

Name : Samarth Mistry**Enrollment no. : 19BEIT30060****Batch : BE-IT I1****Subject : Advance Java Programming (CT506A-N)**

No	Name of Experiment
1	Create a simple calculator application using Swing in Java
2	Implement Student information system using JDBC
3	Create an application using TCP protocol.
4	Create an application using UDP protocol.
5	Develop an EJB application to store and retrieve the student record.
6	Create a Servlet that implements ServletContextAttributeListener interface such that a message dialog is displayed whenever an attribute is added or removed or replaced.
7	Write a servlet that counts the number of times that web page is visited and displays the same information on that page using cookie.
8	Assume that the information regarding the salary and age for all employees of an organization is available in a database. Develop a Servlet application which takes the employee id of an employee as a request parameter and displays the employee information.
9	Create a servlet filter that adds the request processing time in the response page.
10	Create a Login application using servlet and JSP.
11	Create a web page that prints 1 to 10 using JSTL 8.2

12	Create a custom JSP tag that prints current date and time. Use this tag into JSP page.
13	Create a hibernate application for the employee payroll system.
14	Create a "Hello World" application using Spring MVC framework.

PRACTICAL # 01 : Calculator using Java Swing Library

SOURCE CODE practical 01:

```
package p1_calc;
```

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JPanel;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Font;
import javax.swing.SwingConstants;
```

```
public class calc_gui {
    private JFrame frmPracticalBasic;
    private JTextField lcd;
    private String lcd_text="";
    private char operator;
    private Double op1,op2;
    private boolean is_new = false;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    calc_gui window = new calc_gui();
                    window.frmPracticalBasic.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

}
public calc_gui() {
    initialize();
}
private void refreshLcd() {
    lcd.setText(lcd_text);
}
private void checkIsNew() {
    if(is_new) {
        lcd_text = "";
        lcd.setText("");
        op2 = op1 = null;
        is_new = false;
    }
}
private void initialize() {
    frmPracticalBasic = new JFrame();
    frmPracticalBasic.setTitle("Practical 01 Basic Calulator");
    frmPracticalBasic.setResizable(false);
    frmPracticalBasic.setBounds(100, 100, 316, 289);
    frmPracticalBasic.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmPracticalBasic.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBackground(Color.GRAY);
    panel.setBounds(0, 0, 341, 252);
    frmPracticalBasic.getContentPane().add(panel);
    panel.setLayout(null);

    lcd = new JTextField();
    lcd.setFont(new Font("FreeSans", Font.BOLD, 16));
    lcd.setEditable(false);
    lcd.setBounds(12, 5, 288, 35);
    lcd.setText("0.0");
    panel.add(lcd);
    lcd.setColumns(10);

    JButton b7 = new JButton("7");
    b7.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            checkIsNew();
            lcd_text+="7";
            refreshLcd();
        }
    }

```

```
});  
b7.setBounds(22, 52, 42, 24);  
panel.add(b7);  
  
JButton b8 = new JButton("8");  
b8.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="8";  
        refreshLcd();  
    }  
});  
b8.setBounds(78, 52, 42, 24);  
panel.add(b8);  
  
JButton b9 = new JButton("9");  
b9.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="9";  
        refreshLcd();  
    }  
});  
b9.setBounds(132, 52, 42, 24);  
panel.add(b9);  
  
JButton b4 = new JButton("4");  
b4.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="4";  
        refreshLcd();  
    }  
});  
b4.setBounds(22, 83, 42, 24);  
panel.add(b4);  
  
JButton b5 = new JButton("5");  
b5.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="5";  
        refreshLcd();  
    }  
});
```

```
});  
b5.setBounds(79, 84, 42, 24);  
panel.add(b5);  
  
JButton b6 = new JButton("6");  
b6.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="6";  
        refreshLcd();  
    }  
});  
b6.setBounds(133, 83, 42, 24);  
panel.add(b6);  
  
JButton b1 = new JButton("1");  
b1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="1";  
        refreshLcd();  
    }  
});  
b1.setBounds(22, 116, 42, 24);  
panel.add(b1);  
  
JButton b2 = new JButton("2");  
b2.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="2";  
        refreshLcd();  
    }  
});  
b2.setBounds(78, 116, 42, 24);  
panel.add(b2);  
  
JButton b3 = new JButton("3");  
b3.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        checkIsNew();  
        lcd_text+="3";  
        refreshLcd();  
    }  
});
```

```

});
b3.setBounds(132, 116, 42, 24);
panel.add(b3);

JButton b0 = new JButton("0");
b0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        lcd_text+="0";
        refreshLcd();
    }
});
b0.setBounds(78, 149, 42, 24);
panel.add(b0);

JButton bc = new JButton("C");
bc.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        lcd_text="";
        refreshLcd();
    }
});
bc.setBackground(Color.ORANGE);
bc.setBounds(245, 52, 55, 25);
panel.add(bc);

JButton b00 = new JButton("00");
b00.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        lcd_text+="00";
        refreshLcd();
    }
});
b00.setHorizontalAlignment(SwingConstants.LEFT);
b00.setBounds(22, 183, 55, 24);
panel.add(b00);

JButton b_period = new JButton(".");
b_period.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        lcd_text+=".";
    }
});

```

```

        refreshLcd();
    }
});
b_period.setBounds(79, 183, 42, 24);
panel.add(b_period);

JButton b_equals = new JButton("=");
b_equals.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        op2 = Double.parseDouble(lcd.getText());
        Double result;
        if(operator == '+') {
            result = op1+op2;
            lcd.setText(result.toString());
        } else if(operator == '-') {
            result = op1-op2;
            lcd.setText(result.toString());
        } else if(operator == '*') {
            result = op1*op2;
            lcd.setText(result.toString());
        } else if(operator == '/') {
            result = op1/op2;
            lcd.setText(result.toString());
        }
        is_new = true;
    }
});
b_equals.setBounds(132, 183, 44, 24);
panel.add(b_equals);

JButton bPlus = new JButton("+");
bPlus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        operator = '+';
        op1 = Double.parseDouble(lcd.getText());
        lcd_text = "";
        refreshLcd();
    }
});
bPlus.setBackground(Color.ORANGE);
bPlus.setBounds(245, 83, 55, 25);
panel.add(bPlus);

```

```

JButton bSub = new JButton("-");
bSub.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        operator = '-';
        op1 = Double.parseDouble(lcd.getText());
        lcd_text = "";
        refreshLcd();
    }
});
bSub.setBackground(Color.ORANGE);
bSub.setBounds(245, 116, 55, 25);
panel.add(bSub);

```

```

JButton bMul = new JButton("x");
bMul.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        operator = '*';
        op1 = Double.parseDouble(lcd.getText());
        lcd_text = "";
        refreshLcd();
    }
});
bMul.setBackground(Color.ORANGE);
bMul.setBounds(245, 149, 55, 25);
panel.add(bMul);

```

```

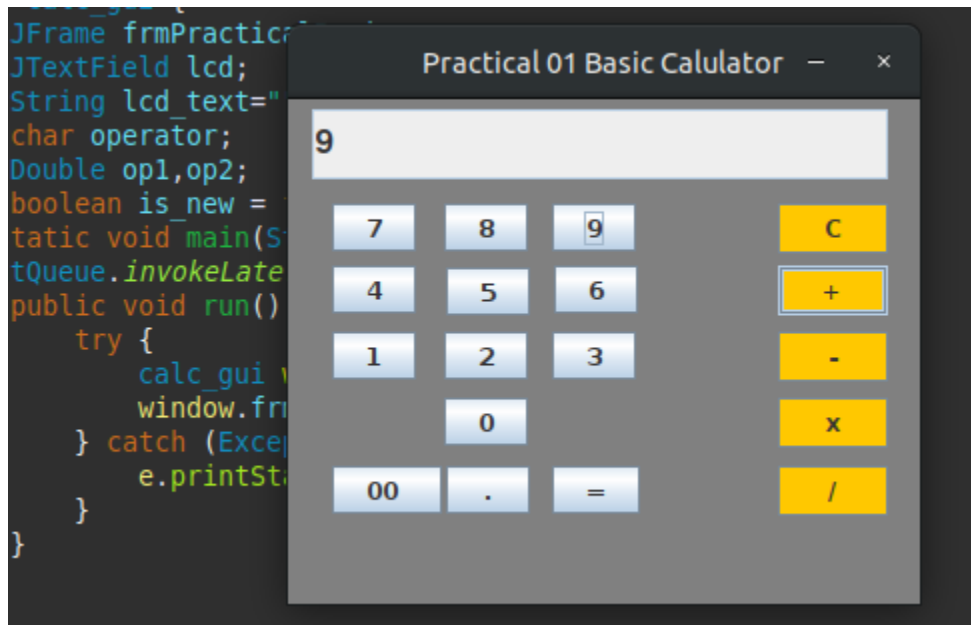
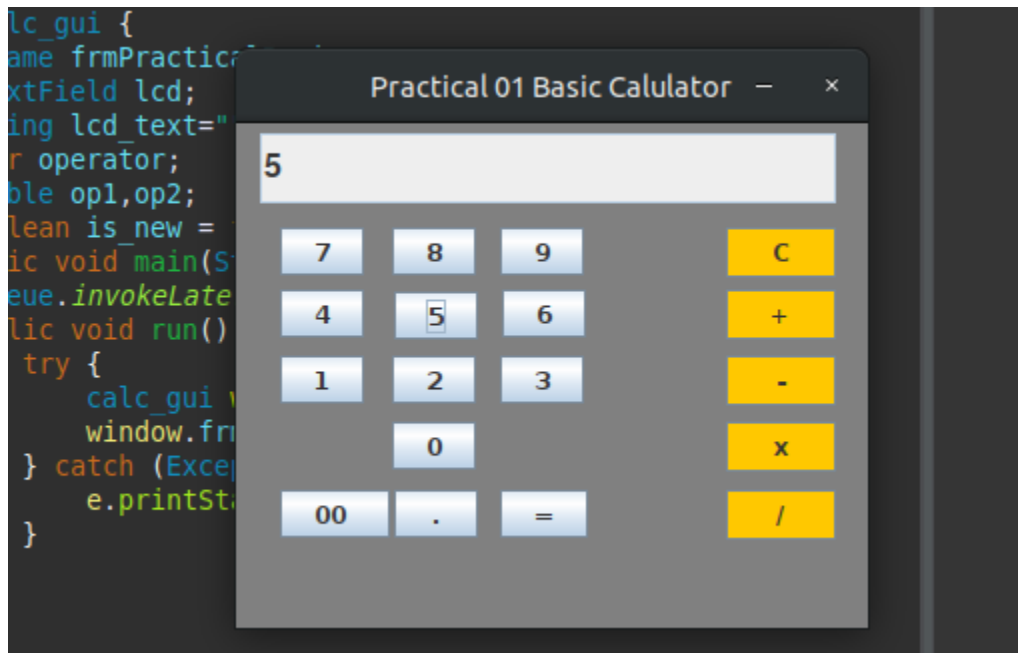
JButton bDivi = new JButton("/");
bDivi.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        checkIsNew();
        operator = '/';
        op1 = Double.parseDouble(lcd.getText());
        lcd_text = "";
        refreshLcd();
    }
});
bDivi.setBackground(Color.ORANGE);
bDivi.setBounds(245, 183, 55, 25);
panel.add(bDivi);

```

```

}
}

```


OUTPUT practical 01 :**Figure 01.1****Figure 01.2**

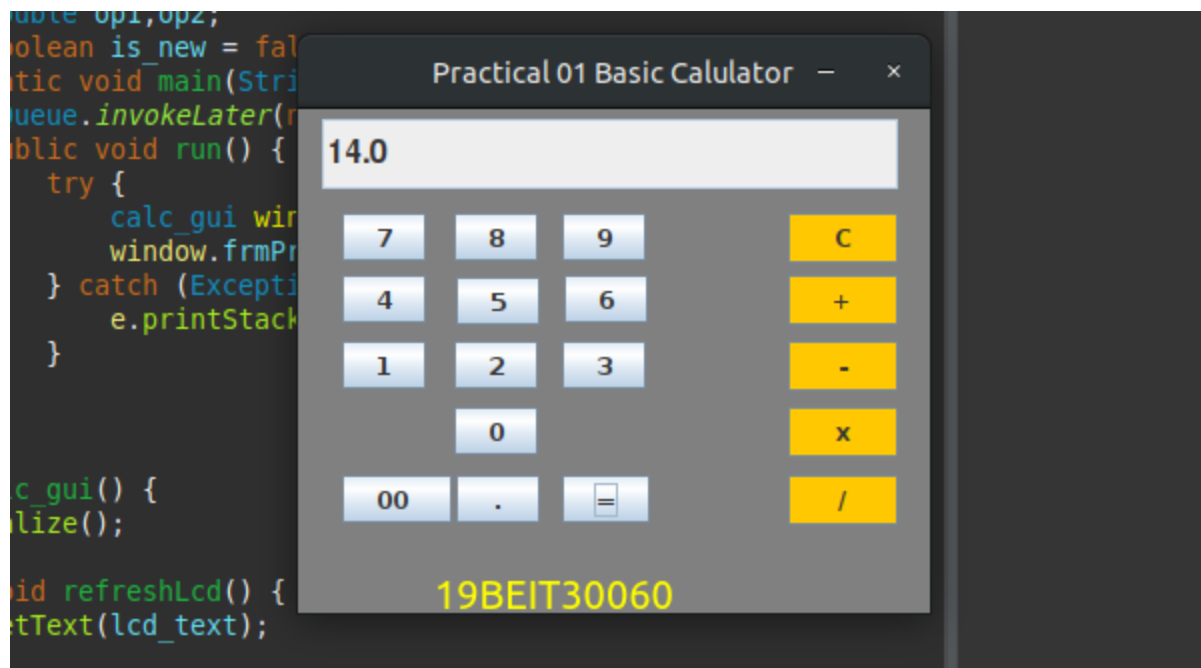


Figure 01.3

PRACTICAL # 02 : Student Information System using Java Swing & JDBC Libraries

SOURCE CODE practical 02:

[FILE NAME : StudentInformation.java]

```
package p2_jdbc;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import java.awt.Color;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.awt.event.ActionEvent;
import java.awt.SystemColor;
import javax.swing.UIManager;
import javax.swing.JRadioButton;
import javax.swing.JTextField;

public class StudentInformation {

    private JFrame frame;
    private JTextField nameTf;
    private JTextField rollnoTf;

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    StudentInformation window = new StudentInformation();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

}
/*
 * Mistry
 */
public StudentInformation() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 634, 424);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JPanel panel = new JPanel();
    panel.setBackground(new Color(218, 165, 32));
    panel.setBounds(12, 30, 337, 334);
    frame.getContentPane().add(panel);

    JLabel lblEnrollment_1 = new JLabel("Gender :");
    lblEnrollment_1.setBounds(12, 68, 127, 15);
    panel.add(lblEnrollment_1);

    JRadioButton maleRd = new JRadioButton("Male");
    maleRd.setActionCommand("male");
    maleRd.setBackground(new Color(218, 165, 32));
    maleRd.setBounds(35, 84, 59, 23);
    panel.add(maleRd);

    JRadioButton femaleRd = new JRadioButton("Female");
    femaleRd.setBackground(new Color(218, 165, 32));
    maleRd.setActionCommand("male");
    femaleRd.setBounds(110, 83, 101, 25);
    panel.add(femaleRd);

    ButtonGroup genderGrp = new ButtonGroup();
    genderGrp.add(femaleRd);
    genderGrp.add(maleRd);

    JButton saveBtn = new JButton("SAVE");
    saveBtn.setBounds(242, 297, 69, 25);
    saveBtn.setBackground(UIManager.getColor("Tree.selectionBorderColor"));
    saveBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {

```

```

String query = "INSERT INTO student_infos (name, gender, eno)
VALUES (?, ?, ?)";

Connection db_conn = xampp_connector.getConnection();
try {
    PreparedStatement pst =
        db_conn.prepareStatement(query);
    pst.setString(1, nameTf.getText());
    if(genderGrp.getSelection().getActionCommand() == "M") {
        System.out.println("Male");
        pst.setString(2, "0");
    } else {
        System.out.println("Female");
        pst.setString(2, "1");
    }
    pst.setString(3, rollNoTf.getText());
    if(pst.executeUpdate() > 0) {
        System.out.println("Successfully Inserted");
        //need to show it on UI not console
    }
} catch (SQLException e) {
    //Exception handling test.....
    e.printStackTrace();
}

});
panel.setLayout(null);
panel.add(saveBtn);

nameTf = new JTextField();
nameTf.setBounds(25, 31, 286, 25);
panel.add(nameTf);
nameTf.setColumns(10);

rollNoTf = new JTextField();
rollNoTf.setColumns(10);
rollNoTf.setBounds(25, 141, 286, 25);
panel.add(rollNoTf);

JLabel lblEnrollment = new JLabel("Enrollment #: ");
lblEnrollment.setBounds(12, 114, 127, 15);
panel.add(lblEnrollment);

```

```
JLabel lblEnrollment_1_1 = new JLabel("Name :");
lblEnrollment_1_1.setBounds(12, 0, 127, 15);
panel.add(lblEnrollment_1_1);
```

```
JLabel lblStudentInformationEntry = new JLabel("Student Information Entry");
lblStudentInformationEntry.setBounds(380, 30, 207, 21);
frame.getContentPane().add(lblStudentInformationEntry);
```

```
}
```

```
}
```

[FILE NAME : xamppConnector.java]

```
package p2_jdbc;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
public class xampp_connector {
```

```
    private static Connection conn;
```

```
    public static Connection getConnection() {
```

```
        try {
```

```
            String url= String.format("jdbc:mysql://localhost:3306/app1");
```

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

```
            conn= DriverManager.getConnection(url,"root","");
```

```
            System.out.println("Connection Established");
```

```
        }catch(Exception e) {
```

```
            System.out.print("Connection Abort");
```

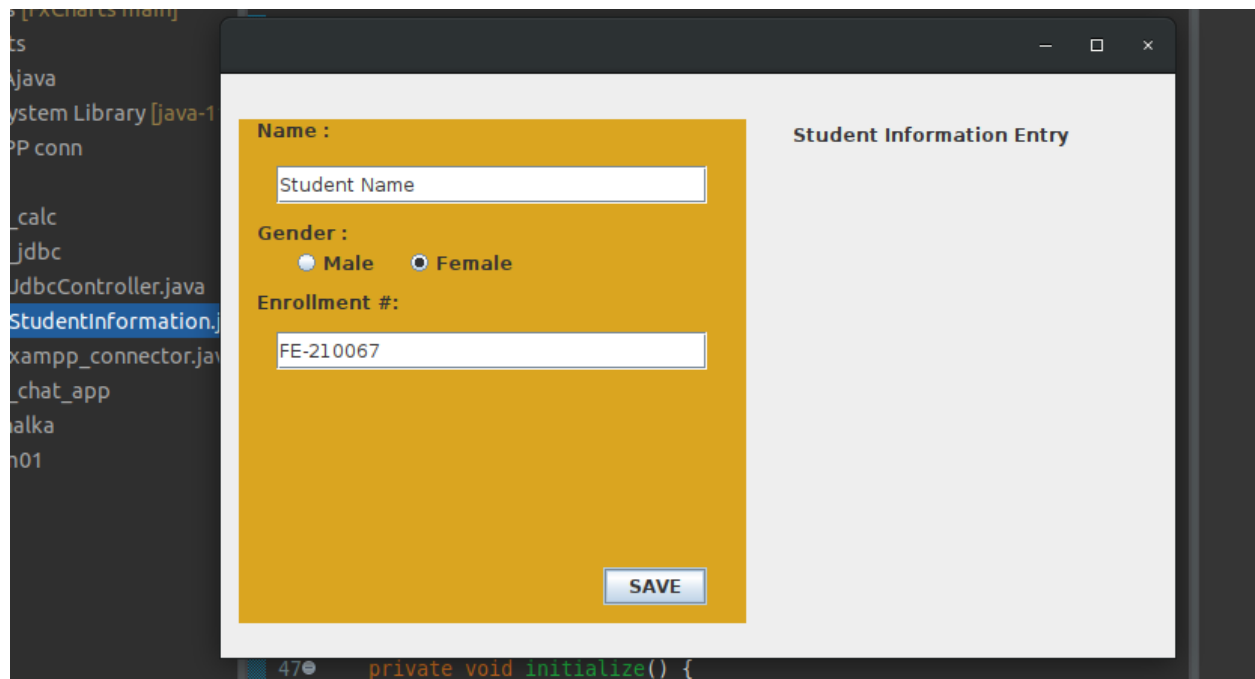
```
            e.printStackTrace();
```

```
        }
```

```
        return conn;
```

```
    }
```

```
}
```

OUTPUT practical 02 :**STEP 1 : Enter the required fields**

The screenshot shows a Java Swing window titled "Student Information Entry" with a yellow background. The window contains three input fields: "Name : Student Name", "Gender : Male (selected) Female", and "Enrollment #: FE-210067". A "SAVE" button is located at the bottom right of the form. The background shows an IDE with Java files like StudentInformation.java.

Figure 02.1**STEP 2 : Press SAVE to enter the data in DB using JDBC driver**

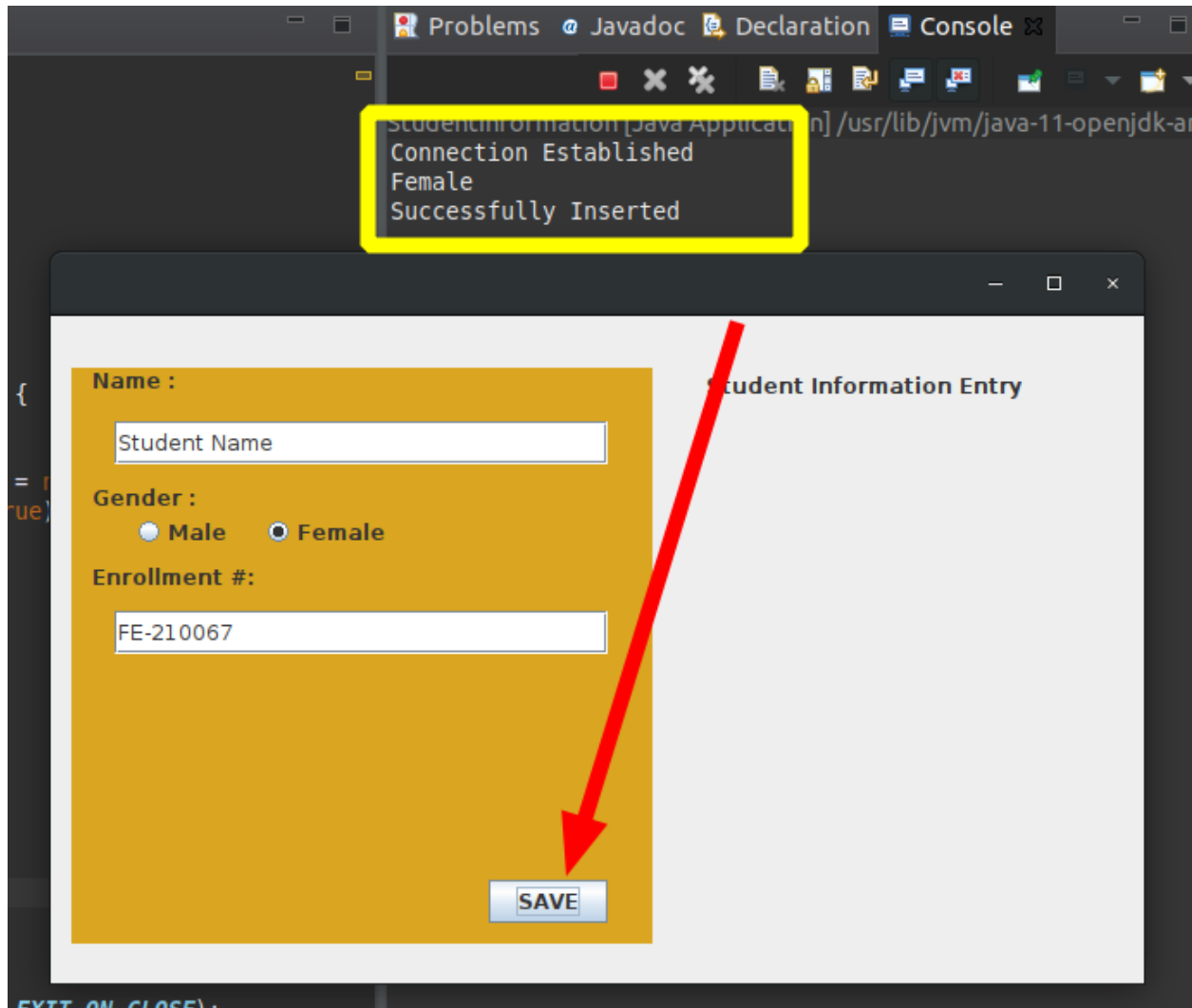


Figure 02.2

DB Entry done

				id	name	gender	eno
<input type="checkbox"/>	Edit	Copy	Delete	1	Test Student	1	19BEIT30060
<input type="checkbox"/>	Edit	Copy	Delete	2	Student Name	1	FE-210067

Figure 02.3

PRACTICAL # 03 : Chat Application using Java.net package & TCP

SOURCE CODE practical 03:

[FILE NAME : ClientTcp.java]

```
package p3_chat_tcp;
import java.io.*;
import java.net.*;
public class ClientTcp {
    public static void main(String[] args) {
        try {
            Socket s=new Socket("localhost",6666);
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            dout.writeUTF("Hello Server");
            dout.flush();
            dout.close();
        } catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

[FILE NAME : ServerTcp.java]

```
package p3_chat_tcp;
import java.io.*;
import java.net.*;
public class ServerTcp {
    public static void main(String[] args){
        try{
            ServerSocket ss=new ServerSocket(6666);
            Socket s=ss.accept();//establishes connection
            DataInputStream dis=new DataInputStream(s.getInputStream());
            String str=(String)dis.readUTF();
            System.out.println("message= "+str);
        } catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT practical 03 :

First message from client side

```

1 package p3_chat_app;
2
3 import java.net.*;
4 public class client_machine {
5     public static void main(String args[]) throws Exception {
6         Socket s = new Socket("localhost", 3333);
7         DataInputStream din = new DataInputStream(s.getInputStream());
8         DataOutputStream dout = new DataOutputStream(s.getOutputStream());
9         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
10
11         String str = "", str2 = "";
12         while (!str.equals("stop")) {
13             str = br.readLine();
14             dout.writeUTF(str);
15         }
16     }
17 }

```

Server is typing...

server_machine [Java Application] /usr/lib/jvm/java-11-openj
 Client: Well, Hello Server
 Hi, I am Responding....

Client is typing....

```

exception {
    getInputStream();
    (s.getOutputStream());
    InputStreamReader(System.in);
}

```

client_machine [Java Application] /usr/lib/jvm/java-11-openj
 Well, Hello Server
 Server: Hi, I am Responding....
 Thank u for Responding

client write stop to end the communication

```

1 package p3_chat_app;
2
3 import java.net.*;
4 public class client_machine {
5     public static void main(String args[]) throws Exception {
6         Socket s = new Socket("localhost", 3333);
7         DataInputStream din = new DataInputStream(s.getInputStream());
8         DataOutputStream dout = new DataOutputStream(s.getOutputStream());
9         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
10
11         String str = "", str2 = "";
12         while (!str.equals("stop")) {
13             str = br.readLine();
14             dout.writeUTF(str);
15             dout.flush();
16         }
17     }
18 }

```

client_machine [Java Application] /usr/lib/jvm/java-11-openj
 Well, Hello Server
 Server: Hi, I am Responding....
 Thank u for Responding
 Server:
 stop

PRACTICAL # 04 : Chat Application using Java.net package & UDP

SOURCE CODE practical 04:

[FILE NAME : UdpSender.java]

```
package p3_chat_udp;

import java.net.*;
public class UdpSender{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        String str = "Welcome java";
        InetAddress ip = InetAddress.getByName("127.0.0.1");

        DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(),ip, 3000);

        ds.send(dp);
        ds.close();
    }
}
```

[FILE NAME : UdpReceiver.java]

```
package p3_chat_udp;

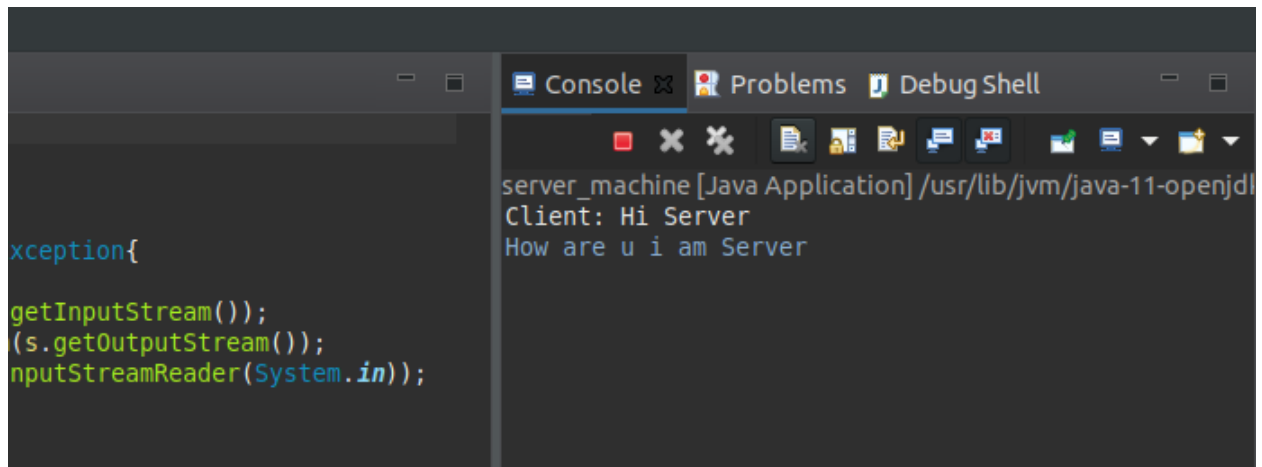
import java.net.*;
public class UdpReceiver{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(3000);
        byte[] buf = new byte[1024];

        System.out.println("Receiver is ready to receive.....");
        DatagramPacket dp = new DatagramPacket(buf, 1024);
        ds.receive(dp);

        String str = new String(dp.getData(), 0, dp.getLength());
        System.out.println(str);

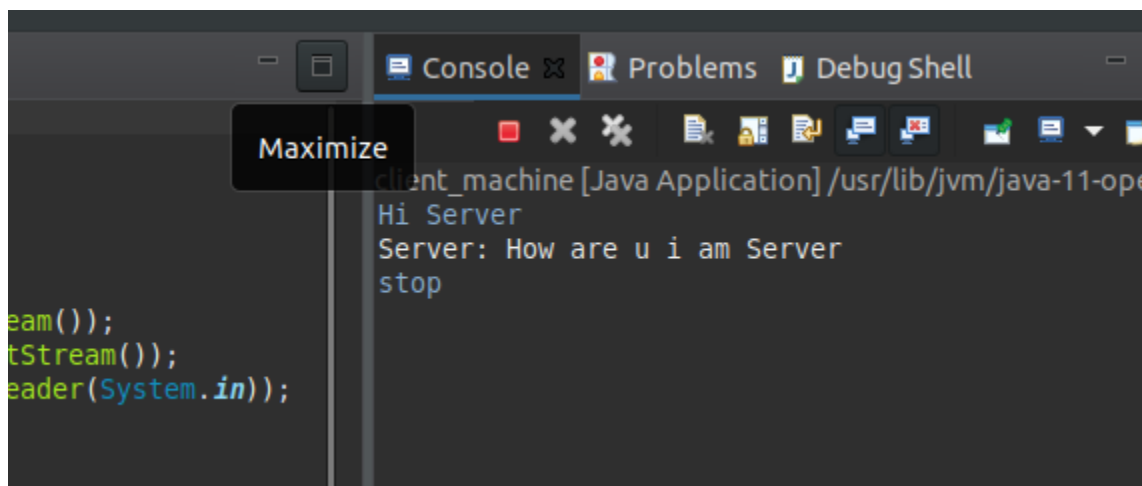
        ds.close();
    }
}
```

OUTPUT practical 04 :



```
exception{  
    getInputStream();  
    (s.getOutputStream());  
    inputSteamReader(System.in));  
}
```

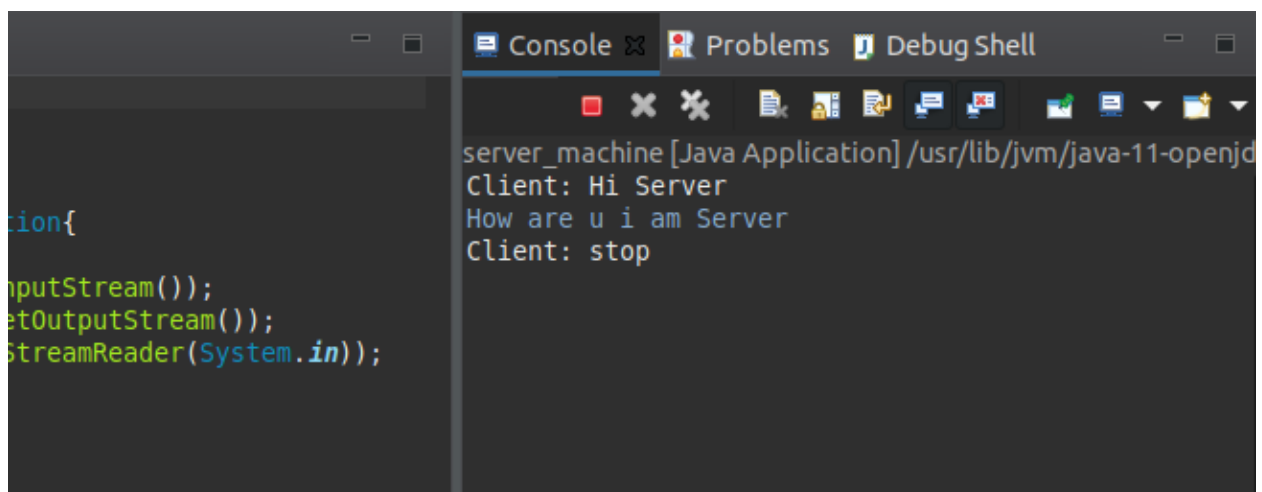
server_machine [Java Application] /usr/lib/jvm/java-11-openjdk
Client: Hi Server
How are u i am Server



Maximize

```
eam();  
tStream();  
eader(System.in));
```

client_machine [Java Application] /usr/lib/jvm/java-11-openjdk
Hi Server
Server: How are u i am Server
stop



```
tion{  
    getInputStream();  
    getOutputStream();  
    inputSteamReader(System.in));  
}
```

server_machine [Java Application] /usr/lib/jvm/java-11-openjdk
Client: Hi Server
How are u i am Server
Client: stop

PRACTICAL # 05 : Develop an EJB application to store and retrieve the student record.

SOURCE CODE practical 05:

[FILE NAME : Student.java]

```
package p4_ejb_app;
import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@Table(name = "STUDENT")
@XmlRootElement
@NamedQueries({
    @NamedQuery(
        name = "Student.findAll",
        query = "SELECT s FROM Student s"
    ),
    @NamedQuery(
        name = "Student.findByEnrollmentno",
        query = "SELECT s FROM Student s WHERE s.enrollmentno =:enrollmentno"
    ),
    @NamedQuery(
        name = "Student.findByName",
        query = "SELECT s FROM Student s WHERE s.name = :name"
    ),
    @NamedQuery(
        name = "Student.findByAddress",
        query = "SELECT s FROM Student s WHERE s.address = :address"
    )
})

public class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 255)
```

```

@Column(name = "ENROLLNMENTNO")

private String enrollmentno;
@Size(max = 255)
@Column(name = "NAME")

private String name;
@Size(max = 255)
@Column(name = "ADDRESS")

private String address;
public Student() {}
public Student(String enrollmentno) {
    this.enrollmentno = enrollmentno;
}
public String getEnrollmentno() {
    return enrollmentno;
}
public void setEnrollmentno(String enrollmentno) {
    this.enrollmentno = enrollmentno;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getAddress() {
    return address;
}
public void setAddress(String address) {
    this.address = address;
}
@Override
public int hashCode() {
    int hash = 0;
    hash += (enrollmentno != null ?
        enrollmentno.hashCode() : 0);
    return hash;
}
@Override
public boolean equals(Object object) {
    if (!(object instanceof Student)) {
        return false;
    }
    Student other = (Student) object;

```

```

        if ((this.enrollmentno == null && other.enrollmentno != null) || (this.enrollmentno !=
        null && !this.enrollmentno.equals(other.enrollmentno))) {
            return false;
        }
        return true;
    }
    @Override
    public String toString() {
        return "stu.Student[ enrollmentno=" + enrollmentno + " ]";
    }
}
    @Override
    public String toString() {
        return "stu.Student[ enrollmentno=" + enrollmentno + " ]";
    }
}
}

```

[FILE NAME : AbstractFacade.java]

```

package stu;

import java.util.List;
import javax.persistence.EntityManager;

public abstract class AbstractFacade<T> {
    private Class<T> entityClass;
    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }
    protected abstract EntityManager getEntityManager();
    public void create(T entity) {
        getEntityManager().persist(entity);
    }
    public void edit(T entity) {
        getEntityManager().merge(entity);
    }
    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }
    public T find(Object id) {
        return getEntityManager().find(entityClass, id);
    }
    public List<T> findAll() {
        javax.persistence.criteria.CriteriaQuery cq =
        getEntityManager().getCriteriaBuilder().createQuery();

        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }
}

```

```

    }
    public List<T> findRange(int[] range) {
        javax.persistence.criteria.CriteriaQuery cq =
            getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));

        javax.persistence.Query q =
            getEntityManager().createQuery(cq);

        q.setMaxResults(range[1] - range[0] + 1);

        q.setFirstResult(range[0]);
        return q.getResultList();
    }
    public int count() {
        javax.persistence.criteria.CriteriaQuery cq =
            getEntityManager().getCriteriaBuilder().createQuery();

        javax.persistence.criteria.Root<T> rt = cq.from(entityClass);
        cq.select(getEntityManager().getCriteriaBuilder().count(rt));

        javax.persistence.Query q =
            getEntityManager().createQuery(cq);

        return ((Long) q.getSingleResult()).intValue();
    }
}

```

[FILE NAME :StudentFacade.java]

```

package stu;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
@Stateless
public class StudentFacade extends AbstractFacade<Student> implements StudentFacadeLocal {
    @PersistenceContext(unitName = "Student-EJB-ejbPU")
    private EntityManager em;
    @Override
    protected EntityManager getEntityManager() {
        return em;
    }
    public StudentFacade() {
        super(Student.class);
    }
    public String insertRecord(String eno,String n,String a){
        try{

```



```

        Student ob=new Student();
        ob.setEnrollmentno(eno);
        ob.setName(n);
        ob.setAddress(a);
        em.persist(ob);
        return " Record Added Successfully ";
    }
    catch(Exception e){ return "Error "; }
}

```

[FILE NAME :StudentFacadeLocal.java]

```

package stu;

import java.util.List;
import javax.ejb.Local;
@Local
public interface StudentFacadeLocal {
    void create(Student student);
    void edit(Student student);
    void remove(Student student);
    Student find(Object id);
    List<Student> findAll();
    List<Student> findRange(int[] range);
    int count();
    public String insertRecord(String eno, String n, String a);
}

```

[FILE NAME : StudentServlet.java]

```

package stu;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class StudentServlet extends HttpServlet {
    @EJB
    private StudentFacadeLocal studentFacade;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String e,n,a;

```

```

        e=request.getParameter("eno");
        n=request.getParameter("name");
        a=request.getParameter("address");
        String result=studentFacade.insertRecord(e,n,a);
        out.println(" "+result);
        List<Student> res = studentFacade.findAll();
        for(Student i:res){
            out.print("<br/> Enrollment No is : "
                +i.getEnrollmentno());
            out.print("<br/> Name is : "+i.getName());
            out.print("<br/> Address is : "+i.getAddress());
            out.print("<br/>");
        }
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {
    return "Short description";
}
}

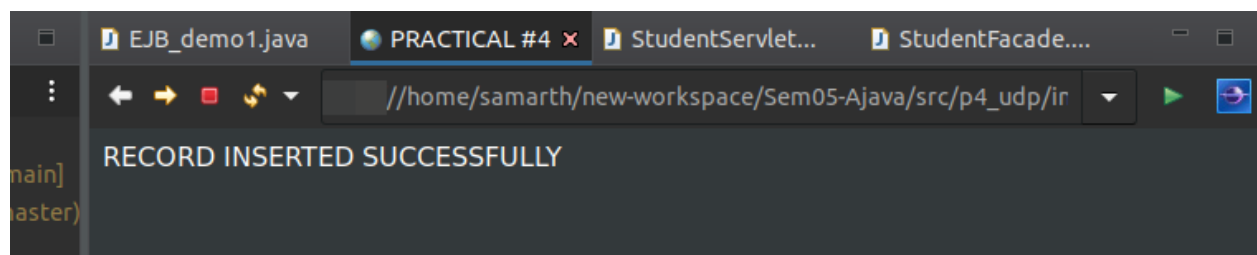
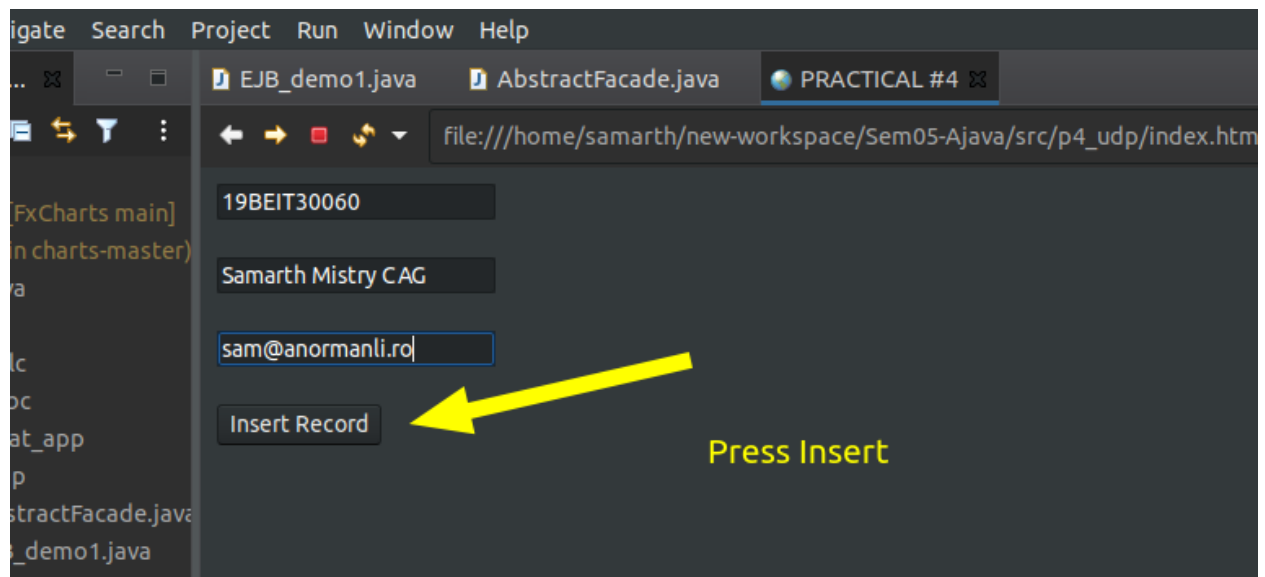
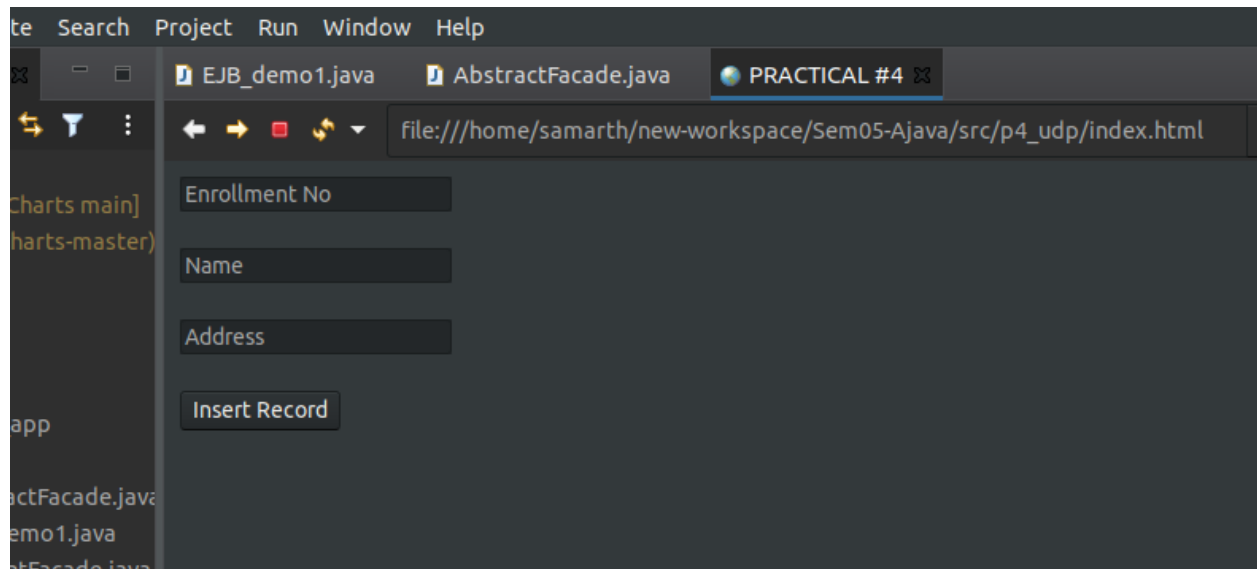
```

[FILE NAME : index.html]

```

<html>
    <head>
        <title>PRACTICAL #4</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="StudentServlet">
            <input type="text" placeholder="Enrollment No " name="eno"><br/><br/>
            <input type="text" placeholder="Name " name="name"><br/><br/>
            <input type="text" placeholder="Address " name="address"><br/><br/>
            <input type="submit" value="Insert Record">
        </form>
    </body>
</html>

```

OUTPUT practical 05 :

PRACTICAL # 06 : Create a Servlet that implements ServletContextAttributeListener interface such that a message dialog is displayed whenever an attribute is added or removed or replaced.

SOURCE CODE practical 06 :

[FILE NAME : SContextListener.java]

```
import javax.servlet.ServletContextAttributeEvent;
import javax.servlet.ServletContextAttributeListener;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

public class SContextListener implements ServletContextListener, ServletContextAttributeListener {

    @Override
    public void contextInitialized(ServletContextEvent sce) {
        System.out.println("INITIALIZED");
    }

    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        System.out.println("CONTEXT DESTROYED");
    }

    @Override
    public void attributeAdded(ServletContextAttributeEvent event) {
        System.out.println("ATTR. ADDED");
    }

    @Override
    public void attributeRemoved(ServletContextAttributeEvent event) {
        System.out.println("ATTR. REMOVED");
    }

    @Override
    public void attributeReplaced(ServletContextAttributeEvent arg0) {
        System.out.println("ATTR. REPLACED");
    }
}
```

[FILE NAME : Servlet.java]

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```

public class Servlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("welcome to servlet.....");
            ServletContext ob=getServletContext();
            ob.setAttribute("name","abc");
            ob.setAttribute("name","aaa");
            ob.removeAttribute("name");

        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```

OUTPUT practical 06 :

```

init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsp:
Incrementally deploying http://localhost:8080/Servelet
Completed incremental distribution of http://localhost:8080/Servelet
Incrementally redeploying http://localhost:8080/Servelet
Reload is in progress...
reload?path=/Servelet
OK - Reloaded application at context path [/Servelet]
run-deploy:
Browsing: http://localhost:8080/Servelet
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 0 seconds)
|

```

PRACTICAL # 07 : Write a servlet that counts the number of times that web page is visited and displays the same information on that page using cookie.

SOURCE CODE practical 07 :

[FILE NAME : index.html]

```

<!DOCTYPE html>
<html>

```

```

<head>
    <title>COUNT</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
</head>
<body>
    <p>Sample web page for counting</p>
</body>
</html>

```

[FILE NAME : CookieCount.java]

```

package aa;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "CookieCount", urlPatterns = {"/CookieCount"})
public class CookieCount extends HttpServlet {
    static int count=1;
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String str=String.valueOf(count);
            Cookie ck = new Cookie("visitcount",str);
            response.addCookie(ck);
            out.println(" You have visited this page : "+count+" times.");
            count++;
            RequestDispatcher rd = request.getRequestDispatcher("S2");
            rd.forward(request, response);
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws
    ServletException, IOException {
        processRequest(request, response);
    }
}

```

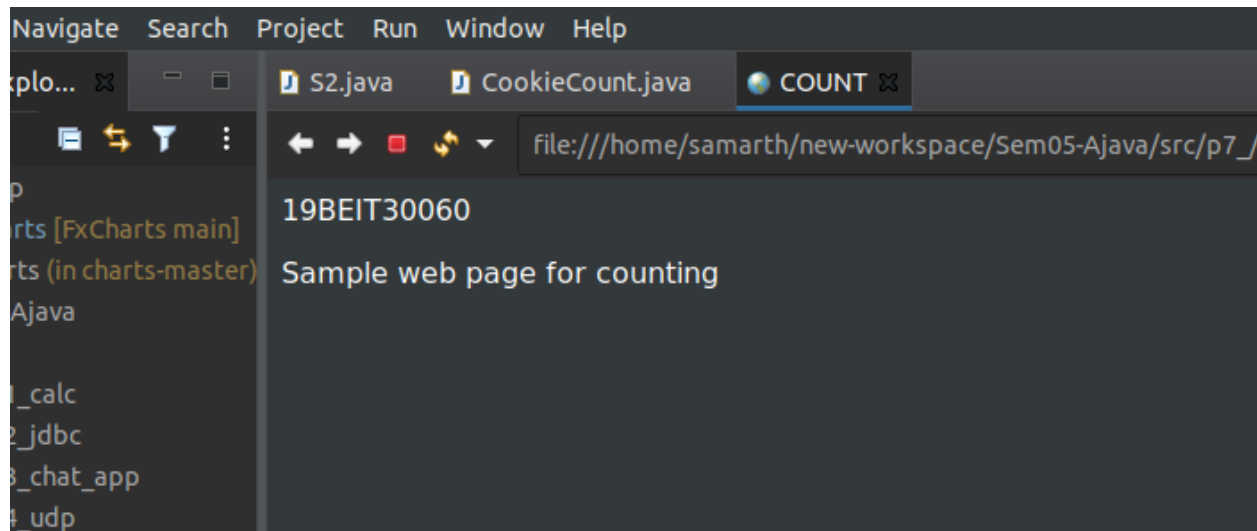
```

        @Override
        public String getServletInfo() {
            return "Short description";
        }
    }
}

[FILE NAME : S2.java]
package aa;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "S2", urlPatterns = {"/S2"})
public class S2 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            Cookie ck[]=request.getCookies();
            out.print(" Value of Cookie is : "+ck[0].getValue());
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```


OUTPUT practical 07 :

PRACTICAL # 08 : Assume that the information regarding the salary and age for all employees of an organization are available in a database. Develop a Servlet application which takes the employee id of an employee as a request parameter and displays the marksheet for the student.

SOURCE CODE practical 08 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
    <head>
        <title>COUNT</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
    </head>
    <body>
        <form action="EmployeeServlet">
            <input type="text" placeholder="Emp ID" name="empid"/>
            <br>
            <input type="submit" value="Get Details"/>
        </form>
    </body>
</html>
```

[FILE NAME : EmployeeServlet.java]

```
package p7_emp_details

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
```

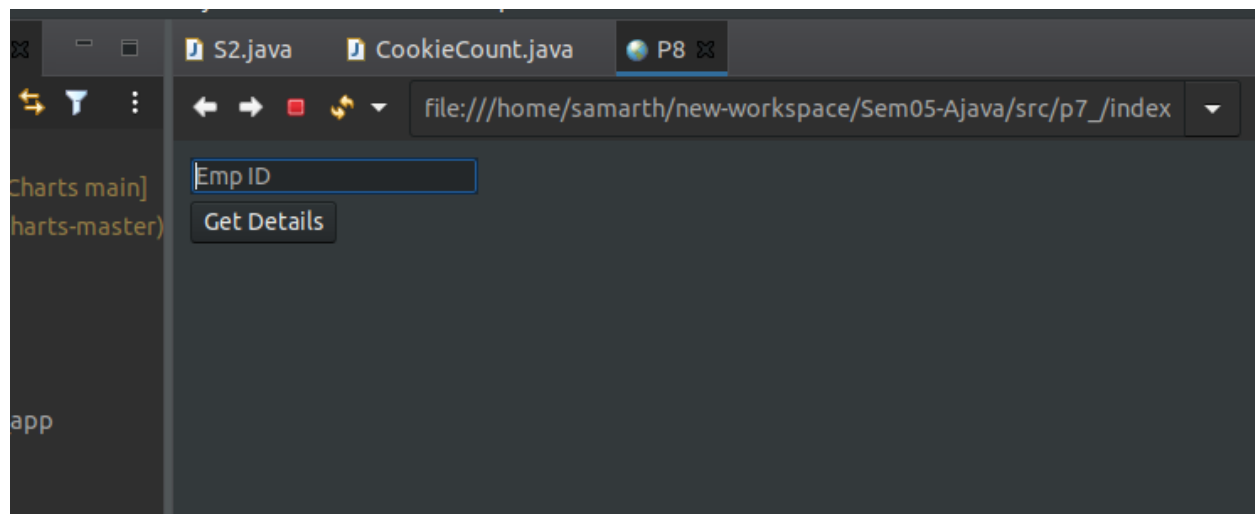
```

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(urlPatterns = {"/EmployeeServlet"})
public class EmployeeServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String no=request.getParameter("empid");
            try {
                Class.forName("org.apache.derby.jdbc.ClientDriver");
                Connection
                cn=DriverManager.getConnection("jdbc:derby://localhost:8000/p7","root","");
                Statement st=cn.createStatement();
                ResultSet rs=st.executeQuery("select * from employee_info where
                empid='"+no+"'");
                while(rs.next()) {
                    out.println("<table border='2' cellpadding='5'>");
                    out.println("<tr><th>");
                    out.println(" Employee ID ");
                    out.println("</th><th>");
                    out.println(" Name ");
                    out.println("</th><th>");
                    out.println(" Salary ");
                    out.println("</th><th>");
                    out.println(" Age ");
                    out.println("</th></tr><tr><td>");
                    out.println(" "+rs.getString(1));
                    out.println("</td><td>");
                    out.println(" "+rs.getString(2));
                    out.println("</td><td>");
                    out.println(" "+rs.getInt(3));
                    out.println("</td><td>");
                    out.println(" "+rs.getInt(4));
                    out.println("</td></tr></table>");
                }
                cn.close();
            }
            catch(Exception e) {out.println("Error "+e);}
        }
    }
}
@Override

```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}
@Override
public String getServletInfo() {
    return "Short description";
}
}
```

OUTPUT practical 08 :



PRACTICAL # 09 : Create a servlet filter that adds the request processing time in the response page.

SOURCE CODE practical 09 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
    <head>
        <title>COUNT</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
    </head>
    <body>
        <form action="S1">
            <input type="submit" value="Click Here ">
        </form>
    </body>
</html>
```

[FILE NAME : EmployeeServlet.java]

```
package aa;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class S1 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            for(float i=0;i<100000;i++){
                out.println(i);
            }
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }
}
```

```

    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}

```

[FILE NAME : newfilter.java]

```

package aa;
import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
public class newfilter implements Filter {
    @Override
    public void init(FilterConfig filterConfig) throws ServletException {}
    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
    throws IOException, ServletException {
        long beforetime = System.currentTimeMillis();
        chain.doFilter(request, response);
        long aftertime = System.currentTimeMillis();
        System.out.println("Time taken By Servlet is : "+(aftertime-beforetime)+" ");
    }
    @Override
    public void destroy() {}
}

```

[FILE NAME : web.xml]

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd">
    <filter>
        <filter-name>a</filter-name>
        <filter-class>aa.newfilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>a</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <servlet>
        <servlet-name>S1</servlet-name>
        <servlet-class>aa.S1</servlet-class>

```

```

</servlet>
<servlet-mapping>
    <servlet-name>S1</servlet-name>
    <url-pattern>/S1</url-pattern>
</servlet-mapping>
<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
</web-app>

```

OUTPUT practical 09 :



PRACTICAL # 10 : Create a Login application using servlet and JSP.

SOURCE CODE practical 10 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
    <head>
        <title>COUNT</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
    </head>
    <body>
        <form action="Login">
            <input type="text" name="uname" placeholder="Username "/>
            <br>
            <input type="password" name="upass" placeholder="Password "/>
            <br>
            <input type="submit" value=" LOGIN "/>
        </form>
    </body>
</html>
```

[FILE NAME : Login.java]

```
package p10_login_servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.swing.JOptionPane;
@WebServlet(urlPatterns = {"/Login"})
public class Login extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String uname=request.getParameter("uname");
            String upass=request.getParameter("upass");
```

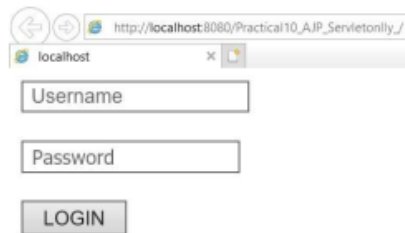


```

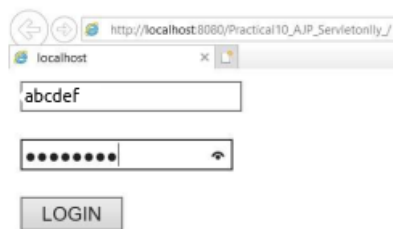
try{
    Class.forName("org.apache.derby.jdbc.ClientDriver");
    Connection
    cn=DriverManager.getConnection("jdbc:derby://localhost:8000/p10","r
oot","");
    Statement st=cn.createStatement();
    ResultSet rs=st.executeQuery("select * from admin where
username='"+uname+"'");
    if(rs.next()) {
        String s=rs.getString(2);
        if(s.equals(upass))
            out.println("Welcome "+uname);
        else{
            out.println(" Invalid Password for Username"+uname);
            out.println("<br/><br/>");
            RequestDispatcher
            rd=request.getRequestDispatcher("index.html");
            rd.include(request, response);
        }
    }
    else {
        out.println(" Invalid Username ");
        out.println("<br/><br/>");
        RequestDispatcher
        rd=request.getRequestDispatcher("index.html");
        rd.include(request, response);
    }
}
catch(Exception e){out.println("Error "+e);}
}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request, response);
}
@Override
public String getServletInfo() {
    return "Short description";
}
}

```

OUTPUT practical 10(A) Servlet:

A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/Practical10_AJP_Servletonly_/_`. The browser tab is labeled `localhost`. The page content includes a login form with two input fields: `Username` and `Password`, and a `LOGIN` button below them.



A screenshot of the same web browser window. The `Username` field now contains the text `abcdef`. The `Password` field contains masked characters (dots) and a small icon on the right. The `LOGIN` button remains below the fields.

Welcome abcdef

PRACTICAL # 10(B) : Create a Login application using servlet and JSP.**SOURCE CODE practical 10(B) :****[FILE NAME : index.html]**

```

<!DOCTYPE html>
<html>
    <head>
        <title>COUNT</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
    </head>
    <body>
        <form action="Login.jsp">
            <input type="text" name="uname" placeholder="Username "/>
            <br>
            <input type="password" name="upass" placeholder="Password "/>
            <br>
            <input type="submit" value=" LOGIN "/>
        </form>
    </body>
</html>

```

[FILE NAME : Login.jsp]

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
    </head>
    <body>
        <%
            String username=request.getParameter("username");
            String password=request.getParameter("password");

            if(username.equals("Samarth") && password.equals("00000000")) {
                out.print("Login Successful");

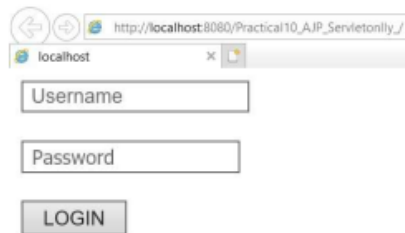
                <jsp:forward page="inbox.jsp">
                <jsp:param name="username" value="<%=username%>"/>
                </jsp:forward>
            }
            else
            {

```

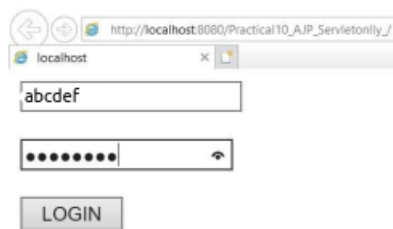
```
        out.print("Login Failed");
    %>
    <jsp:include page="index.html"/>
    <%
        }
    %>
</body>
</html>
```

[FILE NAME : inbox.jsp]

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>P10(B) JSP</title>
    </head>
    <body>
        <%
            String name=request.getParameter("username");
            out.print("Welcome "+name);
        %>
    </body>
</html>
```

OUTPUT practical 10(B) JSP :

A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/Practical10_AJP_Servletonly_/_`. The browser tab is labeled `localhost`. The page content includes a form with two input fields: `Username` and `Password`, followed by a `LOGIN` button.



A screenshot of the same web browser window. The `Username` field now contains the text `abcdef`. The `Password` field is masked with ten dots. The `LOGIN` button remains visible.

Welcome abcdef

PRACTICAL # 11 : Create a web page that prints 1 to 10 using JSTL

8.2

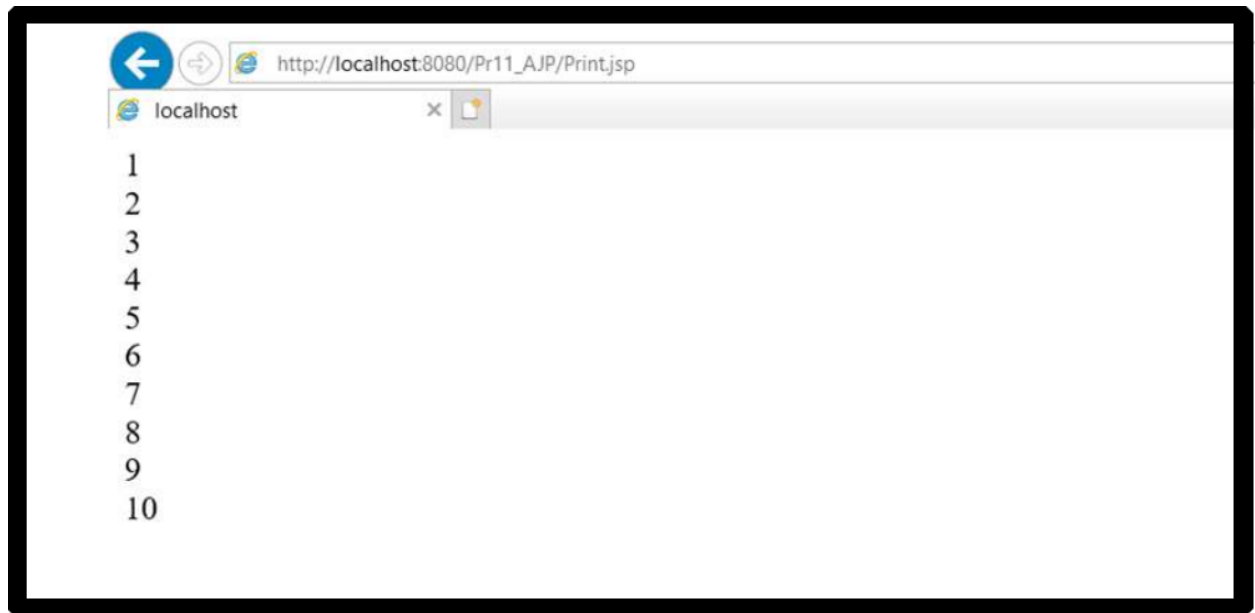
SOURCE CODE practical 11 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form action="Print.jsp">
      <input type="submit" value="Print">
    </form>
  </body>
</html>
```

[FILE NAME : Print.jsp]

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="o"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    <o:forEach var="i" begin="1" end="10">
      <o:out value="${i}"/>
      <br/>
    </o:forEach>
  </body>
</html>
```

OUTPUT practical 11 :

PRACTICAL # 12 : Create a custom JSP tag that prints current date and time. Use this tag into JSP page.

SOURCE CODE practical 12 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
    </head>
    <body>
        <form action="GetTime.jsp">
            <br/>
            <input type="submit" value="Get Time ">
        </form>
    </body>
</html>
```

[FILE NAME : GetTime.jsp]

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="/WEB-INF/tlds/Library" prefix="o" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
    </head>
    <body>
        <%
            String username=request.getParameter("username");
        %>
        <o:printtime/>
    </body>
</html>
```


[FILE NAME : Library.tld]

```

<?xml version="1.0" encoding="UTF-8"?>
<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
jsptaglibrary_2_1.xsd">
    <tlib-version>1.0</tlib-version>
    <short-name>library</short-name>
    <uri>/WEB-INF/tlds/Library</uri>
    <tag>
        <name> printtime </name>
        <tag-class> aa.PrintTimeandDate </tag-class>
        <body-content> empty </body-content>
    </tag>
</taglib>

```

[FILE NAME : PrintTimeandDate.java]

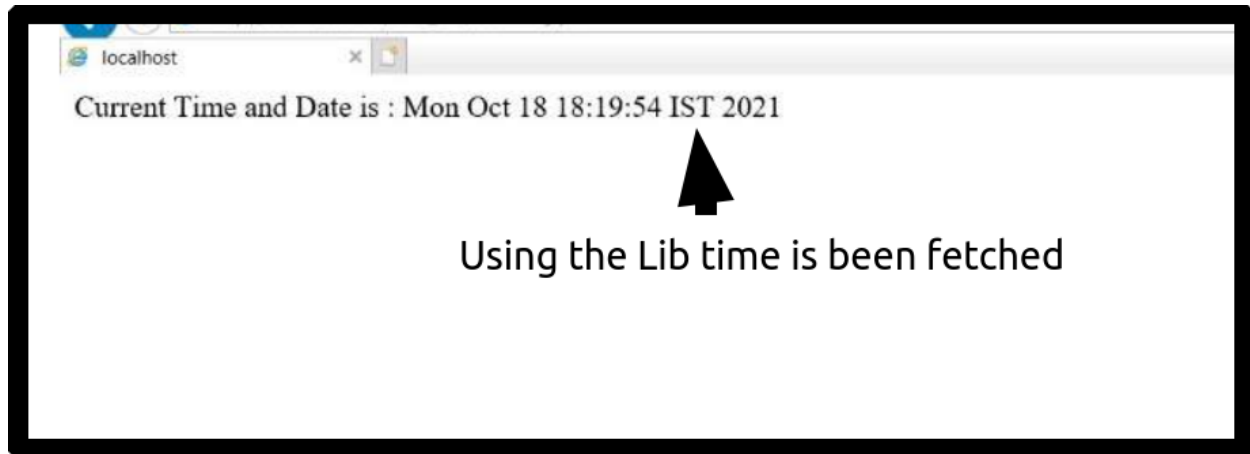
```

package aa;

import java.io.IOException;
import java.util.Calendar;
import java.util.Date;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.SimpleTagSupport;

public class PrintTimeandDate extends SimpleTagSupport {
    public void doTag() throws IOException, JspException {
        JspWriter ob=getJspContext().getOut();
        ob.print("Current Time and Date is : "+new Date());
    }
}

```

OUTPUT practical 12 :

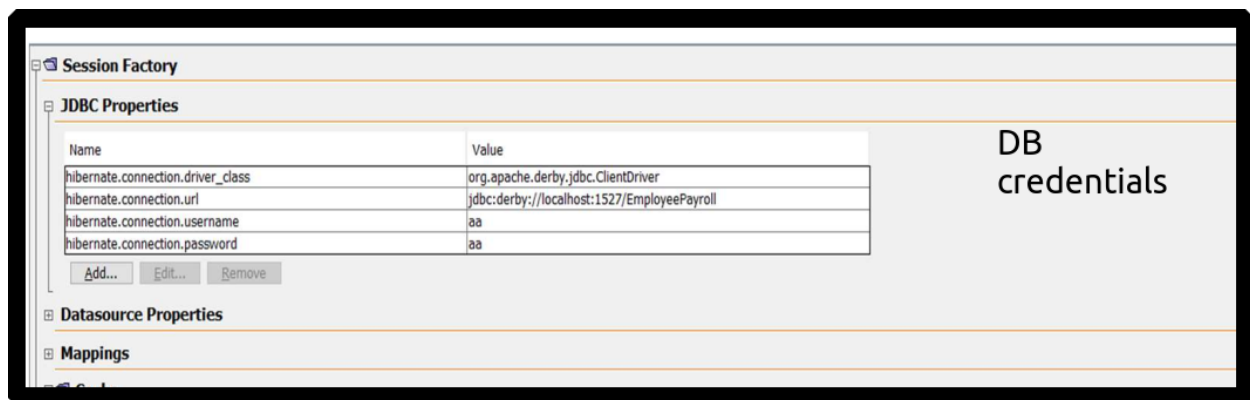
PRACTICAL # 13 : Create a Hibernate application for employee payroll system

SOURCE CODE practical 13 :

[FILE NAME : index.html]

```
<!DOCTYPE html>
<html>
    <head>
        <title>Employee Payroll </title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="Servlet1">
            <input type="submit" value="Get Data ">
        </form>
    </body>
</html>
```

[FILE NAME : hibernate.cfg.xml]



[FILE NAME : hibernate.hbm.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="p1.employeebean" table="EMPLOYEE">
        <id name="empid" type="int" >
            <column name="EMPID"></column>
        </id>
        <property name="name" type="string" >
            <column name="EMPNAME"></column>
        </property>
        <property name="address" type="string">
            <column name="EMPADDRESS"></column>
        </property>
    </class>
</hibernate-mapping>
```

```

        </property>
    </class>
</hibernate-mapping>
[FILE NAME :employeebean.java]

```

```

package p1;
public class employeebean {
    private int empid;
    private String name,address;

    public int getEmpid() {
        return empid;
    }
    public void setEmpid(int empid) {
        this.empid = empid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}

```

[FILE NAME : HibernateUtil.java]

```

package p1;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static {
        try {
            sessionFactory = new
                Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {

```

```

        return sessionFactory;
    }
    public static void ShutDown(){
        sessionFactory.close();
    }
}

```

[FILE NAME : HibernateService.java]

```
package p1;
```

```
import java.util.List;
import org.hibernate.Query;
import org.hibernate.Session;
```

```
public class HibernateService {
    public static employeebean getEmployeeById(int id){
        Session s=HibernateUtil.getSessionFactory().openSession();
        employeebean ob=(employeebean)s.get(employeebean.class,id);
        return ob;
    }
    public static void saveRecord(employeebean ob){
        Session s=HibernateUtil.getSessionFactory().openSession();
        s.beginTransaction();
        s.save(ob);
        s.getTransaction().commit();
    }
    public static void deleteRecord(){
        Session s=HibernateUtil.getSessionFactory().openSession();
        Query q=s.createQuery("delete from employeebean where empid=1001");
        s.beginTransaction();
        q.executeUpdate();
        s.getTransaction().commit();
    }
    public static List<employeebean> getRecords(){
        List<employeebean> ob;
        Session s=HibernateUtil.getSessionFactory().openSession();
        ob=s.createQuery("from employeebean").list();
        return ob;
    }
}

```

[FILE NAME : Servlet1.java]

```
package p1;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.servlet.ServletException;
```

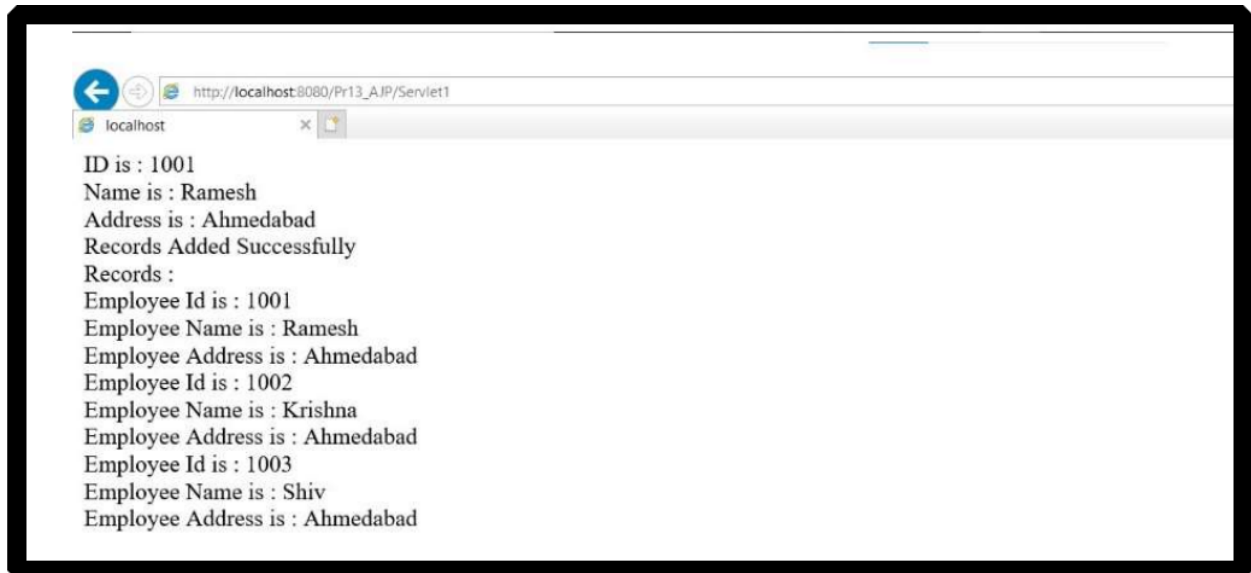
```

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class Servlet1 extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            employeebean ob=HibernateService.getEmployeeByld(1001);
            out.print(" ID is : "+ob.getEmpid());
            out.print("<br>");
            out.print(" Name is : "+ob.getName());
            out.print("<br>");
            out.print(" Address is : "+ob.getAddress());
            employeebean o=new employeebean();
            o.setEmpid(1002);
            o.setName("Krishna");
            o.setAddress("Ahmedabad");
            HibernateService.saveRecord(o);
            o.setEmpid(1003);
            o.setName("Shiv");
            o.setAddress("Ahmedabad");
            HibernateService.saveRecord(o);
            out.print("<br> Records Added Successfully ");
            out.print("<br>Records : ");
            List<employeebean> obj=HibernateService.getRecords();
            for(employeebean i:obj) {
                out.print("<br> Employee Id is : "+i.getEmpid());
                out.print("<br> Employee Name is : "+i.getName());
                out.print("<br> Employee Address is : "+i.getAddress());
            }
            HibernateUtil.ShutDown();
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {

```

```
        return "Short description";  
    }  
}
```

OUTPUT practical 13 :



PRACTICAL # 14 : Create a “Hello World ” application using Spring MVC framework

SOURCE CODE practical 14 :

[FILE NAME : redirect.jsp]

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<% response.sendRedirect("index.htm"); %>
```

[FILE NAME : dispatcher-servlet.xml]

```
<?xml version='1.0' encoding='UTF-8' ?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">
<bean
class="org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping"/>
<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
        <props>
            <prop key="index.htm">indexController</prop>
        </props>
    </property>
</bean>
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />
<bean name="indexController"
class="org.springframework.web.servlet.mvc.ParameterizableViewController" p:viewName="index" />
</beans>
```

[FILE NAME : web.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <context-param>
        <param-name>contextConfigLocation</param-name>
```



```

        <param-value>/WEB-INF/applicationContext.xml</param-value>
    </context-param>
    <listener>
        <listener-class>
            org.springframework.web.context.ContextLoaderListener
        </listener-class>
    </listener>
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>2</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>dispatcher</servlet-name>
        <url-pattern>*.htm</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>30</session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>redirect.jsp</welcome-file>
    </welcome-file-list>
</web-app>
[FILE NAME : index.jsp]
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Spring MVC Framework </title>
    </head>
    <body>
        <p> Hello World </p>
    </body>
</html>

```

OUTPUT practical 14 :

