“Project Report On”

COLLEGE MANAGEMENT SYSTEM USING DBMS

Submitted by

N. LALITHA KAVYA

DEPARTMENT OF MATHEMATICS

Dr.V.S. Krishna Government Degree College (A)

VISAKHAPATNAM

Guide

Mis. Y. Jahnavi

CERTIFICATE

This is to certify that the project report entitled "**COLLEGE MANAGEMENT SYSTEM USING DBMS**"

Submitted By

N LALITTHA KAVYA

record of work carried out by him in partial fulfilment of the requirement for the award of the Degree of Bachelor of Science (Computer Science), as prescribed by the Dr. V.S. Krishna Govt. Degree & P. G. College(A) in the Academic Year 2019 –2020.

Guide

Mis. Y. Jahnavi

Head of Department

ACKNOWLEDGEMENT

We take this opportunity to thank those who helped us immensely throughout this project and whose efforts we will always appreciate and remember.

We thank to **Mis. Y. JAHNAVI** for their valuable suggestions, their indefatigable efforts to review our work, inspiring us and helping us in all possible ways at any time.

We would also like to extend our sincere appreciation to all staff members of Computer Department, without their help and co-operation the Project report would not have been a success.

Last but not the least we express our extreme gratitude to the Almighty, without whose blessing nothing is possible.

DECLARATION

I hereby declare that this declare that this project report entitled “COLLEGE MANAGEMENT SYSTEM” is the result of original work done by me and to the best of my knowledge similar work has been submitted previously to any otheruniversity or published any time before. This project is submitted on the partialfulfilment the requirement for the award of degree of Bachelor of science.

N. LALITHA KAVYA

TABLE OF CONTENT

COLLEGE MANAGEMENT SYSTEM

ABSTRACT:

This project is based on COLLEGE MANAGEMENT SYSTEM. It manages the college information, student information, placement information, various different types of events going on in our college. It also keeps track records of all the information regarding students those who are placed in the various organization. It has a notice board which contains information about various cultural or technical or any sports which is supposed to be held soon.

OBJECTIVE:

The objects of system are –

- To reduce paperwork
- Reduce operational time
- Increased accuracy and reliability
- Increased operational efficiency
- Data security
- Reliable for student to create account and apply for job posts
- Placement head can maintain daily update
- New features can be added as per requirement

Java, JSP And MySQL Project on College Management System:

I have developed this project College Management System in Java, JSP and MySQL. This is a mini java project. There is a single user in this project. Admin is a user. In this project admin will manage all the things in this project like admin can add student, fees and attendance and also admin can see the student reports, fees reports and attendance reports. Admin can edit, update and delete the student, fees and attendance report. Admin will be able to change the account password. Here students can get java projects with source code free download

I have attached the Screens and demo of the project below the description. Look into more details.

Modules and Description of College Management System Project:

Student Module:

The main aim of this module is to provide all the detailed functionality related to college. We have developed all types of CRUDS (Create, Read, Update and Delete) operations of the College. This College Module is an important module in this mini project College Management System which has been developed on Java, JSP and MySQL. Students can get complete easy java projects with source code and documentation for free download.

Features of Student Module:

- Admin can add new Student records
- Admin can see the list of Student details
- Only admin can edit and update the record of the student
- Admin will be able to delete the records of the student
- All Student forms are validated on client side using JavaScript

Attendance Module:

The main purpose for developing this module is to manage the attendance of the student's data wise. So, all attendance will be marked by admin, students will be able to see only his/her attendance. Admin can see the list of all attendance and filter it according to the students.

Features of Attendance Module:

- Admin can mark the student attendance
- Admin can edit/delete the student attendance
- Admin can see the list of all student attendance
- Student can see his attendance

Fees Module:

This module has been developed for managing all the information of the fees, here admin can upload the fees and students can view and download the fees

Features of Fees Module:

- Admin can upload new fees
- Student can view and download the fees
- Admin can edit/update the existing fees
- Admin can delete the existing fees
- Admin can see the list of all fees

Login Module:

This is the login form, from where Admin can login into the system.

Features of Login Module:

- Admin can add new Login records
- Admin can see the Login details
- Only admin can edit and update the record of the Login
- Admin can change the password

Functionality and modules of the project College Management System:

- Login – This module is used for admin login.
- Logout Functionality
- Change Password Functionality
- Dashboard – Admin dashboard related to all Students, Students detail, Students listing.

Student Management Module

- ✓ Adding New Student Details
- ✓ Edit the Existing Student Details
- ✓ View all the details of the student
- ✓ Listing of all Student

→ Fees Module

- ✓ Adding New Fees Details Edit the Existing Fees Details
- ✓ View all the details of the Fees
- ✓ Listing of all Fees

→ Attendance Module

- ✓ Adding New Details Attendance
- ✓ Edit the Existing Attendance Details
- ✓ View all the details of the Attendance
- ✓ Listing of all Attendance

→ Reports of the project College Management System

- ✓ Report of all Students
- ✓ Report of all Fees
- ✓ Report of all Attendance

Static Pages and other sections:

These static pages will be available in project College Management System

- ✓ Home Page with good UI
- ✓ Home Page will contain an JavaScript animated slider for images banner
- ✓ About us page will be available which will describe about the project
- ✓ Contact us page will be available in the project

Technology Used in the project College Management System:

- ✓ HTML: Page layout has been designed in HTML
- ✓ CSS: CSS has been used for all the designing part
- ✓ JavaScript: All the validation task and animations has been developed by JavaScript
- ✓ JSP: All the end front logic has been written in JSP
- ✓ Java: All the business logic has been written in Java
- ✓ MySQL: MySQL database has been used as database for the project
- ✓ Tomcat: Project will be run over the Tomcat server

Supported Operating System:

We can configure this project on the following operating system.

- ✓ Windows: This project can easily be configured on the Windows operating system. For running this project on a Windows system, you will have to install Tomcat 7, JDK 7, MySQL 5.
- ✓ Linux: We can run this project also on all versions of Linux operating system
- ✓ Mac: We can also easily configure this project on the Mac operating system.

CHAPTER-1

Introduction

1.Introduction

With the popularization in all kinds of school and education management departments at all levels, the information construction of' education management has entered a new stage of development. The campus network based on Internet technology has gradually broken the original closed and independent mode of education management and started to transform to an interconnected and open system [1]. However, the current research on teaching management information systems has begun to take shape, but the actual application is still in its infancy [2]. In addition to the insufficient investment in education funds and the weak basic environment of school informatization, the main reasons arc as follows:

The lack of standards for teaching management information system results in great data redundancy and waste of resources, which makes it difficult to realize resource sharing and system interoperability; or just standardizing data cannot solve the problem of interoperability between different systems. Many original independent teaching management information systems have made outstanding contributions to the scientization of education management. However, due to the mutual closure and independent operation of teaching, finance, personnel, equipment, scientific research, and other single management systems developed in different periods and departments, it not only causes great data redundancy and resource waste but also makes it difficult to realize information sharing through the network [3].

The information quality of educational administrators is relatively backward; in particular, whether the leading cadres at all levels have the awareness of modern information management is the key factor in the development and application of campus network management information system. If school leaders and education administrators do not have the awareness of modern information management and lack of understanding of the role and characteristics of modern information management, they will not pay attention to and care about the development and use of teaching management information system, which will bring about insurmountable obstacles to the development and application of the system [4]. Recently, some schools even have established a relatively perfect teaching management information system, but, due to the defects of some managers in consciousness,

ability, and other aspects, as well as the role of conservative factors in the work process, the teaching management information system cannot be fully used and developed.

Information security management refers to the establishment of information security policies and objectives in the overall or specific scope. The organization is required to establish an information security management system by determining the scope of the information security management system, formulating the information security policy, clarifying the management responsibilities, and choosing control objectives and control rethought ds based on risk assessment. To get a truly comprehensive cloud computing service, security is a priority. At the same time, how to better establish the enterprise's own information security management has become the focus of the industry. However, there are some problems in the above methods, such as poor data recall and long query time.

Therefore, this paper proposes a Web-based university education and teaching management information system. It is a comprehensive teaching management software system which takes teaching management as the core and integrates educational administration management, student status management, teacher student management, school production management, book management, and management. Through the network teaching management information system, education administrators, teachers, and students can query, analyse, and process the information in the process of education anytime and anywhere, so as to accurately and timely reflect the current state of the school 's work, use the past data to predict the future, and assist the schoo1's functional departments and principles to manage the school from the overall situation. The network teaching management information system also has a clear meaning of the times, which can meet the requirements of the change of work style in the information age and realize the requirements of information, schematization, and modernization of school management to meet the needs of assisting school administrators to make correct decisions.

Our contribution is threefold:

(1) The traditional university education and teaching management information system has the problems of low information recall, po or information precision, and long query time. Therefore, this paper designs a university education and teaching management information system based on Web.

(2) Through the analysis of the requirements of the higher education and teaching management information system, the design principle of the system is determined, and the structure design of the higher education and teaching management information system is realized.

(3) Experimental results show that this method can effectively improve the low information recall and poor information precision and shorten the query time.

1.1 NEED FOR STUDY

Today in colleges student details are entered manually. The student details in separate records are tedious task. Referring to all these records and updating is needed. There is a chance for more manual errors.

Problems in existing system:

It was limited to a single system.

- It was less user-friendly.
- It have a lots of manual work (Manual system do cs not mean that we are working with pen and paper, it also include working on spread sheets and other simple software's
- It requires more no of employees need to work.
- It was fine consuming process.
- The present system was very less secure
- It is unable to generate different kinds of report.

1.2 SCOPE OF THE STUDY

- Access/ Search information.
- Login in to the system through the first page.
- Change the password
- View/change details.
- An admin privilege who can read as well as remove any uploads

CHAPTER-2

LITERATURE REVIEW

2.LITERATURE REVIEW

Until recently the place of the College Management System (CMS). Now-a-days, education is playing very significant role in the society. Day-by-day, the percentage of illiterates are decreasing and the percentage of literates is increasing. Education will change the society in all the aspects and everyone wants to study higher professional degrees.

Admissions are increasing day by day so there by. Ratio of establishment new colleges and schools are also increasing. But the actual challenge is starting from now. Most of the schools and colleges are maintain student information in records. When the number of records increased, it is difficult to maintain the information of each student in the old manual system. Maintaining the records manually leads to error prone and required more man power and it consumes more time for processing the record

EXISTING SYSTEM

In the existing system of the project, we had just resulted that is usage through manually, here student can check through the computers from a personal computer located at a particular point of place.

2.1 DISADVANTAGES OF EXISTING SYSTEM

1 More time
2 More money
2 More manpower
3 More work
4 Less work results
5 No accuracy

2.2 METHODOLOGY

1. Apache tomcat is used as a web server to host the application.
2. All the environment variables are set.
3. The application is pasted in the webapps folder.
4. Web server is started now.
5. Application is run using the web browser by typing http://localhost/cis
6. Web.xml file is used to control the flow and user actions.

PROPOSED SYSTEM

We resolve all the disadvantages of existing system in our proposed system, CMS. Old students can share their experience, ideas, motivations to their juniors in the college. They can provide information regarding higher studies. RESULT Automation which yields aggregate percentage up to that instant. Time will be saved, no chance of getting error.

2.3 Main Modules Of system :

1. W Administrator module.
2. Staff Mainer (teacher, etc.) Module.
3. Student Module.
4. W Login Module.
5. Registration Module
6. Notice/Upload Module.

Present state :

- No System present at all.
- Time consumed in accessing the records of the students.
- W Manual maintaining student records.
- M Time consumed for registering students and Staff members.
- Manually providing hard copy for any notice or important Information.

After implementation of project :

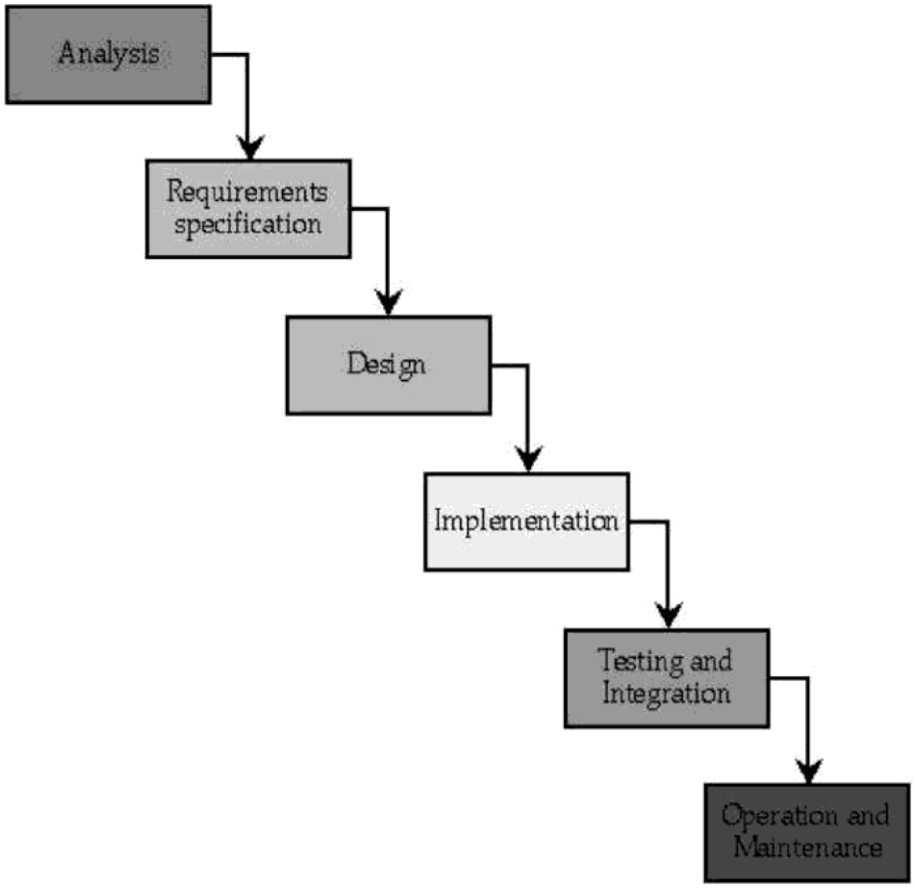
- Collage management system will be created.
- Easy to use GUI.
- Registration of Students and Staff members can be done easily and remotely.
- W Records can be maintained in a more efficient and organized manner.
- Easy to maintain database and secured storage.
- B widen of Administration decreases by a lot.

CHAPTER 3

DETAILED LIFE CYCLE OF PROJECT

DETAILED LIFE CYCLE OF PROJECT

We have used Waterfall Model as Software Engineering life Cycle Process. It is the simplest; oldest and most widely used process model for software development. This model acquires its name from the fact that classic software life cycle is represented as a sequence of descending steps.



3.1 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

hardware Required :

- Processor Dual core or higher | | lghr clock speed or higher
- RAM 2gb or higher Storage 80gb or higher
- Display Adapter 256mb or high

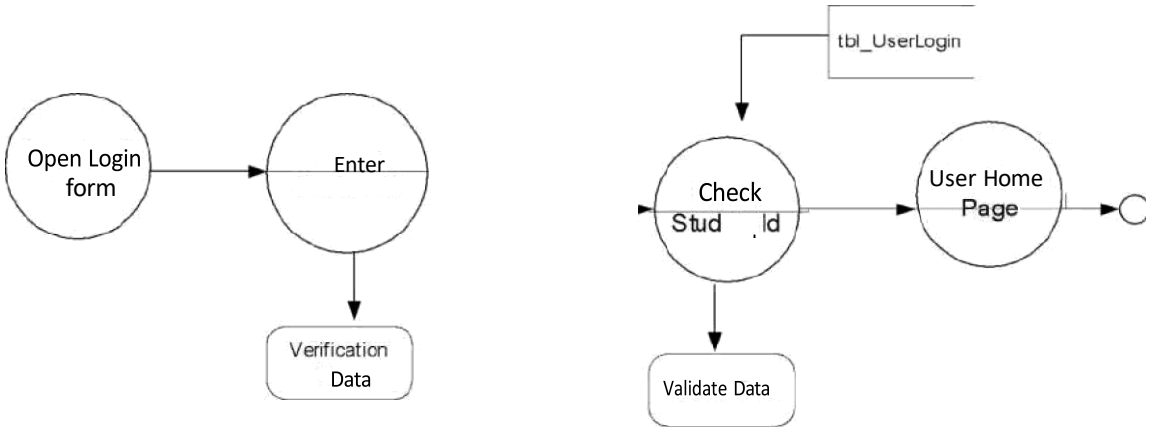
Software Required:

- OS Windows, Unix, Linux
- Web Browser Chrome, Firefox, any Language PHP, HTML, CSS, JAVASCRIPT
- Database MySQL, SQL Server

Tools Used:

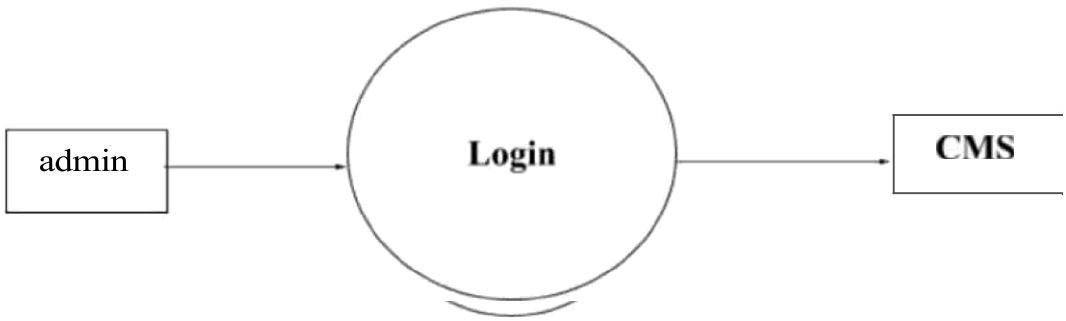
- XAMPP server
- W PHP v5 or higher
- Notepad (Brackets recommended)

3.2 DATA FLOW DIAGRAM

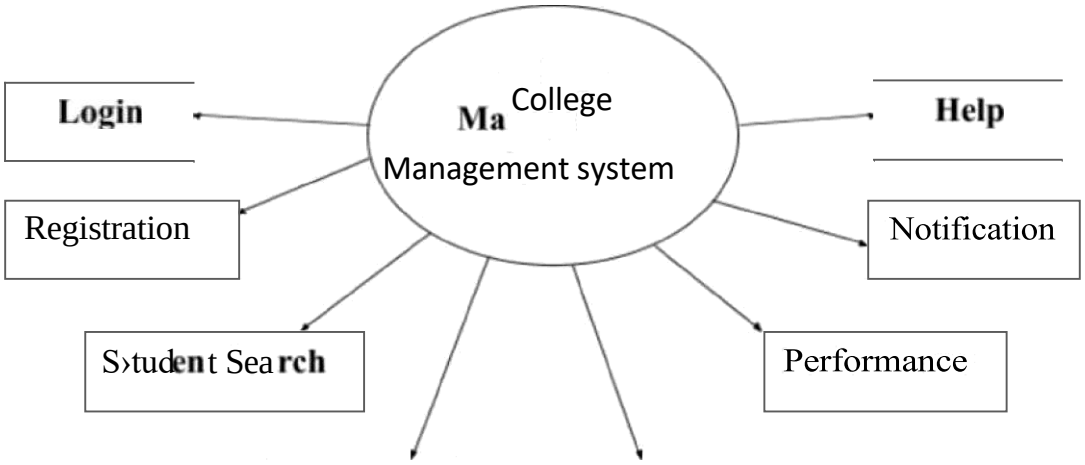


Context level Diagrams:

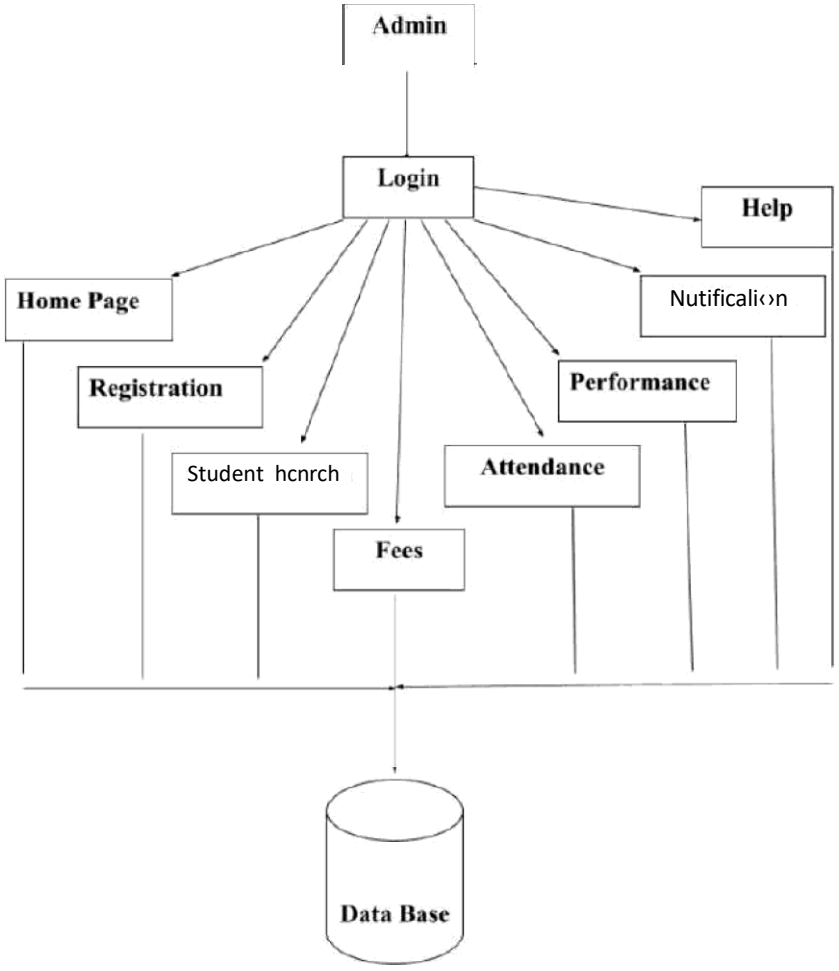
Level 0:



Level 1 :



Level 2:



3.3 Study of the System:

3.3.1 Graphical user interface

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

Administrative user interface

1. The operational or generic user interface
2. The administrative user interface concentrates on the consistent
3. That is practically, part or the organizational activities and which needs proper
4. Authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Date update along with the extensive data search capabilities.
5. The operational or generic user interface helps the users upon the system in transactions through the existing data and required services.

3.3.2 Number of Modules

The system after careful analysis has been identified to be presented with the following modules:

The modules involved are:

1. College information: Through this service one can access the complete information about the college campus such as courses available, admission procedure, placements, college events, achievements etc.
2. Student tracking: Any company or any organization that want to check the summary abo at the student of the college, so that they will be able to choose the particular students for their campus placement And for that purpose they will be given a particular link through which they can access the information required.
3. Student attendance status: It gives the attendance status of students.
4. Faculty will update the attendance periodically and can be seen by

students and parents.

5. Student's performance in exams: This facility provides the performance of the student in each exam which is conducted by university or college such a's midterm performance, Marks obtained by students in exams.
6. Exam Notification: This facility notifies students and parents about examination schedule.
7. Events: it will give information about different events that will be conducted
8. by college time to time. Information about these events will be updated by administrator.
9. Information about staff: It will help in maintaining complete information about college faculty members such as their department, cadre, date of joining, salary, etc. Administrator will register new faculties and remove their account when they leave the college.

3.3.3 FORM DESIGNING

- **Login Form:-**

This is login form of the Software, it ask for the id and password of the user, access is given to the authorized persons only. Types of logins are there Admin only.

- **MDI form:**

This acts as the home screen for this software. Through this MDI form

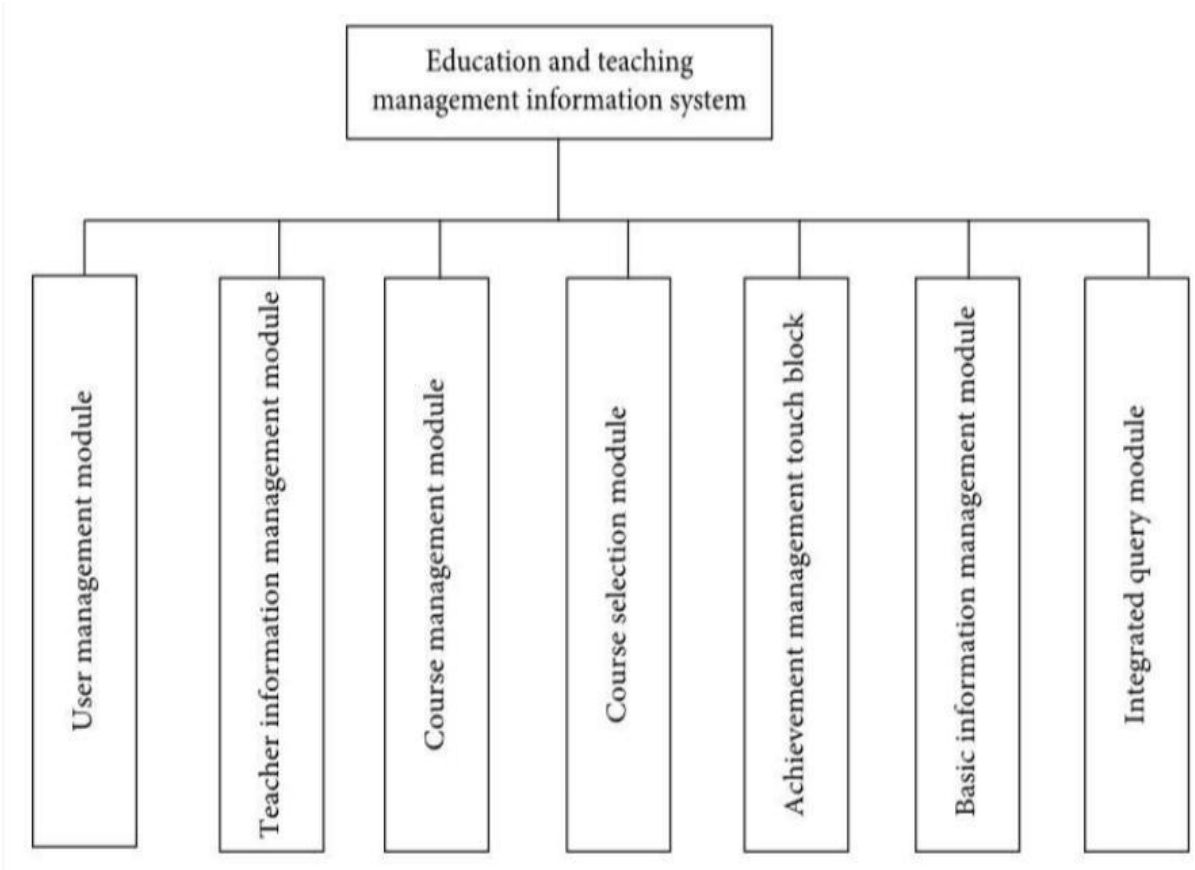
we can access its child forms. 'this form gives the user further 5 options:-

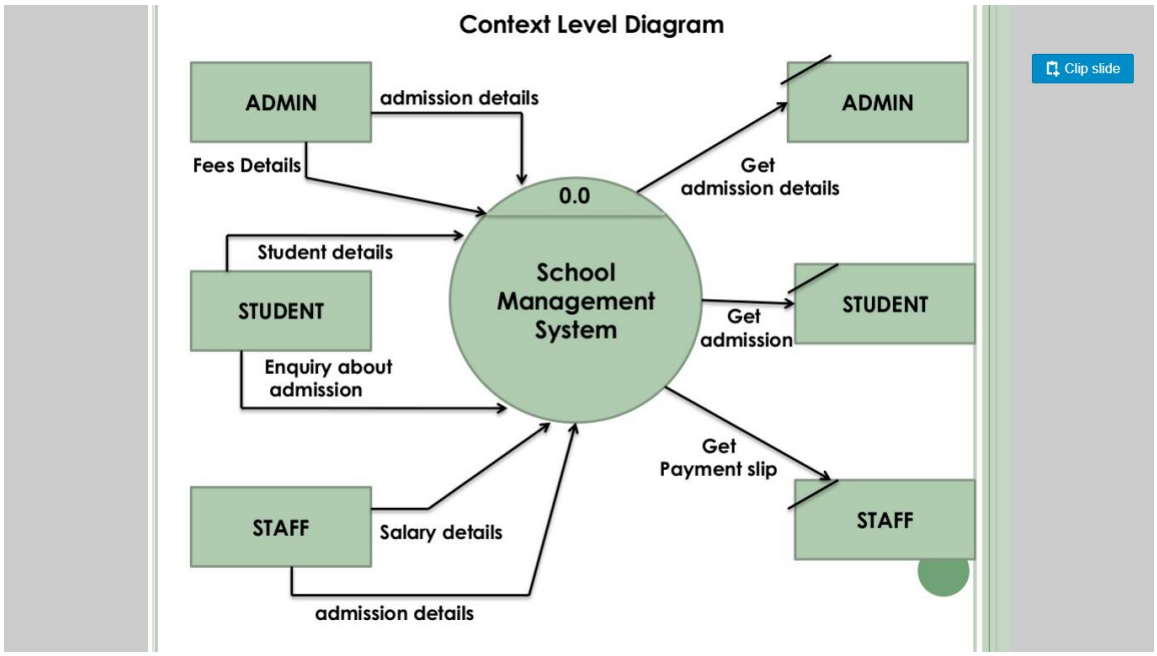
- Login Page it contains LOGIN, LOG OUT, and EXIT option to exit from Software.
- Registration: This module is designed for registration

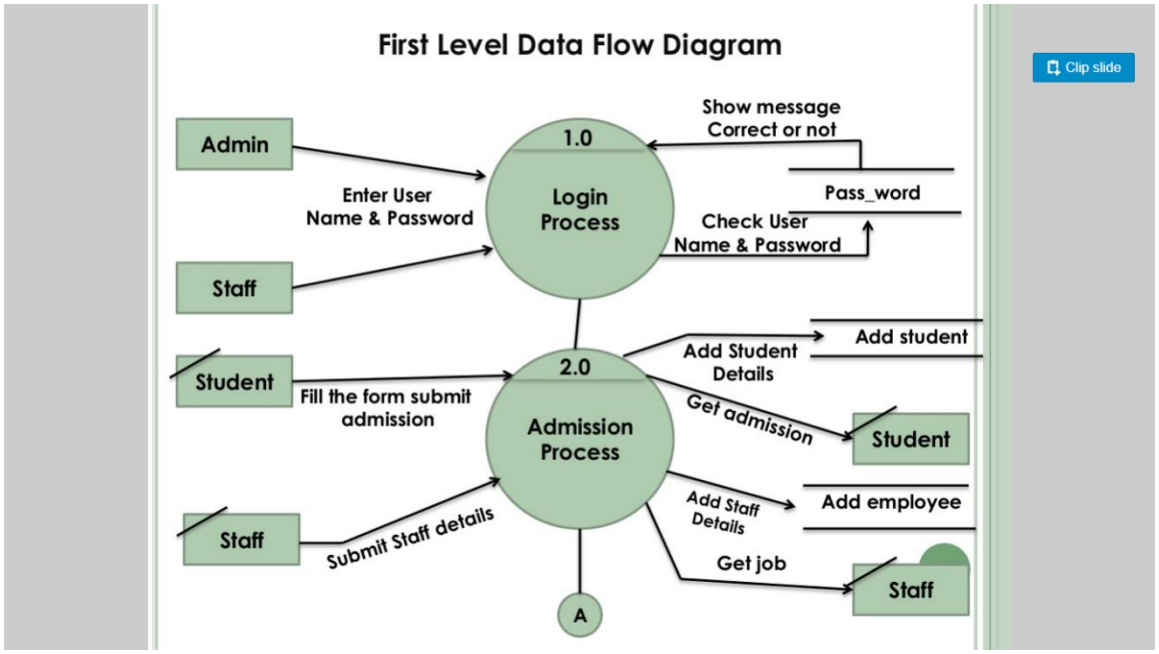
- **Form list and details**
    1. **Registration Entry Form**: registration entry form can be used to enter the new Entry of student.
    2. **Student Search Form**: student search form can be used by admin, for Searching particular student related entering branches and year.
    3. **Notification Form**: notification form can be used to add notification related to the exams and schedule of the college. Full authority is given to the admin.
    4. **Fees Form:** this can be generated the fees report of the students.

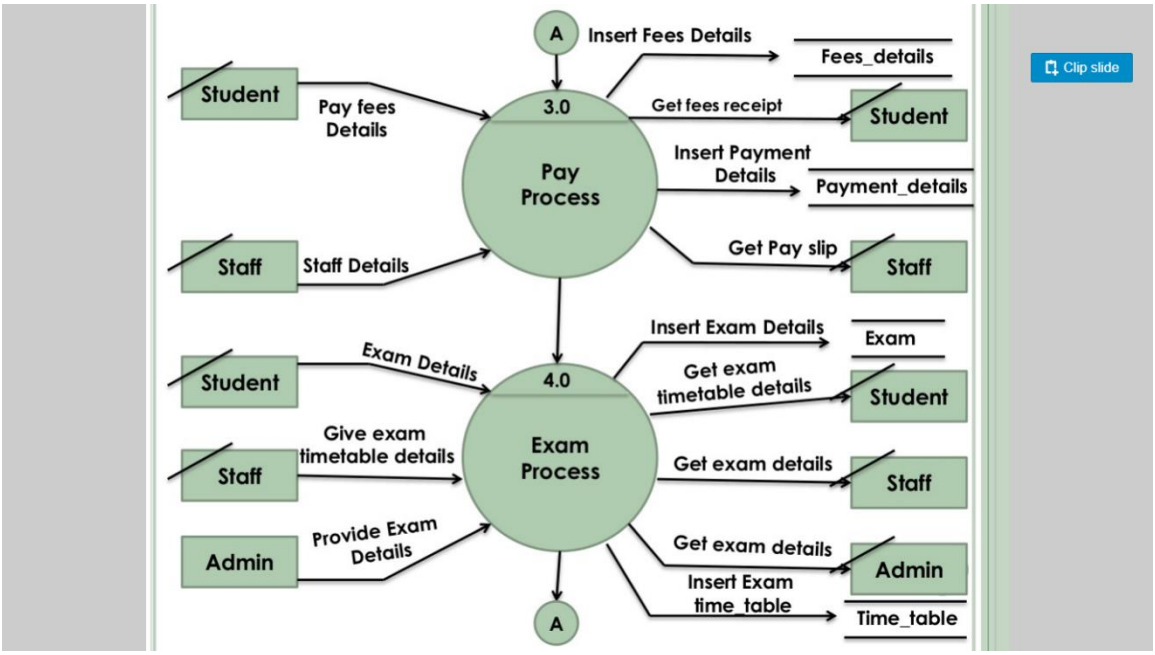
3.4 Flow chart

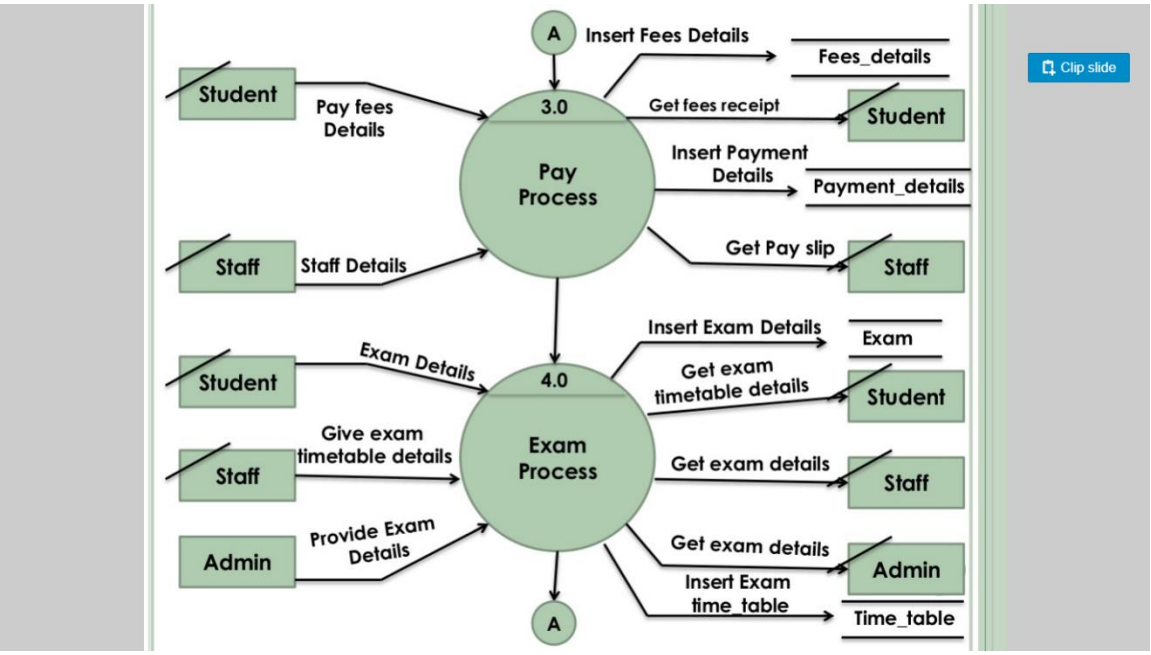
Figure 1:




Context Level Diagram
Clip slide
ADMIN
admission details
ADMIN
Fees Details
Get
admission details
0.0
Student details
School
Management
System
STUDENT
STUDENT
Get
admission
Enquiry about
admission
Get
Payment slip
STAFF
STAFF
Salary details
admission details

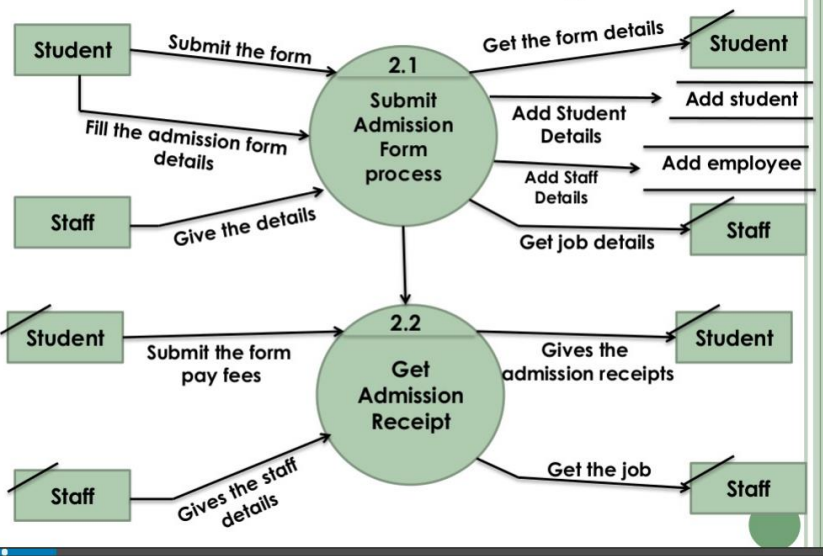

First Level Data Flow Diagram
Clip slide
Admin
1.0
Show message
Correct or not
Login
Process
Pass_word
Enter User
Name & Password
Check User
Name & Password
Staff
Add student
Add Student
Details
Student
2.0
Fill the form submit
admission
Get admission
Admission
Process
Student
Add employee
Add Staff
Details
Staff
Submit Staff details
Get job
Staff
A


A
Insert Fees Details
Fees_details
Student
Pay fees Details
3.0
Pay Process
Get fees receipt
Student
Insert Payment Details
Payment_details
Get Pay slip
Staff
Staff Details
Staff
Insert Exam Details
Exam
Student
Exam Details
4.0
Get exam timetable details
Student
Give exam timetable details
Staff
Exam Process
Get exam details
Staff
Admin
Provide Exam Details
Get exam details
Admin
Insert Exam time_table
Time_table
A
Clip slide


A
Insert Fees Details
Fees_details
Student
Pay fees Details
3.0
Pay Process
Get fees receipt
Student
Insert Payment Details
Payment_details
Get Pay slip
Staff
Staff Details
Staff
Insert Exam Details
Exam
Student
Exam Details
4.0
Get exam timetable details
Student
Give exam timetable details
Staff
Exam Process
Get exam details
Staff
Admin
Provide Exam Details
Get exam details
Admin
Insert Exam time_table
Time_table
A
Clip slide

School Management System

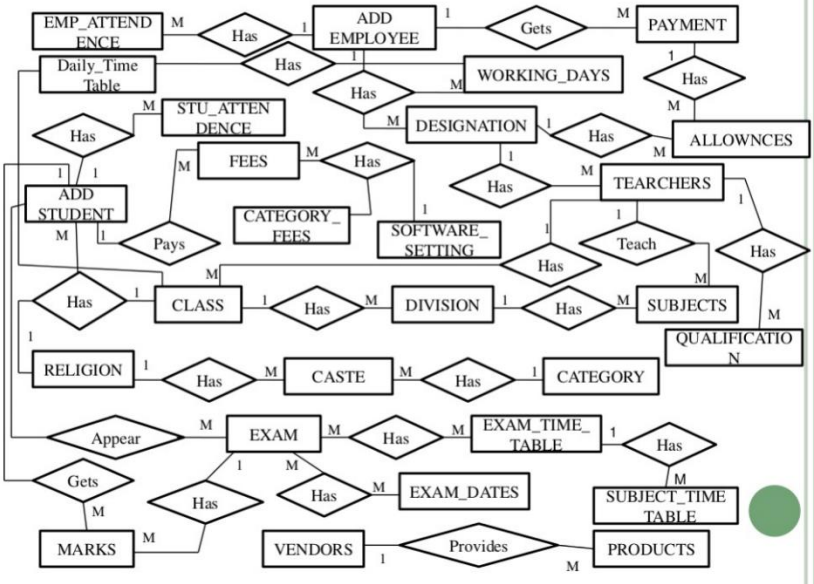
Second Level Data Flow Diagram



Clip slide

15 of 78

Entity Relationship Diagram



Clip slide

CHAPTER-4

IMPLEMENTATION

IMPLEMENTATION

This project designs a Web-based university educator and teaching management information system. By analyzing the requirements of university education and teaching management information system, the design principle of the system is determined, the structure design of university education and teaching management information system is realized, and the menagerie process of the system is determined. On the basis of the above, by calculating the complexity of university education and teaching management information, the priority query information is determined to effectively improve the processing effect of the system. Finally, the relational database model is designed to realize the design of universityeducation and teaching management information system.

The results are as follows:

(1) When the amount of teaching information data is 1 00 GB, the precision rate of this method is 94%, which shoes that this method has better query accuracy for university education and teaching management information.

(2) When the data of teaching information is 200 GB, the recall rate of the information of the teaching management in colleges and universities is 96%, which shows that the method has a good effect on the information retrieval of the teaching management in colleges and universities,

(3) When the amount of teaching information n data is 300 GB, the query time of this method is only 2.3 s. It shows that the query efficiency of this method is better.

4.1 System Database Core Design

The database of university education and teaching management information system is divided into two parts: the general control center database and the sub control center database. The database of the main control center is the co re part of the remote network teaching system. It is particularly important to support the operation of the whole system. In order to efficiently integrate all the information and resource data of the whole system, the capacity, stability,

and operation efficiency of the database must be fully considered in the construction of the database of the general control center. In order to make the main control center database able to effectively meet the daily information reading, hac1«ip, and other needs, one way is to select the appropriate hardware platform to provide the necessary hardware support for the system, and another is to select an efficient and stable database management system to realize the complete management of data. Database construction is the core part of the network distance education system, which plays a vital role.

From the functional point of view, irritatingly covers system use case analysis, data management, financial management, identify authentication, security management, and so on, As the data storage point of the whole system, the central host system of the system construction manages and processes the data storage and transmission of the whole system. The system design adopts a typical three-flier system architecture. In its architecture design, it covers two functional servers: central database server and teaching management platform. Among them, teaching management platform includes data management server, security server, identity authentication server, and other related equipment servers.

4.2 Relational Database Model

The logic structure of the system is divided into two types: conceptual mode and external mode. According to these two modes, the database design of the system can be carried out, and the object patterns can be transformed into fictitious database tables to get the table pattern (conceptual model).

4.3 work / coding

4.3.1 NDEX AND HOMEPAGE CODING

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>College Management System</title>

<link rel="stylesheet" type="text/css" href="css/i.css">

</head> <body>

<div class="menu">

<div id="home"><a href="home.php" target="frame1">Home</a></div> <div
id="login"><a href="login.php" target="frame1">Login</a></div>

<div id="sign_up"><a href=registration.php

target="frame1">SignUp</a></div>

<div id="contact"><a href=contact.php target="frame1">Contact

Us</a></div>

<div id="about_us"><a href=# target="frame1">About Us</a></div>
</div>

<div class="box"> <div class="frame">

<iframe id="frame1" name="frame1" width="100%" height="2048px"
```

```
frameborder="0px"></iframe>

</div>

</div>

<div class="footer">

<p align="center"><font color="red">&copy; 1999 Maharaja Agrasen Institute of Technology, New Delhi. All Rights Reserved.</font></p>

</div> </body>

</html>
```

4.3.2 STUDENT REGISTRATION PORTAL

```
<?php

require_once("config.php");

?>

<!DOCTYPE html>

<html>

<head> <meta charset="utf-8">

<meta name="vierport" content="width=device-width, initial-scale=1.0">
```

```
<link rel="stylesheet" type="text/css" href="css/registration.css">
<title>Registration Form</title>

</head> <body>

<div class="header">

<?php

if(isset($_POST["submit"}}}

$enroll=$_POST["enroll"]; $name= $_POST["name"]:

Semail=$_POST["email"]:

Sphone=$_POST["phone"];

Sdob-S_POST["dob"]; $branch=$_POST["branch"];

$sql = "INSERT INTO users (enroll, name, email, phone, dob, branch)

VALUES(??????)":

$stmtinsert = $db->prepare($sql);

$result = $stmtinsert->execute([Senroll, Sname, Semail, Sphone, $dob.
$branch]);
```

```
if($result){

echo 'Successfully saved.';}

else{ echo 'There were erros while saving the data."; }

} ?>

</div>

<div class="header">

<form action="registration.php" method="post"> <div class="container">

<div class="row">

<div class="data">

<h1 id="h1">User Registration.</h1><hr>

<h2 id="h2">Fill Details,</h2><hr>

<label for="enroll">Enrollment ID:</label>

<input type="text" name="enroll" id="enroll" placeholder="enter your
enrollment id" required><br><br>

<label for="name">Full Name:</label>
```

<input type="text" name="name" id="name" placeholder="enter

your name" required>

<label for="email">Email:</label>

<input type="email" name="email" id="email" placeholder="Email id/Username" required>

<label for="phone">Phone no.</label> <input type="text" name="phone" id="phone" placeholder="0123456789" required>

<label for="dob">Date Of Birth:</label>

<input type="date" name="dob" id="dob" required>

<label for="branch">Branch</label>

<input type="text" name="branch" id="branch" placeholder="enter your branch" required>

value="submit">

<input class="submit" type="submit" name="submit"

 Dr.V.S.Krishna govt degree college portal

 </form>

© Dr.V.S.Krishna govt degree college(A), madhhilapalem,vishkapatnam.

</body>

</html>

4.3.3 Contact us page;

<!DOCTYPE html>

<html>
<head>

<meta charset="utf-8">

```
<link rel="stylesheet" type="text/css" href="css/contact.css">

<title>Contact Us</title>

</head>

<body align="center">

<div class="header">

<div class="left">

<h1>Dr.v.s. krishna govt degree college(A)</h1><hr> <table class="table">

<tr> <th>Postal Address:</th>

<td>Dr.v.s. krishna govt degree college(A)<br>   krishna college road,
Maddilapalem.

<br>Visakhapatnam-530022<br>INDIA.</td>

</tr><hr> <u>

<th>Phone:</th>

<td>+917894578949</td
</tr>
```

<tr>

<th>Fax:</th>

<td>911127582095</td>

</tr>

<tr>

<th>Email Address:</th> <td>infinityconnectsmedia@gmail.com</td>

</tr>

</table>
<hr>

<h1>Director Office.</h1> <table class="table">

<tr>

<th>Address:</th>

<td>Prof.Dr.K.Ravibabu

Dr.v.s. krishna govt degree college(A).</td>

</tr>

<tr>

<th>Phone No. </th> <td>+918448186931</td>

</tr>

<tr>

<th>Email address:</th>

<td>infinityconnectsmedia@gmail.com</td>

</tr>

</table>
<hr>

<h1>Registrar Office.</h1>

<table class="table">

<tr>

<th>Address:</th>

<td>Registrar
Administrative Block, g-12,
Dr.v.s. krishna govt degree college(A.</td> </tr>

<tr>

<th>Phone No. :</th>

<td>+918448186942</td>

</tr> <tr>

<th>Email address:</th>

<td>registrar@mait.ac.in</td>

</tr>

</table>
<hr>

<h1>Accounts Department.</h1>

<table class="table">

<tr>

<th>Address:</th>

<td>Accounts Department.
Administrative Block, A-01,

Dr.v.s. krishna govt degree college(A.</td>

</tr> <tr>

<th>Phone No.:</th> <td>+919999985214</td>

</tr> <tr>

<th>Email address:</th>

<td>account@mait.ac.in</td>

</tr>

</table>
<hr>

<h1>Department of Training & Placement</h1>

<table class="table">

<tr>

<th>Brig. S.K.Kakar</th> <td>Head of the Department
 Dr.v.s. krishna govt degree college(A</td>

</tr>

<tr>

<th>Phone No. :</th> <td>+918448186933</td>

</tr>

<tr>

<th>Fax:</th>

<td>(011)27582095</td>

</tr> <tr>

<th>Email address:</th>

<td>placement@mait.ac.in</td>

</tr>

4.3.4 Building a Sign-up system

The system. Our first step is to create a HTML registration form. The form is pretty In this part, We will create a signup system that allows users to create a new account to simple to create. It only asks for a name, email, password, and confirm password. Email addresses will be unique for every user. Multiple accounts for the same email address are not allowed. It will show an error message to the users who try to create multiple accounts with the same email address.

Step 1: Creating Registration Form in HTML

We will create a PHP file named register with the following code in it. This is a simple HTML form with some basic validation. If you are not familiar with HTML then you can get it from many online sites who give ready-made html5 login form templates.

HTML

```
<!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8"> <title>Sign Up</title>

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css
">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-12"> <h2>Register</h2>

<p>Please fill this form to create an account.</p>

<form action="""" method="post">

<div class="form-group"> <label>Full Name</label>

<input type="text" name="name" class="form-control" required>
</div>

<div class="form-group">

<label>Email Address</label>

<input type="email" name="email" class="form-control" required />
```

```
</div>

<div class="form-group">

<label>Password</label>

<input type="password" name="password" class="form-control"

required>

</div>

<div class="form-group">

<label>Confirm Password</label>

<input type="password" name="confirm_password"

class="form-control"

required> </div>

<div class="form-group"> <input type="submit" name="submit" class="btn btn-primary"

value="Submit">

</div>

<p>Already have an account? <a href="login.php">Login

here</a>.</p>

</form>

</div>
```

</body>

</html>

All the input fields are required by adding the "required" attribute which is the default

HTML attribute. The use of type="email" will validate the email address provided by users and gives an error if the email address is not valid. For the registration form, we have used bootstrap for rapid development. If you want to save your time on HTML code you can always use some free html5 templates for your project.

Step 2: Creating the MySQL Database Table

You will need to create a new database with any suitable name you want. After that please execute the below SQL query to create the user's table inside your newly

created MySQL database.

MySQL

CREATE TABLE `users` (

`id` int(11) unsigned NOT NULL AUTO_INCREMENT,

'name' varchar(75) NOT NULL,

'password' varchar(255) NOT NULL, 'email' varchar(100) NOT NULL,

PRIMARY KEY ("id"). UNIQUE KEY 'email' ('email')

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO INCREMENT=1;

Step 3: Creating Database Configuration File

Now, we have created the user's table. Let's create a new PHP file named config.php to connect with the MySQL database, Paste the following code in the config.php file and change the database name to whatever you choose while creating the database.

PHP

Q H

<?php

define('DBSERVER', 'localhost'); // Database server
define('DBUSERNAME', 'root'); // Database username

define('DBPASSWORD'', ''); // Database password define('DBNAME', 'demo'); // Database name

/* connect to MySQL database */ $db = mysqli_connect(DBSERVER, DBUSERNAME, DEPASSWORD, DBNAME).

// Check db connection if($db === false){

die("Error: connection error.". mysqli_connect_error());

?>

Step 4: Creating a Session File

Let's create a file named session. In this file, we will start the session and check if a user is already logged in, if yes then we will redirect the user to welcome file.
PHP

```
<?php

// Start the session

session_start():

// if the user is already logged in then redirect user to welcome page if (isset($_SESSION["userid"]) && $_SESSION["userid"] === true) {

header("location: welcome.php");

exit;

}
```

Step 5: Create Registration Form in PHP

Finally, it's time to create a PHP code that allows users to register their accounts into the system. This PHP code will alert users with an error if any user is already
registered with the same email address.

Replace the following code in the register.php file.

PHP

```
<?php

require_once "config.php":

require_once "session.php"; if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {

$fullname = trim($_POST['name']);

$email = trim($_POST['email']);
```

$password = trim($_POST['password']]: Sconfirm password = trim($_POST["confirm_password"]);

Spassword_hash=password_hash($password, PASSWORD_BCRYPT); if($query = $db->prepare("SELECT FROM users WHERE email = ?")) {

Serror = '';

// Bind parameters (s = string, = int, b = blob, etc), in our case the username a string so we use "s"

$query->bind_param('s', $email): $query->execute();

// Store the result so we can check if the account exists in the database. $query->store_result():
if ($query->num_rows > 0) { $error = ' The email address is already registered! ';

} else {

// Validate password

if (strlen($password) <6) { $error = ' Password must have atleast 6 characters. ';

}
// Validate confirm password

if (empty($confirm_password)) {

Serror.= ' Please enter confirm password. ';

} else {

if (empty(Serror) && ($password != $confirm_password)) {

$error = ' Password did not match. ';

} if (empty($error)) {

SinsertQuery = $db->prepare("INSERT INTO users (name, email.

password) VALUES (?,?,?);");

SinsertQuery->bind_param("sss", $fullname, $email, $password_hash); $result = $insertQuery->execute();

if ($result) { $error = ' Your registration was successfull ';

} else {

$error = ' Something went wrong! ";

}

$query->close():

SinsertQuery->close(); // Close DB connection mysqli_close($db):

?>
<label>Email Address</label>

<input type="email" name="email" class="form-control" required />

<label>Password</label>

<input type="password" name="password" class="form-control"

required>

<label>Confirm Password</label>

<input type="password" name="confirm_password" class="form-control" required>

<input type="submit" name="submit" class="btn btn-primary" value="Submit">

Already have an account? Login here.

</form>

</body>

</html>

Once user click on submit button it will check if $_SERVER["REQUEST_METHOD"] ==

"POST" and $_POST['submit'] a variable has been set. For security concerns, we always suggest not to store the password as plain text in the database. We have used password hash () function which creates a new password hash using a strong one-way hashing algorithm.

The above PHP script will validate that no user is registered with the same email address and also validate the password. After validation is confirmed, we store the user-provided information in the users' table and alert the user that registration was successful.

4.3.5 Building a Login System

In this part, we will create a login form to allow users to access the restricted area of the system. In our case, the restricted area is a welcome page which we will cover in the next part.

Step 1: Creating a Login Form in HTML

Below is the Login Form in HTML. Paste it in a file named login.php

HTML

<!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8"> <title>Login</title>

<link rel="stylesheet"

</head>

<body>

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css
">

```
<div class="container"> <div class="row">


<div class="col-md-12">

<h2>Login</h2>

<p>Please fill in your email and password. </p>

<form action="" method="post">

<div class="form-group">

<label>Email Address</label> <input type="email" name="email" class="form-control" required />

</div>

<div class="form-group"> <label>Password</label>
<input type="password" name="password" class="form-control"

required>

</div>

<div class="form-group">

<input type="submit" name="submit" class="btn btn-primary"

value="Submit">

</div> <p>Don't have an account? <a href="register.php">Register

here</a>.</p>

</form>
```

```
</div>

</div>

</div>

</body>

 </html>
```

Step 2: Creating a Login System in PHP

After creating the login form in HTML, we will write a code to validate login credentials, On form submit we will check that the email and password are filled. If they filled then we will execute a SELECT query to find the record in a database on the basis of email and password. If any record found, then we will store the "userID" in session and the user is redirected to the welcome.php file, otherwise, the user is alerted with an error message.

Let's replace the following code in the login.php file.

PHP

```
<?php

require once "config.php":

require_once "session.php": $error = '';

if ($_SERVER["REQUEST_METHOD"]=="POST" && isset($_POST['submit']}} {

$email = trim($_POST['email']): Spassword= trim($_POST['password']):

// validate if email is empty
```

```
if (empty($email)) {

Serror.= '<p class="error">Please enter email.</p>';

// validate if password is empty if (empty($password)) {

$error = '<p class="error">Please enter your password. </p>';
if (empty($error)) {

if($query = $db->prepare("SELECT FROM users WHERE email = ?")) {

$query->bind_param('s'. $email);

$query->execute(); $row = $query->fetch():

if ($row) {

if (password_verify($password, Srow['password'])) {

$_SESSION["userid"] = $row["id"]; $_SESSION["user"] = $row:

// Redirect the user to welcome page

header("location: welcome.php");

exit; } else {

$error = '<p class="error">The password is not valid.</p>';

}

} else {

Serror.= '<p class="error">No User exist with that email address.</p>';

}
```

```
} $query->close();

}

// Close connection mysqli_close($db);
<!DOCTYPE html> <html lang="en">

<head>

<meta charset="UTF-8">

<title>Login</title>

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css
">

</head> <body>

<div class="container">

<div class="row">

<div class="col-md-12">

<h2>Login</h2> <p>Please fill in your email and password. </p>

<?php echo Serror; ?>

<form action="" method="post">

<div class="form-group"> <label>Email Address</label>

</div>

<input type="email" name="email" class="form-control" required />
```

```
<div class="form-group"> <label>Password</label>
required>

<input type="password" name="password" class="form-control"

</div>

<div class="form-group">

<input type="submit" name="submit" class="btn btn-primary"
value="Submit"> </div>

<p>Don't have an account? <a href="register.php">Register

here</a></p>

</form>

</div> </div>

</div> </body>

</html>
```

4.3.6 Creating a Welcome Page

Below is the code for the welcome php file. Users will be redirected to this page after a successful login process. We have added some code at the top of the page to check if the user is not logged in, then redirect the user to the login page.

Let's create a welcome. Php file and paste the following code in it.

PHP

```
<?php
```

```
// start the session

session_start();

// Check if the user is not logged in, then redirect the user to login page if (isset($_SESSION["userid") || $_SESSION["userid"] !== true) { header("location: login.php");

exit;

?>

<!DOCTYPE html>

<html lang="en"> <head>

<meta charset="UTF-8">

<title>Welcome <?php echo $_SESSION["name"]; ?></title> <link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css
">
</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-12">

<h1>Hello, <strong><?php echo $_SESSION["name"]; ?></strong> Welcome to
demo site.</h1>
```

```
</div>

<p> <a href="logout.php" class="btn btn-secondary btn-lg active"

role="button" aria-pressed="true">Log Out</a>

</p>

</div>

</div>

</body>

</html>
```

4.3.7 The Logout Script

Finally, Let's create a logout.php file with the following code in it.

PHP

```
<?php

// Start the session

session_start():

// Destroy the session

if (session destroy ()) {
//redirect to the login page

header ("Location: login.php");

exit;
```

Once the user clicks on the Log Out link, the above script, will be called to destroy the session and redirect the user to the login.php file.

4.3.8 Config file which links webpage(frontend) with database(backend):

<?php

Sdb_user = "root": $db_pass=""

$db_name = "cms";

$db = new PDO('mysql:host=localhost;dbname='.$db_name. ;charset=utf8', $db_user. $db_pass); $db->setAttribute(PDO:ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

4.3.9 Login Module:

<!-- PHP Starts Here -->

<?php session_start():

require_once "../connection/connection.php": $message="Email Or Password Does Not Match";

if(isset($_POST["btnlogin"]}}

$username=$_POST["email"]; $password=$_POST["password"];

$query="select * from login where user_id='$username' and Password='Spassword" ";

$result-mysqli_query($con,$query);

if (mysqli_num_rows($result)>0) {

```
while (Srow=mysqli_fetch_array($result)) { if ($row["Role"]=="Admin")

$_SESSION['LoginAdmin']=$row["user_id"];
header('Location: ../admin/admin-index.php');

else if ($row["Role"]=="Teacher" and $row["account"]=="Activate")
$_SESSION['Login Teacher']=$row["user_id"];

header("Location: ../teacher/teacher-index.php"):

else if ($row["Role"]=="Student" and $row["account"]=="Activate")

$_SESSION['LoginStudent']=$row['user_id'];
header('Location: ../student/student-index.php"):

else

header("Location: Login.php");

}

<!doctype html>

<html lang="en">

<head>

<title>Login ICBS</title>

</head>

<body class="login-background">

<?php include(../common/common-header.php') ?>

<div class="login-div mt-3 rounded">
```

 <img src="../Images/icbs_logo.png" alt="" class="align-items-center"

width="100"

height="100">

<h2 class="text-center text-white">LOGIN</h2> <form class="p-1" action="login.php" method="POST">

<label><h6>Enter Your ID/Email:</h6></label>

<input type="text" name="email" placeholder="Enter User ID" class="form-control" required>

<label><h6>Enter Password:</h6></label> <input type="Password" name="password" placeholder="Enter

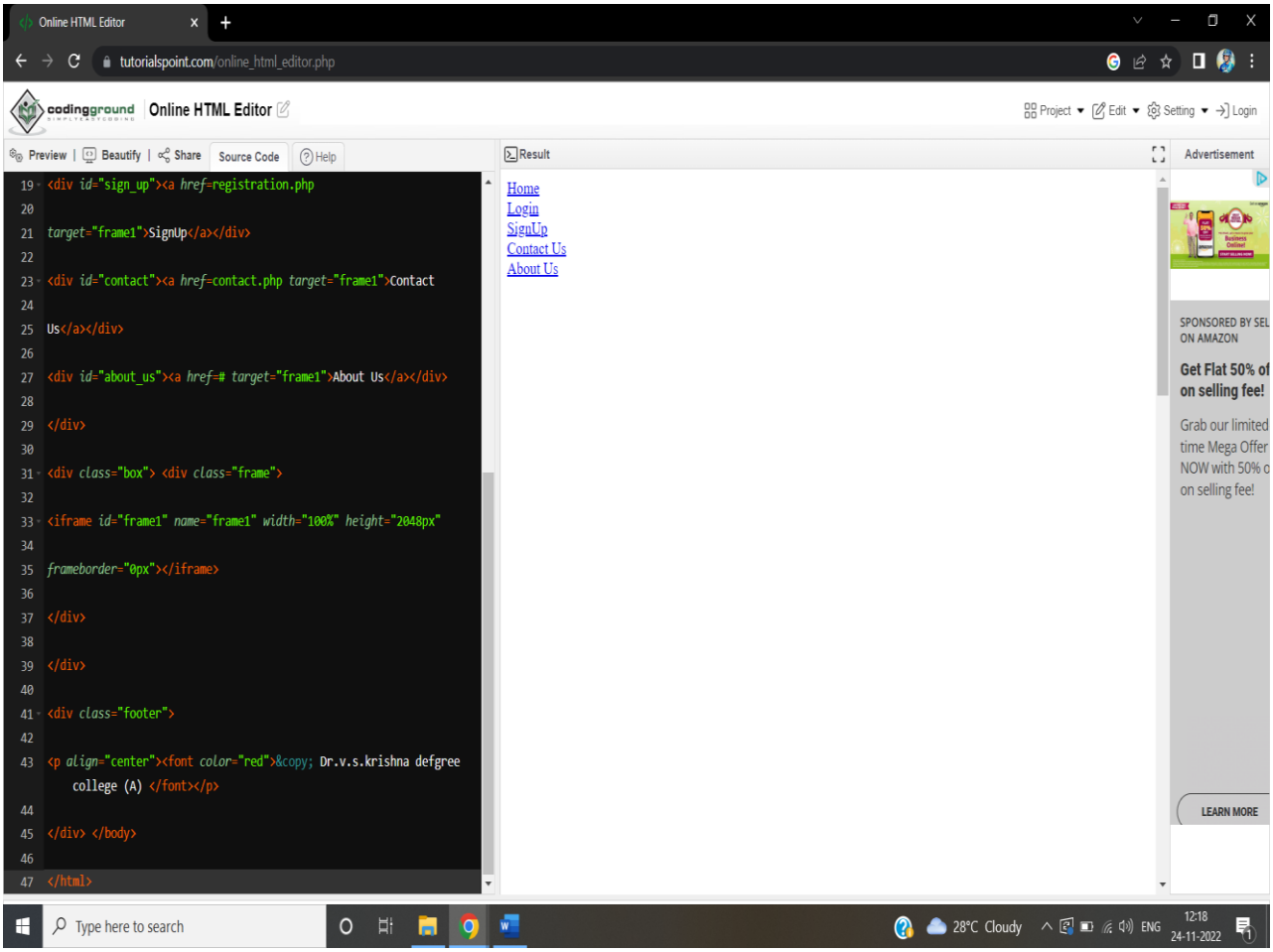
Password" class="form-control border-bottom" required>

<--<?php echo $message; ?> -->

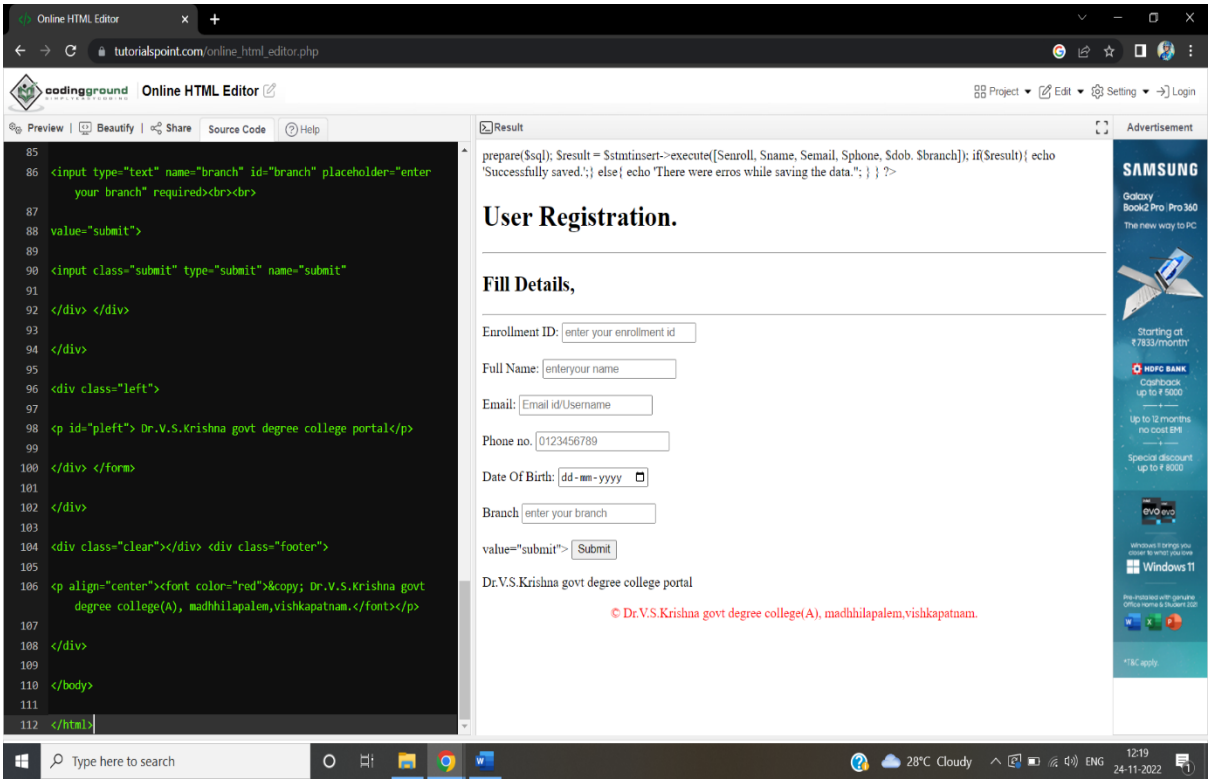
 <input type="submit" name="btnlogin" value="LOGIN" class="btn btn-white pl-5 pr-5">

4.4 OUTPUT SCREENSHOTS;

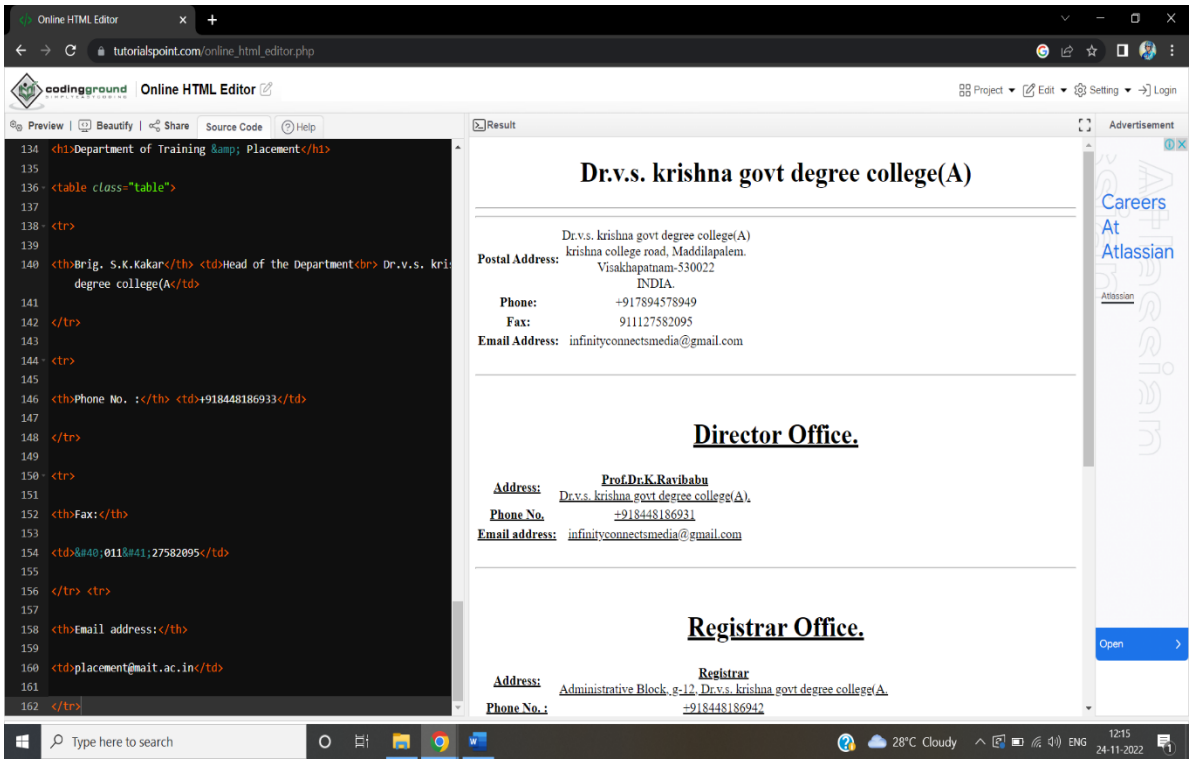
INDEX AND HOMEPAGE OUTPUT;

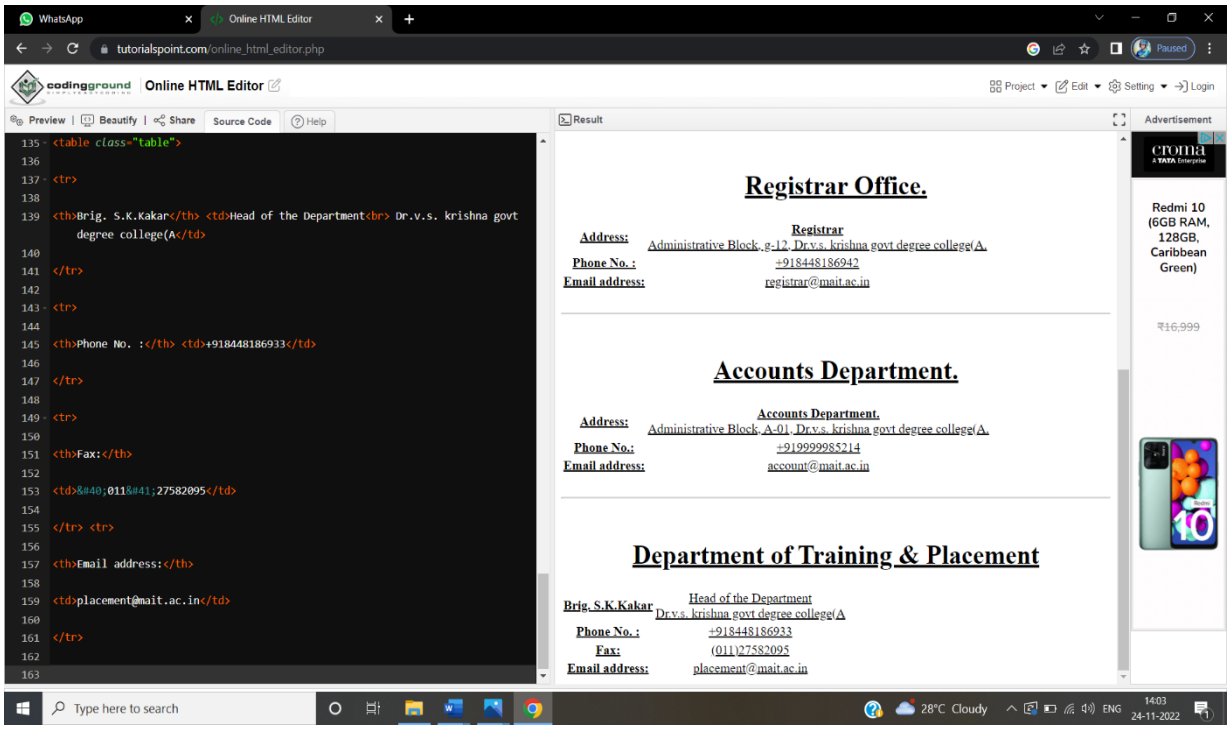


STUDENT REGISTRATION PORTAL

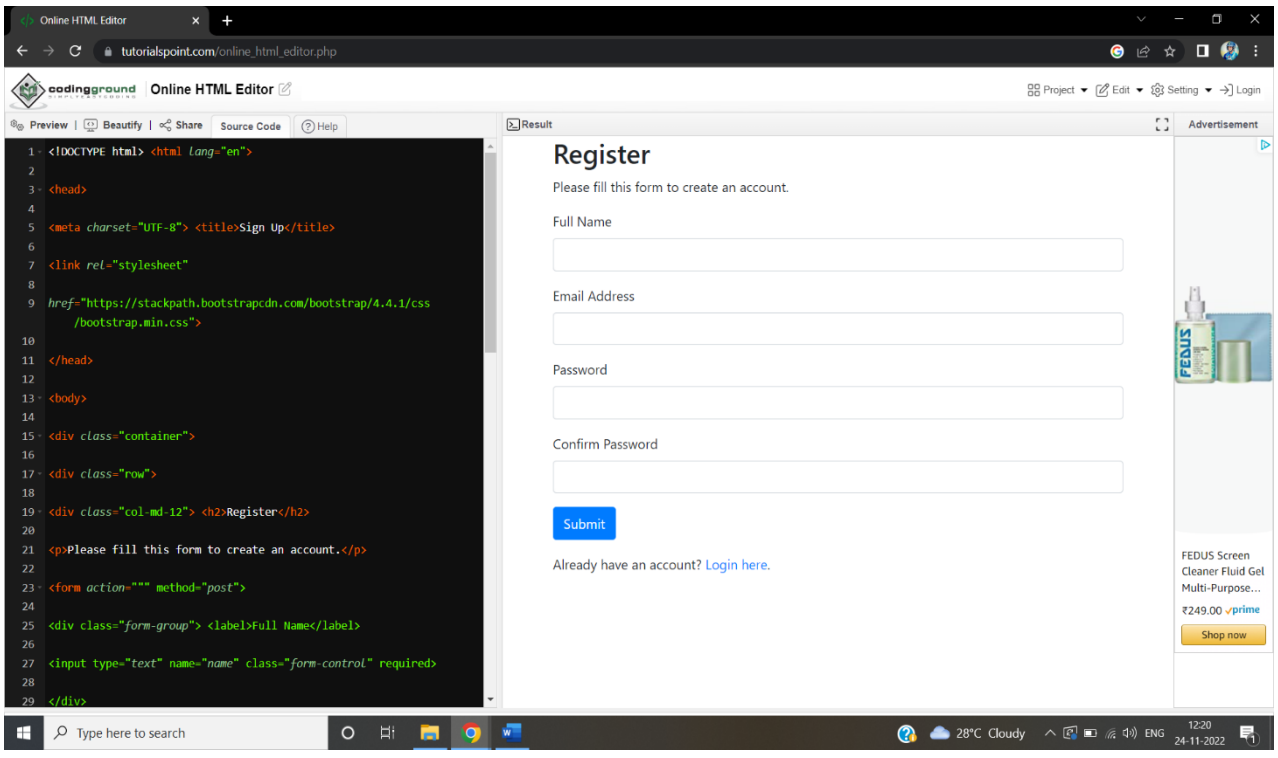


Contact us page



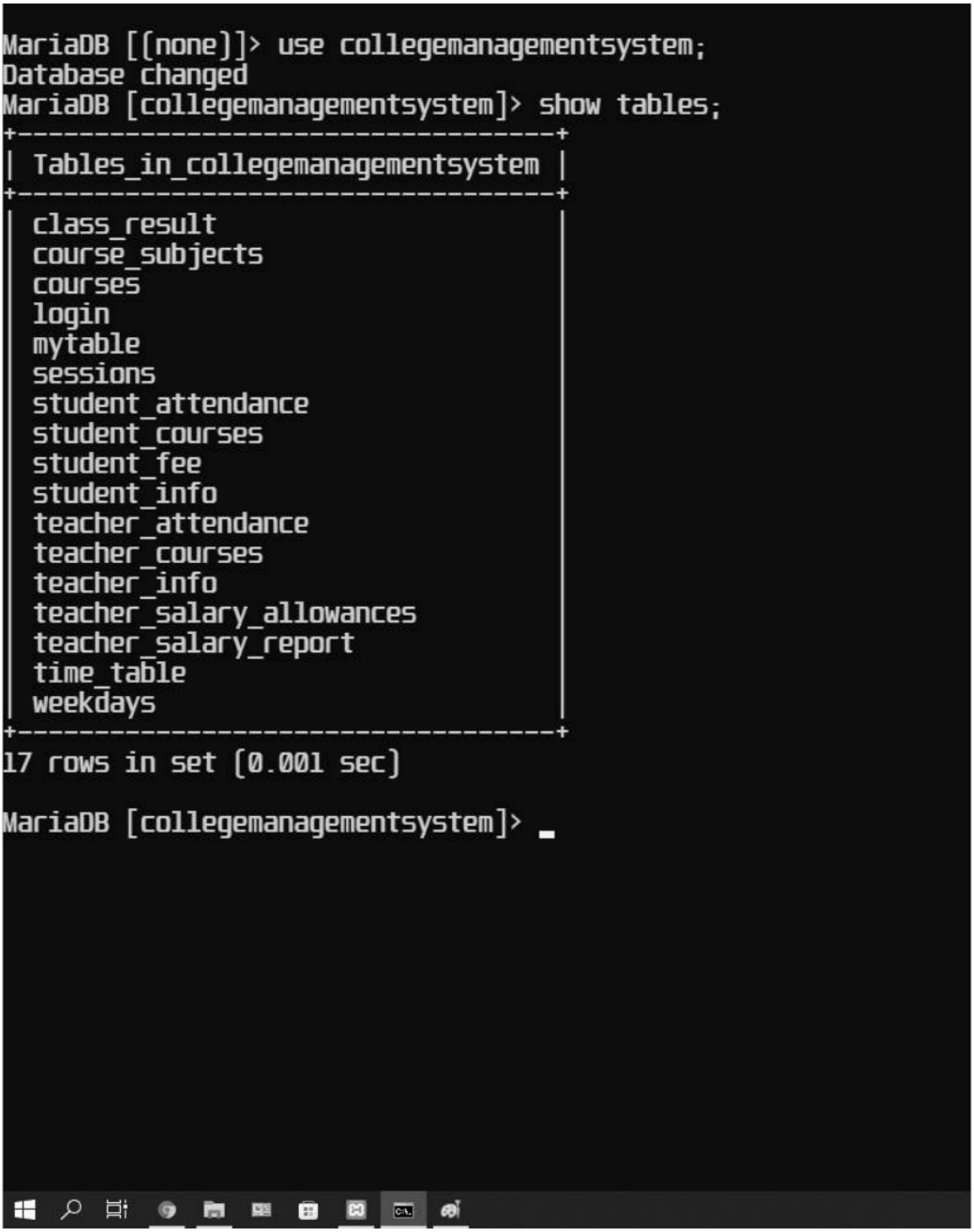


Creating Registration Form in HTML

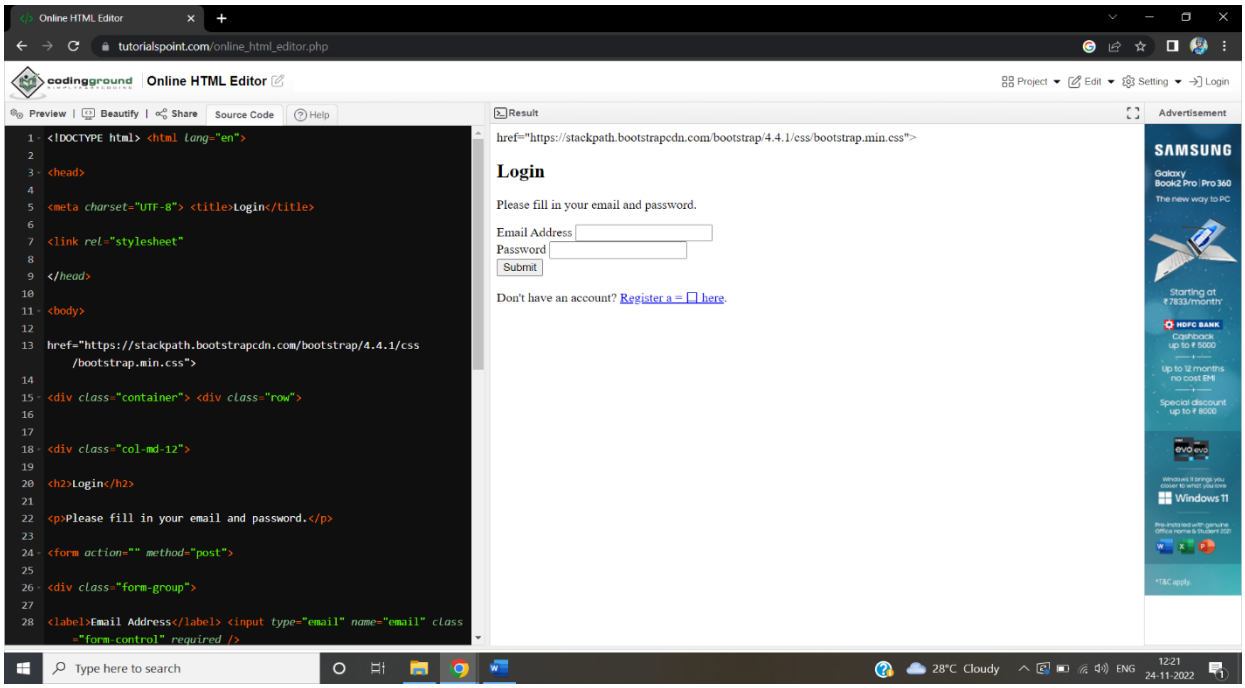


Creating the MySQL Database Table

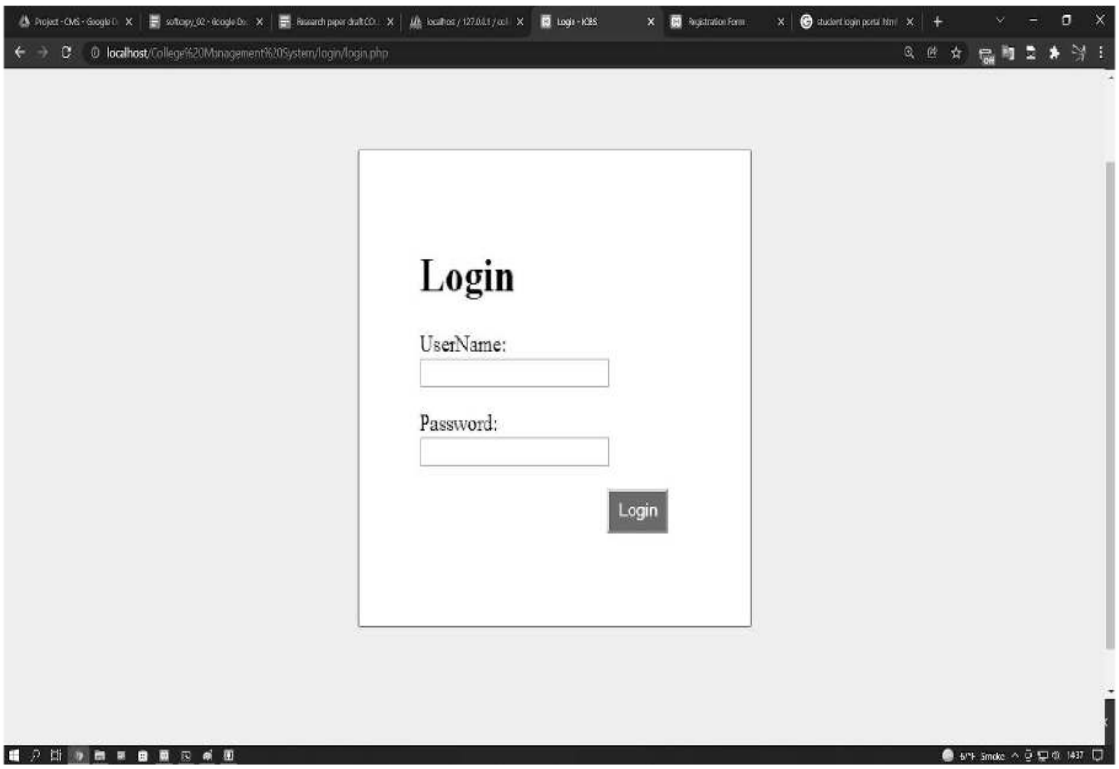
Required databases and tables:

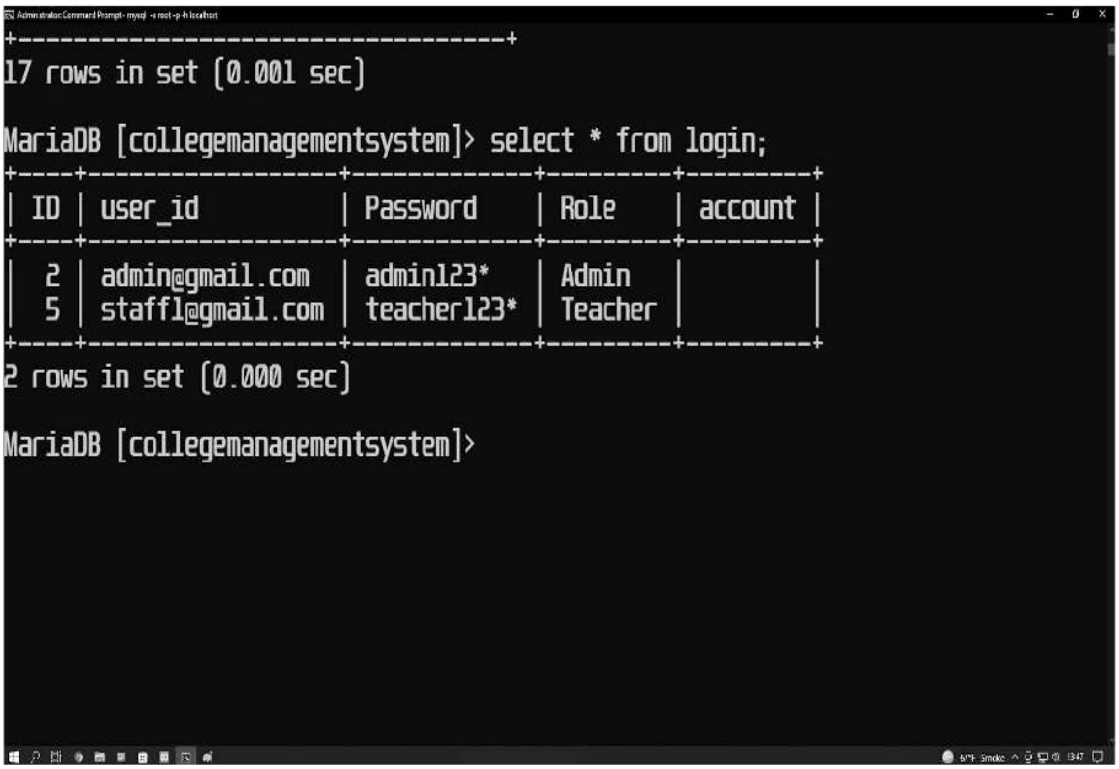


Creating a Login Form in HTML



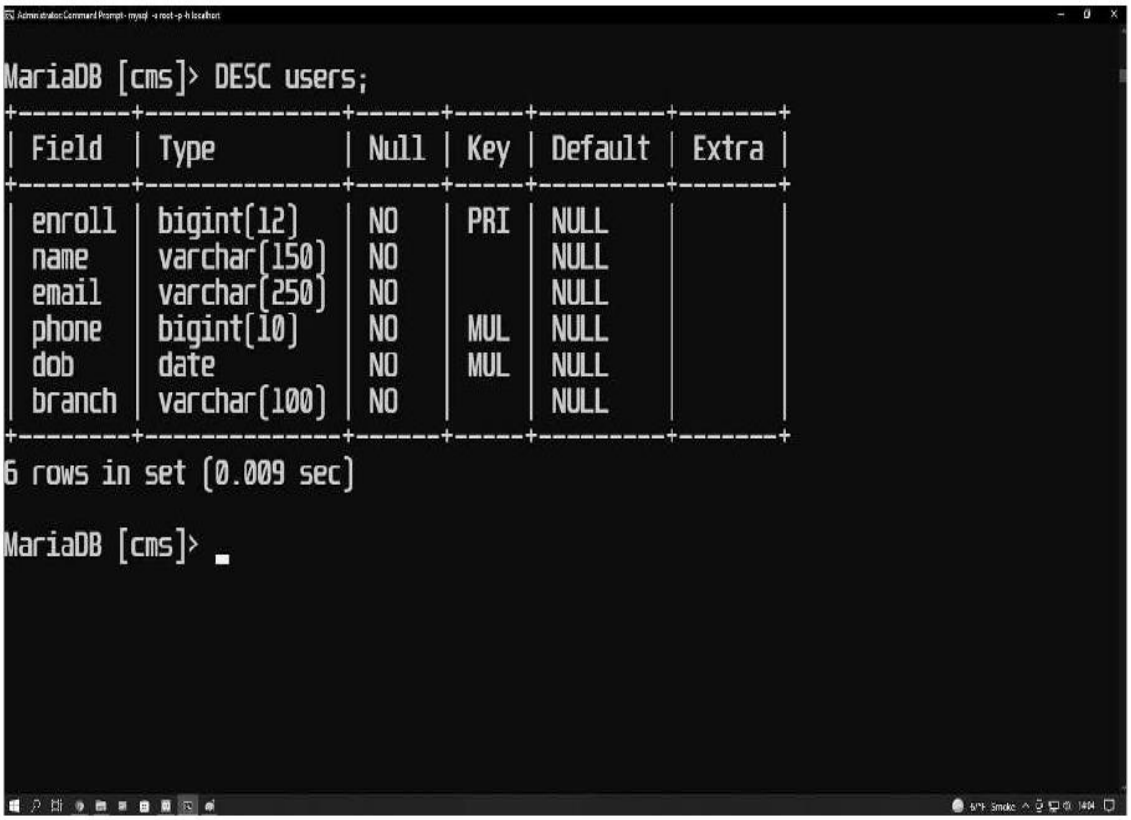
Student registration database's table





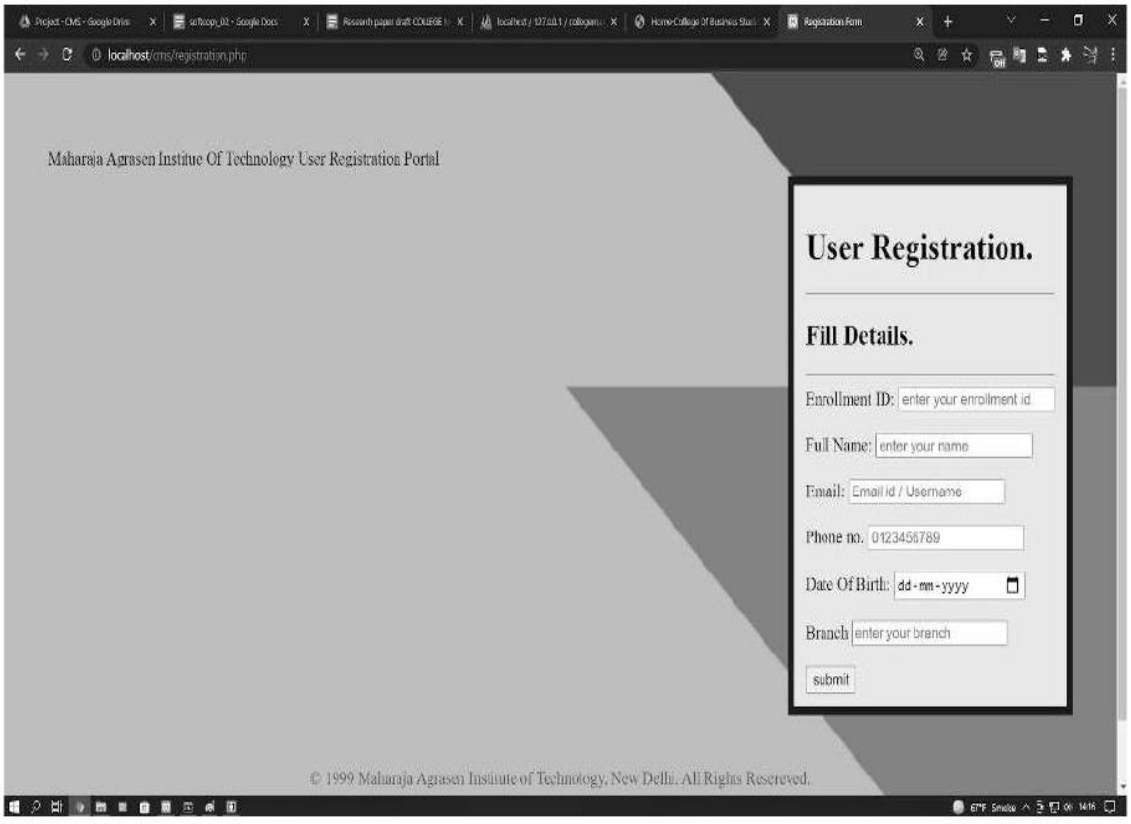
Student registration database's table:

User registration database's table;



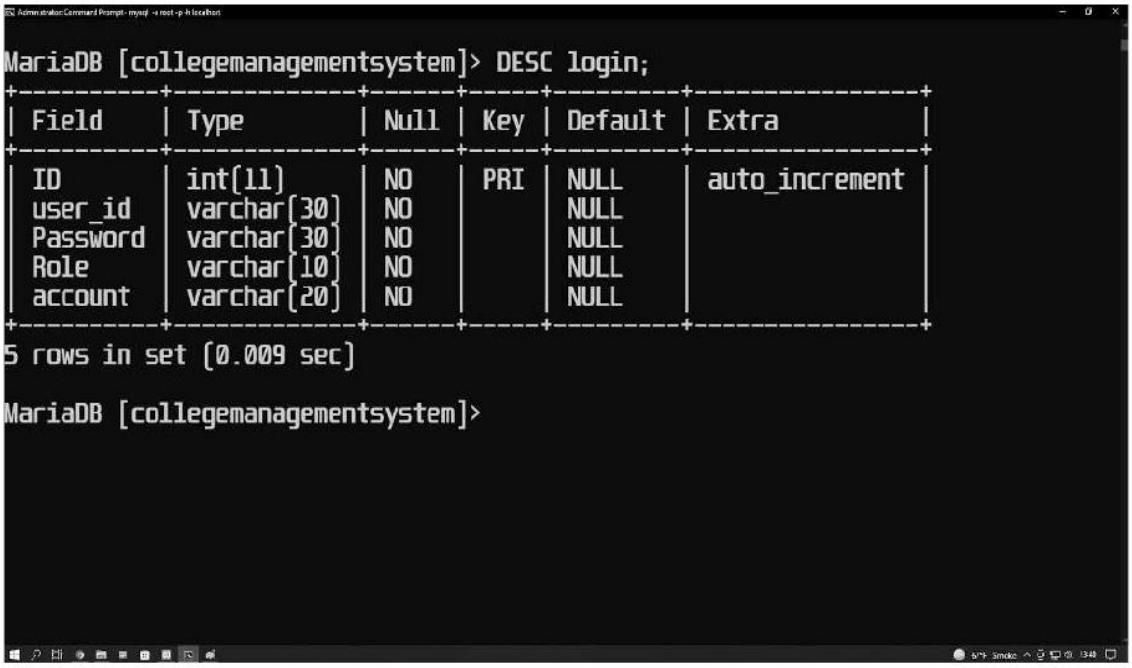
```
MariaDB [cms]> DESC users;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| enroll | bigint(12)   | NO   | PRI | NULL    |       |
| name   | varchar(150) | NO   |     | NULL    |       |
| email  | varchar(250) | NO   |     | NULL    |       |
| phone  | bigint(10)   | NO   | MUL | NULL    |       |
| dob    | date         | NO   | MUL | NULL    |       |
| branch | varchar(100) | NO   |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
6 rows in set (0.009 sec)

MariaDB [cms]>
```



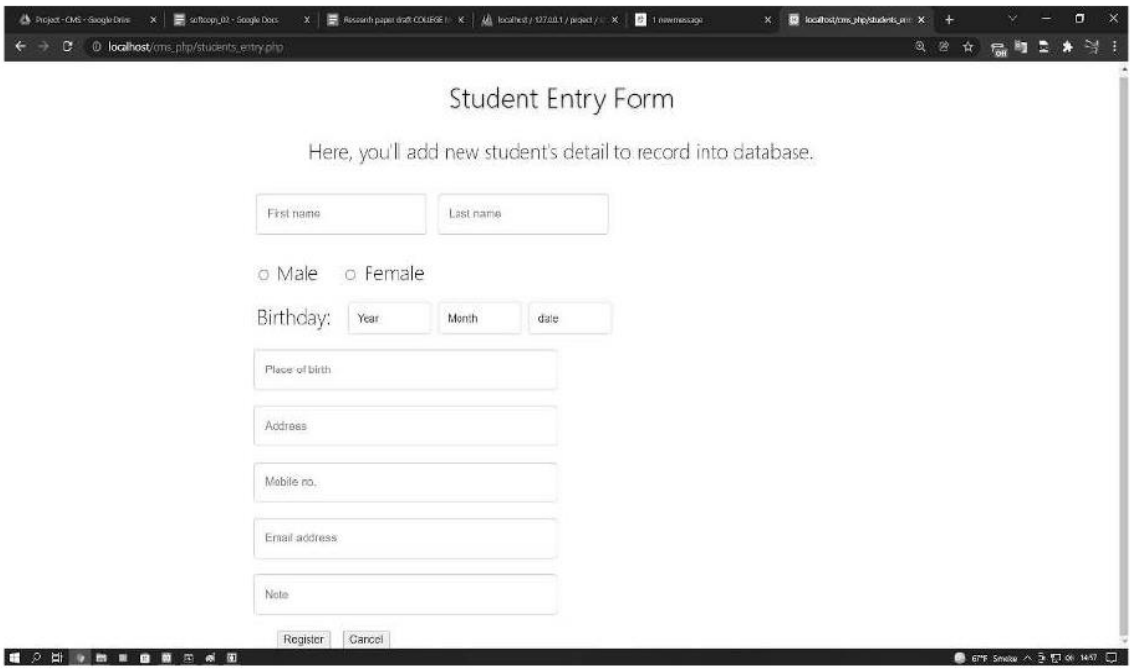
Login page credential table;

Login page credentials table:



```
MariaDB [collegemanagementsystem]> DESC login;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| ID       | int(11)     | NO   | PRI | NULL    | auto_increment |
| user_id  | varchar(30) | NO   |     | NULL    |                |
| Password | varchar(30) | NO   |     | NULL    |                |
| Role     | varchar(10) | NO   |     | NULL    |                |
| account  | varchar(20) | NO   |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
5 rows in set (0.009 sec)

MariaDB [collegemanagementsystem]>
```



Teacher/Staff members:

CHAPTER-5

Experimental Results

Experimental Results

College Management System is a internet based Web Portal that aims at providing information to all levels of management system for the College, This system can be used as an information management system for the college.

For a given registrar / staff / student (Technical / Non-technical) the Administrator creates a login id & password, using these registrar/ staff / students (Technical /Non-technical) can access the system to either upload or download some informational the database. College Management System Software allows college authorities to streamline the education process and manage students with ease and provide a unified environment where teaching & learning can thrive.

CHAPTER 6

CONCLUSION AND FUTURESCOPE

CONCLUSION AND FUTURESCOPE

This project assists in modifying the existing system to a site-based system. This is a paperless work it can be monitored and controlled remotely. It reduces the manpower required. It always provides accurate information; Malpractice can be reduced. All gathered and extra information can be saved and can be accessed at any time, The data which is stored in the project helps in taking intelligent and quick decisions by the management. So, it is better to have a Web-Based Information Management system. All the stakeholders, staff members can get the desired information without delay. This system is essential in colleges and universities.
1 have taken a lot of effort in completing this project.

However, it could not have been possible without the support and guidance of individuals and organizations. We would like to express my sincere thanks to all of them, Query mode, and resource management, in order to conduct accurate queries in the case of complex data.

Today, the market place is flooded with several school management options for colleges and institute to choose from. A variety of innovative products and services are being offered spoiling customers for choice, College Management Systemin more a privilege enjoyed by managers and employees. In the last couple of years, the growth of IT industry has been phenomenal as more have started discovering, the benefits of using this platform. Therefore, we will make this system live and provide Software as Service in future.

CHAPTER 7

REFERENCES

REFERENCES

- PHP-https://www.tutorial.spoint.com/php/-
- HTML-https://www.tutorial.spoint.com/html/-
- CSS -https://www.tutorial.sopint.com/css/-
- JAVASCRIPT- https://www.tutorial.spoint.com/javascript/-
- Web server/Apache -https://www.apachefriends.org/-