Code:

```
 import java.awt.Container;
 import java.awt.event.ActionListener;
 import java.awt.GridLayout;
 import java.awt.event.ActionEvent;
 import java.sql.Connection;
 import java.sql.DriverManager;
 import java.sql.PreparedStatement;
 import javax.swing.JButton;
 Import javax.swing.JComboBox;
 import javax.swing.JInternalFrame;
 import javax.swing.JLabel;
 import javax.swing.JOptionPane;
 import javax.swing.JTextField;

 public class AddSubjects extends JInternalFrame
implements ActionListener{
 private JLabel
lblSubcode,lblSubName,lblCredit,lblMaxmark,lblType,lbl
P ract;
 private JTextField
txtSubcode,txtSubName,txtCredit,txtMaxmark;
 private JComboBox cmbType,cmbPract;
 private JButton btnSave,btnCancel;
 public AddSubjects(){
 super("Add Subjects",false,true,false,true);
 setSize(200,200);
 setLocation(400,200);
 setDefaultCloseOperation(DISPOSE_ON_CLOSE);
Container c=this.getContentPane();
getContentPane().setLayout(new GridLayout(7,2));

lblSubcode=new JLabel(" Code");
lblSubName=new JLabel(" Name");
lblCredit=new JLabel(" Credit");
lblMaxmark=new JLabel(" Max.Mark");
lblType=new JLabel(" Type");
lblPract=new JLabel("Practical");
txtSubcode=new JTextField();
txtSubName=new JTextField();
txtCredit=new JTextField();
txtMaxmark=new JTextField();
cmbType=new JComboBox();
cmbPract=new JComboBox();
cmbType.addItem("COUNTING");
cmbType.addItem("NON COUNTING");
cmbType.setSelectedIndex(0);
cmbPract.addItem("YES");
cmbPract.addItem("NO");
cmbPract.setSelectedIndex(0);
btnSave=new JButton("Save");
btnCancel=new JButton("Cancel");

c.add(lblSubcode);
c.add(txtSubcode);
c.add(lblSubName);
c.add(txtSubName);
c.add(lblCredit);
c.add(txtCredit);
c.add(lblMaxmark);
c.add(txtMaxmark);
c.add(lblType);
c.add(cmbType);
c.add(lblPract);
c.add(cmbPract);
c.add(btnSave);
c.add(btnCancel);

btnSave.addActionListener(this);
btnCancel.addActionListener(this);
}//constructor closed


public void actionPerformed(ActionEvent ae){
if(ae.getSource()==btnSave){
try{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:student");
String sql="INSERT INTO Subject values(?,?,?,?,?,?)";
PreparedStatement ps=con.prepareStatement(sql);
ps.setString(1, txtSubcode.getText());
ps.setString(2, txtSubName.getText());
ps.setInt(3, Integer.parseInt(txtCredit.getText()));
ps.setInt(4, Integer.parseInt(txtMaxmark.getText()));
ps.setString(5,cmbType.getSelectedItem().toString() );
ps.setString(6, cmbPract.getSelectedItem().toString());
int retval=ps.executeUpdate();
if(retval>0){
JOptionPane.showMessageDialog(null, "Record
successfully added to database");
txtCredit.setText("");
txtMaxmark.setText("");
txtSubName.setText("");
txtSubcode.setText("");
cmbType.setSelectedIndex(0);
cmbPract.setSelectedIndex(0);
}
}catch(Exception ex){
JOptionPane.showMessageDialog(null,"Error connection
not established or record already
exists","Error",JOptionPane.ERROR_MESSAGE);
}
}else if(ae.getSource()==btnCancel){
this.dispose();
}
}
}//class closed
```

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Deletedetails extends JInternalFrame
implements ActionListener {

private JPanel panel;
private JButton btnDelete, btnCancel, btnView;
private JLabel lblName, lblAdNo, lblPhNo, lblSex,
lblFatherName, lblOccupation;
private JLabel lblMotherName, lblDOB, lblAge,
lblCaste, lblReligion, lblHouseName;
private JLabel lblCity, lblDistrict, lblState, lblPin,
lblYear, lblQualification;
private JTextField txtName, txtAdNo, txtPhNo,
txtFatherName, txtOccupation;
private JTextField txtMotherName, txtDOB, txtAge,
txtCaste, txtReligion, txtHouseName;
private JTextField txtCity, txtDistrict, txtState, txtPin,
txtYear, txtQualification, txtSex;
private Connection con;
private Statement st;

public Deletedetails() {
super("Edit Student Registration", true, true, true, true);
this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
this.setSize(1000, 1000);

panel = new JPanel();
lblAdNo = new JLabel("Admission no");
lblName = new JLabel("Name");
lblPhNo = new JLabel("Phone no");
lblSex = new JLabel("Sex");
lblFatherName = new JLabel("Father's name");
lblOccupation = new JLabel("Occupation");
lblMotherName = new JLabel("Mother's name");
lblDOB = new JLabel("DOB");
lblAge = new JLabel("Age");
lblCaste = new JLabel("Caste");
lblReligion = new JLabel("Religion");
lblHouseName = new JLabel("House Name");
lblCity = new JLabel("City");
lblDistrict = new JLabel("District");
lblState = new JLabel("State");
lblPin = new JLabel("Pin");
lblYear = new JLabel("Year");
lblQualification = new JLabel("Qualification");

txtName = new JTextField();
txtAdNo = new JTextField();
txtPhNo = new JTextField();
txtSex = new JTextField();
txtFatherName = new JTextField();
txtOccupation = new JTextField();
txtMotherName = new JTextField();
txtDOB = new JTextField();
txtAge = new JTextField();
txtCaste = new JTextField();
txtReligion = new JTextField();
txtHouseName = new JTextField();
txtCity = new JTextField();
txtDistrict = new JTextField();
txtState = new JTextField();
txtPin = new JTextField();
txtQualification = new JTextField();
txtYear = new JTextField();

txtName.setEditable(false);
txtPhNo.setEditable(false);
txtSex.setEditable(false);
txtFatherName.setEditable(false);
txtOccupation.setEditable(false);
txtMotherName.setEditable(false);
txtDOB.setEditable(false);
txtAge.setEditable(false);
txtCaste.setEditable(false);
txtReligion.setEditable(false);
txtHouseName.setEditable(false);
txtCity.setEditable(false);
txtDistrict.setEditable(false);
txtState.setEditable(false);
txtPin.setEditable(false);
txtQualification.setEditable(false);
txtYear.setEditable(false);
btnDelete = new JButton("Delete", new
ImageIcon(ClassLoader.getSystemResource("Images/delusr.png")));
btnDelete.setEnabled(false);
btnCancel = new JButton("Cancel", new
ImageIcon(ClassLoader.getSystemResource("Images/cancel.png")));
btnView = new JButton("View", new
ImageIcon(ClassLoader.getSystemResource("Images/view.png")));
btnDelete.addActionListener(this);
```

```
btnCancel.addActionListener(this);
btnView.addActionListener(this);

panel.setLayout(null);

lblAdNo.setBounds(100, 50, 150, 25);
txtAdNo.setBounds(200, 50, 200, 25);
lblName.setBounds(100, 90, 150, 25);
txtName.setBounds(200, 90, 200, 25);
lblPhNo.setBounds(100, 130, 150, 25);
txtPhNo.setBounds(200, 130, 200, 25);
lblSex.setBounds(100, 170, 150, 25);
txtSex.setBounds(200, 170, 200, 25);
lblFatherName.setBounds(100, 210, 150, 25);
txtFatherName.setBounds(200, 210, 200, 25);
lblOccupation.setBounds(100, 250, 100, 25);
txtOccupation.setBounds(200, 250, 200, 25);
lblMotherName.setBounds(100, 290, 100, 25);
txtMotherName.setBounds(200, 290, 200, 25);
lblDOB.setBounds(100, 330, 150, 25);
txtDOB.setBounds(200, 330, 200, 25);
lblCaste.setBounds(100, 370, 150, 25);
txtCaste.setBounds(200, 370, 200, 25);

lblAge.setBounds(500, 90, 150, 25);
txtAge.setBounds(600, 90, 200, 25);
lblReligion.setBounds(500, 130, 100, 25);
txtReligion.setBounds(600, 130, 200, 25);
lblHouseName.setBounds(500, 170, 100, 25);
txtHouseName.setBounds(600, 170, 200, 25);
lblCity.setBounds(500, 210, 150, 25);
txtCity.setBounds(600, 210, 200, 25);
lblDistrict.setBounds(500, 250, 100, 25);
txtDistrict.setBounds(600, 250, 200, 25);
lblState.setBounds(500, 290, 50, 25);
txtState.setBounds(600, 290, 200, 25);
lblPin.setBounds(500, 330, 150, 25);
txtPin.setBounds(600, 330, 200, 25);
lblYear.setBounds(500, 370, 100, 25);
txtYear.setBounds(600, 370, 200, 25);
lblQualification.setBounds(100, 420, 80, 25);
txtQualification.setBounds(200, 420, 600, 25);
btnDelete.setBounds(300, 470, 100, 25);
btnCancel.setBounds(450, 470, 100, 25);
btnView.setBounds(600, 470, 100, 25);

panel.add(lblAdNo);
panel.add(txtAdNo);
panel.add(lblName);
panel.add(txtName);
panel.add(lblPhNo);
panel.add(txtPhNo);
panel.add(lblSex);
panel.add(txtSex);
```

```
panel.add(lblFatherName);
panel.add(txtFatherName);
panel.add(lblOccupation);
panel.add(txtOccupation);
panel.add(lblMotherName);
panel.add(txtMotherName);
panel.add(lblDOB);
panel.add(txtDOB);
panel.add(lblAge);

panel.add(txtAge);
panel.add(lblCaste);
panel.add(txtCaste);
panel.add(lblReligion);
panel.add(txtReligion);

panel.add(lblHouseName);
panel.add(txtHouseName);
panel.add(lblCity);
panel.add(txtCity);
panel.add(lblDistrict);
panel.add(txtDistrict);
panel.add(lblState);
panel.add(txtState);
panel.add(lblPin);
panel.add(txtPin);
panel.add(txtYear);
panel.add(lblYear);
panel.add(lblQualification);
panel.add(txtQualification);
panel.add(btnDelete);
panel.add(btnCancel);
panel.add(btnView);
add(panel, BorderLayout.CENTER);

try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con =
DriverManager.getConnection("jdbc:odbc:student");
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Error on
connection to database, cannot continue updation process",
"Error", JOptionPane.ERROR_MESSAGE);
}//outer try catch closed
}//constructor closed
public void actionPerformed(ActionEvent e) {
if (e.getActionCommand().equalsIgnoreCase("Delete")) {
try {
//Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
if (e.getSource() == btnDelete) {
String sql = "DELETE FROM student WHERE RollNo="
+ txtAdNo.getText();
st.executeUpdate(sql);
JOptionPane.showMessageDialog(null, "Record
successfully deleted", "Success",
JOptionPane.INFORMATION_MESSAGE);
ClearForm();
btnDelete.setEnabled(false);
}//if closed
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Error on database
operation", "Error", JOptionPane.ERROR_MESSAGE);
}//outer try catch closed
}//if closed
if (e.getActionCommand().equalsIgnoreCase("View")) {
try {
st = con.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM student
WHERE RollNo=" + txtAdNo.getText());
if (rs.next()) {
txtName.setText(rs.getString("SName"));
txtSex.setText(rs.getString("Sex"));
txtFatherName.setText(rs.getString("FName"));
txtMotherName.setText(rs.getString("MName"));
txtAge.setText(rs.getString("Age"));
txtOccupation.setText(rs.getString("Occupation"));
txtPhNo.setText(rs.getString("phno"));
txtReligion.setText(rs.getString("Religion"));
txtHouseName.setText(rs.getString("Hname"));
txtCaste.setText(rs.getString("Caste"));
txtCity.setText(rs.getString("City"));
txtState.setText(rs.getString("State"));
txtDOB.setText(rs.getDate("Dob").toString());
txtDistrict.setText(rs.getString("District"));
txtPin.setText(rs.getString("Pin"));
txtYear.setText(rs.getString("Year"));
txtQualification.setText(rs.getString("Qualification"));
btnDelete.setEnabled(true);
} else {
JOptionPane.showMessageDialog(null, "Roll No not
found in database", "Not found",
JOptionPane.INFORMATION_MESSAGE);
ClearForm();
btnDelete.setEnabled(false);
}//if else closed
} catch (Exception x) {
JOptionPane.showMessageDialog(null, "Error on database
operation,Updation failure", "Error",
JOptionPane.ERROR_MESSAGE);
}//inner try catch closed

}//if closed
if (e.getActionCommand().equalsIgnoreCase("Cancel")) {
this.dispose();
}//if closed
}

private void ClearForm() {
```

```
txtAdNo.setText("");
txtName.setText("");
txtSex.setText("");
txtFatherName.setText("");
txtOccupation.setText("");
txtMotherName.setText("");
txtPhNo.setText("");
txtDOB.setText("");
txtAge.setText("");
txtCaste.setText("");
txtReligion.setText("");
txtHouseName.setText("");
txtCity.setText("");
txtDistrict.setText("");
txtState.setText("");
txtPin.setText("");
txtYear.setText("");
txtQualification.setText("");
}//clearform() closed
}//class closed
private JLabel lblUName = new JLabel("User name");
private JComboBox cmbUName = new JComboBox();
private JButton btnDelete = new JButton("Delete", new
ImageIcon(ClassLoader.getSystemResource("Images/delu
sr.png")));

public Deleteuser() {
super("Delete user accounts", false, true, false, true);
setBounds(350, 200, 300, 200);
setDefaultCloseOperation(DISPOSE_ON_CLOSE);
setLayout(null);
lblUName.setBounds(30, 30, 100, 30);
cmbUName.setBounds(120, 30, 130, 25);
btnDelete.setBounds(100, 90, 100, 25);
add(lblUName);
add(cmbUName);
add(btnDelete);
btnDelete.addActionListener(this);
LoadUserames();
}

public void actionPerformed(ActionEvent e) {
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con =
DriverManager.getConnection("jdbc:odbc:student");
Statement st = con.createStatement();
String Type;
try {
if (e.getSource() == btnDelete) {
String sql = "delete from UAD where Username='" +
cmbUName.getSelectedItem().toString() + "'";
st.executeUpdate(sql);
JOptionPane.showMessageDialog(null, "Database
successfully updated", "Success",
JOptionPane.INFORMATION_MESSAGE);
LoadUserames();
}
} catch (Exception x) {
JOptionPane.showMessageDialog(null, "Error,Unable to
perform database operation", "Error",
JOptionPane.ERROR_MESSAGE);
}//inner try catch closed
} catch (Exception ex1) {
JOptionPane.showMessageDialog(null, "Error on database
connection, Cannot perform database operation", "Error",
JOptionPane.ERROR_MESSAGE);
}//outer try catch closed
}//actionPerformed() closed
private void LoadUserames() {
String name;
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con =
DriverManager.getConnection("jdbc:odbc:student");
Statement st = con.createStatement();
ResultSet rs = st.executeQuery("select Username from
UAD");
cmbUName.removeAllItems();
while (rs.next()) {
name=rs.getString(1);
if (!(name.equalsIgnoreCase("admin"))) {
cmbUName.addItem(name);
}
}
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, "Error while
loading users list, Closing window" + ex.toString());
this.dispose();
}//try catch clsoed
}
}//class closed
```

```
public class DelSubjects extends JInternalFrame
implements ActionListener {

private JLabel lblSubcode, lblSubName, lblCredit,
lblMaxmark;
private JTextField txtSubName, txtCredit, txtMaxmark;
private JComboBox cmbCode;
private JButton btnDelete, btnCancel;

public DelSubjects() {
super("Add Subjects", false, true, false, true);
setSize(200, 150);
setLocation(400, 200);
setDefaultCloseOperation(DISPOSE_ON_CLOSE);
Container c = this.getContentPane();
getContentPane().setLayout(new GridLayout(5, 2));

lblSubcode = new JLabel("Code");
lblSubName = new JLabel("Name");
lblCredit = new JLabel("Credit");
lblMaxmark = new JLabel("Max.Mark");
cmbCode = new JComboBox();
txtSubName = new JTextField();
txtCredit = new JTextField();
txtMaxmark = new JTextField();
btnDelete = new JButton("Delete");
btnCancel = new JButton("Cancel");

c.add(lblSubcode);
c.add(cmbCode);
c.add(lblSubName);
c.add(txtSubName);
c.add(lblCredit);
c.add(txtCredit);
c.add(lblMaxmark);
c.add(txtMaxmark);
c.add(btnDelete);
c.add(btnCancel);
btnDelete.addActionListener(this);
btnCancel.addActionListener(this);
cmbCode.addActionListener(this);
txtCredit.setEditable(false);
txtMaxmark.setEditable(false);
txtSubName.setEditable(false);
loadSubjectCodes();
}//constructor closed

public void actionPerformed(ActionEvent ae) {
if (ae.getSource() == btnDelete) {
try {
if(cmbCode.getSelectedItem()==null)
return;
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con =
DriverManager.getConnection("jdbc:odbc:student");
String sql = "DELETE FROM Subject WHERE
Subjectcode=?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, cmbCode.getSelectedItem().toString());
int retval = ps.executeUpdate();
if (retval > 0) {
JOptionPane.showMessageDialog(null, "Record
successfully deleted");
cmbCode.removeActionListener(this);
loadSubjectCodes();
txtMaxmark.setText("");
txtSubName.setText("");
txtCredit.setText("");
cmbCode.addActionListener(this);
cmbCode.setSelectedIndex(0);

}
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, ex.toString());
}
}
if (ae.getSource() == btnCancel) {
this.dispose();
}
if (ae.getSource() == cmbCode) {
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con =
DriverManager.getConnection("jdbc:odbc:student");
String sql = "SELECT Subjectname,CreditMark,MaxMark
FROM Subject WHERE Subjectcode=?";
PreparedStatement ps = con.prepareStatement(sql);
ps.setString(1, cmbCode.getSelectedItem().toString());
ResultSet rs = ps.executeQuery();
if (rs.next()) {
txtSubName.setText(rs.getString(1));
txtCredit.setText(String.valueOf(rs.getInt(2)));
txtMaxmark.setText(String.valueOf(rs.getString(3)));
}

} catch (Exception ex) {
JOptionPane.showMessageDialog(null,"Error on deletion
process","Error",JOptionPane.ERROR_MESSAGE);
}
}
}

private void loadSubjectCodes() {
try {
cmbCode.removeAllItems();
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con =
DriverManager.getConnection("jdbc:odbc:student");
String sql = "SELECT Subjectcode FROM Subject";
PreparedStatement ps = con.prepareStatement(sql);
ResultSet rs = ps.executeQuery();
while (rs.next()) {
cmbCode.addItem(rs.getString(1));
}
} catch (Exception ex) {
JOptionPane.showMessageDialog(null, ex.toString());
}
}
}
```

```
void register()// for regitsering a new user { s = new
JFrame(); Container c = s.getContentPane();
c.setBackground(Color.CYAN); s.setLayout(null);
s.setVisible(true); s.setSize(500,500);
s.setTitle("Register Here");
s.setBackground(Color.red); fname = new
JLabel("First Name");
fname.setBounds(20,50,100,20); s.add(fname);
getfname = new JTextField();
getfname.setBounds(110,50,150,20);
s.add(getfname); lname = new JLabel("Last Name");
lname.setBounds(20,80,100,20); s.add(lname);
getlname = new JTextField();
getlname.setBounds(110,80,150,20);
s.add(getlname);



email = new JLabel("Email ID");
email.setBounds(20,110,100,20); s.add(email);
getemail = new JTextField();
getemail.setBounds(110,110,150,20);
s.add(getemail);


uname = new JLabel("Username");
uname.setBounds(20,140,100,20); s.add(uname);
getuname = new JTextField();
getuname.setBounds(110,140,150,20);
s.add(getuname); passwd = new
JLabel("Password");
passwd.setBounds(20,170,100,20); s.add(passwd);
getpasswd = new JPasswordField();
getpasswd.setBounds(110,170,150,20);
s.add(getpasswd); registerme = new
JButton("Register Me!!");
registerme.setBounds(80,230,140,30);
s.add(registerme);
registerme.addActionListener(this);
registerme.setForeground(Color.white);
registerme.setBackground(Color.black); } public
void ip() //it will find the IP address of localhost { try
{ InetAddress addr = InetAddress.getLocalHost();
String ipAddress = addr.getHostAddress(); disp_ip =
new JLabel("Your IP is =>
```