PRACTICAL NO 1 :

Aim: Develop the presentation layer of Library Management software application with suitable menus.

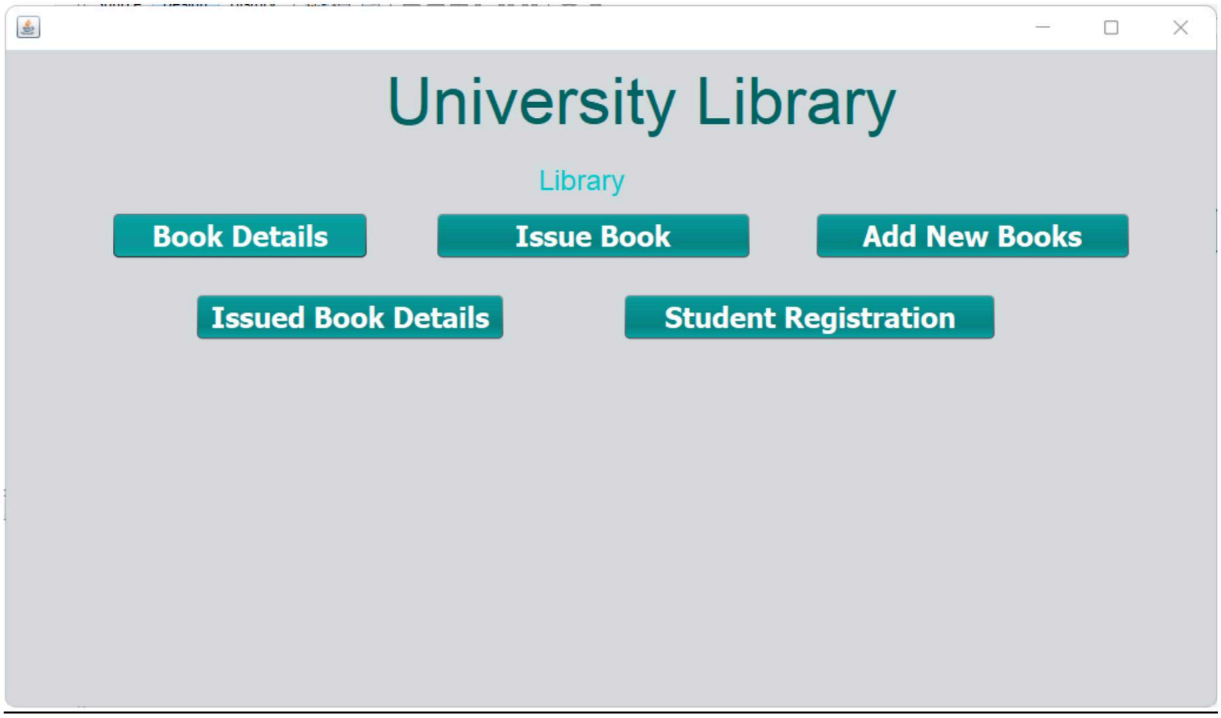
Step 1: Create a java application using IDE --> Click on file --> New Project --> Java --> Java Application -->Give name of the project --> Click on finish.

Step 2: Create a JFrame form inside the source package --> Right click on package --> Select JFrame Form --> Give a name to your JFrame Form --> Then click on finish.

Step 3: Open JFrame Form for designing the interface.

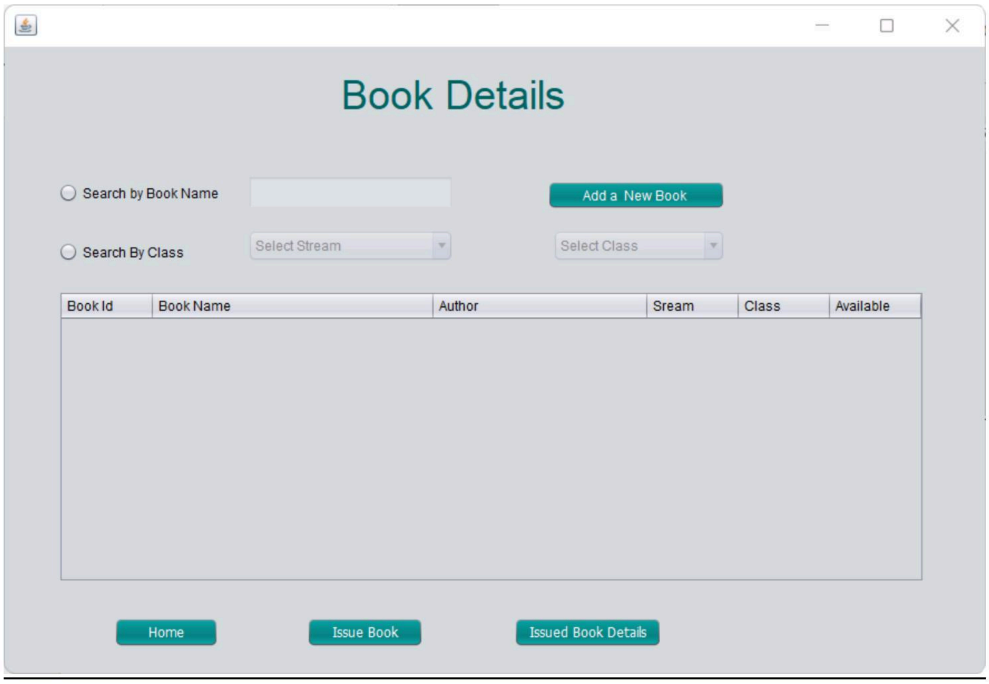
MainLayer.java

Drag and Drop the Swing Palette in the design part which are needed for this frame



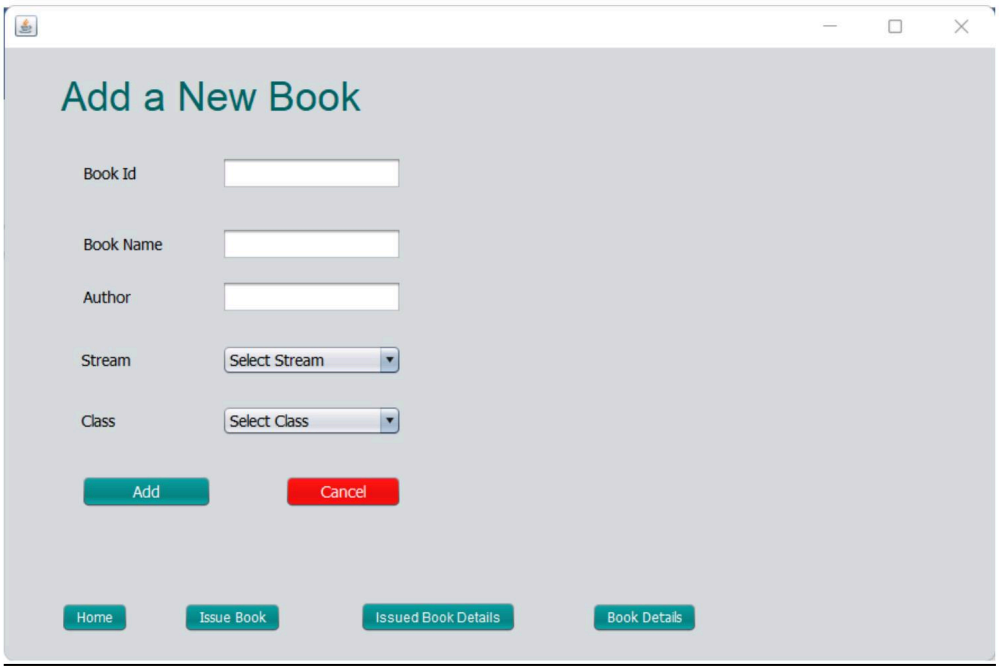
BookDetails.java

Drag and Drop the Swing Palette in the desing part which are needed for this frame



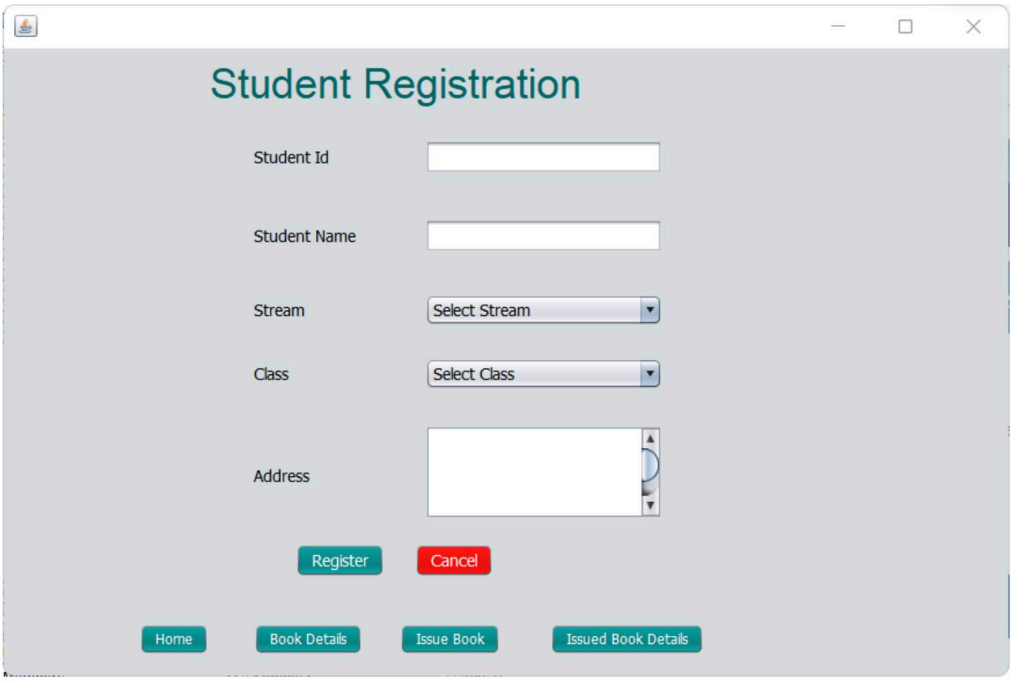
AddNewBook.java

Drag and Drop the Swing Palette in the desing part which are needed for this frame



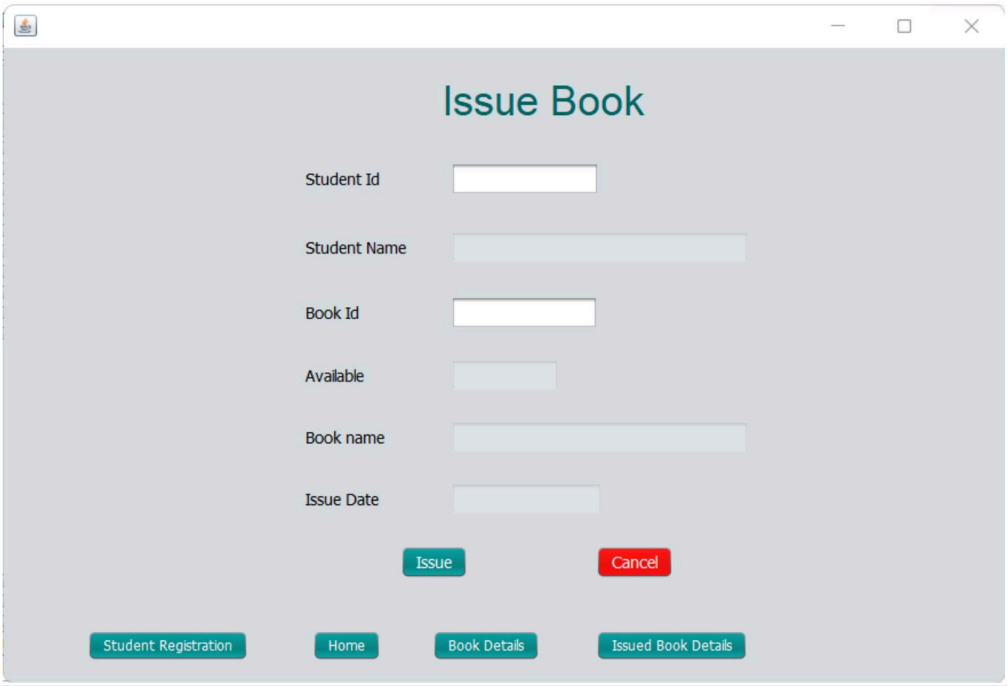
RegisterStudent.java

Drag and Drop the Swing Palette in the desing part which are needed for this frame



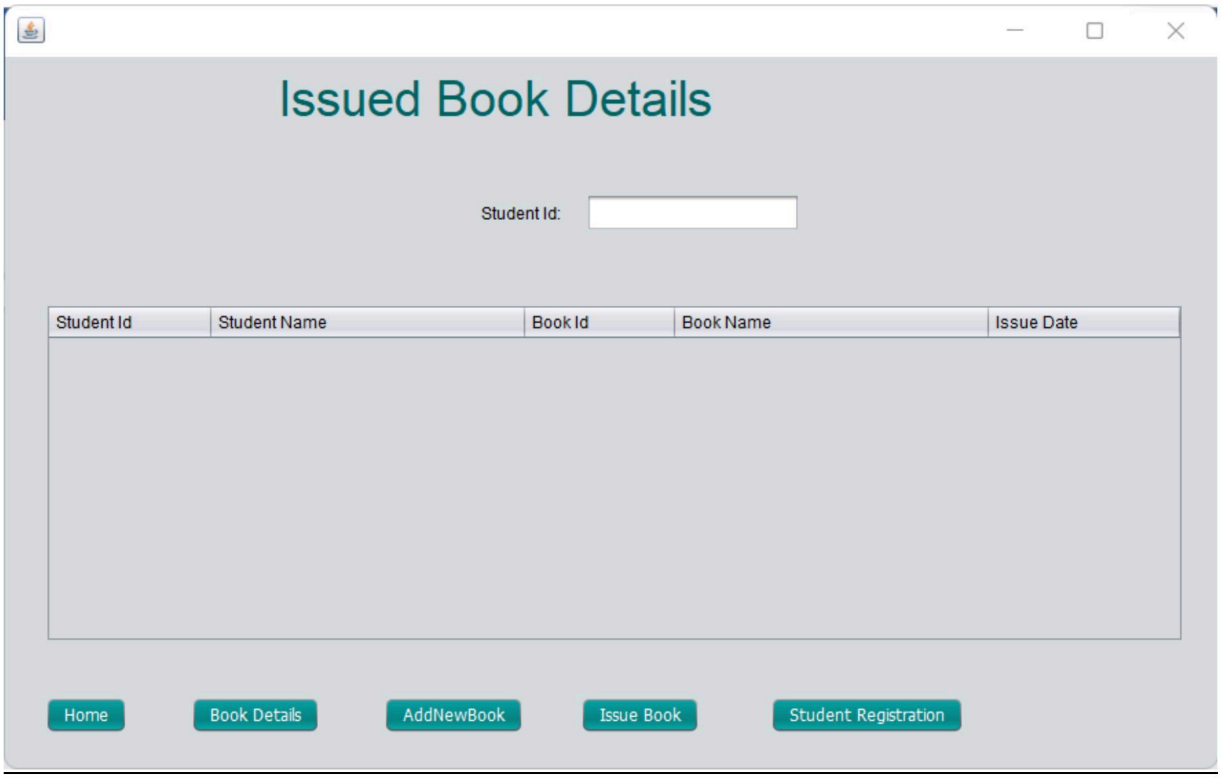
IssueBook.java

Drag and Drop the Swing Palette in the desing part which are needed for this frame



IssuedBook.java

Drag and Drop the Swing Palette in the desing part which are needed for this frame



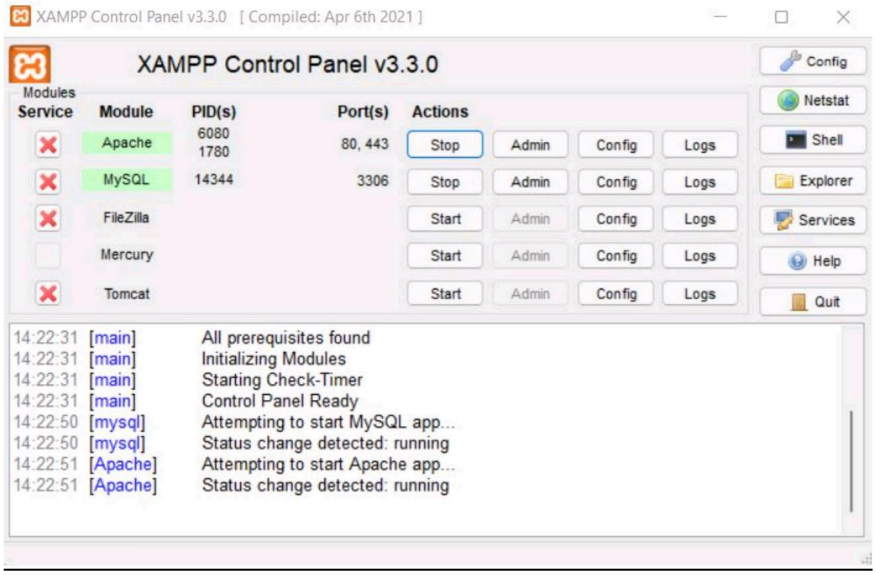
Conclusion: Successfully implemented presentation layer of Library management software using java.

PRACTICAL NO 2:

Aim: Design suitable database for Library Management System.

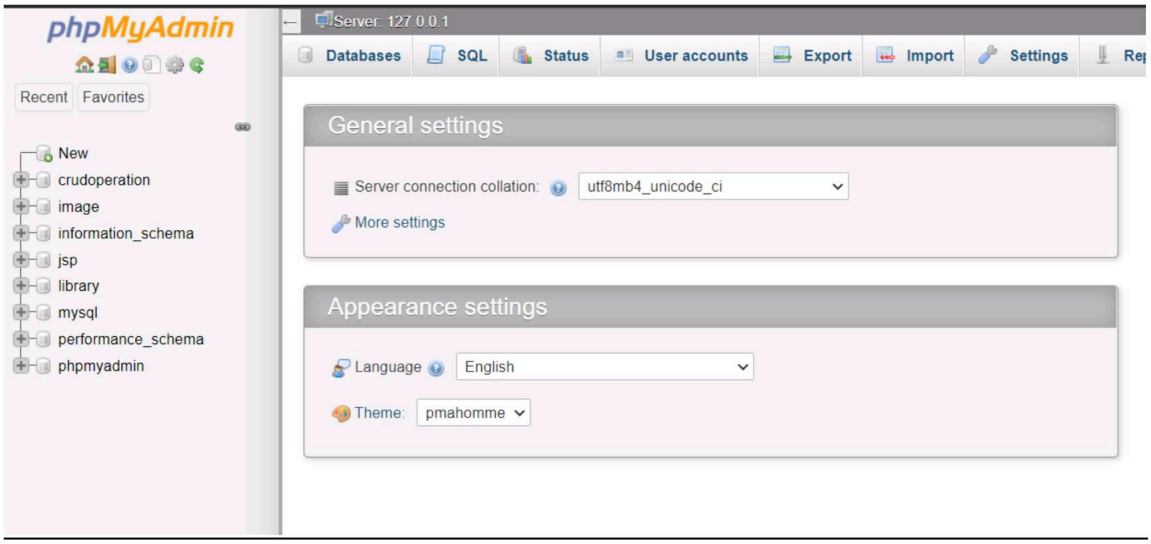
Step 1: Download and install xampp

Step 2: Start apache and mysql

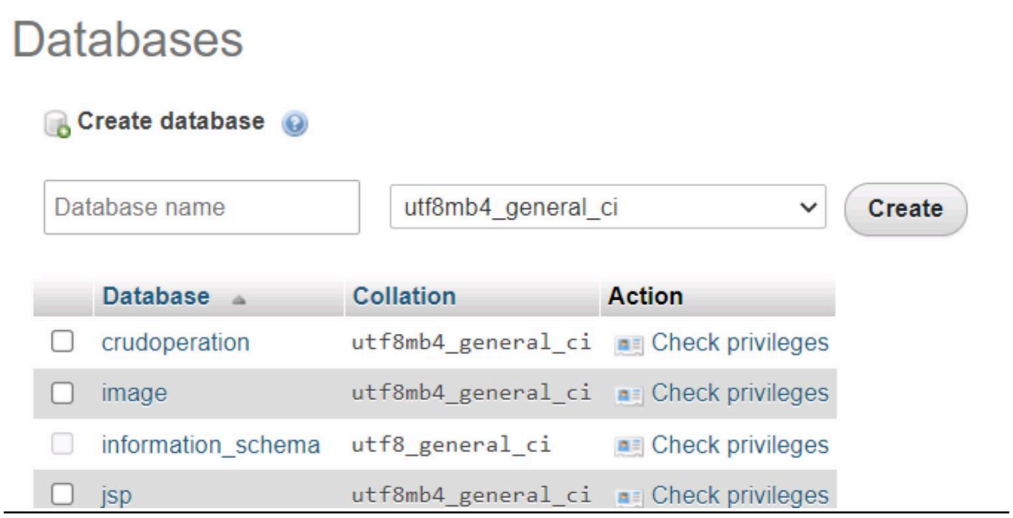


Step 3: Visit to 127.0.0.1 for xampp page and clicl on phpMyadmin to create a database

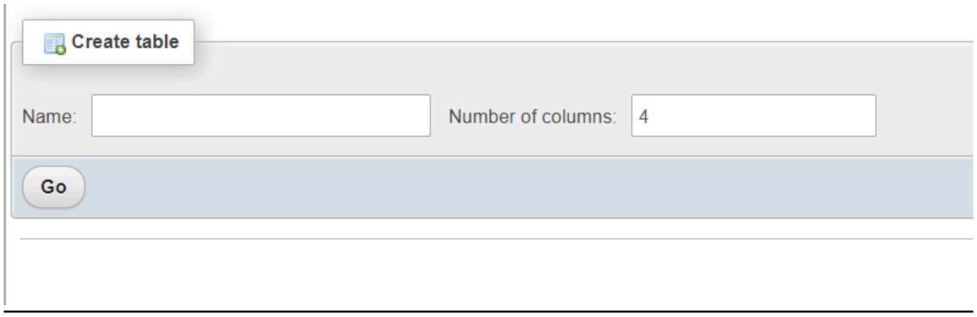
Step 4: On left side of the screen all the database name are their click on new to create a new database



Step 5: Enter database name and click on create to create a new database



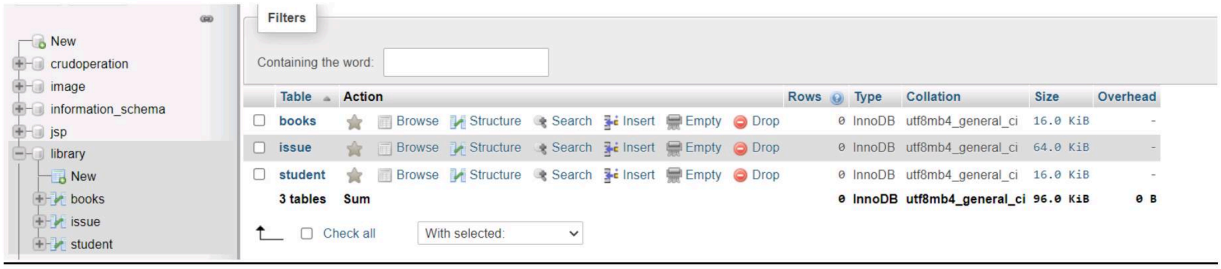
Step 6: Then enter your table name and number of columns needed and then click on GO



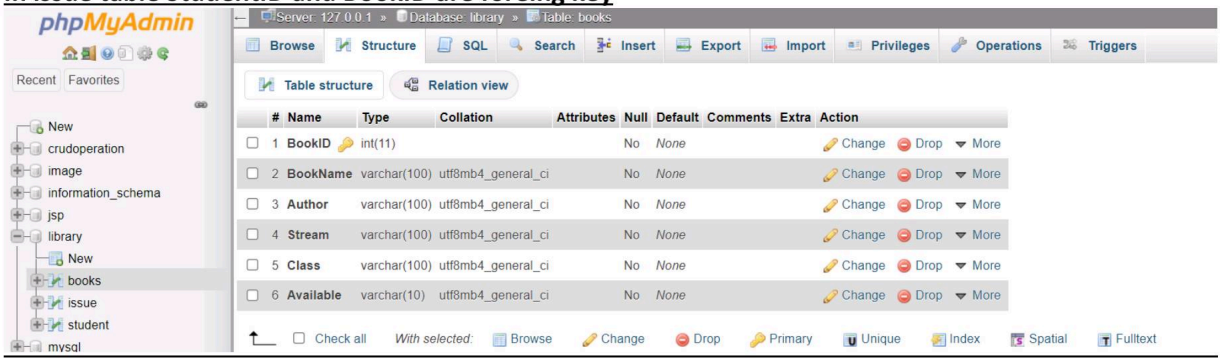
Step7: Then eneter name of the column in name section --> add proper data type next to it --> then give appropriate length values --> If you want to make it primary key click on null_index their you will

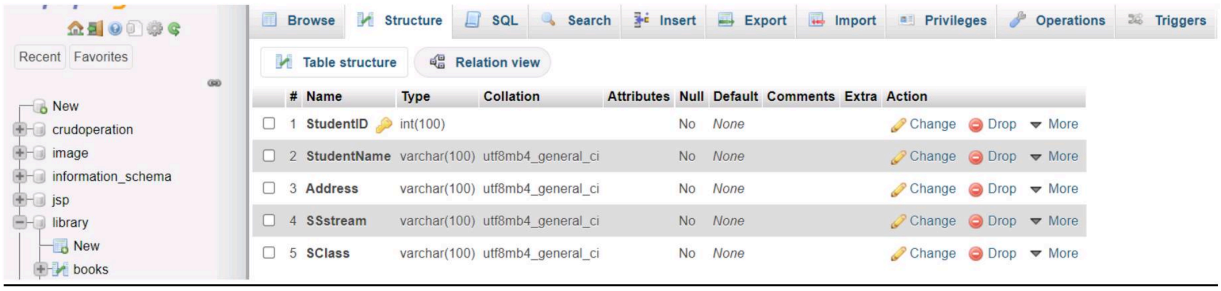
get that option --> If you want to make that column auto increment then check on that A_I option. (Note: You can keep all the columns as it is except name,type and length values).

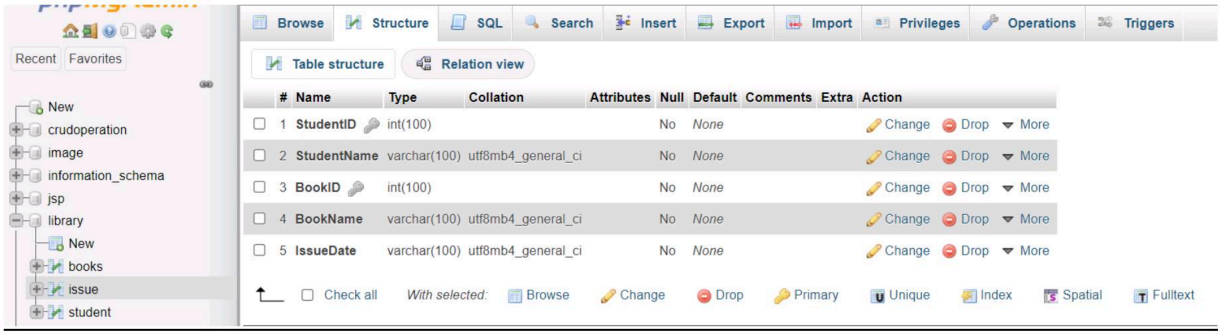
Step 8: Create appropriate database and table for library management software, For adding one then more table in database expand you database and click on new.



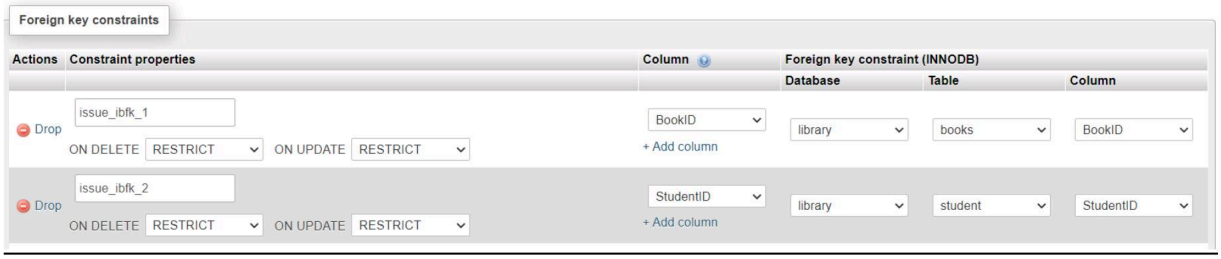
Step 9: As per presentation layer in practical 1 following database and tables are created.
In books table BookID is a primary key
In student table StudentID is a primary key
In issue table StudentID and BookID are foreing key







Step 10: To add foreign key click on realtion view on the top the window then follow the below steps



Keep Constraint properties to default values --> In 1st column select column which you want to make foreign key --> In database select database where reference table is stored --> In table select the reference table name -->In column select the reference column.

Conclusion: Successfully created a database for Library management software.

PRACTICAL NO 4:

Aim: Develop Java application to store image in a database as well as retrieve image from database.

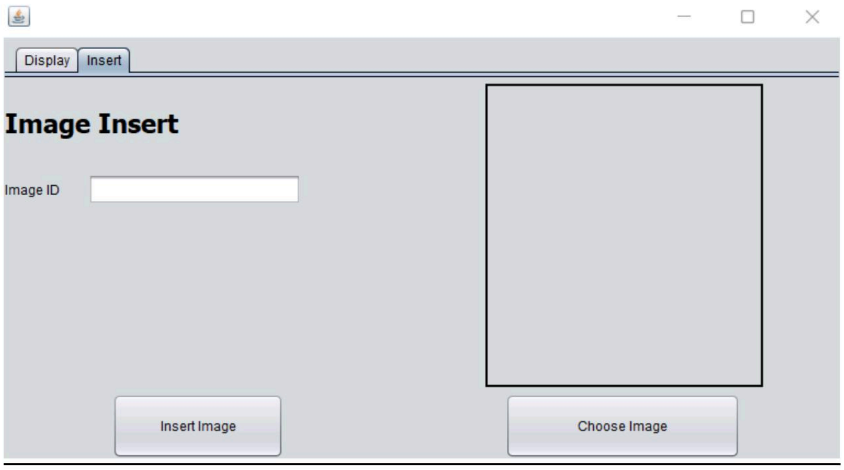
Step 1: Create a java application --> Add one JFrame form and java class file.

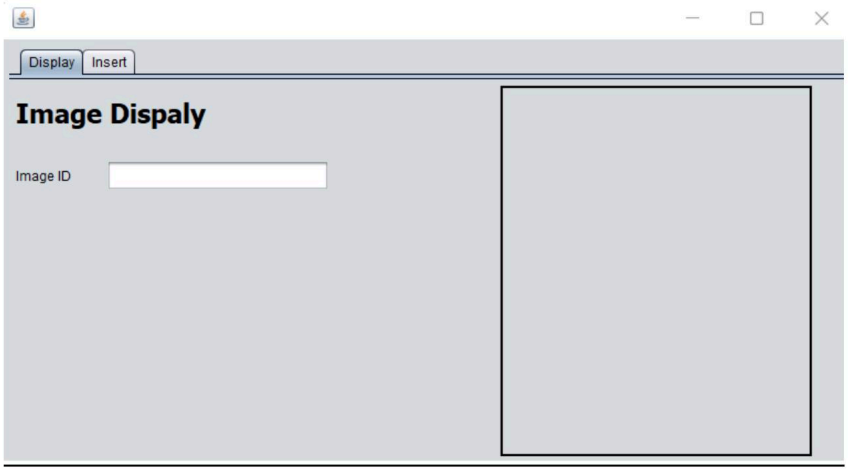
Step 2: Start xampp --> Add database with one table of two columns --> Keep data type of one column as integer to store id and keep data type as BLOB to store image in it.

Step 3: Expand your project and add Mysql JDBC library in library folder.

Step 4: Click on services in IDE --> Right click on database --> Click on New connection --> Select Mysql(connector/Jdriver --> Then one window will appear in which you have to enter your database name and test the connection, If connection is established click on finish.

Step 5: Make the design in Jframe form as shown in the bleow snapshots.





Step 6: Now write the below code for this JFrame Form

Main.java

```
package image;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import java.awt.Image;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


public class Main extends javax.swing.JFrame {
private ImageIcon format=null;
  String fname=null;
  int s=0;
  byte[] pimage=null;
  Connection conn=null;

   public Main() {
      initComponents();
      conn=DBConnect.connect();
   }

   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {

      jTabbedPane1 = new javax.swing.JTabbedPane();
      jPanel1 = new javax.swing.JPanel();
      jLabel3 = new javax.swing.JLabel();
      jLabel4 = new javax.swing.JLabel();
      txtdisid = new javax.swing.JTextField();
      lbldisplay = new javax.swing.JLabel();
      Insert = new javax.swing.JPanel();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      txtid = new javax.swing.JTextField();
      lblimage = new javax.swing.JLabel();
      jButton1 = new javax.swing.JButton();
      jButton2 = new javax.swing.JButton();

      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
    jLabel3.setText("Image Dispaly");

    jLabel4.setText("Image ID");

    txtdisid.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtdisidActionPerformed(evt);
      }
    });
    txtdisid.addKeyListener(new java.awt.event.KeyAdapter() {
      public void keyReleased(java.awt.event.KeyEvent evt) {
        txtdisidKeyReleased(evt);
      }
    });

    lbldisplay.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0), 2));

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 392,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jLabel4)
            .addGap(32, 32, 32)
            .addComponent(txtdisid, javax.swing.GroupLayout.PREFERRED_SIZE, 202,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 43,
Short.MAX_VALUE)
        .addComponent(lbldisplay, javax.swing.GroupLayout.PREFERRED_SIZE, 282,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(33, 33, 33))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(lbldisplay, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 49,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(jLabel4)
                  .addComponent(txtdisid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(0, 240, Short.MAX_VALUE)))
            .addContainerGap())
        );

        jTabbedPane1.addTab("Display", jPanel1);

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel1.setText("Image Insert");

        jLabel2.setText("Image ID");

        txtid.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtidActionPerformed(evt);
            }
        });

        lblimage.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0), 2));

        jButton1.setText("Choose Image");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setText("Insert Image");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout InsertLayout = new javax.swing.GroupLayout(Insert);
        Insert.setLayout(InsertLayout);
        InsertLayout.setHorizontalGroup(
            InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(InsertLayout.createSequentialGroup()
                .addGap(98, 98, 98)
```

```
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 214,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(90, 90, 90))
          .addGroup(InsertLayout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addGroup(InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 210,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGroup(InsertLayout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(26, 26, 26)
                .addComponent(txtid, javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(168, 168, 168)
            .addComponent(lblimage, javax.swing.GroupLayout.PREFERRED_SIZE, 253,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(69, 69, 69))
        );
        InsertLayout.setVerticalGroup(
          InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(InsertLayout.createSequentialGroup()
            .addGroup(InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(InsertLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblimage, javax.swing.GroupLayout.PREFERRED_SIZE, 276,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(InsertLayout.createSequentialGroup()
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(jLabel2)
                  .addComponent(txtid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(InsertLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 59, Short.MAX_VALUE)
              .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        );

        jTabbedPane1.addTab("Insert", Insert);
```

```
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jTabbedPane1)
            .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jTabbedPane1)
            .addContainerGap())
    );

    pack();
}// </editor-fold>

private void txtidActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // using this code you can brows image and image set to lable
    JFileChooser fchoser=new JFileChooser();
    fchoser.showOpenDialog(null);
    File f=fchoser.getSelectedFile();
    fname=f.getAbsolutePath();
    ImageIcon micon=new ImageIcon(fname);
    try {
        File image=new File(fname);
        FileInputStream fis=new FileInputStream(image);
        ByteArrayOutputStream baos=new ByteArrayOutputStream();
        byte[] buf=new byte[1024];
        for(int readnum; (readnum=fis.read(buf)) !=-1;)
        {
            baos.write(buf,0,readnum);
        }
        pimage=baos.toByteArray();
        lblimage.setIcon(resizeImage(fname, buf));
    } catch (Exception e) {
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // This code use to insert image to database
```

```
    String id=txtid.getText();
    try {
        String q= "INSERT INTO `img`(`id`, `image`) VALUES (?,?)";
         PreparedStatement pst=conn.prepareStatement(q);
          pst.setString(1, id);
          pst.setBytes(2, pimage);
          pst.execute();
    } catch (Exception e) {
    }
  }

  private void txtdisidActionPerformed(java.awt.event.ActionEvent evt) {

  }

  private void txtdisidKeyReleased(java.awt.event.KeyEvent evt) {
    //code for displaying the image
     try {
      String sql="SELECT `id`, `image` FROM `img` WHERE id='"+txtdisid.getText()+"'";
      PreparedStatement pst=conn.prepareStatement(sql);
      ResultSet rs=pst.executeQuery();
      if(rs.next())
      {
         byte[] imagedata=rs.getBytes("image");
         format=new ImageIcon(imagedata);
         Image mm=format.getImage();
         Image img2=mm.getScaledInstance(lbldisplay.getWidth(), lbldisplay.getHeight(),
Image.SCALE_SMOOTH);
         ImageIcon image=new ImageIcon(img2);
         lbldisplay.setIcon(image);
       }
      else
      {

      }
    } catch (Exception e) {

    }
  }

  public ImageIcon resizeImage(String imagePath, byte[] pic){
     // This code use to resize image to fit lable
    ImageIcon myImage=null;
    if(imagePath !=null)
    {
    myImage=new ImageIcon(imagePath);

    }else{
```

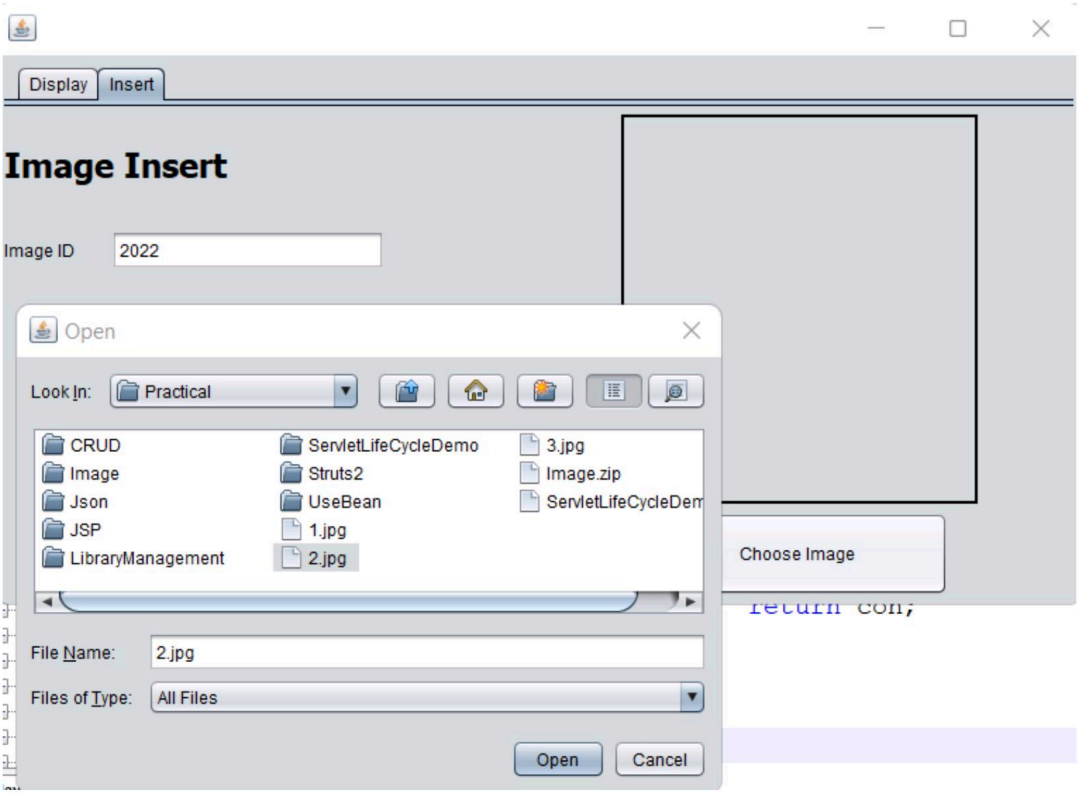
```
        myImage=new ImageIcon(pic);
      }

      Image img=myImage.getImage();
      Image img2=img.getScaledInstance(lblimage.getHeight(), lblimage.getWidth(),
Image.SCALE_SMOOTH);
      ImageIcon image=new ImageIcon(img2);
      return image;
   }

   /**
    * @param args the command line arguments
    */
   public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
      try {
         for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
               javax.swing.UIManager.setLookAndFeel(info.getClassName());
               break;
            }
         }
      } catch (ClassNotFoundException ex) {
         java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
      } catch (InstantiationException ex) {
         java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
      } catch (IllegalAccessException ex) {
         java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
         java.util.logging.Logger.getLogger(Main.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
      }
      //</editor-fold>

      /* Create and display the form */
      java.awt.EventQueue.invokeLater(new Runnable() {
         public void run() {
            new Main().setVisible(true);
         }
      });
```

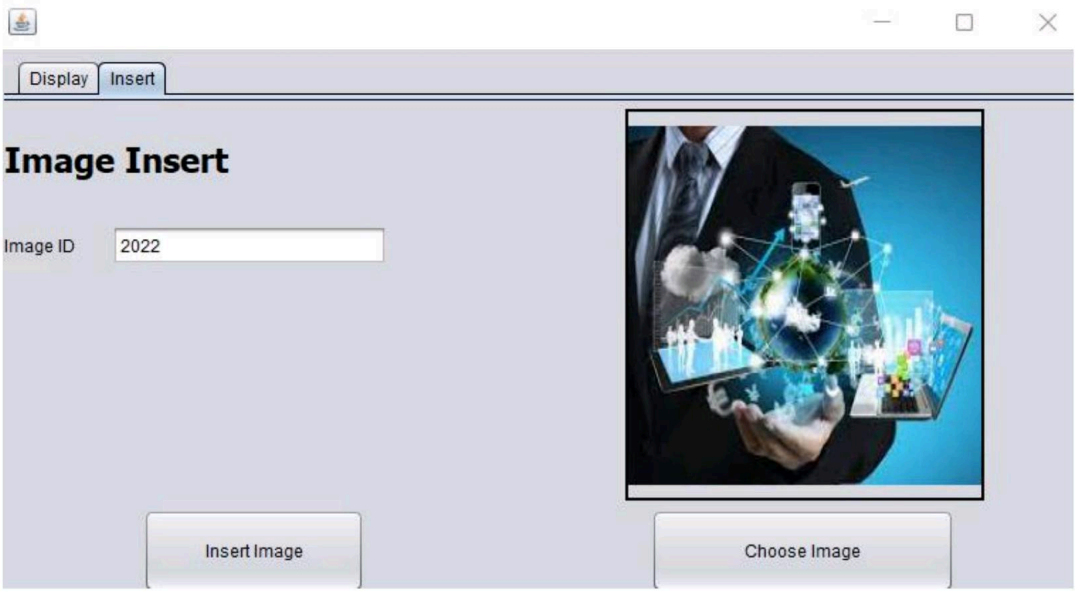
```
    }

    // Variables declaration - do not modify
    private javax.swing.JPanel Insert;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JLabel lbldisplay;
    private javax.swing.JLabel lblimage;
    private javax.swing.JTextField txtdisid;
    private javax.swing.JTextField txtid;
    // End of variables declaration
}
```

DBConnect.java

```
package image;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnect {
    public static Connection connect()
    {
     Connection con=null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/image?","root","");
 }
catch (Exception e)
    {
        System.out.println("inter.DBConnect.connect()");
    }
   return con;
}

}
```

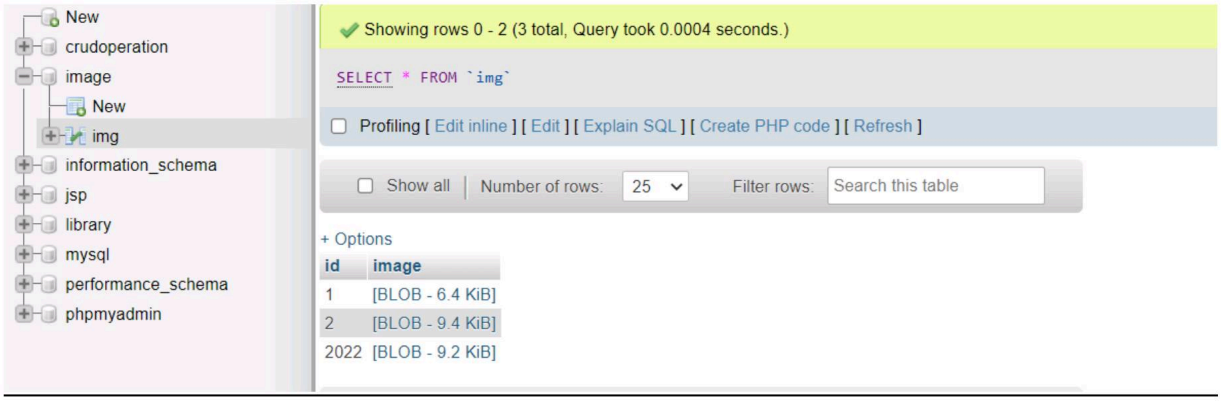
Output:





After clicking on insert button this image will be added in database with id 2022.

As you can after entering the same id on display image side it is displaying the same image which was stored in database.



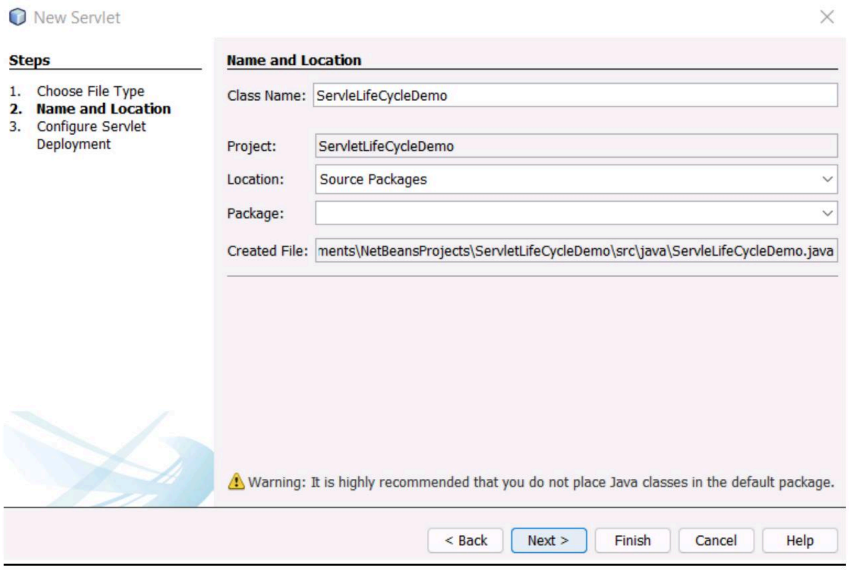
Conclusion: Successfully stored and retrived image from database using java.

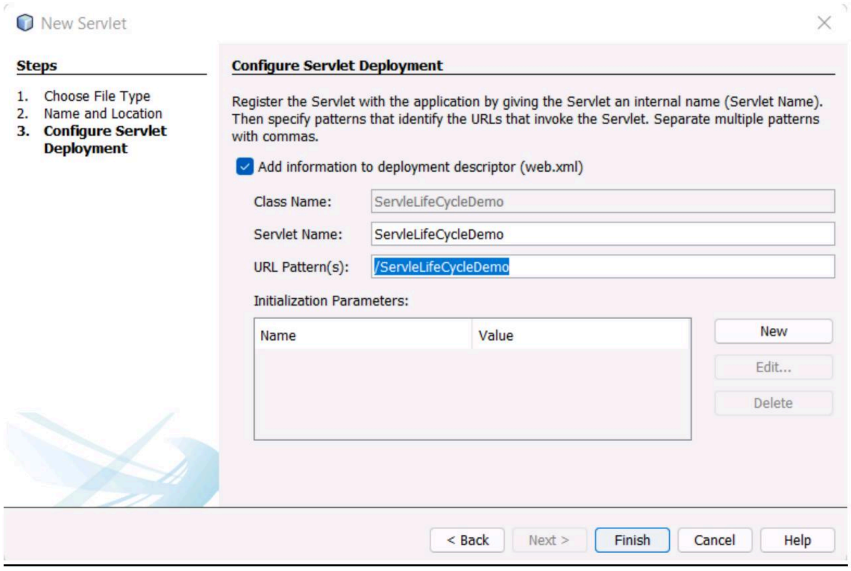
PRACTICAL NO 5:

Aim: Write a Java Application to demonstrate servlet life cycle.

Step 1: Create java a web application --> Click on file --> New Project --> Java Web --> Web Application -->Give name of the project --> Select Server --> Click on finish.

Step 2: Create a Servlet file inside the source package --> Right click on package --> Select servlet --> Give a name to your servlet file --> Check on the deployment descripter (web.xml) --> Note down the url pattern written over their --> Then click on finish.





Step 3: Write the servlet life cycle code inside servlet file you have created. (Note: Their will be some auto-generated code in servlet file keep only that blocks of code which is needed).

ServletLifeCycleDemo.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;


public class ServletLifeCycleDemo extends HttpServlet
{

   public void init() throws ServletException
   {
      System.out.println("Servlet has been Initialized...");
   }


   public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException
   {
      PrintWriter out = res.getWriter();
      out.println("Servlet Life Cycle Has three stages \n. Initialization \n. Service \n. Destroy");
      System.out.println("Servlet started servicing...");
   }


   public void destroy()
   {
      System.out.println("Servlet has been Destroyed...");
   }
}
```

Step 4: Write the html page code inside index.html file present inside Web Pages folder.

Index.html

```
  <html>
 <head>
      <title>Servlet Life Cycle</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <form action="ServletLifeCycleDemo" method="get">
         <input type="submit" value="Click Here">
      </form>
   </body>
</html>
```

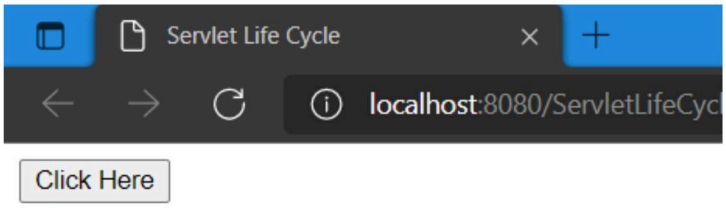
Step 5: Make sure you keep the url name same inside html file as it is written inside web.xml file for that url, web.xml file is present inside configuration file.

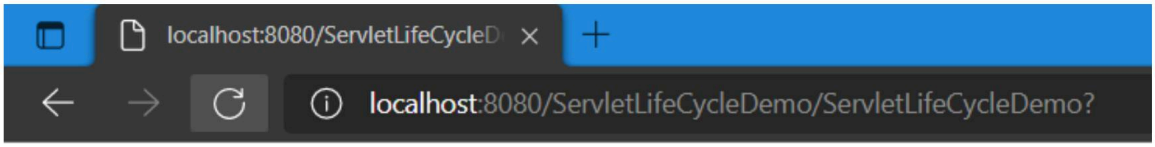
Web.xml

```
<?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">
   <servlet>
      <servlet-name>ServletLifeCycleDemo</servlet-name>
      <servlet-class>ServletLifeCycleDemo</servlet-class>
   </servlet>
   <servlet-mapping>
      <servlet-name>ServletLifeCycleDemo</servlet-name>
      <url-pattern>/ServletLifeCycleDemo</url-pattern>
   </servlet-mapping>
</web-app>
```

Output:

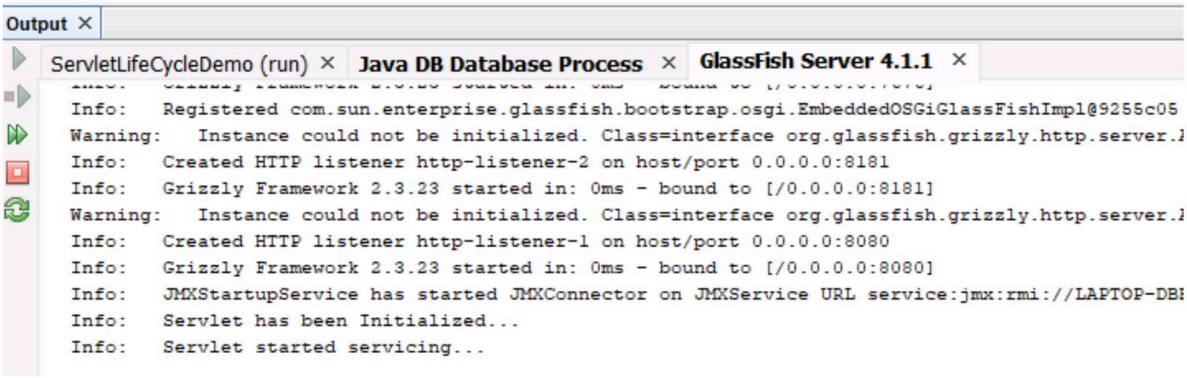
After Clicking on button as you can see appropriate output is displayed to the user.

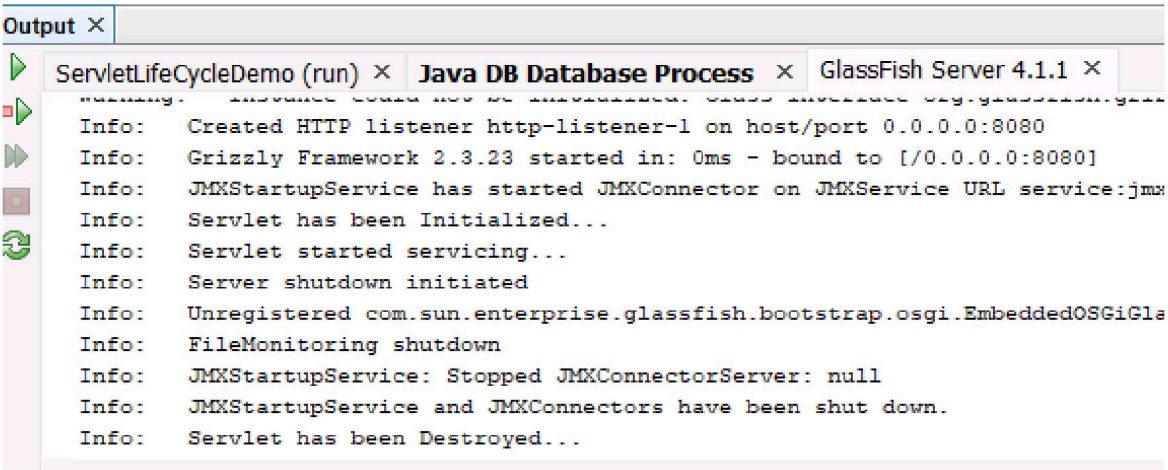





```
Servlet Life Cycle Has three stages
. Initialization
. Service
. Destroy
```

As you can see below simultaneously when servlet is running its init,service and destroy methods are also giving us the output in terminal.





Conclusion: Successfully implemented Servlet Life Cycle using java.

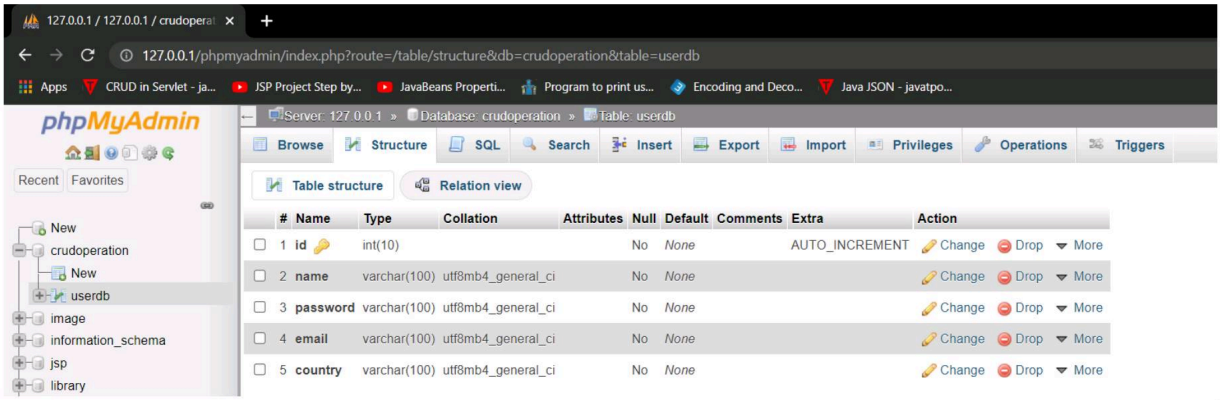
PRACTICAL NO 6:

Aim: Design database for employee administration. Develop servlet(s) to perform CRUD operations.

Step 1: Create a java web application --> Create different Servlet files and 2 java class file.

Step 2: Expand your project and add Mysql JDBC library in libraries folder.

Step 3: Create database on xampp.



Step 4: Click on services in IDE --> Right click on database --> Click on New connection --> Select Mysql(connector/Jdriver) --> Then one window will appear in which you have to enter your database name and test the connection, If connection is established click on finish.

Step 5: In servlet file write the codes for CRUD operation.

Step 6: Inside one class file write database connection and CRUD statements and in other class file write getters and setters code variables.

Emp.java

```
public class Emp
{
private int id;
private String name,password,email,country;

   public int getId() {
      return id;
   }

   public void setId(int id) {
      this.id = id;
```

```
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}
```

EmpDao.java

```
import java.util.*;
import java.sql.*;

public class EmpDao
{
    public static Connection getConnection(){
        Connection con=null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/crudoperation?","root","");
        }
```

```
    catch(Exception e)
    {System.out.println(e);}
    return con;
  }
  public static int save(Emp e){
    int status=0;
    try{
      Connection con=EmpDao.getConnection();
      PreparedStatement ps=con.prepareStatement("insert into userdb(name,password,email,country)
values (?,?,?,?)");
      ps.setString(1,e.getName());
      ps.setString(2,e.getPassword());
      ps.setString(3,e.getEmail());
      ps.setString(4,e.getCountry());

      status=ps.executeUpdate();

      con.close();
    }catch(Exception ex){ex.printStackTrace();}

    return status;
  }
  public static int update(Emp e){
    int status=0;
    try{
      Connection con=EmpDao.getConnection();
      PreparedStatement ps=con.prepareStatement("update userdb set
name=?,password=?,email=?,country=? where id=?");
      ps.setString(1,e.getName());
      ps.setString(2,e.getPassword());
      ps.setString(3,e.getEmail());
      ps.setString(4,e.getCountry());
      ps.setInt(5,e.getId());

      status=ps.executeUpdate();

      con.close();
    }catch(Exception ex){ex.printStackTrace();}

    return status;
  }
  public static int delete(int id){
    int status=0;
    try{
      Connection con=EmpDao.getConnection();
      PreparedStatement ps=con.prepareStatement("delete from userdb where id=?");
      ps.setInt(1,id);
      status=ps.executeUpdate();
```

```
            con.close();
        }catch(Exception e){e.printStackTrace();}

        return status;
    }
    public static Emp getEmployeeById(int id){
        Emp e=new Emp();

        try{
            Connection con=EmpDao.getConnection();
            PreparedStatement ps=con.prepareStatement("select * from userdb where id=?");
            ps.setInt(1,id);
            ResultSet rs=ps.executeQuery();
            if(rs.next()){
                e.setId(rs.getInt(1));
                e.setName(rs.getString(2));
                e.setPassword(rs.getString(3));
                e.setEmail(rs.getString(4));
                e.setCountry(rs.getString(5));
            }
            con.close();
        }catch(Exception ex){ex.printStackTrace();}

        return e;
    }
    public static List<Emp> getAllEmployees(){
        List<Emp> list=new ArrayList<Emp>();

        try{
            Connection con=EmpDao.getConnection();
            PreparedStatement ps=con.prepareStatement("select * from userdb");
            ResultSet rs=ps.executeQuery();
            while(rs.next()){
                Emp e=new Emp();
                e.setId(rs.getInt(1));
                e.setName(rs.getString(2));
                e.setPassword(rs.getString(3));
                e.setEmail(rs.getString(4));
                e.setCountry(rs.getString(5));
                list.add(e);
            }
            con.close();
        }catch(Exception e){e.printStackTrace();}
        return list;
    }
}
```

ViewServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
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("/ViewServlet")
public class ViewServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
   {
      response.setContentType("text/html");
      PrintWriter out=response.getWriter();
      out.println("<a href='index.html'>Add New Employee</a>");
      out.println("<h1>Employees List</h1>");

      List<Emp> list=EmpDao.getAllEmployees();

      out.print("<table border='1' width='100%'");

out.print("<tr><th>Id</th><th>Name</th><th>Password</th><th>Email</th><th>Country</th><th>Edit
</th><th>Delete</th></tr>");
      for(Emp e:list){

out.print("<tr><td>"+e.getId()+"</td><td>"+e.getName()+"</td><td>"+e.getPassword()+"</td><td>"+e.
getEmail()+"</td><td>"+e.getCountry()+"</td><td><a
href='EditServlet?id="+e.getId()+"'>edit</a></td><td><a
href='DeleteServlet?id="+e.getId()+"'>delete</a></td></tr>");
      }
      out.print("</table>");
      out.close();
   }
}
```

SaveServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
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("/SaveServlet")
```

```
public class SaveServlet extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
   {
      response.setContentType("text/html");
      PrintWriter out=response.getWriter();

      String name=request.getParameter("name");
      String password=request.getParameter("password");
      String email=request.getParameter("email");
      String country=request.getParameter("country");

      Emp e=new Emp();
      e.setName(name);
      e.setPassword(password);
      e.setEmail(email);
      e.setCountry(country);

      int status=EmpDao.save(e);
      if(status>0){
         out.print("<p>Record saved successfully!</p>");
         request.getRequestDispatcher("index.html").include(request, response);
      }else{
         out.println("Sorry! unable to save record");
      }
      out.close();
   }
}
```

EditServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
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("/EditServlet")
public class EditServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
   {
      response.setContentType("text/html");
      PrintWriter out=response.getWriter();
      out.println("<h1>Update Employee</h1>");
      String sid=request.getParameter("id");
      int id=Integer.parseInt(sid);
```

```
        Emp e=EmpDao.getEmployeeById(id);

        out.print("<form action='EditServlet2' method='post'>");
        out.print("<table>");
        out.print("<tr><td></td><td><input type='hidden' name='id' value='"+e.getId()+"'/></td></tr>");
        out.print("<tr><td>Name:</td><td><input type='text' name='name'
value='"+e.getName()+"'/></td></tr>");
        out.print("<tr><td>Password:</td><td><input type='password' name='password'
value='"+e.getPassword()+"'/></td></tr>");
        out.print("<tr><td>Email:</td><td><input type='email' name='email'
value='"+e.getEmail()+"'/></td></tr>");
        out.print("<tr><td>Country:</td><td>");
        out.print("<select name='country' style='width:150px'>");
        out.print("<option>India</option>");
        out.print("<option>USA</option>");
        out.print("<option>UK</option>");
        out.print("<option>Other</option>");
        out.print("</select>");
        out.print("</td></tr>");
        out.print("<tr><td colspan='2'><input type='submit' value='Edit & Save '/></td></tr>");
        out.print("</table>");
        out.print("</form>");

        out.close();
    }
}
```

EditServlet2.java

```
import java.io.IOException;
import java.io.PrintWriter;

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("/EditServlet2")
public class EditServlet2 extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();

        String sid=request.getParameter("id");
        int id=Integer.parseInt(sid);
```

```
        String name=request.getParameter("name");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String country=request.getParameter("country");

        Emp e=new Emp();
        e.setId(id);
        e.setName(name);
        e.setPassword(password);
        e.setEmail(email);
        e.setCountry(country);

        int status=EmpDao.update(e);
        if(status>0){
            response.sendRedirect("ViewServlet");
        }else{
            out.println("Sorry! unable to update record");
        }
        out.close();
    }
}
```

DeleteServlet.java

```
import java.io.IOException;
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("/DeleteServlet")
public class DeleteServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
    {
        String sid=request.getParameter("id");
        int id=Integer.parseInt(sid);
        EmpDao.delete(id);
        response.sendRedirect("ViewServlet");
    }
}
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Insert title here</title>
```

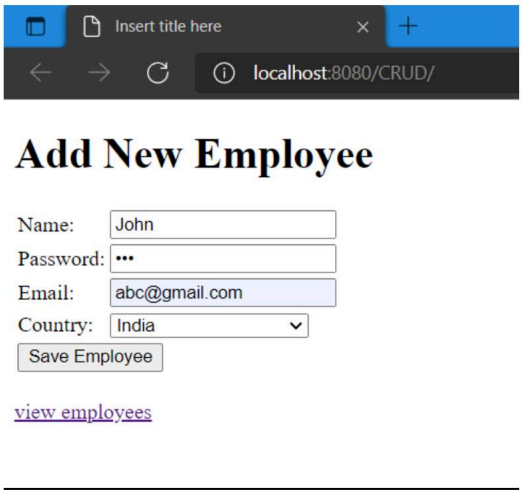
```
</head>
<body>
<h1>Add New Employee</h1>
<form action="SaveServlet" method="post">
<table>
<tr><td>Name:</td><td><input type="text" name="name"/></td></tr>
<tr><td>Password:</td><td><input type="password" name="password"/></td></tr>
<tr><td>Email:</td><td><input type="email" name="email"/></td></tr>
<tr><td>Country:</td><td>
<select name="country" style="width:150px">
<option>India</option>
<option>USA</option>
<option>UK</option>
<option>Other</option>
</select>
</td></tr>
<tr><td colspan="2"><input type="submit" value="Save Employee"/></td></tr>
</table>
</form>
<br/>
<a href="ViewServlet">view employees</a>
</body>
</html>
```

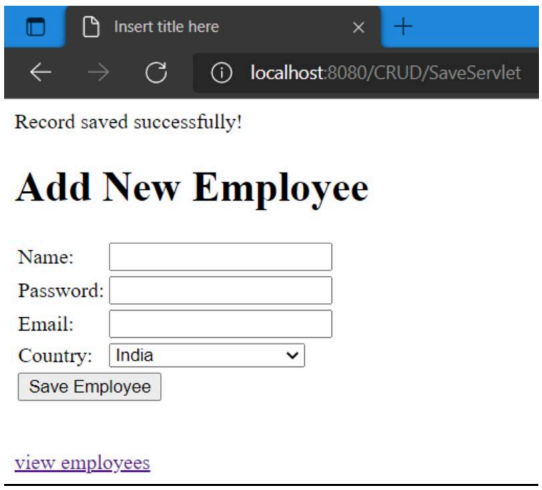
Web.xml

```
<?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">
  <servlet>
    <servlet-name>SaveServlet</servlet-name>
    <servlet-class>SaveServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>EditServlet</servlet-name>
    <servlet-class>EditServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>EditServlet2</servlet-name>
    <servlet-class>EditServlet2</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>DeleteServlet</servlet-name>
    <servlet-class>DeleteServlet</servlet-class>
  </servlet>
  <servlet>
```

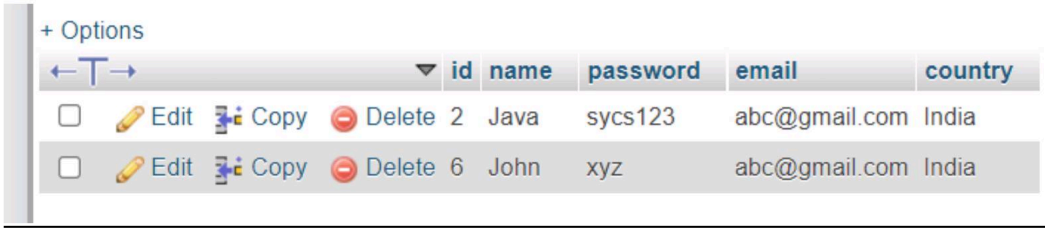
```
    <servlet-name>ViewServlet</servlet-name>
    <servlet-class>ViewServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>SaveServlet</servlet-name>
    <url-pattern>/SaveServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>EditServlet</servlet-name>
    <url-pattern>/EditServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>EditServlet2</servlet-name>
    <url-pattern>/EditServlet2</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>DeleteServlet</servlet-name>
    <url-pattern>/DeleteServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>ViewServlet</servlet-name>
    <url-pattern>/ViewServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
</web-app>
```

Output:





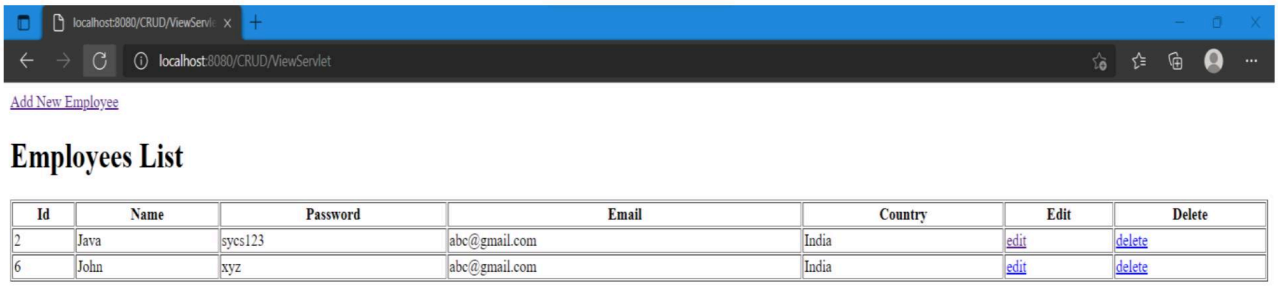
As you can see in above snapshot its showing record is successfully saved lets check whether the data is updated on Xampp or not



+ Options

	id	name	password	email	country
Edit Copy Delete	2	Java	sycs123	abc@gmail.com	India
Edit Copy Delete	6	John	xyz	abc@gmail.com	India

After Clicking on View Employees



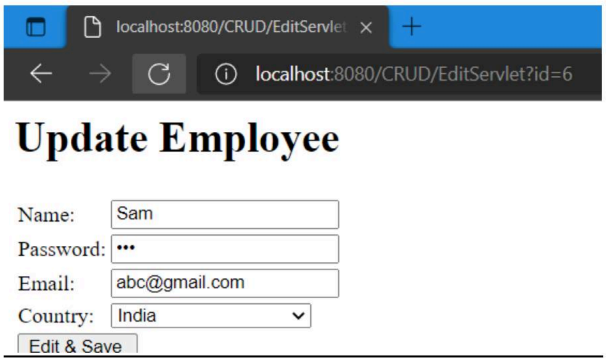
localhost:8080/CRUD/ViewServlet

Add New Employee

Employees List

Id	Name	Password	Email	Country	Edit	Delete
2	Java	sycs123	abc@gmail.com	India	edit	delete
6	John	xyz	abc@gmail.com	India	edit	delete

After clicking on Edit



localhost:8080/CRUD/EditServlet?id=6

Update Employee

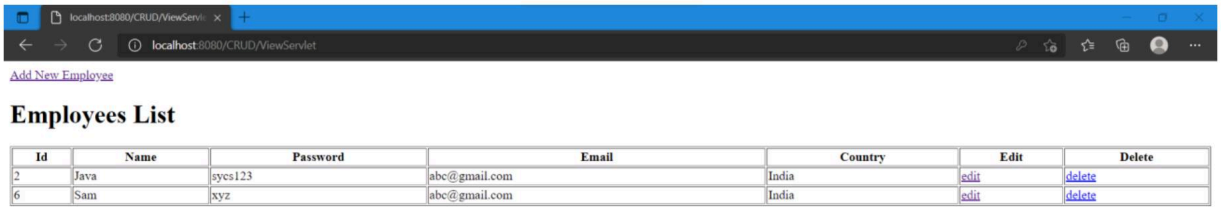
Name: Sam

Password: •••

Email: abc@gmail.com

Country: India

Edit & Save



localhost:8080/CRUD/ViewServlet

Add New Employee

Employees List

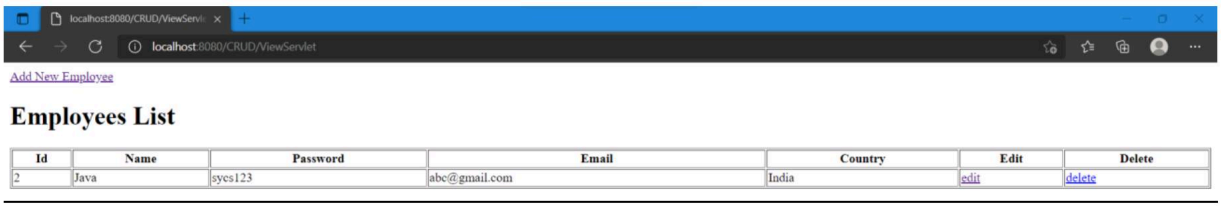
Id	Name	Password	Email	Country	Edit	Delete
2	Java	sycs123	abc@gmail.com	India	edit	delete
6	Sam	xyz	abc@gmail.com	India	edit	delete

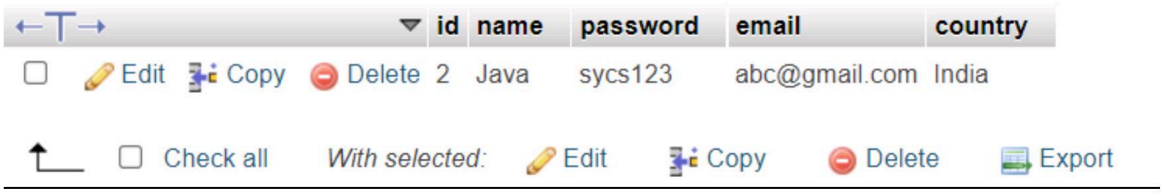
+ Options

	id	name	password	email	country
Edit Copy Delete	2	Java	sycs123	abc@gmail.com	India
Edit Copy Delete	6	Sam	xyz	abc@gmail.com	India

As you can see in above snapshot record is updated in xampp as well as in the web browser

After clicking on delete





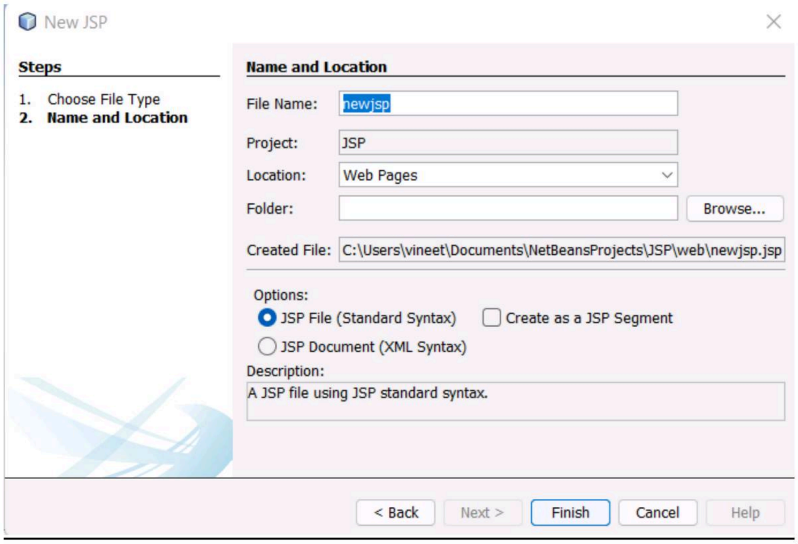
As you can see in above snapshot record is deleted in xampp as well as in the web browser

Conclusion: Successfully implemented CRUD operation using servlet in java.

PRACTICAL NO 7:

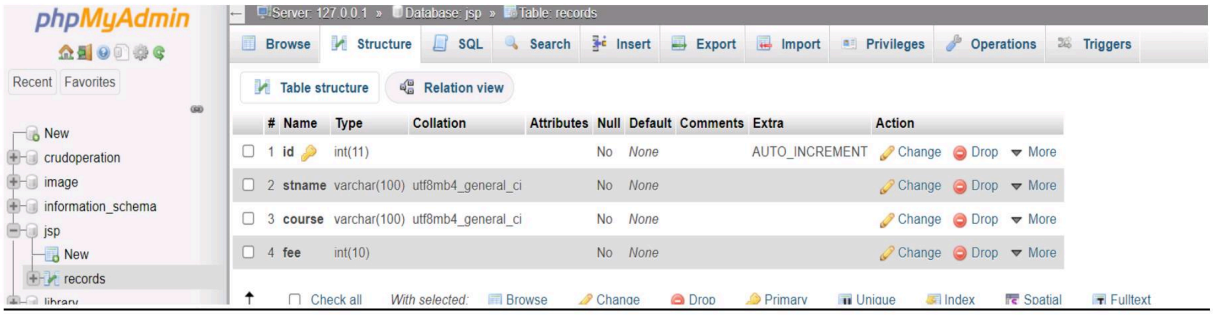
Aim: Create Employees table in EMP database. Perform select, insert, update, delete operations on Employee table using JSP.

Step 1: Create a java web application --> Create JSP files --> Expand your project -> Right click on Web Pages folder --> Select JSP file --> Give name of the file and click on finish.



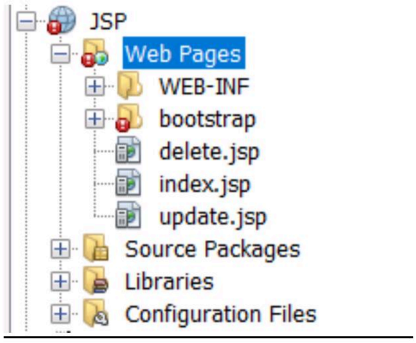
Step 2: Expand your project and add Mysql JDBC library in libraries folder.

Step 3: Create database on xampp.



Step 4: Click on services in IDE --> Right click on database --> Click on New connection --> Select Mysql(connector/Jdriver) --> Then one window will appear in which you have to enter your database name and test the connection, If connection is established click on finish.

Step 5: Download bootstarp 5..1.3 zip file for designing faster --> Extract to It in folder -> Then import that folder inside your project in Web pages folder. (Note: you can ignore the error in bootstrap file)



Step 6: Create JSP files and write code inside it.

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%@page import="java.sql.*" %>
<%

   if(request.getParameter("submit")!=null)
   {
      String name = request.getParameter("sname");
      String course = request.getParameter("course");
      String fee = request.getParameter("fee");

      Connection con;
      PreparedStatement pst;
      ResultSet rs;

      Class.forName("com.mysql.jdbc.Driver");
      con = DriverManager.getConnection("jdbc:mysql://localhost/jsp","root","");
      pst = con.prepareStatement("insert into records(stname,course,fee)values(?,?,?)");
      pst.setString(1, name);
      pst.setString(2, course);
      pst.setString(3, fee);
      pst.executeUpdate();

      %>
   <script>
      alert("Record Added");
   </script>
   <%
   }
  %>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet" type="text/css"/>
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <h1>Student Registation System Crud using-JSP</h1>
    </br>
    <div class="row">
      <div class="col-sm-4">
        <form  method="POST" action="#" >

          <div alight="left">
            <label class="form-label">Student Name</label>
            <input type="text" class="form-control" placeholder="Student Name" name="sname"
id="sname" required >
          </div>

          <div alight="left">
            <label class="form-label">Course</label>
            <input type="text" class="form-control" placeholder="Course" name="course" id="course"
required >
          </div>

          <div alight="left">
            <label class="form-label">Fee</label>
            <input type="text" class="form-control" placeholder="Fee" name="fee" id="fee" required
>
          </div>
            </br>

          <div alight="right">
            <input type="submit" id="submit" value="submit" name="submit" class="btn btn-info">
            <input type="reset" id="reset" value="reset" name="reset" class="btn btn-warning">
          </div>

        </form>
      </div>

      <div class="col-sm-8">
        <div class="panel-body">
          <table id="tbl-student" class="table table-responsive table-bordered" cellpadding ="0"
width="100%">
```

```
            <thead>
              <tr>
                  <th>Student Name</th>
                  <th>Course</th>
                  <th>Fee</th>
                  <th>Edit</th>
                  <th>Delete</th>
              </tr>

              <%

                Connection con;
                PreparedStatement pst;
                ResultSet rs;

                Class.forName("com.mysql.jdbc.Driver");
                con = DriverManager.getConnection("jdbc:mysql://localhost/jsp","root","");

                 String query = "select * from records";
                 Statement st = con.createStatement();

                  rs =  st.executeQuery(query);

                    while(rs.next())
                    {
                       String id = rs.getString("id");
                 %>

              <tr>
                 <td><%=rs.getString("stname") %></td>
                 <td><%=rs.getString("course") %></td>
                 <td><%=rs.getString("fee") %></td>
                 <td><a href="update.jsp?id=<%=id%>">Edit</a></td>
                 <td><a href="delete.jsp?id=<%=id%>">Delete</a></td>
              </tr>

                <%

                 }
                %>
            </table>
          </div>

        </div>
      </div>

    </body>
</html>
```

Update.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*" %>

<%
   if(request.getParameter("submit")!=null)
   {
      String id = request.getParameter("id");
      String name = request.getParameter("sname");
      String course = request.getParameter("course");
      String fee = request.getParameter("fee");

      Connection con;
      PreparedStatement pst;
      ResultSet rs;

      Class.forName("com.mysql.jdbc.Driver");
      con = DriverManager.getConnection("jdbc:mysql://localhost/jsp","root","");
      pst = con.prepareStatement("update records set stname = ?,course =?,fee= ? where id = ?");
      pst.setString(1, name);
      pst.setString(2, course);
      pst.setString(3, fee);
       pst.setString(4, id);
      pst.executeUpdate();

      %>

      <script>
         alert("Record Updated");
     </script>
   <%
   }

%>

<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>

       <link href="bootstrap/css/bootstrap.css" rel="stylesheet" type="text/css"/>
      <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>
```

```
</head>
<body>
   <h1>Student Update</h1>


   <div class="row">
      <div class="col-sm-4">
         <form  method="POST" action="#" >

            <%
                Connection con;
               PreparedStatement pst;
               ResultSet rs;

                Class.forName("com.mysql.jdbc.Driver");
                 con = DriverManager.getConnection("jdbc:mysql://localhost/jsp","root","");

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

               pst = con.prepareStatement("select * from records where id = ?");
               pst.setString(1, id);
               rs = pst.executeQuery();

                while(rs.next())
                {

            %>
            <div alight="left">
               <label class="form-label">Student Name</label>
               <input type="text" class="form-control" placeholder="Student Name" value="<%=
rs.getString("stname")%>" name="sname" id="sname" required >
             </div>

            <div alight="left">
               <label class="form-label">Course</label>
               <input type="text" class="form-control" placeholder="Course" name="course" value="<%=
rs.getString("course")%>" id="course" required >
             </div>

             <div alight="left">
               <label class="form-label">Fee</label>
               <input type="text" class="form-control" placeholder="Fee" name="fee" id="fee"
value="<%= rs.getString("fee")%>" required >
             </div>

            <% }  %>
```

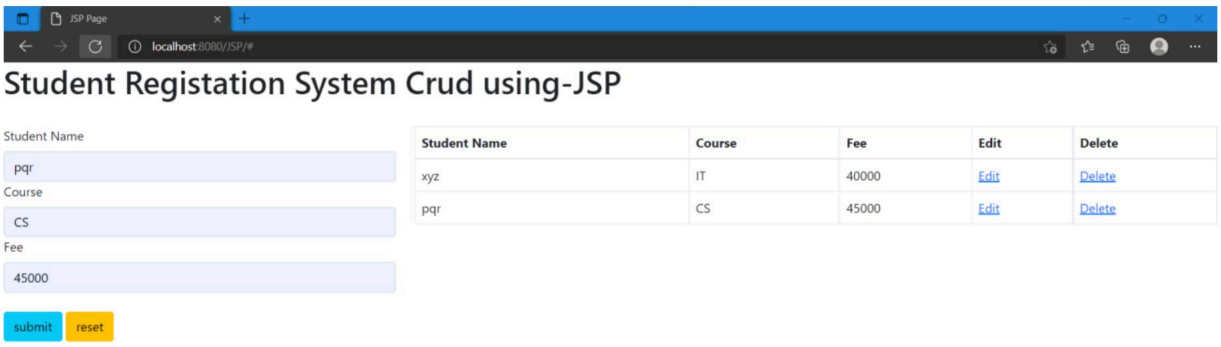
```
                </br>

              <div alight="right">
                <input type="submit" id="submit" value="submit" name="submit" class="btn btn-info">
                <input type="reset" id="reset" value="reset" name="reset" class="btn btn-warning">
              </div>

                <div align="right">

                  <p><a href="index.jsp">Click Back</a></p>


                </div>

            </form>
          </div>
        </div>

    </body>
</html>
```

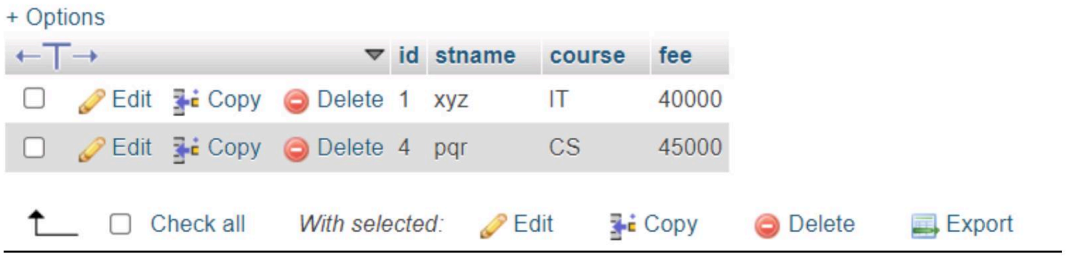
Delete.jsp

```
<%@page import="java.sql.*" %>


<%
      String id = request.getParameter("id");
      Connection con;
      PreparedStatement pst;
      ResultSet rs;

      Class.forName("com.mysql.jdbc.Driver");
      con = DriverManager.getConnection("jdbc:mysql://localhost/jsp","root","");
      pst = con.prepareStatement("delete from records where id = ?");
       pst.setString(1, id);
      pst.executeUpdate();

      %>

      <script>

         alert("Record Deleted");

      </script>
```

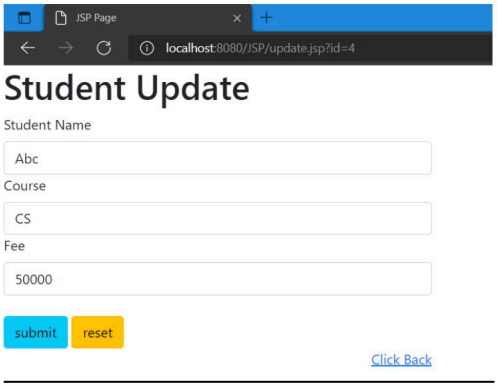
Output:

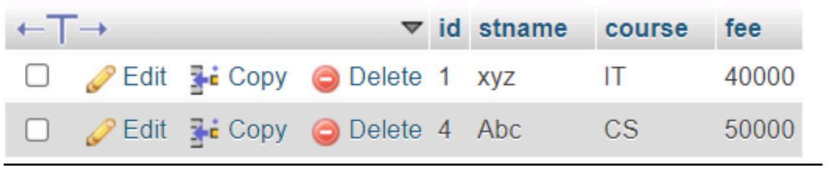
Entering the value and click on submit the value will get stored and it will be displayed in the table



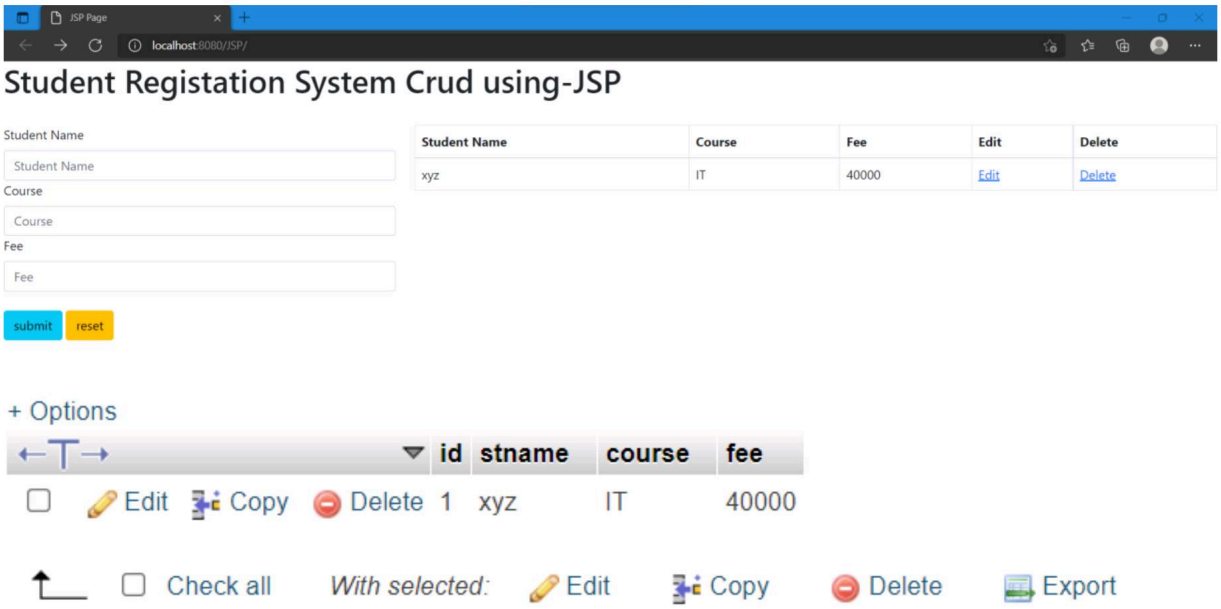


After Clicking on edit





After clicking on delete



Conclusion: Successfully implemented servlet select, insert, update, delete operation using JSP in java.

PRACTICAL NO 8:

Aim: Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write Java Application to access JavaBeans Properties.

Step 1: Create a java web application --> Create JSP files --> Expand your project -> Right click on Web Pages folder --> Select JSP file --> Give name of the file and click on finish.

Step 2: Create 1 java class file and import getters and setters in it.

StudentBean.java

```
package bean;

public class StudentBean {

   private String course;

   public String getCourse() {
      return course;
   }

   public void setCourse(String course) {
      this.course = course;
   }


   private String lastname;

   public String getLastname() {
      return lastname;
   }


   public void setLastname(String lastname) {
      this.lastname = lastname;
   }


   private String firstname;


   public String getFirstname() {
      return firstname;
```

```
    }


    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }


}
```

Step 3: Create 2 JSP files, 1 html and write code inside it.

Index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="first.jsp" method="get">
         Enter First Name: <input type="text" name="firstname"/><br/><br/>
         Enter Last Name: <input type="text" name="lastname"/><br/><br/>
         Enter Course: <input type="text" name="course"/><br/><br/>
         <input type="submit" value="submit"/>

        </form>
    </body>
</html>
```

First.jsp

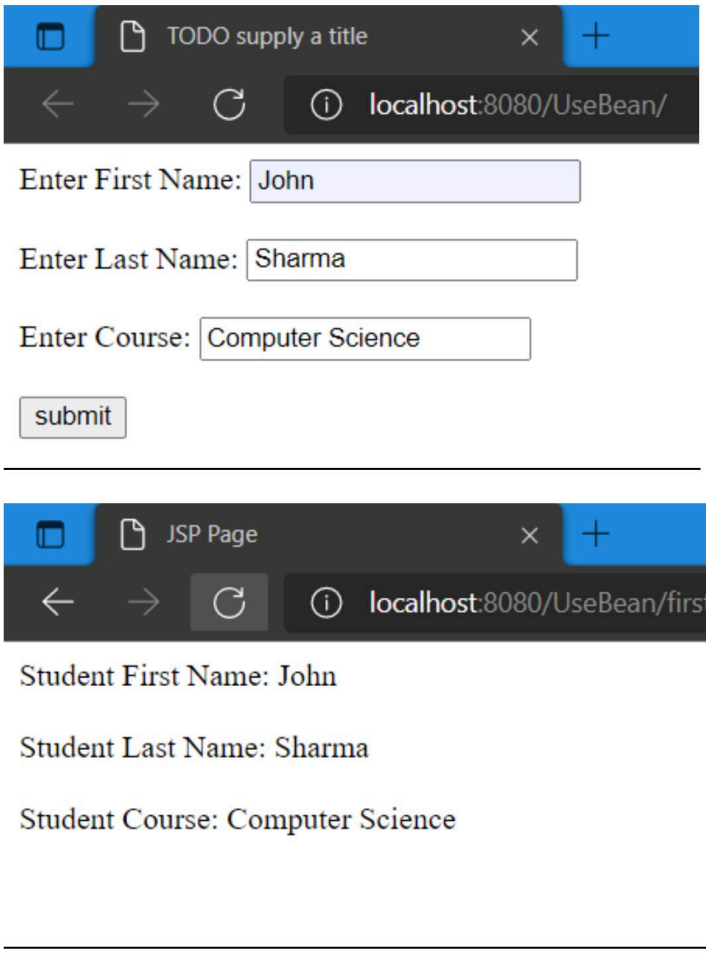
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <% String firstname=request.getParameter("firstname");
        String lastname=request.getParameter("lastname");
        String course=request.getParameter("age");
        %>
        <jsp:useBean id = "b1" class = "bean.StudentBean" scope="request"/>
         <jsp:setProperty name = "b1" property = "firstname"/>
```

```
        <jsp:setProperty name = "b1" property = "lastname"/>
        <jsp:setProperty name = "b1" property = "course"/>

        <jsp:forward page="second.jsp"/>
    </body>
</html>
```

Second.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <% String firstname=request.getParameter("firstname");
        String lastname=request.getParameter("lastname");
        String course=request.getParameter("age");
        %>
        <jsp:useBean id = "b1" class = "bean.StudentBean" scope="request"/>
         Student First Name: <jsp:getProperty name = "b1" property = "firstname"/><br/><br/>
         Student Last Name: <jsp:getProperty name = "b1" property = "lastname"/><br/><br/>
         Student Course: <jsp:getProperty name = "b1" property = "course"/><br/><br/>
    </body>
</html>
```

Output:



Conclusion: Successfully created Java web application to access JavaBeans properties and also created useBean action that declares a JavaBean for use in a JSP.

PRACTICAL NO 9:

Aim: Design application using Struts2. Application must accept user name and greet user when command button is pressed.

Step 1: Download struts2-suite-1.3.5 zip file --> Extraxt it in a folder --> Open netbeans 8.2 --> Click on tools --> Click on pulgins --> Go to downloaded --> Click on add plugins --> Access that folder where you have extracted the file --> You will get 3 files with extestion .nbm in that folder import them and click on install.

Step 2: Create java a web application --> Click on file --> New Project --> Java Web --> Web Application -->Give name of the project --> Select Server -->Select Struts2 --> Click on finish.

Step 3: Create 1 package and inside that add 1 java class file --> create getters and setter in it.

Hello_action.java

```
package action_jsp;

import com.opensymphony.xwork2.ActionSupport;

// everything in here must be kept public to have package level access.

public class hello_action extends ActionSupport
{
  private String name;
  // getter and setter methods for the form element value

  public String getName()
   {
      return name;
   }
   public void setName(String name)
   {
      this.name = name;
   }
//method that returns success or failure on action
   public String execute()
   {
      return "success";
   }
}
```

Step 4: Create 1 folder in Web Pages folder and inside that and add 2 jsp file.

index.jsp

```
<%@page contentType = "text/html" pageEncoding = "UTF-8"%>
<%@taglib  prefix = "s" uri = "/struts-tags" %>

<html>
  <body>
    <s:form method = "post" action = "hello_action">

       <s:label name = "label1"/>
       <s:textfield value="Enter your name" name = "name"/>
       <s:submit value = "CLICK" name = "submit" align="left"/>

    </s:form>
  </body>
</html>
```

result.jsp

```
<%@page contentType = "text/html" pageEncoding = "UTF-8"%>
<%@taglib prefix = "s" uri = "/struts-tags" %>
<html>
<body>
      <h1>Hello <s:property value = "name"/></h1>
</body>
</html>
```

Step 5: Check you web.xml and struts.xml both files will be in configuration file.

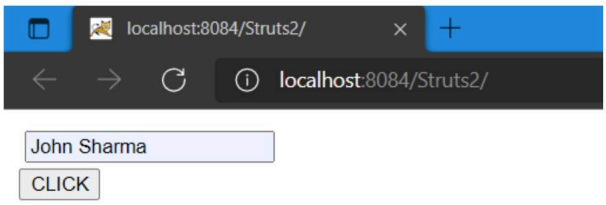
Struts.xml

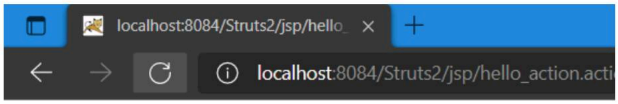
```
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
   <include file="example.xml"/>
   <!-- Configuration for the default package. -->
   <package name="default" extends="struts-default">
      <action name = "hello_action" class = "action_jsp.hello_action" method = "execute">
      <result name = "success">result.jsp</result>
      <result name = "failure">index.jsp</result>
      </action>
   </package>
</struts>
```

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>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>jsp/index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Output:





Hello John Sharma

Conclusion: Successfully designed application using Struts2.