JAVA AWT BASED – UNIVERSITY DORMITORY ALLOTMENT PORTAL - SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

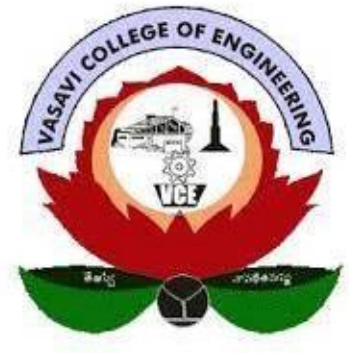
INFORMATION TECHNOLOGY

By

M.Dheeraj <1602-19-737-010>V.Siddhartha <

1602-19-737-040>

Under the guidance of Ms. S. Sreelakshmi



Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

BONAFIDE CERTIFICATE

This is to certify that this project report titled ‘University Dormitory Allotment Portal’ is a project work by Mr.M.Dheeraj and Mr.V.Siddhartha bearing rollnos. 1602-19-737-010 and 1602-19-737-040 who carried out the project under my supervision in the IV semester for the academic year 2020-2021.

Signature
Internal Examiner

Signature
External Examiner

ABSTRACT

The purpose of University Dormitory Allotment Portal is to automate the existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. University Dormitory Allotment Portal
, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records
without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.
The aim is to automate its existing manual system by the help of computerized
equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage the nuances of a University's Hospitality.

INTRODUCTION

REQUIREMENTS

TABLES	ATTRIBUTES
Student	Sid Number(10) Sname varchar(20) Roll_number Number(10) Gender varchar(10) Hostel_name varchar(20) Room_number number(10)
Department	Sid Number(10) Dept_name varchar(20) Faculty_advisor varchar(20)
Room	Room_number number(10) Hostel_name varchar(20) Alloted varchar(10)
Hostel	Hostel_id number(10) Hostel_name varchar(20) Warden varchar(20) Gender varchar(10)

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based of University Dormitory Allotment Portal which take the values related to student information, hostel information, department details etc. These values are to be updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

Software Requirement Specifications:

Operating System Front End Back End Server Documentation: Windows 10

Frontend Software: Java NetBeans 12.0: JDK 14

Backend Software: SQL Developer

Java AWT:

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

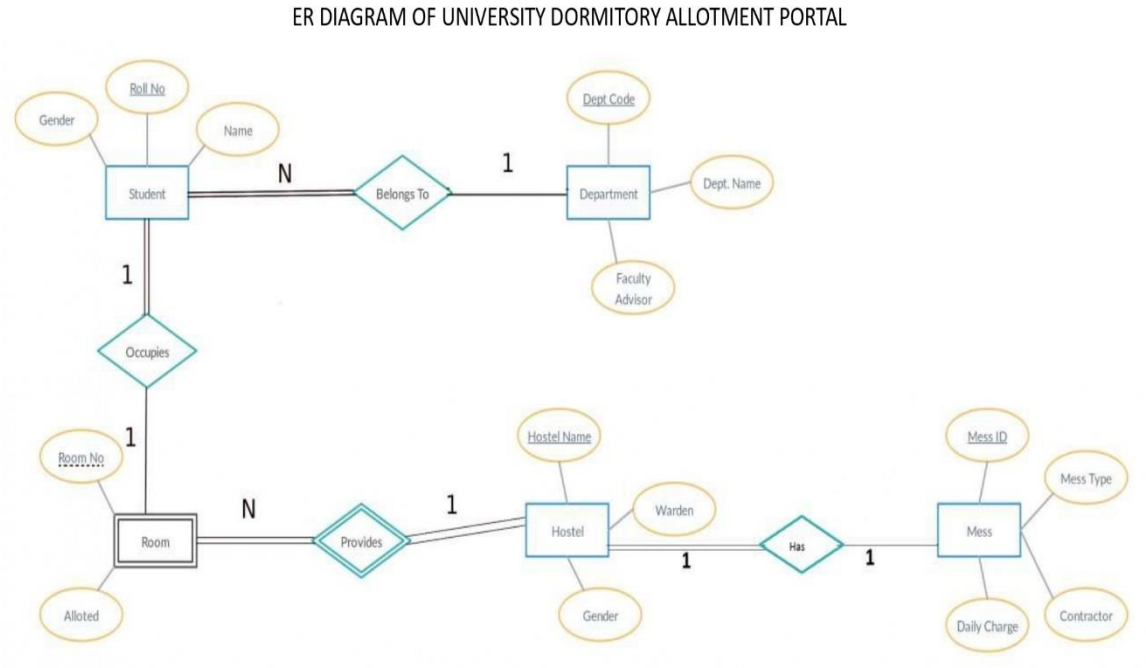
SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN

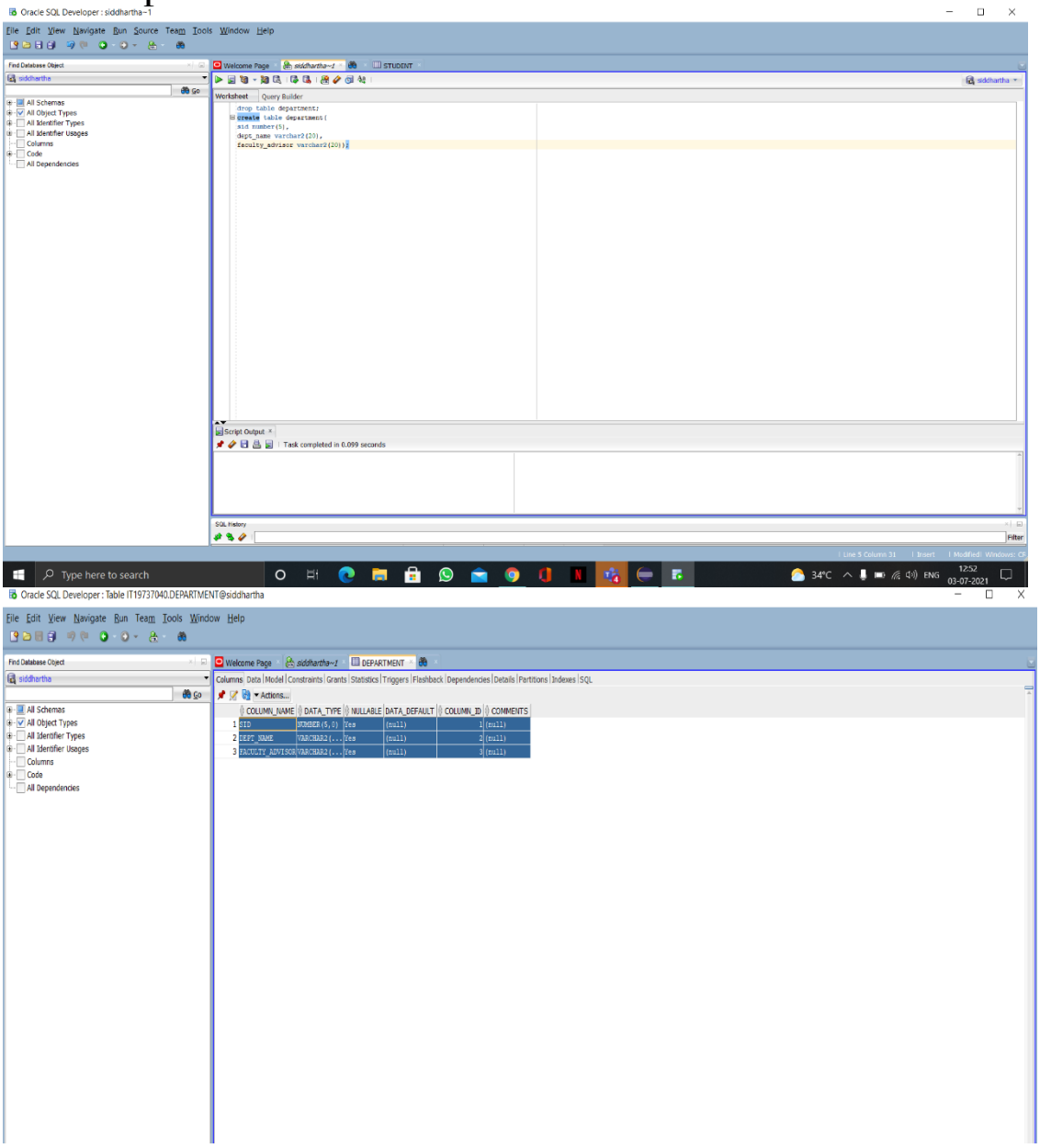
Entity relationship diagram

University Dormitory Allotment Portal

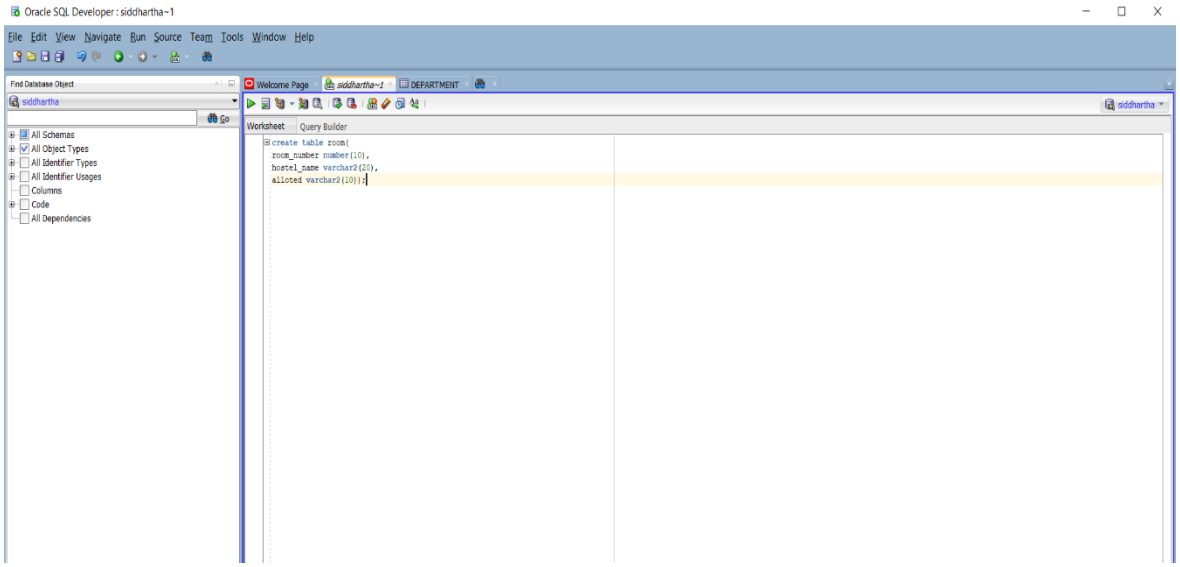


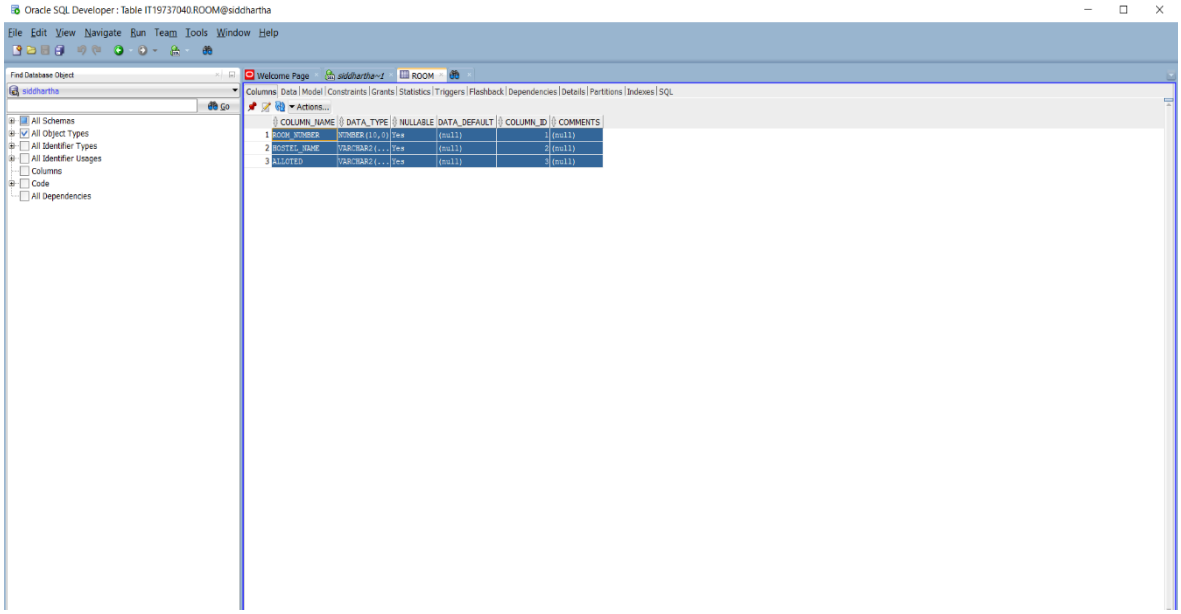
Database Design:

- Department table

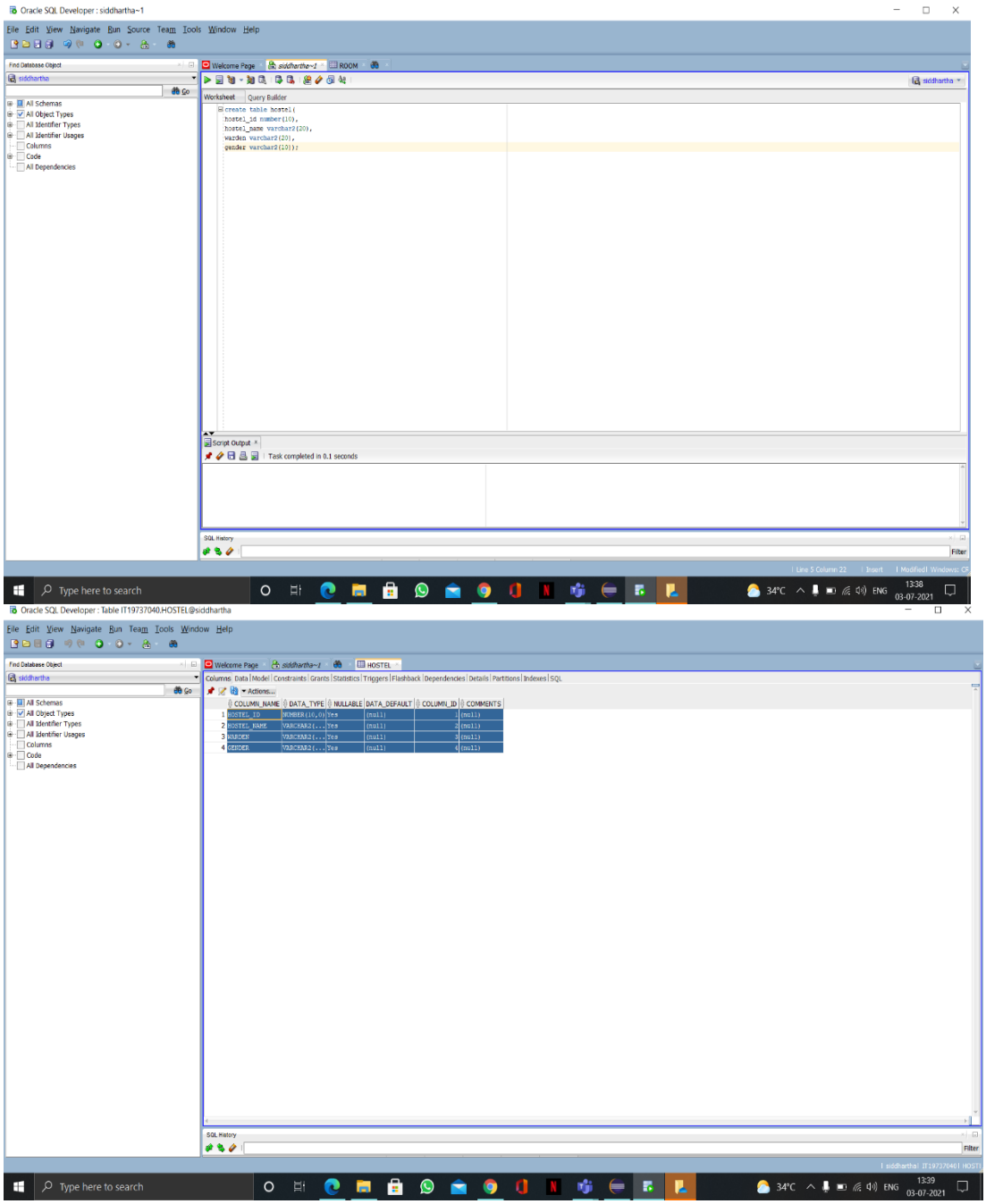


- Room table

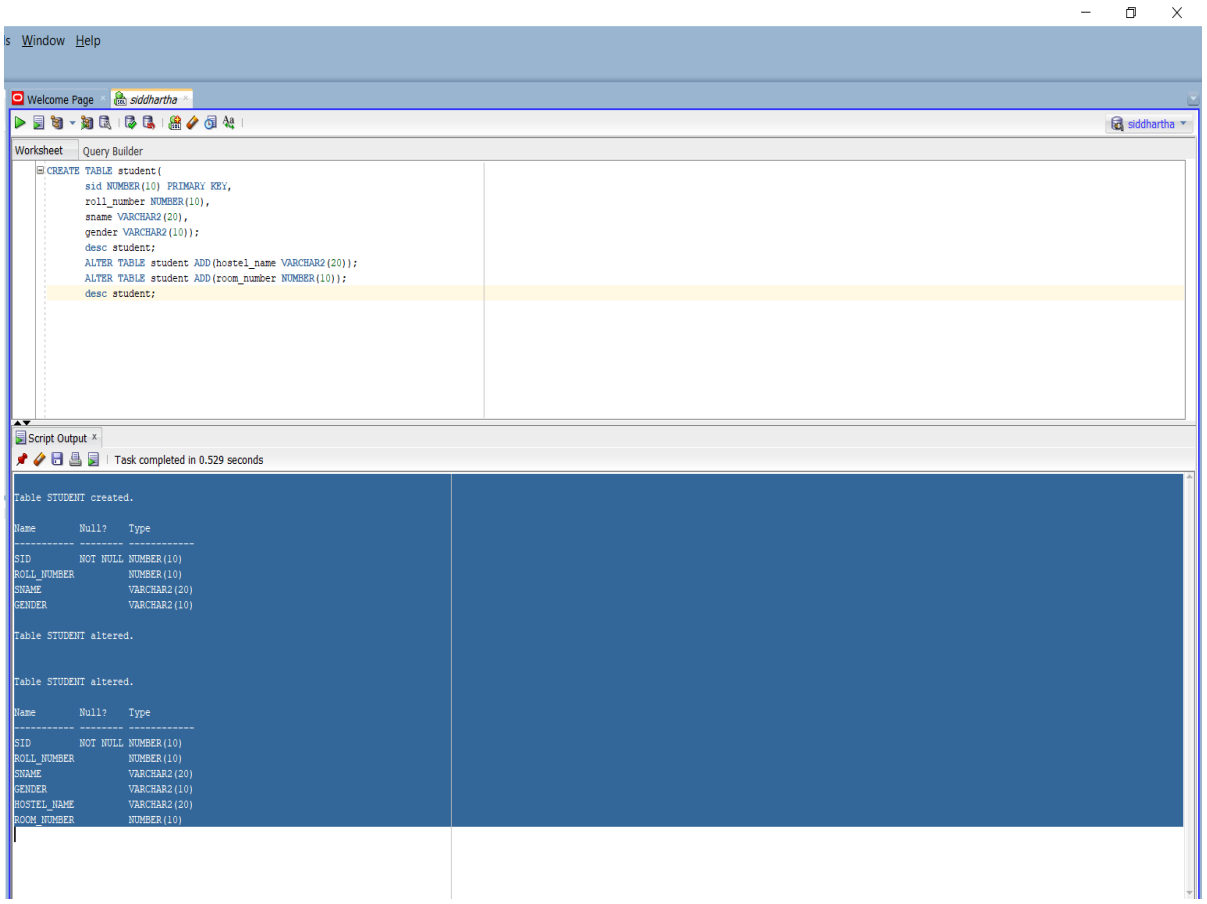

Oracle SQL Developer : siddhartha~1
File Edit View Navigate Run Source Team Tools Window Help
Welcome Page
siddhartha~1
DEPARTMENT
Worksheet
Query Builder
create table room(
room_number number(10),
hostel_name varchar2(20),
alloted varchar2(10));


Oracle SQL Developer : Table IT19737040.ROOM@siddhartha
File Edit View Navigate Run Team Tools Window Help
Welcome Page
siddhartha~1
ROOM
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
COLUMN_NAME DATA_TYPE NULLABLE DATA_DEFAULT COLUMN_ID COMMENTS
1 ROOM_NUMBER NUMBER(10,0) Yes (null) 1 (null)
2 HOSTEL_NAME VARCHAR2(... Yes (null) 2 (null)
3 ALLOTED VARCHAR2(... Yes (null) 3 (null)

➢ Hostel table



➢ Student table


Window Help
Welcome Page
siddhartha
siddhartha
Worksheet
Query Builder
CREATE TABLE student(
sid NUMBER(10) PRIMARY KEY,
roll_number NUMBER(10),
sname VARCHAR2(20),
gender VARCHAR2(10));
desc student;
ALTER TABLE student ADD(hostel_name VARCHAR2(20));
ALTER TABLE student ADD(room_number NUMBER(10));
desc student;
Script Output
Task completed in 0.529 seconds
Table STUDENT created.
Name Null? Type
SID NOT NULL NUMBER(10)
ROLL_NUMBER NUMBER(10)
SNAME VARCHAR2(20)
GENDER VARCHAR2(10)
Table STUDENT altered.
Table STUDENT altered.
Name Null? Type
SID NOT NULL NUMBER(10)
ROLL_NUMBER NUMBER(10)
SNAME VARCHAR2(20)
GENDER VARCHAR2(10)
HOSTEL_NAME VARCHAR2(20)
ROOM_NUMBER NUMBER(10)

Front end programs and its connectivity

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Normalization

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Mainframe

```
package university.management.system;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class Mainframe extends JFrame{
public static void main(String[] args) {
new Mainframe().setVisible(true);
}

public Mainframe() {
super("UNIVERSITY DORMITORY ALLOTEMENT PORTAL");
initialize();
}
private void initialize() {
setForeground(Color.CYAN);
setLayout(null);
JLabel UniversityManagementSystem = new JLabel("UNIVERSITY DORMITORY
ALLOTEMENT PORTAL");
UniversityManagementSystem.setForeground(Color.RED);
UniversityManagementSystem.setFont(new Font("Tahoma", Font.PLAIN, 45));
UniversityManagementSystem.setBounds(260, 330, 1000, 55);
add(UniversityManagementSystem);
JMenuBar menuBar = new JMenuBar();
setJMenuBar(menuBar);
JMenu student = new JMenu("STUDENT          ");
student.setForeground(Color.RED);
menuBar.add(student);

JMenuItem insertstudent = new JMenuItem("Insert");
student.add(insertstudent);

JMenuItem updatestudent = new JMenuItem("Modify");
student.add(updatestudent);

JMenuItem deletestudent = new JMenuItem("Delete");
```

```
student.add(deletestudent);
JMenu department = new JMenu("DEPARTMENT            ");
department.setForeground(Color.RED);
menuBar.add(department);
JMenuItem insertdepartment = new JMenuItem("Insert");
department.add(insertdepartment);

JMenuItem updatedepartment = new JMenuItem("Modify");
department.add(updatedepartment);

JMenuItem deletedepartment = new JMenuItem("Delete");
department.add(deletedepartment);


JMenu room = new JMenu("ROOM          ");
room.setForeground(Color.RED);
menuBar.add(room);

JMenuItem insertroom = new JMenuItem("Insert");
room.add(insertroom);

JMenuItem updateroom = new JMenuItem("Modify");
room.add(updateroom);

JMenuItem deleteroom = new JMenuItem("Delete");
room.add(deleteroom);
JMenu hostel = new JMenu("HOSTEL    ");
hostel.setForeground(Color.RED);
menuBar.add(hostel);

JMenuItem inserthostel = new JMenuItem("Insert");
hostel.add(inserthostel);

JMenuItem updatehostel = new JMenuItem("Modify");
hostel.add(updatehostel);

JMenuItem deletehostel = new JMenuItem("Delete");
```

```
hostel.add(deletehostel);
insertstudent.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new insertstudent();
} catch (Exception e) {e.printStackTrace();
}
}
});

updatestudent.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new updatestudent();
} catch (Exception e) {
e.printStackTrace();
}
}
});
deletestudent.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
new deletestudent();
}
});

insertdepartment.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new insertdepartment();
} catch (Exception e) {
e.printStackTrace();
}
}
});

updatedepartment.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
```

```
try {
new updatedepartment();
} catch (Exception e) {
e.printStackTrace();
}_
}
});


deletedepartment.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
new deletedepartment();
}});

insertroom.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new insertroom();
} catch (Exception e) {
e.printStackTrace();
}_
}
});

updateroom.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new updateroom();
} catch (Exception e) {
e.printStackTrace();
}_
}
});


deleteroom.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
```

```
new deleteroom();
}
});

inserthostel.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new inserthostel();
} catch (Exception e) {
e.printStackTrace();
}
}
});

updatehostel.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
try {
new updatehostel();
} catch (Exception e) {
e.printStackTrace();
}}
});



deletehostel.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent ae){
new deletehostel();
}
});
setSize(700,500);
setLocation(285,100);
setVisible(true);
}
}
```

INSERT STUDENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class insertstudent extends Frame
{
Button insertstudentButton;
TextField sid,sname,roll_number,gender,hostel_name,room_number;
TextArea errorText;
Connection connection;
Statement statement;
public insertstudent()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737040","v
asavi");

statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
public void buildGUI()
{
```

```
insertstudentButton = new Button("Insert");
insertstudentButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

String query= "INSERT INTO student VALUES("+"'" +sid.getText() + "'," +
sname.getText() + "'," + roll_number.getText() + "'," +gender.getText() + "',"
+hostel_name.getText()+ "'," +room_number.getText() + "'" +")";
int i = statement.executeUpdate(query);
errorText.append("\nInserted " + i + " rows successfully");
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
sid = new TextField(15);
sname = new TextField(15);
roll_number = new TextField(15);
gender = new TextField(15);
hostel_name = new TextField(15);
room_number = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(6, 2));
first.add(new Label("Sid : "));
first.add(sid);
first.add(new Label("Name : "));
first.add(sname);
first.add(new Label("Roll Number: "));
first.add(roll_number);
first.add(new Label("Gender : "));
first.add(gender);
first.add(new Label("hostel name: "));
first.add(hostel_name);
first.add(new Label("Room Number: "));
first.add(room_number);
```

```
first.setBounds(125,90,200,100);
Panel second = new Panel(new GridLayout(6, 1));
second.add(insertstudentButton);
second.setBounds(125,220,150,100);
Panel third = new Panel();
third.add(errorText);
third.setBounds(125,320,300,200);
setLayout(null);
add(first);
add(second);
add(third);
setTitle("Insert Gets");
setSize(500, 600);
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}

public static void main(String[] args)
{
insertstudent i = new insertstudent();
i.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
i.buildGUI();
}
}
```

UPDATE STUDENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updatestudent extends Frame
{
```

```
Button updatestudentButton;
List sidList;
TextField
sidText,snameText,roll_numberText,genderText,hostel_nameText,room_numberText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public updatestudent()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loadstudent()
{
try
{
```

```
rs = statement.executeQuery("SELECT sid FROM student");
while (rs.next())
{
sidList.add(rs.getString("sid"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
sidList = new List(10);
loadstudent();
add(sidList);
sidList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM student where sid
="'+sidList.getSelectedItem()+"'");
rs.next();
sidText.setText(rs.getString("sid"));
snameText.setText(rs.getString("sname"));
roll_numberText.setText(rs.getString("roll_number"));
genderText.setText(rs.getString("gender"));
hostel_nameText.setText(rs.getString("hostel_name"));
room_numberText.setText(rs.getString("room_number"));
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
updatestudentButton = new Button("Update");
updatestudentButton.addActionListener(new ActionListener()
```

```
{
public void actionPerformed(ActionEvent e)
{
try
{

Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("UPDATE student "+ "SET sname='" +
snameText.getText() + "', "+ "roll_number='" + roll_numberText.getText() + "', "+
"gender ='"+ genderText.getText()+"',"+"hostel_name='"
+hostel_nameText.getText()+"',"+"room_number='" +room_numberText.getText() + "'
WHERE sid= '"+ sidList.getSelectedItem()+"'");
errorText.append("\nUpdated " + i + " rows successfully");
sidList.removeAll();
loadstudent();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
sidText = new TextField(15);
sidText.setEditable(false);
snameText = new TextField(15);
roll_numberText = new TextField(15);
genderText = new TextField(15);
hostel_nameText = new TextField(15);
room_numberText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(6, 2));
first.add(new Label("Sid : "));
first.add(sidText);
first.add(new Label("Name : "));
first.add(snameText);
```

```
first.add(new Label("Roll Number: "));
first.add(roll_numberText);
first.add(new Label("Gender : "));
first.add(genderText);
first.add(new Label("hostel name: "));
first.add(hostel_nameText);
first.add(new Label("Room Number: "));
first.add(room_numberText);
Panel second = new Panel(new GridLayout(6, 1));
second.add(updatestudentButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Update student ");
setSize(500, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
updatestudent ups = new updatestudent();
ups.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
ups.buildGUI();
}
}
```

DELETE STUDENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deletestudent extends Frame
{
Button deletestudentButton;
List sidList;
TextField
sidText,snameText,roll_numberText,genderText,hostel_nameText,room_numberText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deletestudent()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
```

```
System.out.println(e);
}
}
private void loadstudent()
{
try
{
rs = statement.executeQuery("SELECT * FROM student");
while (rs.next())
{
sidList.add(rs.getString("sid"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
sidList = new List(10);
loadstudent();
add(sidList);
sidList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM student where sid
='"+sidList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("sid").equals(sidList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
sidText.setText(rs.getString("sid"));
```

```
snameText.setText(rs.getString("sname"));
roll_numberText.setText(rs.getString("roll_number"));
genderText.setText(rs.getString("gender"));
hostel_nameText.setText(rs.getString("hostel_name"));
room_numberText.setText(rs.getString("room_number"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deletestudentButton = new Button("Delete student");
deletestudentButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con=
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM student WHERE sid =
'"+sidList.getSelectedItem()+"' and sname='"+snameText.getText()+"' and
roll_number='"+roll_numberText.getText()+"' and gender ='"+genderText.getText()+"'
and  hostel_name='"+hostel_nameText.getText()+"' and
room_number='"+room_numberText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
snameText.setText(null);
roll_numberText.setText(null);
genderText.setText(null);
hostel_nameText.setText(null);
room_numberText.setText(null);
sidList.removeAll();
loadstudent();
}
catch (SQLException insertException)
{
```

```
displaySQLErrors(insertException);
}
}
});
sidText = new TextField(15);
snameText = new TextField(15);
roll_numberText = new TextField(15);
genderText = new TextField(15);
hostel_nameText = new TextField(15);
room_numberText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(6, 2));
first.add(new Label("Sid : "));
first.add(sidText);
first.add(new Label("Name : "));
first.add(snameText);
first.add(new Label("Roll Number: "));
first.add(roll_numberText);
first.add(new Label("Gender : "));
first.add(genderText);
first.add(new Label("hostel name: "));
first.add(hostel_nameText);
first.add(new Label("Room Number: "));
first.add(room_numberText);
Panel second = new Panel(new GridLayout(6, 1));
second.add(deletestudentButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  student ");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
```

```
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
deletestudent del = new deletestudent();
del.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
del.buildGUI();
}
}
```

INSERT DEPARTMENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class insertdepartment extends Frame
{
Button insertdepartmentButton;
TextField sid,dept_name,faculty_advisor;
TextArea errorText;
Connection connection;
Statement statement;
public insertdepartment()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
```

```
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");

statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
public void buildGUI()
{

insertdepartmentButton = new Button("Insert");
insertdepartmentButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

String query= "INSERT INTO department VALUES("+"'" +sid.getText() + "'," +
dept_name.getText() + "'," + faculty_advisor.getText()  + "'" +")";
int i = statement.executeUpdate(query);
errorText.append("\nInserted " + i + " rows successfully");
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
```

```
});
sid = new TextField(15);
dept_name = new TextField(15);
faculty_advisor = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("Sid : "));
first.add(sid);
first.add(new Label("dept_name : "));
first.add(dept_name);
first.add(new Label("faculty_advisor: "));
first.add(faculty_advisor);

first.setBounds(125,90,200,100);
Panel second = new Panel(new GridLayout(3, 1));
second.add(insertdepartmentButton);
second.setBounds(125,220,150,100);
Panel third = new Panel();
third.add(errorText);
third.setBounds(125,320,300,200);
setLayout(null);
add(first);
add(second);
add(third);
setTitle("Insert Gets");
setSize(500, 600);
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}

public static void main(String[] args)
{
insertdepartment i = new insertdepartment();
```

```
i.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
i.buildGUI();
}
}
```

UPDATE DEPARTMENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updatedepartment extends Frame
{
Button updatedepartmentButton;
List sidList;
TextField sidText,dept_nameText,faculty_advisorText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public updatedepartment()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
```

```
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loaddepartment()
{
try
{
rs = statement.executeQuery("SELECT sid FROM department");
while (rs.next())
{
sidList.add(rs.getString("sid"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
sidList = new List(10);
loaddepartment();
add(sidList);
sidList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM department where sid
='"+sidList.getSelectedItem()+"'");
```

```
rs.next();
sidText.setText(rs.getString("sid"));
dept_nameText.setText(rs.getString("dept_name"));
faculty_advisorText.setText(rs.getString("faculty_advisor"));
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
updatedepartmentButton = new Button("Update");
updatedepartmentButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("UPDATE department "+ "SET dept_name='" +
dept_nameText.getText() + "', "+ "faculty_advisor='" + faculty_advisorText.getText()
+ "' WHERE sid= '"+ sidList.getSelectedItem()+"'");
errorText.append("\nUpdated " + i + " rows successfully");
sidList.removeAll();
loaddepartment();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
sidText = new TextField(15);
sidText.setEditable(false);
dept_nameText = new TextField(15);
faculty_advisorText = new TextField(15);
```

```
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("Sid : "));
first.add(sidText);
first.add(new Label("dept_name : "));
first.add(dept_nameText);
first.add(new Label("faculty_advisor: "));
first.add(faculty_advisorText);
Panel second = new Panel(new GridLayout(3, 1));
second.add(updatedepartmentButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Update department ");
setSize(500, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
updatedepartment ups = new updatedepartment();
ups.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
ups.buildGUI();
}
}
```

DELETE DEPARTMENT

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deletedepartment extends Frame
{
Button deletedepartmentButton;
List sidList;
TextField sidText,dept_nameText,faculty_advisorText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deletedepartment()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
```

```
System.out.println(e);
}
}
private void loaddepartment()
{
try
{
rs = statement.executeQuery("SELECT * FROM department");
while (rs.next())
{
sidList.add(rs.getString("sid"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
sidList = new List(10);
loaddepartment();
add(sidList);
sidList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM department where sid
='"+sidList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("sid").equals(sidList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
sidText.setText(rs.getString("sid"));
```

```
dept_nameText.setText(rs.getString("dept_name"));
faculty_advisorText.setText(rs.getString("faculty_advisor"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deletedepartmentButton = new Button("Delete department");
deletedepartmentButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con=
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM department WHERE sid =
'"+sidList.getSelectedItem()+"' and dept_name='"+dept_nameText.getText()+"' and
faculty_advisor='"+faculty_advisorText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
dept_nameText.setText(null);
faculty_advisorText.setText(null);
sidList.removeAll();
loaddepartment();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
sidText = new TextField(15);
dept_nameText = new TextField(15);
faculty_advisorText = new TextField(15);
errorText = new TextArea(10, 40);
```

```
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("Sid : "));
first.add(sidText);
first.add(new Label("dept_name : "));
first.add(dept_nameText);
first.add(new Label("faculty_advisor: "));
first.add(faculty_advisorText);
Panel second = new Panel(new GridLayout(3, 1));
second.add(deletedepartmentButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  department ");
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
deletedepartment del = new deletedepartment();
del.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
del.buildGUI();
}
}
```

INSERT ROOM

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class insertroom extends Frame
{
Button insertroomButton;
TextField room_number,hostel_name,alloted;
TextArea errorText;
Connection connection;
Statement statement;
public insertroom()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");

statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
```

```
}
public void buildGUI()
{

insertroomButton = new Button("Insert");
insertroomButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

String query= "INSERT INTO room VALUES("+"'" +room_number.getText() + "','"
+hostel_name.getText() + "','" + alloted.getText()  + "'" +")";
int i = statement.executeUpdate(query);
errorText.append("\nInserted " + i + " rows successfully");
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
room_number = new TextField(15);
hostel_name = new TextField(15);
alloted = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("room_number : "));
first.add(room_number);
first.add(new Label("hostel_name : "));
first.add(hostel_name);
first.add(new Label("alloted: "));
first.add(alloted);

first.setBounds(125,90,200,100);
Panel second = new Panel(new GridLayout(3, 1));
second.add(insertroomButton);
```

```
second.setBounds(125,220,150,100);
Panel third = new Panel();
third.add(errorText);
third.setBounds(125,320,300,200);
setLayout(null);
add(first);
add(second);
add(third);
setTitle("Insert Gets");
setSize(500, 600);
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}

public static void main(String[] args)
{
insertroom i = new insertroom();
i.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
i.buildGUI();
}
}
```

UPDATE ROOM

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updateroom extends Frame
{
Button updateroomButton;
List room_numberList;
```

```
TextField room_numberText,hostel_nameText,allotedText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public updateroom()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loadroom()
{
try
{
rs = statement.executeQuery("SELECT room_number FROM room");
while (rs.next())
{
```

```
room_numberList.add(rs.getString("room_number"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
room_numberList = new List(10);
loadroom();
add(room_numberList);
room_numberList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM room where room_number
="'+room_numberList.getSelectedItem()+"'");
rs.next();
room_numberText.setText(rs.getString("room_number"));
hostel_nameText.setText(rs.getString("hostel_name"));
allotedText.setText(rs.getString("alloted"));
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
updateroomButton = new Button("Update");
updateroomButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
```

```
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("UPDATE room "+ "SET hostel_name='"
+hostel_nameText.getText() + "', "+ "alloted='" + allotedText.getText()  + "' WHERE
room_number= '"+ room_numberList.getSelectedItem()+"'");
errorText.append("\nUpdated " + i + " rows successfully");
room_numberList.removeAll();
loadroom();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
room_numberText = new TextField(15);
room_numberText.setEditable(false);
room_numberText = new TextField(15);
hostel_nameText = new TextField(15);
allotedText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("room_number : "));
first.add(room_numberText);
first.add(new Label("hostel_name : "));
first.add(hostel_nameText);
first.add(new Label("alloted: "));
first.add(allotedText);
Panel second = new Panel(new GridLayout(3, 1));
second.add(updateroomButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Update room ");
```

```
setSize(500, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
updateroom ups = new updateroom();
ups.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
ups.buildGUI();
}
}
```

DELETE ROOM

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deleteroom extends Frame
{
Button deleteroomButton;
List room_numberList;
TextField room_numberText,hostel_nameText,allotedText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deleteroom()
{
try
```

```
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loadroom()
{
try
{
rs = statement.executeQuery("SELECT * FROM room");
while (rs.next())
{
room_numberList.add(rs.getString("room_number"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
```

```
public void buildGUI()
{
room_numberList = new List(10);
loadroom();
add(room_numberList);
room_numberList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM room where room_number
="'+room_numberList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("room_number").equals(room_numberList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
room_numberText.setText(rs.getString("room_number"));
hostel_nameText.setText(rs.getString("hostel_name"));
allotedText.setText(rs.getString("alloted"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deleteroomButton = new Button("Delete room");
deleteroomButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con=
```

```
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it19737040","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM room WHERE room_number = '"+room_numberList.getSelectedItem()+"' and hostel_name='"+hostel_nameText.getText()+"' and alloted='"+allotedText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
hostel_nameText.setText(null);
allotedText.setText(null);
room_numberList.removeAll();
loadroom();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
room_numberText = new TextField(15);
hostel_nameText = new TextField(15);
allotedText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(3, 2));
first.add(new Label("room_number : "));
first.add(room_numberText);
first.add(new Label("hostel_name : "));
first.add(hostel_nameText);
first.add(new Label("alloted: "));
first.add(allotedText);
Panel second = new Panel(new GridLayout(3, 1));
second.add(deleteroomButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  room ");
```

```
setSize(450, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
deleteroom del = new deleteroom();
del.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
del.buildGUI();
}
}
```

INSERT HOSTEL

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class inserthostel extends Frame
{
Button inserthostelButton;
TextField hostel_id,hostel_name,warden,gender;
TextArea errorText;
Connection connection;
Statement statement;
public inserthostel()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");

statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
public void buildGUI()
{

inserthostelButton = new Button("Insert");
inserthostelButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

String query= "INSERT INTO hostel VALUES("+"'" +hostel_id.getText() + "'," +
hostel_name.getText() + "'," + warden.getText() + "'," +gender.getText() + "'" +")";
int i = statement.executeUpdate(query);
errorText.append("\nInserted " + i + " rows successfully");
}
```

```
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
hostel_id = new TextField(15);
hostel_name = new TextField(15);
warden = new TextField(15);
gender = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("hostel_id : "));
first.add(hostel_id);
first.add(new Label("hostel_name : "));
first.add(hostel_name);
first.add(new Label("warden: "));
first.add(warden);
first.add(new Label("Gender : "));
first.add(gender);
first.setBounds(125,90,200,100);
Panel second = new Panel(new GridLayout(4, 1));
second.add(inserthostelButton);
second.setBounds(125,220,150,100);
Panel third = new Panel();
third.add(errorText);
third.setBounds(125,320,300,200);
setLayout(null);
add(first);
add(second);
add(third);
setTitle("Insert Gets");
setSize(500, 600);
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
```

```
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}

public static void main(String[] args)
{
inserthostel i = new inserthostel();
i.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
i.buildGUI();
}
}
```

UPDATE HOSTEL

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class updatehostel extends Frame
{
Button updatehostelButton;
List hostel_idList;
TextField hostel_idText,hostel_nameText,wardenText,genderText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public updatehostel()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
```

```
System.err.println("Unable to find and load driver");
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loadhostel()
{
try
{
rs = statement.executeQuery("SELECT hostel_id FROM hostel");
while (rs.next())
{
hostel_idList.add(rs.getString("hostel_id"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
hostel_idList = new List(10);
loadhostel();
add(hostel_idList);
```

```
hostel_idList.addItemListener(new ItemListener()
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM hostel where hostel_id
="'+hostel_idList.getSelectedItem()+"'");
rs.next();
hostel_idText.setText(rs.getString("hostel_id"));
hostel_nameText.setText(rs.getString("hostel_name"));
wardenText.setText(rs.getString("warden"));
genderText.setText(rs.getString("gender"));

}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
updatehostelButton = new Button("Update");
updatehostelButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{

Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("UPDATE hostel "+ "SET hostel_name='" +
hostel_nameText.getText() + "', "+ "warden='" + wardenText.getText() + "', "+ "gender
='"+ genderText.getText() + "' WHERE hostel_id= '"+
hostel_idList.getSelectedItem()+"'");
errorText.append("\nUpdated " + i + " rows successfully");
hostel_idList.removeAll();
loadhostel();
```

```
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
hostel_idText = new TextField(15);
hostel_idText.setEditable(false);
hostel_nameText = new TextField(15);
wardenText = new TextField(15);
genderText = new TextField(15);

errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("hostel_id : "));
first.add(hostel_idText);
first.add(new Label("hostel_name : "));
first.add(hostel_nameText);
first.add(new Label("warden: "));
first.add(wardenText);
first.add(new Label("Gender : "));
first.add(genderText);
Panel second = new Panel(new GridLayout(4, 1));
second.add(updatehostelButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Update hostel ");
setSize(500, 600);
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
```

```
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
updatehostel ups = new updatehostel();
ups.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
ups.buildGUI();
}
}
```

DELETE HOSTEL

```
package university.management.system;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class deletehostel extends Frame
{
Button deletehostelButton;
List hostel_idList;
TextField hostel_idText,hostel_nameText,wardenText,genderText;
TextArea errorText;
Connection connection;
Statement statement;
ResultSet rs;
public deletehostel()
{
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e)
{
System.err.println("Unable to find and load driver");
```

```
System.exit(1);
}
connectToDB();
}
public void connectToDB()
{
try
{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
statement = con.createStatement();
System.out.println("connected");
}
catch (SQLException e)
{
System.out.println(e);
}
}
private void loadhostel()
{
try
{
rs = statement.executeQuery("SELECT * FROM hostel");
while (rs.next())
{
hostel_idList.add(rs.getString("hostel_id"));
}
}
catch (SQLException e)
{
displaySQLErrors(e);
}
}
public void buildGUI()
{
hostel_idList = new List(10);
loadhostel();
add(hostel_idList);
hostel_idList.addItemListener(new ItemListener()
```

```
{
public void itemStateChanged(ItemEvent e)
{
try
{
rs = statement.executeQuery("SELECT * FROM hostel where hostel_id
="'+hostel_idList.getSelectedItem()+"'");
while (rs.next())
{
if
(rs.getString("hostel_id").equals(hostel_idList.getSelectedItem()))
break;
}
if (!rs.isAfterLast())
{
hostel_idText.setText(rs.getString("hostel_id"));
hostel_nameText.setText(rs.getString("hostel_name"));
wardenText.setText(rs.getString("warden"));
genderText.setText(rs.getString("gender"));
}
}
catch (SQLException selectException)
{
displaySQLErrors(selectException);
}
}
});
deletehostelButton = new Button("Delete hostel");
deletehostelButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
Connection con=
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it197370
40","vasavi");
Statement statement1 = con.createStatement();
int i = statement1.executeUpdate("DELETE FROM hostel WHERE hostel_id =
"'+hostel_idList.getSelectedItem()+"' and
```

```
hostel_name='"+hostel_nameText.getText()+"' and
warden='"+wardenText.getText()+"' and gender ='"+genderText.getText()+"'");
errorText.append("\nDeleted " + i + " rows successfully");
hostel_nameText.setText(null);
wardenText.setText(null);
genderText.setText(null);
hostel_idList.removeAll();
loadhostel();
}
catch (SQLException insertException)
{
displaySQLErrors(insertException);
}
}
});
hostel_idText = new TextField(15);
hostel_nameText = new TextField(15);
wardenText = new TextField(15);
genderText = new TextField(15);
errorText = new TextArea(10, 40);
errorText.setEditable(false);
Panel first = new Panel();
first.setLayout(new GridLayout(4, 2));
first.add(new Label("hostel_id : "));
first.add(hostel_idText);
first.add(new Label("hostel_name : "));
first.add(hostel_nameText);
first.add(new Label("warden: "));
first.add(wardenText);
first.add(new Label("Gender : "));
first.add(genderText);
Panel second = new Panel(new GridLayout(4, 1));
second.add(deletehostelButton);
Panel third = new Panel();
third.add(errorText);
add(first);
add(second);
add(third);
setTitle("Remove  hostel ");
setSize(450, 600);
```

```
setLayout(new FlowLayout());
setVisible(true);
}
private void displaySQLErrors(SQLException e)
{
errorText.append("\nSQLException: " + e.getMessage() + "\n");
errorText.append("SQLState: " + e.getSQLState() + "\n");
errorText.append("VendorError: " + e.getErrorCode() + "\n");
}
public static void main(String[] args)
{
deletehostel del = new deletehostel();
del.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
del.buildGUI();
}
}
```

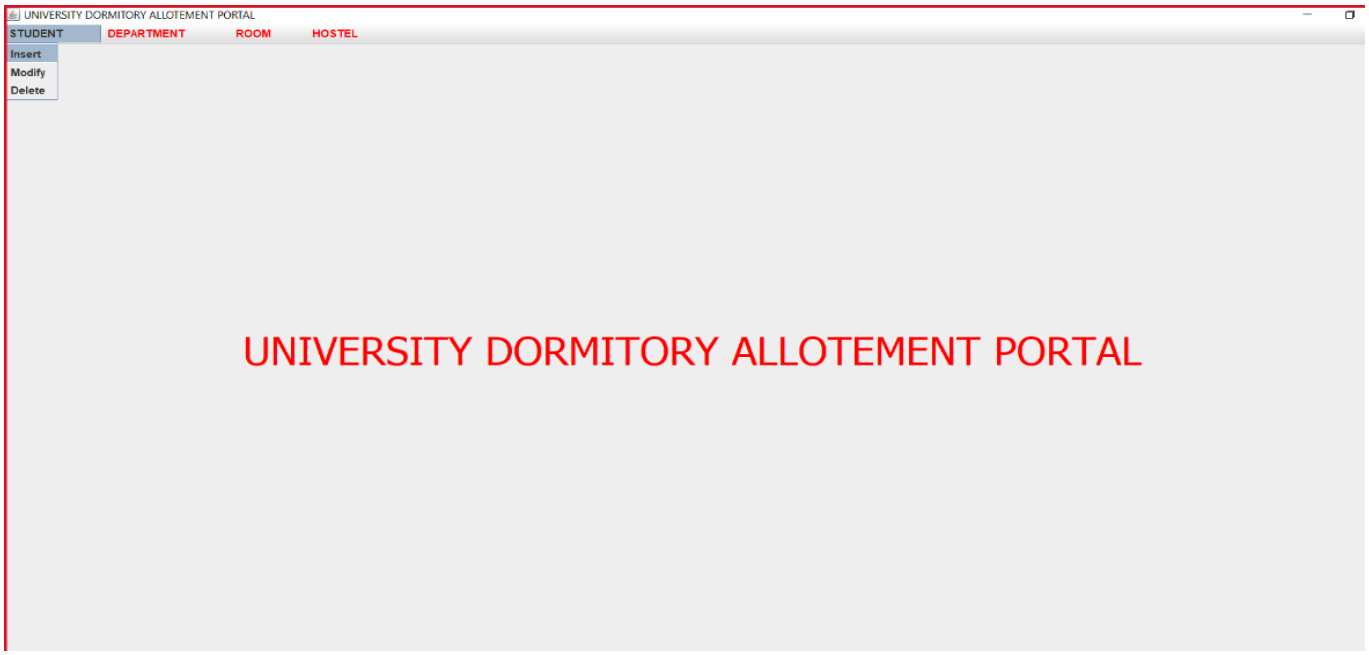
Github links and folder structure:

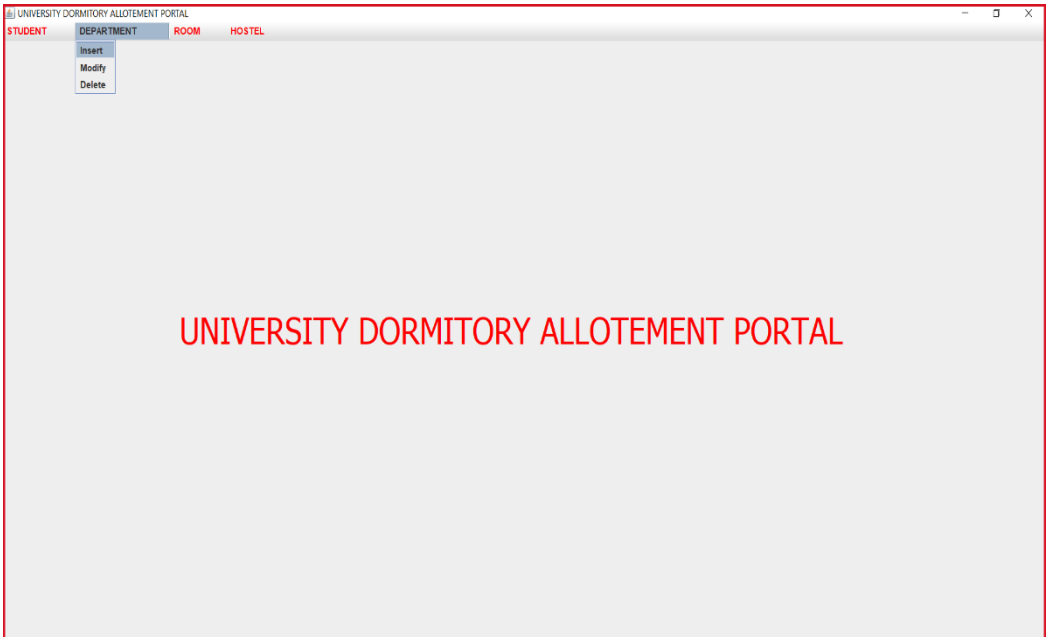
Link: https://github.com/mukiralad/University-Dormitory-Allotment-Portal

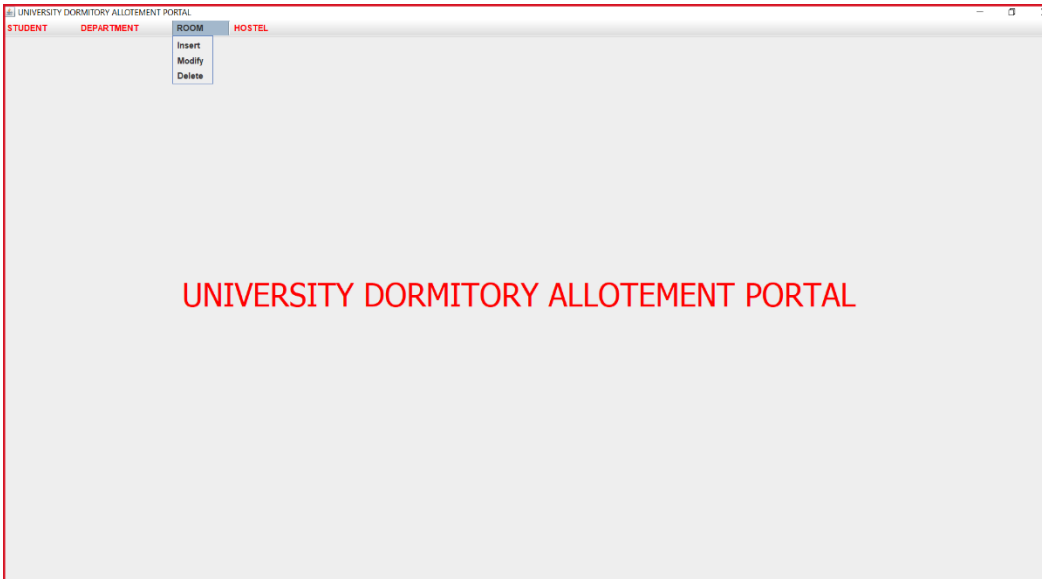
Folder Structure: -

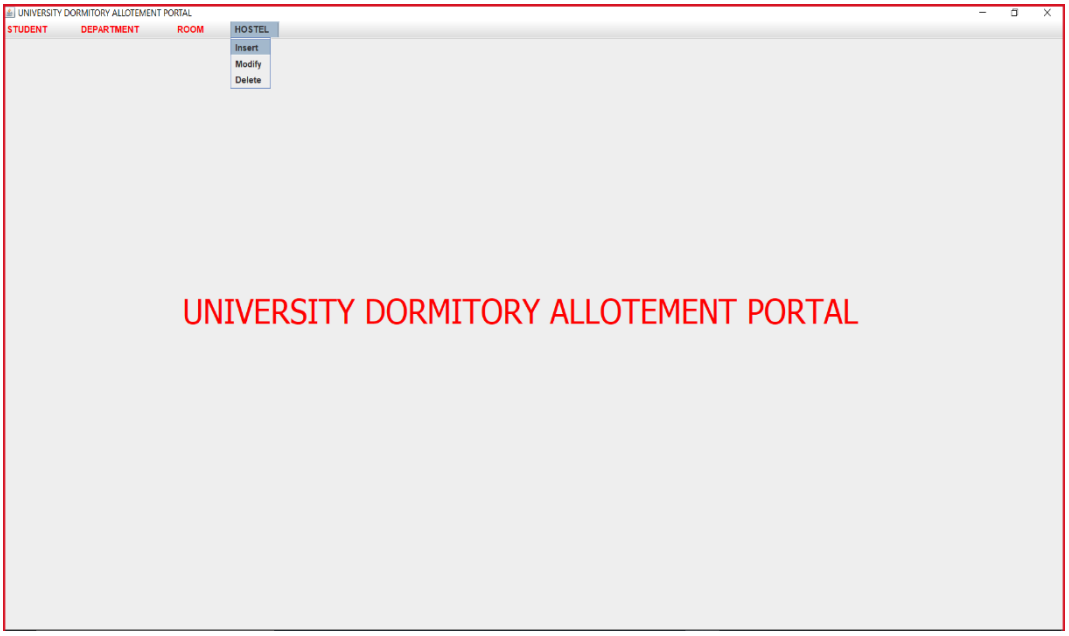
1. Codes.
2. 1602-19-737-010,40 DBMS Assignment 2.
3. 1602-19-737-010,40 Project Report.

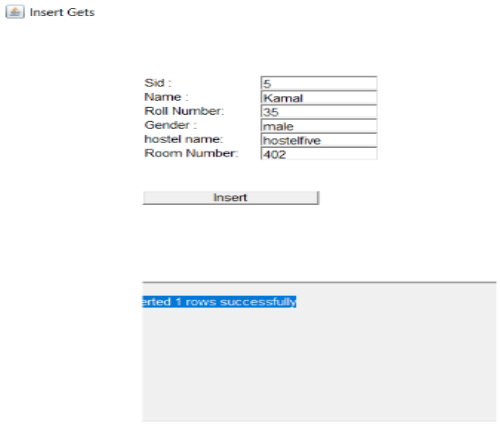
TESTING AND RESULT
MAIN FRAME

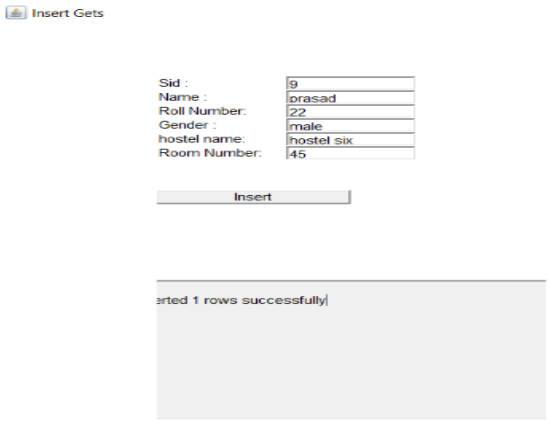


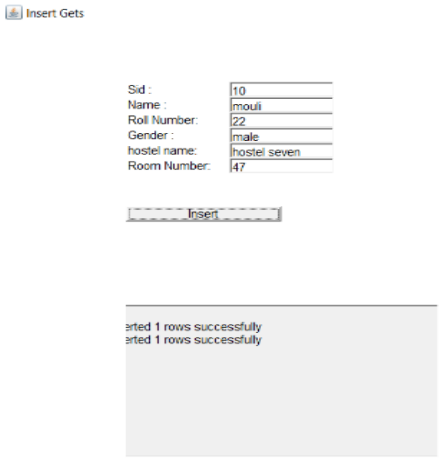


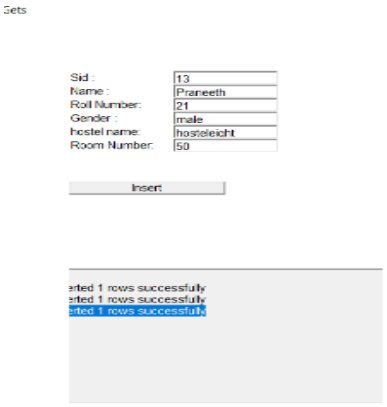

UNIVERSITY DORMITORY ALLOTEMENT PORTAL
STUDENT
DEPARTMENT
ROOM
HOSTEL
Insert
Modify
Delete
UNIVERSITY DORMITORY ALLOTEMENT PORTAL

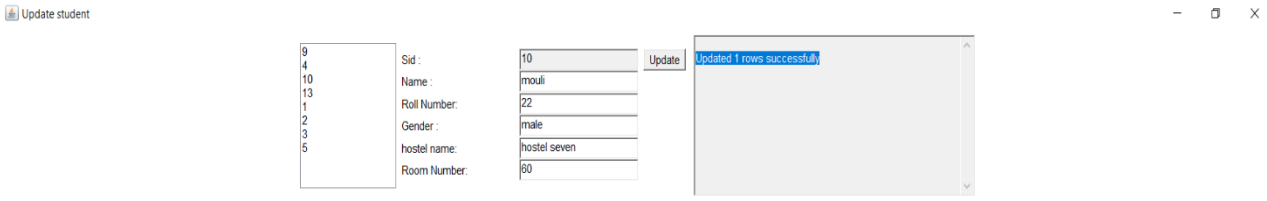

UNIVERSITY DORMITORY ALLOTEMENT PORTAL
STUDENT
DEPARTMENT
ROOM
HOSTEL
Insert
Modify
Delete
UNIVERSITY DORMITORY ALLOTEMENT PORTAL

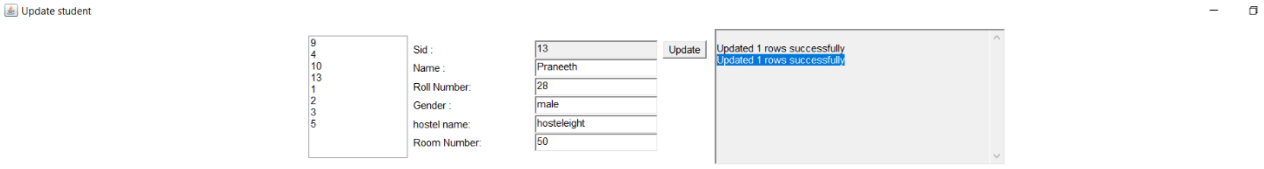

Insert Gets
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
5
Kamal
35
male
hostelfive
402
Insert
erted 1 rows successfully

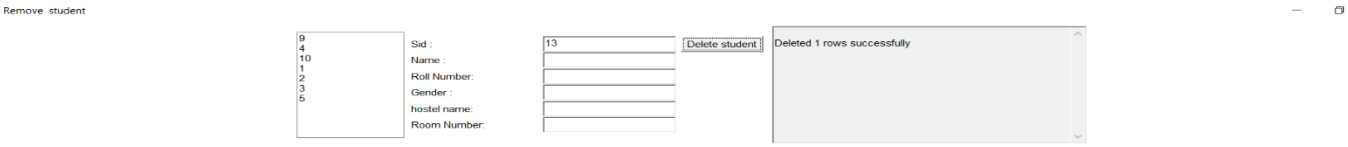

Insert Gets
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
9
prasad
22
male
hostel six
45
Insert
erted 1 rows successfully

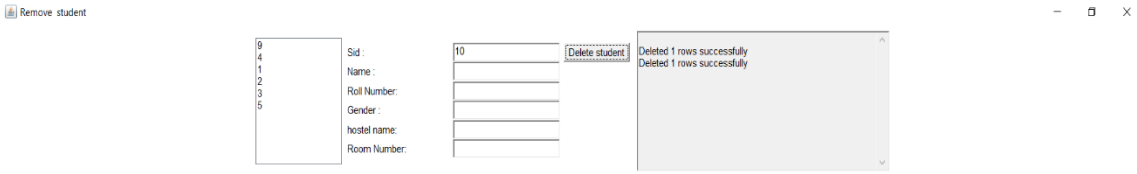

Insert Gets
Sid :
10
Name :
mouli
Roll Number:
22
Gender :
male
hostel name:
hostel seven
Room Number:
47
Insert
erted 1 rows successfully
erted 1 rows successfully


Gets
Sid :
13
Name :
Praneeth
Roll Number:
21
Gender :
male
hostel name:
hosteleight
Room Number:
50
Insert
erted 1 rows successfully
erted 1 rows successfully
erted 1 rows successfully

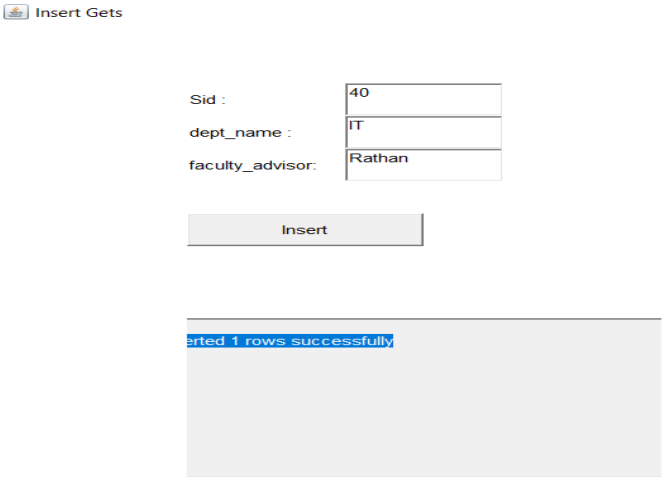

Update student
9
4
10
13
1
2
3
5
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
10
mouli
22
male
hostel seven
80
Update
Updated 1 rows successfully

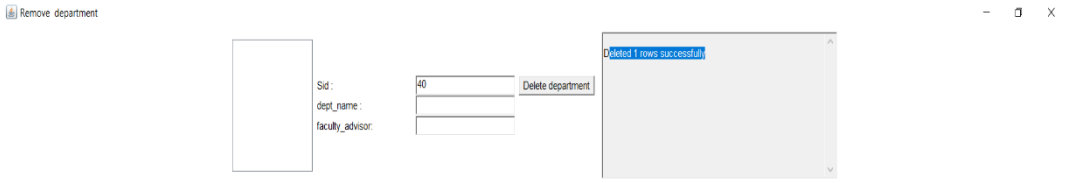

Update student
9
4
10
13
1
2
3
5
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
13
Praneeth
28
male
hosteleight
50
Update
Updated 1 rows successfully
Updated 1 rows successfully

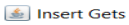

Remove student
9
4
10
1
2
3
5
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
13
Delete student
Deleted 1 rows successfully


Remove student
9
4
1
2
3
5
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
10
Delete student
Deleted 1 rows successfully
Deleted 1 rows successfully


Remove student
9
4
1
2
3
Sid :
Name :
Roll Number:
Gender :
hostel name:
Room Number:
5
Delete student
Deleted 1 rows successfully
Deleted 1 rows successfully
Deleted 1 rows successfully


Insert Gets
Sid :
dept_name :
faculty_advisor:
40
IT
Rathan
Insert
erted 1 rows successfully





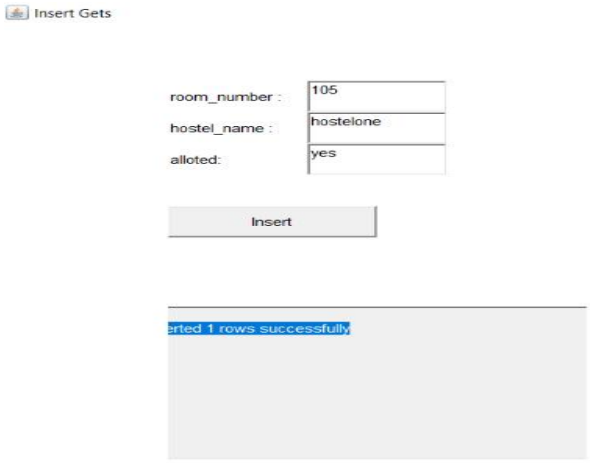
Sid : 18

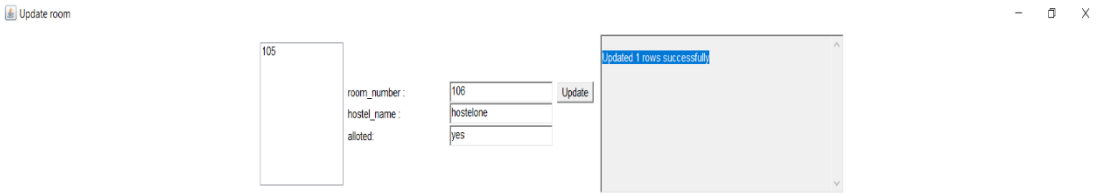
dept_name : sociology

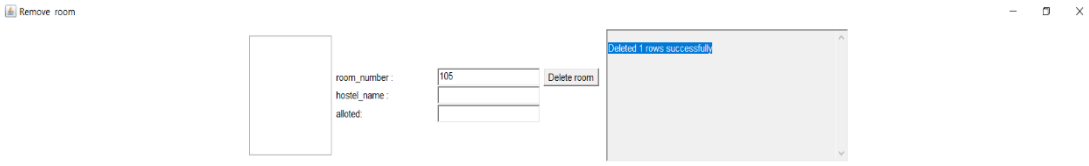
faculty_advisor: preetham

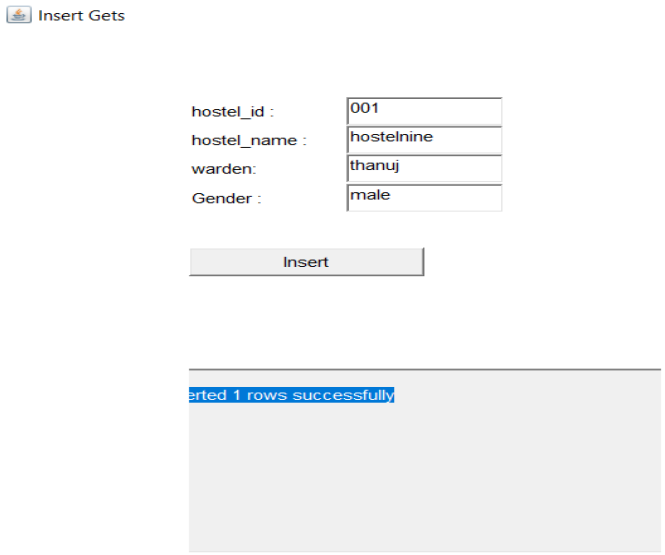
Insert

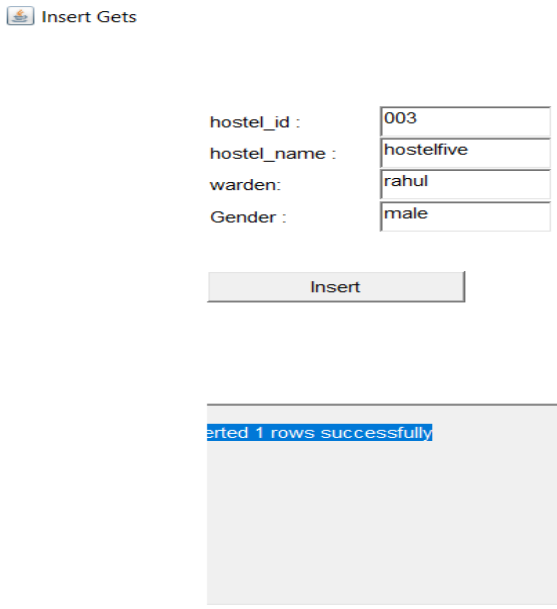
erted 1 rows successfully

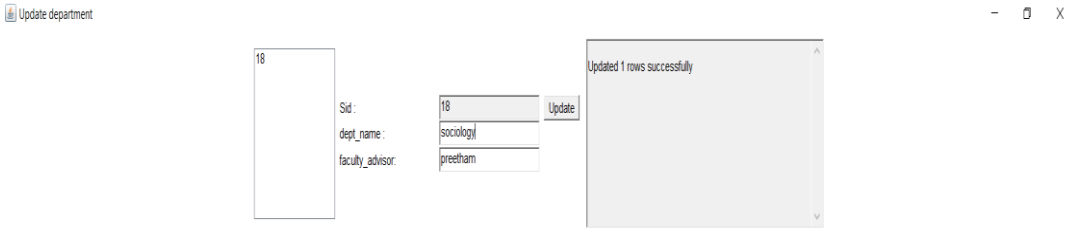

Insert Gets
room_number :
105
hostel_name :
hostelone
alloted:
yes
Insert
erted 1 rows successfully

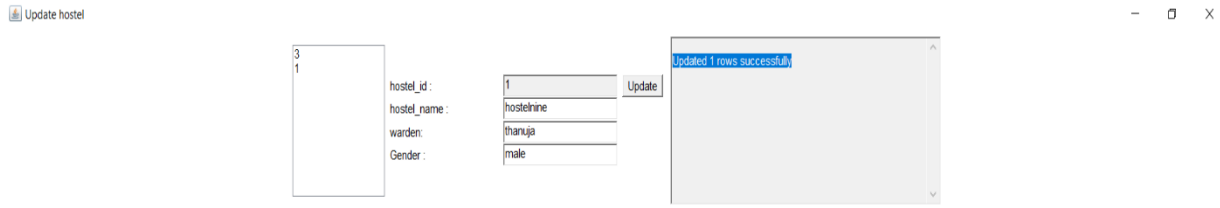

Update room
105
room_number :
106
Update
hostel_name :
hostelone
alloted:
yes
Updated 1 rows successfully

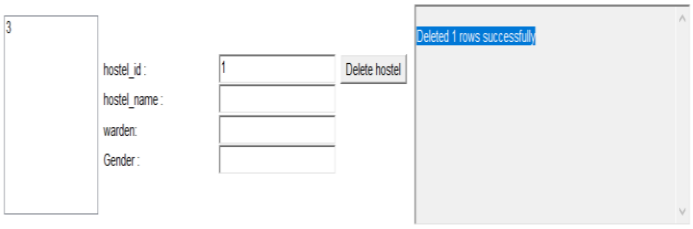

Remove room
room_number :
105
Delete room
hostel_name :
alloted:
Deleted 1 rows successfully


Insert Gets
hostel_id :
001
hostel_name :
hostelnine
warden:
thanuj
Gender :
male
Insert
erted 1 rows successfully


Insert Gets
hostel_id :
003
hostel_name :
hostelfive
warden:
rahul
Gender :
male
Insert
erted 1 rows successfully


Update department
18
Sid :
18
Update
dept_name :
sociology
faculty_advisor:
preetham
Updated 1 rows successfully


Update hostel
3
1
hostel_id :
1
Update
hostel_name :
hostelnine
warden:
thanuja
Gender :
male
Updated 1 rows successfully


3
hostel_id :
1
Delete hostel
hostel_name :
warden:
Gender :
Deleted 1 rows successfully

DISCUSSION AND FUTURE WORK

This project on University Dormitory Allotment Portal is the automation of registration process of students at University Hostels. The system is able to provide much information like student details, hostel details and the department details. The system allows us to add records when a student signs up for a hostel. It also allows to delete and update the records based on student's requirements. This project has guided our path through various aspects of computer science where developing online application plays a major role.

In future the most probable aspects would be including more tables like the student hobbies, student cities, department events etc. Lastly, there could be a room for including advanced software and other technologies that could make the project more purposeful and better for future use.

REFERENCES

https://www.oracle.com/in/database/technologies/appdev/sql.html

https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/package-summary.html

https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/

THANK YOU