RECIPE RECOMMENDATION

A MINI-PROJECT BY:

SHALINI R K   230701304

SOWMYA R   230701328

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **“RECIPE RECOMMENDATION”** is the bonafide work of **“SOWMYA R , SHALINI R K”** who carried out the project work under my supervision.

Submitted for the practical examination held on ____________________

SIGNATURE

Mr.G.SARAVANA GOKUL
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

SIGNATURE

Ms.V.JANANEE
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Recipe Recommendation App is designed to revolutionize meal planning and cooking by providing users with tailored recipe suggestions based on their available ingredients, dietary preferences, and cuisine interests. The app simplifies the cooking process with features like ingredient-based searches, dynamic filtering, and step-by-step instructions. It enhances user experience by offering personalized recommendations, saving favorite recipes, and generating shopping lists for missing ingredients. This innovative solution promotes efficient use of resources, reduces food waste, and caters to diverse culinary needs. With its intuitive design and versatile functionality, the app serves as an essential tool for home cooks and food enthusiasts.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

A Recipe Recommendation App is a user-friendly platform designed to make meal planning and cooking easier. The app allows users to find recipes based on the ingredients they have, dietary preferences, and favorite cuisines. With features like ingredient-based searches, personalized recommendations, and dynamic filters, it simplifies the process of discovering new and exciting dishes. Users can save their favorite recipes, access step-by-step cooking instructions, and even generate shopping lists for missing ingredients. This app serves as a practical tool for saving time, reducing food waste, and enhancing creativity in the kitchen.

1.2 IMPLEMENTATION

The **RECIPE RECOMMENDATION** project discussed here is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.3 SCOPE OF THE PROJECT

The Recipe Recommendation App offers a wide scope, including ingredient-based recipe suggestions, personalized recommendations, and meal planning support. It caters to diverse dietary needs, reduces food waste, and enhances cooking convenience. By integrating features like dynamic filters, favorites, and shopping lists, the app serves as a versatile tool for everyday culinary exploration.

1.4 WEBSITE FEATURES

1.4.1 Home page.

1.4.2 Countries to be chosen.

1.4.3 Recipe recommendation on foods.

1.4.4 Instructions for preparing.

1.4.5 Visualization of foods.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR	:	Intel i5
MEMORY SIZE	:	4GB(Minimum)
HARD DISK	:	500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE	:	Java, MySQL
FRONT-END	:	Java
BACK-END	:	MySQL
OPERATING SYSTEM	:	Windows 11

SAMPLE CODE

3.1 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);
    }
 }
```

3.2 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_AS_NEEDE
D);
        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

}
```

3.3 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_AS_NEEDE
D);
        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());
        }
    }
}
```

3.4 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_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 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
  }
```

3.5 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

}
```

3.6 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_AS_NEEDE
D);
        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());
        }
    }

}
```

3.7 FRANCHISE FOOD PAGE DESIGN

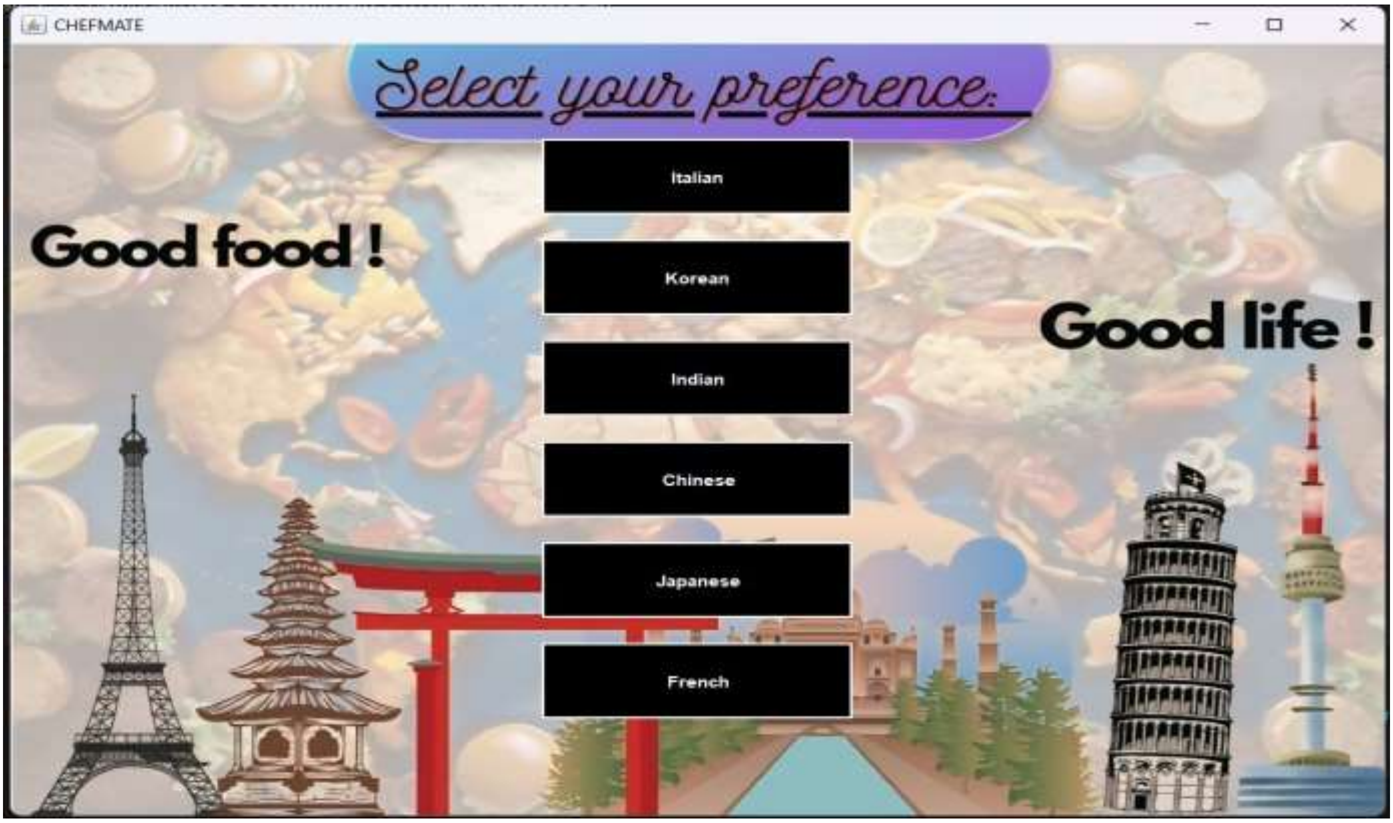
```
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

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_AS_NEEDE
D);
        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);
```

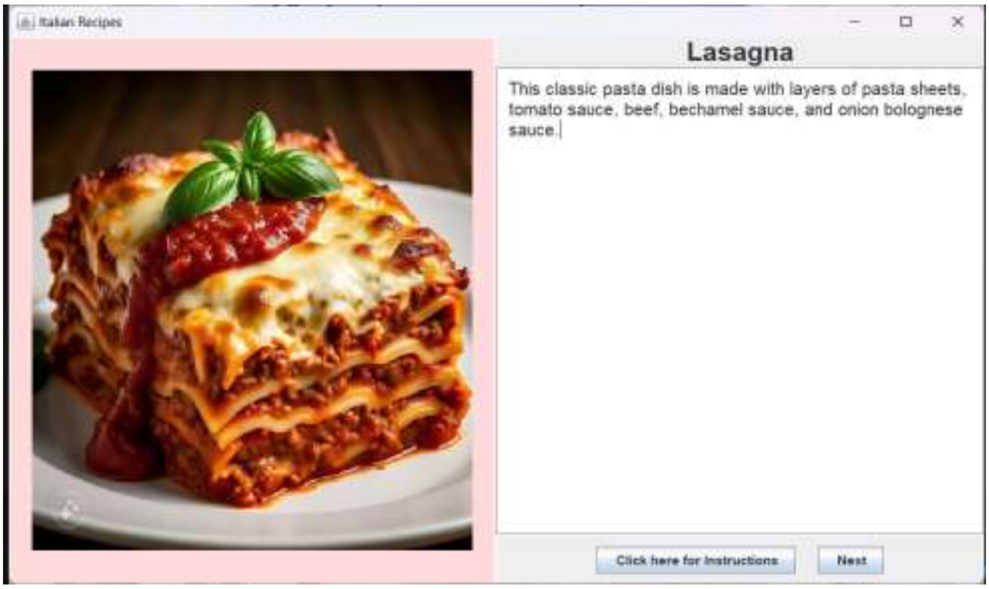
```
            instructionArea.setEditable(false);
            instructionArea.setFont(new Font("Arial", Font.PLAIN, 14));
            instructionArea.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

            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());
        }
    }

}
```

SNAPSHOTS

4.1 HOME PAGE

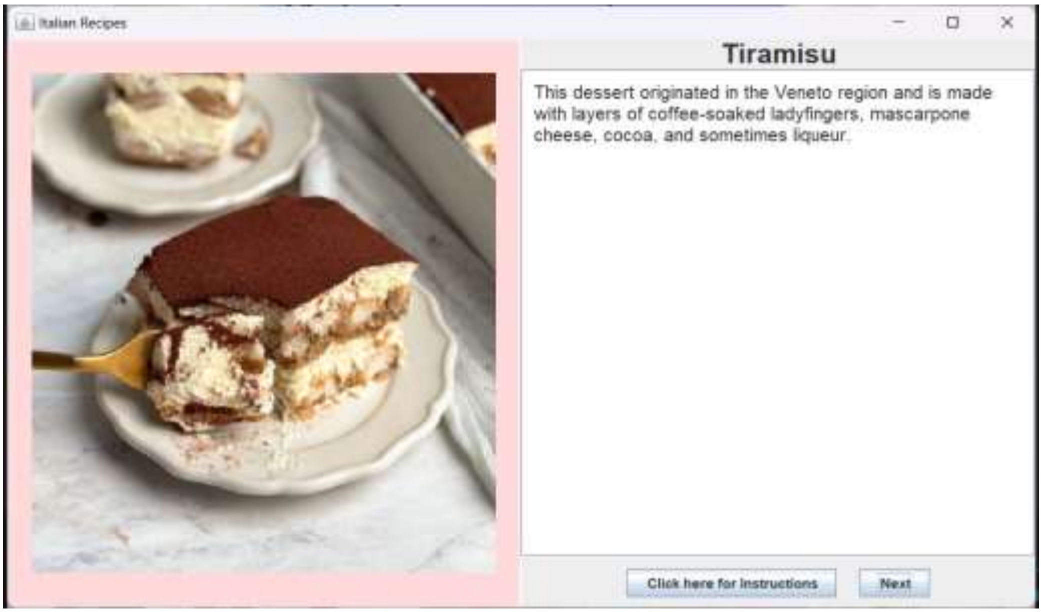


4.2 ITALIAN FOOD PAGE



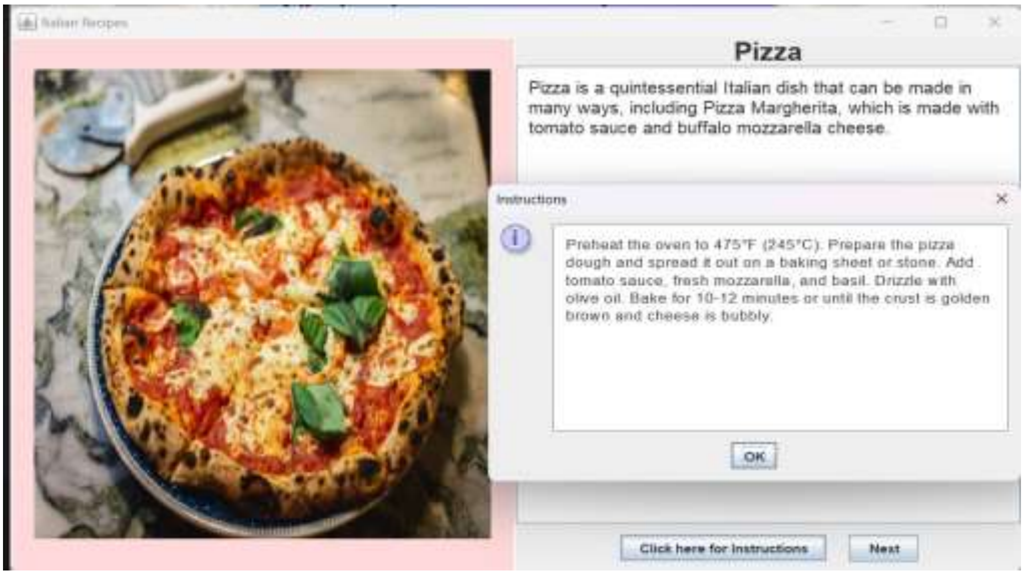
Lasagna

This classic pasta dish is made with layers of pasta sheets, tomato sauce, beef, bechamel sauce, and onion bolognese sauce.



Tiramisu

This dessert originated in the Veneto region and is made with layers of coffee-soaked ladyfingers, mascarpone cheese, cocoa, and sometimes liqueur.

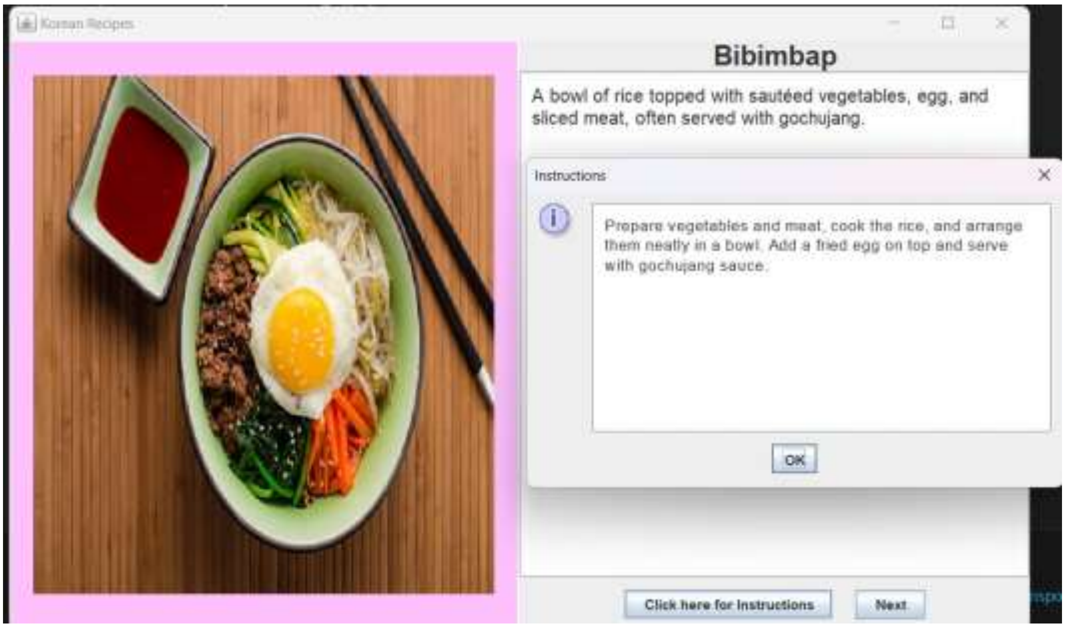


Pizza

Pizza is a quintessential Italian dish that can be made in many ways, including Pizza Margherita, which is made with tomato sauce and buffalo mozzarella cheese.

Preheat the oven to 475°F (245°C). Prepare the pizza dough and spread it out on a baking sheet or stone. Add tomato sauce, fresh mozzarella, and basil. Drizzle with olive oil. Bake for 10-12 minutes or until the crust is golden brown and cheese is bubbly.

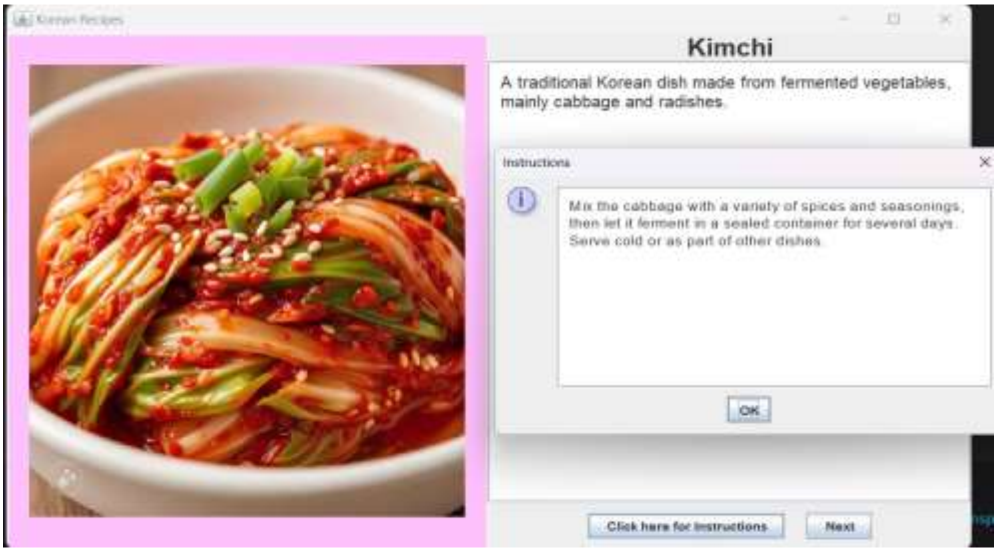
4.3 KOREAN FOOD PAGE



Bibimbap

A bowl of rice topped with sautéed vegetables, egg, and sliced meat, often served with gochujang.

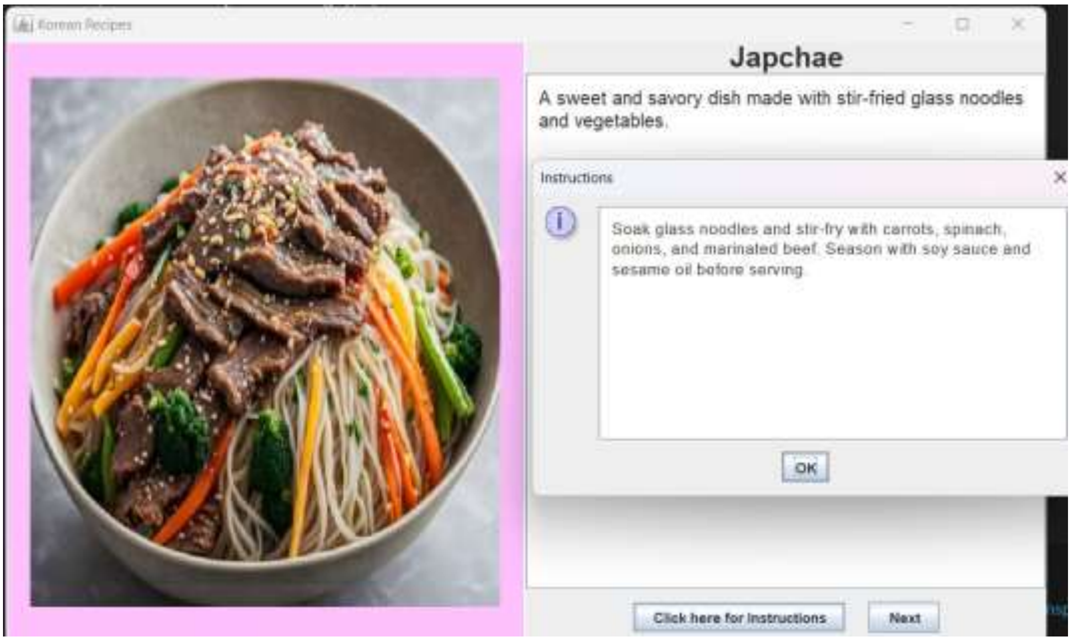
Prepare vegetables and meat, cook the rice, and arrange them neatly in a bowl. Add a fried egg on top and serve with gochujang sauce.



Kimchi

A traditional Korean dish made from fermented vegetables, mainly cabbage and radishes.

Mix the cabbage with a variety of spices and seasonings, then let it ferment in a sealed container for several days. Serve cold or as part of other dishes.

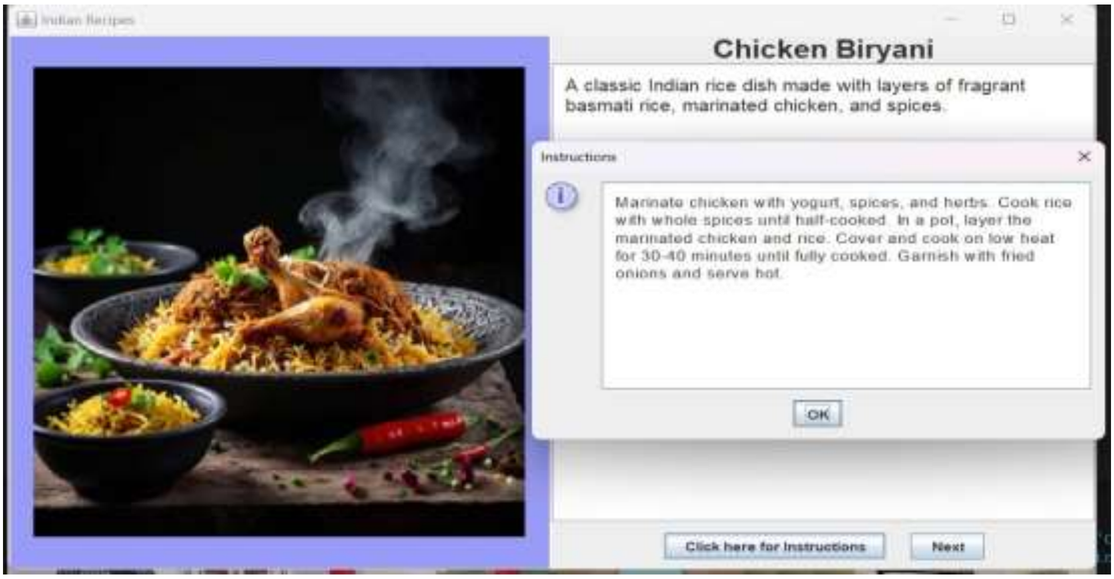


Japchae

A sweet and savory dish made with stir-fried glass noodles and vegetables.

Soak glass noodles and stir-fry with carrots, spinach, onions, and marinated beef. Season with soy sauce and sesame oil before serving.

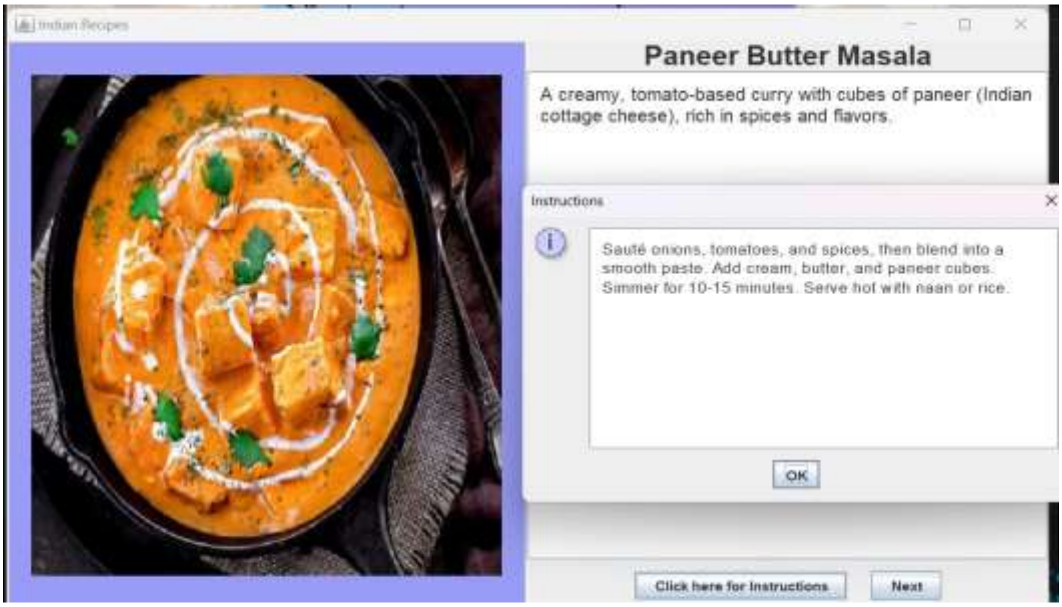
4.4 INDIAN FOOD PAGE



Chicken Biryani

A classic Indian rice dish made with layers of fragrant basmati rice, marinated chicken, and spices.

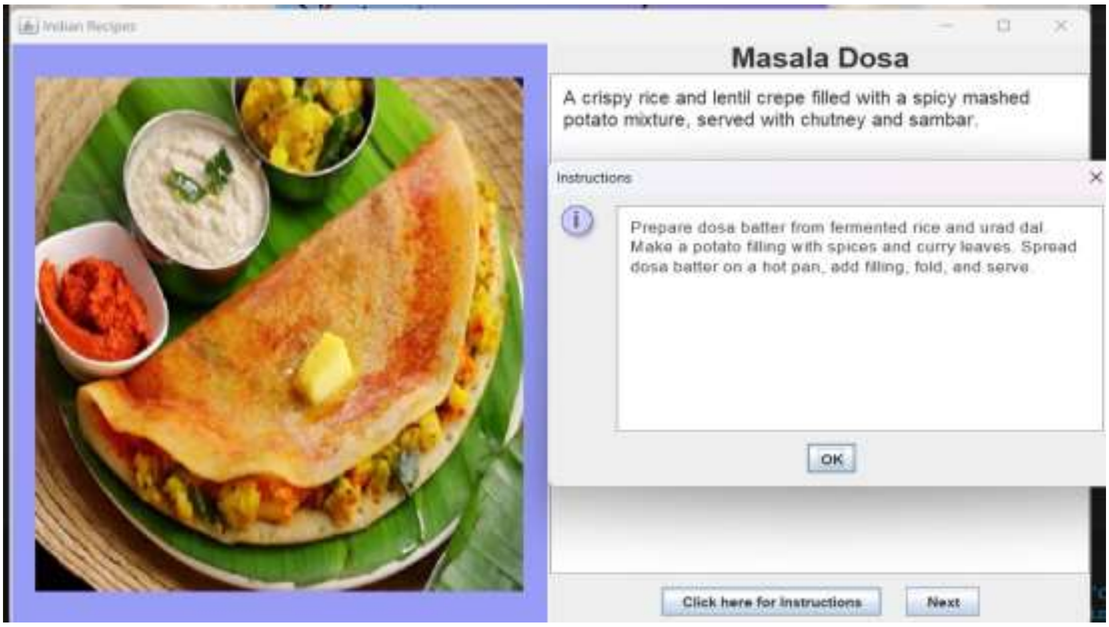
Marinate chicken with yogurt, spices, and herbs. Cook rice with whole spices until half-cooked. In a pot, layer the marinated chicken and rice. Cover and cook on low heat for 30-40 minutes until fully cooked. Garnish with fried onions and serve hot.



Paneer Butter Masala

A creamy, tomato-based curry with cubes of paneer (Indian cottage cheese), rich in spices and flavors.

Sauté onions, tomatoes, and spices, then blend into a smooth paste. Add cream, butter, and paneer cubes. Simmer for 10-15 minutes. Serve hot with naan or rice.

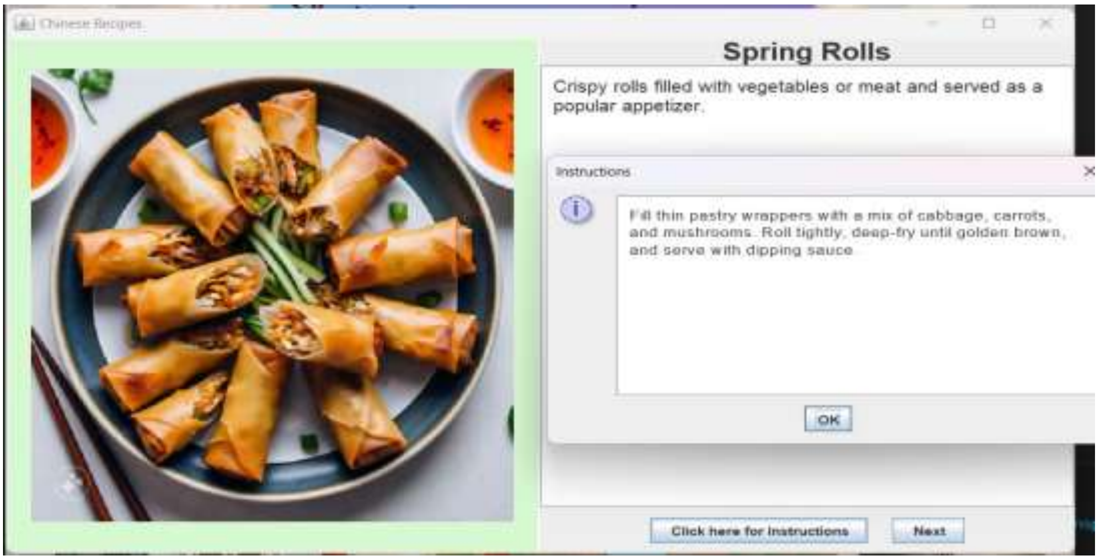


Masala Dosa

A crispy rice and lentil crepe filled with a spicy mashed potato mixture, served with chutney and sambar.

Prepare dosa batter from fermented rice and urad dal. Make a potato filling with spices and curry leaves. Spread dosa batter on a hot pan, add filling, fold, and serve.

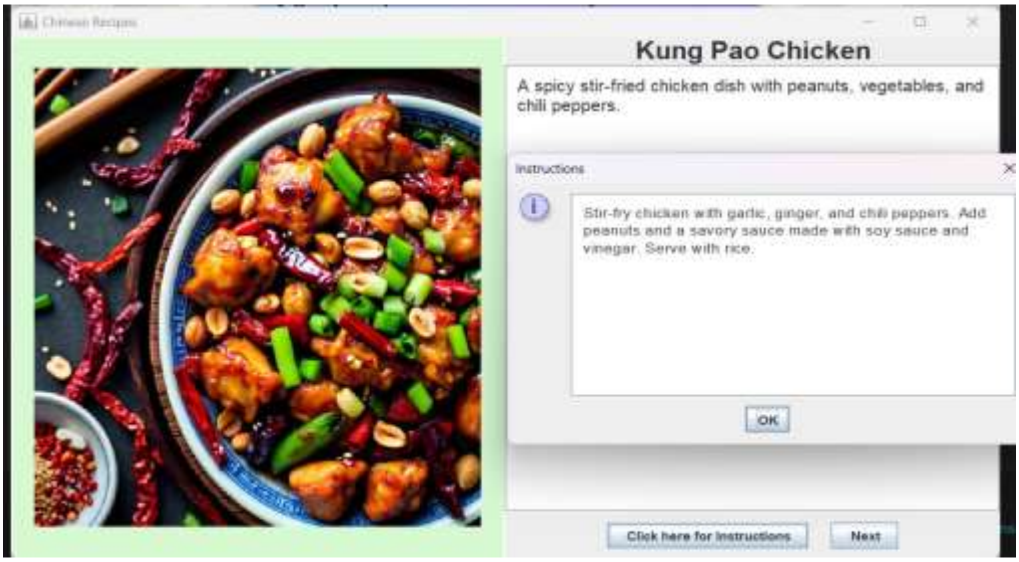
4.5 CHINESE FOOD PAGE



Spring Rolls

Crispy rolls filled with vegetables or meat and served as a popular appetizer.

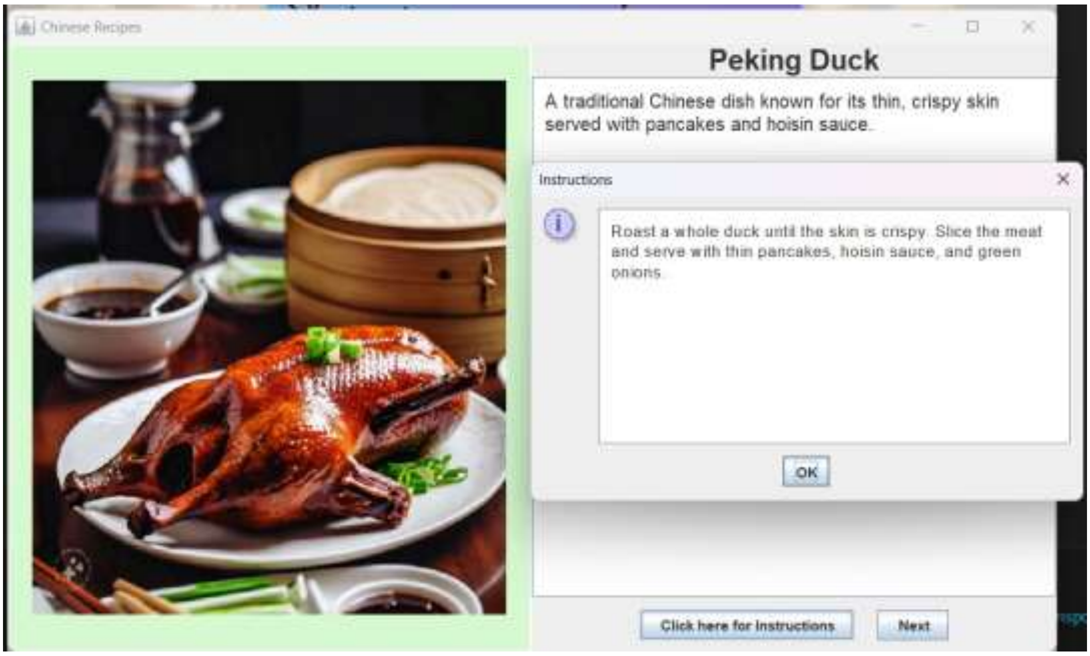
Fill thin pastry wrappers with a mix of cabbage, carrots, and mushrooms. Roll tightly, deep-fry until golden brown, and serve with dipping sauce.



Kung Pao Chicken

A spicy stir-fried chicken dish with peanuts, vegetables, and chili peppers.

Stir-fry chicken with garlic, ginger, and chili peppers. Add peanuts and a savory sauce made with soy sauce and vinegar. Serve with rice.

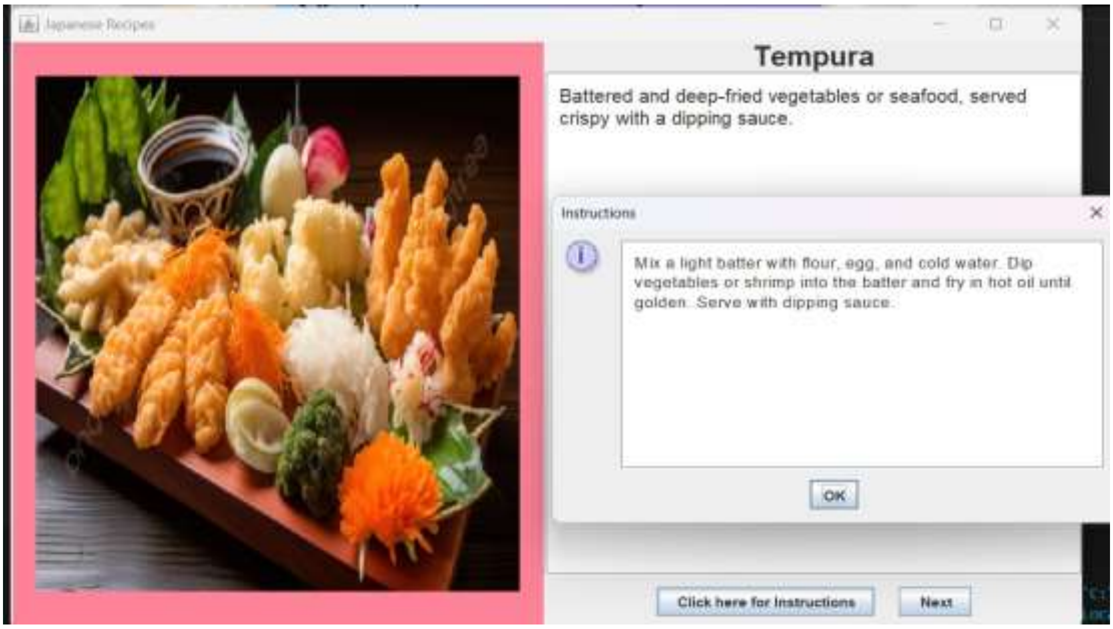


Peking Duck

A traditional Chinese dish known for its thin, crispy skin served with pancakes and hoisin sauce.

Roast a whole duck until the skin is crispy. Slice the meat and serve with thin pancakes, hoisin sauce, and green onions.

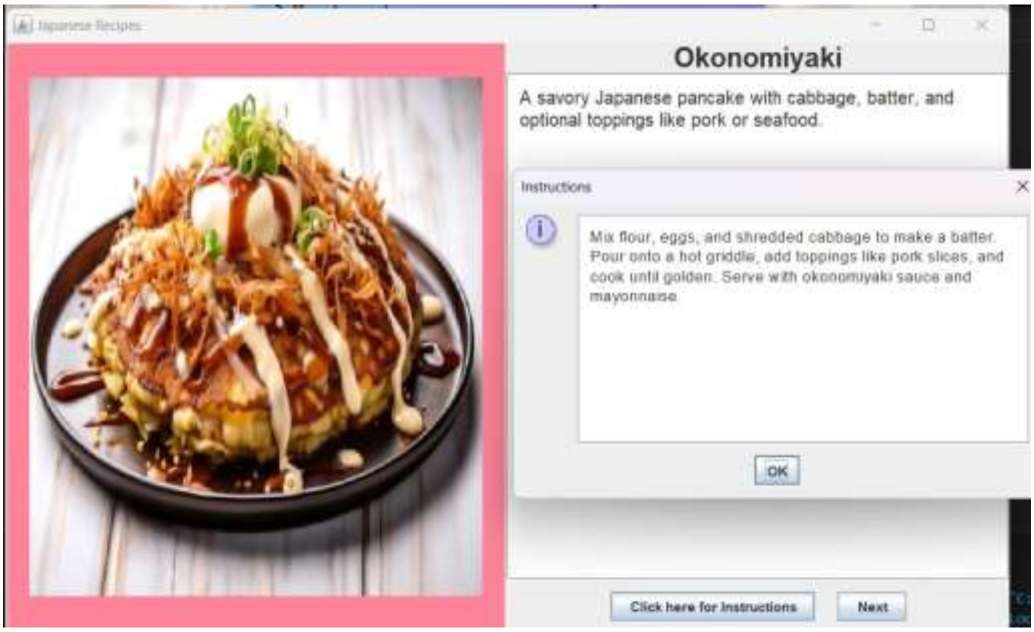
4.6 JAPANESE FOOD PAGE



Tempura

Battered and deep-fried vegetables or seafood, served crispy with a dipping sauce.

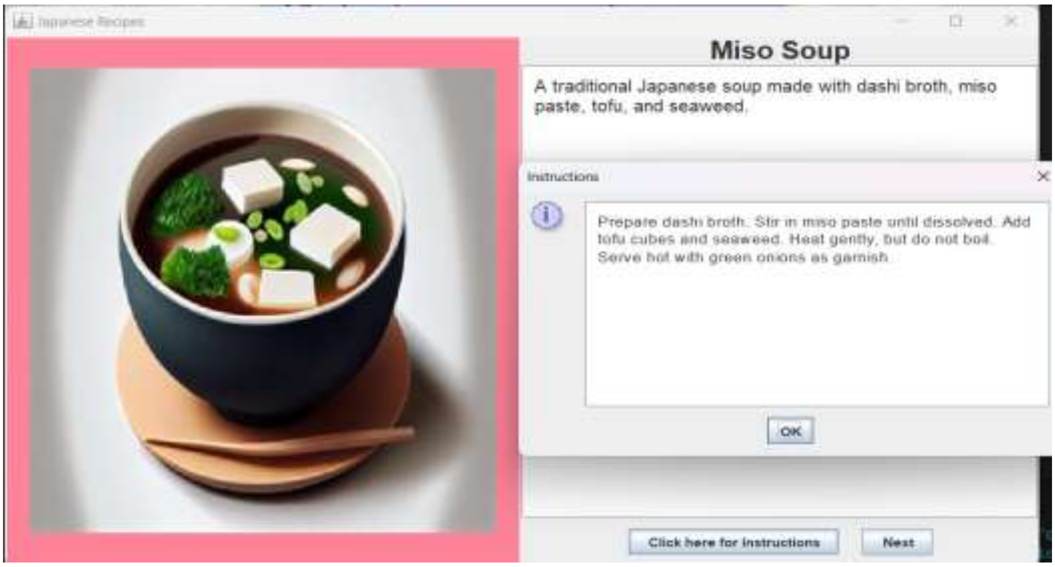
Mix a light batter with flour, egg, and cold water. Dip vegetables or shrimp into the batter and fry in hot oil until golden. Serve with dipping sauce.



Okonomiyaki

A savory Japanese pancake with cabbage, batter, and optional toppings like pork or seafood.

Mix flour, eggs, and shredded cabbage to make a batter. Pour onto a hot griddle, add toppings like pork slices, and cook until golden. Serve with okonomiyaki sauce and mayonnaise.

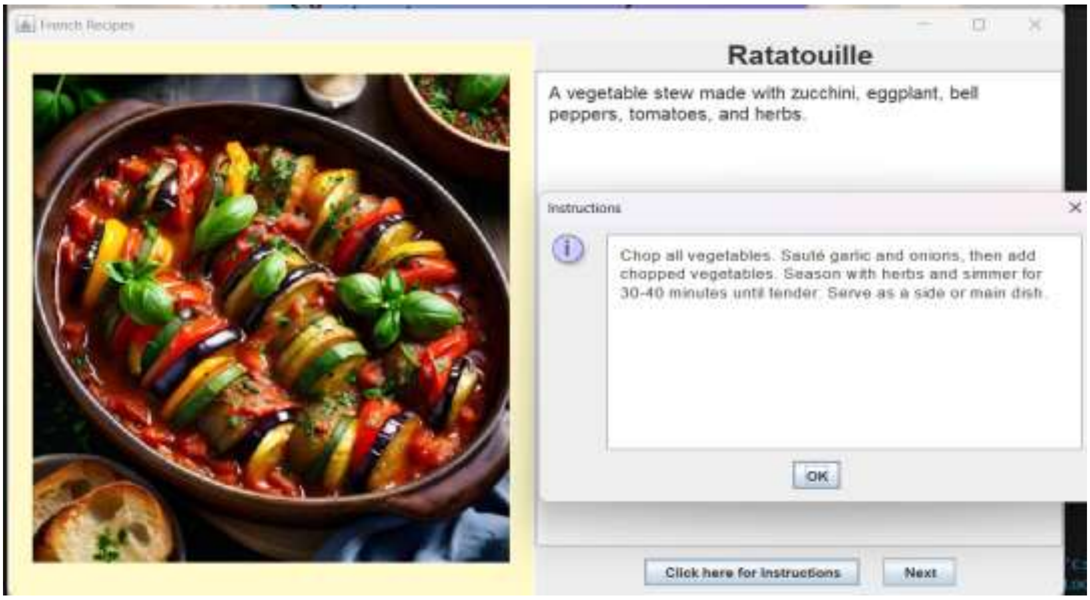


Miso Soup

A traditional Japanese soup made with dashi broth, miso paste, tofu, and seaweed.

Prepare dashi broth. Stir in miso paste until dissolved. Add tofu cubes and seaweed. Heat gently, but do not boil. Serve hot with green onions as garnish.

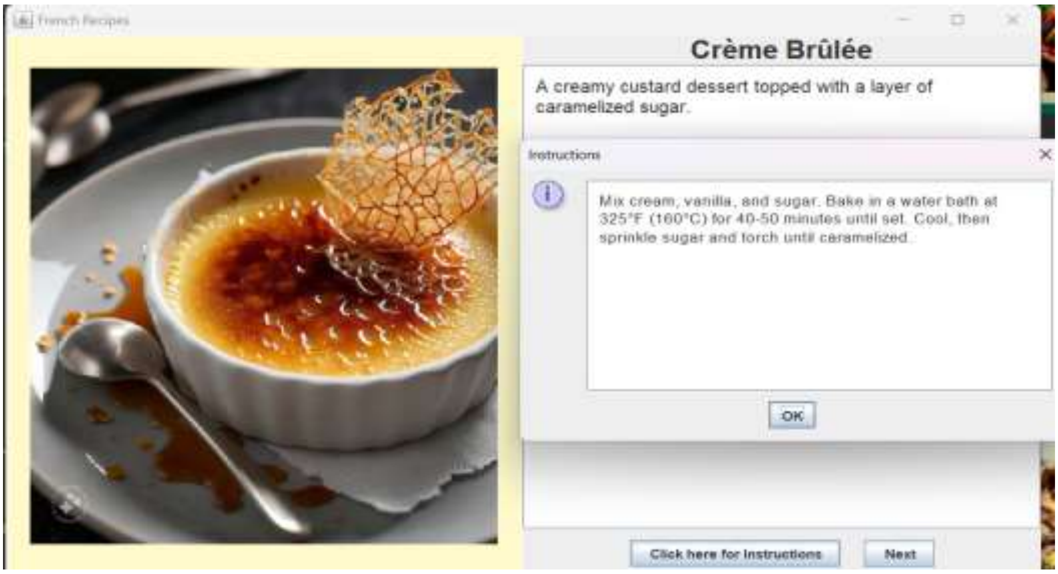
4.7 FRANCHISE FOOD PAGE



Ratatouille

A vegetable stew made with zucchini, eggplant, bell peppers, tomatoes, and herbs.

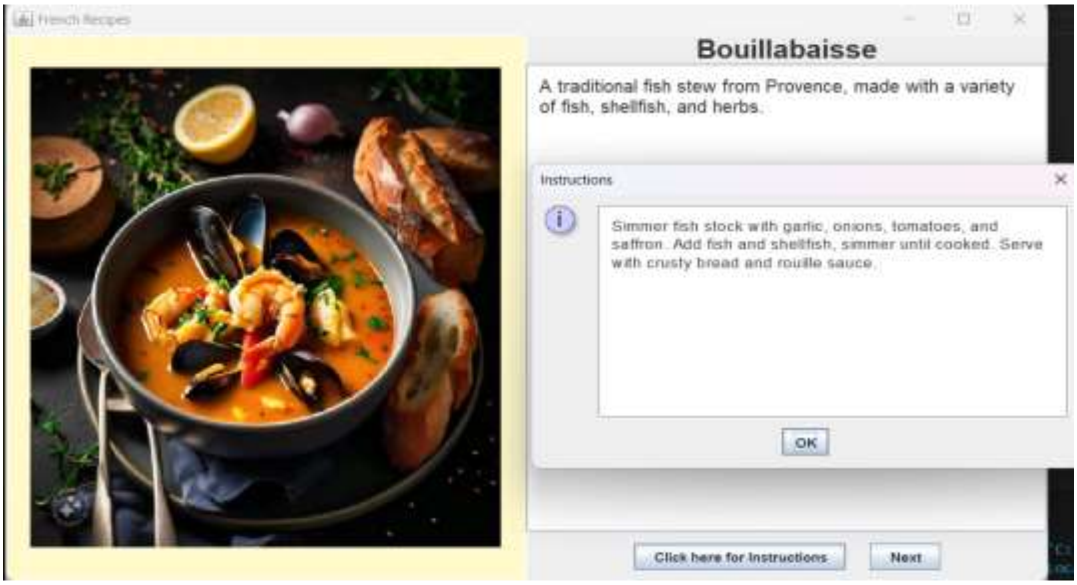
Chop all vegetables. Sauté garlic and onions, then add chopped vegetables. Season with herbs and simmer for 30-40 minutes until tender. Serve as a side or main dish.



Crème Brûlée

A creamy custard dessert topped with a layer of caramelized sugar.

Mix cream, vanilla, and sugar. Bake in a water bath at 325°F (160°C) for 40-50 minutes until set. Cool, then sprinkle sugar and torch until caramelized.



Bouillabaisse

A traditional fish stew from Provence, made with a variety of fish, shellfish, and herbs.

Simmer fish stock with garlic, onions, tomatoes, and saffron. Add fish and shellfish, simmer until cooked. Serve with crusty bread and rouille sauce.

CONCLUSION

A recipe recommendation app revolutionizes meal planning by providing personalized suggestions based on user preferences, dietary restrictions, and available ingredients. By analyzing users' tastes, dietary needs, and even health goals, it helps them discover diverse, nutritious, and easy-to-make recipes. The app simplifies the cooking process, making it easier for individuals to try new dishes without the stress of meal planning. With features like ingredient-based searches, nutritional information, and step-by-step instructions, it encourages healthier eating habits, reduces food waste, and promotes culinary creativity. Ultimately, the app enhances the overall cooking experience, making it enjoyable and accessible for everyone.

REFERENCES

1. **https://www.javatpoint.com/java-tutorial**
2. https://www.wikipedia.org/
3. https://www.w3schools.com/sql/
4. SQL | Codecademy