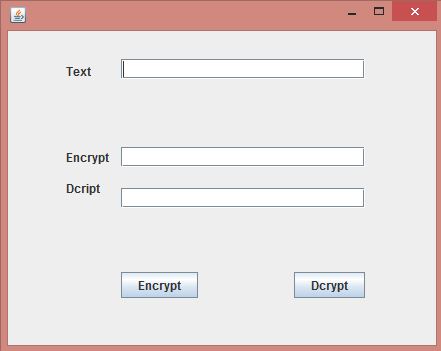
**01.Java Encrypt Decrypt**



private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// Button encrypt

try {

if(!txt.getText().isEmpty()){

Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider()); //type of security

input = txt.getText().getBytes();

SecretKeySpec key = new SecretKeySpec(keyBytes, "DES");

IvParameterSpec ivSpec = new IvParameterSpec(ivBytes);

cipher = Cipher.getInstance("DES/CTR/NoPadding", "BC"); //CTR of encryption

cipher.init(Cipher.ENCRYPT\_MODE, key, ivSpec);

chiperText = new byte[cipher.getOutputSize(input.length)];

ctLenght = cipher.update(input, 0, input.length, chiperText, 0);

ctLenght += cipher.doFinal(chiperText, ctLenght);

Etxt.setText(new String(chiperText));}

else{

JOptionPane.showMessageDialog(null, "Please enter a text");}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

**//dycrypt**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try {

if(!Etxt.getText().isEmpty()){

cipher.init(Cipher.DECRYPT\_MODE, key,ivSpec);

byte[] plainText = new byte[cipher.getOutputSize(ctLenght)];

int ptLength = cipher.update(chiperText, 0,ctLenght,plainText,0);

ptLength += cipher.doFinal(plainText, ptLength);

Dtxt.setText(new String(plainText));

}else{JOptionPane.showMessageDialog(null, "Please enter a text and encrypt it ");}

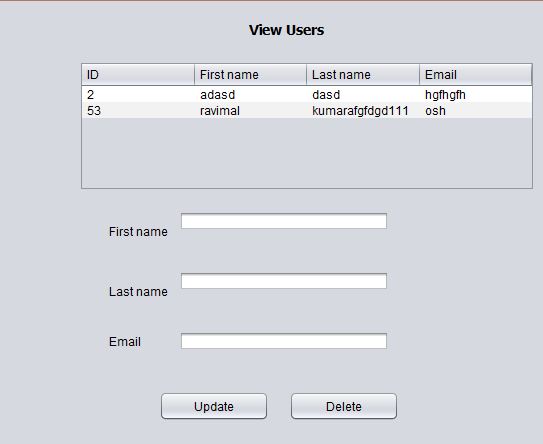
} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

**2.mongoDB**



**//DBManager class**

package mongodb;

import com.mongodb.DB;

import com.mongodb.MongoClient;

import java.net.UnknownHostException;

public class DBManager {

private static DB database;

public static DB getDatabase() throws java.net.UnknownHostException {

if(database == null) {

MongoClient mongo;

try {

mongo = new MongoClient("localhost", 27017);

database = mongo.getDB("usermanager");

} catch (UnknownHostException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return database;

}

}

**// Userr class**

package mongodb;

public class Userr {

private int id;

private String firstName;

private String lastName;

private String email;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

**//Add button // setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);//use in constructor**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

if(txtE.getText().equals("")||txtID.getText().equals("")||txtF.getText().equals("")||txtF.getText().equals(""))

{

JOptionPane.showMessageDialog(null, "Please fill all text fields");

}

else{

Userr us= new Userr();

us.setId(Integer.parseInt(txtID.getText()));

us.setFirstName(txtF.getText());

us.setLastName(txtL.getText());

us.setEmail(txtE.getText());

DBObject doc = createDBObject(us);

DB userDB = DBManager.getDatabase();

DBCollection col = userDB.getCollection("user");

WriteResult result = col.insert(doc);

JOptionPane.showMessageDialog(null, "Date inserted");

txtE.setText(""); txtID.setText(""); txtF.setText(""); txtL.setText("");

}

} catch (UnknownHostException ex) {

Logger.getLogger(Insert.class.getName()).log(Level.SEVERE, null, ex);

}

catch(NumberFormatException e){

JOptionPane.showMessageDialog(null, "ID must be a number");

}

private static DBObject createDBObject(Userr user) {

BasicDBObjectBuilder docBuilder = BasicDBObjectBuilder.start();

docBuilder.append("\_id", user.getId());

docBuilder.append("firstName", user.getFirstName());

docBuilder.append("lastName", user.getLastName());

docBuilder.append("email", user.getEmail());

return docBuilder.get();

}

**//Load values in to a table ..use this method in constructor** ..

public void load(){

try {

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

DB userDB = DBManager.getDatabase();

DBCollection col = userDB.getCollection("user");

DBCursor cursor = col.find();

while (cursor.hasNext()) {

DBObject user=(DBObject) cursor.next();

model.addRow(new Object[] {user.get("\_id"),user.get("firstName"),user.get("lastName"),user.get("email")});

System.out.println(user);

System.out.println(user.get("\_id"));

}

} catch (UnknownHostException ex) {

Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);

JOptionPane.showMessageDialog(null, ex);

}

}

**//Update buttonclick button**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String val= jTable1.getModel().getValueAt(jTable1.getSelectedRow(), 0).toString();

try {

DB userDB = DBManager.getDatabase();

DBCollection col = userDB.getCollection("user");

BasicDBObject query = new BasicDBObject("\_id", new BasicDBObject("$eq", Integer.parseInt(val)));

BasicDBObject newDocument = new BasicDBObject();

newDocument.put("firstName", jTextField1.getText());

newDocument.put("lastName", jTextField2.getText());

newDocument.put("email", jTextField3.getText());

BasicDBObject updateObj = new BasicDBObject();

updateObj.put("$set", newDocument);

col.update(query, updateObj, false, true);

JOptionPane.showMessageDialog(rootPane, "User Update SucessFully...!");

} catch (UnknownHostException ex) {

Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);

}

}

**//Delete button click**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String val= jTable1.getModel().getValueAt(jTable1.getSelectedRow(), 0).toString();

try {

DB userDB = DBManager.getDatabase();

DBCollection col = userDB.getCollection("user");

BasicDBObject query = new BasicDBObject("\_id", new BasicDBObject("$eq", Integer.parseInt(val)));

col.remove(query);

// col.D(query, updateObj, false, true);

JOptionPane.showMessageDialog(rootPane, "User deleted SucessFully...!");

this.dispose();

new View().setVisible(true);

} catch (UnknownHostException ex) {

Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);

}

}

**//table click event**

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

try {

String val= jTable1.getModel().getValueAt(jTable1.getSelectedRow(), 0).toString();

DB userDB = DBManager.getDatabase();

DBCollection col = userDB.getCollection("user");

BasicDBObject query = new BasicDBObject("\_id", new BasicDBObject("$eq", Integer.parseInt(val)));

DBCursor cursor = col.find(query);

DBObject user = (DBObject) cursor.next();

jTextField1.setText(user.get("firstName").toString());

jTextField2.setText(user.get("lastName").toString());

jTextField3.setText(user.get("email").toString());

} catch (UnknownHostException ex) {

Logger.getLogger(View.class.getName()).log(Level.SEVERE, null, ex);

}

}

**WebService //db class**

package db;

import java.sql.Connection;

import java.sql.DriverManager;

public class Dbconection {

public static Connection getConnection(){

Connection connection = null;

try {

connection = DriverManager.getConnection("jdbc:derby://localhost:1527/Test", "ead", "ead");//.getConnection("jdbc:derby://localhost:1527/sample");

} catch (Exception e) {

e.printStackTrace();

}

return connection;

}

}

**//person class**

package bs;

public class person {

int id;

String name;

String address;

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 getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

package bs;

import db.Dbconection;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

public class Oparations {

Connection con = Dbconection.getConnection();

Statement statement = null;

ResultSet resultSet = null;

public List<person> getdata()

{

List<person> personlist = new ArrayList<person>();

try {

statement = con.createStatement();

resultSet = statement.executeQuery("select \* from EAD.PERSON");

while (resultSet.next()) {

person p = new person();

p.setId(resultSet.getInt("id"));

p.setName(resultSet.getString("name"));

p.setAddress(resultSet.getString("address"));

System.out.println("name");

personlist.add(p);

}

} catch (SQLException ex) {

} finally {

}

return personlist;

}

/\*

public void add(Supplier supplier) {

try {

String queryString = "INSERT INTO Supplier(sup\_name,sup\_address,sup\_status) values('"+supplier.getName()+"','"+supplier.getAddress()+"',"+true+")";

connection = getConnection();

ptmt = connection.prepareStatement(queryString);

ptmt.executeUpdate();

queryString="INSERT INTO Supplier\_contact values(?,?,?,?) ";

ptmt = connection.prepareStatement(queryString);

ptmt.setInt(1,supid );

ptmt.setInt(2, supplier.getPhone());

ptmt.setString(3, supplier.getEmail());

ptmt.setInt(4, supplier.getFax());

ptmt.executeUpdate();

JOptionPane.showMessageDialog(null, "Supplier '"+supplier.getName()+"' information added successfully", "Successfull", 3);//successful message

} catch (SQLException e) {

e.printStackTrace();

}

}

}

//use in webservices

@WebMethod(operationName = "searchpersons")

public List<person> searchpersons() {

Oparations obj = new Oparations();

return obj.getdata();

}

**EJB**

**Messagedrivenbean**

**Add following**

**@Resource**

**private MessageDrivenContext mdc;**

**public void onMessage(Message message) {**

**ObjectMessage msg = null;**

**try {**

**if (message instanceof ObjectMessage) {**

**msg = (ObjectMessage) message;**

**NewsEntity e = (NewsEntity) msg.getObject();**

**save(e);**

**}**

**}catch(JMSException e){**

**e.printStackTrace();**

**mdc.setRollbackOnly();**

**}**

**catch(Throwable te)**

**{ te.printStackTrace();**

**}**

**}**

**Servlet post**

**@Resource(mappedName="jms/NewMessageFactory")**

**private ConnectionFactory connectionFactory;**

**@Resource(mappedName="jms/NewMessage")**

**private Queue queue;**

**//////////////**

**String title=request.getParameter("title");**

**String body=request.getParameter("body");**

**if ((title!=null) && (body!=null)) {**

**try {**

**Connection connection = connectionFactory.createConnection();**

**Session session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);**

**MessageProducer messageProducer = session.createProducer(queue);**

**ObjectMessage message = session.createObjectMessage();**

**// here we create NewsEntity, that will be sent in JMS message**

**NewsEntity e = new NewsEntity();**

**e.setTitle(title);**

**e.setBody(body);**

**message.setObject(e);**

**messageProducer.send(message);**

**messageProducer.close();**

**connection.close();**

**response.sendRedirect("ListNews");**

**} catch (JMSException ex) {**

**ex.printStackTrace();**

**}**

**}**

**///////////////////////**

**out.println("<form>");**

**out.println("Title: <input type='text' name='title'><br/>");**

**out.println("Message: <textarea name='body'></textarea><br/>");**

**out.println("<input type='submit'><br/>");**

**out.println("</form>");**