**Practical no. 1**

**Aim**: Design User Login application using swing components

**Source code:**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class UserLoginForm{

// declaration of components

JFrame mainFrame;

JLabel heading;

JLabel usernameLabel;

JLabel passwordLabel;

JTextField usernameField;

JPasswordField passwordField;

JButton loginButton;

JLabel statusBar;

// creating constructor coz it is a good practice

UserLoginForm(){

// initialization

mainFrame = new JFrame("User login form");

heading = new JLabel("USER LOGIN FORM");

usernameLabel = new JLabel("Username: ");

passwordLabel = new JLabel("Password: ");

usernameField = new JTextField("Username");

passwordField = new JPasswordField("");

loginButton = new JButton("Login");

statusBar = new JLabel("Status.. ");

// setting bounds for each coz using layout as "null"

heading.setBounds(80,10,150,30);

usernameLabel.setBounds(50,50,100,20);

passwordLabel.setBounds(50,90,100,20);

usernameField.setBounds(120,50,100,25);

passwordField.setBounds(120,90,100,25);

loginButton.setBounds(50,130,170,30);

statusBar.setBounds(70,170,150,30);

mainFrame.add(heading);

mainFrame.add(usernameLabel);

mainFrame.add(passwordLabel);

mainFrame.add(usernameField);

mainFrame.add(passwordField);

mainFrame.add(loginButton);

mainFrame.add(statusBar);

// imp things to be added before anything

// frame properties

mainFrame.setSize(300,250);

mainFrame.setLayout(null);

mainFrame.setVisible(true);

// if frame closes then terminate the program

mainFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// adding actionlistener for loginButton

loginButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

performLogin();

}

});

}

// method to be perform after login button click

public void performLogin(){

String username = usernameField.getText();

char[] passwordChar = passwordField.getPassword();

String password = new String(passwordChar);

if (username.equals("Jayesh") && password.equals("123")){

statusBar.setText("User login successful :)");

} else {

statusBar.setText("User login failed :(");

}

}

// main method or entry point

public static void main(String[] args){

// instance of a class

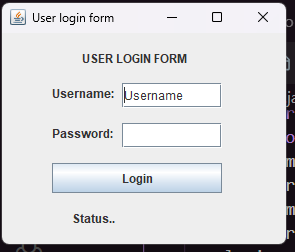
new UserLoginForm();

}

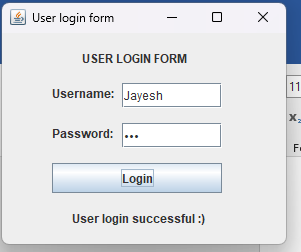
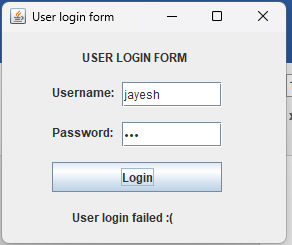
}

**Output:**

How application looks like :



After successful login: After login failed:

**Practical No. 2**

**Aim**: Design Library Management System using swing components

**Source code:**

package pkg1st.prac;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

class prac2 extends JFrame {

JTextField studentNameField, classNameField, rollNumberField, bookNameField, authorField, issueDateField, returnDateField;

JButton addButton;

JTextArea bookListTextArea;

ArrayList<String> bookList;

prac2() {

bookList = new ArrayList<>();

setLayout(new BorderLayout());

bookListTextArea = new JTextArea();

JScrollPane scrollPane = new JScrollPane(bookListTextArea);

add(scrollPane, BorderLayout.CENTER);

JPanel inputPanel = new JPanel(new GridLayout(8, 2, 10, 10));

studentNameField = new JTextField();

classNameField = new JTextField();

rollNumberField = new JTextField();

bookNameField = new JTextField();

authorField = new JTextField();

issueDateField = new JTextField();

returnDateField = new JTextField();

addButton = new JButton("Add Book");

inputPanel.add(new JLabel("Student Name:"));

inputPanel.add(studentNameField);

inputPanel.add(new JLabel("Class Name:"));

inputPanel.add(classNameField);

inputPanel.add(new JLabel("Roll Number:"));

inputPanel.add(rollNumberField);

inputPanel.add(new JLabel("Book Name:"));

inputPanel.add(bookNameField);

inputPanel.add(new JLabel("Author:"));

inputPanel.add(authorField);

inputPanel.add(new JLabel("Issue Date:"));

inputPanel.add(issueDateField);

inputPanel.add(new JLabel("Return Date:"));

inputPanel.add(returnDateField);

inputPanel.add(new JLabel()); // Empty space for layout

inputPanel.add(addButton);

add(inputPanel, BorderLayout.SOUTH);

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

addBook();

}

});

setTitle("Library Management System");

setSize(600, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setVisible(true);

}

void addBook() {

String studentName = studentNameField.getText();

String className = classNameField.getText();

String rollNumber = rollNumberField.getText();

String bookName = bookNameField.getText();

String author = authorField.getText();

String issueDate = issueDateField.getText();

String returnDate = returnDateField.getText();

if (!studentName.isEmpty() && !className.isEmpty() && !rollNumber.isEmpty()&& !bookName.isEmpty() && !author.isEmpty() && !issueDate.isEmpty() && !returnDate.isEmpty()) {

String bookDetails = "Student Name: " + studentName + ", Class Name: " + className+ ", Roll Number: " + rollNumber + "\n"+ "Book Name: " + bookName + ", Author: " + author+ ", Issue Date: " + issueDate + ", Return Date: " + returnDate;

bookList.add(bookDetails);

updateBookList();

clearInputFields();

} else {

JOptionPane.showMessageDialog(this, "Please fill in all fields.");

}

}

void updateBookList() {

StringBuilder stringBuilder = new StringBuilder();

for (String bookDetails : bookList) {

stringBuilder.append(bookDetails).append("\n\n");

}

bookListTextArea.setText(stringBuilder.toString());

}

void clearInputFields() {

studentNameField.setText("");

classNameField.setText("");

rollNumberField.setText("");

bookNameField.setText("");

authorField.setText("");

issueDateField.setText("");

returnDateField.setText("");

}

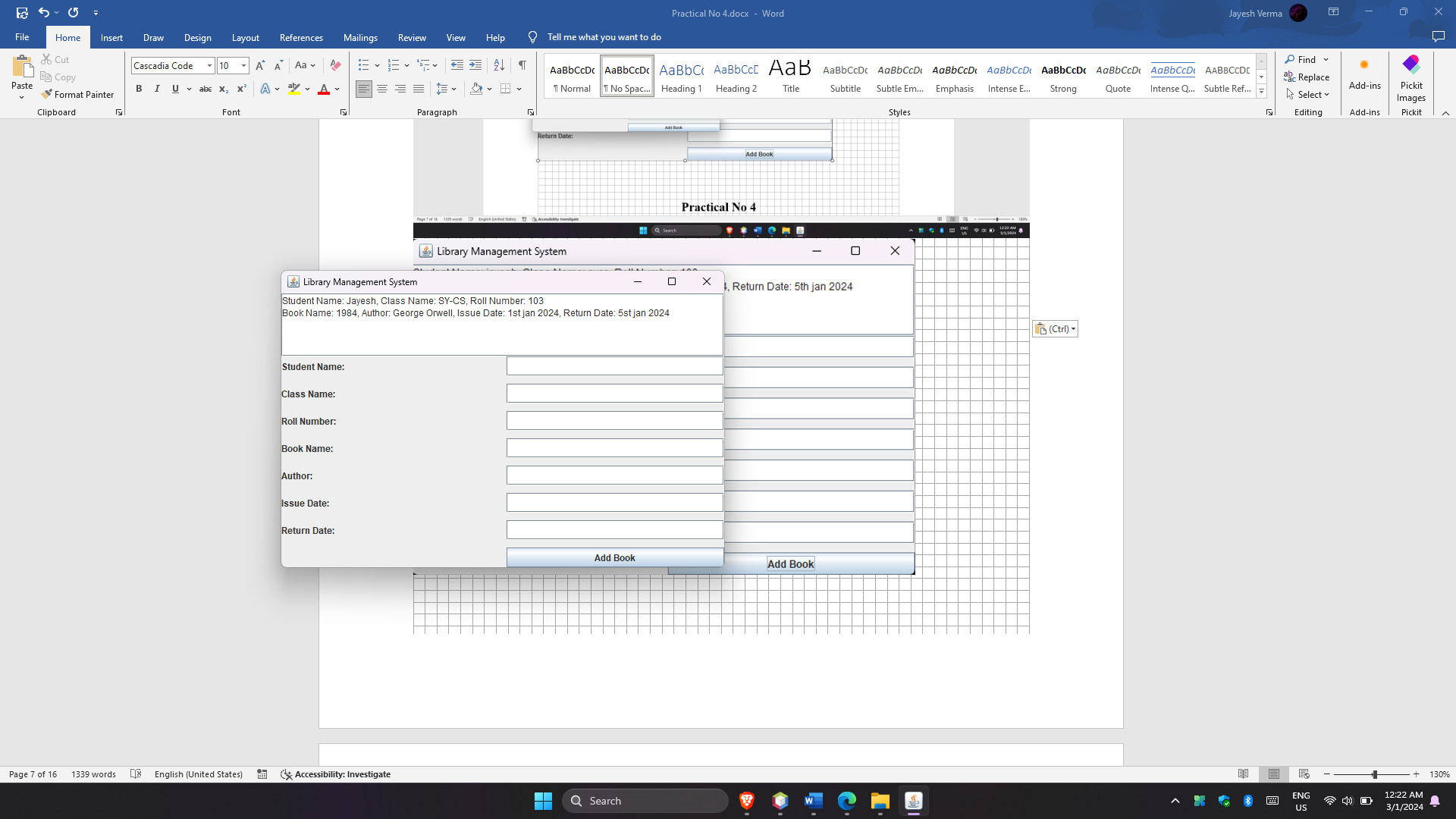
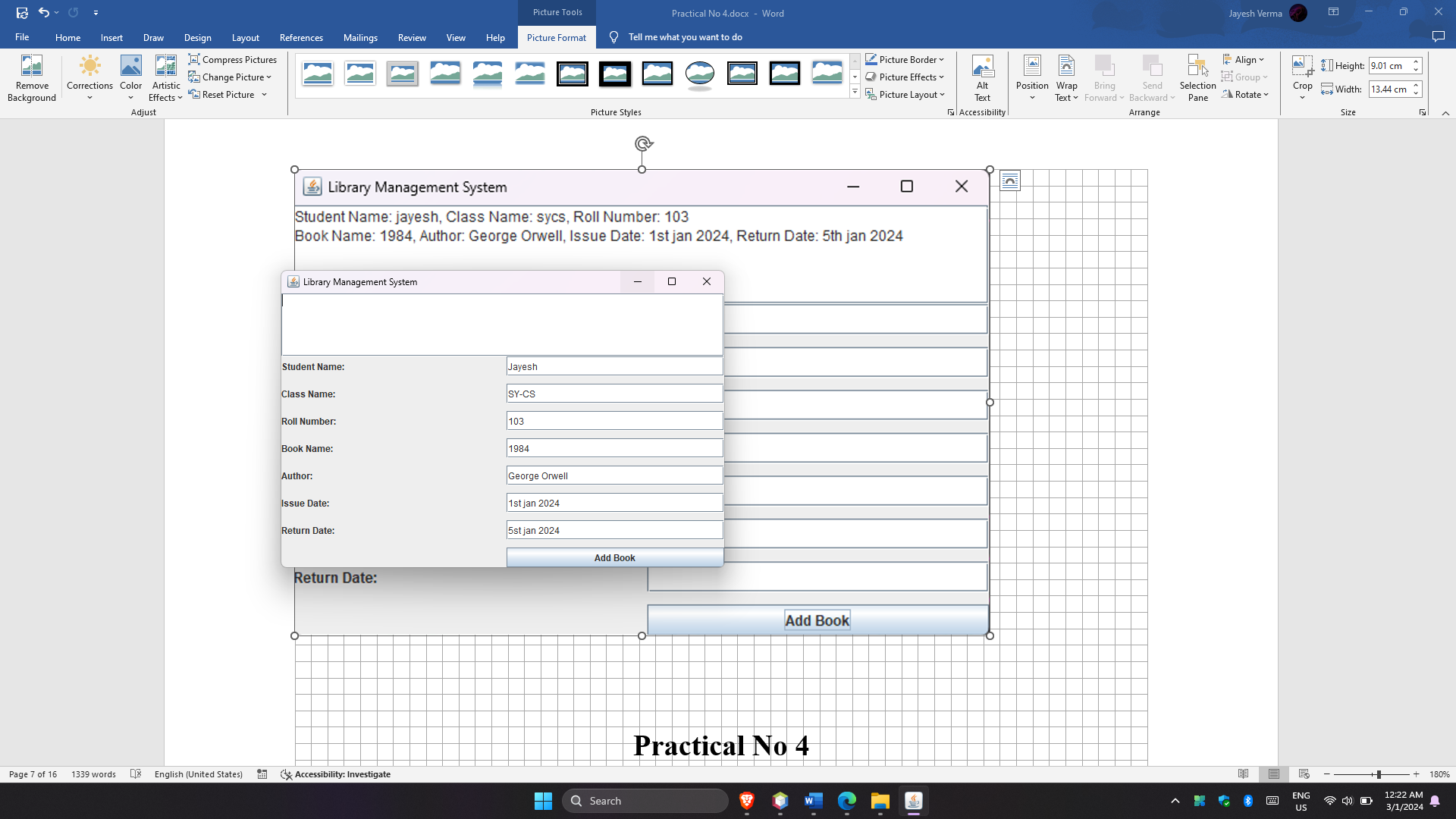
public static void main(String[] args) {

new prac2();

}

}

**Output:**



**Practical No. 3**

**Aim:** Design Pizza Order System using swing components

**Source code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

class PizzaOrderApplication extends JFrame {

JTextField customerNameField, addressField, mobileNumberField, quantityField,

priceField;

JComboBox pizzaTypeComboBox, paymentOptionComboBox;

JCheckBox pepperoniCheckBox, mushroomsCheckBox, extraCheeseCheckBox;

JButton orderButton;

ArrayList<Double> pizzaPrices;

PizzaOrderApplication() {

setTitle("Pizza Order Application");

setSize(400, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

initUI();

setVisible(true);

}

private void initUI() {

setLayout(new BorderLayout());

JPanel formPanel = new JPanel(new GridLayout(10, 2, 10, 10));

formPanel.add(new JLabel("Customer Name:"));

customerNameField = new JTextField();

formPanel.add(customerNameField);

formPanel.add(new JLabel("Address:"));

addressField = new JTextField();

formPanel.add(addressField);

formPanel.add(new JLabel("Mobile Number:"));

mobileNumberField = new JTextField();

formPanel.add(mobileNumberField);

formPanel.add(new JLabel("Pizza Type:"));

pizzaTypeComboBox = new JComboBox(new String[]{"Margherita", "Pepperoni",

"Vegetarian"});

formPanel.add(pizzaTypeComboBox);

formPanel.add(new JLabel("Toppings:"));

pepperoniCheckBox = new JCheckBox("Pepperoni");

mushroomsCheckBox = new JCheckBox("Mushrooms");

extraCheeseCheckBox = new JCheckBox("Extra Cheese");

JPanel toppingsPanel = new JPanel(new FlowLayout());

toppingsPanel.add(pepperoniCheckBox);

toppingsPanel.add(mushroomsCheckBox);

toppingsPanel.add(extraCheeseCheckBox);

formPanel.add(toppingsPanel);

formPanel.add(new JLabel("Quantity:"));

quantityField = new JTextField();

formPanel.add(quantityField);

formPanel.add(new JLabel("Price:"));

priceField = new JTextField();

priceField.setEditable(false);

formPanel.add(priceField);

formPanel.add(new JLabel("Payment Option:"));

paymentOptionComboBox = new JComboBox(new String[]{"Credit Card", "Cash",

"Online Payment"});

formPanel.add(paymentOptionComboBox);

formPanel.add(new JLabel()); // Empty label for layout

orderButton = new JButton("Place Order");

formPanel.add(orderButton);

add(formPanel, BorderLayout.CENTER);

orderButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

placeOrder();

}

});

pizzaPrices = new ArrayList<>();

pizzaPrices.add(8.99); // Margherita

pizzaPrices.add(10.99); // Pepperoni

pizzaPrices.add(9.99); // Vegetarian

pizzaTypeComboBox.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

updatePrice();

}

});

pepperoniCheckBox.addActionListener(e -> updatePrice());

mushroomsCheckBox.addActionListener(e -> updatePrice());

extraCheeseCheckBox.addActionListener(e -> updatePrice());

}

void placeOrder() {

String customerName = customerNameField.getText();

String address = addressField.getText();

String mobileNumber = mobileNumberField.getText();

String pizzaType = (String) pizzaTypeComboBox.getSelectedItem();

String toppings = getSelectedToppings();

String quantity = quantityField.getText();

String price = priceField.getText();

String paymentOption = (String) paymentOptionComboBox.getSelectedItem();

String orderDetails = "Order Details:\n"

+ "Customer Name: " + customerName + "\n"

+ "Address: " + address + "\n"

+ "Mobile Number: " + mobileNumber + "\n"

+ "Pizza Type: " + pizzaType + "\n"

+ "Toppings: " + toppings + "\n"

+ "Quantity: " + quantity + "\n"

+ "Price: $" + price + "\n"

+ "Payment Option: " + paymentOption;

JOptionPane.showMessageDialog(this, orderDetails, "Order Confirmation",

JOptionPane.INFORMATION\_MESSAGE);

clearInputFields();

}

void updatePrice() {

int selectedIndex = pizzaTypeComboBox.getSelectedIndex();

double basePrice = pizzaPrices.get(selectedIndex);

int selectedToppings = getSelectedToppingsCount();

double totalPrice = basePrice + (selectedToppings \* 1.50);

try {

int quantity = Integer.parseInt(quantityField.getText());

totalPrice \*= quantity;

} catch (NumberFormatException e) {

// Ignore invalid quantity input

}

priceField.setText(String.format("%.2f", totalPrice));

}

String getSelectedToppings() {

StringBuilder toppings = new StringBuilder();

if (pepperoniCheckBox.isSelected()) {

toppings.append("Pepperoni ");

}

if (mushroomsCheckBox.isSelected()) {

toppings.append("Mushrooms ");

}

if (extraCheeseCheckBox.isSelected()) {

toppings.append("Extra Cheese ");

}

return toppings.toString().trim();

}

int getSelectedToppingsCount() {

int count = 0;

if (pepperoniCheckBox.isSelected()) {

count++;

}

if (mushroomsCheckBox.isSelected()) {

count++;

}

if (extraCheeseCheckBox.isSelected()) {

count++;

}

return count;

}

void clearInputFields() {

customerNameField.setText("");

addressField.setText("");

mobileNumberField.setText("");

pizzaTypeComboBox.setSelectedIndex(0);

pepperoniCheckBox.setSelected(false);

mushroomsCheckBox.setSelected(false);

extraCheeseCheckBox.setSelected(false);

quantityField.setText("");

priceField.setText("");

paymentOptionComboBox.setSelectedIndex(0);

}

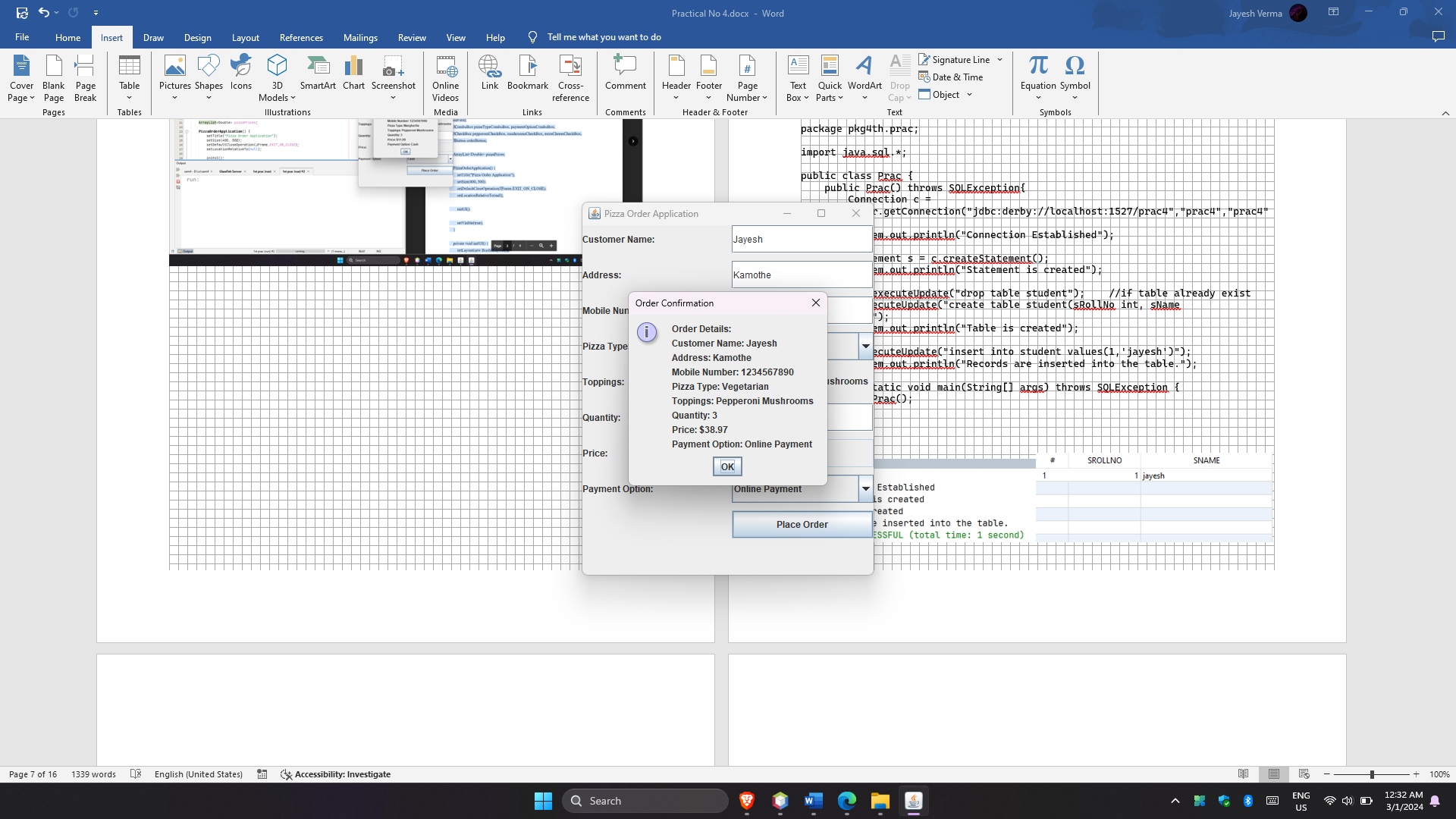
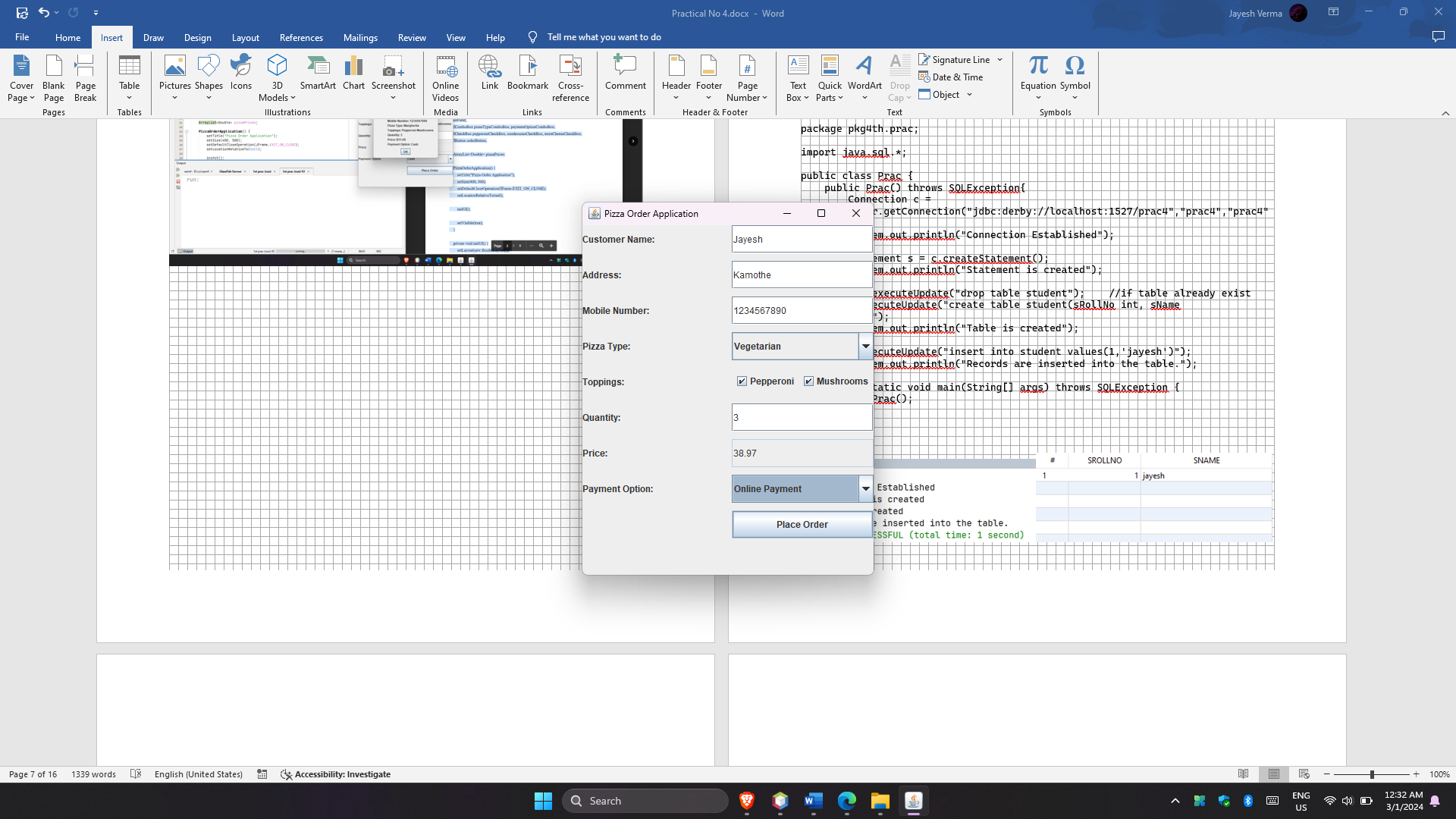
public static void main(String[] args) {

new PizzaOrderApplication();

}

}

**Output:**



**Practical No 4**

**Aim:** Write a JDBC program to create table and insert values in it.

**Source Code:**

package pkg4th.prac;

import java.sql.\*;

public class Prac {

public Prac() throws SQLException{

Connection c = DriverManager.getConnection("jdbc:derby://localhost:1527/prac4","prac4","prac4");

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

Statement s = c.createStatement();

System.out.println("Statement is created");

// s.executeUpdate("drop table student"); //if table already exist

s.executeUpdate("create table student(sRollNo int, sName varchar(20))");

System.out.println("Table is created");

s.executeUpdate("insert into student values(1,'jayesh')");

System.out.println("Records are inserted into the table.");

}

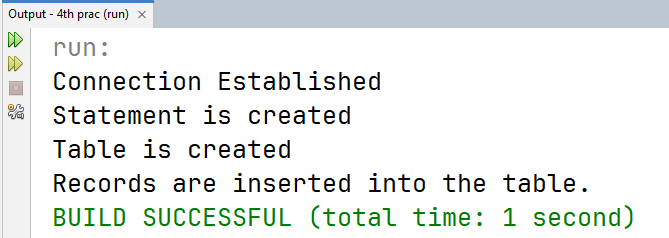
public static void main(String[] args) throws SQLException {

new Prac();

}

}

Output:



**Practical No. 5**

**Aim:** Develop a Java application to perform CRUD operation using JDBC.

**Source Code:**

package pkg5th.prac;

import java.sql.\*;

public class Prac {

public Prac() throws SQLException {

Connection c = DriverManager.getConnection("jdbc:derby://localhost:1527/prac5", "prac5", "prac5");

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

Statement s = c.createStatement();

System.out.println("Statment is created");

// create

s.executeUpdate("create table student(sRollNo int, sName varchar(20))");

System.out.println("Table is created");

// read

s.executeUpdate("insert into student values(1,'jayesh'),(2,'ramesh')");

System.out.println("Values are inserted");

// update

s.executeUpdate("update student set sname='Jayesh' where sRollNo=1");

System.out.println("Name\t\tRoll No.\n");

// delete

// s.executeUpdate("delete from student where sname='Jayesh'");

ResultSet r = s.executeQuery("select \* from student");

while (r.next()) {

int RollNo = r.getInt("sRollNo");

String Name = r.getString("sName");

System.out.println(Name + "\t\t " + RollNo);

}

}

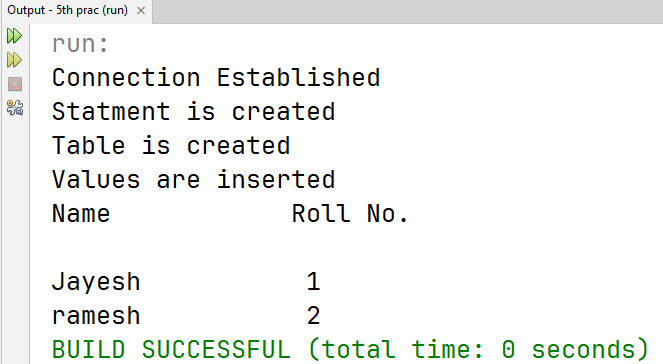
public static void main(String[] args) throws SQLException {

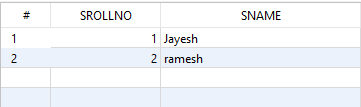
new Prac();

}

}

**Output:**





**Practical No. 6.1**

**Aim:** Develop a Java application to store image in a database and display all values on the output window in the same application.

**Source code:**

package pkg6th.prac;

import java.beans.Statement;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.OutputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class Prac {

public static void main(String[] args) throws SQLException, FileNotFoundException {

try {

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/prac6", "prac6", "prac6");

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

java.sql.Statement stmt = con.createStatement();

String createTable = "CREATE TABLE Tutorial(Name VARCHAR(255), Type VARCHAR(50), Logo BLOB)";

stmt.execute(createTable);

System.out.println("table created");

//Inserting values

String query = "INSERT INTO Tutorial(Name, Type, Logo) VALUES (?, ?, ?)";

PreparedStatement pstmt = con.prepareStatement(query);

pstmt.setString(1, "Advance Java");

pstmt.setString(2, "A Java Practical");

FileInputStream fin = new FileInputStream("D:\\img.png");

pstmt.setBinaryStream(3, fin);

pstmt.execute();

System.out.println("Data inserted");

ResultSet rs = stmt.executeQuery("Select \* from Tutorial");

while (rs.next()) {

System.out.print("Name: " + rs.getString("Name") + ", ");

System.out.print("Tutorial Type: " + rs.getString("Type") + ", ");

System.out.print("Logo: " + rs.getBlob("Logo"));

System.out.println();

}

} catch (Exception e) {

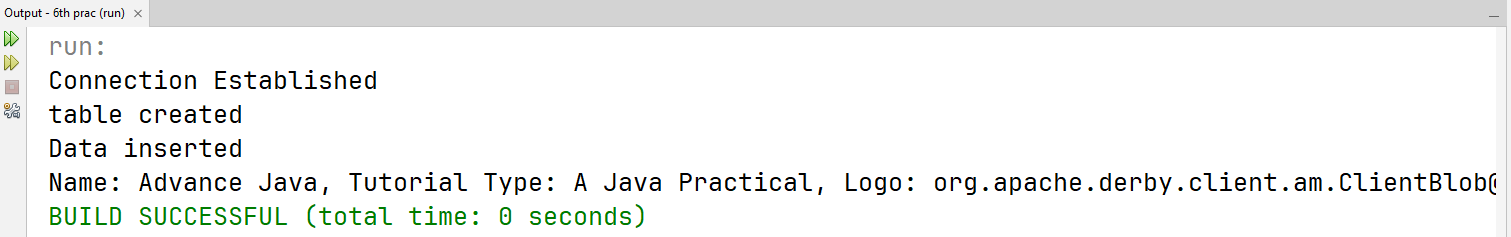
System.out.println("Exception generated: " + e);

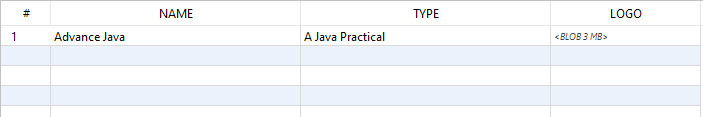
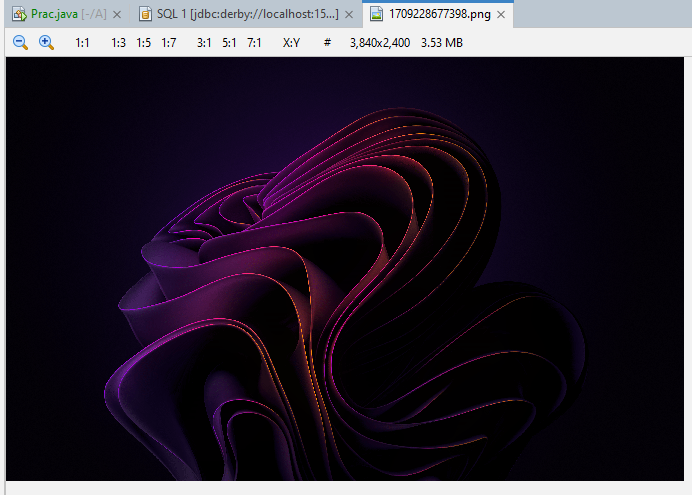
}

}

}

**Output:**



**Practical No. 6.2**

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

**Source Code:**

package pkg6th.prac;

import java.beans.Statement;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.OutputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class prac\_6\_b {

public static void main(String[] args) throws SQLException, FileNotFoundException {

try {

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/prac6", "prac6", "prac6");

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

java.sql.Statement stmt = con.createStatement();

//If table is already created, then comment below statements

/\* String createTable = "CREATE TABLE Tutorial( "

+ "Name VARCHAR(255), "

+ "Type VARCHAR(50), "

+ "Logo BLOB)";

stmt.execute(createTable);

System.out.println("table created");\*/

//Inserting values

String query = "INSERT INTO Tutorial(Name, Type, Logo) VALUES (?, ?, ?)";

PreparedStatement pstmt = con.prepareStatement(query);

pstmt.setString(1, "Advance Java");

pstmt.setString(2, "A Java Practical");

FileInputStream fin = new FileInputStream("D:\\img.png");

pstmt.setBinaryStream(3, fin);

pstmt.execute();

System.out.println("Data inserted");

// ResultSet rs = stmt.executeQuery("Select \*from Tutorial");

// selecting only logo field from the table

ResultSet rs = stmt.executeQuery("Select logo from Tutorial");

int i = 0;

while (rs.next()) {

// below code is used to display output on the output window in the application itself

/\* System.out.print("Name: "+rs.getString("Name")+", ");

System.out.print("Tutorial Type: "+rs.getString("Type")+", ");

System.out.print("Logo: "+rs.getBlob("Logo"));

System.out.println();\*/

InputStream in = rs.getBinaryStream(1);

OutputStream f = new FileOutputStream(new File("test" + i + ".jpg"));

i++;

int c = 0;

while ((c = in.read()) > -1) {

f.write(c);

}

f.close();

in.close();

}

System.out.println("File is created.");

//To check the image file, Go in Files tab, select your project & expand build.xml, all images with test name will be displayed.");

}catch (Exception e) {

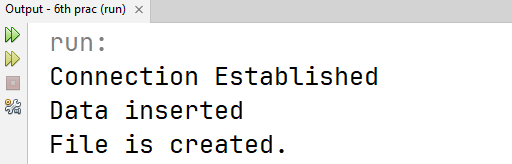
System.out.println("Exception generated : " + e);

}

}

}

**Output:**



**Practical No. 7**

**Aim:** Develop a Java application to display current date and time using servlet

**Source code:**

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

@WebServlet(urlPatterns = {"/prac7"})

public class prac7 extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet displaydate</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet displaydate at " + request.getContextPath() + "</h1>");

java.util.Date date = new java.util.Date();

out.println("<h2>" + "Current Date &amp; Time: " + date.toString() + "</h2>");

out.println("</body>");

out.println("</html>");

}

}

}

**Output:**

init:

deps-module-jar:

deps-ear-jar:

deps-jar:

library-inclusion-in-archive:

library-inclusion-in-manifest:

compile:

compile-jsps:

Incrementally deploying 7th\_prac

Completed incremental distribution of 7th\_prac

run-deploy:

Browsing: http://localhost:8080/7th\_prac/prac7

run-display-browser:

run:

BUILD SUCCESSFUL (total time: 0 seconds)

**On browser:**

