Chapter 4

IMPLEMENTATION

Hardware Requirements

Processor	Pentium i3, i5 or more
RAM	2GB or Higher
Disk Space	10GB

Software Requirements

Operating System	Windows 7 or higher
Language	JSP, HTML, CSS,
Database	MySQL
Tools	Apache Tomcat

4.1 Front end and Back end used

MySQL

MySQL can be used for a variety of application, but it is most commonly found on Web servers. A Website that uses MySQL may includes Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Website that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a programming language like Java to access information from the database. MySQL commands can be incorporated into the Java code, allowing part or all of a web page to be generated from database information. Because both MySQL and Java are open source, the Java/MySQL combination has become a popular choice for database-driven websites.

MySQL is used as back end. MySQL is a powerful Relational Database Management System (RDBMS) which we will use the learn the basic principles of database using Structures Query Language (SQL) statement. SQL is a database language that is used to retrieve, insert, delete, update store data. This is achieved by constructing conditional statements that conform to a specific syntaxes.

How does MySQL works

MySQL is a database server program and as such is installed on the machine, but can be used as a 'server' for the database to a variety of location. Figure 4.1 shows the working of MySQL

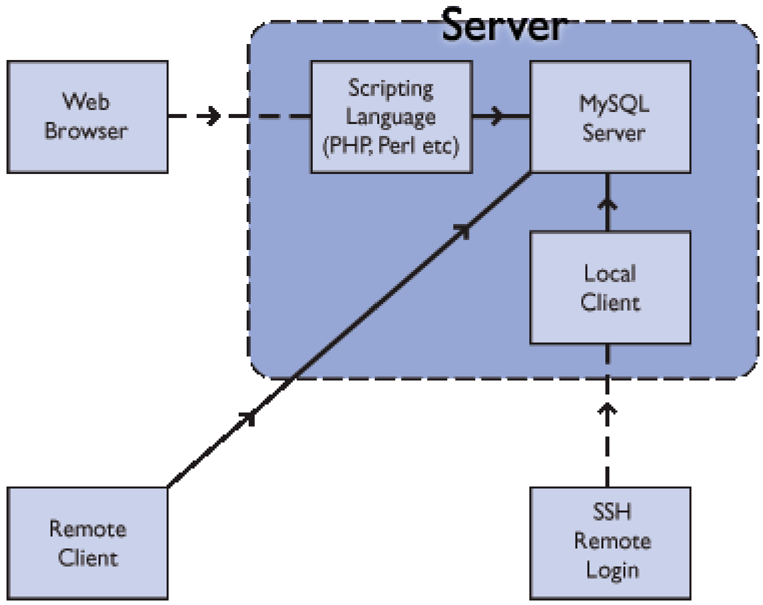


Figure 4.1 MySQL Server

The MySQL Server is installed on a Server and can be accessed directly via various client interfaces, which send SQL statements to the server and then display the results to a user. Some of these are:

A Local Client - a program on the same machine as the server. An example of this is the command line MySQL client software we will be using in the rest of the MySQL workshops (although there are other programs including graphical interfaces).

A Scripting Language - can pass SQL queries to the server and display the result.

A Remote Client - a programme on a different machine that can connect to the server and run SQL statements.

You can also use two more indirect methods.

Remote Login - You may be able to connect to the Server Machine to run one of its local clients.

Web Browser - you can use a web browser and scripts that someone has written (we're going to use this method for the rest of the workshop).

JSP stands for Java Server Pages. Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of Servlets, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the Servlet application program interface (API). JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as VBScript or JScript) before the page is sent to the user.

HTML

HTML is the standard markup language for creating Web pages. HTML stands for Hyper Text Markup Language. Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages.

CSS

CSS stands for Cascading Style Sheets. It is a language used for describing the presentation of the document written in markup language like HTML. It is used for describing the presentation of Web pages, including colors, layouts and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based Markup Language

TOMCAT

Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions. Tomcat can be used as either a standalone product with its own internal Web server or together with other Web servers, including Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later.

4.2 Coding

Code for Login page

```
@WebServlet("/login")

public class login extends HttpServlet {

        protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException

        {

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

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

        if(uname.equals("admin")&& pass.equals("root"))

        {

                HttpSession session = request.getSession();

                session.setAttribute("username", uname);

                session.setAttribute("password",pass);

                response.sendRedirect("enter.html");
        }

        else

        {

                response.sendRedirect("admlogin.jsp");

        }



}

}
```

Code for Insertion

```
String url="jdbc:mysql://localhost:3306/srms";

String usrname="root";

String pwd="loseyourself";

String query="insert into marks values(?,?,?,?,?,?,?,?,?,?)";

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(url,usrname,pwd);

        String usn=(String)session.getAttribute("usn1");

        String course_id=(String)session.getAttribute("course_id1");

        int sem=Integer.parseInt(request.getParameter("sem"));

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

        int internal=Integer.parseInt(request.getParameter("internal"));

        int external=Integer.parseInt(request.getParameter("external"));

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

        String query="insert into result values(?,?,?,?,?,?,?,?,?)";
```

```
 PreparedStatement st1=con.prepareStatement(query);
st1.setString(1,usn.toUpperCase());
st1.setString(2,sub_code1.toUpperCase());
st1.setString(3,course_id.toLowerCase());
st1.setInt(4,internal1);
st1.setInt(5,external1);
st1.setString(6,grade1.toUpperCase());
st1.setInt(7,point1);
st1.setString(8,status1.toUpperCase());
st1.setInt(9,sem);
int count1=st1.executeUpdate();
st1.close();
con.close();%>
```

Code for Search

```
$searchword=$_POST['txt_search'];
$query_Recordset1="select * from tbl_userreg where firstname LIKE '%".$searchword."%'";#
```

Code for View Result

```
String usn=request.getParameter("usn");
int sem=Integer.parseInt(request.getParameter("sem"));
st.setString(1,usn);//result table
st.setInt(2,sem);//result table
pst.setString(1,usn);//student table
```

```
ResultSet rs=st.executeQuery();//result
ResultSet rse=pst.executeQuery();//student
rse.next();
%>
 <%=rse.getString(1) %>
<br>
 <%=rse.getString(2)+""+rse.getString(3) %>
<br>
 <%=rse.getString(4) %>
<br>
<%=rse.getString(5) %>
<br>
<%=rse.getString(6) %>
<br>
<%=rse.getString(7) %><br>
<br>
<%out.println(sgpa); %>
<%while(rs.next())
{<tr><td><%=rs.getString(2)%></td>
<td><%=rs.getInt(4)%></td>
<td><%=rs.getInt(5)%></td>
<td><%=rs.getString(6)%></td>
<td><%=rs.getInt(7)%></td>
<td><%=rs.getString(8)%></td></tr>

<%} %></table>

</body>

</html>
```

Code for Connection

```
<%

String url="jdbc:mysql://localhost:3306/srms";

String usrname="root";

String pwd="loseyourself";

String query="select* from result where usn=? and sem=?";

String querys="select* from student where usn=?";

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection(url,usrname,pwd);

PreparedStatement st=con.prepareStatement(query);//result table

PreparedStatement pst=con.prepareStatement(querys);//student table

    %>
```

Code for logout

```
public class logout extends HttpServlet {

        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        HttpSession session=request.getSession();

        session.removeAttribute("username");

        session.removeAttribute("password");
```

```
            session.invalidate();

            response.sendRedirect("intro.html");

            }

}
```

Code for Calculating SGPA

```
if(sem==2)

{   int sum0=0;

            int sum4=0;

            int sum2=0;

            while(rs.next())

              {   String scr=rs.getString(2);

                        if(scr.equals("15CIV28"))

                          {

                                    sum0=sum0+(0*rs.getInt(7));

                          }

                        elseif(scr.equals("15CHEL27")||scr.equals("15CPL26"))

                          {

                         sum2=sum2+(2*rs.getInt(7));

                          }

                        else

                          {

                          sum4=sum4+(4*(rs.getInt(7)));

                          }
```

```
         }
        sgpa=(sum0+sum4+sum2)/24.0;
        rs.beforeFirst();
}
else
{int sum2=0
;int sum4=0;

while(rs.next())
{ String scr=rs.getString(2);
if(scr.equals("15CPL26")||scr.equals("15CSL37")||scr.equals("15CSL38")||scr.equals("15CSL47")||scr.equals("15CSL48")||scr.equals("15EEL37")||scr.equals("15EEL38")||scr.equals("15EEL47")||scr.equals("15EEL48")||scr.equals("15ECL37")||scr.equals("15ECL38")||scr.equals("15ECL47")||scr.equals("15ECL48")||scr.equals("15EIL37")||scr.equals("15EIL38")||scr.equals("15EIL47")||scr.equals("15EIL48")||scr.equals("15MEA306")||scr.equals("15MEA307")||scr.equals("15MEA308")||scr.equals("15MEB406")||scr.equals("15MEB407")||scr.equals("15MEB408")||scr.equals("15CVL37")||scr.equals("15CVL47")||scr.equals("15CVL38")||scr.equals("15CVL48"))
{
sum2=sum2+(2*rs.getInt(7));
}
else
{sum4=sum4+(4*(rs.getInt(7)));
}
sgpa=(sum2+sum4)/28.0;
```

}

4.3Applications of Project Work

i. Student Registration Form is to retrieve and update student information.

ii. Print view of result can be done.

iii. Ability to show SGPA along with result marks.

iv. Growth of data is supported.

v. Helps to delete student details.

vi. Eligibility status of members can be easily viewed.

4.4Discussion of the Results

Title Page: In Title page it allows the user to use the website either as admin for insertion or as a student to check result as shown in Figure 4.1.

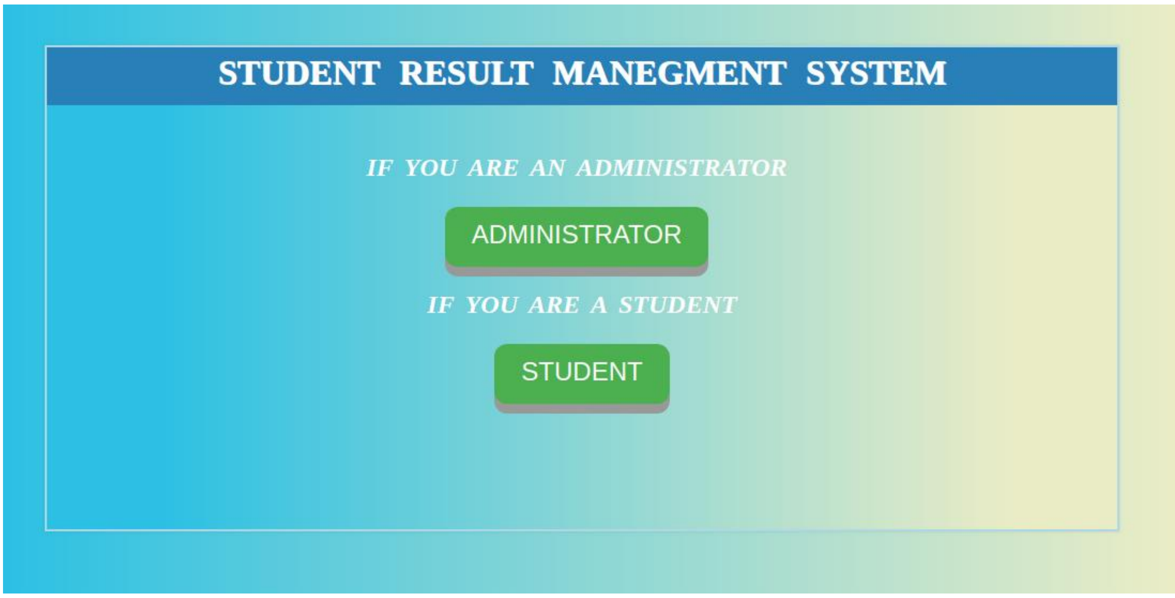
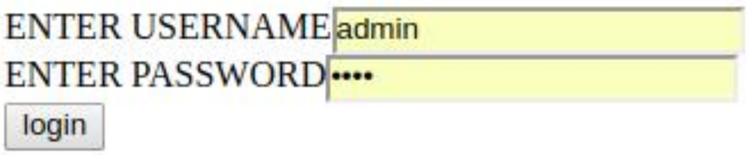


Figure 4.1 Title Screen

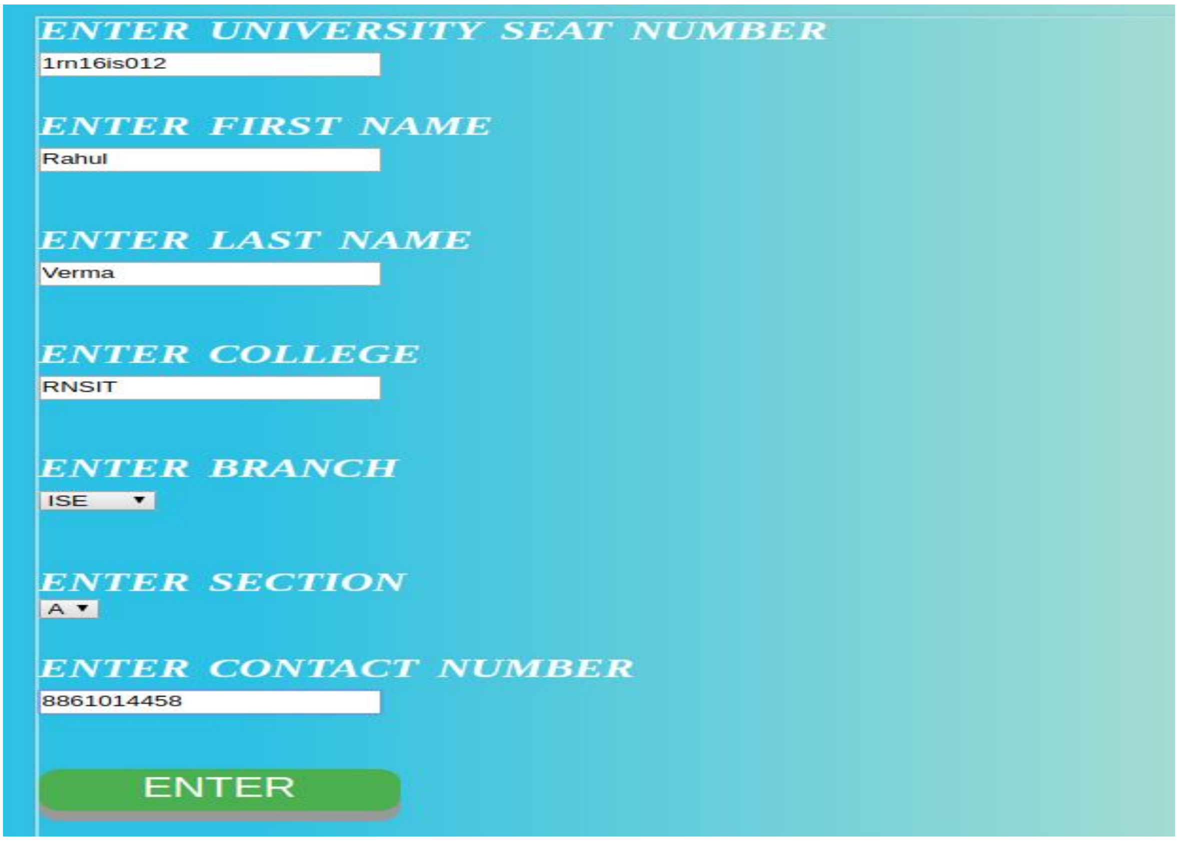
Login Page: In login page it allows the admin to enter his username and password as shown in Figure 4.2. If login is successful it goes to main frame.



ENTER USERNAME admin
ENTER PASSWORD ••••

Figure 4.2 Login Screen

Student Insertion Page: In this page it allows the admin the enter his student details such as usn, name, course and college as shown in Figure 4.3. After insertion admin can click on submit button after which he will get an option to logout or to enter marks.



ENTER UNIVERSITY SEAT NUMBER
1rn16is012

ENTER FIRST NAME
Rahul

ENTER LAST NAME
Verma

ENTER COLLEGE
RNSIT

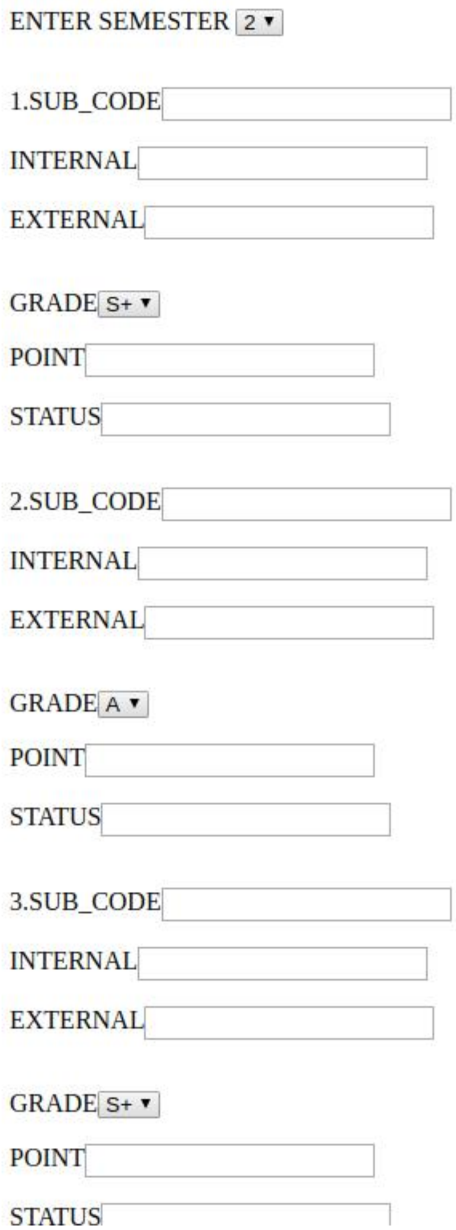
ENTER BRANCH
ISE

ENTER SECTION
A

ENTER CONTACT NUMBER
8861014458

Figure 4.3 student insertion

Result Insertion Page: In this page it allows the admin to enter student's marks in respective subjects as shown in Figure 4.4. At submission, admin will get logout option. .



ENTER SEMESTER 2 ▾

1.SUB_CODE

INTERNAL

EXTERNAL

GRADE S+ ▾

POINT

STATUS

2.SUB_CODE

INTERNAL

EXTERNAL

GRADE A ▾

POINT

STATUS

3.SUB_CODE

INTERNAL

EXTERNAL

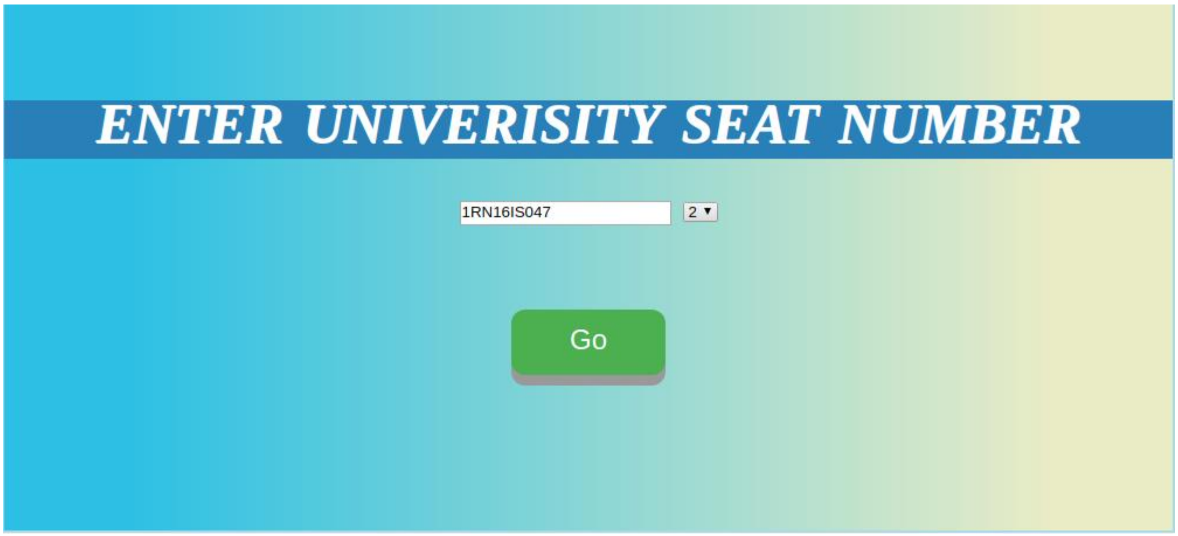
GRADE S+ ▾

POINT

STATUS

Figure 4.4 Result Insertion

Student login Page: In this page it allows the student to enter his university seat number and select a semester. Then click on submit button to view the result as shown in figure 4.5.

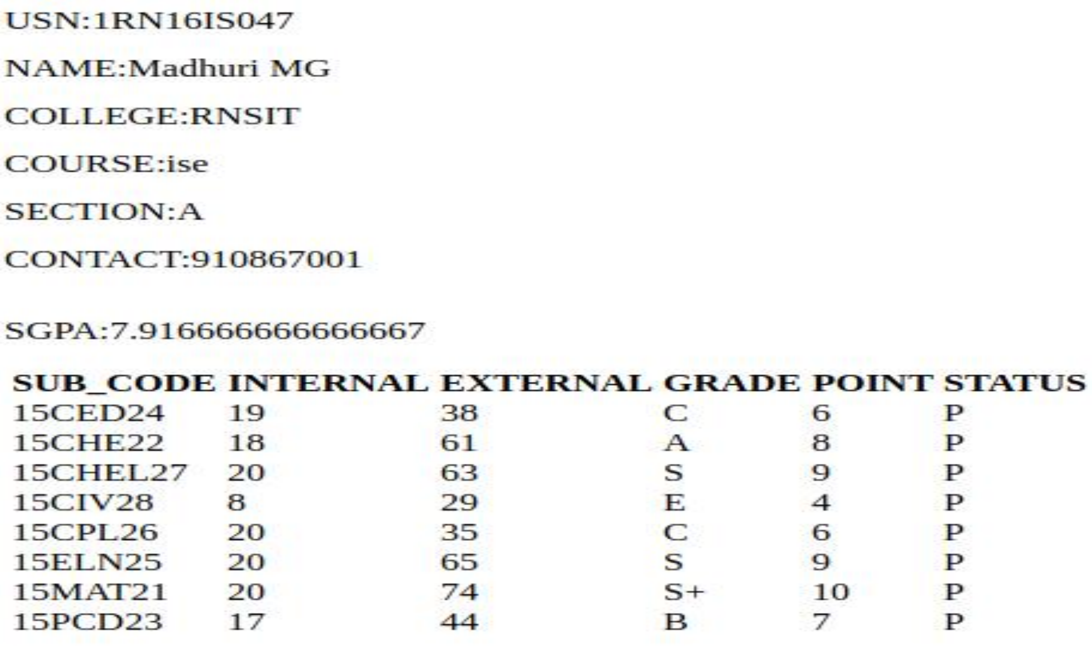


ENTER UNIVERISITY SEAT NUMBER

1RN16IS047 2 ▾

Figure 4.5 Enter University Seat Number

View Results Page: In view results the student can check their results of the selected semester with marks scored in respective subjects, grades, eligibility and grade points scored. This page also shows the student details such as his course and his section. The page view is shown in Figure 4.6



USN:1RN16IS047

NAME:Madhuri MG

COLLEGE:RNSIT

COURSE:ise

SECTION:A

CONTACT:910867001

SGPA:7.916666666666667

SUB_CODE	INTERNAL	EXTERNAL	GRADE	POINT	STATUS
15CED24	19	38	C	6	P
15CHE22	18	61	A	8	P
15CHEL27	20	63	S	9	P
15CIV28	8	29	E	4	P
15CPL26	20	35	C	6	P
15ELN25	20	65	S	9	P
15MAT21	20	74	S+	10	P
15PCD23	17	44	B	7	P

Figure 4.6 View Result