercuryMailTransport System	A simple, open-source mail server	4.62
FileZilla FTP Server	A FTP (File Transfer Protocol) server to make file transfers smoother	0.9.41
Tomcat	A freeware Java servlet for serving Java applications	7.0.42
Strawberry Perl	A popular distribution of Perl for Windows	5.16.3.1

Unless you are running a live web server, We won't need anything beyond Apache, MySQL and PHP, although it is a good practice to install all other components as well.We also have the option of installing a smaller 'XAMPP Portable Lite' version, which only includes essential Apache, MySQL, PHP and phpMyAdmin components.

3.1.2 Notepad++ is a text editor and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window. The project's name comes from the C increment operator. Notepad++ is distributed as free software. At first the project was hosted on SourceForge.net, from where it has been downloaded over 28 million times, and twice won the Source Forge Community Choice Award for Best Developer Tool. The project was hosted on Tux Family from 2010 to 2015; since 2015 Notepad++ has been hosted on GitHub. Notepad++ uses the Scintilla editor component.

Features:
Notepad++ is a source code editor. It features syntax highlighting, code folding and limited autocompletion for programming, scripting, and mark-up languages, but not intelligent code completion or syntax checking. As such it may properly highlight code written in a supported schema but whether the syntax is internally sound or compliable cannot be verified. As of version 4.7.2, Notepad++ can highlight the syntactic elements of:
ActionScript
Ada
ASN.1
ASP
Assembly
AutoIt

AviSynth scripts
BaanC
Bash scripts
batch files
Blitz Basic
C
C#
C++
Caml
CMake
Cobol
CoffeeScript
Csound
CSS
D
Diff
Erlang
escript
Forth
Fortran
FreeBASIC
Gui4Cli
Haskell
HTML
INI files
Inno Setupscripts
Java
JavaScript
JSON
KiXtart
LaTeX
LISP
Lua
Makefile
Matlab
MMIX
Nimrod
nnCron
NSIS scripts
Objective-C
OScript
Pascal
Perl
PHP

- PostScript
- PowerShell
- PureBasic
- Python
- R

- Rebol
- Resource file
- Ruby
- Rust
- Scheme
- Smalltalk
- SPICE
- SQL
- Swift
- S-Record
- Tcl
- TeX
- txt2tags
- Visual Basic
- VHDL
- Verilog
- XML
- YAML

The language list also displays two special-case items for ordinary plain text: "Normal text" (default) or "MS-DOS Style", which tries to emulate DOS-era text editors.
Notepad++ has features for consuming and creating cross-platform plain text files. It recognizes three newline representations (CR, CR+LF and LF) and can convert between them on the fly. In addition, it supports reinterpreting plain text files in various character encodings and can convert them to ASCII, UTF-8 or UCS-2. As such, it can fix plain text that seem gibberish only because their character encoding is not properly detected.

Notepad++ also has features that improve plain text editing experience in general, such as:
Autosave
Finding and replacing strings of text with regular expressions
Guided indentation
Line bookmarking
Macros
Simultaneous editing
Split screen editing and synchronized scrolling
Line operations, including sorting, case conversion (Uppercase, lowercase, camel case, sentence case), and removal of redundant whitespace

3.2 Framework

Bootstrap is a free front-end framework for faster and easier web development
Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
Bootstrap also gives you the ability to easily create responsive designs

Advantages of Bootstrap:

- Easy to use**:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- Responsive features**:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- Mobile-first approach**:** In Bootstrap 3, mobile-first styles are part of the core framework
- Browser compatibility**:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

1. Add the HTML5 doctype

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.
Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  </head>
</html>
```

2. Bootstrap 3 is mobile-first

Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.
To ensure proper rendering and touch zooming, add the following `<meta>` tag inside the `<head>`element:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
The `initial-scale=1` part sets the initial zoom level when the page is first loaded by the browser.

3. Containers

Bootstrap also requires a containing element to wrap site contents.
There are two container classes to choose from:
The `.container` class provides a responsive fixed width container
The `.container-fluid` class provides a full width container, spanning the entire width of the viewport
.container
.container-fluid

Two Basic Bootstrap Pages
The following example shows the code for a basic Bootstrap page (with a responsive fixed width container):

Example

```
<!DOCTYPE html>
<html lang="en">
<head>  <title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/boo
```

```
tstrap.min.css"><script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.
min.js"></script>
<scriptsrc="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></s
cript>
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

The following example shows the code for a basic Bootstrap page (with a full width container):

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
 <title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bots
trap.min.css"><script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.
min.js"></script><scriptsrc="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

3.2.1 Usecase Diagram

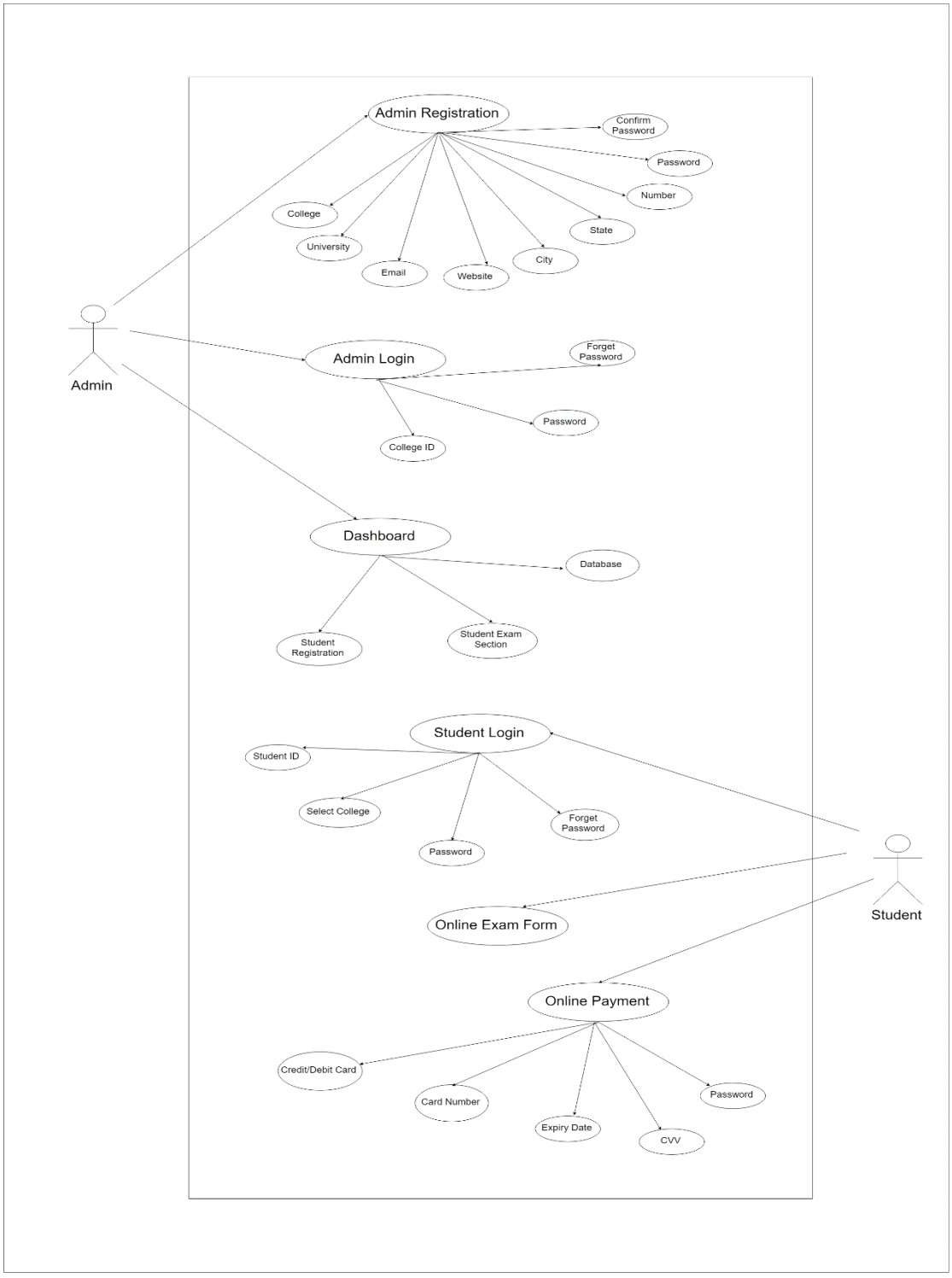


fig: 3.2.2.1 Use Case diagram

3.2.2 Flow chart

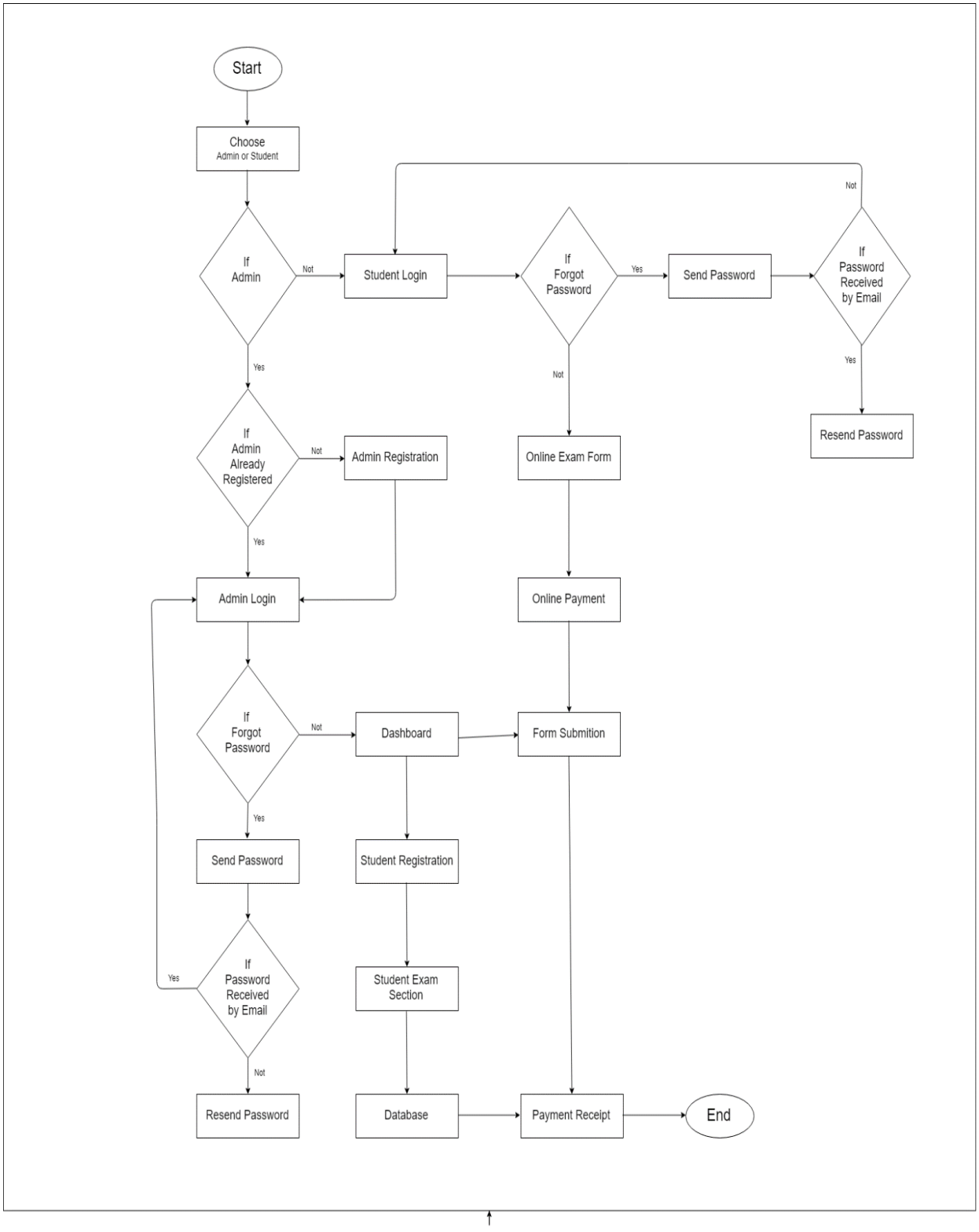


fig: 3.2.1.2 Work flow of system

Above Work Flow diagram can be shown follows:

Mainly it has three modules:

- Admin Panel
- Student Panel
- Payment Gateway

- **Admin Panel:**
  First admin panel sign in the system and if admin is not signing the sign up then admin can be successfully login into the system. Admin Panel can be do the generating student exam form it gives the privileges to edit delete add modify student data. Admin panel has given privileges to add department data, edit department, all department. Add course, edit course, all course, entering result of all student and generate exam form of student, admin can be also entering the result of student to extract the courses of student in database. Admin can be adding department courses student and see the payment and form data. Admin can be generating the exam form data of student and gives the permission for payment after submitting form also admin can be sent the email to student if any query. Admin can be seeing the ratio of how many students can be submit the form and how many students can be in the databased and how many payments can be done and how many departments in the system and how many courses in database.

- **Student panel:**
  Student can be directly login into the system and the checking the form is available or not if the form is available then fill up the form and after filling up the form then go to payment option. Student can be seeing the start date of exam form end date of exam form date fees date and late fees per day.

- **Payment Gateway:**
  After submitting exam form student can be go to payment mode total amount can be display after clicking on payment mode then choose the card in payment mode three type of card are available.1. credit card2.debit card.3. net banking. Then student can be choosing the card after that enter the card details and then payment can be success. But in payment gateway gives the extra charges because it is free payment gateway. And after that student will receive the payment receipt. In case student click on payment mode and not enter card detail then student will get the email to requesting for payment.

3.2.3 System diagram

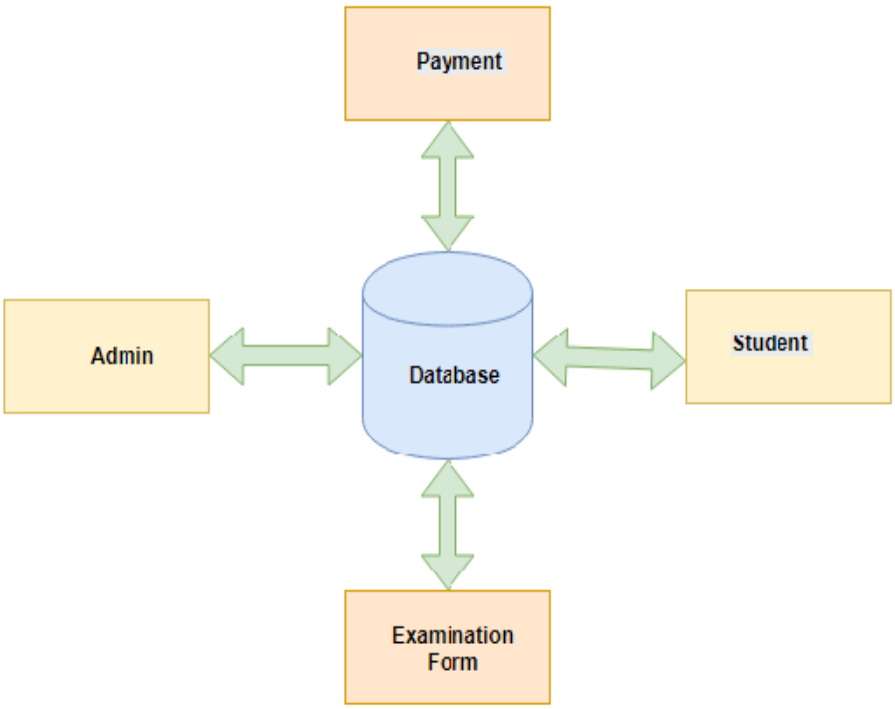


fig: 3.2.3 System Diagram

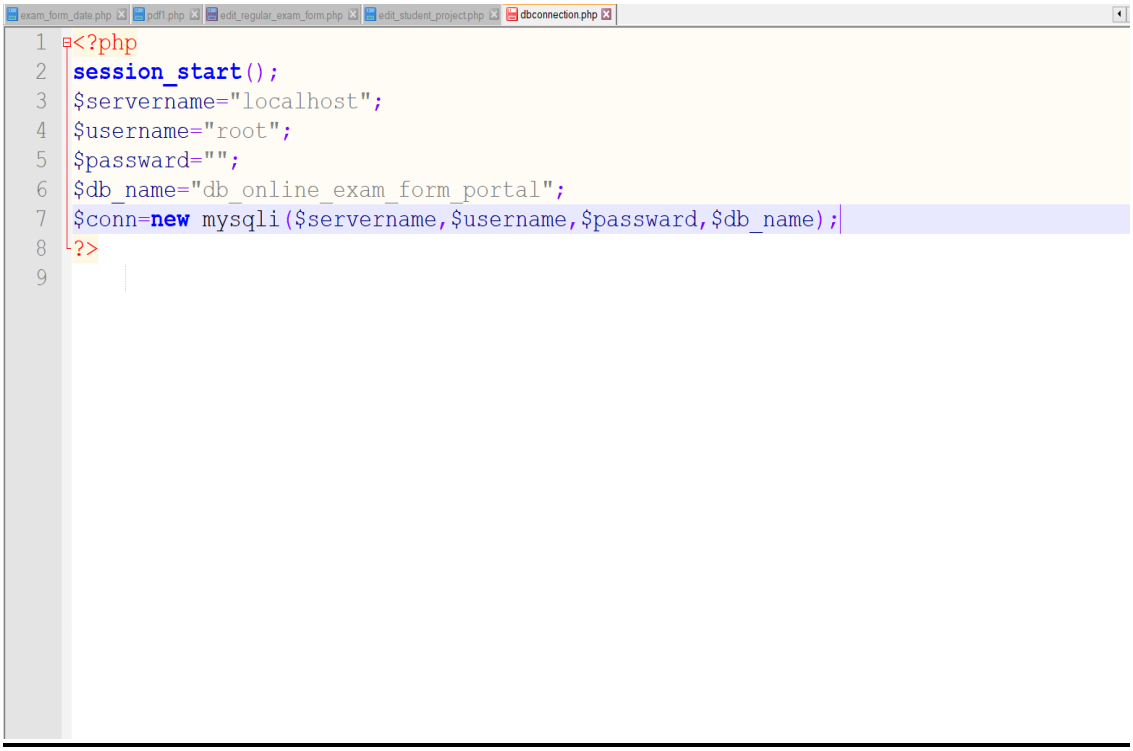
4. Coding

4.1 Snapshots of coding

exam_form_date.php | pdf1.php | edit_regular_exam_form.php | edit_student_project.php | dbconnection.php | mainpage.php | studentlogin.php

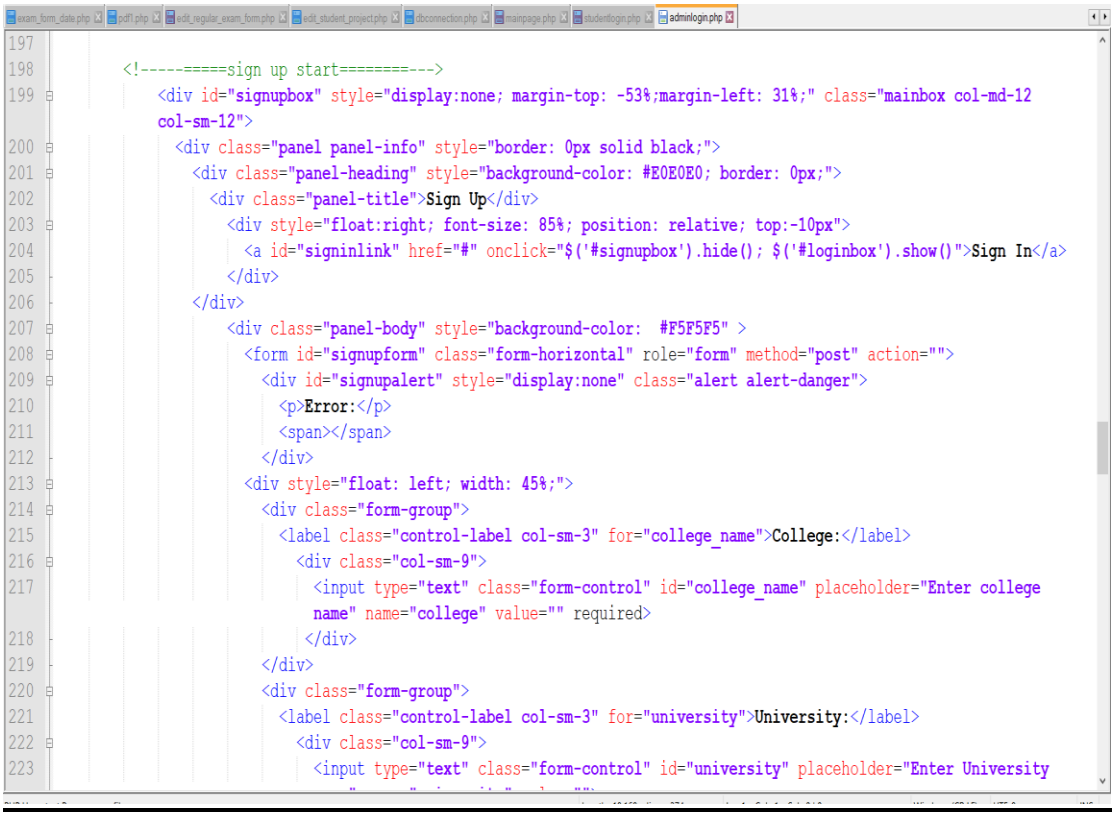
```
</head>
<body class="bgg" style="background-color: #FAFAD2;"> <!--style="background-color: #F5F5F5;"-->

<div class="" style="background-color: #303030;">
  <div class="page-header text-center top_header_heading" style="border-bottom: 0px;">
    <h1 style="font-size: 50px; color: white;">Examination Form Portal</h1>
  </div>
</div>
<div class="container" style="margin-top: 50px;">
  <div style="margin-left: 5%; float: left;">
    <img src="male-icon-32.png" width= "300" height= "300"><br>

    <a href="http://localhost/project/adminlogin.php" target="_blank" ><!--target="_blank"-->
    <button type="button"  class="btn btn-primary col-md-8" style="margin-left: 15%;" >Admin Login</button>
    </a>
  </div>
  <div style="float:right; margin-right: 5%;">
    <img src="student_icon.png" width= "300" height= "300"><br >
    <a href="http://localhost/project/studentlogin.php" target="_blank"><!--target="_blank"-->
    <button type="button" class="btn btn-primary col-md-8" style="margin-left: 20%;">Student Login</button>
    </a>
  </div>
</div>
<div class="footer">
  <p>2018 @College Online Examination Form With Online Payment</p>
</div>
</body>
</html>
```

fig: 4.1.1 Main Page



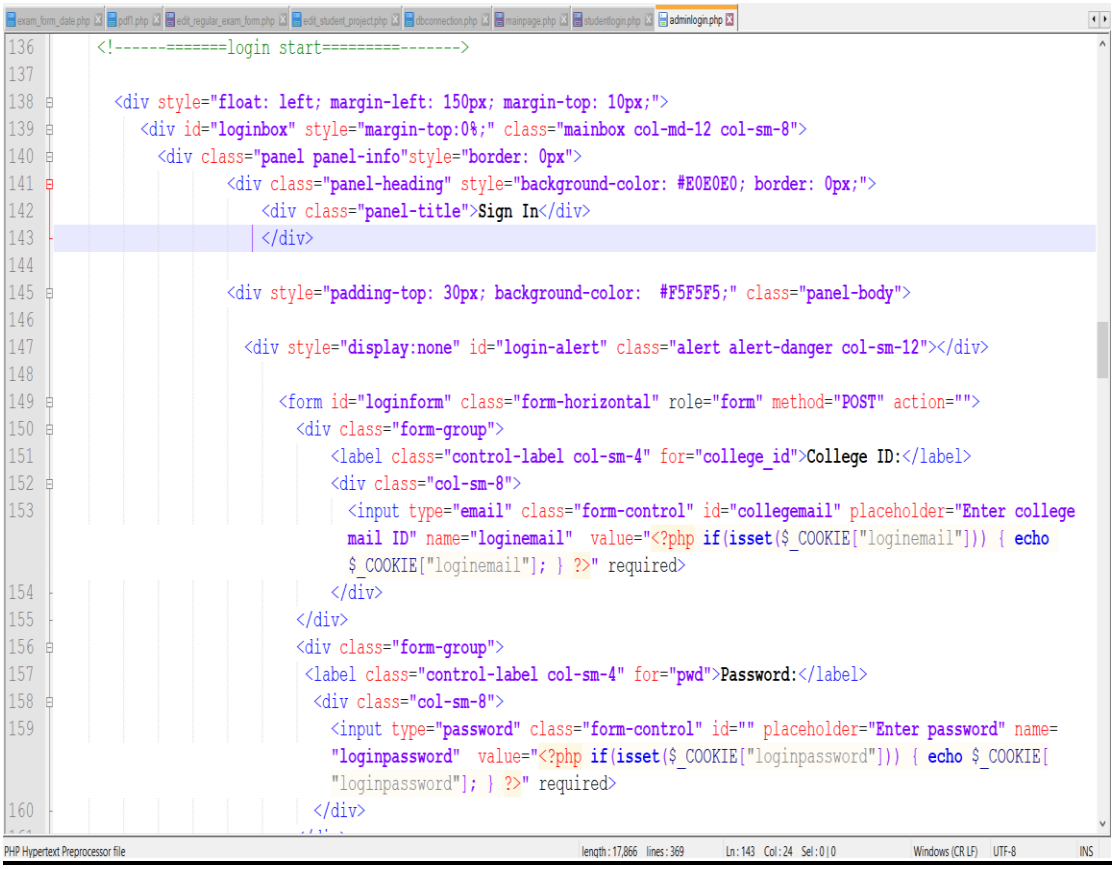
```php
<?php
session_start();
$servername="localhost";
$username="root";
$passward="";
$db_name="db_online_exam_form_portal";
$conn=new mysqli($servername,$username,$passward,$db_name);
?>
```

fig: 4.1.2 Database configuration



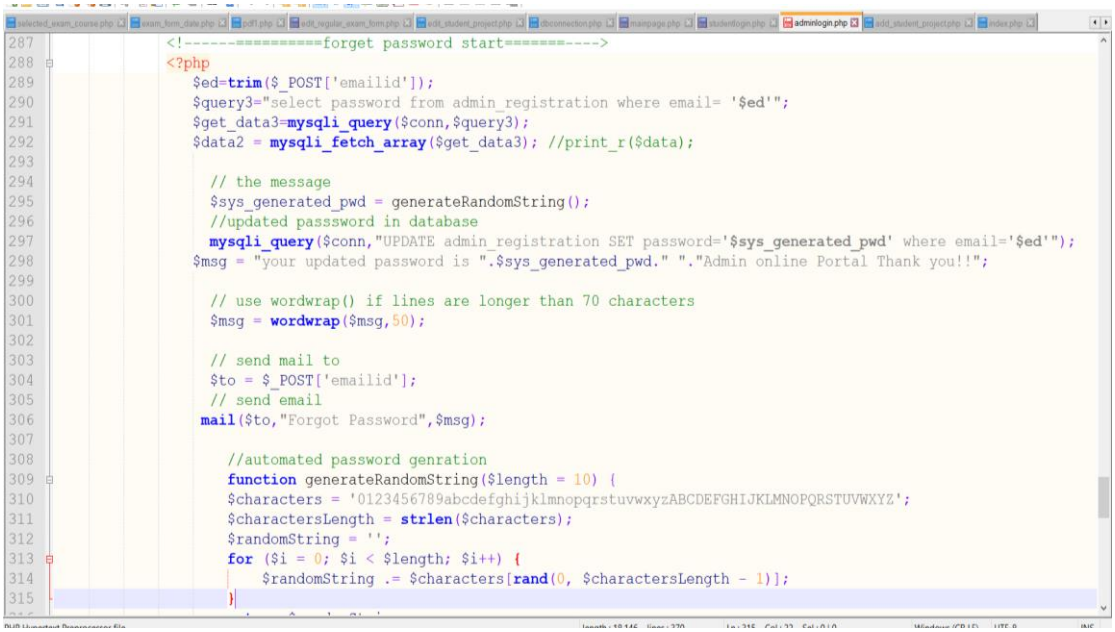
```
<!----------sign up start========--->
    <div id="signupbox" style="display:none; margin-top: -53%;margin-left: 31%;" class="mainbox col-md-12
    col-sm-12">
      <div class="panel panel-info" style="border: 0px solid black;">
        <div class="panel-heading" style="background-color: #E0E0E0; border: 0px;">
          <div class="panel-title">Sign Up</div>
            <div style="float:right; font-size: 85%; position: relative; top:-10px">
              <a id="signinlink" href="#" onclick="$('#signupbox').hide(); $('#loginbox').show()">Sign In</a>
            </div>
        </div>
            <div class="panel-body" style="background-color:  #F5F5F5" >
              <form id="signupform" class="form-horizontal" role="form" method="post" action="">
                <div id="signupalert" style="display:none" class="alert alert-danger">
                  <p>Error:</p>
                  <span></span>
                </div>
                <div style="float: left; width: 45%;">
                  <div class="form-group">
                    <label class="control-label col-sm-3" for="college_name">College:</label>
                      <div class="col-sm-9">
                        <input type="text" class="form-control" id="college_name" placeholder="Enter college
                        name" name="college" value="" required>
                      </div>
                  </div>
                  <div class="form-group">
                    <label class="control-label col-sm-3" for="university">University:</label>
                      <div class="col-sm-9">
                        <input type="text" class="form-control" id="university" placeholder="Enter University
```

fig: 4.1.3 Admin Signup



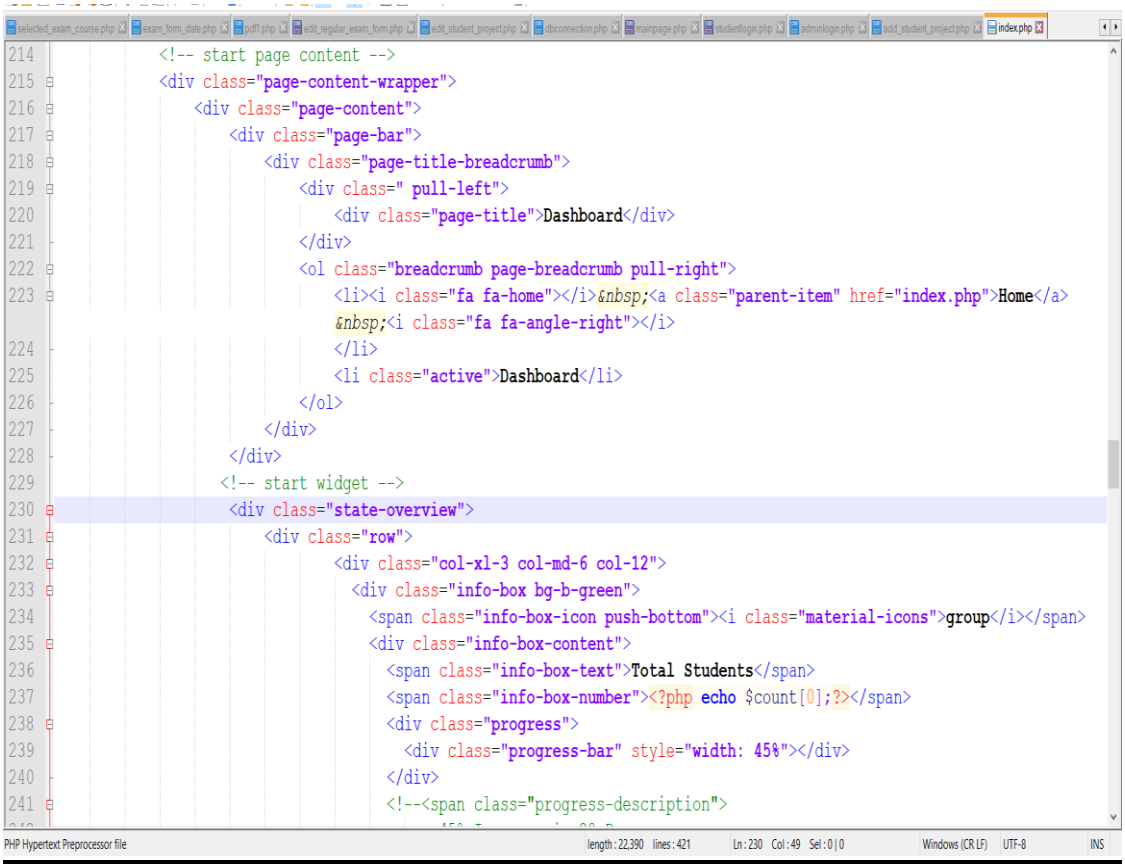
```
<!----------=======login start=========------->

  <div style="float: left; margin-left: 150px; margin-top: 10px;">
    <div id="loginbox" style="margin-top:0%;" class="mainbox col-md-12 col-sm-8">
      <div class="panel panel-info"style="border: 0px">
            <div class="panel-heading" style="background-color: #E0E0E0; border: 0px;">
                <div class="panel-title">Sign In</div>
              </div>

            <div style="padding-top: 30px; background-color:  #F5F5F5;" class="panel-body">

             <div style="display:none" id="login-alert" class="alert alert-danger col-sm-12"></div>

                <form id="loginform" class="form-horizontal" role="form" method="POST" action="">
                  <div class="form-group">
                      <label class="control-label col-sm-4" for="college_id">College ID:</label>
                      <div class="col-sm-8">
                        <input type="email" class="form-control" id="collegemail" placeholder="Enter college
                        mail ID" name="loginemail"  value="<?php if(isset($_COOKIE["loginemail"])) { echo
                        $_COOKIE["loginemail"]; } ?>" required>
                      </div>
                  </div>
                  <div class="form-group">
                   <label class="control-label col-sm-4" for="pwd">Password:</label>
                    <div class="col-sm-8">
                      <input type="password" class="form-control" id="" placeholder="Enter password" name=
                      "loginpassword"  value="<?php if(isset($_COOKIE["loginpassword"])) { echo $_COOKIE[
                      "loginpassword"]; } ?>" required>
                    </div>
```

fig: 4.1.4 Admin SignIn



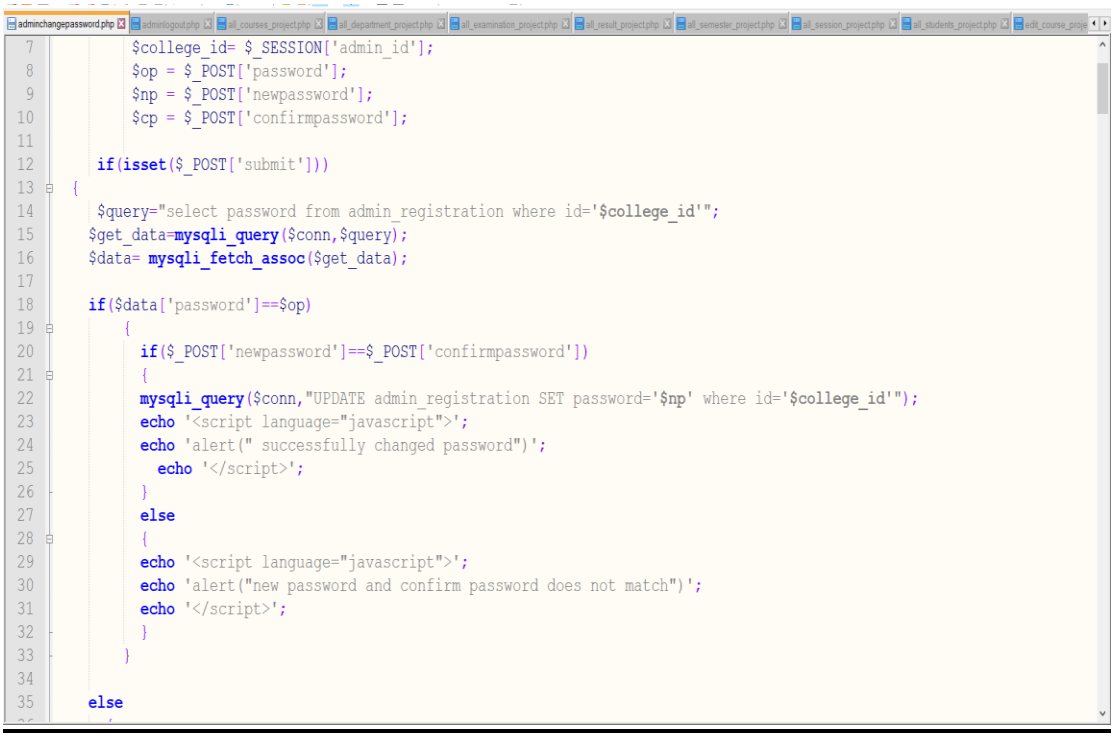
```
<!----------==========forget password start==========---->
<?php
   $ed=trim($_POST['emailid']);
   $query3="select password from admin_registration where email= '$ed'";
   $get_data3=mysqli_query($conn,$query3);
   $data2 = mysqli_fetch_array($get_data3); //print_r($data);

    // the message
    $sys_generated_pwd = generateRandomString();
    //updated passsword in database
    mysqli_query($conn,"UPDATE admin_registration SET password='$sys_generated_pwd' where email='$ed'");
   $msg = "your updated password is ".$sys_generated_pwd." "."Admin online Portal Thank you!!";

    // use wordwrap() if lines are longer than 70 characters
    $msg = wordwrap($msg,50);

    // send mail to
    $to = $_POST['emailid'];
    // send email
   mail($to,"Forgot Password",$msg);

     //automated password genration
     function generateRandomString($length = 10) {
     $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
     $charactersLength = strlen($characters);
     $randomString = '';
     for ($i = 0; $i < $length; $i++) {
         $randomString .= $characters[rand(0, $charactersLength - 1)];
     }
```

fig: 4.1.5 Admin ForgotPassword



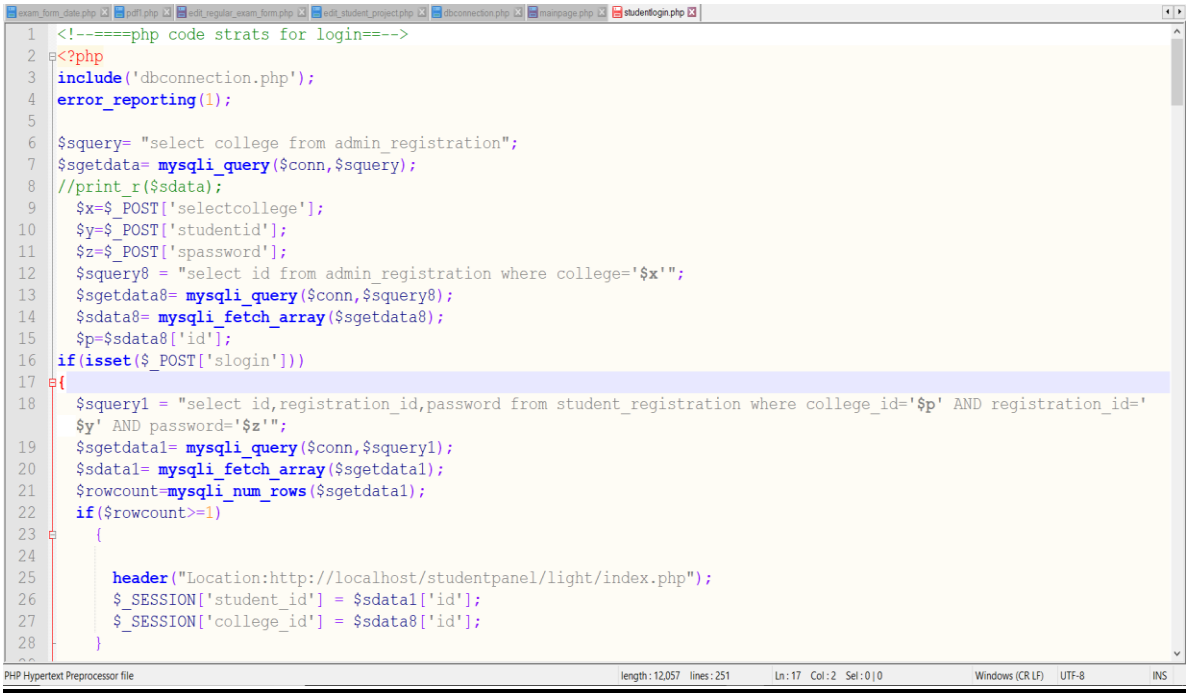
```
<!-- start page content -->
<div class="page-content-wrapper">
    <div class="page-content">
        <div class="page-bar">
            <div class="page-title-breadcrumb">
                <div class=" pull-left">
                    <div class="page-title">Dashboard</div>
                </div>
                <ol class="breadcrumb page-breadcrumb pull-right">
                    <li><i class="fa fa-home"></i> <a class="parent-item" href="index.php">Home</a>
                     <i class="fa fa-angle-right"></i>
                    </li>
                    <li class="active">Dashboard</li>
                </ol>
            </div>
        </div>
        <!-- start widget -->
        <div class="state-overview">
            <div class="row">
                <div class="col-xl-3 col-md-6 col-12">
                  <div class="info-box bg-b-green">
                    <span class="info-box-icon push-bottom"><i class="material-icons">group</i></span>
                    <div class="info-box-content">
                      <span class="info-box-text">Total Students</span>
                      <span class="info-box-number"><?php echo $count[0];?></span>
                      <div class="progress">
                        <div class="progress-bar" style="width: 45%"></div>
                      </div>
                      <!--<span class="progress-description">
```

fig: 4.1.6 Admin Dashboard



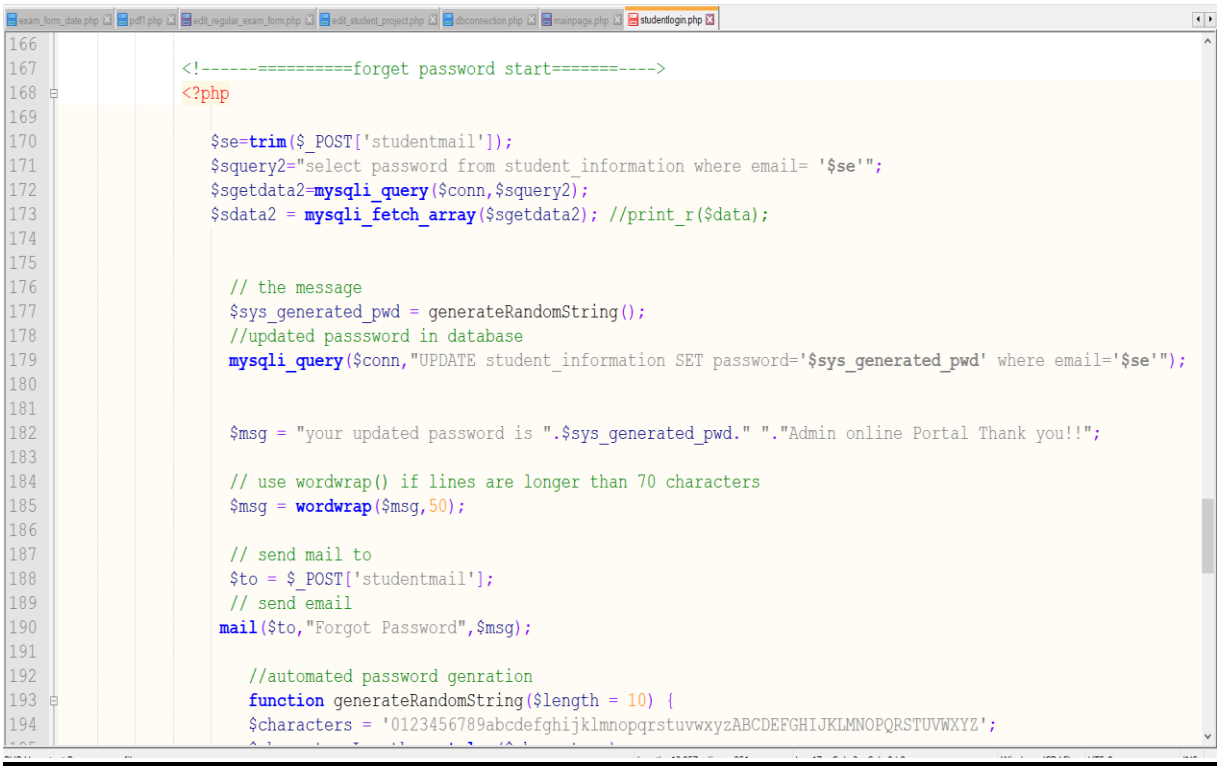
```
$college_id= $_SESSION['admin_id'];
$op = $_POST['password'];
$np = $_POST['newpassword'];
$cp = $_POST['confirmpassword'];

 if(isset($_POST['submit']))
{
 $query="select password from admin_registration where id='$college_id'";
$get_data=mysqli_query($conn,$query);
$data= mysqli_fetch_assoc($get_data);

if($data['password']==$op)
    {
      if($_POST['newpassword']==$_POST['confirmpassword'])
      {
      mysqli_query($conn,"UPDATE admin_registration SET password='$np' where id='$college_id'");
      echo '<script language="javascript">';
      echo 'alert(" successfully changed password")';
        echo '</script>';
      }
      else
      {
      echo '<script language="javascript">';
      echo 'alert("new password and confirm password does not match")';
      echo '</script>';
      }
    }

else
```

fig: 4.1.7 Admin ChangePassword



```php
<!--====php code strats for login==-->
<?php
include('dbconnection.php');
error_reporting(1);

$squery= "select college from admin_registration";
$sgetdata= mysqli_query($conn,$squery);
//print_r($sdata);
  $x=$_POST['selectcollege'];
  $y=$_POST['studentid'];
  $z=$_POST['spassword'];
  $squery8 = "select id from admin_registration where college='$x'";
  $sgetdata8= mysqli_query($conn,$squery8);
  $sdata8= mysqli_fetch_array($sgetdata8);
  $p=$sdata8['id'];
if(isset($_POST['slogin']))
{
  $squery1 = "select id,registration_id,password from student_registration where college_id='$p' AND registration_id='
  $y' AND password='$z'";
  $sgetdata1= mysqli_query($conn,$squery1);
  $sdata1= mysqli_fetch_array($sgetdata1);
  $rowcount=mysqli_num_rows($sgetdata1);
  if($rowcount>=1)
    {

      header("Location:http://localhost/studentpanel/light/index.php");
      $_SESSION['student_id'] = $sdata1['id'];
      $_SESSION['college_id'] = $sdata8['id'];
    }
```

fig: 4.1.8 Student Login



```php
<!------------forget password start=======---->
<?php

   $se=trim($_POST['studentmail']);
   $squery2="select password from student_information where email= '$se'";
   $sgetdata2=mysqli_query($conn,$squery2);
   $sdata2 = mysqli_fetch_array($sgetdata2); //print_r($data);


    // the message
    $sys_generated_pwd = generateRandomString();
    //updated passsword in database
    mysqli_query($conn,"UPDATE student_information SET password='$sys_generated_pwd' where email='$se'");


    $msg = "your updated password is ".$sys_generated_pwd." "."Admin online Portal Thank you!!";

    // use wordwrap() if lines are longer than 70 characters
    $msg = wordwrap($msg,50);

    // send mail to
    $to = $_POST['studentmail'];
    // send email
   mail($to,"Forgot Password",$msg);

     //automated password genration
     function generateRandomString($length = 10) {
     $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
```

fig: 4.1.9 Student Login Forgot Password

```
                                    <th>Sr.No</th>
                                    <th>Course Name</th>
                                    <th>Status</th>
                                    <th>Theory/Practicle</th>
                                    <th>Credit</th>
                                    <th>Tick/Untick</th>
                                </tr>
                            </thead>
                            <tbody>
                                <?php
                                $i=0;
                                $query2="select * from course where semester_id='$semsester_id
                                ' AND department_id='$department_id' AND college_id='
                                $college_id'";
                                $getdata2=mysqli_query($conn,$query2);
                                //$data2=mysqli_fetch_array($getdata2);
                                while($data2=mysqli_fetch_array($getdata2))
                                {
                                ?>
                                  <tr>
                                    <td><?php echo ++$i ?> </td>
                                    <td><?php echo $data2['course_code'] ?>-<?php echo $data2[
                                    'course_name'] ?> </td>
                                    <td><?php if($rowcount11>=1){ echo "EX";}else{ echo
                                    "Regular";} ?></td>
                                    <td> <?php echo $data2['course_type'] ?> </td>
                                    <td ><?php echo $data2['credit'] ?> </td>
                                    <td>
                                        <div class="form-check" style="margin-left:35%;">
```

fig: 4.1.10 Student Examination Form

```
<?php
$product_name = $_POST["product_name"];
$price = $_POST["product_price"];
$name = $_POST["name"];
$phone = $_POST["phone"];
$email = $_POST["email"];
include 'src/instamojo.php';
$api = new Instamojo\Instamojo('test_dbd3dd8adf9dca8aa1d9cdf9704', 'test_0cb70147d46b984de5b7ef5b5b2',
'https://test.instamojo.com/api/1.1/');
try {
    $response = $api->paymentRequestCreate(array(
        "purpose" => $product_name,
        "amount" => $price,
        "buyer_name" => $name,
        "phone" => $phone,
        "send_email" => true,
        "send_sms" => true,
        "email" => $email,
        'allow_repeated_payments' => false,
        "redirect_url" =>"http://onlineexamform.org/thankyou.php",
        "webhook" => "http://onlineexamform.org/webhook.php"
        ));
    //print_r($response);
    $pay_ulr = $response['longurl'];
    //Redirect($response['longurl'],302); //Go to Payment page
    header("Location: $pay_ulr");
    exit();

}
```

fig: 4.1.11 Payment gateway Configuration

5. Results and Discussions

5.1 Snapshots of System

A. Login Page for Admin as well as Student

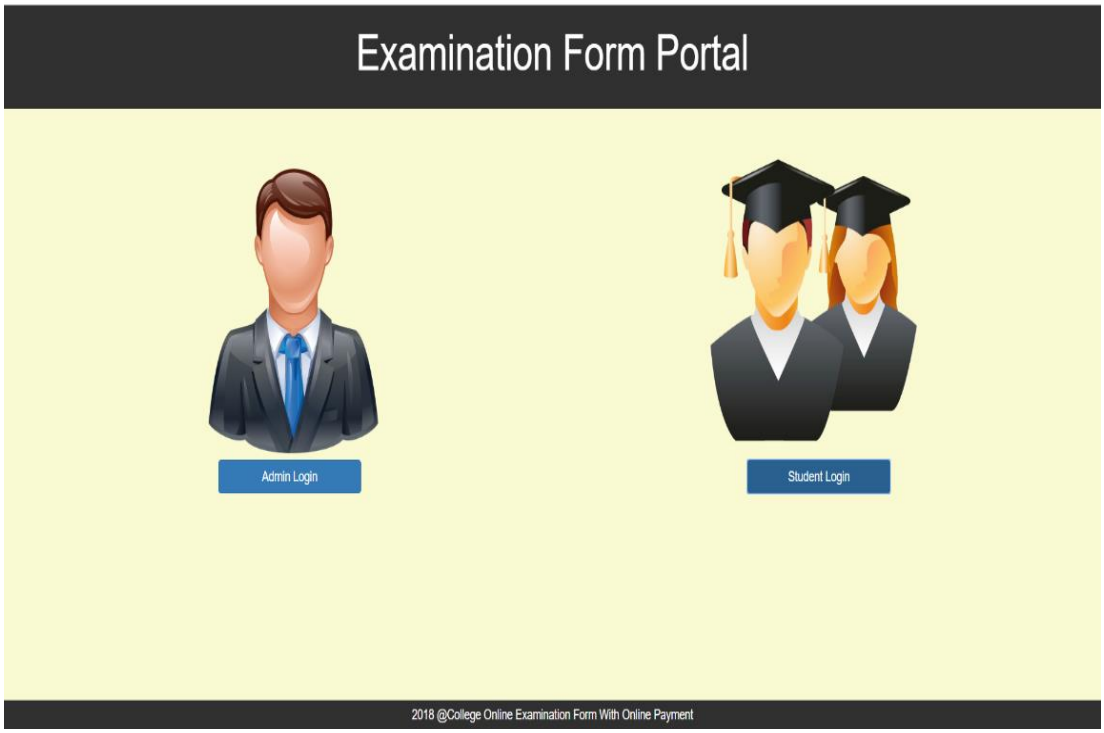


fig: 5.1.1 Login Page for Users

The above figure 5.1.1 shows the main page of our project that is Examination form portal. The main page contains two users option that are 1.Admin Login and 2.Student Login. If the user is admin then he should click on Admin Login button otherwise if student, the should click on Student Login button. This system used only register student, student can be registered option student can be directly login in to the system because student has already provided the registration-id and password from there college. And when click on Admin login admin can be login by its email-id and password. If admin has does not have email-id and password, then admin can be also register and then login into the system.

B. Login as Admin

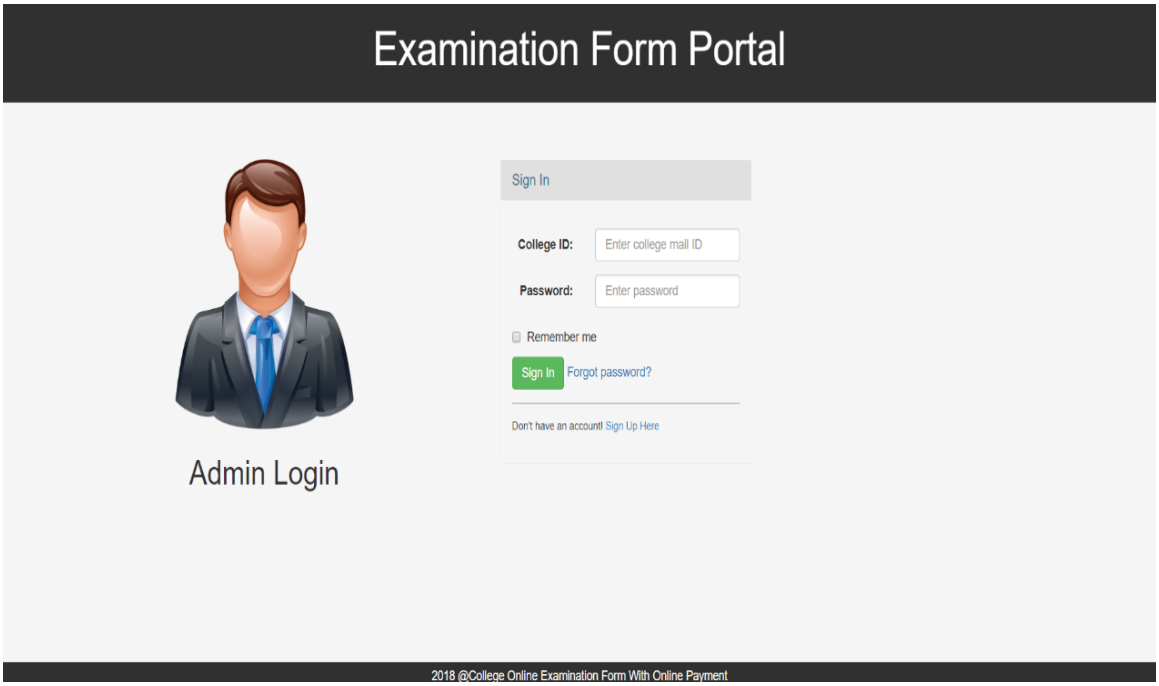


fig: 5.1.2 Admin's Login details

This above figure 5.1.2 shows the Admin login system. If Admin has been already registered with the system then he needs to fill the login details (college id, Password) For login to the admin dashboard. In this page one option is provided to the user that is Remember me with checkbox, after clicking over checkbox it stored the value inserted inside the input box in cookies for 24hr, if user again login in that time period then he does not need to re-enter the college id and his password. If user forget his password the he can click on Forgot password link. And if admin have not College-id and password then admin can be click on Sign Up Here link then admin can be registered then it can be login into the student.

C. Admin Sign Up:

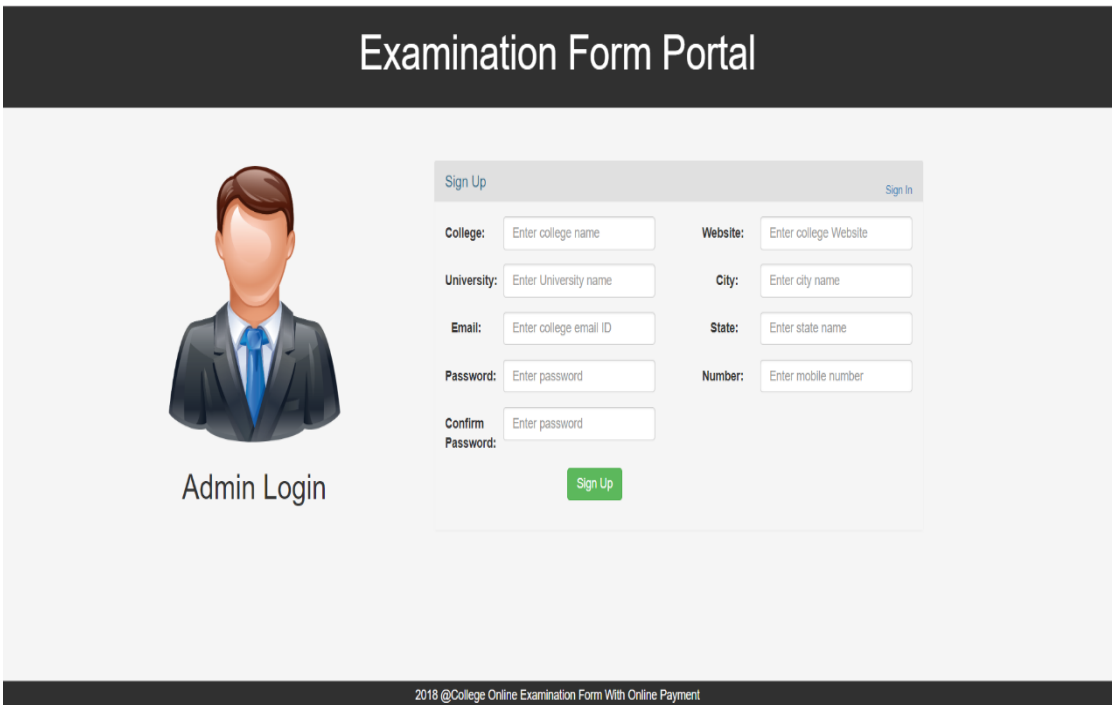


fig: 5.1.3 Admin's Signup

This above figure 5.1.3 shows Admin registration process to our system. If Admin user has not registered, then he needs to signup first after that he can directly sign in to the system anytime and can use our system. At the time of sign up admin then entered the college, university, email, password, confirm password, website, city, state, number.

D. Admin Forgot Password:

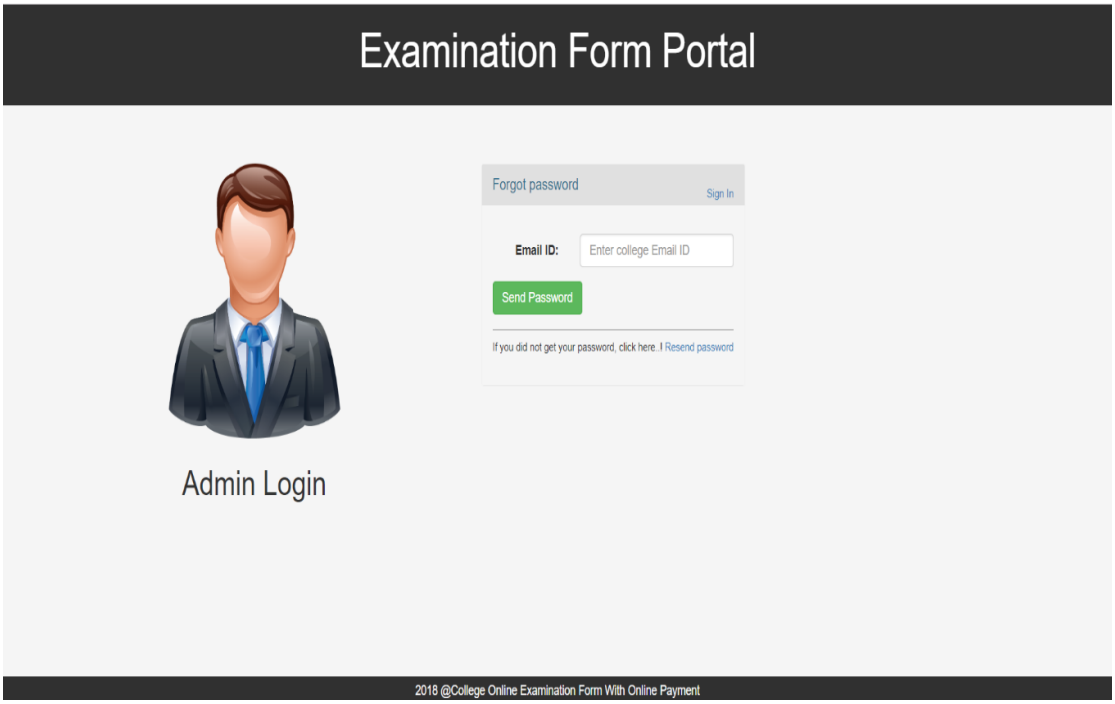


fig: 5.1.4 Admin Forgot Password

This above figure 4.1.4 shows password resets system. If the admin forgets his/her password then he can reset his/her password. The user need to enter his/her registered email id to get the password reset, If user does not get the mail there is an option of resend password. So, again he need to click on resend password option to his mail-id and then he can login to the system using that new password.

E. Admin Dashboard:

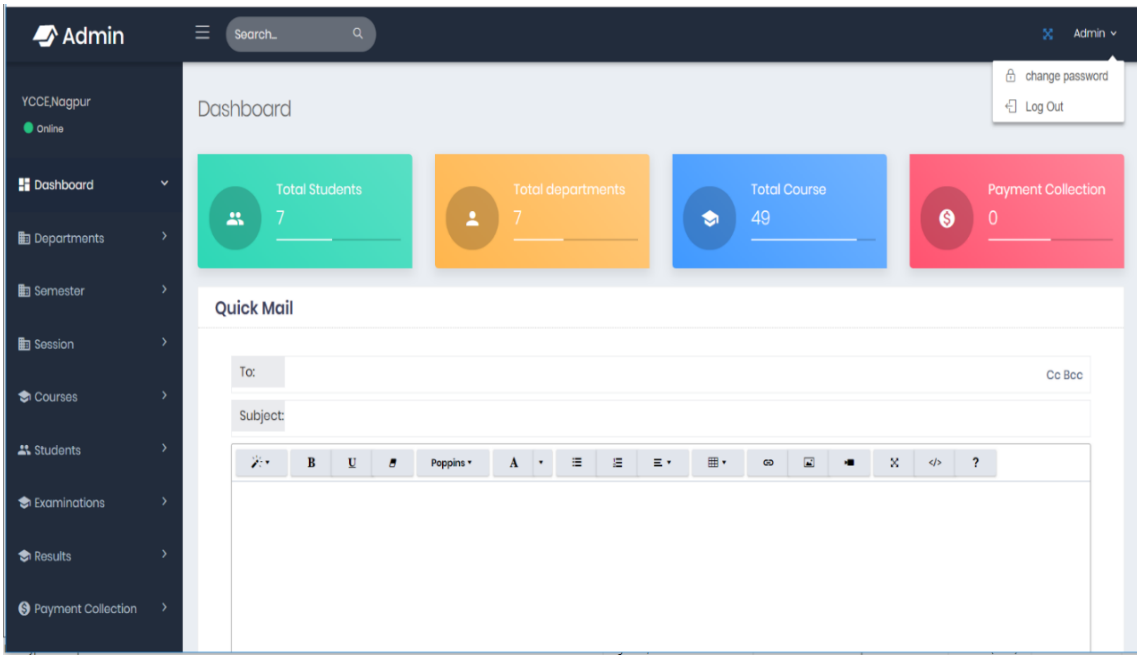


fig: 5.1.5 Admin Dashboard

When admin logs in figure 5.15 page opens. This is admin dashboard where admin need to do registration of Departments, Semesters, Session, courses, students. Admin can send mail to the students through this system for some certain notification. Also, admin can be looking the how many students can be registered in the system, how many department can be registered in the system, how many courses can be registered in the system, how many payment can be done. All privileges can be given to the admin, admin can be edit delete add course, student data, exam form, result, department. In dashboard gives the option of change password and log out in to the system.

F. Admin Change Password:

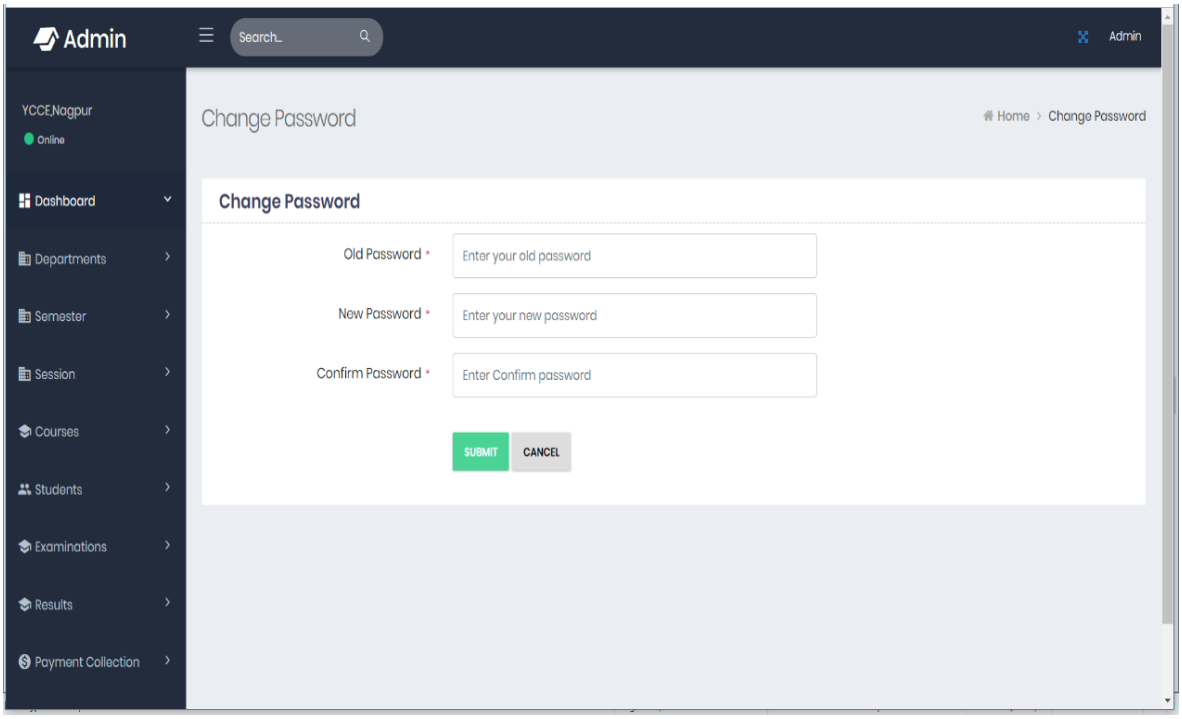


fig: 5.1.6 Admin Change Password

This above figure 5.1.6 shows admin change password system. If user(admin) wants to change his password so he can enter his old password(existing Password) New password(change password) and confirm password then the password get change and updates with new one. Now user can login by using his/her new password or change password.

G. All Student Data:

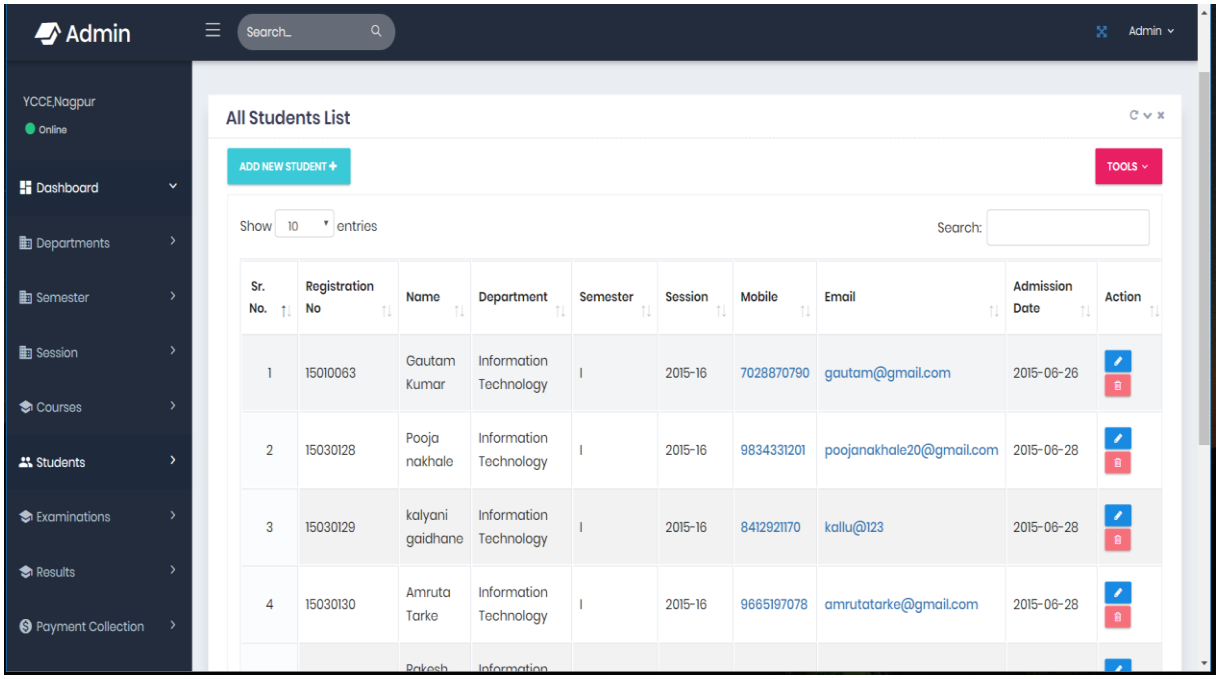


fig: 5.1.7 Admin (Student data table)

If admin clicks on a request tab it opens all request in which id, subject, requester name, assigns to, due by, status, created date, site, priority attributes are shown. Here admin can add new request, assign and can see all the actions. Admin can be registration of all student . either admin can be accessing all student data, edit student data. Delete student data and search the student data. Admin can have the all privileges to maintain the all over system and student data. For generating exam form of student. Admin also acknowledgment of payment which student done the payment or not.

H. Student Login:

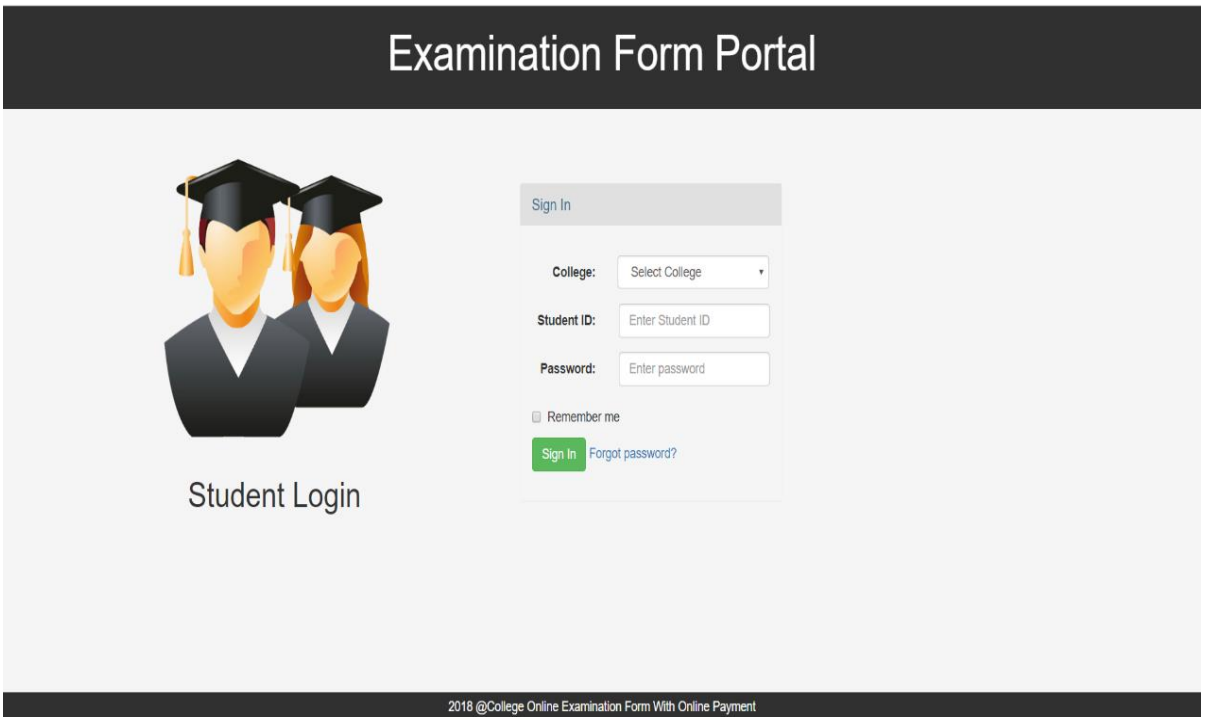


fig: 5.1.8 Student Login

This above figure 5.1.8 shows the Student login system. If Admin has been already registered with the system then he needs to fill the login details (registration id, Password) For login to the Student dashboard. In this page one option is provided to the user that is Remember me with checkbox, after clicking over checkbox it stored the value inserted inside the input box in cookies for 24hr, if user again login in that time period then he does not need to re-enter the choose college registration- id and his password. If user forget his password the he can click on Forgot password link. In Login page the college field can be contain which colleges are registered in the system only this colleges are available in college field and student-id and password then student can be login successfully in the system.

I. Student Dashboard:

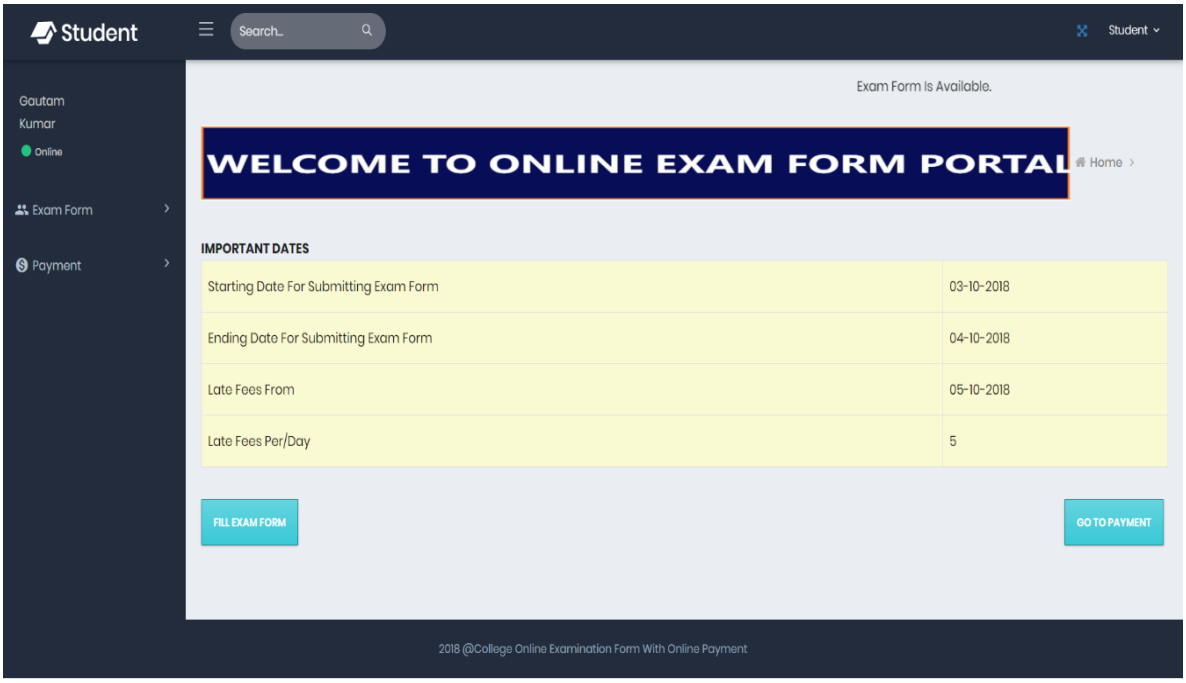


fig: 5.1.9 Student Dashboard

Above figure 5.1.9 shows the student dashboard. Student dashboard can be contain mainly exam form and payment option. When student can be click on the exam form then exam form can be display to the student, student can be filling this form and click on submit button for submit the form, and then click on payment then click on make payment button for payment of exam form. Also, student can be display the start date of exam form, end date of exam form, late fees of form and late fees per/day.

J. Exam Form of Student:

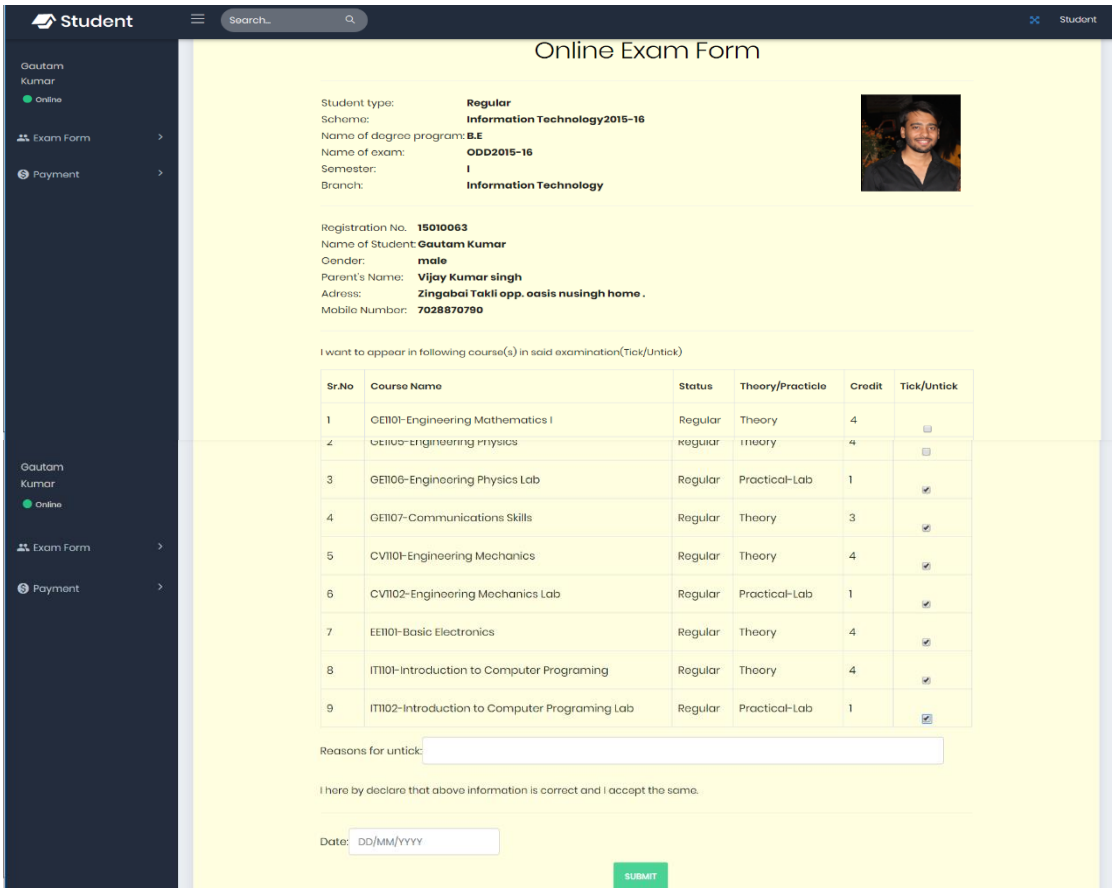


fig: 5.1.10 Student Examination Form

This above figure5.1.10 shows the student's examination form. Here the automated examination form generated with checkbox, student needs to click over the checkbox of his/her desired subject for which he wants to attend the exam after that he will get the list of subject which he had submitted there he will get three option before payment option that are 1. Edit the form, 2. Make Payment, 3. Save as a pdf option provided to the student.

K. Payment Gateway:

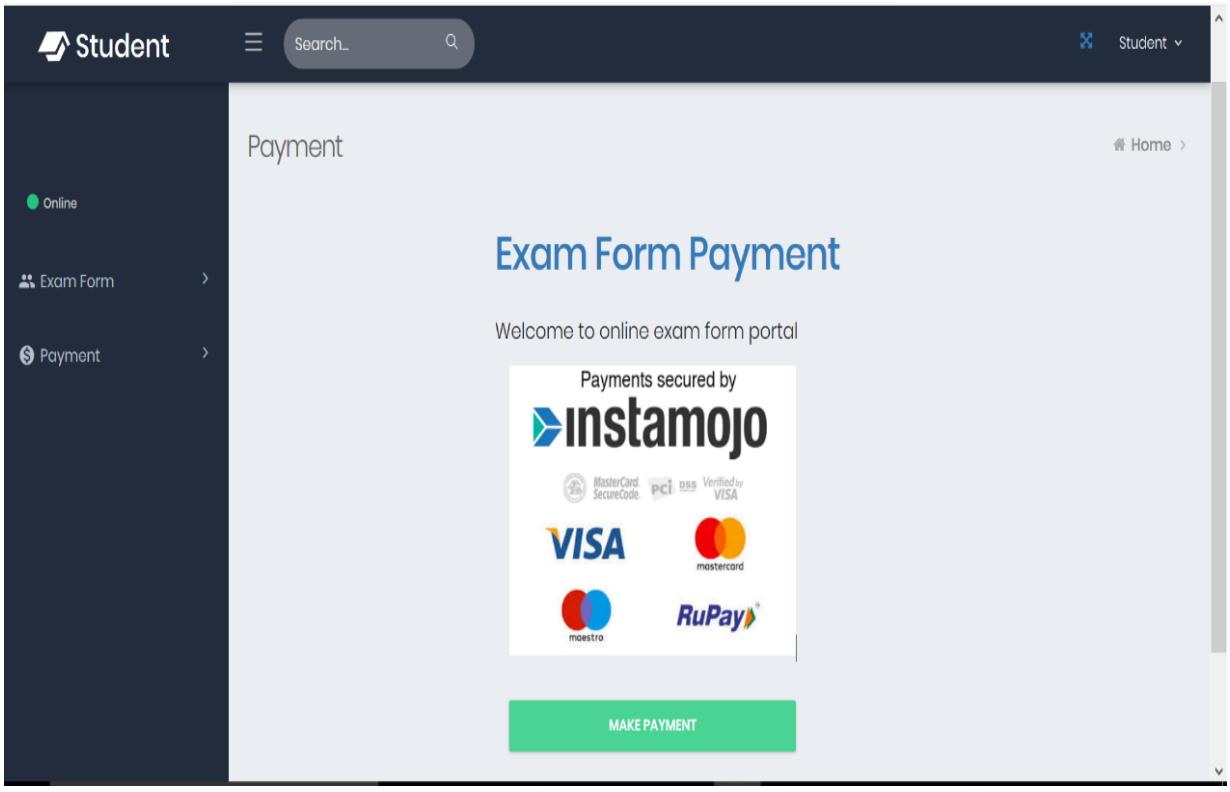


fig: 5.1.11 Instamojo Payment gateway

Above figure5.1.11 shows the instamojo payment gateway. When click on make payment button then student can be going to the payment mode. Why choose instamojo payment gateway because instamojo is an online payment gateway just like PayPal other payment gateway which allow users to Send, Receive Payments through many online options like net banking, Credit-Card, Debit Card etc. So, using Instamojo it is easier to send and receive payments online. Instamojo allow us to integrate the checkout system in the website, Apps etc for faster and easier checkout. As well as instamojo is fully secured and trusted website in all over the World. You Can receive payment from any place and Send payments to anywhere in the world.

L. Payment Amount Display:

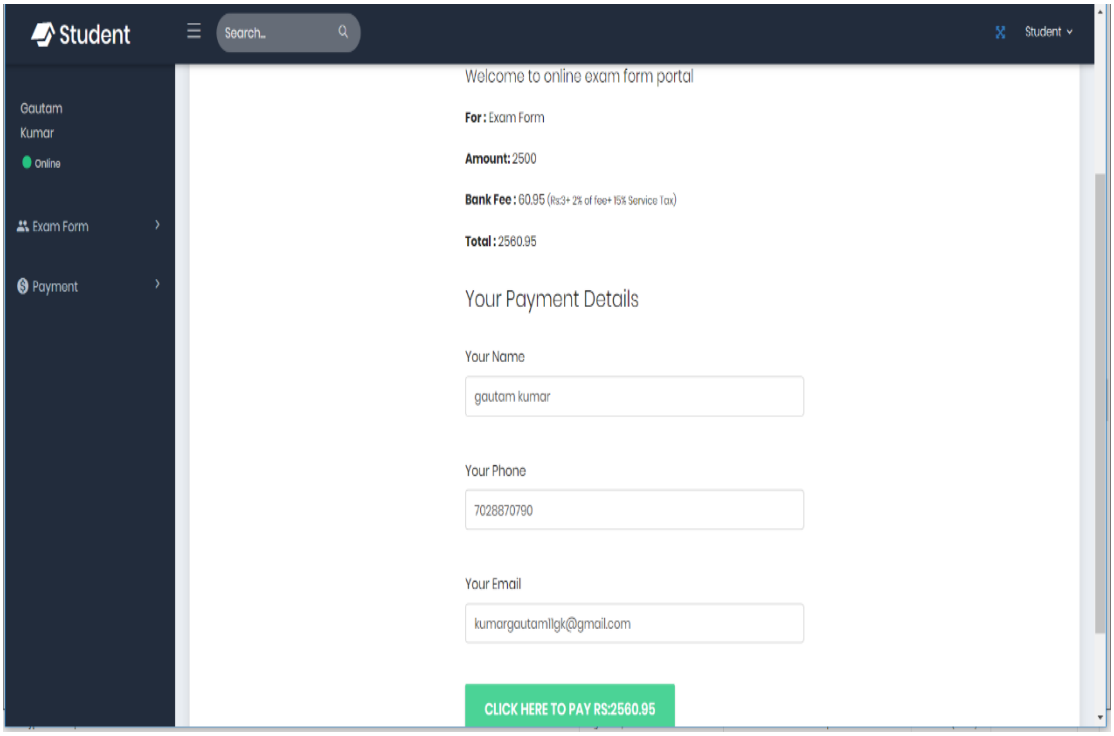


fig: 5.1.12 Make Payment

Above page shows the payment amount, how many total amounts can be calculated for form this display on this page as per the student type. If the student is regular, then 2500 amount can be display. Else Ex- student form amount calculated as per his subject calculation. But both amounts can be adding the extra 3% charges from student. Also entered the student information name, mobile number, email and then click on the click here to pay button. Then student can get the email from instamojo for requesting for payment.

M. Payment Card Select Option and Card Details:

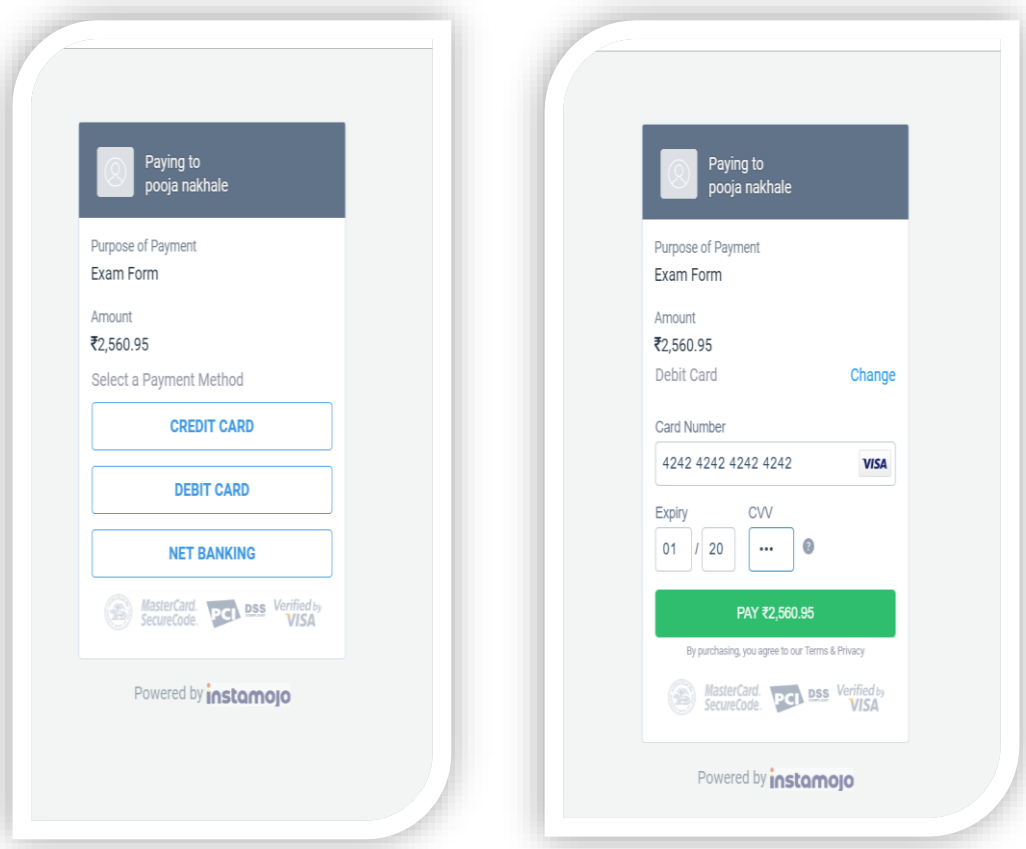


fig: 5.1.13 Choose Payment Option

This fig5.1.13 can be gives the payment type by using different card. Instamojo can gives the three type of card accepted 1. Credit Card, 2. Debit Card, 3.Net Banking. Student can be choose only one card. And entered the card number, expiry date and CVV then click on pay button.

N. Payment Receipt:

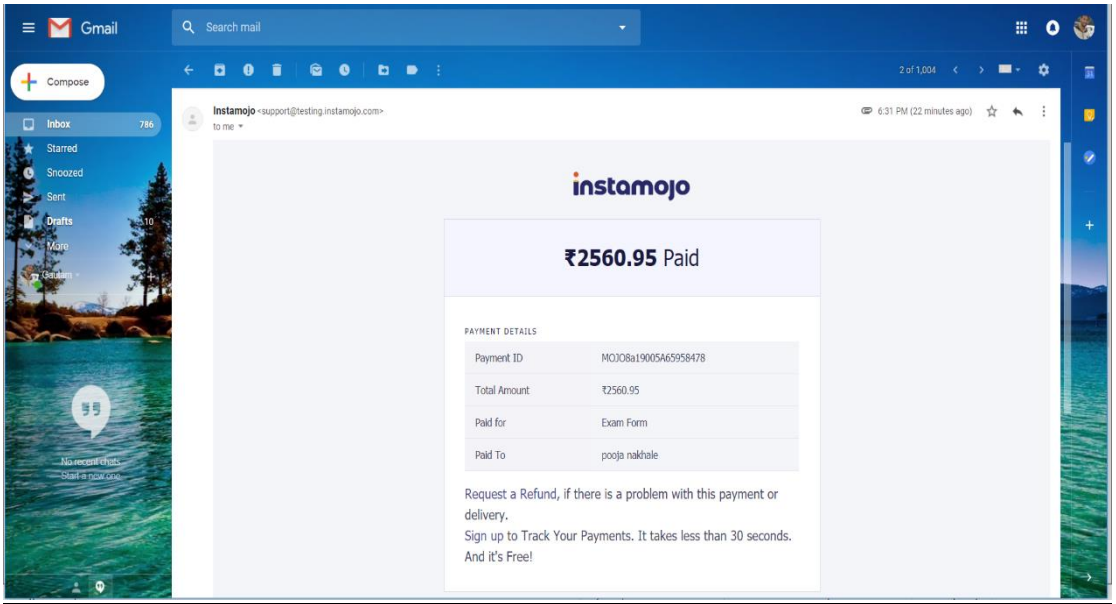


fig: 5.1.14 Payment Receipt

This fig5.1.15 can be show the payment receipt. The payment can be successfully then receipt can be automatically generated and sent to the student email-id in the form of message and pdf format.

6. Summary and Conclusions

Using an open source language gives us more flexibility, but at the same time it required more time to be programmed. The proposed Examination form Postal System can be easily adopted by universities and institutions in order to make the exam form more secure and more flexible. The system is subdivided into two main subsystems (student and administrator) that are designed to give the system maximum benefit by demonstrating carefully each subsystem service. The administrator's functions are clearly identified to be able to manipulate student information such as add (register), delete student data and managing the exam materials and content such as add, delete data. Thus, the proposed system is easy and flexible because for future maintenance and development because each subsystem can be handled separately without influence on other system.

- It saves paper.
- It saves more time
- It's more secure.

7. References

1. Bai Yi-chen, "The Design and Implementation of the Online Examination System Based on MVC Model [D]", *Shandong University*, 2012.
2. Chen Ying-ying, "The Design and Implementation of Online Examination System with Characteristics of Cloud Service [D]", *Beijing University of Posts and Telecommunications*, 2013.
3. Man Yong, "The Design and Implementation of Online Examine System [D]", *Hunan University*, 2012
4. Wang Kun, "The Design and Implementation of Online Examination System Based on. NET [D]", *Shandong University*, 2013.
5. Wang Shan, Sa Shi-xuan, "Introduction to Database System [M]" in , Beijing:Higher Education Press, 2006