RECIPE RECOMMENDATION

A mini project

By

SOWMYA R 230701328

SHALINI R K 230701304

PROJECT CODING

SAMPLE CODE

1.1 HOME PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;

public class Dashboard extends JFrame implements ActionListener {
    private JPanel contentPanel;
    private BufferedImage backgroundImage;

    public Dashboard() {
        loadBackgroundImage();
        setupFrame();
        setupContentPanel();
        setVisible(true);
    }

    // Load the background image from resources
    private void loadBackgroundImage() {
        try {
            backgroundImage =
ImageIO.read(getClass().getResource("/resources/images/food.jpg"));
```

```
        } catch (IOException | NullPointerException e) {
            JOptionPane.showMessageDialog(this, "Error loading background image: 
" + e.getMessage());
            e.printStackTrace();
        }
    }

    // Set up the main frame properties
    private void setupFrame() {
        setTitle("CHEFMATE");
        setSize(900, 650);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        setLocationRelativeTo(null);
    }

    // Set up the main content panel with a background image and buttons in the
    // center
    private void setupContentPanel() {
        contentPanel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                if (backgroundImage != null) {
                    g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
                }
            }
        };
        contentPanel.setLayout(new GridBagLayout()); // Center the components
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 0, 10, 0); // Add vertical spacing between buttons
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Add buttons for each cuisine
        addCenteredButton("Italian", gbc);
        addCenteredButton("Korean", gbc);
        addCenteredButton("Indian", gbc);
        addCenteredButton("Chinese", gbc);
        addCenteredButton("Japanese", gbc);
        addCenteredButton("French", gbc);
        add(contentPanel, BorderLayout.CENTER);
    }

    // Create and style centered buttons
    private void addCenteredButton(String text, GridBagConstraints gbc) {
        JButton button = new JButton(text);
        button.setPreferredSize(new Dimension(200, 60));
        button.setForeground(Color.WHITE);
        button.setBackground(Color.BLACK);
        Border border = BorderFactory.createLineBorder(Color.WHITE,2);
        button.setBorder(border);
        button.setFocusPainted(false);
        button.addActionListener(this);
        gbc.gridy++; // Move the button position down for each new button
        contentPanel.add(button, gbc);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String action = e.getActionCommand();
        switch (action) {
            case "Italian":
                openItalianPage();
                break;
            case "Korean":
                openKoreanPage();
                break;
            case "Indian":
                openIndianPage();
                break;
            case "Chinese":
                openChinesePage();
                break;
            case "Japanese":
                openJapanesePage();
                break;
            case "French":
                openFrenchPage();
                break;
            default:
                JOptionPane.showMessageDialog(this, "Unknown action!");
        }
    }

    // Method to display a specific cuisine page
    private void openItalianPage() {
        ItalianPage ipage = new ItalianPage();
        ipage.setLocationRelativeTo(null);
        ipage.setVisible(true);

    }
    private void openIndianPage() {
        IndianPage inpage = new IndianPage();
        inpage.setLocationRelativeTo(null);
        inpage.setVisible(true); // Open the ClientPage window
    }
    private void openChinesePage() {
        ChinesePage cpage = new ChinesePage();
        cpage.setLocationRelativeTo(null);
        cpage.setVisible(true);
    }
    private void openJapanesePage() {
        JapanesePage jpage = new JapanesePage();
        jpage.setLocationRelativeTo(null);
        jpage.setVisible(true);
    }
    private void openKoreanPage() {
        KoreanPage kpage = new KoreanPage();
        kpage.setLocationRelativeTo(null);
        kpage.setVisible(true);
    }
   private void openFrenchPage() {
        FrenchPage fpage = new FrenchPage();
        fpage.setLocationRelativeTo(null);
        fpage.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Dashboard::new);
    }
 }
```

1.2 ITALIAN FOOD PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class ItalianPage extends JFrame {
    private JLabel nameLabel;
    private JTextArea descriptionArea;
    private JButton instructionButton;
    private JButton nextButton;
    private JLabel imageLabel;
    private Connection connection;
    private ResultSet resultSet;

    public ItalianPage() {
        // Frame settings
        setTitle("Italian Recipes");
        setSize(900, 550);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create main panel with two sections: image and text
        JPanel mainPanel = new JPanel(new GridLayout(1, 2));
        add(mainPanel, BorderLayout.CENTER);

        // Image panel
        JPanel imagePanel = new JPanel();
        imagePanel.setLayout(new BorderLayout());
        imageLabel = new JLabel();
        imageLabel.setHorizontalAlignment(JLabel.CENTER);
        imagePanel.add(imageLabel, BorderLayout.CENTER);
        imagePanel.setBackground(new Color(255,218,218));
        mainPanel.add(imagePanel);

        // Text panel
        JPanel textPanel = new JPanel();
        textPanel.setLayout(new BorderLayout());

        nameLabel = new JLabel("", JLabel.CENTER);
        nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
        textPanel.add(nameLabel, BorderLayout.NORTH);

        descriptionArea = new JTextArea();
        descriptionArea.setLineWrap(true);
        descriptionArea.setWrapStyleWord(true);
        descriptionArea.setEditable(false);
        descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_A
S_NEEDED);
        textPanel.add(scrollPane, BorderLayout.CENTER);

        // Button panel
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,
20, 10));
        instructionButton = new JButton("Click here for Instructions");
        nextButton = new JButton("Next");

        buttonPanel.add(instructionButton);
        buttonPanel.add(nextButton);
        textPanel.add(buttonPanel, BorderLayout.SOUTH);

        mainPanel.add(textPanel);

        // Database connection setup
        try {
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root",
"Shalini@2005");
            Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            resultSet = statement.executeQuery("SELECT * FROM italian");

            if (resultSet.next()) {
                displayRecipe(resultSet);
            } else {
                JOptionPane.showMessageDialog(this, "No recipes found in the
database.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());
        }

        // Action listeners
        instructionButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showInstructions();
            }
        });

        nextButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (resultSet.next()) {
                        displayRecipe(resultSet);
                    } else {
                        // Move to the first row if we reach the end
                        resultSet.beforeFirst();
                        resultSet.next();
                        displayRecipe(resultSet);
                    }
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(ItalianPage.this, "Error retrieving
next recipe: " + ex.getMessage());
                }
            }
        });
    }

    private void displayRecipe(ResultSet resultSet) {
        try {
            String name = resultSet.getString("name");
            String description = resultSet.getString("description");

            nameLabel.setText(name);
            descriptionArea.setText(description);

            // Load the image from resources and display it
            ImageIcon icon = new
ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
            Image scaledImage = icon.getImage().getScaledInstance(400, 450,
Image.SCALE_SMOOTH); // Adjust image size as needed
            imageLabel.setIcon(new ImageIcon(scaledImage));
            Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
change
            imageLabel.setBorder(border);

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
e.getMessage());
        }
    }

    private void showInstructions() {
        try {
            String instructions = resultSet.getString("instruction");
            JTextArea instructionArea = new JTextArea(instructions);
            instructionArea.setWrapStyleWord(true);
            instructionArea.setLineWrap(true);
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 
10));

            JScrollPane scrollPane = new JScrollPane(instructionArea);
            scrollPane.setPreferredSize(new Dimension(400, 200));

            JOptionPane.showMessageDialog(this, scrollPane, "Instructions", 
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error retrieving instructions: " + 
e.getMessage());
        }
    }

    // Main method for running the page

}
```

1.3 KOREAN FOOD PAGE DESIGN:

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class KoreanPage extends JFrame {
```

```
private JLabel nameLabel;
private JTextArea descriptionArea;
private JButton instructionButton;
private JButton nextButton;
private JLabel imageLabel;
private Connection connection;
private ResultSet resultSet;

public KoreanPage() {
    // Frame settings
    setTitle("Chinese Recipes");
    setSize(900, 550);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());

    // Create main panel with two sections: image and text
    JPanel mainPanel = new JPanel(new GridLayout(1, 2));
    add(mainPanel, BorderLayout.CENTER);

    // Image panel
    JPanel imagePanel = new JPanel();
    imagePanel.setLayout(new BorderLayout());
    imageLabel = new JLabel();
    imageLabel.setHorizontalAlignment(JLabel.CENTER);
    imagePanel.add(imageLabel, BorderLayout.CENTER);
    imagePanel.setBackground(new Color(255,193,254));
    mainPanel.add(imagePanel);

    // Text panel
    JPanel textPanel = new JPanel();
    textPanel.setLayout(new BorderLayout());

    nameLabel = new JLabel("", JLabel.CENTER);
    nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
    textPanel.add(nameLabel, BorderLayout.NORTH);

    descriptionArea = new JTextArea();
    descriptionArea.setLineWrap(true);
    descriptionArea.setWrapStyleWord(true);
    descriptionArea.setEditable(false);
```

```
        descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_A
S_NEEDED);
        textPanel.add(scrollPane, BorderLayout.CENTER);

        // Button panel
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,
20, 10));
        instructionButton = new JButton("Click here for Instructions");
        nextButton = new JButton("Next");

        buttonPanel.add(instructionButton);
        buttonPanel.add(nextButton);
        textPanel.add(buttonPanel, BorderLayout.SOUTH);

        mainPanel.add(textPanel);

        // Database connection setup
        try {
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root",
"Shalini@2005");
            Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            resultSet = statement.executeQuery("SELECT * FROM korean");

            if (resultSet.next()) {
                displayRecipe(resultSet);
            } else {
                JOptionPane.showMessageDialog(this, "No recipes found in the
database.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());
        }

        // Action listeners
        instructionButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showInstructions();
            }
        });

        nextButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (resultSet.next()) {
                        displayRecipe(resultSet);
                    } else {
                        // Move to the first row if we reach the end
                        resultSet.beforeFirst();
                        resultSet.next();
                        displayRecipe(resultSet);
                    }
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(KoreanPage.this, "Error retrieving
next recipe: " + ex.getMessage());
                }
            }
        });
    }

    private void displayRecipe(ResultSet resultSet) {
        try {
            String name = resultSet.getString("name");
            String description = resultSet.getString("description");

            nameLabel.setText(name);
            descriptionArea.setText(description);

            // Load the image from resources and display it
            ImageIcon icon = new
ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
            Image scaledImage = icon.getImage().getScaledInstance(400, 450,
Image.SCALE_SMOOTH); // Adjust image size as needed
            imageLabel.setIcon(new ImageIcon(scaledImage));
            Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
change
            imageLabel.setBorder(border);

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
e.getMessage());
        }
    }

    private void showInstructions() {
        try {
            String instructions = resultSet.getString("instruction");
            JTextArea instructionArea = new JTextArea(instructions);
            instructionArea.setWrapStyleWord(true);
            instructionArea.setLineWrap(true);
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

            JScrollPane scrollPane = new JScrollPane(instructionArea);
            scrollPane.setPreferredSize(new Dimension(400, 200));

            JOptionPane.showMessageDialog(this, scrollPane, "Instructions",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error retrieving instructions: " +
e.getMessage());
        }
    }
}
```

1.4 INDIAN FOOD PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class IndianPage extends JFrame {
    private JLabel nameLabel;
    private JTextArea descriptionArea;
    private JButton instructionButton;
    private JButton nextButton;
    private JLabel imageLabel;
    private Connection connection;
    private ResultSet resultSet;

    public IndianPage() {
        // Frame settings
        setTitle("Indian Recipes");
        setSize(900, 550);//change
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create main panel with two sections: image and text
        JPanel mainPanel = new JPanel(new GridLayout(1, 2));
        add(mainPanel, BorderLayout.CENTER);

        // Image panel
```

```
  JPanel imagePanel = new JPanel();
  imagePanel.setLayout(new BorderLayout());
  imageLabel = new JLabel();
  imageLabel.setHorizontalAlignment(JLabel.CENTER);
  imagePanel.add(imageLabel, BorderLayout.CENTER);
  imagePanel.setBackground(new Color(153,156,246));//image color
  mainPanel.add(imagePanel);

  // Text panel
  JPanel textPanel = new JPanel();
  textPanel.setLayout(new BorderLayout());

  nameLabel = new JLabel("", JLabel.CENTER);
  nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
  textPanel.add(nameLabel, BorderLayout.NORTH);
  descriptionArea = new JTextArea();
  descriptionArea.setLineWrap(true);
  descriptionArea.setWrapStyleWord(true);
  descriptionArea.setEditable(false);
  descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
  descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

  JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_A
S_NEEDED);
  textPanel.add(scrollPane, BorderLayout.CENTER);

  // Button panel
  JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20,
10));
  instructionButton = new JButton("Click here for Instructions");
  nextButton = new JButton("Next");

  buttonPanel.add(instructionButton);
  buttonPanel.add(nextButton);
  textPanel.add(buttonPanel, BorderLayout.SOUTH);

  mainPanel.add(textPanel);

  // Database connection setup
  try {
      connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root",
"Shalini@2005");
      Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
      resultSet = statement.executeQuery("SELECT * FROM indian");

      if (resultSet.next()) {
          displayRecipe(resultSet);
      } else {
          JOptionPane.showMessageDialog(this, "No recipes found in the
database.");
      }
  } catch (SQLException e) {
      JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());
  }

  // Action listeners
  instructionButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
            showInstructions();
        }
    });

    nextButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                if (resultSet.next()) {
                    displayRecipe(resultSet);
                } else {
                    // Move to the first row if we reach the end
                    resultSet.beforeFirst();
                    resultSet.next();
                    displayRecipe(resultSet);
                }
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(IndianPage.this, "Error retrieving next
  recipe: " + ex.getMessage());
            }
        }
    });
}

private void displayRecipe(ResultSet resultSet) {
    try {
        String name = resultSet.getString("name");
        String description = resultSet.getString("description");

        nameLabel.setText(name);
        descriptionArea.setText(description);
        // Load the image from resources and display it
        ImageIcon icon = new
  ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
        Image scaledImage = icon.getImage().getScaledInstance(400, 450,
  Image.SCALE_SMOOTH); // change image height
        imageLabel.setIcon(new ImageIcon(scaledImage));
        Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
  change
        imageLabel.setBorder(border);//border change

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
  e.getMessage());
    }
}

private void showInstructions() {
    try {
        String instructions = resultSet.getString("instruction");
        JTextArea instructionArea = new JTextArea(instructions);
        instructionArea.setWrapStyleWord(true);
        instructionArea.setLineWrap(true);
        instructionArea.setEditable(false);
        instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
        instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        JScrollPane scrollPane = new JScrollPane(instructionArea);
        scrollPane.setPreferredSize(new Dimension(400, 200));

        JOptionPane.showMessageDialog(this, scrollPane, "Instructions",
  JOptionPane.INFORMATION_MESSAGE);
```

```
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error retrieving instructions: " +
  e.getMessage());
    }
}

// Main method for running the page
  }
```

1.5 CHINESE FOOD PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class ChinesePage extends JFrame {
    private JLabel nameLabel;
    private JTextArea descriptionArea;
    private JButton instructionButton;
    private JButton nextButton;
    private JLabel imageLabel;
    private Connection connection;
    private ResultSet resultSet;

    public ChinesePage() {
        // Frame settings
        setTitle("Chinese Recipes");
        setSize(900, 550);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create main panel with two sections: image and text
        JPanel mainPanel = new JPanel(new GridLayout(1, 2));
        add(mainPanel, BorderLayout.CENTER);
        // Image panel
        JPanel imagePanel = new JPanel();
        imagePanel.setLayout(new BorderLayout());
        imageLabel = new JLabel();
        imageLabel.setHorizontalAlignment(JLabel.CENTER);
        imagePanel.add(imageLabel, BorderLayout.CENTER);
        imagePanel.setBackground(new Color(215,249,210));
        mainPanel.add(imagePanel);

        // Text panel
        JPanel textPanel = new JPanel();
        textPanel.setLayout(new BorderLayout());

        nameLabel = new JLabel("", JLabel.CENTER);
        nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
        textPanel.add(nameLabel, BorderLayout.NORTH);

        descriptionArea = new JTextArea();
        descriptionArea.setLineWrap(true);
        descriptionArea.setWrapStyleWord(true);
        descriptionArea.setEditable(false);
        descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_
NEEDED);
        textPanel.add(scrollPane, BorderLayout.CENTER);

        // Button panel
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20,
10));
        instructionButton = new JButton("Click here for Instructions");
        nextButton = new JButton("Next");

        buttonPanel.add(instructionButton);
        buttonPanel.add(nextButton);
        textPanel.add(buttonPanel, BorderLayout.SOUTH);
        mainPanel.add(textPanel);

        // Database connection setup
        try {
            connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root", 
"Shalini@2005");
            Statement statement = 
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, 
ResultSet.CONCUR_READ_ONLY);
            resultSet = statement.executeQuery("SELECT * FROM chinese");

            if (resultSet.next()) {
                displayRecipe(resultSet);
            } else {
                JOptionPane.showMessageDialog(this, "No recipes found in the 
database.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Database connection failed: " + 
e.getMessage());
        }

        // Action listeners
        instructionButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showInstructions();
            }
        });

        nextButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (resultSet.next()) {
                        displayRecipe(resultSet);
                    } else {
                        // Move to the first row if we reach the end
                        resultSet.beforeFirst();
                        resultSet.next();
                        displayRecipe(resultSet);
                    }
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(ChinesePage.this, "Error retrieving
next recipe: " + ex.getMessage());
                }
            }
        });
}

private void displayRecipe(ResultSet resultSet) {
        try {
            String name = resultSet.getString("name");
            String description = resultSet.getString("description");

            nameLabel.setText(name);
            descriptionArea.setText(description);

            // Load the image from resources and display it
            ImageIcon icon = new
ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
            Image scaledImage = icon.getImage().getScaledInstance(400, 450,
Image.SCALE_SMOOTH); // Adjust image size as needed
            imageLabel.setIcon(new ImageIcon(scaledImage));
            Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
change
            imageLabel.setBorder(border);

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
e.getMessage());
        }
}

private void showInstructions() {
        try {
            String instructions = resultSet.getString("instruction");
            JTextArea instructionArea = new JTextArea(instructions);
            instructionArea.setWrapStyleWord(true);
```

```
            instructionArea.setLineWrap(true);
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

            JScrollPane scrollPane = new JScrollPane(instructionArea);
            scrollPane.setPreferredSize(new Dimension(400, 200));

            JOptionPane.showMessageDialog(this, scrollPane, "Instructions",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error retrieving instructions: " +
e.getMessage());
        }
    }

    // Main method for running the page

}
```

1.6 JAPANESE FOOD PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class JapanesePage extends JFrame {
    private JLabel nameLabel;
    private JTextArea descriptionArea;
    private JButton instructionButton;
    private JButton nextButton;
    private JLabel imageLabel;
    private Connection connection;
    private ResultSet resultSet;

    public JapanesePage() {
        // Frame settings
        setTitle("Japanese Recipes");
        setSize(900, 550);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create main panel with two sections: image and text
        JPanel mainPanel = new JPanel(new GridLayout(1, 2));
        add(mainPanel, BorderLayout.CENTER);

        // Image panel
        JPanel imagePanel = new JPanel();
        imagePanel.setLayout(new BorderLayout());
        imageLabel = new JLabel();
        imageLabel.setHorizontalAlignment(JLabel.CENTER);
        imagePanel.add(imageLabel, BorderLayout.CENTER);
        imagePanel.setBackground(new Color(255,131,152));
        mainPanel.add(imagePanel);

        // Text panel
        JPanel textPanel = new JPanel();
        textPanel.setLayout(new BorderLayout());

        nameLabel = new JLabel("", JLabel.CENTER);
        nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
        textPanel.add(nameLabel, BorderLayout.NORTH);

        descriptionArea = new JTextArea();
        descriptionArea.setLineWrap(true);
        descriptionArea.setWrapStyleWord(true);
        descriptionArea.setEditable(false);
        descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_A
S_NEEDED);
        textPanel.add(scrollPane, BorderLayout.CENTER);
        // Button panel
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,
20, 10));
        instructionButton = new JButton("Click here for Instructions");
        nextButton = new JButton("Next");

        buttonPanel.add(instructionButton);
        buttonPanel.add(nextButton);
        textPanel.add(buttonPanel, BorderLayout.SOUTH);

        mainPanel.add(textPanel);

        // Database connection setup
        try {
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root",
"Shalini@2005");
            Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            resultSet = statement.executeQuery("SELECT * FROM japanese");

            if (resultSet.next()) {
                displayRecipe(resultSet);
            } else {
                JOptionPane.showMessageDialog(this, "No recipes found in the
database.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());
        }

        // Action listeners
        instructionButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showInstructions();
            }
        });

        nextButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (resultSet.next()) {
                        displayRecipe(resultSet);
                    } else {
                        // Move to the first row if we reach the end
                        resultSet.beforeFirst();
                        resultSet.next();
                        displayRecipe(resultSet);
                    }
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(JapanesePage.this, "Error retrieving
next recipe: " + ex.getMessage());
                }
            }
        });
    }

    private void displayRecipe(ResultSet resultSet) {
        try {
            String name = resultSet.getString("name");
            String description = resultSet.getString("description");

            nameLabel.setText(name);
            descriptionArea.setText(description);

            // Load the image from resources and display it
            ImageIcon icon = new
ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
            Image scaledImage = icon.getImage().getScaledInstance(400, 450,
Image.SCALE_SMOOTH); // Adjust image size as needed
            imageLabel.setIcon(new ImageIcon(scaledImage));
            Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
change
            imageLabel.setBorder(border);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
e.getMessage());
        }
    }

    private void showInstructions() {
        try {
            String instructions = resultSet.getString("instruction");
            JTextArea instructionArea = new JTextArea(instructions);
            instructionArea.setWrapStyleWord(true);
            instructionArea.setLineWrap(true);
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

            JScrollPane scrollPane = new JScrollPane(instructionArea);
            scrollPane.setPreferredSize(new Dimension(400, 200));

            JOptionPane.showMessageDialog(this, scrollPane, "Instructions",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error retrieving instructions: " +
e.getMessage());
        }
    }

}
```

1.7 FRANCHISE FOOD PAGE DESIGN

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
```

```
public class FrenchPage extends JFrame {
    private JLabel nameLabel;
    private JTextArea descriptionArea;
    private JButton instructionButton;
    private JButton nextButton;
    private JLabel imageLabel;
    private Connection connection;
    private ResultSet resultSet;

    public FrenchPage() {
        // Frame settings
        setTitle("French Recipes");
        setSize(900, 550);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Create main panel with two sections: image and text
        JPanel mainPanel = new JPanel(new GridLayout(1, 2));
        add(mainPanel, BorderLayout.CENTER);

        // Image panel
        JPanel imagePanel = new JPanel();
        imagePanel.setLayout(new BorderLayout());
        imageLabel = new JLabel();
        imageLabel.setHorizontalAlignment(JLabel.CENTER);
        imagePanel.add(imageLabel, BorderLayout.CENTER);
        imagePanel.setBackground(new Color(255,249,203));
        mainPanel.add(imagePanel);

        // Text panel
        JPanel textPanel = new JPanel();
        textPanel.setLayout(new BorderLayout());

        nameLabel = new JLabel("", JLabel.CENTER);
        nameLabel.setFont(new Font("Arial", Font.BOLD, 24));
        textPanel.add(nameLabel, BorderLayout.NORTH);

        descriptionArea = new JTextArea();
        descriptionArea.setLineWrap(true);
        descriptionArea.setWrapStyleWord(true);
        descriptionArea.setEditable(false);
        descriptionArea.setFont(new Font("Arial", Font.PLAIN, 16));
        descriptionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JScrollPane scrollPane = new JScrollPane(descriptionArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_A
S_NEEDED);
        textPanel.add(scrollPane, BorderLayout.CENTER);

        // Button panel
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER,
20, 10));
        instructionButton = new JButton("Click here for Instructions");
        nextButton = new JButton("Next");

        buttonPanel.add(instructionButton);
        buttonPanel.add(nextButton);
        textPanel.add(buttonPanel, BorderLayout.SOUTH);

        mainPanel.add(textPanel);

        // Database connection setup
        try {
            connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/cuisine", "root",
"Shalini@2005");
            Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            resultSet = statement.executeQuery("SELECT * FROM french");

            if (resultSet.next()) {
                displayRecipe(resultSet);
            } else {
                JOptionPane.showMessageDialog(this, "No recipes found in the
database.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Database connection failed: " +
e.getMessage());
        }

        // Action listeners
        instructionButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                showInstructions();
            }
        });

        nextButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (resultSet.next()) {
                        displayRecipe(resultSet);
                    } else {
                        // Move to the first row if we reach the end
                        resultSet.beforeFirst();
                        resultSet.next();
                        displayRecipe(resultSet);
                    }
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(FrenchPage.this, "Error retrieving
next recipe: " + ex.getMessage());
                }
            }
        });
    }

    private void displayRecipe(ResultSet resultSet) {
        try {
            String name = resultSet.getString("name");
            String description = resultSet.getString("description");

            nameLabel.setText(name);
            descriptionArea.setText(description);
            // Load the image from resources and display it
            ImageIcon icon = new
ImageIcon(getClass().getResource("/resources/images/"+ name+".jpg"));
            Image scaledImage = icon.getImage().getScaledInstance(400, 450,
Image.SCALE_SMOOTH); // Adjust image size as needed
            imageLabel.setIcon(new ImageIcon(scaledImage));
            Border border = BorderFactory.createLineBorder(Color.WHITE,2);//border
change
            imageLabel.setBorder(border);

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error displaying recipe: " +
e.getMessage());
        }
    }

    private void showInstructions() {
        try {
            String instructions = resultSet.getString("instruction");
            JTextArea instructionArea = new JTextArea(instructions);
            instructionArea.setWrapStyleWord(true);
            instructionArea.setLineWrap(true);
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

            JScrollPane scrollPane = new JScrollPane(instructionArea);
            scrollPane.setPreferredSize(new Dimension(400, 200));

            JOptionPane.showMessageDialog(this, scrollPane, "Instructions",
JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "Error retrieving instructions: " +
e.getMessage());
        }
}
}
```