**The Project entitled**

**M.V.S.R NOTIFICATIONS**

Submitted in partial fulfillment of academic requirements for the award of the degree of

Bachelor of Engineering (Computer Science and Engineering)

**By**

**VADALI SAIKRISHNA (1608-14-733-099)**

**RUDRAVARAM SANDEEP KUMAR (2451-14-733-171)**

**KATURI VINAY BABU (2451-14-733-170)**

****

**Department of Computer Science and Engineering**

**M.V.S.R. ENGINEERING COLLEGE**

**(Affiliated to Osmania University & Recognized by AICTE)**

**Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510**

**2016-17.**

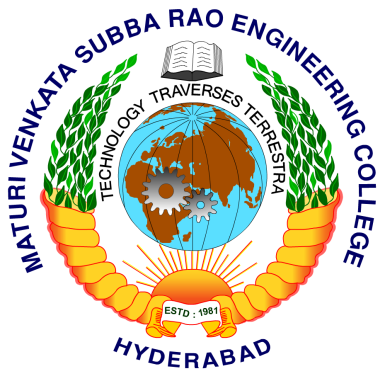
**Department of Computer Science and Engineering**

**M.V.S.R. ENGINEERING COLLEGE**

**(Sponsored by Matrusri Education Society, Estd. 1980)**

**(Affiliated to Osmania University & Recognized by AICTE)**

**Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510**

****

**CERTIFICATE**

This is to certify that the project entitled **“M.V.S.R NOTIFICATIONS”,** is being submitted by Mr.**V.SAIKRISHNA** bearing H.T No **1608-14-733-099,** Mr.**R.SANDEEP KUMAR** bearing H.T No **2451-14-733-171** &Mr.**K.VINAY BABU** bearing H.T No **2451-14-733-170** in partial fulfillment of academic requirements for the award of the degree of BACHELOR OF ENGINEERING in COMPUTER SCIENCE AND ENGINEERING from MVSR Engineering College, affiliated to OSMANIA UNIVERSITY, is a record of bonafide work carried out by us under the guidance and supervision of the faculty (CSED). The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of my knowledge and belief.

|  |  |  |
| --- | --- | --- |
| Guide |  | Project Coordinator |
| **M.V.R JYOTHISREE** |  | **VIKRAM NARAYANDAS** |
| Assistant Professor, |  | Assistant Professor |
| Dept. of CSE. |  | Dept. of CSE. |
| MVSR Engineering College |  | MVSR Engineering College |

**ACKNOWLEDGMENT**

I take this opportunity to express my profound and sincere gratitude to all those who helped us in carrying out this project successfully.

At the very outset, I am thankful to our principal Sri **Dr.V.CHANDRA SEKHAR** and Sri **J.PRASANNA KUMAR** ,Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for their consent to do the project work as a part of our B.E Degree (CSE). We thank them for their valuable suggestions and advice throughout our stay at the college, during our project work.

I would like to thank our Internal Project Guide Ms.**M.V.R JYOTHISREE**, Assistant Professor, Department of Computer Science and Engineering, MVSR Engineering College for her useful suggestions, guidance and encouragement.

We thank the teaching and non-teaching staff of CSE for extending their support.

Finally we are thankful to our parents for their cooperation and support throughout all endeavors in our life.

V.SAIKRISHNA (1608-14-733-099)

R.SANDEEP KUMAR (2451-14-733-171)

K.VINAY BABU (2451-14-733-170)

**ABSTRACT**

**MVSR NOTIFICATIONS** is an android app developed based on STUDENT INFORMATION SYSTEM DESIGN. This app is developed for the purpose of accessing of M.V.S.R. engineering college details through the most renowned android operating system application

This app was designed based on the principles of android system of the GOOGLE Company. The app is developed on ANDROID STUDIO BUNDLE provided by the GOOGLE.

College website was found to be the only system for accessing the complete information regarding the MVSR engineering college. But, an INTERNET option is definitely required for accessing that website. But, now-a-days, android system is found to be most useful way of accessing information regarding anything.

For this purpose we have come up with a new idea of developing an android app for the college.This app helps to not only get latest info regarding the college but also to be in regular contact with the college faculty , students , fests, etc., And other information can be obtained.

**List of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Chapter** | **Pg.no.** |
| **1** | Chapter 1 – **Introduction** | **1** |
| **2** | * 1. – motivation | **1** |
| **3** | * 1. – Problem Definition | **2** |
| **4** | 1.3 – Objectives | **2** |
| **5** | Chapter 2 **- System Analysis** | **3** |
| **6** | 2.1 - existing system | **3** |
| **7** | 2.2 - Proposed system | **3** |
| **8** | 2.3 - Requirements | **3** |
| **9** | Chapter 3 - **System Design** | **5** |
| **10** | 3.1 – Database Design | **5** |
| **11** | 3.2 – Activity Diagrams | **7** |
| **12** | Chapter 4 – **Implementation** | **10** |
| **13** | 4.1 – Project Sample Code | **12** |
| **14** | 4.2 - Installation Procedure | **20** |
| **15** | 4.3 - Test Cases | **21** |
| **16** | 4.4 - Output Screens | **22** |
| **17** | Chapter 5 – **Conclusion** | **29** |
| **18** | Future Enchancement | **29** |
| **19** | REFERENCES | **30** |

**1. INTRODUCTION**

A Student Information System (SIS) is a System that manages the records of student regarding admission and examination part.

A Student Information System (SIS) is designed to help collages for management of student. Extensive information is available at your fingertips through this System. Viewing student data, managing admission and reshuffling ,managing seats, quota, board, semester, faculty, category and for examination, block allocation, subject management , scheduling exam, notice ,assignments, result and related issues are made simple and easy. There are custom search capabilities to aid in finding student information and working on student records. This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources. SIS allows the keeping of personnel data in a form that can be easily accessed and analyzed in a consistent way.

The SIS module is a component covering many other student aspects from application to retirement. The system records basic personal information, admission information, education information regarding student. Leading edge systems provide the ability to "read" applications and enter relevant data to applicable database fields, notify student and provide result. Student management function involves

**1.1 MOTIVATION**

Choosing Student information system literally means the general information systems for maintaining and providing student information. It exists in all the schools, colleges, universities and any other education institutions. However, those information systems vary. Some of them are paper based; heavily manual work is involved in managing and maintaining information such as student personal records files. However, recently, most schools, even down to the very smallest, utilize computers in some way or anther. The uses to which the computers are put vary enormously, ranging from word processing and spreadsheet through to worldwide on-line access, complicated user access permission system and vast functionalities.

**1.2 PROBLEM DEFINITION**

Manual system not able to reduce redundancy and as a result increase in data duplication process and wastage of space, paper work and problem in finding reports and details on time. Each and every year during admission process and assigning new tasks to group of people various information has to be maintained by the organization and for making this process successful, they have to maintain various type of information in different files which is access by different departments of the organizations.

**1.3 OBJECTIVES**

With the increase of information, it is unwise to adopting the traditional paper based system which is slow to access and therefore, inefficient. Recent years, many systems were developed either by the universities or the software companies in order to partially automate many of the processes carried out by the department. Those developments dramatically reduce the time take in searching information and should enable the school to maintain precise and up-to-date information. For example, previously,

Students would have to enquire for much of his information from the school student office, which

Required more time and effort , particularly from members of staff. The new system is more efficient

And also often presents more accurate information. However, those systems vary even within the same

University . Some are quite well developed and implemented, some still adopting the inferior and out of-date technology. This is also one of the project’s aims: to analysis some sample systems in order to attain the strengths and eliminate the weaknesses in developing the new potential student information system.

**2. SYSTEM ANALYSIS**

**2.1 Existing System**.

The present system consisting of only entry of Attendance, results, assignment marks And updates to parents, the major drawback of the existing system is there is no accessing of student in order to view either attendance or marks allotted.

**2.2 PROPOSED SYSTEM**

The new proposed Student Information system will surely eradicate all these issues that they are facing now and this software is created in such a way that it automatically update the student details such as attendance, marks, notices and payment of exam fee etc... This software also keeps track of Student’s attendance and assignments submission.

**2.3 REQUIREMENTS**

The following are minimum hardware and requirements that should be present to run the project successfully.

Table 2.1 Tools and Technology

|  |  |
| --- | --- |
| Development technologies | HTML 5, SQL server 2005, JSP, JavaScript, XML |
| Development tools | NOTEPAD++ or Edit++ |
| Application server | IIS 7.5 |
| Database | SQL |
| Operating system | Windows XP Profession Edition, Windows Vista Enterprise Edition, Windows 7 and latest versions. |
| Web browser | Internet Explorer 6.0 and above |
| Hardware | intel, 1 GB RAM, 80 GB HDD |

Table 2.2:

|  |
| --- |
| **Client Configuration** |
| Intel, 750 MHz, 20GB HDD  Operating System: Windows XP/2000  RAM: 256 MB minimum  400MB Minimum Free Space on Drive  Microsoft Office |
| **Server Configuration** |
| 2 Servers, each with following configuration  2 CPU  Operating System: Win XP  RAM: 512MB Minimum  40GB Minimum Free Space on Drive  750MHz |
| One Server with Sql Application Server |
| Other Server with Sql Database Server |

Table 2.3:

|  |
| --- |
| **System Software** |
| MS Sql Application Server enterprise edition |
| MS Sql Database Server enterprise edition |
| Microsoft Office |
| Reporting server |
| IIS 6.0 or higher |

**3. SYSTEM DESIGN**

**3.1Database Design** 

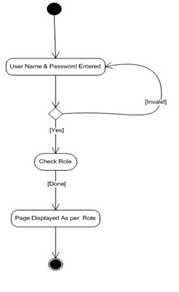
Figure 3.1.1 Data Diagram (Admission Module)



Figure 3.1.2 Data Diagram (Examination Module)

**3.2 ACTIVITY DIAGRAM**

**3.2.1 Activity Diagrams :**

Fig 3.2: activity Diagram for entering username and password

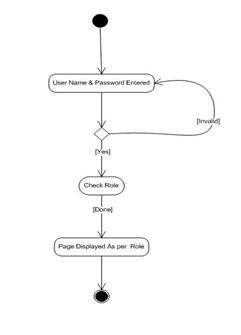


Fig 3.3 Activity Diagram for Login

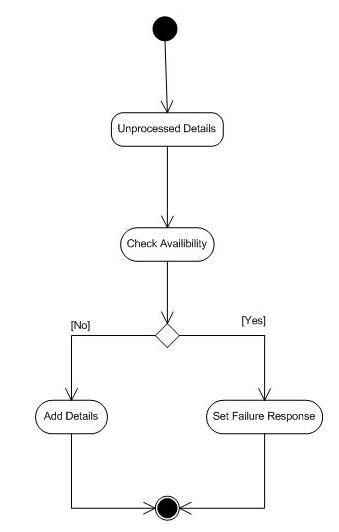


Fig 3.4 Activity Diagram for Adding Board, Quota and Designation

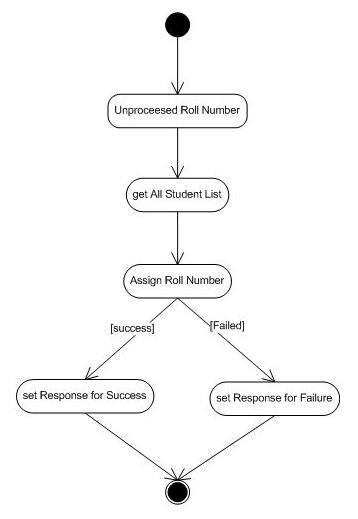


Fig 3.5 Activity Diagram for Assign Roll Numbers

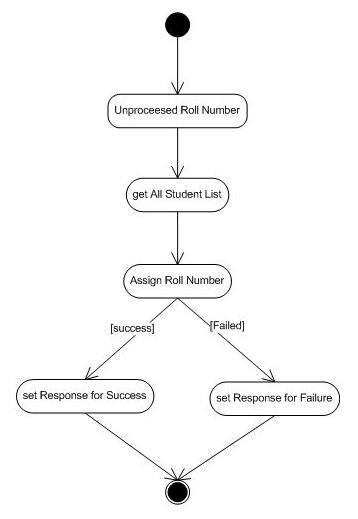


Fig 3.6 Activity Diagram for configuring Fees Details

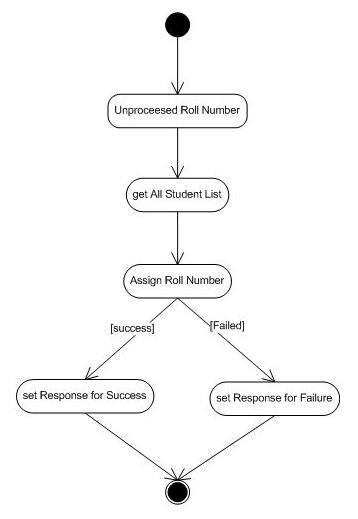


Fig 3.7 Activity Diagram for Getting Admission

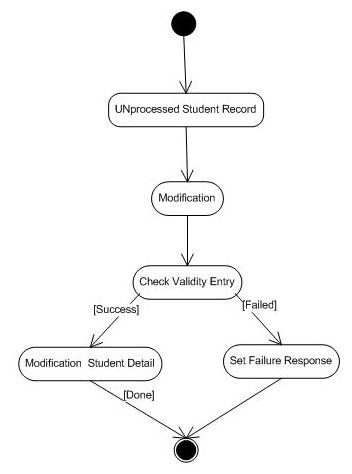


Fig 3.8 Activity Diagram for Modifying Student Details

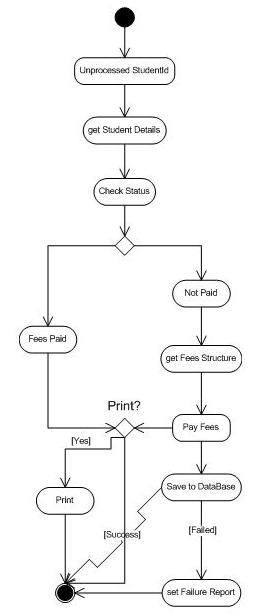


Fig 3.9 Activity Diagram for Pay Fees

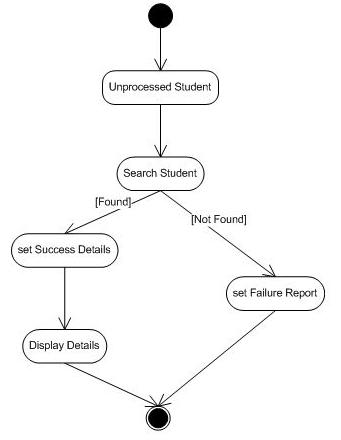
****

Fig 3.10 Activity Diagram for Searching Student

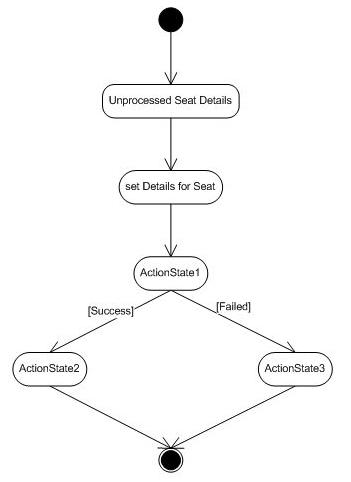
****

Fig 3.11 Activity Diagram for setting Seat for Admission

**4. IMPLEMENTATION**

We have also followed the web based **4-tier architecture** as the implementation architecture which is as follows:

**Modules:**

Admin

* Add (faculty, students)
* Delete

Faculty

* Add (students details)
* Entry of attendance
* Entry of marks
* Upload notices
* Upload assignments

Student

* View of attendance
* View marks
* View notice
* Queries

**4.1 SAMPLE CODE**

**1. Code for the 1st html login screen:**

</head>

<body>

<div class="wrapper col0">

<div id="topbar">

<div id="slidepanel">

<div class="topbox">

<h2>Welcome</h2>

<p>Welcome to MVSR Engineering College</p>

<p class="readmore"><a href="#">Continue Reading &raquo;</a></p>

</div>

<div class="topbox">

<h2>Teachers Login Here</h2>

<form action="tech.jsp" method="post">

<fieldset>

<legend>Teachers Login Form</legend>

<label for="teachername">Username:

<input type="text" name="tn" id="teachername" value="" Required />

</label>

<label for="teacherpass">Password:

<input type="password" name="tp" id="teacherpass" value="" Required />

</label>

<label for="teacherremember">

<input class="checkbox" type="checkbox" name="teacherremember" id="teacherremember" checked="checked" />

Remember me</label>

<p>

<input type="submit" name="teacherlogin" id="teacherlogin" value="Login" />

&nbsp;

<input type="reset" name="teacherreset" id="teacherreset" value="Reset" />

</p>

</fieldset>

</form>

</div>

<div class="topbox last">

<h2>Pupils Login Here</h2>

<form action="pupil.jsp" method="post">

<fieldset>

<legend>Pupils Login Form</legend>

<label for="pupilname">Username:

<input type="text" name="pn" id="pupilname" required value="" placeholder="username" />

</label>

<label for="pupilpass">Password:

<input type="password" name="pd" id="pupilpass" required value="" placeholder="password"/>

</label>

<label for="pupilremember">

<input class="checkbox" type="checkbox" name="pupilremember" id="pupilremember" checked="checked" />

Remember me</label>

<p>

<input type="submit" name="pupillogin" id="pupillogin" value="Login" />

&nbsp;

<input type="reset" name="pupilreset" id="pupilreset" value="Reset" />

</p>

</fieldset>

</form>

</div>

<br class="clear" />

</div>

<div id="loginpanel">

<ul>

<li class="left">Log In Here &raquo;</li>

<li class="right" id="toggle"><a id="slideit" href="#slidepanel">Administration</a><a id="closeit" style="display: none;" href="#slidepanel">Close Panel</a></li>

</ul>

</div>

<br class="clear" />

</div>

</div>

**2. Submit.java – code for inserting details into the page :**

importjavax.servlet.\*;

importjavax.servlet.http.\*;

import java.io.\*;

importjava.sql.\*;

public class submit extends HttpServlet

{

public void doGet(HttpServletRequestrequest,HttpServletResponse response)throws IOException,ServletException

{

response.setContentType("text/html");

PrintWriter out=response.getWriter();

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:Student1");

String fun=request.getParameter("t1");

intfpw=Integer.parseInt(request.getParameter("t2"));

PreparedStatementpstmt=con.prepareStatement("select \* from admin where username=?");

pstmt.setString(1,fun);

ResultSetrs=pstmt.executeQuery();

while(rs.next())

{

intdpwd=rs.getInt(2);

if(fpw==dpwd)

{

out.println("<a href=form.html>continue</a>");

}

else

{

out.println("<a href=index.html>try again</a>");

}

}

con.commit();

pstmt.close();

con.close();

}

catch(Exception e)

{

}

}

}

**3. Form.html**

<html>

<head>

</head>

<body>

<form name="f1" action="insert">

ename:<input type=text name="t1"><br>

eno:<input type=text name="t2"><br>

designation :<input type=text name="t3"><br>

salary:<input type=text name="t4"><br>

phoneno:<input type=text name="t5"<br>

<input type=submit value="submit">

<input type=reset value="reset">

</form>

</body></html>

**4.Student Insert’s :**

<%@page import="java.sql.\*" %>

<%

String user=request.getParameter("name");

String pwd=request.getParameter("pwd");

String email=request.getParameter("emailid");

String cno=request.getParameter("cno");

String dname=request.getParameter("dname");

String roll=request.getParameter("roll");

try

{

Class.forName("oracle.jdbc.OracleDriver");

Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","root");

PreparedStatement pst=con.prepareStatement("insert into studentdetails values(?,?,?,?,?,?)");

pst.setString(1,user);

pst.setString(2,pwd);

pst.setString(3,roll);

pst.setString(4,email);

pst.setString(5,cno);

pst.setString(6,dname);

int i=pst.executeUpdate();

PreparedStatement pst1=con.prepareStatement("insert into user\_login values(?,?,?)");

pst1.setString(1,user);

pst1.setString(2,pwd);

pst1.setInt(3,3);

int j=pst1.executeUpdate();

out.println("<font size=5 color='blue'>Successfully inserted <a href='adminhome.html'>&nbsp;&nbsp;Back</a></font>");

}

catch(Exception e)

{

out.println(e);

}

%>

**5. Web.xml**

<web-app>

<servlet>

<servlet-name>submit</servlet-name>

<servlet-class>submit</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>submit</servlet-name>

<url-pattern>submit</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>payslip</servlet-name>

<servlet-class>payslip</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>payslip</servlet-name>

<url-pattern>payslip</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>insert</servlet-name>

<servlet-class>insert</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>insert</servlet-name>

<url-pattern>insert</url-pattern>

</servlet-mapping>

</web-app>

**4.2 Installation procedure**

In Installation procedure first we have to know what technologies we are using in project , in Steganography project the technologies are Java and swings .

First we need to install jdk 1.4 or higher; in this we can execute java programs like applets and swings...Etc.

After installing java, set environment variables.

**Procedure to run the project**

* Go to the command prompt where your project had,
* Compile your project
* Run your project after successful compilation

**4.3 TEST CASES**

The following test-cases being mentioned here are done from starting of the application to its ending on an android emulator provided by android studio bundle emulator.

The functioning of the app and the data information being entered into the app as well as the output obtained are thoroughly tested. Following are the outcomes of the tests done:

|  |  |  |
| --- | --- | --- |
| **DESCRIPTION** | **EXPECTED OUTCOMES** | **ORIGINAL OUTCOME** |
| Clicking on the apk file of the application | The application opens normally as any other android app | The app opened as usual as other applications |
| Presence of tiles which will take you to next screen if clicked | The tiles will be visible on the screen with names and small descriptions | The tiles will take you to a link of a website like student information system |
| Presence of buttons like settings and other default icons | Clicking the buttons will enable you to either make settings to the app or know its features | As expected buttons helped in the convenience of app function |
| Pressing the back button of the android phone | The app runs in the background and gets paused | As expected app runs in background and gets paused |

**4.4 SCREENSHOTS**

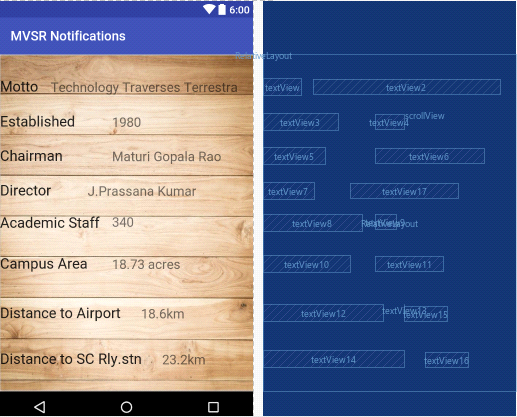


Fig 4.4.1: Details of the college data entered.

Screenshot and test case of a layout of the app “MVSR NOTIFICATIONS”

This screen shows the output on the left side and the layout test case on the right side.

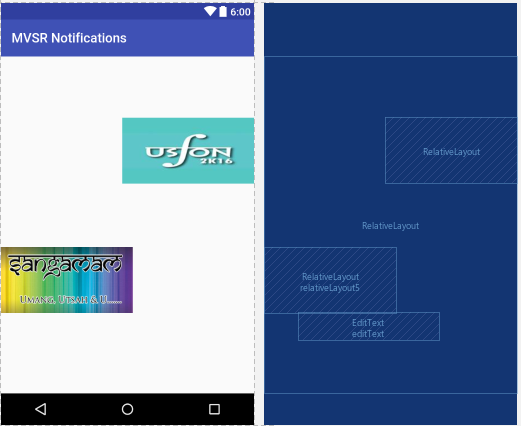
****

Fig 4.4.2: Sample layout of fests names being entered into the app

Screenshot of fests layout of the app…This screen shows sample icons of fests.

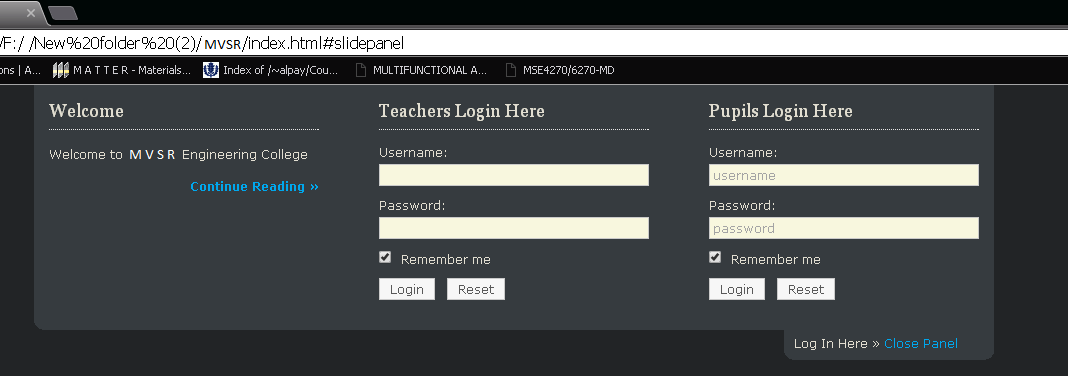
****

Fig 4.4.3: This screen shows the login page of the application

This is the first page of the student information system.

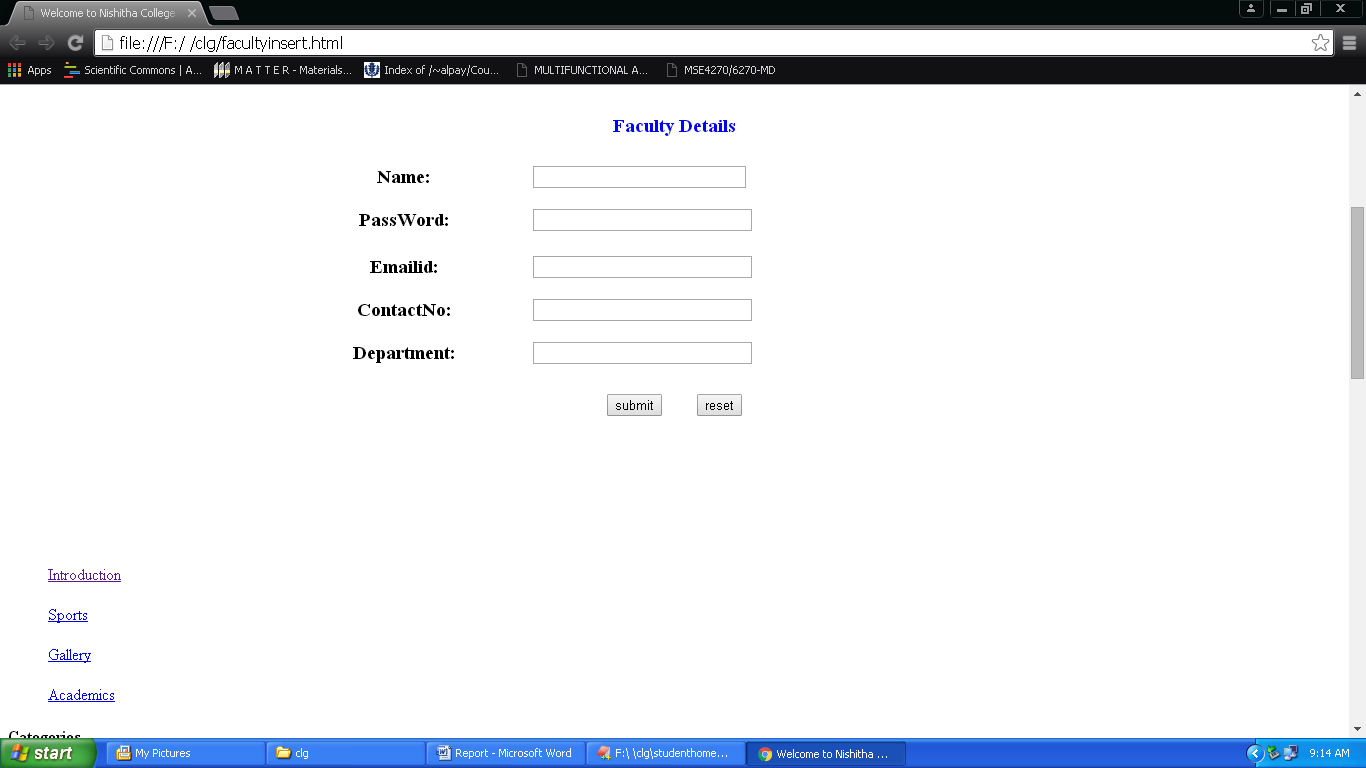


Fig 4.4.4: This screen shows the login page of details for faculty.

Faculty details like name, email ID, etc., have to be entered here.

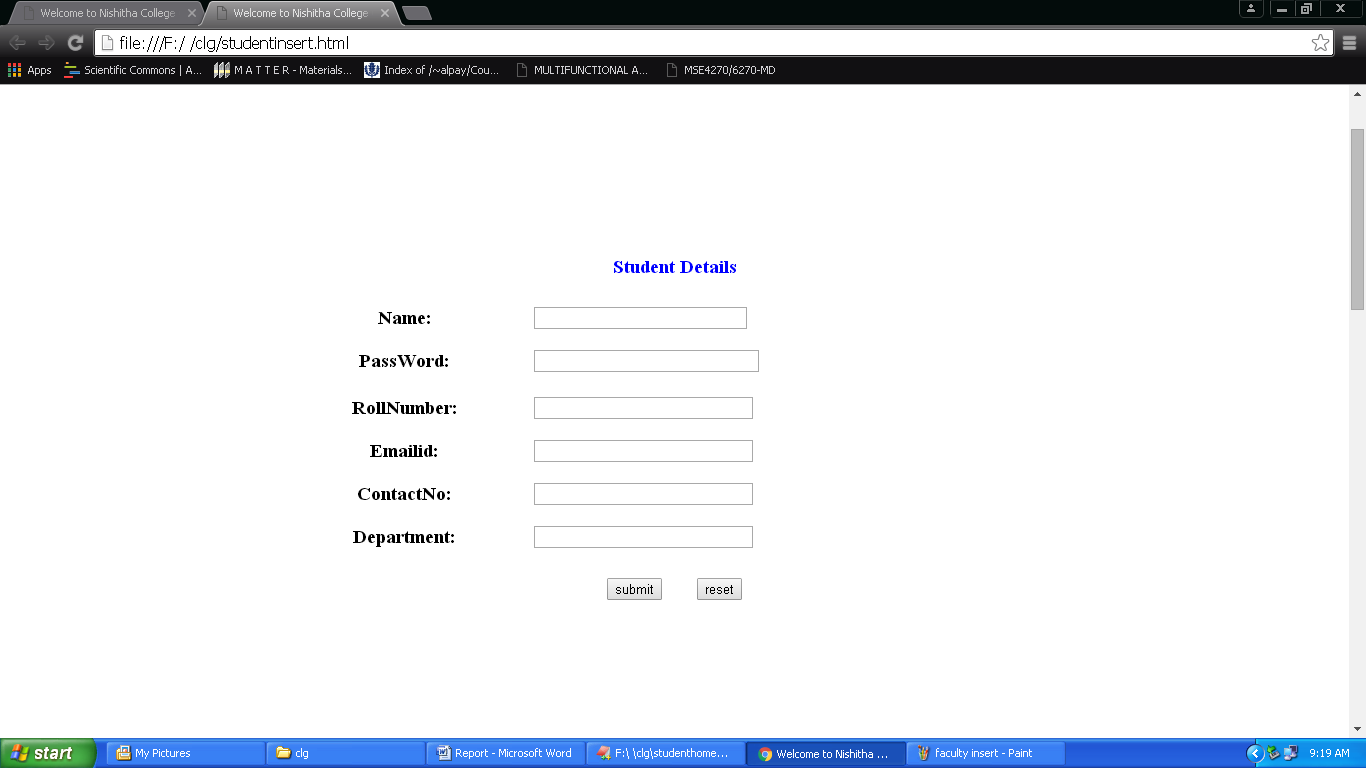


Figure 4.4.5: Add Student Details

Details of the student like name, email ID, password, etc., have to be entered here

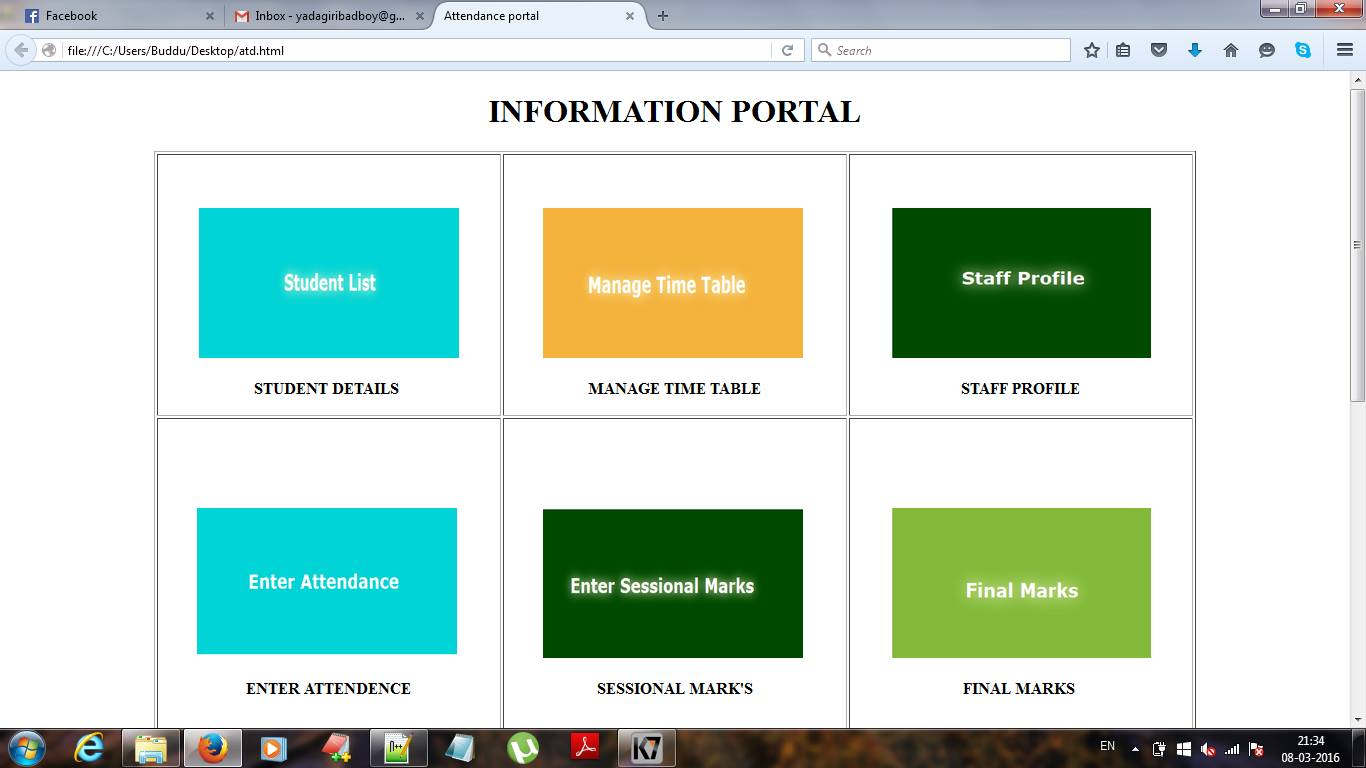


Figure 4.4.6: Faculty Home Page

Faculty can view their details like their students list, timetables, profiles, etc.

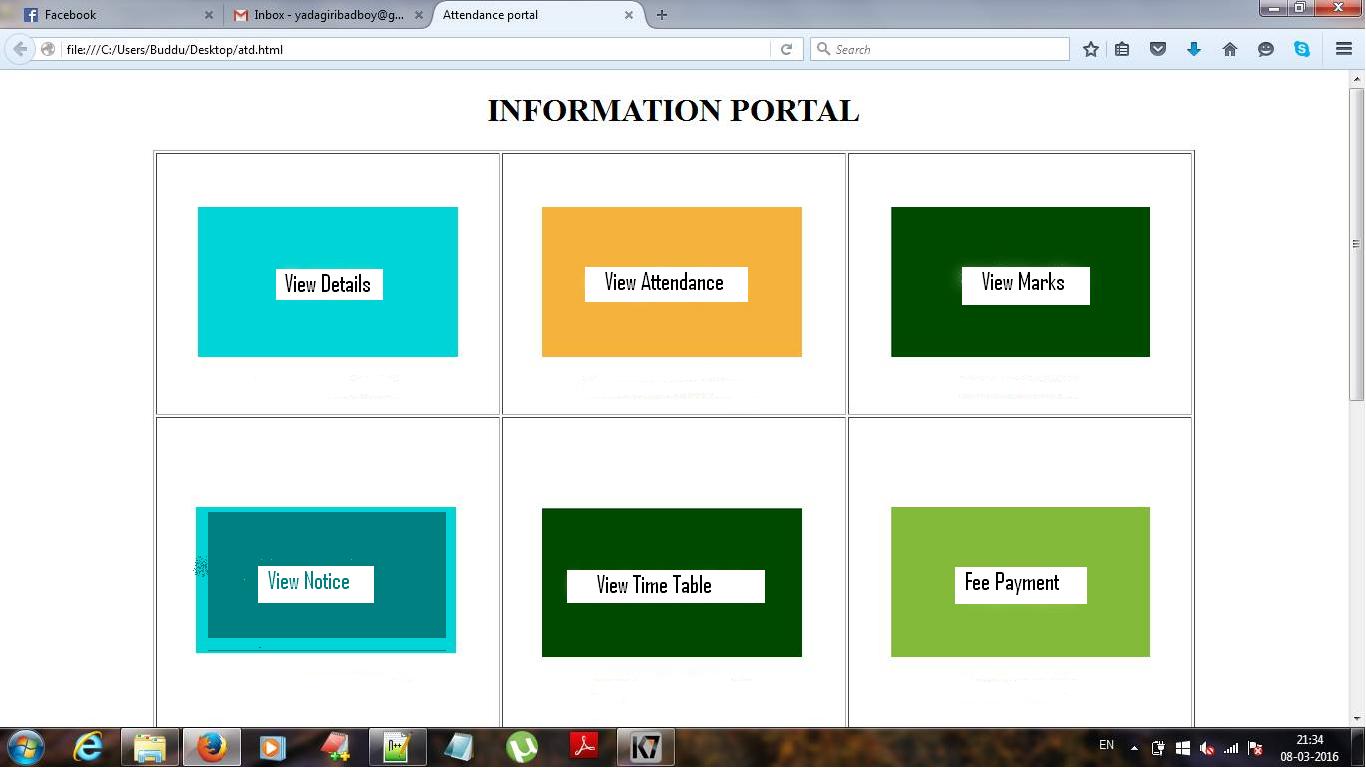


Figure 4.4.7: Student Home Page

The student can comfortably view his timetable, marks, fee payment details, etc.

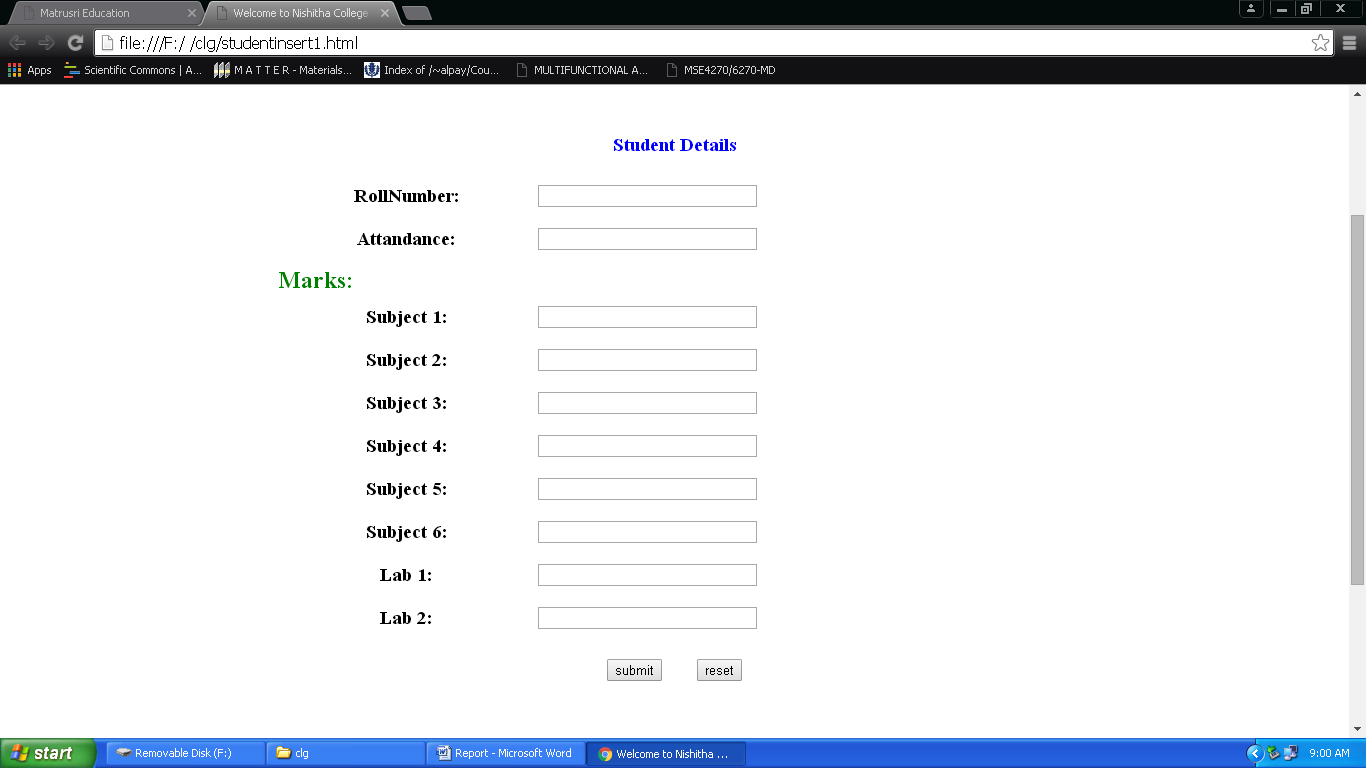


Figure 4.4.8: Enter student marks

The screen is prompting to enter student details and his marks respectively



4.4.9: Successful insertion of the Student Marks

Hence the details of a student are successfully and comfortably are inserted.

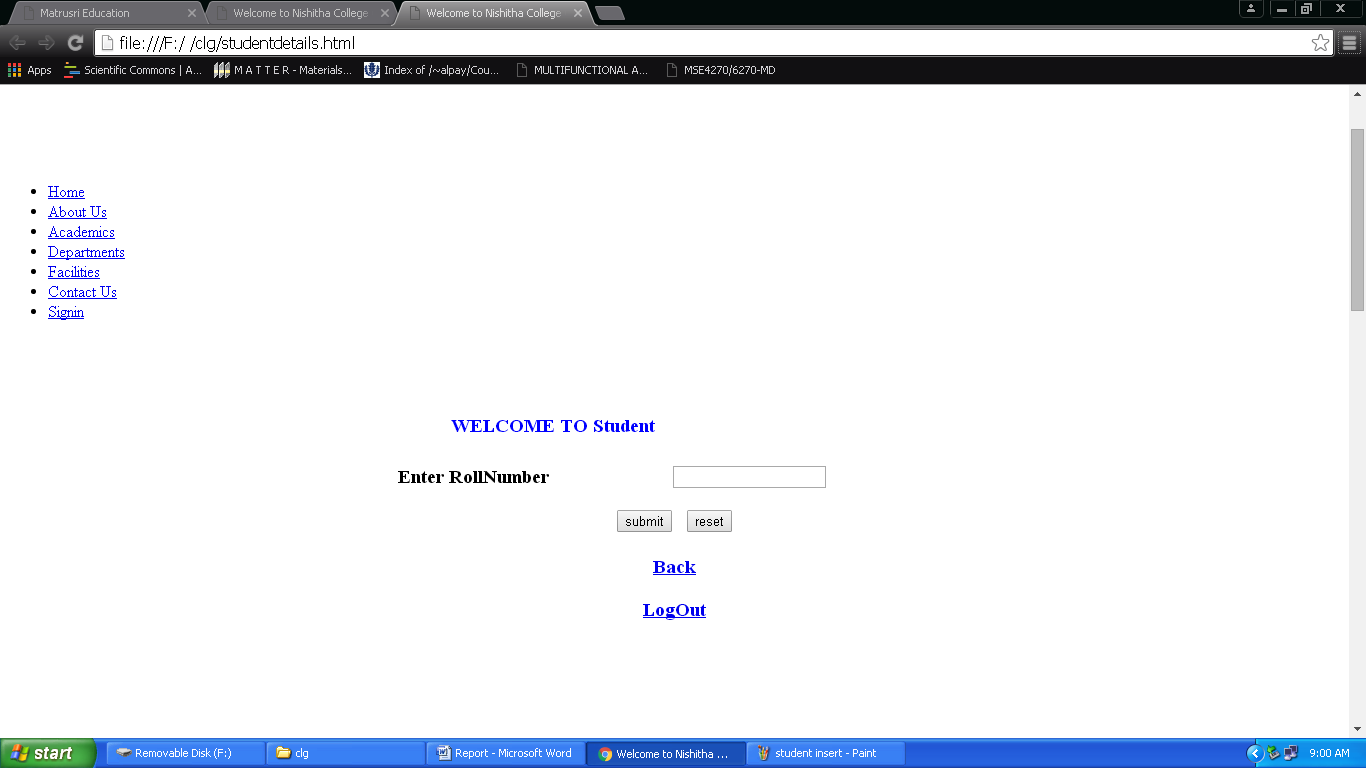


Figure 4.4.10: View the Student details

This screen asks for entering the roll number so that the details of the student can be seen.

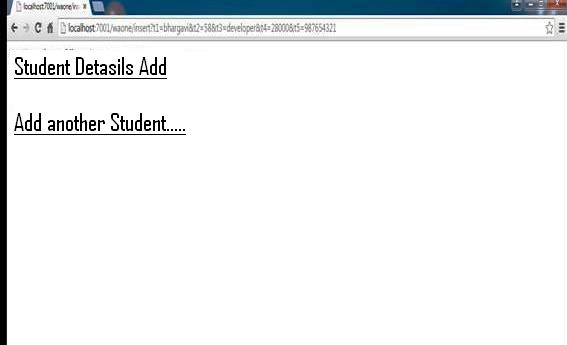


Figure 4.4.11: successful insertion of the Student details

This screen indicates that details of a student are successfully entered and would you like to add another student details.



Figure 4.4.12: indication of the Student details have inserted into database

This screen indicates that details are successfully inserted into the database using student information system.

**5. CONCLUSION**

The ANDROID APP “M.V.S.R NOTIFICATIONS” development will be helpful for

1. a student to access his required info regarding the college info, latest fest news, his attendance percentage, library facilities, information regarding the canteen, college timings etc.,
2. A parent to get acknowledged with the latest information regarding the MVSR ENGINEERING COLLEGE.
3. A guest to get introduced to the college
4. Easy to access from the android mobile phone itself as it’s an APP
5. College administration to get in contact with the outside world through android app
6. This Student Information processing software will improve the maintained of Student Information process, saving time ,paper work and money, and provided services with real-time access to accurate and up-to-date information.

**FUTURE ENHANCEMENT**

This app can be developed further into adding maps, gps, linking layouts with one another, developing an APK file so that it can be accessed from Google play store.

.

**REFERENCES**

1. Software engineering principles-roger s pressman

2. Database management system – Abraham silberschatz

3. Programming the World Wide Web – Robert w sebesta

4. Java server programming - Subramanian allamraju

5. <http://developer.android.com/guide/index.html>

6. <https://www.youtube.com/watch?v=SUOWNXGRc6g&list=PL2F07DBCDCC01493A> ---- new Boston android tutorials