“STUDENTS MANAGEMENT SYSTEM”

1.0 RATIONALE:

This Student Management System is developed using Java as the Frontend and MS Access Database integration. It consist of Action Listener event and also uses database for storing information. This system can add, edit/update, delete, and search a particular student. The aim of this Student Management System project is to build a student registration system that will completely automate the process of new student registration in a university. The system will handle the document submission and registration of new students. The system will be web based using database. The database operation will be to update, view, add new students and also displaying the database. With the use of Microsoft access database one can easily insert data in the database through a registration form. This project displays the entire data stored in a database in a JTable. This project can be useful for any college to store the database of the students. One can easily search the record of a particular student by applying conditions. There is no need to search the data of a particular student for the admin, the system manually will display the desired output of the choice

Advantages-

1. Eco-Friendly: paperwork can be avoided
2. Efficient control over student data
3. Monitor student performance
4. Cost-efficient and User-friendly
5. Single solution for total College management
6. Easy access marks, grades and students data

Disadvantages-

1. Only, people who are accustomed to regular use of smartphones or computers can operate this software
2. With huge flow in traffic the application is prone to performance issues
3. Absence of proper internet-network makes it difficult for a user to access information, which is a significant disadvantage
4. The risk of data mishandling might be bothersome; but all these drawbacks can be evaded by choosing proper, cost-efficient and best software that best benefits an organization.

2.0 COURSE OUTCOMES INTEGRATED:

1. Firstly we got knowledge how to create a database in Microsoft Access
2. Learned java database connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through java programs
3. Learned the basics of advanced java programming and got to know how to create a form through a Frame window
4. We learned how to store the data entered in a form into a database and also how to display it in a JTable
5. Got knowledge about how to do modifications in a database through advanced java programming

3.0 LITERATURE REVIEW:

References:

1. https://www.dummies.com/software/microsoft-office/access-2019-how-to-create-an-access-database/ :
   This link was useful for how to create a Database using MS access so that we can easily insert the data in it. By reading this page we got the idea about how to create a basic database
2. https://www.codejava.net/java-se/jdbc/java-jdbc-example-connect-to-microsoft-access-database :
   By this link we learnt how to connect to a database so that once we inserted the data in a form it can easily be stored in a database without any error
3. https://www.kashipara.com/project/java/613/download-student-management-system-java-project-source-code :
   We got detailed information and basics about Students Management System project about which modules to include in our project and what type of data it will contain. Also what advantages and limitations it will have, which software's will be required to do the project
4. https://www.sourcecodester.com/java/display-record-ms-access-database-jtable.html:
   We got the idea of how to display the data from database to a JTable. So that it will be useful for the user to search any data from database by applying certain conditions and displaying the necessary output as per required choice

4.0 ACTUAL PROCEDURE FOLLOWED:

1. The first step was to find the project over the internet and syllabus
2. Selected the appropriate topic as “Students Management System” by discussing with our group members
3. Gained information about the overall concept of the project and discussed with group members about how to develop the project and which team member will do what work
4. Created the database named as “college” and table named as “Info”
5. Each column contains the data bout a particular student and the columns are named as roll no, name, marks, address, enrollment no, city, semester and email-id
6. Used the MS Access Database to create our project database
7. Coding was done by the team member as per the task assigned to each team member
8. Each team member tested the project to check whether we are getting the expected output or not
9. Last but not the least, each and every team member contributed to create the final project documentation

5.0 ACTUAL RESOURCES USED:

Sr. No.	Name of Resource/material	Specification	Qty	Remarks
1	Microsoft Word	Version 2013	-	
2	NetBean	-	-	
3	Microsoft Access	Version 2013	-	
4	Software	Jdk 1.6.0	-	
5	Command Prompt	-	-	
6	Wikipedia	-	-	

6.0 OUTPUT OF MICRO-PROJECT:

PROGRAM CODE-

COVER PAGE:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
new cover().dispose();
new add_stud().setVisible(true);
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
new cover().dispose();
new update().setVisible(true);
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
 {
new cover().dispose();
new delete_stud().setVisible(true);
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
 {
new cover().dispose();
new view().setVisible(true);
}
```

ADD RECORD:

```
private void b2ActionPerformed(java.awt.event.ActionEvent evt)
{
 int roll=Integer.parseInt(t1.getText())
 String name=t2.getText();
 String address=t3.getText();
 int marks=Integer.parseInt(t4.getText());
 int sem=Integer.parseInt(t5.getText());
 int enroll=Integer.parseInt(t6.getText());
 int phone=Integer.parseInt(t7.getText());
 String email=t8.getText();
 String sql="UPDATE student SET
NAME=?,ADDRESS=?,MARKS=?,SEMESTER=?,ENROLL=?,PHONE=?,EMAIL=?  WHERE
ROLL=?";
Try
{
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con=DriverManager.getConnection
("jdbc:derby://localhost:1527/student","gpr","123");
PreparedStatement s=con.prepareStatement(sql);
s.setString(1, name);
s.setString(2, address);
s.setInt(3, marks);
s.setInt(4, sem);
s.setInt(5, enroll);
s.setInt(6, phone);
s.setString(7,email);
 s.setInt(8, roll);
 s.executeUpdate();
 l9.setText("Data updated");
 }
 catch (ClassNotFoundException | SQLException ex)
```

```
{
}
```

DELETE RECORD:

```
private void b1ActionPerformed(java.awt.event.ActionEvent evt)

{

int i= Integer.parseInt(t1.getText());

String sql="delete from student where ROLL=?";

Try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection
("jdbc:derby://localhost:1527/student","gpr","123");

PreparedStatement s=con.prepareStatement(sql);

s.setInt(1,i);

s.executeUpdate();

l2.setText("Record deleted");

}

catch (ClassNotFoundException | SQLException ex)
{
}
```

UPDATE RECORD:

1. **Update Button**

```
private void b1ActionPerformed(java.awt.event.ActionEvent evt)
{
int  n= Integer.parseInt(t1.getText());

String sql="select * from student where roll=?";
```

```
Try
{
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con=DriverManager.getConnection
("jdbc:derby://localhost:1527/student","gpr","123");
PreparedStatement s=con.prepareStatement(sql);
s.setInt(1, n);
ResultSet rs=s.executeQuery();
while(rs.next())
{
t2.setText(rs.getString(2));
t3.setText(rs.getString(3));
t4.setText(Integer.toString(rs.getInt(4)));
t5.setText(Integer.toString(rs.getInt(5)));
t6.setText(Integer.toString(rs.getInt(6)));
t7.setText(Integer.toString(rs.getInt(7)));
t8.setText(rs.getString(8));
}
}
catch (ClassNotFoundException | SQLException ex)
{
}
```

2. **View Button:**

```
private void b2ActionPerformed(java.awt.event.ActionEvent evt)
{
int roll=Integer.parseInt(t1.getText());
```

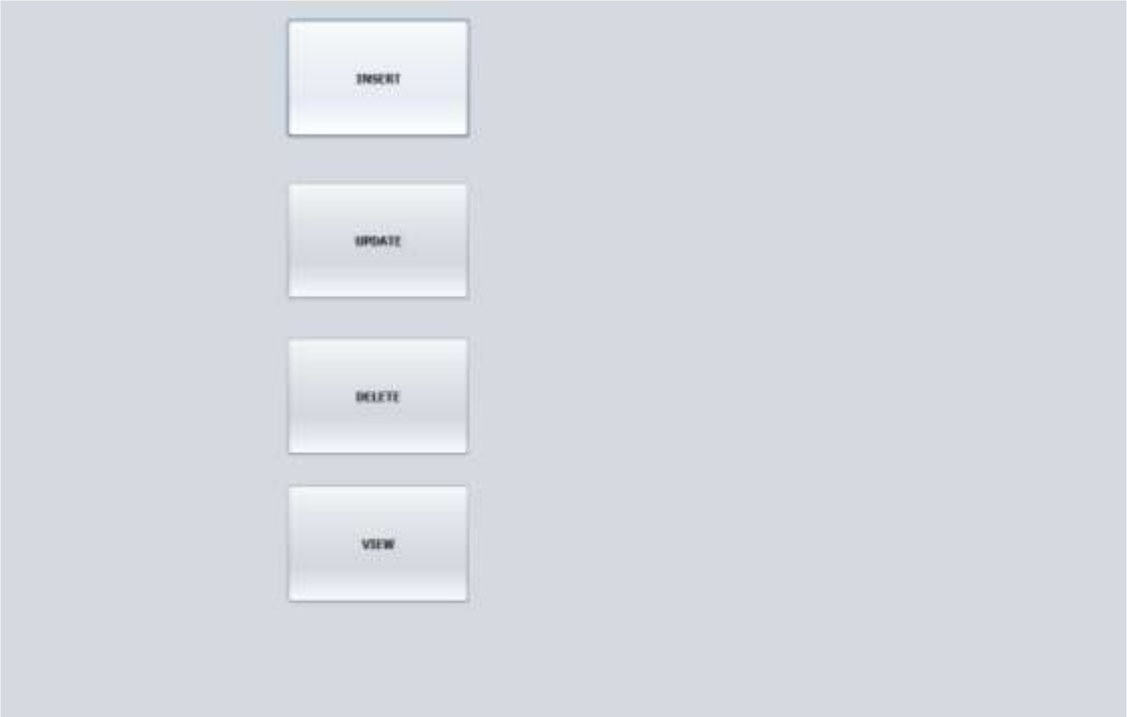
```
String name=t2.getText();

String address=t3.getText());

int marks=Integer.parseInt(t4.getText());

int sem=Integer.parseInt(t5.getText());

int enroll=Integer.parseInt(t6.getText());

int phone=Integer.parseInt(t7.getText());

String email=t8.getText();

String sql="UPDATE student SET
NAME=?,ADDRESS=?,MARKS=?,SEMESTER=?,ENROLL=?,PHONE=?,EMAIL=? WHERE
ROLL=?"

Try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection
("jdbc:derby://localhost:1527/student","gpr","123");

PreparedStatement s=con.prepareStatement(sql);

s.setString(1, name);

s.setString(2, address);

s.setInt(3, marks);

s.setInt(4, sem);

s.setInt(5, enroll);

s.setInt(6, phone);

s.setString(7,email);

s.setInt(8, roll);
```

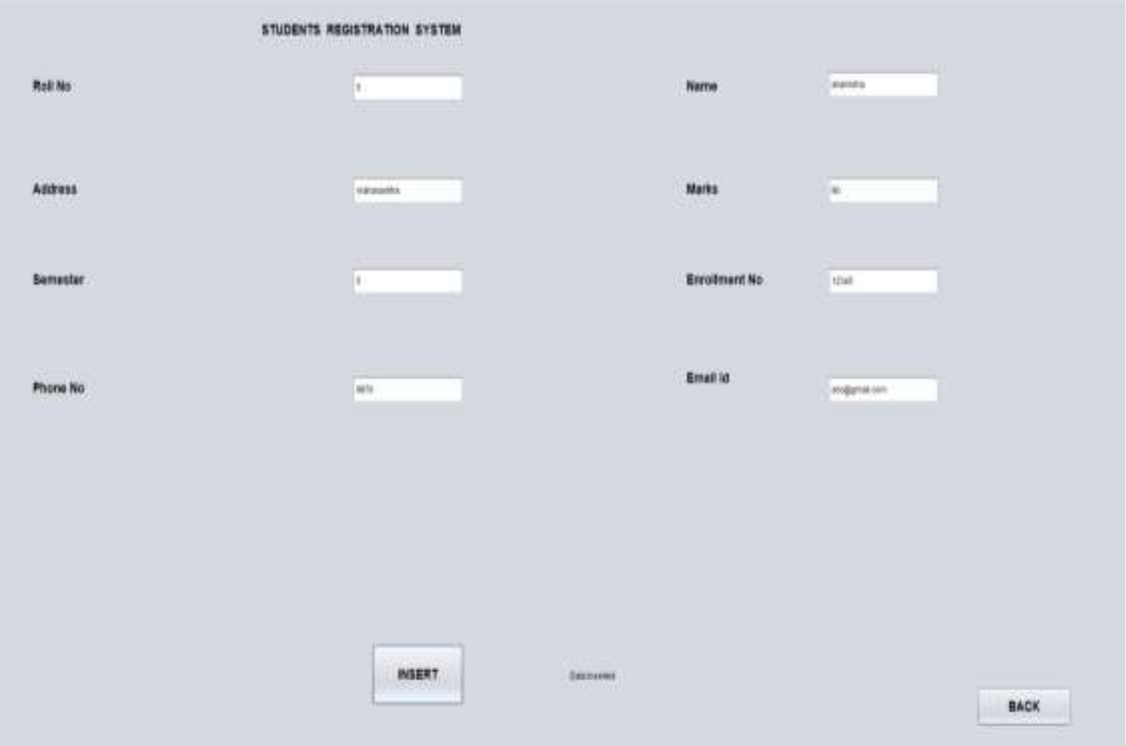
```
s.executeUpdate();

l9.setText("Data updated");

}

catch (ClassNotFoundException | SQLException ex)

{

}
```

VIEW RECORD:

```
public view()
{
initComponents();
try
{
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con=(Connection) DriverManager.getConnection
("jdbc:derby://localhost:1527/student","gpr","123");
PreparedStatement ps=con.prepareStatement("select * from student");
ResultSet rs=ps.executeQuery();
ResultSetMetaData rsmt=rs.getMetaData();
int c=rsmt.getColumnCount();
Vector col=new Vector(c);
for(int i=1;i<=c;i++)
{
col.add(rsmt.getColumnName(i));
}
Vector data = new Vector();
Vector row= new Vector();
while(rs.next())
{
row= new Vector(c);
for(int i=1;i<=c;i++)
{
row.add(rs.getString(i));
}
data.add(row);
}
JTable t=new JTable(data,col);
JScrollPane js=new JScrollPane(t);
jPanel1.setLayout(new BorderLayout());
jPanel1.add(js,BorderLayout.CENTER);
}
catch
(ClassNotFoundException | SQLException ex)
{
}
}
```

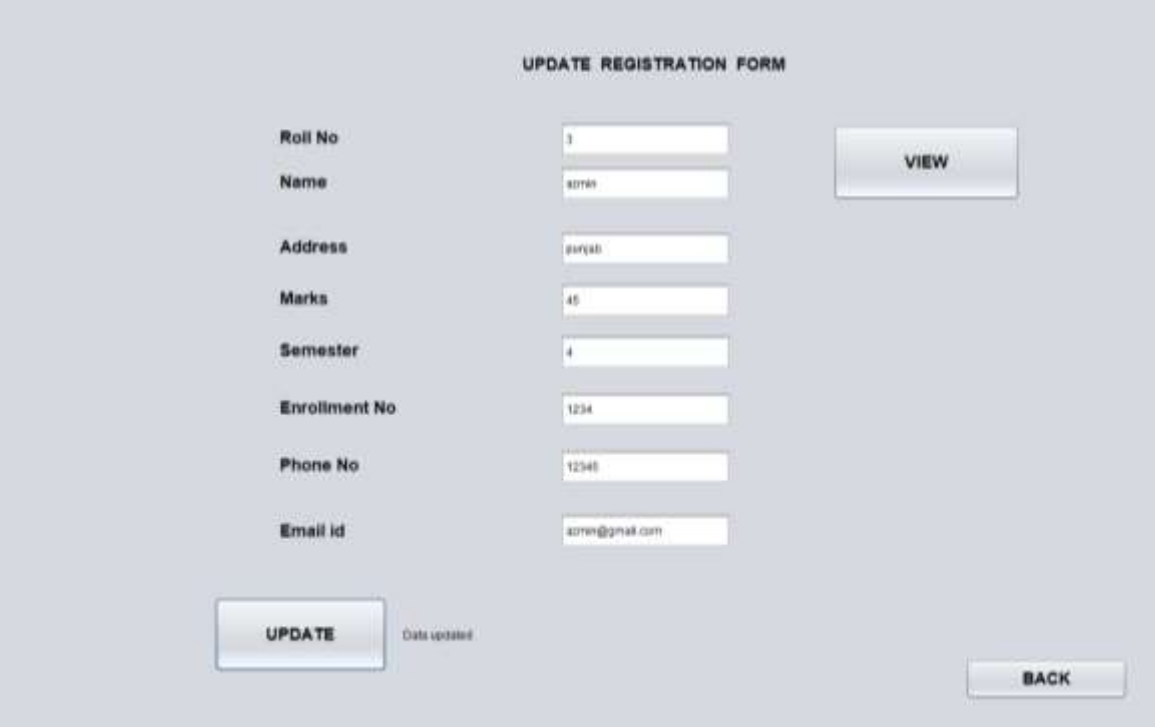
OUTPUT-





STUDENTS REGISTRATION SYSTEM

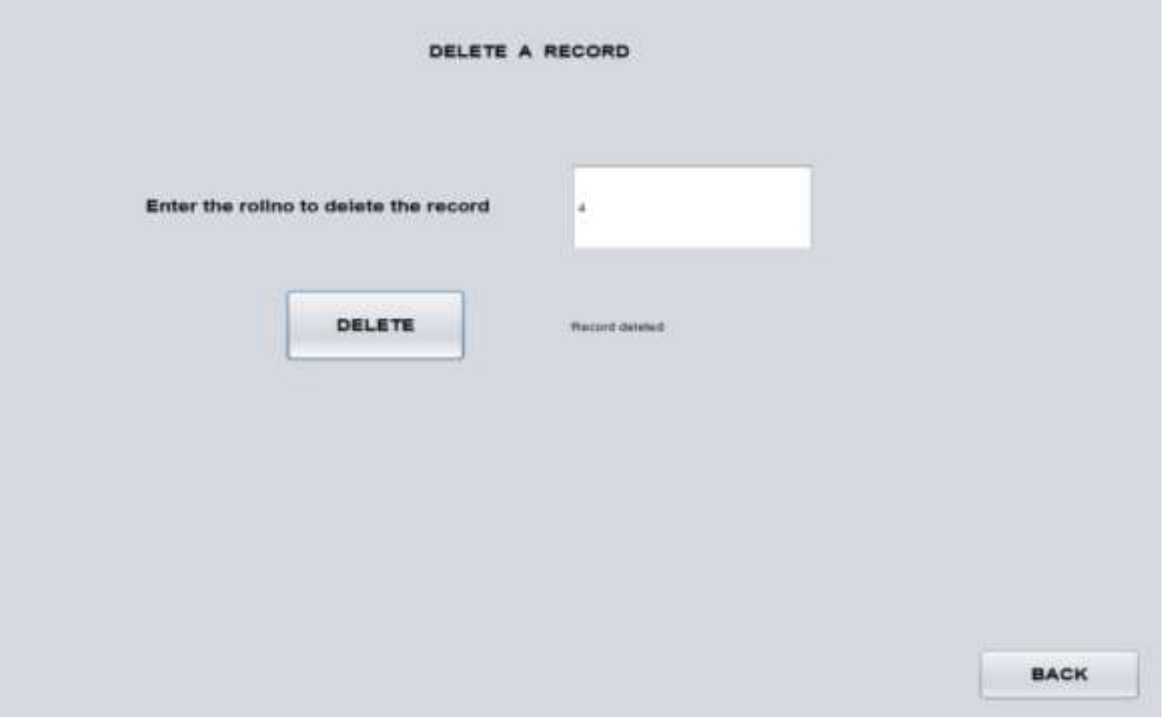
Roll No	1	Name	[illegible]
Address	maharashtra	Marks	[illegible]
Semester	1	Enrollment No	12345
Phone No	9876	Email id	abc@gmail.com



UPDATE REGISTRATION FORM

Roll No	3
Name	admin
Address	punjab
Marks	45
Semester	4
Enrollment No	1234
Phone No	12345
Email id	admin@gmail.com

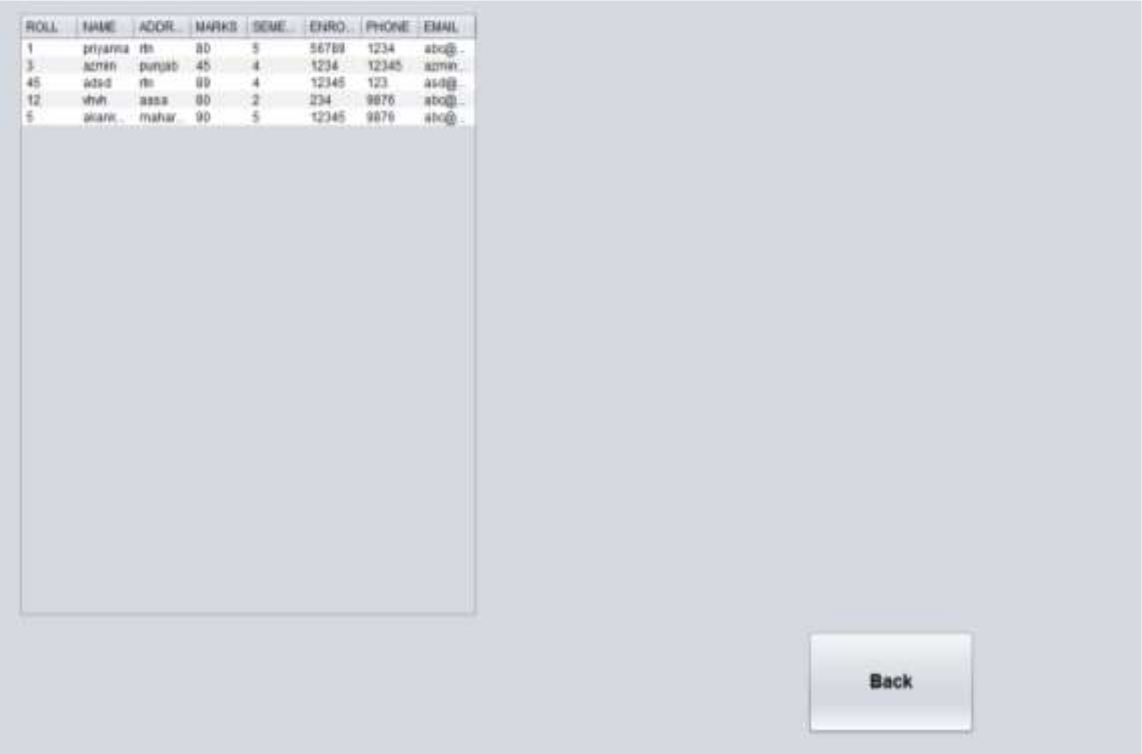
Data updated



DELETE A RECORD

Enter the rollno to delete the record	4

Record deleted



ROLL	NAME	ADDR...	MARKS	SEME...	ENRO...	PHONE	EMAIL
1	priyanka	rtn	80	5	56789	1234	abc@...
3	admin	punjab	45	4	1234	12345	admin...
45	adsd	rtn	89	4	12345	123	asd@...
12	vhvh	aasa	80	2	234	9876	abc@...
5	akank...	mahar...	90	5	12345	9876	abc@...

7.0 SKILLS DEVELOPED OUT OF THIS MICROPROJECT:

1. Implementation of database related operations in Advanced java programming
2. Implementation of basic concepts in Advanced java programming
3. Got knowledge about this type of programming language

8.0 APPLICATIONS OF THIS MICROPROJECT:

1. It can be used in college for students registration
2. One can easily search a particular data of his choice
3. Students management system allows you to keep the students records and manage them when needed
4. Students management system is a software that is used for monitoring and controlling the system in colleges
5. It features a familiar and well thought-out, an attractive user interface, combined with strong searching insertion and reporting capabilities

9.0 AREA OF FUTURE IMPROVEMENT:

The student’s registration system is the next generation address book which will provide these two basic services like portability and security. The future scope includes expand the technologies like HTML, PHP many more for improving the efficiency of the software. The project will be useful for any schools and colleges with slightly modifications. Project is flexible i.e. any changes/modifications in database in database may be performing easily and also this project could be made web enabled.