1. Design your biodata by using various AWT components.

ANS:

Code-

import java.awt.\*;

import java.awt.event.\*;

public class Biodata extends Frame {

    Label head, email, mob, line, aboutLabel, skillLabel, personalLabel, personalDetails[];

    TextArea aboutMe;

    List technicalSkills;

    Biodata(String title) {

        // Frame setup

        setLayout(null);

        setTitle(title);

        setVisible(true);

        setSize(500, 600);

        // Header

        head = new Label("ADITYA MAKWANA", Label.CENTER);

        head.setFont(new Font("Arial", Font.BOLD, 20));

        email = new Label("E-MAIL: adityamak707@gmail.com");

        mob = new Label("MOBILE: 9321033722");

        line = new Label("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

        // About Section

        aboutLabel = new Label("ABOUT ME");

        aboutLabel.setFont(new Font("Arial", Font.BOLD, 14));

        String aboutText = "A detail-oriented web developer with 2+ years of experience in web development and project management.\n " +

                 "Currently pursuing a post-SSC diploma at Vidyalankar Polytechnic (Third year). " +

                 "Skilled in public speaking, having anchored college events and guest lectures. \n" +

                 "Excelled academically with top grades, and proficient in SQL, computer graphics, and software engineering. " +

                 "Committed to continuous learning and exploring emerging technologies.";

        aboutMe = new TextArea(aboutText, 6, 50, TextArea.SCROLLBARS\_VERTICAL\_ONLY);

        // Technical Skills Section

        skillLabel = new Label("TECHNICAL SKILLS");

        skillLabel.setFont(new Font("Arial", Font.BOLD, 14));

        technicalSkills = new List(10);

        technicalSkills.add("Advanced Java");

        technicalSkills.add("HTML");

        technicalSkills.add("CSS");

        technicalSkills.add("JavaScript");

        technicalSkills.add("Node.js");

        // Personal Details Section

        personalLabel = new Label("PERSONAL DETAILS");

        personalLabel.setFont(new Font("Arial", Font.BOLD, 14));

        personalDetails = new Label[] {

                new Label("DOB: 12th June 2002"),

                new Label("Nationality: Indian"),

                new Label("Languages: English, Hindi, Marathi")

        };

        // Positioning Components

        head.setBounds(100, 40, 300, 30);

        email.setBounds(160, 80, 400, 20);

        mob.setBounds(300, 80, 200, 20);

        line.setBounds(20, 110, 460, 15);

        aboutLabel.setBounds(50, 140, 200, 20);

        aboutMe.setBounds(50, 170, 400, 150);

        skillLabel.setBounds(50, 330, 200, 20);

        technicalSkills.setBounds(50, 360, 400, 100);

        personalLabel.setBounds(50, 470, 200, 20);

        int personalY = 500;

        for (Label detail : personalDetails) {

            detail.setBounds(50, personalY, 400, 20);

            add(detail);

            personalY += 30;

        }

        // Add Components to Frame

        add(head);

        add(email);

        add(mob);

        add(line);

        add(aboutLabel);

        add(aboutMe);

        add(skillLabel);

        add(technicalSkills);

        add(personalLabel);

        // Window Closing Action

        addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {

                System.exit(0);

            }

        });

    }

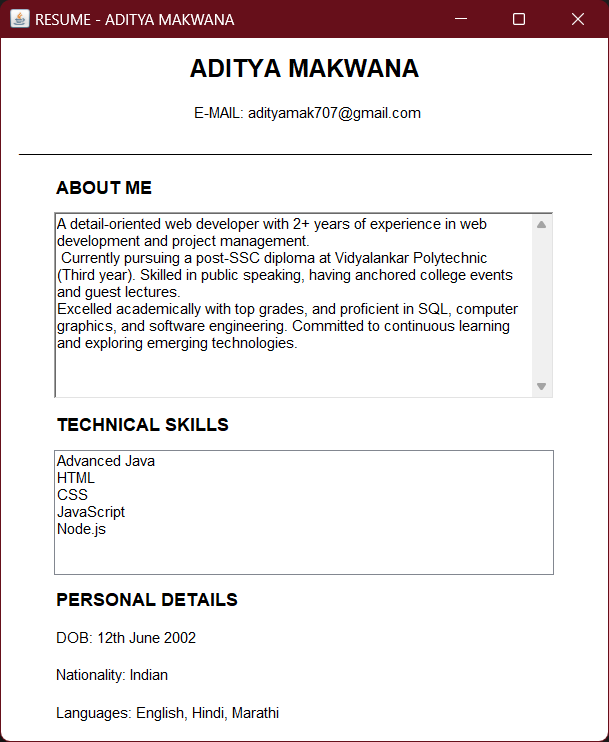
    public static void main(String[] args) {

        new Biodata("RESUME - ADITYA MAKWANA");

    }

}

OUTPUT-



1. Design an applet/Application using List components to add names of 10 different cities.

ANS:

Code-

import java.awt.\*;

public class CityListFrame extends Frame

{

 public CityListFrame()

 {

 // Set frame properties

 setTitle("City List Frame");

 setSize(400, 300);

 setLayout(new FlowLayout());

 setVisible(true);

 // Create a List component

 List cityList = new List(10, false); // 10 visible rows, single selection

 // Add city names to the List component

 cityList.add("New York");

 cityList.add("Los Angeles");

 cityList.add("Chicago");

 cityList.add("Houston");

 cityList.add("Phoenix");

 cityList.add("Philadelphia");

 cityList.add("San Antonio");

 cityList.add("San Diego");

 cityList.add("Dallas");

 cityList.add("San Jose");

 // Add the List component to the frame

 add(cityList);

 }

 public static void main(String[] args)

 {

 new CityListFrame();

 }

}

1. WAP to use Border Layout .

ANS:

Code-

import java.awt.\*;

public class BorderLayoutDemo extends Frame

{

 public BorderLayoutDemo()

 {

// Set frame properties

 setTitle("BorderLayout Demo");

 setSize(400, 300);

 setVisible(true);

 setLayout(new BorderLayout(5, 5)); // 5px horizontal and vertical gap

 // Add components to the frame for each BorderLayout region

 add(new Button("North"), BorderLayout.NORTH);

 add(new Button("South"), BorderLayout.SOUTH);

 add(new Button("East"), BorderLayout.EAST);

 add(new Button("West"), BorderLayout.WEST);

 add(new Button("Center"), BorderLayout.CENTER);

 }

 public static void main(String[] args)

 {

 new BorderLayoutDemo();

 }

}

1. WAP which creates Menu of different colors and disable menu item for Black color.

ANS:

Code-

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ColorMenuExample {

    public static void main(String[] args) {

        // Create the frame

        JFrame frame = new JFrame("Color Menu Example");

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

        frame.setSize(400, 300);

        // Create a menu bar

        JMenuBar menuBar = new JMenuBar();

        // Create a menu

        JMenu colorMenu = new JMenu("Colors");

        // Create menu items for different colors

        JMenuItem redItem = new JMenuItem("Red");

        JMenuItem greenItem = new JMenuItem("Green");

        JMenuItem blueItem = new JMenuItem("Blue");

        JMenuItem blackItem = new JMenuItem("Black");

        // Add action listeners to the menu items

        redItem.addActionListener(e -> frame.getContentPane().setBackground(Color.RED));

        greenItem.addActionListener(e -> frame.getContentPane().setBackground(Color.GREEN));

        blueItem.addActionListener(e -> frame.getContentPane().setBackground(Color.BLUE));

        // Disable the black color menu item

        blackItem.setEnabled(false);

        // Add menu items to the menu

        colorMenu.add(redItem);

        colorMenu.add(greenItem);

        colorMenu.add(blueItem);

        colorMenu.add(blackItem); // Black is disabled

        // Add the menu to the menu bar

        menuBar.add(colorMenu);

        // Set the menu bar for the frame

        frame.setJMenuBar(menuBar);

        // Set the initial background color

        frame.getContentPane().setBackground(Color.WHITE);

        // Make the frame visible

        frame.setVisible(true);

    }

}

1. WAP to develop a frame to select the different states of India using JComboBox

ANS:

Code-

import javax.swing.\*;

import java.awt.event.\*;

public class StateSelector {

    public static void main(String[] args) {

        // Create frame

        JFrame frame = new JFrame("State Selection");

        frame.setSize(300, 200);

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

        frame.setLayout(null);

        // Create components

        JLabel label = new JLabel("Select a State:");

        label.setBounds(30, 30, 100, 30);

        String[] states = { "Andhra Pradesh", "Assam", "Bihar", "Gujarat", "Karnataka", "Kerala", "Maharashtra",

                "Punjab", "Tamil Nadu", "West Bengal" };

        JComboBox<String> comboBox = new JComboBox<>(states);

        comboBox.setBounds(140, 30, 120, 30);

        JLabel selectedLabel = new JLabel("Selected: None");

        selectedLabel.setBounds(30, 80, 200, 30);

        // Add action listener

        comboBox.addActionListener(e -> selectedLabel.setText("Selected: " + comboBox.getSelectedItem()));

        // Add components to frame

        frame.add(label);

        frame.add(comboBox);

        frame.add(selectedLabel);

        frame.setVisible(true);

    }

}

1. Develop a program to demonstrate the use of tree component in swing.

ANS:

Code-

import javax.swing.\*;

import javax.swing.tree.DefaultMutableTreeNode;

import java.awt.\*;

public class SimpleTreeExample {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Simple JTree Example");

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

        frame.setSize(400, 300);

        DefaultMutableTreeNode root = new DefaultMutableTreeNode("File System");

        DefaultMutableTreeNode documents = new DefaultMutableTreeNode("Documents");

        DefaultMutableTreeNode pictures = new DefaultMutableTreeNode("Pictures");

        DefaultMutableTreeNode program = new DefaultMutableTreeNode("Program");

        root.add(documents);

        root.add(pictures);

        root.add(program);

        documents.add(new DefaultMutableTreeNode("Resume.docx"));

        documents.add(new DefaultMutableTreeNode("CoverLetter.docx"));

        pictures.add(new DefaultMutableTreeNode("Vacation.png"));

        pictures.add(new DefaultMutableTreeNode("Family.jpg"));

        program.add(new DefaultMutableTreeNode("Java.mp3"));

        program.add(new DefaultMutableTreeNode("Javascript.mp3"));

        JTree tree = new JTree(root);

        JScrollPane treeView = new JScrollPane(tree);

        frame.getContentPane().add(treeView, BorderLayout.CENTER);

        frame.setVisible(true);

    }

}

1. Develop a program to demonstrate the use of JTable.

ANS:

Code-

import javax.swing.\*;

import java.awt.\*;

public class JTableExample {

    public static void main(String[] args) {

       JFrame frame = new JFrame("JTable Example");

                // Sample data for the JTable

        String[][] data = {

            {"101", "Amit", "670000"},

            {"102", "Jai", "780000"},

            {"103", "Sachin", "700000"}

        };

        String[] columnNames = {"ID", "NAME", "SALARY"};

        JTable table = new JTable(data, columnNames);

        table.setAutoCreateRowSorter(true);

        JScrollPane scrollPane = new JScrollPane(table);

        frame.add(scrollPane);

        frame.setSize(500, 300);

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

        frame.setVisible(true);

    }

}

1. WAP to demonstrate various mouse events using MouseListener and MouseMotionListener interface

ANS:

Code-

import java.awt.\*;

import java.awt.event.\*;

public class MouseEventsFrame extends Frame implements MouseListener, MouseMotionListener {

    String msg = "";

    int mousex = 0, mousey = 50; // Initial coordinates of the mouse

    public MouseEventsFrame() {

        // Frame setup

        setTitle("Mouse Events Demo");

        setSize(400, 300);

        setLayout(null);

        setVisible(true);

        // Add Mouse Listeners

        addMouseListener(this);

        addMouseMotionListener(this);

        // Handle frame closing

        addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {

                System.exit(0);

            }

        });

    }

    // Handle mouse clicked

    public void mouseClicked(MouseEvent me) {

        msg = "Mouse clicked.";

        repaint();

    }

    // Handle mouse entered

    public void mouseEntered(MouseEvent me) {

        msg = "Mouse entered.";

        repaint();

    }

    // Handle mouse exited

    public void mouseExited(MouseEvent me) {

        msg = "Mouse exited.";

        repaint();

    }

    // Handle mouse pressed

    public void mousePressed(MouseEvent me) {

        mousex = me.getX();

        mousey = me.getY();

        msg = "Mouse pressed.";

        repaint();

    }

    // Handle mouse released

    public void mouseReleased(MouseEvent me) {

        mousex = me.getX();

        mousey = me.getY();

        msg = "Mouse released.";

        repaint();

    }

    // Handle mouse dragged

    public void mouseDragged(MouseEvent me) {

        mousex = me.getX();

        mousey = me.getY();

        msg = "Mouse dragging.";

        repaint();

    }

    // Handle mouse moved

    public void mouseMoved(MouseEvent me) {

        msg = "Mouse moved at " + me.getX() + ", " + me.getY();

        repaint();

    }

    // Paint the message on the frame

    public void paint(Graphics g) {

        g.drawString(msg, mousex, mousey);

    }

    public static void main(String[] args) {

        new MouseEventsFrame();

    }

}

1. WAP to demonstrate the use of JTextfield and JPasswordField using Listener interface

ANS:

Code-

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class JTextFieldJPasswordFieldExample {

    public static void main(String[] args) {

        // Create a new JFrame

        JFrame frame = new JFrame("Login Form Example");

        frame.setSize(250, 250);

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

        frame.setLayout(new FlowLayout());

        // Create labels

        JLabel userLabel = new JLabel("Username:");

        JLabel passLabel = new JLabel("Password:");

        JLabel messageLabel = new JLabel("Enter your details and click Login.");

        // Create text field and password field

        JTextField usernameField = new JTextField(15);

        JPasswordField passwordField = new JPasswordField(15);

        // Create a button

        JButton loginButton = new JButton("Login");

        // Add components to the frame

        frame.add(userLabel);

        frame.add(usernameField);

        frame.add(passLabel);

        frame.add(passwordField);

        frame.add(loginButton);

        frame.add(messageLabel); // Add message label

        // Add an ActionListener to the button

        loginButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                // Retrieve the username and password

                String username = usernameField.getText();

                String password = new String(passwordField.getPassword());

                // Update the message label to display username and password

                messageLabel.setText("<html>Username: " + username + "<br>Password: " + password + "</html>");

            }

        });

        // Make the frame visible

        frame.setVisible(true);

    }

}

1. WAP to demonstrate the use of WindowAdapter class

ANS:

Code-

import java.awt.\*;

import java.awt.event.\*;

public class WindowAdapterExample {

    public static void main(String[] args) {

        Frame frame = new Frame("WindowAdapter Example");

        frame.setSize(400, 300);

        frame.setLayout(null);

        Label label = new Label("Close the window to see the adapter in action!");

        label.setBounds(50, 100, 300, 30);

        frame.add(label);

        frame.addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {

                frame.dispose();

            }

        });

        frame.setVisible(true);

        frame.addWindowListener(new WindowAdapter() {

            public void windowClosed(WindowEvent e) {

                System.exit(0);

            }

        });

    }

}

1. WAP to demonstrate the use of InetAddress class and its factory methods

ANS:

Code-

import java.io.\*;

import java.net.\*;

public class InetDemo {

    public static void main(String[] args) {

        try {

            InetAddress ip = InetAddress.getByName("localhost");

            System.out.println("Host Name: " + ip.getHostName());

            System.out.println("IP Address: " + ip.getHostAddress());

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}

1. WAP to demonstrate the use of URL and URLConnection class and its methods

ANS:

URL CLASS CODE-

*import java.net.URL;*

*import java.net.MalformedURLException;*

*public class URLRetrive*

*{*

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

*URL url = new URL("https://msbte.org.in/");*

*System.out.println("Authority: "+ url.getAuthority());*

*System.out.println("Default Port: "+ url.getDefaultPort());*

*System.out.println("File: "+ url.getFile());*

*System.out.println("Path: "+ url.getPath());*

*System.out.println("Protocol: "+ url.getProtocol());*

*System.out.println("Reference: "+ url.getRef());*

*}*

*}*

URL CONNECTION CODE-

import java.net.\*;

import java.io.\*;

import java.util.Date;

public class UCDemo {

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

        int c;

        URL hp = new URL("http://www.internic.net");

        URLConnection hpCon = hp.openConnection();

        // get date

        long d = hpCon.getDate();

        if (d == 0)

            System.out.println("No date information.");

        else

            System.out.println("Date: " + new Date(d));

        // get content type

        System.out.println("Content-Type: " + hpCon.getContentType());

        // get expiration date

        d = hpCon.getExpiration();

        if (d == 0)

            System.out.println("No expiration information.");

        else

            System.out.println("Expires: " + new Date(d));

        // get last-modified date

        d = hpCon.getLastModified();

        if (d == 0)

            System.out.println("No last-modified information.");

        else

            System.out.println("Last-Modified: " + new Date(d));

        // get content length

        int len = hpCon.getContentLength();

        if (len == -1)

            System.out.println("Content length unavailable.");

        else

            System.out.println("Content-Length: " + len);

        if (len != 0) {

            System.out.println("=== Content ===");

            InputStream input = hpCon.getInputStream();

            int i = len;

            while (((c = input.read()) != -1)) { // && (--i > 0)) {

                System.out.print((char) c);

            }

            input.close();

        } else {

            System.out.println("No content available.");

        }

    }

}

1. WAP to insert and retrieve the data from database using JDBC

ANS:

Code-

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package databasepractice;

import java.sql.\*;

public class DATABASEPRACTICE {

static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

static final String DB\_URL = "jdbc:mysql://localhost/student";

static final String USER = "root";

static final String PASS = "";

public static void main(String[] args) {

Connection conn = null;

Statement stmt = null;

try{

//STEP 2: Register JDBC driver

Class.forName(JDBC\_DRIVER);

//STEP 3: Open a connection

System.out.println("Connecting to a selected database...");

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

System.out.println("Connected database successfully...");

//STEP 4: Execute a query

System.out.println("Inserting records into the table...");

stmt = conn.createStatement();

String sql = "INSERT INTO registration (name,percentage) VALUES ('ADITYA',99)";

stmt.executeUpdate(sql);

sql = "INSERT INTO registration (name,percentage) VALUES ('SIDDHARTH',99)";

stmt.executeUpdate(sql);

System.out.println("Inserted records into the table...");

String query = "SELECT \* FROM registration";

// execute the query, and get a java resultset

ResultSet rs = stmt.executeQuery(query);

// iterate through the java resultset

while (rs.next())

{

int stud\_id = rs.getInt("stud\_id");

String name = rs.getString("name");

int per = rs.getInt("percentage");

// print the results

System.out.format("%s, %s, %s\n", stud\_id, name, per);

}

}catch(SQLException se){

//Handle errors for JDBC

se.printStackTrace();

}catch(Exception e){

//Handle errors for Class.forName

e.printStackTrace();

}finally{

//finally block used to close resources

try{

if(stmt!=null)

conn.close();

}catch(SQLException se){

}// do nothing

try{

if(conn!=null)

conn.close();

}catch(SQLException se){

se.printStackTrace();

}//end finally try

}//end try

System.out.println("Goodbye!");

}

}

OUTPUT-

A white background with black text

Description automatically generated

1. WAP servlet to send username and password using HTML forms and authenticate the user

ANS:

Servlet code-

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* LoginServlet handles user login authentication.

\*/

@WebServlet("/LoginServlet")

public class LoginServlet extends HttpServlet {

/\*\*

\* Handles HTTP POST requests for user authentication.

\*

\* @param request HttpServletRequest object

\* @param response HttpServletResponse object

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

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

response.setContentType("text/html");

PrintWriter out = response.getWriter();

try {

if ("admin".equals(username) && "password123".equals(password)) {

out.println("<h2>Login Successful!</h2>");

} else {

out.println("<h2>Invalid Username or Password.</h2>");

}

} finally {

out.close(); // Ensure PrintWriter is closed to free resources

}

}

/\*\*

\* Provides a brief description of the servlet.

\*

\* @return A String describing the servlet

\*/

@Override

public String getServletInfo() {

return "LoginServlet: Handles user login authentication.";

}

}

Html code-

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <title>Login</title> |
|  | </head> |
|  | <body> |
|  | <h2>Login Form</h2> |
|  | <form action="LoginServlet" method="POST"> |
|  | <label for="username">Username:</label> |
|  | <input type="text" id="username" name="username" required><br><br> |
|  | <label for="password">Password:</label> |
|  | <input type="password" id="password" name="password" required><br><br> |
|  | <button type="submit">Login</button> |
|  | </form> |
|  | </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>LoginServlet</servlet-name>

<servlet-class>LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/LoginServlet</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout>

30

</session-timeout>

</session-config>

</web-app>

Output-

