

SOUTHERN LUZON STATE UNIVERSITY
COLLEGE OF ENGINEERING
COMPUTER DEPARTMENT

2024

CPE18 - SOFTWARE DESIGN & DEVELOPMENT

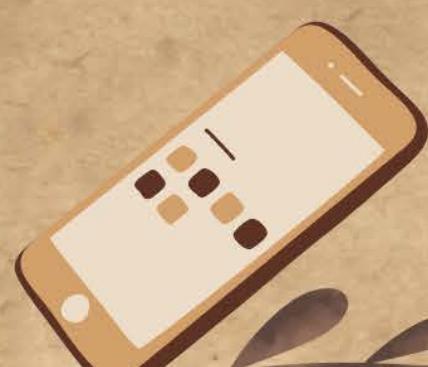
CHEF CACHE'S



JADE WESLEY J.
DEVILLES

BSCPE III - GF

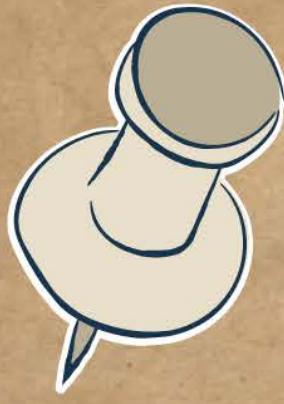
JUNE 10 , 2024



ver
State
a. V.
D C
lixer, ple
urt in M.
ampaign fi
charges
Manafort's un
rectly on Mr M
to Russian inter
6 election or on v
mp has sought to
stigation
was the first test o
counsel's ability to p
case in a federal cou
nd intense criticism from
ident and his allies that
ury is a biased and unjusti
ch hunt. And the outcome
stantial political implicat
only in denying Mr Trump
immunition for his cam
discredit Mr Mueller
e and during the trial, Mr
with whom he had a 1 M

NE,
ANALYS

engulfed his W
collapsed



INTRODUCTION OF CACHE'S

In nunc porta
etus et netus et
ut. Sed ut sit et
netus. Huiusmodi ut
hanc modumque lo-
nsequitur probitatus; et etiam
inclusus. In huiusmodi ve-
ritate regnatur iste sacerdos.
Opis egestas. Proi-
nassa eget pede-
naeos. Donec
eget, consequa-
et magnis dis

s. Etiam at
pus, sed
amplificare
et deinde quod
objisto se-
ion cursu
na. Manu
e posuerit.
Cras dapibus
nunc viverra
ne at tempu
et malesuada
dolores est qui
eget neque at



Welcome to Chef's Cache, the ultimate culinary companion designed to revolutionize your kitchen experience. Chef's Cache goes beyond being a mere recipe app; it is a comprehensive tool tailored to enhance your culinary skills, inspire creativity, and streamline your cooking process. Whether you're taking your first steps into the culinary world or you're an experienced chef seeking new inspirations, Chef's Cache is here to support and elevate your culinary journey.

Why Choose Chef's Cache?

Chef's Cache is built on the core philosophy of making cooking accessible, enjoyable, and convenient for everyone. The app combines an extensive recipe catalog with innovative features such as personalized meal planning, intuitive menu creation, and smart shopping lists. These tools are designed to empower users to take full control of their culinary adventures, ensuring that each meal is not only delicious but also aligns with your dietary preferences and nutritional goals.

Key Features of Chef's Cache:

- Extensive Recipe Catalog:** Discover a world of flavors with our vast collection of recipes. From quick and easy weeknight dinners to gourmet meals for special occasions, our database is continuously updated to provide endless inspiration.
- Personalized Meal Planning:** Tailor your meal plans to suit your lifestyle and dietary needs. Whether you're vegan, gluten-free, or following a specific diet, Chef's Cache helps you create balanced and nutritious meal plans that cater to your preferences.
- Intuitive Menu Creation:** Easily design menus for any occasion with our user-friendly interface. Plan family meals, dinner parties, or holiday feasts with confidence, knowing that every detail is covered.
- Smart Shopping Lists:** Simplify your grocery shopping with automatically generated shopping lists based on your selected recipes and meal plans. Save time and reduce food waste by buying only what you need.
- Cooking Tutorials and Tips:** Enhance your skills with step-by-step cooking tutorials and expert tips. From basic techniques to advanced culinary tricks, Chef's Cache helps you build confidence in the kitchen.
- Dietary Management:** Manage dietary restrictions and allergies with ease. Filter recipes and meal plans to exclude ingredients you want to avoid, ensuring safe and enjoyable meals for everyone.
- Time-Saving:** Overcome common cooking challenges such as lack of time and meal boredom. With Chef's Cache, you can quickly find recipes that fit your schedule and discover new flavors to keep your meals exciting.

Transform Your Culinary Experience:

Chef's Cache is more than just an app, it's your trusted partner in the kitchen. By combining advanced technology with a passion for cooking, we aim to make the culinary process fun, efficient, and deeply rewarding. Whether you're cooking for yourself, your family, or entertaining guests, Chef's Cache provides the tools and inspiration you need to create delicious meals with confidence.

Join the Chef's Cache community today and embark on a culinary journey like no other. Discover new flavors, master new techniques, and enjoy the art of cooking with Chef's Cache by your side.

TABLE OF CONTENTS

I.	SDLC (ACTIVITY 1)	3 – 4
II.	Homepage Design/Menu & Logo	5
III.	Software Date Modeling	6 – 7
IV.	Software Layout Design	8 – 15
V.	Summary of Codes	16 – 120
VI.	Screenshot of Actual Usage	121 –



Software Development Life Cycle

Concept Creation

Managing recipes efficiently is a common challenge for individuals seeking to organize their culinary endeavors. This abstract presents a problem statement for the development of a recipe organizer. Existing methods of managing recipes, such as physical recipe books or scattered digital files, are often cumbersome, disorganized, and not optimized for modern lifestyle.

The design of an offline recipe organizer software application entails a robust local database system for storing recipes, intuitive user interface for easy navigation and management, and offline synchronization capabilities for updating recipes across devices. Additionally, backup and restore functionality ensures data integrity and security.

Title

"CHEF'S CACHE: YOUR OFFLINE RECIPE COMPANION"

Chef Cache is the ultimate offline recipe organizer designed to be your culinary companion, whether you're a seasoned chef, starting out in the kitchen, or starting a business anytime and anywhere.

Intended Audience

Chef Cache is very useful to any individual for learning, practicing or even creating a food stall with offline access anywhere we are.

Features

OFFLINE MODE /ACCESS

This is particularly useful for cooking in areas with limited or no internet access, such as rural areas or while travelling.

MEAL PLANNING

Provide tools for users to plan their meals by scheduling recipes for specific days, creating shopping list based on planned meals and dietary preferences.

RECIPE COLLECTION

User can create and manage their recipe collection and it includes new recipes manually or recipe database.

CATEGORIZATION & TAGGING

Organize recipes into categories for easy searching and filtering.

CUSTOMIZATION

Allows users to customize the app interface like themes.

SOURCE:

Smith, J. & Johnson, A. A Comprehensive Review of Recipe Management System: Challenges and Opportunities Journal of Culinary Technology.

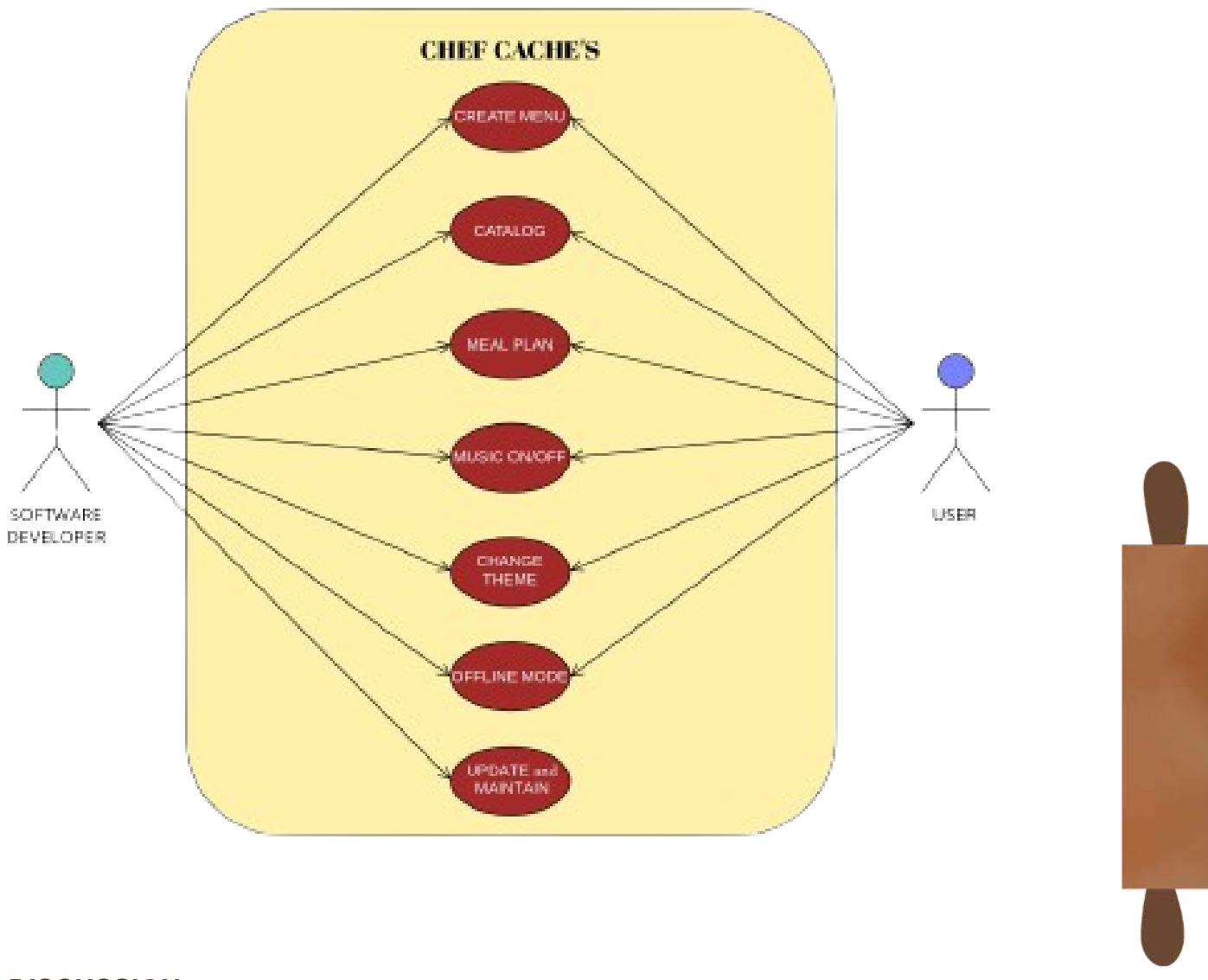


Homepage Design/Menu & Logo



SOFTWARE DATA MODELING

USE CASE DIAGRAM

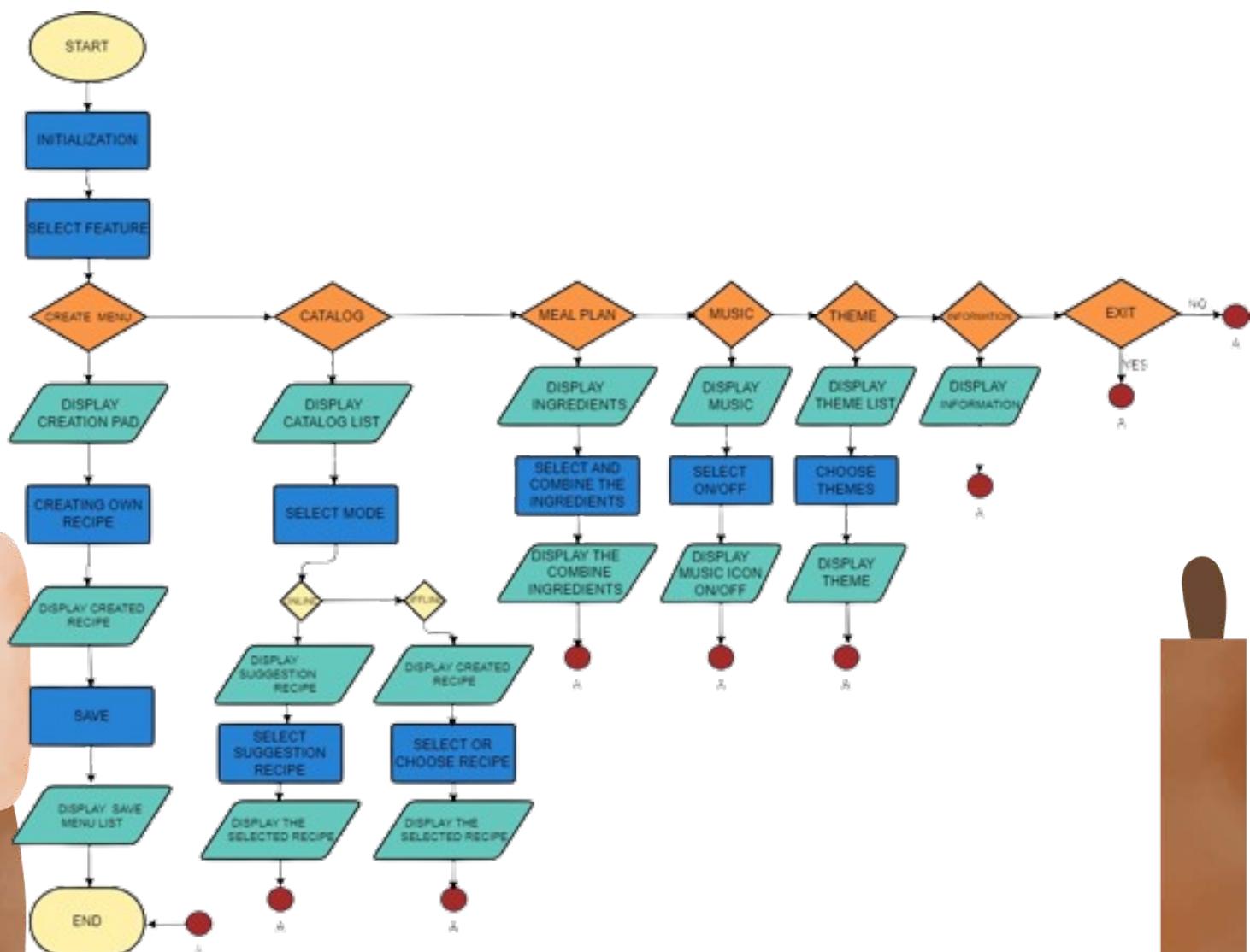


DISCUSSION

This use case diagram for "Chef Cache's" shows the interactions between two types of users, the software developer and the regular user, with the system's main functions. The system has several features: "Create Menu," "Catalog," "Meal Plan," "Music On/Off," "Change Theme," "Offline Mode," and "Update and Maintain." The software developer is responsible for implementing and maintaining these features, while the user interacts with them for various purposes. For example, users can create a meal plan, browse the catalog, and toggle music on or off. They can also change the theme of the interface and use the app in offline mode. Overall, this diagram helps us understand how different users interact with the key functionalities of the "Chef Cache's" system.



FLOWCHART



DISCUSSION

This flowchart for "Chef Cache's" outlines the steps users follow to interact with various features of the application. It starts with the initialization process, where users select a feature to use. Users can choose from options like "Create Menu," "Catalog," "Meal Plan," "Music," "Theme," and "Information."

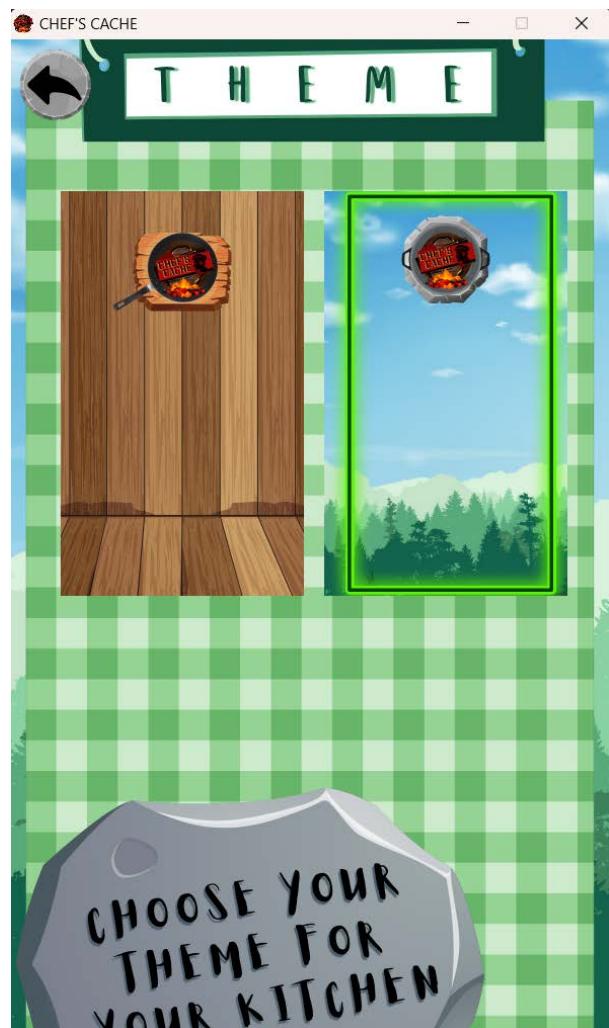
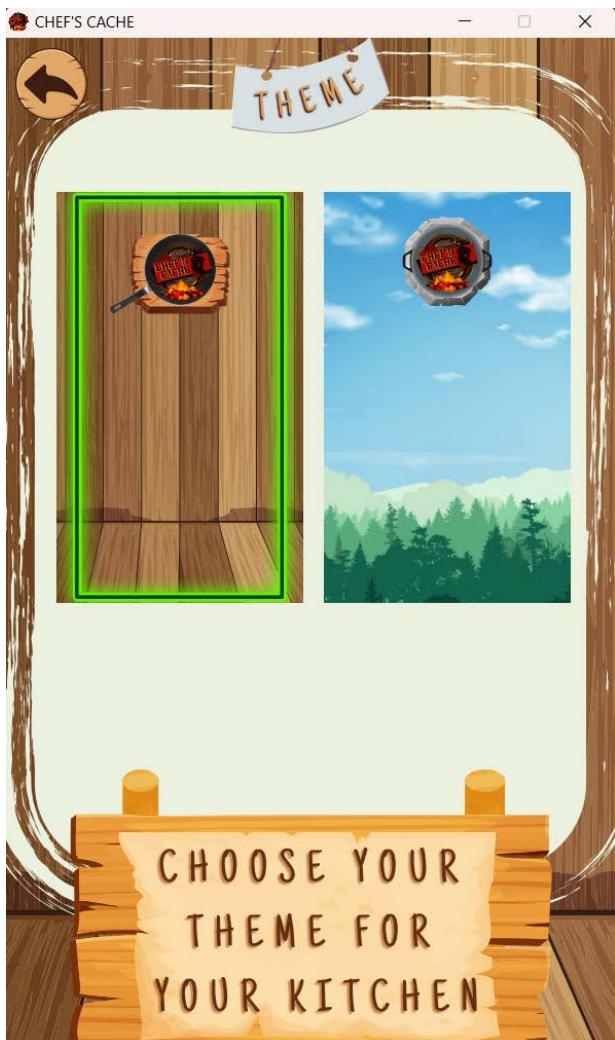
For the "Create Menu" feature, users are presented with a creation pad to create their own recipes, which they can then save and view in a menu list. The "Catalog" feature allows users to view a list of recipes, where they can select between online and offline modes. In online mode, users get recipe suggestions, while offline mode lets them select or choose a recipe they already have.

Lastly, the "Meal Plan" feature helps users display and combine ingredients for meal planning. The "Music" feature allows users to turn the music on or off, while the "Theme" feature lets them choose different themes for the application. The "Information" feature provides additional details about the application. Finally, users can choose to exit the application, which is marked as the end of the process. This flowchart effectively visualizes the user journey through different functionalities of the "Chef Cache's" system.

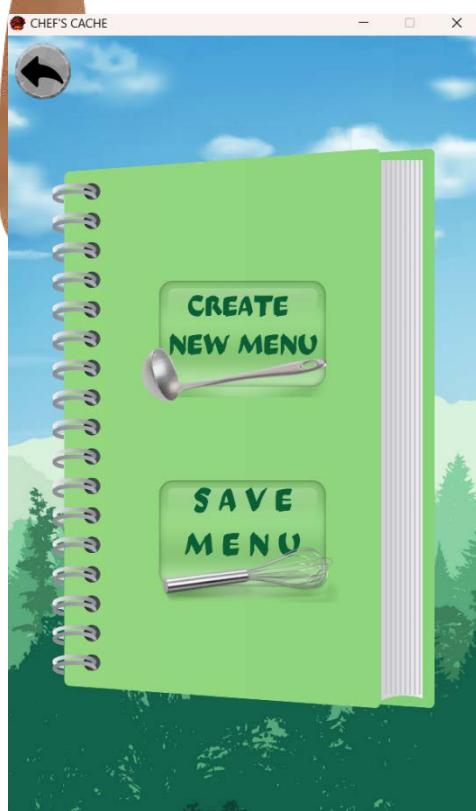


SOFTWARE LAYOUT DESIGN

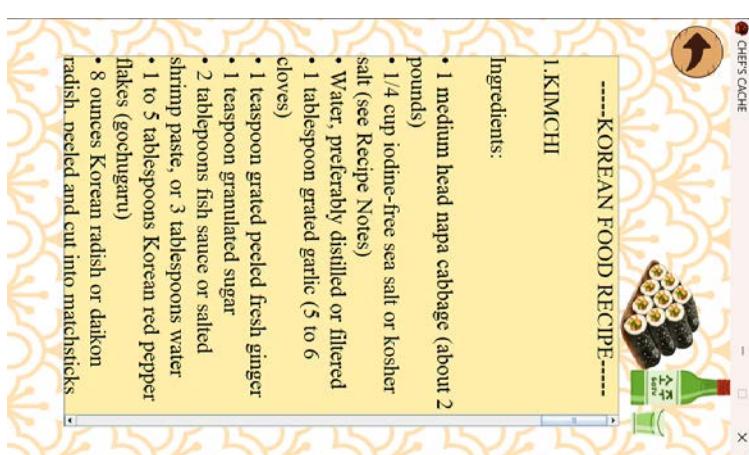
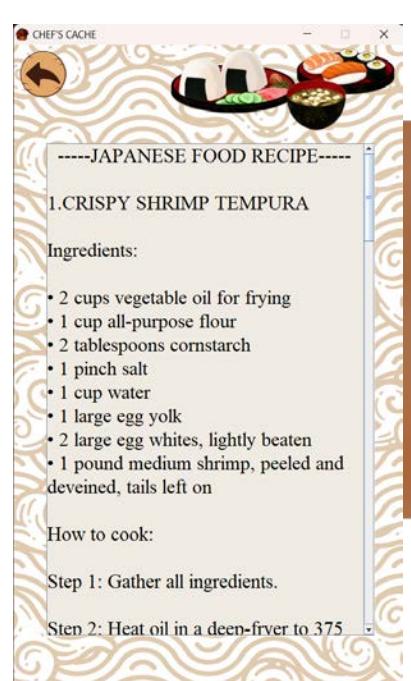
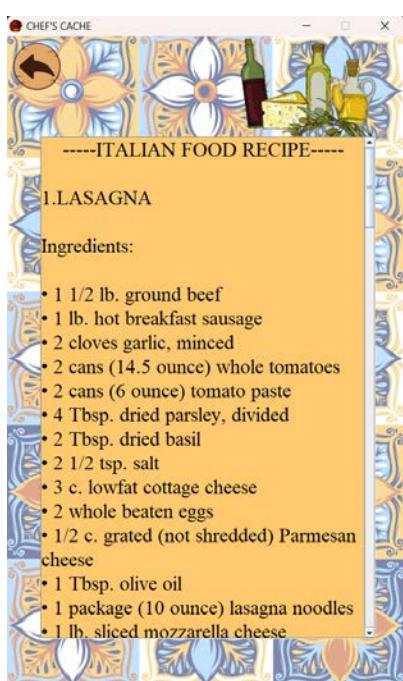
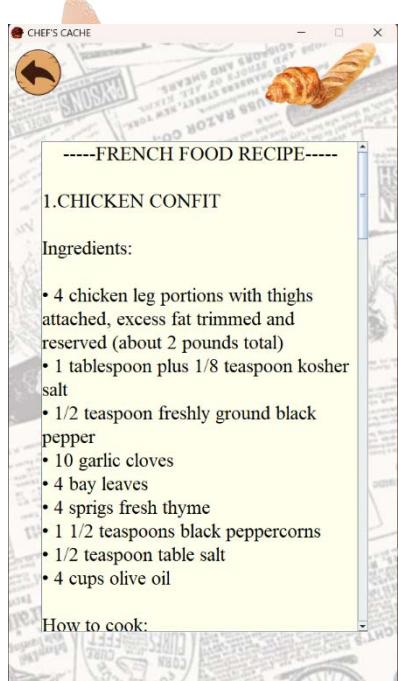
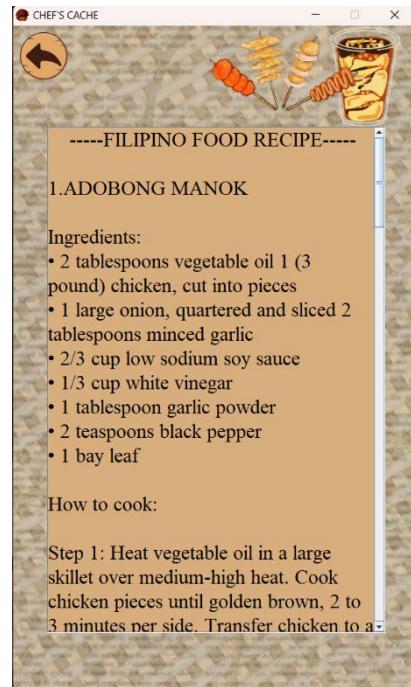
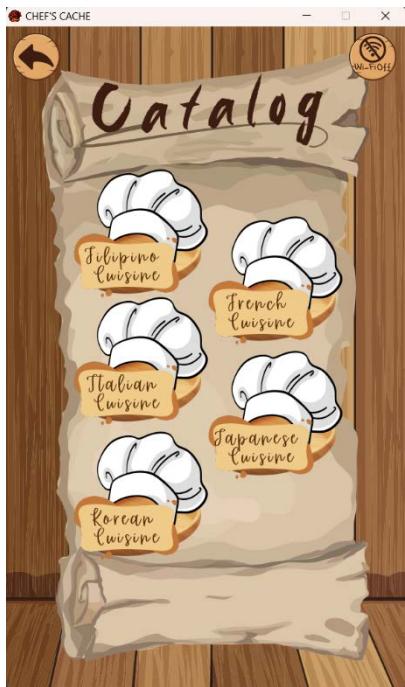
FEATURE 1 (THEME)



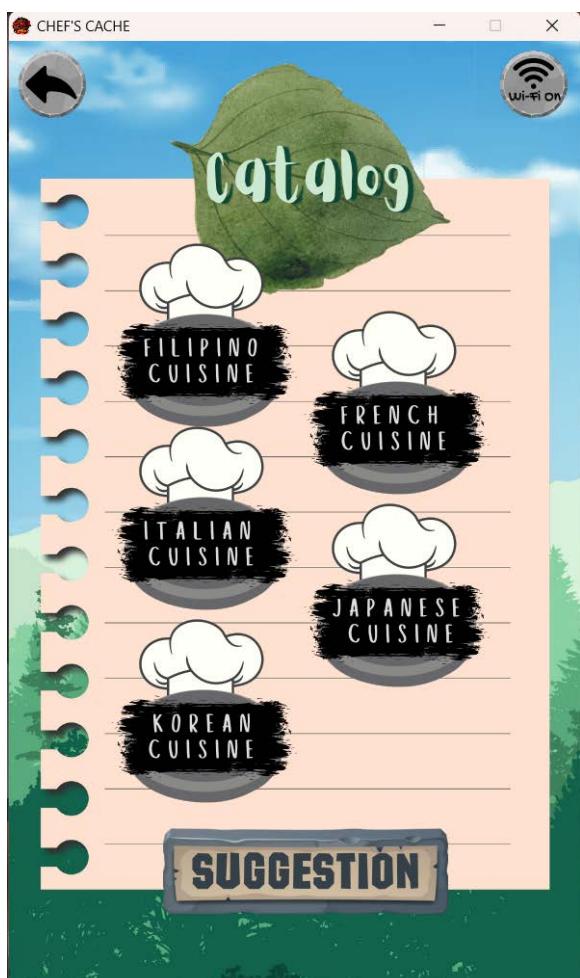
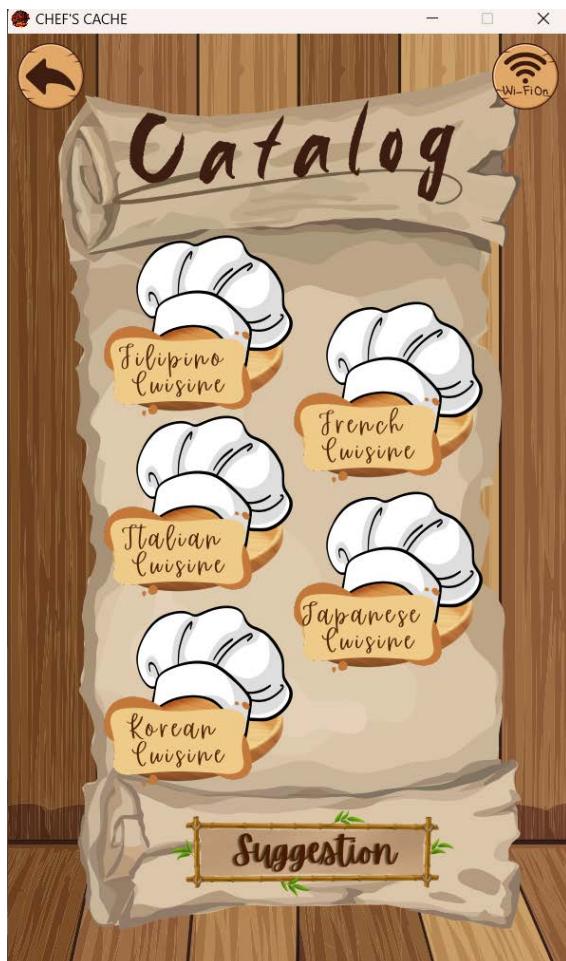
FEATURE 2 (CREATE MENU)



FEATURE 3 (CATALOG)



FEATURE 4 (WI-FI ON/OFF)



Home - Panlasang Pinoy | French Recipes Homepage | Discover Delicious and Simple | Japanese Food | Easy Korean Recipes - My Korean Recipes | Authentic Chinese Recipes

panlasangpinoy.com

Google Meet | Classes | CANVA | Smallpdf.com - A Free Online PDF Editor | Welcome To Colaboo...

About | Cooking Schools | Contact | [f](#) [i](#) [p](#) [y](#)

Your Top Source Of Filipino Recipes



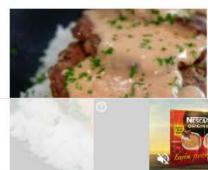
New? Start Here All Recipes Course ▾ Ingredient ▾ Type ▾ Search...

Browse Over 1500+ Filipino Recipes

With over 5 million YouTube subscribers, Panlasang Pinoy is the internet's top source of tried & true, kitchen-tested Filipino recipes.

[EXPLORE RECIPES >](#)

NEW!



BROWSE ALL



...

...

...



French Recipes Homepage - Pardon Your French | Discover Delicious and Simple | Japanese Food | Easy Korean Recipes - My Korean Recipes | Authentic Chinese Recipes

pardonyourfrench.com

Google Meet | Classes | CANVA | Smallpdf.com - A Free Online PDF Editor | Welcome To Colaboo...

New? Start Here.

Pardon your French

HOME ALL RECIPES COOKBOOK ABOUT SEE ALL RECIPES

Rustic French recipes made easy

New Recipes

SEE ALL RECIPES

French-style Cherry Cake (Gâteau aux cerises)

June 6, 2024 | 0 comments

Accounting software for small businesses

90% off for 6 months

Buy now

Eggplant Caviar Dip (Caviar d'Aubergine)

May 15, 2024 | 8 comments

Discover Delicious and Simple | Japanese Food | Easy Korean Recipes - My Korean Recipes | Authentic Chinese Recipes

italianfoodforever.com

Google Meet | Classes | CANVA | Smallpdf.com - A Free Online PDF Editor | Welcome To Colaboo...

ABOUT CONTACT

f i p Search ...



CHI MANGIA BENE, MANGIA ITALIANO

Browse Recipes Kitchen Tips ▾ Italy ❤️ ▾ Mangia Bene

Recent Posts

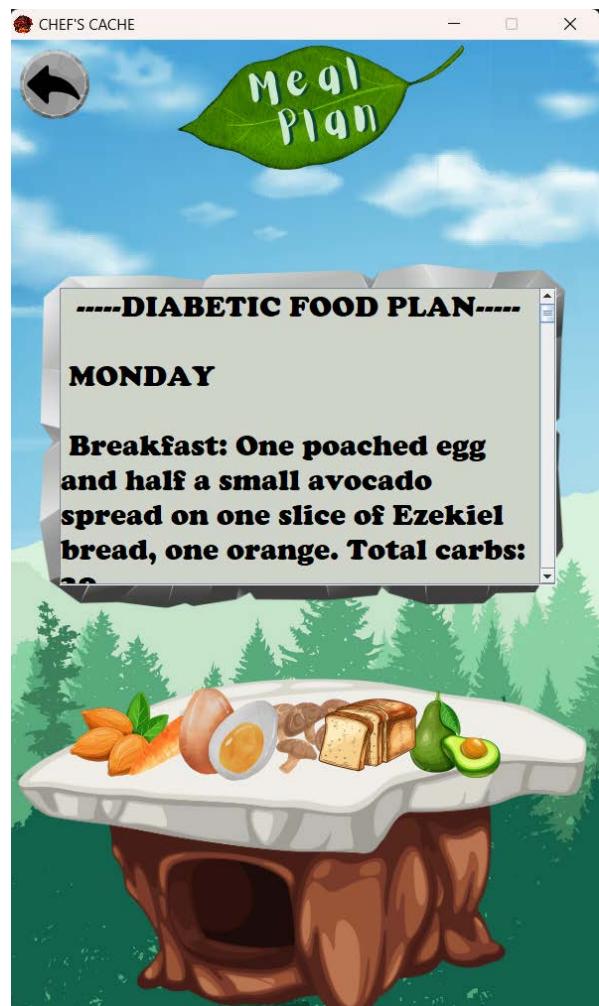
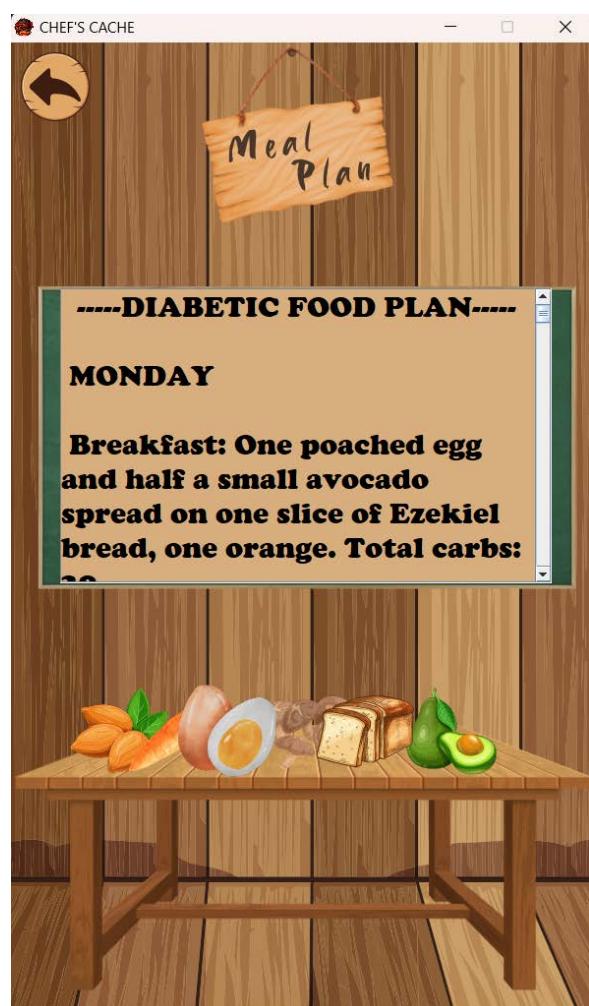
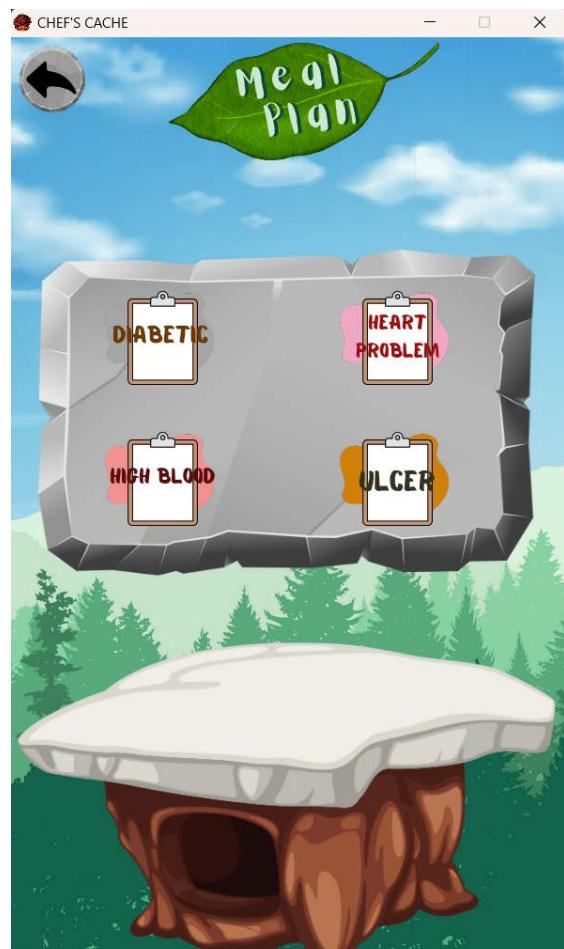
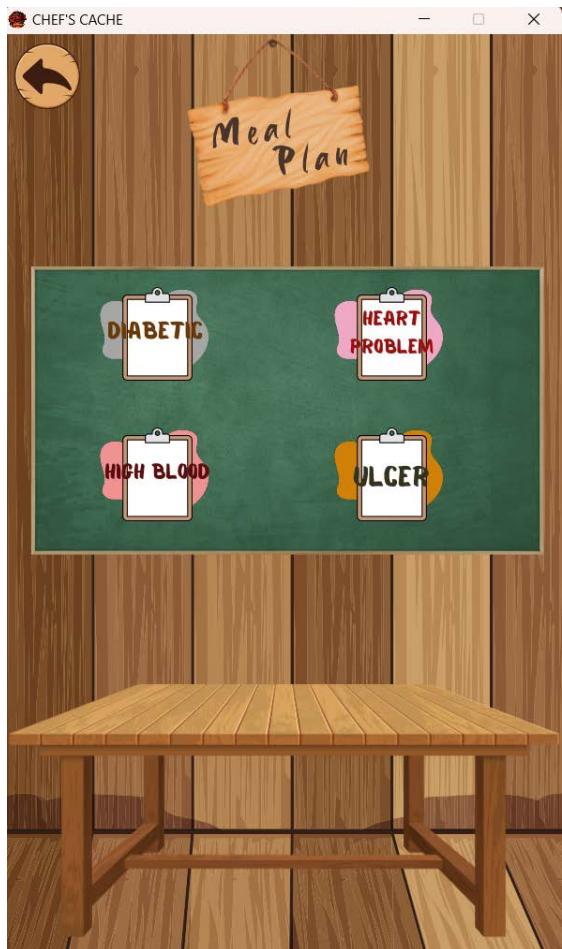


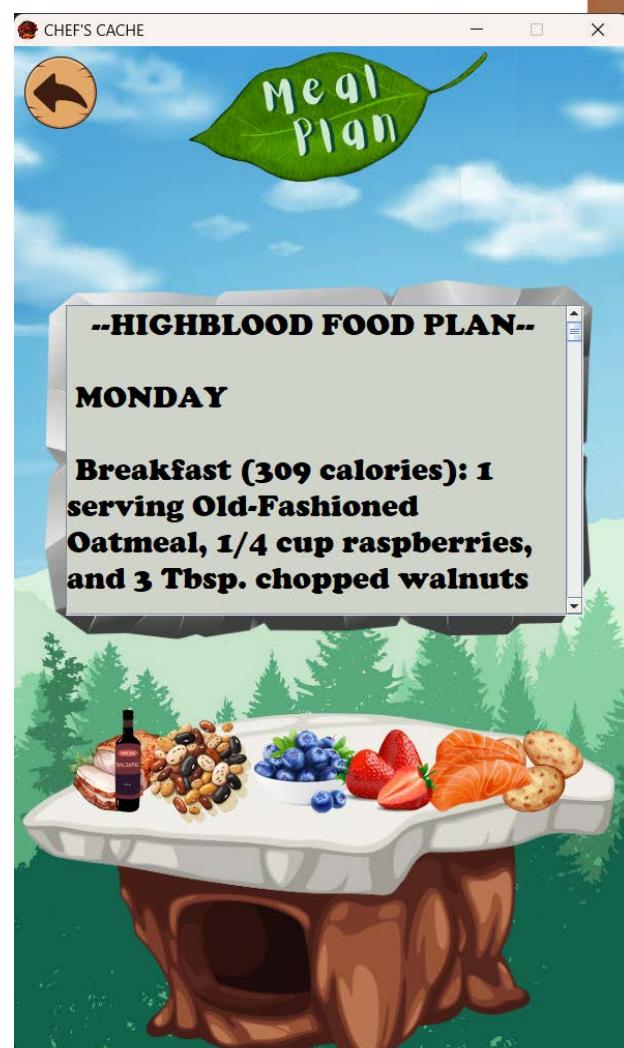
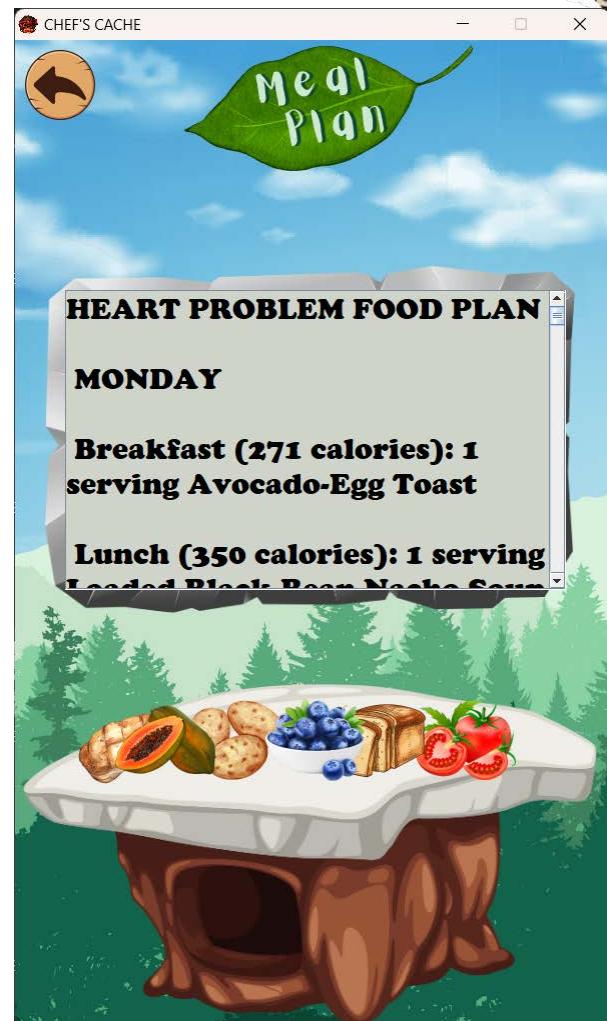
The screenshot shows a web browser window with the URL mykoreankitchen.com. The page features a header with the site's logo and navigation links for ABOUT, RECIPES, EBOOKS, K-DRAMA, SHOP, and CONTACT. A search bar is at the top right. Below the header, there's a portrait of a woman with the text "ANNYEONG & WELCOME!". A bio describes the author as Sue, the author/cook/photographer behind the site. A "GET THE RECIPE" button is visible next to a photo of the dish.

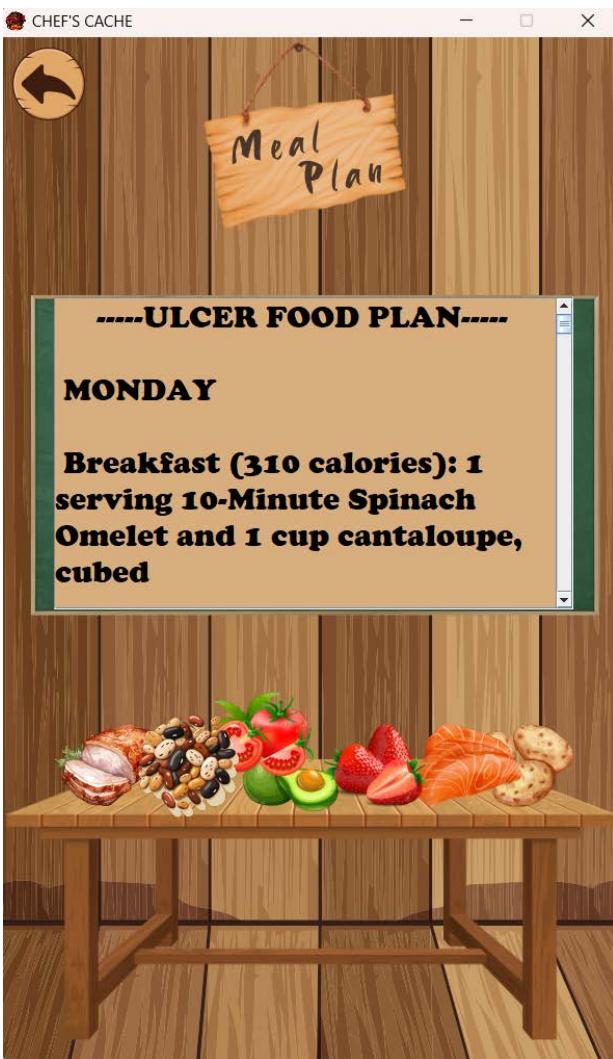
The screenshot shows a web browser window with the URL [japan-guide.com](http://japan-guide.com/e/e620.html). The page has a header with the site's logo and navigation links for Destinations, Interests, Plan, Forum, Videos, and Sign In. Below the header, there's a "Food and Drink" menu with links to Popular Dishes, How to dine out, Types of restaurants, and Table Manners. A large section titled "Japanese Food" features a collage of various Japanese dishes like sashimi,握壽司, and bento boxes.

The screenshot shows a web browser window with the URL chinasichuanfood.com. The page has a header with the site's logo and navigation links for NEWEST, RECIPE INDEX, SICHUAN, BY TYPE, Subscribe, About, and social media links. Below the header, there are four image thumbnails: Mapo Tofu Recipe - Miso Po Tofu, CHINESE PANCAKES - RICE AND FLOUR - TRENDING, Easy Spring Roll Wrappers, and Chinese Eggplants with Minced Pork. A banner for the HUAWEI Pura70 Series is visible at the bottom.

FEATURE 5 (MEAL PLAN)







SUMMARY OF CODES

```
package SD_CC;

import javax.swing.*;
import java.awt.*;
import javax.sound.sampled.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.IOException;
import java.io.InputStream;
import java.net.URL;
import javax.swing.text.BadLocationException;
import javax.swing.text.PlainDocument;
import java.awt.event.*;
import java.io.*;
import java.nio.file.*;
import java.net.URI;

public class MAINPAGE {
    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                if (args.length > 0 && args[0].equalsIgnoreCase("B")) {
                    new HomepageB();
                } else {
                    new Homepage();
                }
            }
        });
    }
}

@SuppressWarnings("serial")
class Homepage extends JFrame implements ActionListener {
    private static Clip clip;
    private static boolean isMusicPlaying = false; // Start with music off
    private static JFrame frame; // Variable for the frame
    private JToggleButton musicToggle; // Variable for music to toggle on and off
    private JButton CM, CTL, MP, THEME, INFO, EXIT; // Variable for every button

    public Homepage() {
        frame = new JFrame();
        frame.setContentPane(new JLabel(new ImageIcon(getClass().getResource("/Main.gif"))));
        Image logo =
            Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frame.setIconImage(logo);
        frame.setTitle("CHEF'S CACHE");

        CM = new JButton();
        CM.setIcon(new ImageIcon(getClass().getResource("/newcm.png")));
        CM.setBounds(190, 480, 240, 160);
```

```

CM.setOpaque(false);
CM.setContentAreaFilled(false);
CM.setBorderPainted(false);
ImageIcon cm1 = new ImageIcon();
CM.setPressedIcon(cm1);
CM.addActionListener(this);
CM.setName("cm");

CTL = new JButton();
CTL.setIcon(new ImageIcon(getClass().getResource("/newctl.png")));
CTL.setBounds(190, 590, 240, 160);
CTL.setOpaque(false);
CTL.setContentAreaFilled(false);
CTL.setBorderPainted(false);
ImageIcon ctl1 = new ImageIcon();
CTL.setPressedIcon(ctl1);
CTL.addActionListener(this);
CTL.setName("ctl");

MP = new JButton();
MP.setIcon(new ImageIcon(getClass().getResource("/newmp.png")));
MP.setBounds(190, 690, 240, 160);
MP.setOpaque(false);
MP.setContentAreaFilled(false);
MP.setBorderPainted(false);
ImageIcon mp1 = new ImageIcon();
MP.setPressedIcon(mp1);
MP.addActionListener(this);
MP.setName("mp");

// where music being toggle
musicToggle = new JToggleButton();
musicToggle.setBounds(510, 20, 70, 70);
setToggleButtonProperties(musicToggle, new
ImageIcon(getClass().getResource("/musicoff.png")),
new ImageIcon(getClass().getResource("/musicon.png")));
musicToggle.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        toggleMusic();
    }
});

INFO = new JButton();
INFO.setIcon(new ImageIcon(getClass().getResource("/SMbutton.png")));
INFO.setOpaque(false);
INFO.setContentAreaFilled(false);
INFO.setBounds(10, 30, 70, 70);
INFO.setBorderPainted(false);
ImageIcon infoA = new ImageIcon();
INFO.setPressedIcon(infoA);
INFO.addActionListener(this);
INFO.setName("info");

THEME = new JButton();
THEME.setIcon(new ImageIcon(getClass().getResource("/themebutton.png")));
THEME.setOpaque(false);
THEME.setContentAreaFilled(false);

```

```

THEME.setBounds(510, 100, 70, 70);
THEME.setBorderPainted(false);
ImageIcon themeA = new ImageIcon();
THEME.setPressedIcon(themeA);
THEME.addActionListener(this);
THEME.setName("theme");

EXIT = new JButton();
EXIT.setIcon(new ImageIcon(getClass().getResource("/newex.png")));
EXIT.setBounds(190, 790, 240, 160);
EXIT.setOpaque(false);
EXIT.setContentAreaFilled(false);
EXIT.setBorderPainted(false);
ImageIcon exit1 = new ImageIcon();
EXIT.setPressedIcon(exit1);
EXIT.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

frame.add(CM);
frame.add(CTL);
frame.add(MP);
frame.add(INFO);
frame.add(THEME);
frame.add(EXIT);
frame.add(musicToggle);

frame.setBounds(660, 0, 600, 1020);
frame.setResizable(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null);
frame.setVisible(true);

// where music stop all throughout the system
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        stopMusic();
    }
});

System.out.println("Homepage constructor called");
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("cm")) {
        new CMMain();
        frame.dispose();
    } else if (choices.equals("ctl")) {
        new CTL();
        frame.dispose();
    } else if (choices.equals("mp")) {
        new MP();
        frame.dispose();
    }
}

```



```
    } else if (choices.equals("theme")) {
        new THEME();
        frame.dispose();
    } else if (choices.equals("info")) {
        new INFO();
        frame.dispose();
    }
}

// music button information
@SuppressWarnings("unused")
private static JButton createButton(String imagePath, String pressedImagePath,
int x, int y) {
    JButton button = new JButton();
    button.setIcon(new ImageIcon(imagePath));
    button.setBounds(x, y, 70, 70);
    button.setOpaque(false);
    button.setContentAreaFilled(false);
    button.setBorderPainted(false);
    button.setPressedIcon(new ImageIcon(pressedImagePath));
    return button;
}

// toggle part of the music where it on and off
private static void toggleMusic() {
    if (isMusicPlaying) {
        stopMusic();
        isMusicPlaying = false;
    } else {
        playMusic("/sounds1.wav");
        isMusicPlaying = true;
    }
}

// Inserting music that being used and condition to play the music
private static void playMusic(String filePath) {
    try {
        URL audioFile = MAINPAGE.class.getResource(filePath);
        AudioInputStream audioInputStream =
        AudioSystem.getAudioInputStream(audioFile);
        clip = AudioSystem.getClip();
        clip.open(audioInputStream);
        clip.loop(Clip.LOOP_CONTINUOUSLY); // Loop the music
        clip.start();
        isMusicPlaying = true; // Music is now playing
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
e) {
        e.printStackTrace();
    }
}

// condition to stop the music
static void stopMusic() {
    if (clip != null) {
        clip.stop();
        clip.close();
        isMusicPlaying = false; // Music is now stopped
    }
}
```



```

        }

    }

// Method to set properties for JToggleButton
private void setToggleButtonProperties(JToggleButton button, Icon icon, Icon
selectedIcon) {
    button.setIcon(icon);
    button.setSelectedIcon(selectedIcon);
    button.setCursor(new Cursor(Cursor.HAND_CURSOR));
    button.setFocusPainted(false);
    button.setBorder(null);
    button.setContentAreaFilled(false);
}

}

@SuppressWarnings("serial")
public class HomepageB extends JFrame implements ActionListener {
    private static Clip clip;
    private static boolean isMusicPlaying = false;
    private JToggleButton musicToggle;
    private static JButton CM, CTL, MP, THEME, INFO, EXIT;

    public HomepageB() {
        setTitle("CHEF'S CACHE");
        setContentPane(new JLabel(loadImage("/Main2.gif")));
        setIconImage(loadImage("/CCLOGO.png").getImage());

        CM = new JButton(loadImage("/cm2.png"));
        CM.setBounds(190, 480, 240, 160);
        setButtonProperties(CM, "cm");

        CTL = new JButton(loadImage("/ctl2.png"));
        CTL.setBounds(190, 590, 240, 160);
        setButtonProperties(CTL, "ctl");

        MP = new JButton(loadImage("/mp2.png"));
        MP.setBounds(190, 690, 240, 160);
        setButtonProperties(MP, "mp");

        INFO = new JButton(loadImage("/SMBUTTON2.png"));
        INFO.setBounds(10, 30, 70, 70);
        setButtonProperties(INFO, "info");

        THEME = new JButton(loadImage("/themebutton2.png"));
        THEME.setBounds(510, 100, 70, 70);
        setButtonProperties(THEME, "theme");

        EXIT = new JButton(loadImage("/exit2.png"));
        EXIT.setBounds(190, 790, 240, 160);
        setButtonProperties(EXIT, null);
        EXIT.addActionListener(e -> System.exit(0));

        musicToggle = new JToggleButton();
        musicToggle.setBounds(510, 20, 70, 70);
        setToggleButtonProperties(musicToggle, loadImage("/musicoff2.png"),
loadImage("/musicon2.png"));
    }
}

```

```
musicToggle.addActionListener(e -> toggleMusic());
```

```
add(CM);  
add(CTL);  
add(MP);  
add(INFO);  
add(THEME);  
add(EXIT);  
add(musicToggle);
```

```
setBounds(660, 0, 600, 1020);  
setResizable(false);  
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
setLayout(null);  
setVisible(true);
```

```
addWindowListener(new WindowAdapter() {  
    @Override  
    public void windowClosing(WindowEvent e) {  
        stopMusic();  
    }  
});
```

```
System.out.println("Homepage constructor called");  
}
```

```
@Override  
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton) e.getSource();  
    String choices = choice.getName();  
    switch (choices) {  
        case "cm":  
            new CMMainB();  
            dispose();  
            break;  
        case "ctl":  
            new CTLB();  
            dispose();  
            break;  
        case "mp":  
            new MPB();  
            dispose();  
            break;  
        case "theme":  
            new THEMEB();  
            dispose();  
            break;  
        case "info":  
            new INFOB();  
            dispose();  
            break;  
        default:  
            break;  
    }  
}
```

```
private static void toggleMusic() {
```

```

if (isMusicPlaying) {
    stopMusic();
} else {
    playMusic("/sounds2.wav");
}
}

private static void playMusic(String filePath) {
    stopMusic();
    try {
        URL audioFile = MAINPAGE.class.getResource(filePath);
        AudioInputStream audioInputStream =
        AudioSystem.getAudioInputStream(audioFile);
        clip = AudioSystem.getClip();
        clip.open(audioInputStream);
        clip.loop(Clip.LOOP_CONTINUOUSLY); // Loop the music
        clip.start();
        isMusicPlaying = true; // Music is now playing
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
e) {
    e.printStackTrace();
}
}

static void stopMusic() {
    if (clip != null) {
        clip.stop();
        clip.close();
        isMusicPlaying = false;
    }
}

private void setButtonProperties(JButton button, String name) {
    button.setOpaque(false);
    button.setContentAreaFilled(false);
    button.setBorderPainted(false);
    button.setName(name);
    button.addActionListener(this);
    button.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

private void setToggleButtonProperties(JToggleButton button, Icon icon, Icon
selectedIcon) {
    button.setIcon(icon);
    button.setSelectedIcon(selectedIcon);
    button.setCursor(new Cursor(Cursor.HAND_CURSOR));
    button.setFocusPainted(false);
    button.setBorder(null);
    button.setContentAreaFilled(false);
}

private ImageIcon loadImage(String path) {
    java.net.URL imgURL = getClass().getResource(path);
    if (imgURL != null) {
        return new ImageIcon(imgURL);
    } else {
        System.err.println("Couldn't find file: " + path);
    }
}

```

```

        return null;
    }

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

class CM extends JFrame implements ActionListener {
    private JFrame framecm;
    private JButton BACK, SAVE;
    private JTextArea textArea;
    private JScrollPane scrollPane;

    public CM() {
        framecm = new JFrame();
        framecm.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CMPage.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framecm.setIconImage(logo);
        framecm.setTitle("CHEF'S CACHE");

        SAVE = new JButton();
        SAVE.setIcon(new ImageIcon(getClass().getResource("/SAVE.png")));
        SAVE.setBounds(170, 870, 240, 100);
        SAVE.setOpaque(false);
        SAVE.setContentAreaFilled(false);
        SAVE.setBorderPainted(false);
        SAVE.addActionListener(this);
        SAVE.setName("save");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10, 10, 70, 70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        textArea = new JTextArea();
        textArea.setText("_____TITLE_____ \n• ");
        textArea.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        textArea.setLineWrap(true); // Enable line wrap
        textArea.setBackground(new Color(255, 239, 213));
        textArea.setFont(new Font("Viner Hand ITC", Font.PLAIN, 30));
        textArea.setBounds(40, 199, 510, 670);

        // Add KeyListener to detect Enter key press
        textArea.addKeyListener(new KeyAdapter() {
            @Override
            public void keyPressed(KeyEvent e) {

```

```

if (e.getKeyCode() == KeyEvent.VK_ENTER) {
    e.consume(); // Consume the Enter key event
    int caretPosition = textArea.getCaretPosition();
    textArea.insert("\n• ", caretPosition);
}
});

// Create a JScrollPane and add the JTextArea to it
scrollPane = new JScrollPane(textArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);

// Set document to limit text length
((PlainDocument) textArea.getDocument()).setDocumentFilter(new
DocumentSizeFilter(300));

// Add the JScrollPane to the JFrame
scrollPane.setBounds(40, 199, 510, 670);
framecm.add(scrollPane);

framecm.add(SAVE);
framecm.add(BACK);

framecm.setBounds(660, 0, 600, 1020);
framecm.setResizable(false);
framecm.setLayout(null);
framecm.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) {
        new CMMain();
        framecm.dispose();
    } else if (choices.equals("save")) {
        // Prompt for file name
        String fileName = JOptionPane.showInputDialog(framecm, "Enter file name:",
        "Save File", JOptionPane.QUESTION_MESSAGE);

        if (fileName != null && !fileName.trim().isEmpty()) {
            String content = textArea.getText();
            try {
                Files.write(Paths.get(fileName + ".txt"), content.getBytes());
                JTextField textField = new JTextField("Your Menu is Saved as " + fileName
                + ".txt");
                textField.setEditable(false);
                textField.setOpaque(false);
                JOptionPane.showMessageDialog(framecm, textField, "Save
Confirmation", JOptionPane.INFORMATION_MESSAGE);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
        }
    }
}

```

```

        }
    }
}

// Document filter to limit text length
class DocumentSizeFilter extends javax.swing.text.DocumentFilter {
    int maxCharacters;

    public DocumentSizeFilter(int maxChars) {
        maxCharacters = maxChars;
    }

    public void insertString(javax.swing.text.DocumentFilter.FilterBypass fb, int offs,
String str, javax.swing.text.AttributeSet a) throws BadLocationException {
        if ((fb.getDocument().getLength() + str.length()) <= maxCharacters) {
            super.insertString(fb, offs, str, a);
        }
    }

    public void replace(javax.swing.text.DocumentFilter.FilterBypass fb, int offs, int
length, String str, javax.swing.text.AttributeSet a) throws BadLocationException {
        if ((fb.getDocument().getLength() + str.length() - length) <= maxCharacters) {
            super.replace(fb, offs, length, str, a);
        }
    }
}

class CM2 implements ActionListener {
    private JFrame framecmain2;
    private JButton BACK, LOAD, DELETE;
    private JTextArea textArea;
    private JScrollPane ScrollPane;
    private DefaultListModel<String> ListModel;
    private JList<String> FileList;
    private JPanel backgroundPanel;

    public CM2() {
        framecmain2 = new JFrame();
        framecmain2.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CMSave.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framecmain2.setIconImage(logo);
        framecmain2.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
    }
}

```

```

BACK.setName("back");

LOAD = new JButton();
LOAD.setIcon(new ImageIcon(getClass().getResource("/load.png")));
LOAD.setOpaque(false);
LOAD.setContentAreaFilled(false);
LOAD.setBounds(100, 5, 240, 100);
LOAD.setBorderPainted(false);
ImageIcon loadA = new ImageIcon();
LOAD.setPressedIcon(loadA);
LOAD.addActionListener(this);
LOAD.setName("load");

DELETE = new JButton();
DELETE.setIcon(new ImageIcon(getClass().getResource("/delete.png")));
DELETE.setOpaque(false);
DELETE.setContentAreaFilled(false);
DELETE.setBounds(350, 5, 240, 100);
DELETE.setBorderPainted(false);
ImageIcon deleteA = new ImageIcon();
DELETE.setPressedIcon(deleteA);
DELETE.addActionListener(this);
DELETE.setName("delete");

ListModel = new DefaultListModel<>();
FileList = new JList<>(ListModel);
FileList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
loadSavedFiles();

JScrollPane fileScrollPane = new JScrollPane(FileList);
fileScrollPane.setBounds(40, 100, 520, 200);

textArea = new JTextArea();
textArea.setFont(new Font("Viner Hand ITC", Font.PLAIN, 30)); // Set larger font
size
textArea.setLineWrap(true);
textArea.setEditable(false);
textArea.setWrapStyleWord(true);
textArea.setOpaque(false); // Make text area background transparent

// Set custom cell renderer for the fileList
FileList.setCellRenderer(new DefaultListCellRenderer() {
    @Override
    public Component getListCellRendererComponent(JList<?> list, Object value,
int index, boolean isSelected, boolean cellHasFocus) {
        JLabel label = (JLabel) super.getListCellRendererComponent(list, value,
index, isSelected, cellHasFocus);
        label.setFont(new Font("Viner Hand ITC", Font.BOLD, 26)); // Enlarge font
size for file names
        return label;
    }
});

// Custom JPanel with background image
backgroundPanel = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {

```

```

super.paintComponent(g);
// Draw background image
ImageIcon imageIcon = new
ImageIcon(getClass().getResource("/loadbg.png"));
Image image = imageIcon.getImage();
g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
}
};

backgroundPanel.setLayout(new BorderLayout());
backgroundPanel.add(textArea, BorderLayout.CENTER);

ScrollPane = new JScrollPane(backgroundPanel);

ScrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
ScrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
ScrollPane.setBounds(40, 320, 520, 650);

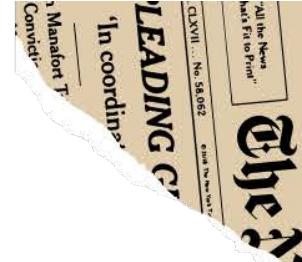
framecmain2.add(BACK);
framecmain2.add(LOAD);
framecmain2.add(DELETE);
framecmain2.add(fileScrollPane);
framecmain2.add(ScrollPane);

framecmain2.setBounds(660, 0, 600, 1020);
framecmain2.setResizable(false);
framecmain2.setLayout(null);
framecmain2.setVisible(true);
}

private void loadSavedFiles() {
File folder = new File(".");
File[] files = folder.listFiles((dir, name) -> name.endsWith(".txt"));
ListModel.clear();
if (files != null) {
for (File file : files) {
ListModel.addElement(file.getName());
}
}
}

public void actionPerformed(ActionEvent e) {
JButton choice = (JButton) e.getSource();
String choices = choice.getName();
if (choices.equals("back")) {
new CMMain();
framecmain2.dispose();
} else if (choices.equals("load")) {
String selectedFile = fileList.getSelectedValue();
if (selectedFile != null) {
try {
String content = new String(Files.readAllBytes(Paths.get(selectedFile)));
textArea.setText(content);
} catch (IOException ioException) {
}
}
}
}

```



```
        ioException.printStackTrace();
    }
}
} else if (choices.equals("delete")) {
    String selectedFile = FileList.getSelectedValue();
    if (selectedFile != null) {
        int confirm = JOptionPane.showConfirmDialog(framecm2, "Are you sure
you want to delete " + selectedFile + "?", "Confirm Delete",
JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {
            try {
                Files.delete(Paths.get(selectedFile));
                loadSavedFiles();
                textArea.setText("");
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
        }
    }
}

class BackgroundPanel extends JPanel {
    private Image backgroundImage;

    public BackgroundPanel(String fileName) {
        try {
            backgroundImage = new ImageIcon(fileName).getImage();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (backgroundImage != null) {
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
        }
    }
}

class CMB extends JFrame implements ActionListener {
    private JFrame framecm;
    private JButton BACK, SAVE;
    private JTextArea textArea;
    private JScrollPane scrollPane;

    public CMB() {
        framecm = new JFrame();
        framecm.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CMPage2.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framecm.setIconImage(logo);
    }
}
```





```
framecm.setTitle("CHEF'S CACHE");

SAVE = new JButton();
SAVE.setIcon(new ImageIcon(getClass().getResource("/SAVE2.png")));
SAVE.setBounds(170, 870, 240, 100);
SAVE.setOpaque(false);
SAVE.setContentAreaFilled(false);
SAVE.setBorderPainted(false);
SAVE.addActionListener(this);
SAVE.setName("save2");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10, 10, 70, 70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

textArea = new JTextArea();
textArea.setText("_____TITLE_____ \n• ");
textArea.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
textArea.setLineWrap(true); // Enable line wrap
textArea.setBackground(new Color(255, 239, 213));
textArea.setFont(new Font("Viner Hand ITC", Font.PLAIN, 30));
textArea.setBounds(40, 199, 510, 670);

// Add KeyListener to detect Enter key press
textArea.addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER) {
            e.consume(); // Consume the Enter key event
            int caretPosition = textArea.getCaretPosition();
            textArea.insert("\n• ", caretPosition);
        }
    }
});

// Create a JScrollPane and add the JTextArea to it
scrollPane = new JScrollPane(textArea);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);

// Set document to limit text length
((PlainDocument) textArea.getDocument()).setDocumentFilter(new DocumentSizeFilter(300));

// Add the JScrollPane to the JFrame
scrollPane.setBounds(40, 199, 510, 670);
```

```

framecm.add(scrollPane);

framecm.add(SAVE);
framecm.add(BACK);

framecm.setBounds(660, 0, 600, 1020);
framecm.setResizable(false);
framecm.setLayout(null);
framecm.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) {
        new CMMainB();
        framecm.dispose();
    } else if (choices.equals("save2")) {
        // Prompt for file name
        String fileName = JOptionPane.showInputDialog(framecm, "Enter file name:",
        "Save File", JOptionPane.PLAIN_MESSAGE);
        if (fileName != null && !fileName.trim().isEmpty()) {
            String content = textArea.getText();
            try {
                Files.write(Paths.get(fileName + ".txt"), content.getBytes());
                JTextField textField = new JTextField("Your Menu is Saved as " + fileName
                + ".txt");
                textField.setEditable(false);
                JOptionPane.showMessageDialog(framecm, textField, "Save
Confirmation", JOptionPane.INFORMATION_MESSAGE);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
        }
    }
}

// Document filter to limit text length
class DocumentSizeFilter extends javax.swing.text.DocumentFilter {
    int maxCharacters;

    public DocumentSizeFilter(int maxChars) {
        maxCharacters = maxChars;
    }

    public void insertString(javax.swing.text.DocumentFilter.FilterBypass fb, int offs,
    String str, javax.swing.text.AttributeSet a) throws BadLocationException {
        if ((fb.getDocument().getLength() + str.length()) <= maxCharacters) {
            super.insertString(fb, offs, str, a);
        }
    }

    public void replace(javax.swing.text.DocumentFilter.FilterBypass fb, int offs, int
    length, String str, javax.swing.text.AttributeSet a) throws BadLocationException {
        if ((fb.getDocument().getLength() + str.length() - length) <= maxCharacters) {
            super.replace(fb, offs, length, str, a);
        }
    }
}

```

```
        }  
    }  
}
```

```
class CMB2 implements ActionListener {  
    private JFrame framecmain2;  
    private JButton BACK, LOAD, DELETE;  
    private JTextArea textArea;  
    private JScrollPane scrollPane;  
    private DefaultListModel<String> listModel;  
    private JList<String> fileList;  
    private JPanel backgroundPanel2;  
  
    public CMB2() {  
        framecmain2 = new JFrame();  
        framecmain2.setContentPane(new JLabel(new  
ImageIcon(getClass().getResource("/CMSave2.png"))));  
        Image logo =  
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));  
        framecmain2.setIconImage(logo);  
        framecmain2.setTitle("CHEF'S CACHE");  
  
        BACK = new JButton();  
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));  
        BACK.setOpaque(false);  
        BACK.setContentAreaFilled(false);  
        BACK.setBounds(10,10,70,70);  
        BACK.setBorderPainted(false);  
        ImageIcon backA = new ImageIcon();  
        BACK.setPressedIcon(backA);  
        BACK.addActionListener(this);  
        BACK.setName("back");  
  
        LOAD = new JButton();  
        LOAD.setIcon(new ImageIcon(getClass().getResource("/load2.png")));  
        LOAD.setOpaque(false);  
        LOAD.setContentAreaFilled(false);  
        LOAD.setBounds(100, 5, 240, 100);  
        LOAD.setBorderPainted(false);  
        ImageIcon loadA = new ImageIcon();  
        LOAD.setPressedIcon(loadA);  
        LOAD.addActionListener(this);  
        LOAD.setName("load");  
  
        DELETE = new JButton();  
        DELETE.setIcon(new ImageIcon(getClass().getResource("/delete2.png")));  
        DELETE.setOpaque(false);  
        DELETE.setContentAreaFilled(false);  
        DELETE.setBounds(350, 5, 240, 100);  
        DELETE.setBorderPainted(false);  
        ImageIcon deleteA = new ImageIcon();  
        DELETE.setPressedIcon(deleteA);  
        DELETE.addActionListener(this);  
        DELETE.setName("delete");  
  
        listModel = new DefaultListModel<>();
```

```

fileList = new JList<>(listModel);
fileList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
loadSavedFiles();

JScrollPane fileScrollPane = new JScrollPane(fileList);
fileScrollPane.setBounds(40, 100, 520, 200);

textArea = new JTextArea();
textArea.setFont(new Font("Viner Hand ITC", Font.PLAIN, 30)); // Set larger font
size
textArea.setLineWrap(true);
textArea.setEditable(false);
textArea.setWrapStyleWord(true);
textArea.setOpaque(false); // Make text area background transparent

// Set custom cell renderer for the fileList
fileList.setCellRenderer(new DefaultListCellRenderer() {
    @Override
    public Component getListCellRendererComponent(JList<?> list, Object value,
int index, boolean isSelected, boolean cellHasFocus) {
        JLabel label = (JLabel) super.getListCellRendererComponent(list, value,
index, isSelected, cellHasFocus);
        label.setFont(new Font("Viner Hand ITC", Font.BOLD, 26)); // Enlarge font
size for file names
        return label;
    }
});

// Custom JPanel with background image
backgroundPanel2 = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        // Draw background image
        ImageIcon imagelcon = new
ImageIcon(getClass().getResource("/loadbg2.png"));
        Image image = imagelcon.getImage();
        g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
    }
};
backgroundPanel2.setLayout(new BorderLayout());
backgroundPanel2.add(textArea, BorderLayout.CENTER);

scrollPane = new JScrollPane(backgroundPanel2);

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
scrollPane.setBounds(40, 320, 520, 650);

framecmain2.add(BACK);
framecmain2.add(LOAD);
framecmain2.add(DELETE);
framecmain2.add(fileScrollPane);
framecmain2.add(scrollPane);

```

```

framecmain2.setBounds(660, 0, 600, 1020);
framecmain2.setResizable(false);
framecmain2.setLayout(null);
framecmain2.setVisible(true);
}

private void loadSavedFiles() {
    File folder = new File(".");
    File[] files = folder.listFiles((dir, name) -> name.endsWith(".txt"));
    listModel.clear();
    if (files != null) {
        for (File file : files) {
            listModel.addElement(file.getName());
        }
    }
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) {
        new CMMainB();
        framecmain2.dispose();
    } else if (choices.equals("load")) {
        String selectedFile = fileList.getSelectedValue();
        if (selectedFile != null) {
            try {
                String content = new String(Files.readAllBytes(Paths.get(selectedFile)));
                textArea.setText(content);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
        }
    } else if (choices.equals("delete")) {
        String selectedFile = fileList.getSelectedValue();
        if (selectedFile != null) {
            int confirm = JOptionPane.showConfirmDialog(framecmain2, "Are you sure  
you want to delete " + selectedFile + "?", "Confirm Delete",  
JOptionPane.YES_NO_OPTION);
            if (confirm == JOptionPane.YES_OPTION) {
                try {
                    Files.delete(Paths.get(selectedFile));
                    loadSavedFiles();
                    textArea.setText("");
                } catch (IOException ioException) {
                    ioException.printStackTrace();
                }
            }
        }
    }
}

class BackgroundPanel extends JPanel {
    private Image backgroundImage;

    public BackgroundPanel(String fileName) {

```



```
try {
    backgroundImage = new ImageIcon(fileName).getImage();
} catch (Exception e) {
    e.printStackTrace();
}

@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    if (backgroundImage != null) {
        g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
    }
}
}

class CMMain implements ActionListener{
    private JFrame framecmain;
    private JButton BACK, CMA, CMB;

    public CMMain(){
        framecmain = new JFrame();
        framecmain.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CMMain.png"))));
        Image logo =
            Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framecmain.setIconImage(logo);
        framecmain.setTitle("CHEF'S CACHE");

        CMA      = new JButton();
        CMA.setIcon(new ImageIcon(getClass().getResource("/CMA.png")));
        CMA.setBounds(150,300,280,160);
        CMA.setOpaque(false);
        CMA.setContentAreaFilled(false);
        CMA.setBorderPainted(false);
        ImageIcon cmA = new ImageIcon();
        CMA.setPressedIcon(cmA);
        CMA.addActionListener(this);
        CMA.setName("cma");

        CMB      = new JButton();
        CMB.setIcon(new ImageIcon(getClass().getResource("/CMB.png")));
        CMB.setBounds(150,550,280,160);
        CMB.setOpaque(false);
        CMB.setContentAreaFilled(false);
        CMB.setBorderPainted(false);
        ImageIcon cmB = new ImageIcon();
        CMB.setPressedIcon(cmB);
        CMB.addActionListener(this);
        CMB.setName("cmb");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == CMA) {
            framecmain.dispose();
        } else if (e.getSource() == CMB) {
            framecmain.dispose();
        } else if (e.getSource() == BACK) {
            framecmain.dispose();
        }
    }
}
```



```

BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framecmain.add(CMA);
framecmain.add(CMB);
framecmain.add(BACK);

framecmain.setBounds(660,0,600,1020);
framecmain.setResizable(false);
framecmain.setLayout(null);
framecmain.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new Homepage();
        framecmain.dispose();
    }
    else if(choices == "cma"){
        new CM();
        framecmain.dispose();
    }
    else if(choices == "cmb"){
        new CM2();
        framecmain.dispose();
    }
}

}

class CMMainB implements ActionListener{
    private JFrame framecmain;
    private JButton BACK, CMA, CMB;

    public CMMainB(){
        framecmain = new JFrame();
        framecmain.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CMMain2.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framecmain.setIconImage(logo);
        framecmain.setTitle("CHEF'S CACHE");

        CMA = new JButton();
        CMA.setIcon(new ImageIcon(getClass().getResource("/CMA2.png")));
        CMA.setBounds(150,300,280,160);
        CMA.setOpaque(false);
        CMA.setContentAreaFilled(false);
        CMA.setBorderPainted(false);
        ImageIcon cmA = new ImageIcon();
        CMA.setPressedIcon(cmA);
    }
}

```

```

CMA.addActionListener(this);
CMA.setName("cma");

CMB = new JButton();
CMB.setIcon(new ImageIcon(getClass().getResource("/CMB2.png")));
CMB.setBounds(150,550,280,160);
CMB.setOpaque(false);
CMB.setContentAreaFilled(false);
CMB.setBorderPainted(false);
ImageIcon cmB = new ImageIcon();
CMB.setPressedIcon(cmB);
CMB.addActionListener(this);
CMB.setName("cmb");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framecmain.add(CMA);
framecmain.add(CMB);
framecmain.add(BACK);

framecmain.setBounds(660,0,600,1020);
framecmain.setResizable(false);
framecmain.setLayout(null);
framecmain.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new HomepageB();
        framecmain.dispose();
    }
    else if(choices == "cma"){
        new CMB();
        framecmain.dispose();
    }
    else if(choices == "cmb"){
        new CMB2();
        framecmain.dispose();
    }
}
}

class CTL implements ActionListener {
    private JFrame framecl;

```

```

private JButton BACK, WIFI, SUGGEST, MENUA, MENUB, MENUUC, MENUD,
MENUE;
private static boolean wifiStatus = false; // Static variable to store WiFi status

public CTL() {
    framectl = new JFrame();
    framectl.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CTLPage.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    framectl.setIconImage(logo);
    framectl.setTitle("CHEF'S CACHE");

    WIFI = new JButton();
    WIFI.setIcon(new ImageIcon(getClass().getResource("/wifioff.png")));
    WIFI.setOpaque(false);
    WIFI.setContentAreaFilled(false);
    WIFI.setBounds(510, 10, 70, 70);
    WIFI.setBorderPainted(false);
    ImageIcon wifiA = new ImageIcon();
    WIFI.setPressedIcon(wifiA);
    WIFI.addActionListener(this); // Add ActionListener to WiFi button
    WIFI.setName("wifi"); // Set the name of the WiFi button

    MENUA = new JButton();
    MENUA.setIcon(new ImageIcon(getClass().getResource("/filipino.png")));
    MENUA.setOpaque(false);
    MENUA.setContentAreaFilled(false);
    MENUA.setBounds(100, 210, 200, 200);
    MENUA.setBorderPainted(false);
    ImageIcon menuA = new ImageIcon();
    MENUA.setPressedIcon(menuA);
    MENUA.addActionListener(this);
    MENUA.setName("menua");

    MENUB = new JButton();
    MENUB.setIcon(new ImageIcon(getClass().getResource("/french.png")));
    MENUB.setOpaque(false);
    MENUB.setContentAreaFilled(false);
    MENUB.setBounds(300, 280, 200, 200);
    MENUB.setBorderPainted(false);
    ImageIcon menuB = new ImageIcon();
    MENUB.setPressedIcon(menuB);
    MENUB.addActionListener(this);
    MENUB.setName("menub");

    MENUUC = new JButton();
    MENUUC.setIcon(new ImageIcon(getClass().getResource("/italian.png")));
    MENUUC.setOpaque(false);
    MENUUC.setContentAreaFilled(false);
    MENUUC.setBounds(100, 400, 200, 200);
    MENUUC.setBorderPainted(false);
    ImageIcon menuC = new ImageIcon();
    MENUUC.setPressedIcon(menuC);
    MENUUC.addActionListener(this);
    MENUUC.setName("menuc");
}

```



```
MENUD = new JButton();
MENUD.setIcon(new ImageIcon(getClass().getResource("/japan.png")));
MENUD.setOpaque(false);
MENUD.setContentAreaFilled(false);
MENUD.setBounds(300, 480, 200, 200);
MENUD.setBorderPainted(false);
ImageIcon menuD = new ImageIcon();
MENUD.setPressedIcon(menuD);
MENUD.addActionListener(this);
MENUD.setName("menuD");

MENUE = new JButton();
MENUE.setIcon(new ImageIcon(getClass().getResource("/korean.png")));
MENUE.setOpaque(false);
MENUE.setContentAreaFilled(false);
MENUE.setBounds(100, 600, 200, 200);
MENUE.setBorderPainted(false);
ImageIcon menuE = new ImageIcon();
MENUE.setPressedIcon(menuE);
MENUE.addActionListener(this);
MENUE.setName("menuE");

SUGGEST = new JButton();
SUGGEST.setIcon(new ImageIcon(getClass().getResource("/SUGGEST.png")));
SUGGEST.setOpaque(false);
SUGGEST.setContentAreaFilled(false);
SUGGEST.setBounds(170, 800, 300, 120);
SUGGEST.setBorderPainted(false);
SUGGEST.setVisible(false); // Initially hidden
ImageIcon suggestA = new ImageIcon();
SUGGEST.setPressedIcon(suggestA);
SUGGEST.addActionListener(this);
SUGGEST.setName("suggest");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10, 10, 70, 70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framectl.add(MENUA);
framectl.add(MENUB);
framectl.add(MENUC);
framectl.add(MENUD);
framectl.add(MENUE);
framectl.add(SUGGEST);
framectl.add(WIFI);
framectl.add(BACK);

framectl.setBounds(660, 0, 600, 1020);
framectl.setResizable(false);
framectl.setLayout(null);
```

```

framectl.setVisible(true);

updateWifilcon(); // Set initial WiFi icon
toggleSuggestButton(wifiStatus); // Set initial visibility of SUGGEST button based
on WiFi status
}

public void actionPerformed(ActionEvent e) {
JButton choice = (JButton) e.getSource();
String choices = choice.getName();
if (choices.equals("back")) {
    new Homepage();
    framectl.dispose();
} else if (choices.equals("menua")) {
    new MENUA();
    framectl.dispose();
} else if (choices.equals("menub")) {
    new MENUB();
    framectl.dispose();
} else if (choices.equals("menuc")) {
    new MENUC();
    framectl.dispose();
} else if (choices.equals("menud")) {
    new MENUD();
    framectl.dispose();
} else if (choices.equals("menue")) {
    new MENUE();
    framectl.dispose();
} else if (choices.equals("suggest")) {
    new SUGGEST(wifiStatus); // Pass WiFi status to SUGGEST class
    framectl.dispose();
} else if (choices.equals("wifi")) {
    wifiStatus = !wifiStatus;
    updateWifilcon(); // Update WiFi icon based on the status
    toggleSuggestButton(wifiStatus); // Toggle suggestion button visibility
}
}

private void updateWifilcon() {
WIFI.setIcon(new ImageIcon(wifiStatus ?
    getClass().getResource("/wifion.png") :
    getClass().getResource("/wifioff.png")));
}

private void toggleSuggestButton(boolean wifiOn) {
    SUGGEST.setVisible(wifiOn);
}
}

class CTLB implements ActionListener {
private JFrame framectl;
private JButton BACK, WIFI, SUGGEST, MENUA, MENUB, MENUC, MENUD,
MENUE;
private static boolean wifiStatusB = false; // Static variable to store WiFi status

```

```

public CTLB(){
    framectl = new JFrame();
    framectl.setContentPane(new JLabel(new
    ImageIcon(getClass().getResource("/CTLPage2.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    framectl.setIconImage(logo);
    framectl.setTitle("CHEF'S CACHE");

    WIFI = new JButton();
    WIFI.setIcon(new ImageIcon(getClass().getResource("/wifioff.png")));
    WIFI.setOpaque(false);
    WIFI.setContentAreaFilled(false);
    WIFI.setBounds(510, 10, 70, 70);
    WIFI.setBorderPainted(false);
    ImageIcon wifiA = new ImageIcon();
    WIFI.setPressedIcon(wifiA);
    WIFI.addActionListener(this); // Add ActionListener to WiFi button
    WIFI.setName("wifi"); // Set the name of the WiFi button

    MENUA = new JButton();
    MENUA.setIcon(new ImageIcon(getClass().getResource("/filipinob.png")));
    MENUA.setOpaque(false);
    MENUA.setContentAreaFilled(false);
    MENUA.setBounds(100, 210, 200, 200);
    MENUA.setBorderPainted(false);
    ImageIcon menuA = new ImageIcon();
    MENUA.setPressedIcon(menuA);
    MENUA.addActionListener(this);
    MENUA.setName("menua");

    MENUB = new JButton();
    MENUB.setIcon(new ImageIcon(getClass().getResource("/frenchb.png")));
    MENUB.setOpaque(false);
    MENUB.setContentAreaFilled(false);
    MENUB.setBounds(300, 280, 200, 200);
    MENUB.setBorderPainted(false);
    ImageIcon menuB = new ImageIcon();
    MENUB.setPressedIcon(menuB);
    MENUB.addActionListener(this);
    MENUB.setName("menub");

    MENUC = new JButton();
    MENUC.setIcon(new ImageIcon(getClass().getResource("/italianb.png")));
    MENUC.setOpaque(false);
    MENUC.setContentAreaFilled(false);
    MENUC.setBounds(100, 400, 200, 200);
    MENUC.setBorderPainted(false);
    ImageIcon menuC = new ImageIcon();
    MENUC.setPressedIcon(menuC);
    MENUC.addActionListener(this);
    MENUC.setName("menuc");

    MENUD = new JButton();
    MENUD.setIcon(new ImageIcon(getClass().getResource("/japanb.png")));
    MENUD.setOpaque(false);
    MENUD.setContentAreaFilled(false);

```

```

MENUD.setBounds(300, 480, 200, 200);
MENUD.setBorderPainted(false);
ImageIcon menuD = new ImageIcon();
MENUD.setPressedIcon(menuD);
MENUD.addActionListener(this);
MENUD.setName("menud");

MENUE = new JButton();
MENUE.setIcon(new ImageIcon(getClass().getResource("/koreanb.png")));
MENUE.setOpaque(false);
MENUE.setContentAreaFilled(false);
MENUE.setBounds(100, 600, 200, 200);
MENUE.setBorderPainted(false);
ImageIcon menuE = new ImageIcon();
MENUE.setPressedIcon(menuE);
MENUE.addActionListener(this);
MENUE.setName("menue");

SUGGEST = new JButton();
SUGGEST.setIcon(new ImageIcon(getClass().getResource("/SUGGESTB.png")));
SUGGEST.setOpaque(false);
SUGGEST.setContentAreaFilled(false);
SUGGEST.setBounds(160, 800, 300, 120);
SUGGEST.setBorderPainted(false);
SUGGEST.setVisible(false); // Initially hidden
ImageIcon suggestA = new ImageIcon();
SUGGEST.setPressedIcon(suggestA);
SUGGEST.addActionListener(this);
SUGGEST.setName("suggest");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10, 10, 70, 70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framectl.add(MENUA);
framectl.add(MENUB);
framectl.add(MENUC);
framectl.add(MENUD);
framectl.add(MENUE);
framectl.add(SUGGEST);
framectl.add(WIFI);
framectl.add(BACK);

framectl.setBounds(660, 0, 600, 1020);
framectl.setResizable(false);
framectl.setLayout(null);
framectl.setVisible(true);

updateWifilcon(); // Set initial WiFi icon

```

```
    toggleSuggestButtonB(wifiStatusB); // Set initial visibility of SUGGEST button  
    based on WiFi status
```

```
}
```

```
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton) e.getSource();  
    String choices = choice.getName();  
    if (choices.equals("back")) {  
        new HomepageB();  
        framectl.dispose();  
    } else if (choices.equals("menua")) {  
        new MENUA_B();  
        framectl.dispose();  
    } else if (choices.equals("menub")) {  
        new MENUB_B();  
        framectl.dispose();  
    } else if (choices.equals("menuc")) {  
        new MENU_C_B();  
        framectl.dispose();  
    } else if (choices.equals("menud")) {  
        new MENU_D_B();  
        framectl.dispose();  
    } else if (choices.equals("menue")) {  
        new MENU_E_B();  
        framectl.dispose();  
    } else if (choices.equals("suggest")) {  
        new SUGGESTB(wifiStatusB); // Pass WiFi status to SUGGEST class  
        framectl.dispose();  
    } else if (choices.equals("wifi")) {  
        wifiStatusB = !wifiStatusB;  
        updateWifilcon(); // Update WiFi icon based on the status  
        toggleSuggestButtonB(wifiStatusB); // Toggle suggestion button visibility  
    }  
}
```

```
private void updateWifilcon() {  
    WIFI.setIcon(new ImageIcon(wifiStatusB ?  
        getClass().getResource("/wifion2.png") :  
        getClass().getResource("/wifioff2.png")));  
}
```

```
private void toggleSuggestButtonB(boolean wifiOn) {  
    SUGGEST.setVisible(wifiOn);  
}  
}
```

```
class MENUA implements ActionListener {  
    private JFrame framemenu;  
    private JButton BACK;  
    private JTextArea menutext;  
    private JScrollPane menuscroll;
```

```
    public MENUA() {  
        framemenu = new JFrame();
```



```
framemenu.setContentPane(new JLabel(new  
ImageIcon(getClass().getResource("/FilipinoBG.png"))));  
Image logo =  
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));  
framemenu.setIconImage(logo);  
framemenu.setTitle("CHEF'S CACHE");  
  
BACK = new JButton();  
BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));  
BACK.setOpaque(false);  
BACK.setContentAreaFilled(false);  
BACK.setBounds(10, 10, 70, 70);  
BACK.setBorderPainted(false);  
ImageIcon backA = new ImageIcon();  
BACK.setPressedIcon(backA);  
BACK.addActionListener(this);  
BACK.setName("back");  
  
menutext = new JTextArea();  
menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger  
font size  
menutext.setEditable(false); // Set as non-editable  
menutext.setOpaque(true); // Make it opaque to show background color  
menutext.setForeground(Color.BLACK); // Set text color  
menutext.setText(" ----FILIPINO FOOD RECIPE----"  
+ "\n"  
+ "\n1.ADOBONG MANOK"  
+ "\n"  
+ "\nIngredients: "  
+ "\n• 2 tablespoons vegetable oil 1 (3 pound) chicken, cut into pieces "  
+ "\n• 1 large onion, quartered and sliced 2 tablespoons minced garlic "  
+ "\n• 2/3 cup low sodium soy sauce "  
+ "\n• 1/3 cup white vinegar"  
+ "\n• 1 tablespoon garlic powder"  
+ "\n• 2 teaspoons black pepper"  
+ "\n• 1 bay leaf"  
+ "\n"  
+ "\nHow to cook:"  
+ "\n"  
+ "\nStep 1: Heat vegetable oil in a large skillet over medium-high heat.  
Cook chicken pieces until golden brown, 2 to 3 minutes per side. Transfer chicken to a  
plate and set aside."  
+ "\n"  
+ "\nStep 2: Add onion and garlic to the skillet; cook until softened and  
brown, about 6 minutes."  
+ "\n"  
+ "\nStep 3: Pour in soy sauce and vinegar and season with garlic powder,  
black pepper, and bay leaf."  
+ "\n"  
+ "\nStep 4: Return chicken to pan, increase heat to high, and bring to a  
boil. Reduce heat to medium-low, cover, and simmer until chicken is tender and cooked  
through, 35 to 40 minutes."  
+ "\n"  
+ "\n2.SINIGANG NA BABOY "  
+ "\n"  
+ "\nIngredients:"  
+ "\n• 4 1/4 cups water "
```



+ "\n• 2 cups pork spare rib cut into chunks"
+ "\n• 1 pc red onion, quartered"
+ "\n• 2 pcs medium sized tomato, quartered"
+ "\n• 8 pcs sitaw, sliced into 2" pieces"
+ "\n• 250 g kangkong stalk and leaves"
+ "\n• 1 pc talong, sliced"
+ "\n• 2 pcs siling pangsigang"
+ "\n• 1 (20g) pack Knorr Sinigang sa Sampalok Mix Original"
+ "\n• Optional Vegetable: Radish(labanos) and Okra"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Begin by washing pork spare ribs in cold running water to remove any visible impurities. Then, place pork ribs submerge in a medium pot with a liter of water. Bring to a boil without covering the pot and allow the scum to float to the surface. Meticulously remove the scums from the surface of the water using a slotted spoon or a spatula. Scums are impurities found in meat."
+ "\n"
+ "\nStep 2: After the broth clears up, turn you heat to medium setting, then it's time toss in your onions and tomatoes to add flavor to the broth. Cover your pot and let it simmer to fully cook your meat. Use a fork to pierce the meat so that the flavors would also enter the meaty parts of the spare ribs, You also do this so you can tell if it's already set and tender."
+ "\n"
+ "\nStep 3: You can already add your vegetables. Start by adding the sitaw, kangkong stalks and talong into the pot and cook until these turn dark green. If you decide to use labanos and okra, you can also add those at this point. This should take roughly around 2-3 minutes."
+ "\n"
+ "\nStep 4: Add in your siling sigang, kangkong leaves and the Knorr Sinigang sa Sampaloc Mix. Stir and let it simmer. It should be done in about 2-3 minutes."
+ "\n"
+ "\nStep 5: All done! Transfer your Sinigang to a serving bowl. This is a happy treat that warms, soothes and satisfies everyone in the family! Remember to prepare extra servings of rice to enjoy the extra asim kilig taste of this Filipino classic soup. Enjoy!"
+ "\n"
+ "\nMORE FILIPINO FOOD TO SERVE SOON!");

```
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as
needed rgb(216, 174, 126)
```

```
menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
framemenu.add(menuscroll); // Add scrollable text area to the frame
framemenu.add(BACK);
```



```

framemenu.setBounds(660, 0, 600, 1020);
framemenu.setResizable(false);
framemenu.setLayout(null);
framemenu.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new CTL();
        framemenu.dispose();
    }
}
}

class MENUA_B implements ActionListener {
private JFrame framemenu;
private JButton BACK;
private JTextArea menutext;
private JScrollPane menuscroll;

public MENUA_B() {
    framemenu = new JFrame();
    framemenu.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/FilipinoBG.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    framemenu.setIconImage(logo);
    framemenu.setTitle("CHEF'S CACHE");

    BACK = new JButton();
    BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
    BACK.setOpaque(false);
    BACK.setContentAreaFilled(false);
    BACK.setBounds(10, 10, 70, 70);
    BACK.setBorderPainted(false);
    ImageIcon backA = new ImageIcon();
    BACK.setPressedIcon(backA);
    BACK.addActionListener(this);
    BACK.setName("back");

    menutext = new JTextArea();
    menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
    menutext.setEditable(false); // Set as non-editable
    menutext.setOpaque(true); // Make it opaque to show background color
    menutext.setForeground(Color.BLACK); // Set text color
    menutext.setText(" ----FILIPINO FOOD RECIPE----"
+ "\n"
+ "\n1.ADOBONG MANOK"
+ "\n"

```



+ "\nIngredients: "
+ "\n• 2 tablespoons vegetable oil 1 (3 pound) chicken, cut into pieces "
+ "\n• 1 large onion, quartered and sliced 2 tablespoons minced garlic "
+ "\n• 2/3 cup low sodium soy sauce "
+ "\n• 1/3 cup white vinegar"
+ "\n• 1 tablespoon garlic powder"
+ "\n• 2 teaspoons black pepper"
+ "\n• 1 bay leaf"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Heat vegetable oil in a large skillet over medium-high heat.
Cook chicken pieces until golden brown, 2 to 3 minutes per side. Transfer chicken to a plate and set aside."
+ "\n"
+ "\nStep 2: Add onion and garlic to the skillet; cook until softened and brown, about 6 minutes."
+ "\n"
+ "\nStep 3: Pour in soy sauce and vinegar and season with garlic powder, black pepper, and bay leaf."
+ "\n"
+ "\nStep 4: Return chicken to pan, increase heat to high, and bring to a boil. Reduce heat to medium-low, cover, and simmer until chicken is tender and cooked through, 35 to 40 minutes."
+ "\n"
+ "\n2. SINIGANG NA BABOY "
+ "\n"
+ "\nIngredients:"
+ "\n• 4 1/4 cups water "
+ "\n• 2 cups pork spare rib cut into chunks"
+ "\n• 1 pc red onion, quartered"
+ "\n• 2 pcs medium sized tomato, quartered"
+ "\n• 8 pcs sitaw, sliced into 2" pieces"
+ "\n• 250 g kangkong stalk and leaves"
+ "\n• 1 pc talong, sliced"
+ "\n• 2 pcs siling pangsigang"
+ "\n• 1 (20g) pack Knorr Sinigang sa Sampalok Mix Original"
+ "\n• Optional Vegetable: Radish(labanos) and Okra"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Begin by washing pork spare ribs in cold running water to remove any visible impurities. Then, place pork ribs submerge in a medium pot with a liter of water. Bring to a boil without covering the pot and allow the scum to float to the surface. Meticulously remove the scums from the surface of the water using a slotted spoon or a spatula. Scums are impurities found in meat."
+ "\n"
+ "\nStep 2: After the broth clears up, turn you heat to medium setting, then it's time toss in your onions and tomatoes to add flavor to the broth. Cover your pot and let it simmer to fully cook your meat. Use a fork to pierce the meat so that the flavors would also enter the meaty parts of the spare ribs, You also do this so you can tell if it's already set and tender."
+ "\n"
+ "\nStep 3: You can already add your vegetables. Start by adding the sitaw, kangkong stalks and talong into the pot and cook until these turn dark green. If you decide to use labanos and okra, you can also add those at this point. This should take roughly around 2-3 minutes."



+ "\n"
 + "\nStep 4: Add in your siling sigang, kangkong leaves and the Knorr Sinigang sa Sampaloc Mix. Stir and let it simmer. It should be done in about 2-3 minutes."
 + "\n"
 + "\nStep 5: All done! Transfer your Sinigang to a serving bowl. This is a happy treat that warms, soothes and satisfies everyone in the family! Remember to prepare extra servings of rice to enjoy the extra asim kilig taste of this Filipino classic soup. Enjoy!"
 + "\n"
 + "\nMORE FILIPINO FOOD TO SERVE SOON!");

```

menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as
needed rgb(216, 174, 126)
```

```

menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_
ALWAYS); // Enable vertical scrolling
```

```

framemenu.add(menuscroll); // Add scrollable text area to the frame
framemenu.add(BACK);

framemenu.setBounds(660, 0, 600, 1020);
framemenu.setResizable(false);
framemenu.setLayout(null);
framemenu.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);
}
```

```

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new CTLB();
        framemenu.dispose();
    }
}
```

```

class MENUB implements ActionListener{
    private JFrame framemenu;
    private JButton BACK;
    private JTextArea menutext;
    private JScrollPane menuscroll;
```

```
    public MENUB(){
```



```
framemenu = new JFrame();
framemenu.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/FrenchBG.png"))));
Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
framemenu.setIconImage(logo);
framemenu.setTitle("CHEF'S CACHE");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

menutext = new JTextArea();
menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("This is some sample text.\nYou can add more text here.");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(255, 255, 236)); // Adjust RGB values as
needed rgb(255, 255, 236)
menutext.setText(" ----FRENCH FOOD RECIPE----"
+ "\n"
+ "\n1.CHICKEN CONFIT"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 4 chicken leg portions with thighs attached, excess fat trimmed and
reserved (about 2 pounds total)"
+ "\n• 1 tablespoon plus 1/8 teaspoon kosher salt"
+ "\n• 1/2 teaspoon freshly ground black pepper"
+ "\n• 10 garlic cloves"
+ "\n• 4 bay leaves"
+ "\n• 4 sprigs fresh thyme"
+ "\n• 1 1/2 teaspoons black peppercorns"
+ "\n• 1/2 teaspoon table salt"
+ "\n• 4 cups olive oil"
+ "\n"
+ "\nHow to cook:"
+ "\nStep 1: Lay the leg portions on a platter, skin side down. Sprinkle with
1 tablespoon of the kosher salt and black pepper. Place the garlic cloves, bay leaves,
and sprigs of thyme on each of 2 leg portions. Lay the remaining 2 leg portions, flesh to
flesh, on top. Put the reserved fat from the chicken in the bottom of a glass or plastic
container. Top with the sandwiched leg portions. Sprinkle with the remaining 1/8
teaspoon kosher salt. Cover and refrigerate for 12 hours."
+ "\n"
```



+ "\nStep 2: Preheat the oven to 200 degrees F."

+ "\n"

+ "\nStep 3: Remove the chicken from the refrigerator. Remove the garlic, bay leaves, thyme, and chicken fat and reserve. Rinse the chicken with cool water, rubbing off some of the salt and pepper. Pat dry with paper towels."

+ "\n"

+ "\nStep 4: Put the reserved garlic, bay leaves, thyme, and chicken fat in the bottom of an enameled cast iron pot. Sprinkle evenly with the peppercorns and salt. Lay the chicken on top, skin side down. Add the olive oil. Cover and bake for 12 to 14 hours, or until the meat pulls away from the bone."

+ "\n"

+ "\nStep 5: Remove the chicken from the fat. Strain the fat and reserve. Pick the meat from the bones and place it in a stoneware container. Cover the meat with some of the strained fat so that there is a 1/4-inch layer of fat on top. The chicken confit can be stored in the refrigerator for up to 1 month."

+ "\n"

+ "\nStep 6: The excess oil can be stored in an airtight container in the refrigerator and used like butter for cooking. The tinge of chicken taste in the oil is wonderful and I use the oil to roast potatoes, cook green beans, and pan-fry veal."

+ "\n"

+ "\n"

+ "\n2.LAMB SHANK NAVARIN"

+ "\n"

+ "\nIngredients:"

+ "\n"

+ "\n• 2 tbsp plain flour"

+ "\n• 6 (about 2kg) lamb shanks, French trimmed"

+ "\n• 2 tbsp olive oil"

+ "\n• 1 brown onion, halved, thinly sliced"

+ "\n• 2 garlic cloves, finely chopped"

+ "\n• 80ml (1/3 cup) white wine"

+ "\n• 500ml (2 cups) Massel chicken style liquid stock"

+ "\n• 500ml (2 cups) passata (tomato pasta sauce)"

+ "\n• 3 small sprigs fresh rosemary"

+ "\n• 2 bay leaves"

+ "\n• 6 baby Coliban potatoes, halved (see note)"

+ "\n• 1 turnip, peeled, cut into wedges"

+ "\n• 1 bunch baby (Dutch) carrots, trimmed, peeled"

+ "\n• 150g (1 cup) frozen peas"

+ "\n• 1/3 cup chopped fresh continental parsley"

+ "\n"

+ "\nHow to cook:"

+ "\n"

+ "\nStep 1: Place the flour on a plate. Season with salt and pepper. Toss the lamb shanks in the flour mixture to coat. Shake off excess. Heat half the oil in a large flameproof casserole dish over medium-high heat. Cook half the lamb, turning occasionally, for 7 minutes or until browned. Transfer to a plate. Repeat with the remaining lamb and oil, reheating the pan between batches."

+ "\n"

+ "\nStep 2: Add the onion and garlic to the dish. Cook, stirring, for 3 minutes or until soft. Add the wine and cook for 1 minute, scraping the pan with a flat-edged wooden spoon to dislodge any bits that have cooked onto the base. Stir in the stock, passata, rosemary and bay leaves. Return the lamb to the dish."

+ "\n"

+ "\nStep 3: Cover and reduce heat to low. Cook, turning lamb halfway through cooking, for 1 hour 45 minutes. Use a metal spoon to skim fat from surface. Add potato and turnip. Cook for 35 minutes or until lamb is tender. Use a slotted spoon



to transfer lamb, potato and turnip to a large plate. Add carrots to the dish. Increase heat to medium. Cook, uncovered, stirring occasionally, for 20 minutes or until carrots are tender and sauce has reduced."

+ "\n"

+ "\nStep 4: Add the peas to the carrot mixture. Return the lamb, potato and turnip to the dish. Cook, turning the lamb in the sauce occasionally, for 5 minutes or until heated through. Stir in the parsley. Season with salt and pepper."

+ "\n"

+ "\nMORE FRENCH FOOD TO SERVE SOON!");

```
menuscroll = new JScrollPane(menuText);
```

```
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
frameMenu.add(menuscroll); // Add scrollable text area to the frame  
frameMenu.add(BACK);
```

```
frameMenu.setBounds(660, 0, 600, 1020);  
frameMenu.setResizable(false);  
frameMenu.setLayout(null);  
frameMenu.setVisible(true);
```

```
// Set the caret position to the beginning  
menuText.setCaretPosition(0);  
}
```

```
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton)e.getSource();  
    String choices = choice.getName();  
    if (choices == "back"){  
        new CTL();  
        frameMenu.dispose();  
    }  
}
```

```
}
```

```
class MENU_B implements ActionListener{  
    private JFrame frameMenu;  
    private JButton BACK;  
    private JTextArea menuText;  
    private JScrollPane menuscroll;
```

```
public MENU_B(){  
    frameMenu = new JFrame();  
    frameMenu.setContentPane(new JLabel(new  
    ImageIcon(getClass().getResource("/FrenchBG.png"))));  
    Image logo =  
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));  
    frameMenu.setIconImage(logo);  
    frameMenu.setTitle("CHEF'S CACHE");  
  
    BACK = new JButton();  
    BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));  
    BACK.setOpaque(false);
```



```
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

menutext = new JTextArea();
menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("This is some sample text.\nYou can add more text here.");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(255, 255, 236)); // Adjust RGB values as
needed rgb(255, 255, 236)
menutext.setText(" ----FRENCH FOOD RECIPE----"
+ "\n"
+ "\n1.CHICKEN CONFIT"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 4 chicken leg portions with thighs attached, excess fat trimmed and
reserved (about 2 pounds total)"
+ "\n• 1 tablespoon plus 1/8 teaspoon kosher salt"
+ "\n• 1/2 teaspoon freshly ground black pepper"
+ "\n• 10 garlic cloves"
+ "\n• 4 bay leaves"
+ "\n• 4 sprigs fresh thyme"
+ "\n• 1 1/2 teaspoons black peppercorns"
+ "\n• 1/2 teaspoon table salt"
+ "\n• 4 cups olive oil"
+ "\n"
+ "\nHow to cook:"
+ "\nStep 1: Lay the leg portions on a platter, skin side down. Sprinkle with
1 tablespoon of the kosher salt and black pepper. Place the garlic cloves, bay leaves,
and sprigs of thyme on each of 2 leg portions. Lay the remaining 2 leg portions, flesh to
flesh, on top. Put the reserved fat from the chicken in the bottom of a glass or plastic
container. Top with the sandwiched leg portions. Sprinkle with the remaining 1/8
teaspoon kosher salt. Cover and refrigerate for 12 hours."
+ "\n"
+ "\nStep 2: Preheat the oven to 200 degrees F."
+ "\n"
+ "\nStep 3: Remove the chicken from the refrigerator. Remove the garlic,
bay leaves, thyme, and chicken fat and reserve. Rinse the chicken with cool water,
rubbing off some of the salt and pepper. Pat dry with paper towels."
+ "\n"
+ "\nStep 4: Put the reserved garlic, bay leaves, thyme, and chicken fat in
the bottom of an enameled cast iron pot. Sprinkle evenly with the peppercorns and salt.
Lay the chicken on top, skin side down. Add the olive oil. Cover and bake for 12 to 14
hours, or until the meat pulls away from the bone."
+ "\n"
```



+ "\nStep 5: Remove the chicken from the fat. Strain the fat and reserve.

Pick the meat from the bones and place it in a stoneware container. Cover the meat with some of the strained fat so that there is a 1/4-inch layer of fat on top. The chicken confit can be stored in the refrigerator for up to 1 month."

+ "\n"

+ "\nStep 6: The excess oil can be stored in an airtight container in the refrigerator and used like butter for cooking. The tinge of chicken taste in the oil is wonderful and I use the oil to roast potatoes, cook green beans, and pan-fry veal."

+ "\n"

+ "\n"

+ "\n2.LAMB SHANK NAVARIN"

+ "\n"

+ "\nIngredients:"

+ "\n"

+ "\n• 2 tbsp plain flour"

+ "\n• 6 (about 2kg) lamb shanks, French trimmed"

+ "\n• 2 tbsp olive oil"

+ "\n• 1 brown onion, halved, thinly sliced"

+ "\n• 2 garlic cloves, finely chopped"

+ "\n• 80ml (1/3 cup) white wine"

+ "\n• 500ml (2 cups) Massel chicken style liquid stock"

+ "\n• 500ml (2 cups) passata (tomato pasta sauce)"

+ "\n• 3 small sprigs fresh rosemary"

+ "\n• 2 bay leaves"

+ "\n• 6 baby Coliban potatoes, halved (see note)"

+ "\n• 1 turnip, peeled, cut into wedges"

+ "\n• 1 bunch baby (Dutch) carrots, trimmed, peeled"

+ "\n• 150g (1 cup) frozen peas"

+ "\n• 1/3 cup chopped fresh continental parsley"

+ "\n"

+ "\nHow to cook:"

+ "\n"

+ "\nStep 1: Place the flour on a plate. Season with salt and pepper. Toss the lamb shanks in the flour mixture to coat. Shake off excess. Heat half the oil in a large flameproof casserole dish over medium-high heat. Cook half the lamb, turning occasionally, for 7 minutes or until browned. Transfer to a plate. Repeat with the remaining lamb and oil, reheating the pan between batches."

+ "\n"

+ "\nStep 2: Add the onion and garlic to the dish. Cook, stirring, for 3 minutes or until soft. Add the wine and cook for 1 minute, scraping the pan with a flat-edged wooden spoon to dislodge any bits that have cooked onto the base. Stir in the stock, passata, rosemary and bay leaves. Return the lamb to the dish."

+ "\n"

+ "\nStep 3: Cover and reduce heat to low. Cook, turning lamb halfway through cooking, for 1 hour 45 minutes. Use a metal spoon to skim fat from surface. Add potato and turnip. Cook for 35 minutes or until lamb is tender. Use a slotted spoon to transfer lamb, potato and turnip to a large plate. Add carrots to the dish. Increase heat to medium. Cook, uncovered, stirring occasionally, for 20 minutes or until carrots are tender and sauce has reduced."

+ "\n"

+ "\nStep 4: Add the peas to the carrot mixture. Return the lamb, potato and turnip to the dish. Cook, turning the lamb in the sauce occasionally, for 5 minutes or until heated through. Stir in the parsley. Season with salt and pepper."

+ "\n"

+ "\nMORE FRENCH FOOD TO SERVE SOON!");

menuscroll = new JScrollPane(menuText);



```

menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling

framemenu.add(menuscroll); // Add scrollable text area to the frame
framemenu.add(BACK);

framemenu.setBounds(660, 0, 600, 1020);
framemenu.setResizable(false);
framemenu.setLayout(null);
framemenu.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new CTLB();
        framemenu.dispose();
    }
}
}

class MENUC implements ActionListener{
    private JFrame framemenu;
    private JButton BACK;
    private JTextArea menutext;
    private JScrollPane menuscroll;

    public MENUC(){
        framemenu = new JFrame();
        framemenu.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/ItalyBG.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemenu.setIconImage(logo);
        framemenu.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10, 10, 70, 70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        menutext = new JTextArea();
        menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
        menutext.setEditable(false); // Set as non-editable
        menutext.setOpaque(true); // Make it opaque to show background color
    }
}

```



```
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("This is some sample text.\nYou can add more text here.");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4500)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(255, 201, 111)); // Adjust RGB values as
needed rgb(255, 201, 111)
menutext.setText(" ----ITALIAN FOOD RECIPE----"
+ "\n"
+ "\n1.LASAGNA"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 1 1/2 lb. ground beef"
+ "\n• 1 lb. hot breakfast sausage"
+ "\n• 2 cloves garlic, minced"
+ "\n• 2 cans (14.5 ounce) whole tomatoes"
+ "\n• 2 cans (6 ounce) tomato paste"
+ "\n• 4 Tbsp. dried parsley, divided"
+ "\n• 2 Tbsp. dried basil"
+ "\n• 2 1/2 tsp. salt"
+ "\n• 3 c. lowfat cottage cheese"
+ "\n• 2 whole beaten eggs"
+ "\n• 1/2 c. grated (not shredded) Parmesan cheese"
+ "\n• 1 Tbsp. olive oil"
+ "\n• 1 package (10 ounce) lasagna noodles"
+ "\n• 1 lb. sliced mozzarella cheese"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Bring a large pot of water to a boil."
+ "\n"
+ "\nStep 2: Meanwhile, in a large skillet or saucepan, combine ground
beef, sausage, and garlic. Cook over medium-high heat until browned. Drain half the fat;
less if you're feeling naughty. Add tomatoes, tomato paste, 2 tablespoons parsley, basil
and 1 teaspoon salt. After adding the tomatoes, the sauce mixture should simmer for 45
minutes while you are working on the other steps."
+ "\n"
+ "\nStep 3: In a medium bowl, mix cottage cheese, beaten eggs, grated
Parmesan, 2 more tablespoons parsley, and 1 more teaspoon salt. Stir together well.
Set aside. Add 1/2 teaspoon salt and the olive oil to the boiling water, then cook the
lasagna noodles until "al dente" (not overly cooked). Drain."
+ "\n"
+ "\nStep 4: To assemble: Arrange 4 cooked lasagna noodles in the
bottom of a baking pan, overlapping if necessary. Spoon half the cottage cheese
mixture over the noodles. Spread evenly. Cover cottage cheese with a layer of
mozzarella cheese. Spoon a little less than half the meat/sauce mixture over the top.
Repeat, ending with meat/sauce mixture. Sprinkle top generously with extra Parmesan."
+ "\n"
+ "\nStep 5: Either freeze, refrigerate for up to two days, or bake
immediately: 350-degree oven for 20 to 30 minutes, or until top is hot and bubbly."
+ "\n"
+ "\n2.MARINATED LAMB LOIN CHOPS"
+ "\n"
+ "\nIngredients:"
```



```
+ "\n"
+ "\n• 4 Large Boneless Lamb Loin Chops"
+ "\n• 1/4 Cup Olive Oil"
+ "\n• 1 Tablespoon Fresh Thyme, Chopped"
+ "\n• 1 Tablespoon Fresh Parsley, Chopped"
+ "\n• 2 Teaspoons Fresh Mint, Chopped"
+ "\n• 2 Cloves Garlic, Minced"
+ "\n• Zest of 1 Lemon"
+ "\n• 1 Teaspoon Salt"
+ "\n• 1/4 Teaspoon Black Pepper"
+ "\n"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Mix together all of the marinade ingredients and place in a shallow casserole dish along with the lamb ensuring the marinade coats the lamb well."
+ "\n"
+ "\nStep 2: Refrigerate for at least two hours or up till 6 hours."
+ "\n"
+ "\nStep 3: Remove from the refrigerator 45 minutes before you are ready to cook to allow the meat to come to room temperature."
+ "\n"
+ "\nStep 4: Preheat the oven to 400 degrees F."
+ "\n"
+ "\nStep 5: Heat a heavy oven-proof frying pan at medium high heat."
+ "\n"
+ "\nStep 6: Sear the lamb very well one both sides until golden brown."
+ "\n"
+ "\nStep 7: Place the pan in the hot oven until the lamb is cooked to your desired doneness."
+ "\n"
+ "\nStep 8: We find 5 to 6 minutes will cook our 2 inch lamb chops to a nice warm rare."
+ "\n"
+ "\nStep 9: Serve hot."
+ "\n"
+ "\nMORE ITALIAN FOOD TO SERVE SOON!"
);
```

```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
frameMenu.add(menuscroll); // Add scrollable text area to the frame
frameMenu.add(BACK);
```

```
frameMenu.setBounds(660, 0, 600, 1020);
frameMenu.setResizable(false);
frameMenu.setLayout(null);
frameMenu.setVisible(true);
```

```
// Set the caret position to the beginning
menuText.setCaretPosition(0);
}
```

```
public void actionPerformed(ActionEvent e) {
```



```

JButton choice = (JButton)e.getSource();
String choices = choice.getName();
if (choices == "back"){
    new CTL();
    framemenu.dispose();
}
}

class MENUC_B implements ActionListener{
    private JFrame framemenu;
    private JButton BACK;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public MENUC_B(){
        framemenu = new JFrame();
        framemenu.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/ItalyBG.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemenu.setIconImage(logo);
        framemenu.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        minutext = new JTextArea();
        minutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
        minutext.setEditable(false); // Set as non-editable
        minutext.setOpaque(true); // Make it opaque to show background color
        minutext.setForeground(Color.BLACK); // Set text color
        minutext.setText("This is some sample text.\nYou can add more text here.");
        minutext.setLineWrap(true);
        minutext.setWrapStyleWord(true);
        minutext.setPreferredSize(new Dimension(500, 4500)); // Set preferred size
        // Set background color using RGB values
        minutext.setBackground(new Color(255, 201, 111)); // Adjust RGB values as
needed rgb(255, 201, 111)
        minutext.setText(" ----ITALIAN FOOD RECIPE----"
+ "\n "
+ "\n1.LASAGNA"
+ "\n "
+ "\nIngredients:"
+ "\n "
+ "\n• 1 1/2 lb. ground beef"
+ "\n• 1 lb. hot breakfast sausage"

```



+ "\n• 2 cloves garlic, minced"
+ "\n• 2 cans (14.5 ounce) whole tomatoes"
+ "\n• 2 cans (6 ounce) tomato paste"
+ "\n• 4 Tbsp. dried parsley, divided"
+ "\n• 2 Tbsp. dried basil"
+ "\n• 2 1/2 tsp. salt"
+ "\n• 3 c. lowfat cottage cheese"
+ "\n• 2 whole beaten eggs"
+ "\n• 1/2 c. grated (not shredded) Parmesan cheese"
+ "\n• 1 Tbsp. olive oil"
+ "\n• 1 package (10 ounce) lasagna noodles"
+ "\n• 1 lb. sliced mozzarella cheese"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Bring a large pot of water to a boil."
+ "\n"
+ "\nStep 2: Meanwhile, in a large skillet or saucepan, combine ground beef, sausage, and garlic. Cook over medium-high heat until browned. Drain half the fat; less if you're feeling naughty. Add tomatoes, tomato paste, 2 tablespoons parsley, basil and 1 teaspoon salt. After adding the tomatoes, the sauce mixture should simmer for 45 minutes while you are working on the other steps."
+ "\n"
+ "\nStep 3: In a medium bowl, mix cottage cheese, beaten eggs, grated Parmesan, 2 more tablespoons parsley, and 1 more teaspoon salt. Stir together well. Set aside. Add 1/2 teaspoon salt and the olive oil to the boiling water, then cook the lasagna noodles until "al dente" (not overly cooked). Drain."
+ "\n"
+ "\nStep 4: To assemble: Arrange 4 cooked lasagna noodles in the bottom of a baking pan, overlapping if necessary. Spoon half the cottage cheese mixture over the noodles. Spread evenly. Cover cottage cheese with a layer of mozzarella cheese. Spoon a little less than half the meat/sauce mixture over the top. Repeat, ending with meat/sauce mixture. Sprinkle top generously with extra Parmesan."
+ "\n"
+ "\nStep 5: Either freeze, refrigerate for up to two days, or bake immediately: 350-degree oven for 20 to 30 minutes, or until top is hot and bubbly."
+ "\n"
+ "\n"
+ "\n2.MARINATED LAMB LOIN CHOPS"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 4 Large Boneless Lamb Loin Chops"
+ "\n• 1/4 Cup Olive Oil"
+ "\n• 1 Tablespoon Fresh Thyme, Chopped"
+ "\n• 1 Tablespoon Fresh Parsley, Chopped"
+ "\n• 2 Teaspoons Fresh Mint, Chopped"
+ "\n• 2 Cloves Garlic, Minced"
+ "\n• Zest of 1 Lemon"
+ "\n• 1 Teaspoon Salt"
+ "\n• 1/4 Teaspoon Black Pepper"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Mix together all of the marinade ingredients and place in a shallow casserole dish along with the lamb ensuring the marinade coats the lamb well."
+ "\n"



+ "\nStep 2: Refrigerate for at least two hours or up till 6 hours."
 + "\n"
 + "\nStep 3: Remove from the refrigerator 45 minutes before you are
 ready to cook to allow the meat to come to room temperature."
 + "\n"
 + "\nStep 4: Preheat the oven to 400 degrees F."
 + "\n"
 + "\nStep 5: Heat a heavy oven-proof frying pan at medium high heat."
 + "\n"
 + "\nStep 6: Sear the lamb very well one both sides until golden brown."
 + "\n"
 + "\nStep 7: Place the pan in the hot oven until the lamb is cooked to your
 desired doneness."
 + "\n"
 + "\nStep 8: We find 5 to 6 minutes will cook our 2 inch lamb chops to a
 nice warm rare."
 + "\n"
 + "\nStep 9: Serve hot."
 + "\n"
 + "\nMORE ITALIAN FOOD TO SERVE SOON!"
);

```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
frameMenu.add(menuscroll); // Add scrollable text area to the frame
frameMenu.add(back);
```

```
frameMenu.setBounds(660, 0, 600, 1020);
frameMenu.setResizable(false);
frameMenu.setLayout(null);
frameMenu.setVisible(true);
```

```
// Set the caret position to the beginning
menuText.setCaretPosition(0);
}
```

```
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new CTLB();
        frameMenu.dispose();
    }
}
```

```
class MENU implements ActionListener{
    private JFrame frameMenu;
    private JButton back;
    private JTextArea menuText;
    private JScrollPane menuscroll;
```

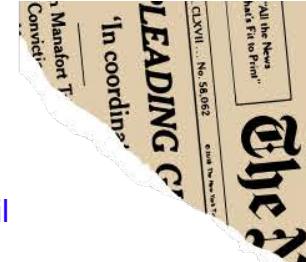
```

public MENUD(){
    framemenu = new JFrame();
    framemenu.setContentPane(new JLabel(new
    ImageIcon(getClass().getResource("/JapanBG.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    framemenu.setIconImage(logo);
    framemenu.setTitle("CHEF'S CACHE");

    BACK = new JButton();
    BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
    BACK.setOpaque(false);
    BACK.setContentAreaFilled(false);
    BACK.setBounds(10,10,70,70);
    BACK.setBorderPainted(false);
    ImageIcon backA = new ImageIcon();
    BACK.setPressedIcon(backA);
    BACK.addActionListener(this);
    BACK.setName("back");

    menutext = new JTextArea();
    menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
    menutext.setEditable(false); // Set as non-editable
    menutext.setOpaque(true); // Make it opaque to show background color
    menutext.setForeground(Color.BLACK); // Set text color
    menutext.setText("This is some sample text.\nYou can add more text here.");
    menutext.setLineWrap(true);
    menutext.setWrapStyleWord(true);
    menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
    // Set background color using RGB values
    menutext.setBackground(new Color(240, 235, 227)); // Adjust RGB values as
needed rgb(240, 235, 227)
    menutext.setText(" ----JAPANESE FOOD RECIPE----"
+ "\n"
+ "\n1.CRISPY SHRIMP TEMPURA"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 2 cups vegetable oil for frying"
+ "\n• 1 cup all-purpose flour"
+ "\n• 2 tablespoons cornstarch"
+ "\n• 1 pinch salt"
+ "\n• 1 cup water"
+ "\n• 1 large egg yolk"
+ "\n• 2 large egg whites, lightly beaten"
+ "\n• 1 pound medium shrimp, peeled and deveined, tails left on"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Gather all ingredients."
+ "\n"
+ "\nStep 2: Heat oil in a deep-fryer to 375 degrees F (190 degrees C)."
+ "\n"
+ "\nStep 3: Whisk flour, cornstarch, and salt together in a large bowl;
make a well in the center."
+ "\n"

```



+ "\nStep 4: Pour water and egg yolk into the well and mix just until moistened; batter will be lumpy. Stir in egg whites."

+ "\n"

+ "\nStep 5: Dip one shrimp at a time into batter to coat; do not batter tails."

+ "\n"

+ "\nStep 6: When three shrimp have been battered, carefully place them into the deep fryer and fry until golden brown, about 1 1/2 minutes."

+ "\n"

+ "\nStep 7: Remove with a slotted spoon and drain on a paper towel-lined plate. Repeat with remaining shrimp, battering a few at a time while the previous batch is cooking."

+ "\n"

+ "\n"

+ "\n2.YAKITORI (GRILLED CHICKEN SKEWERS)"

+ "\n"

+ "\nIngredients:"

+ "\n"

+ "\n• 1 pound chicken livers, gizzards or boneless thigh meat"

+ "\n• 1/2 cup dark soy sauce or tamari"

+ "\n• 1/4 cup mirin"

+ "\n• 2 tablespoons sake or dry sherry"

+ "\n• 1 tablespoon brown sugar"

+ "\n• 2 garlic cloves, peeled and smashed"

+ "\n• 1/2 teaspoon grated fresh ginger"

+ "\n• Scallions, thinly sliced, for garnish"

+ "\n"

+ "\nHow to cook:"

+ "\n"

+ "\nStep 1: Cut chicken into one-inch pieces and place in a shallow dish."

+ "\n"

+ "\nStep 2: In a small saucepan, combine soy sauce or tamari, mirin, sake or sherry, brown sugar, garlic and ginger. Bring to a simmer and cook for 7 minutes, until thickened. Reserve 2 tablespoons sauce for serving. Pour remaining sauce over chicken, cover, and chill for at least one hour (and up to 4 hours)."

+ "\n"

+ "\nStep 3: If using wooden or bamboo skewers, soak them in water for one hour. Preheat grill or broiler. Thread chicken pieces onto skewers, and grill or broil, turning halfway, for about 3 minutes for livers, 10 minutes for gizzards and 6 minutes for thighs. Serve drizzled with reserved sauce and garnished with scallions."

+ "\n"

+ "\nMORE JAPANESE FOOD TO SERVE SOON!");

```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
framemenu.add(menuscroll); // Add scrollable text area to the frame
framemenu.add(BACK);
```

```
framemenu.setBounds(660, 0, 600, 1020);
framemenu.setResizable(false);
framemenu.setLayout(null);
framemenu.setVisible(true);
```



```

// Set the caret position to the beginning
menutext.setCaretPosition(0);
}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new CTL();
        framemenu.dispose();
    }
}
}

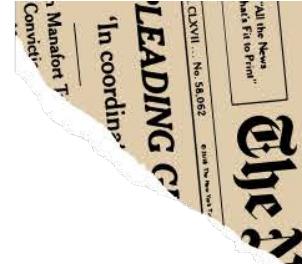
class MENUD_B implements ActionListener{
    private JFrame framemenu;
    private JButton BACK;
    private JTextArea menutext;
    private JScrollPane menuscroll;

    public MENUD_B(){
        framemenu = new JFrame();
        framemenu.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/JapanBG.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemenu.setIconImage(logo);
        framemenu.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        menutext = new JTextArea();
        menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
        menutext.setEditable(false); // Set as non-editable
        menutext.setOpaque(true); // Make it opaque to show background color
        menutext.setForeground(Color.BLACK); // Set text color
        menutext.setText("This is some sample text.\nYou can add more text here.");
        menutext.setLineWrap(true);
        menutext.setWrapStyleWord(true);
        menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
        // Set background color using RGB values
        menutext.setBackground(new Color(240, 235, 227)); // Adjust RGB values as
needed rgb(240, 235, 227)
        menutext.setText(" ----JAPANESE FOOD RECIPE----"
+ "\n"
+ "\n1.CRISPY SHRIMP TEMPURA"

```



+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 2 cups vegetable oil for frying"
+ "\n• 1 cup all-purpose flour"
+ "\n• 2 tablespoons cornstarch"
+ "\n• 1 pinch salt"
+ "\n• 1 cup water"
+ "\n• 1 large egg yolk"
+ "\n• 2 large egg whites, lightly beaten"
+ "\n• 1 pound medium shrimp, peeled and deveined, tails left on"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Gather all ingredients."
+ "\n"
+ "\nStep 2: Heat oil in a deep-fryer to 375 degrees F (190 degrees C)."
+ "\n"
+ "\nStep 3: Whisk flour, cornstarch, and salt together in a large bowl; make a well in the center."
+ "\n"
+ "\nStep 4: Pour water and egg yolk into the well and mix just until moistened; batter will be lumpy. Stir in egg whites."
+ "\n"
+ "\nStep 5: Dip one shrimp at a time into batter to coat; do not batter tails."
+ "\n"
+ "\nStep 6: When three shrimp have been battered, carefully place them into the deep fryer and fry until golden brown, about 1 1/2 minutes."
+ "\n"
+ "\nStep 7: Remove with a slotted spoon and drain on a paper towel-lined plate. Repeat with remaining shrimp, battering a few at a time while the previous batch is cooking."
+ "\n"
+ "\n"
+ "\n2.YAKITORI (GRILLED CHICKEN SKEWERS)"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 1 pound chicken livers, gizzards or boneless thigh meat"
+ "\n• 1/2 cup dark soy sauce or tamari"
+ "\n• 1/4 cup mirin"
+ "\n• 2 tablespoons sake or dry sherry"
+ "\n• 1 tablespoon brown sugar"
+ "\n• 2 garlic cloves, peeled and smashed"
+ "\n• 1/2 teaspoon grated fresh ginger"
+ "\n• Scallions, thinly sliced, for garnish"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Cut chicken into one-inch pieces and place in a shallow dish."
+ "\n"
+ "\nStep 2: In a small saucepan, combine soy sauce or tamari, mirin, sake or sherry, brown sugar, garlic and ginger. Bring to a simmer and cook for 7 minutes, until thickened. Reserve 2 tablespoons sauce for serving. Pour remaining sauce over chicken, cover, and chill for at least one hour (and up to 4 hours)."
+ "\n"



+ "\nStep 3: If using wooden or bamboo skewers, soak them in water for one hour. Preheat grill or broiler. Thread chicken pieces onto skewers, and grill or broil, turning halfway, for about 3 minutes for livers, 10 minutes for gizzards and 6 minutes for thighs. Serve drizzled with reserved sauce and garnished with scallions."

+ "\n"
+ "\nMORE JAPANES FOOD TO SERVE SOON!");

```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
frameMenu.add(menuscroll); // Add scrollable text area to the frame
frameMenu.add(back);
```

```
frameMenu.setBounds(660, 0, 600, 1020);
frameMenu.setResizable(false);
frameMenu.setLayout(null);
frameMenu.setVisible(true);
```

```
// Set the caret position to the beginning
menuText.setCaretPosition(0);
}
```

```
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new CTLB();
        frameMenu.dispose();
    }
}
```

```
}
```

```
class MENU implements ActionListener{
    private JFrame frameMenu;
    private JButton back;
    private JTextArea menuText;
    private JScrollPane menuscroll;
```

```
public MENU(){
    frameMenu = new JFrame();
    frameMenu.setContentPane(new JLabel(new ImageIcon(getClass().getResource("/KoreaBG.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    frameMenu.setIconImage(logo);
    frameMenu.setTitle("CHEF'S CACHE");
```

```
BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
```



```
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

menutext = new JTextArea();
menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("This is some sample text.\nYou can add more text here.");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 6000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(255, 238, 169)); // Adjust RGB values as
needed rgb(255, 238, 169)
menutext.setText(" ----KOREAN FOOD RECIPE----"
+ "\n"
+ "\n1.KIMCHI"
+ "\n"
+ "\nIngredients:"
+ "\n"
+ "\n• 1 medium head napa cabbage (about 2 pounds)"
+ "\n• 1/4 cup iodine-free sea salt or kosher salt (see Recipe Notes)"
+ "\n• Water, preferably distilled or filtered"
+ "\n• 1 tablespoon grated garlic (5 to 6 cloves)"
+ "\n• 1 teaspoon grated peeled fresh ginger"
+ "\n• 1 teaspoon granulated sugar"
+ "\n• 2 tablespoons fish sauce or salted shrimp paste, or 3 tablespoons
water"
+ "\n• 1 to 5 tablespoons Korean red pepper flakes (gochugaru)"
+ "\n• 8 ounces Korean radish or daikon radish, peeled and cut into
matchsticks"
+ "\n• 4 medium scallions, trimmed and cut into 1-inch pieces"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nStep 1: Cut the cabbage. Cut the cabbage lengthwise through the
stem into quarters. Cut the cores from each piece. Cut each quarter crosswise into 2-
inch-wide strips."
+ "\n"
+ "\nStep 2: Salt the cabbage. Place the cabbage in a large bowl and
sprinkle with the salt. Using your hands, massage the salt into the cabbage until it starts
to soften a bit. Add enough water to cover the cabbage. Put a plate on top of the
cabbage and weigh it down with something heavy, like a jar or can of beans. Let stand
for 1 to 2 hours."
+ "\n"
+ "\nStep 3: Rinse and drain the cabbage. Rinse the cabbage under cold
water 3 times. Set aside to drain in a colander for 15 to 20 minutes. Meanwhile, make
the spice paste."
+ "\n"
+ "\nStep 4: Make the spice paste. Rinse and dry the bowl you used for
salting. Add the garlic, ginger, sugar, and fish sauce, shrimp paste, or water and stir into
a smooth paste. Stir in the gochugaru, using 1 tablespoon for mild and up to 5
```



tablespoons for spicy (I like about 3 1/2 tablespoons); set aside until the cabbage is ready."

+ "\n"

+ "\nStep 5: Combine the vegetables and spice paste. Gently squeeze any remaining water from the cabbage and add it to the spice paste. Add the radish and scallions."

+ "\n"

+ "\nStep 6: Mix thoroughly. Using your hands, gently work the paste into the vegetables until they are thoroughly coated. The gloves are optional here but highly recommended to protect your hands from stings, stains, and smells!"

+ "\n"

+ "\nStep 7: Pack the kimchi into the jar. Pack the kimchi into a 1-quart jar. Press down on the kimchi until the brine (the liquid that comes out) rises to cover the vegetables, leaving at least 1 inch of space at the top. Seal the jar."

+ "\n"

+ "\nStep 8: Let it ferment for 1 to 5 days. Place a bowl or plate under the jar to help catch any overflow. Let the jar stand at cool room temperature, out of direct sunlight, for 1 to 5 days. You may see bubbles inside the jar and brine may seep out of the lid."

+ "\n"

+ "\nStep 9: Check it daily and refrigerate when ready. Check the kimchi once a day, opening the jar and pressing down on the vegetables with a clean finger or spoon to keep them submerged under the brine. (This also releases gases produced during fermentation.) Taste a little at this point, too! When the kimchi tastes ripe enough for your liking, transfer the jar to the refrigerator. You may eat it right away, but it's best after another week or two."

+ "\n"

+ "\n"

+ "\n2.KIMCHI STEW"

+ "\n"

+ "\nIngredients:"

+ "\n"

+ "\n• 1 pound kimchi, cut into bite size pieces"

+ "\n• 1/4cup kimchi brine"

+ "\n• 1/2 pound pork shoulder (or pork belly)"

+ "\n• 1/2 package of tofu (optional), sliced into ½ inch thick bite size pieces"

+ "\n• 3 green onions"

+ "\n• 1 medium onion, sliced (1 cup)"

+ "\n• 1 teaspoon kosher salt"

+ "\n• 2 teaspoons sugar"

+ "\n• 2 teaspoons gochugaru (Korean hot pepper flakes)"

+ "\n• 1 tablespoon gochujang (hot pepper paste)"

+ "\n• 1 teaspoon toasted sesame oil"

+ "\n• 2 cups of anchovy stock (or chicken or beef broth)"

+ "\n"

+ "\nFor stock (makes about 2½ cups' worth)"

+ "\n7 large dried anchovies, heads and guts removed"

+ "\n• 1/3 cup Korean radish (or daikon radish), sliced thinly"

+ "\n• 4 by 5 inch dried kelp"

+ "\n• 3 green onion roots"

+ "\n• 4 cups water"

+ "\n"

+ "\nHow to cook:"

+ "\n"

+ "\nMake anchovy stock"



+ "\nStep 1: Put the anchovies, daikon, green onion roots, and dried kelp in a sauce pan."
 + "\nStep 2: Add the water and boil for 20 minutes over medium high heat."
 + "\nStep 3: Lower the heat to low for another 5 minutes."
 + "\nStep 4: Strain"
 + "\n"
 + "\nMake kimchi stew"
 + "\nStep 1: Place the kimchi and kimchi brine in a shallow pot. Add pork and onion"
 + "\nStep 2: Slice 2 green onions diagonally and add them to the pot."
 + "\nStep 3: Add salt, sugar, hot pepper flakes, and hot pepper paste.
 Drizzle sesame oil over top and add the anchovy stock"
 + "\nStep 4: Cover and cook for 10 minutes over medium high heat."
 + "\nStep 5: Open and mix in the seasonings with a spoon. Lay the tofu over top."
 + "\nStep 6: Cover and cook another 10 to 15 minutes over medium heat."
 + "\nStep 7: Chop 1 green onion and put it on the top of the stew. Remove from the heat and serve right away with rice."
 + "\n"
 + "\nMORE KOREAN FOOD TO SERVE SOON!"
);

```

menuscroll = new JScrollPane(minutext);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling

framemenu.add(menuscroll); // Add scrollable text area to the frame
framemenu.add(BACK);

framemenu.setBounds(660,0,600,1020);
framemenu.setResizable(false);
framemenu.setLayout(null);
framemenu.setVisible(true);

// Set the caret position to the beginning
minutext.setCaretPosition(0);
}

public void actionPerformed(ActionEvent e) {
  JButton choice = (JButton)e.getSource();
  String choices = choice.getName();
  if (choices == "back"){
    new CTL();
    framemenu.dispose();
  }
}
}
  
```

```

class MENU_B implements ActionListener{
  private JFrame framemenu;
  private JButton BACK;
  private JTextArea minutext;
  private JScrollPane menuscroll;
  
```

```

public MENU_B(){
    framemenu = new JFrame();
    framemenu.setContentPane(new JLabel(new
    ImageIcon(getClass().getResource("/KoreaBG.png"))));
    Image logo =
    Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    framemenu.setIconImage(logo);
    framemenu.setTitle("CHEF'S CACHE");

    BACK = new JButton();
    BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
    BACK.setOpaque(false);
    BACK.setContentAreaFilled(false);
    BACK.setBounds(10,10,70,70);
    BACK.setBorderPainted(false);
    ImageIcon backA = new ImageIcon();
    BACK.setPressedIcon(backA);
    BACK.addActionListener(this);
    BACK.setName("back");

    menutext = new JTextArea();
    menutext.setFont(new Font("Times New Roman", Font.PLAIN, 30)); // Set larger
font size
    menutext.setEditable(false); // Set as non-editable
    menutext.setOpaque(true); // Make it opaque to show background color
    menutext.setForeground(Color.BLACK); // Set text color
    menutext.setText("This is some sample text.\nYou can add more text here.");
    menutext.setLineWrap(true);
    menutext.setWrapStyleWord(true);
    menutext.setPreferredSize(new Dimension(500, 6000)); // Set preferred size
    // Set background color using RGB values
    menutext.setBackground(new Color(255, 238, 169)); // Adjust RGB values as
needed rgb(255, 238, 169)
    menutext.setText(" ----KOREAN FOOD RECIPE----"
        + "\n"
        + "\n1.KIMCHI"
        + "\n"
        + "\nIngredients:"
        + "\n"
        + "\n• 1 medium head napa cabbage (about 2 pounds)"
        + "\n• 1/4 cup iodine-free sea salt or kosher salt (see Recipe Notes)"
        + "\n• Water, preferably distilled or filtered"
        + "\n• 1 tablespoon grated garlic (5 to 6 cloves)"
        + "\n• 1 teaspoon grated peeled fresh ginger"
        + "\n• 1 teaspoon granulated sugar"
        + "\n• 2 tablespoons fish sauce or salted shrimp paste, or 3 tablespoons
water"
        + "\n• 1 to 5 tablespoons Korean red pepper flakes (gochugaru)"
        + "\n• 8 ounces Korean radish or daikon radish, peeled and cut into
matchsticks"
        + "\n• 4 medium scallions, trimmed and cut into 1-inch pieces"
        + "\n"
        + "\nHow to cook:"
        + "\n"
)
}

```



- + "\nStep 1: Cut the cabbage. Cut the cabbage lengthwise through the stem into quarters. Cut the cores from each piece. Cut each quarter crosswise into 2-inch-wide strips."
- + "\n"
- + "\nStep 2: Salt the cabbage. Place the cabbage in a large bowl and sprinkle with the salt. Using your hands, massage the salt into the cabbage until it starts to soften a bit. Add enough water to cover the cabbage. Put a plate on top of the cabbage and weigh it down with something heavy, like a jar or can of beans. Let stand for 1 to 2 hours."
- + "\n"
- + "\nStep 3: Rinse and drain the cabbage. Rinse the cabbage under cold water 3 times. Set aside to drain in a colander for 15 to 20 minutes. Meanwhile, make the spice paste."
- + "\n"
- + "\nStep 4: Make the spice paste. Rinse and dry the bowl you used for salting. Add the garlic, ginger, sugar, and fish sauce, shrimp paste, or water and stir into a smooth paste. Stir in the gochugaru, using 1 tablespoon for mild and up to 5 tablespoons for spicy (I like about 3 1/2 tablespoons); set aside until the cabbage is ready."
- + "\n"
- + "\nStep 5: Combine the vegetables and spice paste. Gently squeeze any remaining water from the cabbage and add it to the spice paste. Add the radish and scallions."
- + "\n"
- + "\nStep 6: Mix thoroughly. Using your hands, gently work the paste into the vegetables until they are thoroughly coated. The gloves are optional here but highly recommended to protect your hands from stings, stains, and smells!"
- + "\n"
- + "\nStep 7: Pack the kimchi into the jar. Pack the kimchi into a 1-quart jar. Press down on the kimchi until the brine (the liquid that comes out) rises to cover the vegetables, leaving at least 1 inch of space at the top. Seal the jar."
- + "\n"
- + "\nStep 8: Let it ferment for 1 to 5 days. Place a bowl or plate under the jar to help catch any overflow. Let the jar stand at cool room temperature, out of direct sunlight, for 1 to 5 days. You may see bubbles inside the jar and brine may seep out of the lid."
- + "\n"
- + "\nStep 9: Check it daily and refrigerate when ready. Check the kimchi once a day, opening the jar and pressing down on the vegetables with a clean finger or spoon to keep them submerged under the brine. (This also releases gases produced during fermentation.) Taste a little at this point, too! When the kimchi tastes ripe enough for your liking, transfer the jar to the refrigerator. You may eat it right away, but it's best after another week or two."
- + "\n"
- + "\n"
- + "\n2.KIMCHI STEW"
- + "\n"
- + "\nIngredients:"
- + "\n"
- + "\n• 1 pound kimchi, cut into bite size pieces"
- + "\n• 1/4cup kimchi brine"
- + "\n• 1/2 pound pork shoulder (or pork belly)"
- + "\n• 1/2 package of tofu (optional), sliced into 1/2 inch thick bite size pieces"
- + "\n• 3 green onions"
- + "\n• 1 medium onion, sliced (1 cup)"
- + "\n• 1 teaspoon kosher salt"





+ "\n• 2 teaspoons sugar"
+ "\n• 2 teaspoons gochugaru (Korean hot pepper flakes)"
+ "\n• 1 tablespoon gochujang (hot pepper paste)"
+ "\n• 1 teaspoon toasted sesame oil"
+ "\n• 2 cups of anchovy stock (or chicken or beef broth)"
+ "\n"
+ "\nFor stock (makes about 2½ cups' worth)"
+ "\n7 large dried anchovies, heads and guts removed"
+ "\n• 1/3 cup Korean radish (or daikon radish), sliced thinly"
+ "\n• 4 by 5 inch dried kelp"
+ "\n• 3 green onion roots"
+ "\n• 4 cups water"
+ "\n"
+ "\nHow to cook:"
+ "\n"
+ "\nMake anchovy stock"
+ "\nStep 1: Put the anchovies, daikon, green onion roots, and dried kelp in a sauce pan."
+ "\nStep 2: Add the water and boil for 20 minutes over medium high heat."
+ "\nStep 3: Lower the heat to low for another 5 minutes."
+ "\nStep 4: Strain"
+ "\n"
+ "\nMake kimchi stew"
+ "\nStep 1: Place the kimchi and kimchi brine in a shallow pot. Add pork and onion"
+ "\nStep 2: Slice 2 green onions diagonally and add them to the pot."
+ "\nStep 3: Add salt, sugar, hot pepper flakes, and hot pepper paste.
Drizzle sesame oil over top and add the anchovy stock"
+ "\nStep 4: Cover and cook for 10 minutes over medium high heat."
+ "\nStep 5: Open and mix in the seasonings with a spoon. Lay the tofu over top."
+ "\nStep 6: Cover and cook another 10 to 15 minutes over medium heat."
+ "\nStep 7: Chop 1 green onion and put it on the top of the stew. Remove from the heat and serve right away with rice."
+ "\n"
+ "\nMORE KOREAN FOOD TO SERVE SOON!"
);

```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 150, 500, 750); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
frameMenu.add(menuscroll); // Add scrollable text area to the frame
frameMenu.add(BACK);
```

```
frameMenu.setBounds(660, 0, 600, 1020);
frameMenu.setResizable(false);
frameMenu.setLayout(null);
frameMenu.setVisible(true);
```

```
// Set the caret position to the beginning
menuText.setCaretPosition(0);
```

```
}
```



```
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton)e.getSource();  
    String choices = choice.getName();  
    if (choices == "back"){  
        new CTLB();  
        framemenu.dispose();  
    }  
}  
}
```

```
class MP implements ActionListener{  
    private JFrame framemp;  
    private JButton BACK, DIABETIC, HIGHBLOOD, HEART, ULCER;  
  
    public MP(){  
        framemp = new JFrame();  
        framemp.setContentPane(new JLabel(new  
ImageIcon(getClass().getResource("/MPPage.png"))));  
        Image logo =  
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));  
        framemp.setIconImage(logo);  
        framemp.setTitle("CHEF'S CACHE");  
  
        BACK = new JButton();  
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));  
        BACK.setOpaque(false);  
        BACK.setContentAreaFilled(false);  
        BACK.setBounds(10,10,70,70);  
        BACK.setBorderPainted(false);  
        ImageIcon backA = new ImageIcon();  
        BACK.setPressedIcon(backA);  
        BACK.addActionListener(this);  
        BACK.setName("back");  
  
        DIABETIC = new JButton();  
        DIABETIC.setIcon(new ImageIcon(getClass().getResource("/DIABETIC.png")));  
        DIABETIC.setOpaque(false);  
        DIABETIC.setContentAreaFilled(false);  
        DIABETIC.setBounds(100,270,120,100);  
        DIABETIC.setBorderPainted(false);  
        ImageIcon diabeticA = new ImageIcon();  
        DIABETIC.setPressedIcon(diabeticA);  
        DIABETIC.addActionListener(this);  
        DIABETIC.setName("diabetic");  
  
        HIGHBLOOD = new JButton();  
        HIGHBLOOD.setIcon(new  
ImageIcon(getClass().getResource("/HIGHBLOOD.png")));  
        HIGHBLOOD.setOpaque(false);  
        HIGHBLOOD.setContentAreaFilled(false);  
        HIGHBLOOD.setBounds(100,420,120,100);  
        HIGHBLOOD.setBorderPainted(false);  
        ImageIcon highbloodA = new ImageIcon();  
        HIGHBLOOD.setPressedIcon(highbloodA);
```

```

HIGHBLOOD.addActionListener(this);
HIGHBLOOD.setName("high");

HEART = new JButton();
HEART.setIcon(new ImageIcon(getClass().getResource("/HEART.png")));
HEART.setOpaque(false);
HEART.setContentAreaFilled(false);
HEART.setBounds(350,270,120,100);
HEART.setBorderPainted(false);
ImageIcon heartA = new ImageIcon();
HEART.setPressedIcon(heartA);
HEART.addActionListener(this);
HEART.setName("heart");

ULCER = new JButton();
ULCER.setIcon(new ImageIcon(getClass().getResource("/ULCER.png")));
ULCER.setOpaque(false);
ULCER.setContentAreaFilled(false);
ULCER.setBounds(350,420,120,100);
ULCER.setBorderPainted(false);
ImageIcon ulcerA = new ImageIcon();
ULCER.setPressedIcon(ulcerA);
ULCER.addActionListener(this);
ULCER.setName("ulcer");

framemp.add(DIABETIC);
framemp.add(HIGHBLOOD);
framemp.add(HEART);
framemp.add(ULCER);
framemp.add(BACK);

framemp.setBounds(660,0,600,1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new Homepage();
        framemp.dispose();
    }else if (choices == "diabetic") {
        new DIABETIC();
        framemp.dispose();
    }else if (choices == "high") {
        new HIGHBLOOD();
        framemp.dispose();
    }else if (choices == "heart") {
        new HEART();
        framemp.dispose();
    }else if (choices == "ulcer") {
        new ULCER();
        framemp.dispose();
    }
}

```

```
}
```

```
class MPB implements ActionListener{
    private JFrame framemp;
    private JButton BACK, DIABETIC, HIGHBLOOD, HEART, ULCER;

    public MPB(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage2.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        DIABETIC = new JButton();
        DIABETIC.setIcon(new ImageIcon(getClass().getResource("/DIABETIC.png")));
        DIABETIC.setOpaque(false);
        DIABETIC.setContentAreaFilled(false);
        DIABETIC.setBounds(100,270,120,100);
        DIABETIC.setBorderPainted(false);
        ImageIcon diabeticA = new ImageIcon();
        DIABETIC.setPressedIcon(diabeticA);
        DIABETIC.addActionListener(this);
        DIABETIC.setName("diabetic");

        HIGHBLOOD = new JButton();
        HIGHBLOOD.setIcon(new
ImageIcon(getClass().getResource("/HIGHBLOOD.png")));
        HIGHBLOOD.setOpaque(false);
        HIGHBLOOD.setContentAreaFilled(false);
        HIGHBLOOD.setBounds(100,420,120,100);
        HIGHBLOOD.setBorderPainted(false);
        ImageIcon highbloodA = new ImageIcon();
        HIGHBLOOD.setPressedIcon(highbloodA);
        HIGHBLOOD.addActionListener(this);
        HIGHBLOOD.setName("high");

        HEART = new JButton();
        HEART.setIcon(new ImageIcon(getClass().getResource("/HEART.png")));
        HEART.setOpaque(false);
        HEART.setContentAreaFilled(false);
        HEART.setBounds(350,270,120,100);
        HEART.setBorderPainted(false);
        ImageIcon heartA = new ImageIcon();
```

```

HEART.setPressedIcon(heartA);
HEART.addActionListener(this);
HEART.setName("heart");

ULCER = new JButton();
ULCER.setIcon(new ImageIcon(getClass().getResource("/ULCER.png")));
ULCER.setOpaque(false);
ULCER.setContentAreaFilled(false);
ULCER.setBounds(350,420,120,100);
ULCER.setBorderPainted(false);
ImageIcon ulcerA = new ImageIcon();
ULCER.setPressedIcon(ulcerA);
ULCER.addActionListener(this);
ULCER.setName("ulcer");

framemp.add(DIABETIC);
framemp.add(HIGHBLOOD);
framemp.add(HEART);
framemp.add(ULCER);
framemp.add(BACK);

framemp.setBounds(660,0,600,1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new HomepageB();
        framemp.dispose();
    }else if (choices == "diabetic") {
        new DIABETICB();
        framemp.dispose();
    }else if (choices == "high") {
        new HIGHBLOODB();
        framemp.dispose();
    }else if (choices == "heart") {
        new HEARTB();
        framemp.dispose();
    }else if (choices == "ulcer") {
        new ULCERB();
        framemp.dispose();
    }
}
}

```

```

class DIABETIC implements ActionListener{
    private JFrame framemp;
    private JButton BACK, EGG, CARROT, BREAD, MUSH, AVOCADO, ALMOND;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public DIABETIC(){

```



```
framemp = new JFrame();
framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage.png"))));
Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
framemp.setIconImage(logo);
framemp.setTitle("CHEF'S CACHE");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framemp.setBounds(660,0,600,1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

menutext = new JTextArea();
menutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText(" ----DIABETIC FOOD PLAN----"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast: One poached egg and half a small avocado spread on one
slice of Ezekiel bread, one orange. Total carbs: 39."
+ "\n "
+ "\n Lunch: Mexican bowl: Two-thirds of a cup low sodium canned pinto
beans, 1 cup chopped spinach, a quarter cup chopped tomatoes, a quarter cup bell
peppers, 1 ounce (oz) cheese, 1 tablespoon (tbsp) salsa. Total carbs: 30."
+ "\n "
+ "\n Snack: Twenty 1-gram baby carrots with 2 tbsp hummus. Total carbs:
21."
+ "\n "
+ "\n Dinner: 1 cup cooked lentil penne pasta, 2 oz ground lean turkey, 1.5
cups veggie tomato sauce with garlic, mushrooms, greens, zucchini, and eggplant.
Total carbs: 35."
+ "\n "
+ "\n Total carbs for the day: 125."
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast: Three-quarter cup of blueberries, 1 cup cooked oatmeal, 1 oz
almonds, 1 teaspoon (tsp) chia seeds. Total carbs: 34."
+ "\n "
```



```

+ "\n Lunch: Salad: Half a cup chickpeas, 2 cups fresh spinach, 2 oz grilled
chicken breast, half a small avocado, half a cup sliced strawberries, one-quarter cup
shredded carrots, 2 tbsp dressing. Total carbs: 52."
+ "\n "
+ "\n Snack: One small peach diced into one-third cup of 2% cottage cheese.
Total carbs: 16."
+ "\n "
+ "\n Dinner: Mediterranean couscous: Two-thirds of a cup whole wheat
cooked couscous, half a cup sautéed eggplant, four sundried tomatoes, five chopped
jumbo olives, half a diced cucumber, 1 tbsp balsamic vinegar, fresh basil. Total carbs:
38."
+ "\n "
+ "\n Total carbs for the day: 140."
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast: Three-quarter cup blueberries, two-egg veggie omelet with
spinach, mushrooms, bell pepper, avocado, and a half cup of black beans. Total carbs:
34."
+ "\n "
+ "\n Lunch: Sandwich: Two regular slices of 100% whole wheat bread, 1 tbsp
plain, nonfat Greek yogurt, 1 tbsp mustard, 2 oz canned tuna in water mixed with a
quarter cup shredded carrots, 1 tbsp dill relish, 1 cup sliced tomato, half a medium
apple. Total carbs: 40."
+ "\n "
+ "\n Snack: 1 cup unsweetened kefir. Total carbs: 12."
+ "\n "
+ "\n Dinner: Half a cup succotash, 1 tsp butter, 2 oz pork tenderloin, 1 cup
cooked asparagus, half a cup fresh pineapple. Total carbs: 34."
+ "\n "
+ "\n Total carbs for the day: 120."
+ "\n ");

```

```

menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as
needed rgb(216, 174, 126)

```

```

menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position

```

```

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_
ALWAYS); // Enable vertical scrolling

```

```

EGG = new JButton();
EGG.setIcon(new ImageIcon(getClass().getResource("/EGG.png")));
EGG.setBounds(170,650,100,100);
EGG.setOpaque(false);
EGG.setContentAreaFilled(false);
EGG.setBorderPainted(false);
ImageIcon eggA = new ImageIcon();
EGG.setPressedIcon(eggA);

```



```

BREAD = new JButton();
BREAD.setIcon(new ImageIcon(getClass().getResource("/BREAD.png")));
BREAD.setBounds(310,650,100,100);
BREAD.setOpaque(false);
BREAD.setContentAreaFilled(false);
BREAD.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
BREAD.setPressedIcon(breadA);

CARROT = new JButton();
CARROT.setIcon(new ImageIcon(getClass().getResource("/CARROT.png")));
CARROT.setBounds(120,650,100,100);
CARROT.setOpaque(false);
CARROT.setContentAreaFilled(false);
CARROT.setBorderPainted(false);
ImageIcon carrotA = new ImageIcon();
CARROT.setPressedIcon(carrotA);

MUSH = new JButton();
MUSH.setIcon(new ImageIcon(getClass().getResource("/MUSH.png")));
MUSH.setBounds(250,650,100,100);
MUSH.setOpaque(false);
MUSH.setContentAreaFilled(false);
MUSH.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
MUSH.setPressedIcon(mushA);

AVOCADO = new JButton();
AVOCADO.setIcon(new ImageIcon(getClass().getResource("/AVOCADO.png")));
AVOCADO.setBounds(400,650,100,100);
AVOCADO.setOpaque(false);
AVOCADO.setContentAreaFilled(false);
AVOCADO.setBorderPainted(false);
ImageIcon almondA = new ImageIcon();
AVOCADO.setPressedIcon(almondA);

ALMOND = new JButton();
ALMOND.setIcon(new ImageIcon(getClass().getResource("/ALMOND.png")));
ALMOND.setBounds(60,650,100,100);
ALMOND.setOpaque(false);
ALMOND.setContentAreaFilled(false);
ALMOND.setBorderPainted(false);
ImageIcon avocadoA = new ImageIcon();
ALMOND.setPressedIcon(avocadoA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(EGG);
framemp.add(CARROT);
framemp.add(BREAD);
framemp.add(MUSH);
framemp.add(AVOCADO);
framemp.add(ALMOND);

framemp.setBounds(660, 0, 600, 1020);

```



```
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
EGG.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Egg Nutrition Information:\nProtein: 6g\nFat: 5g\nCarbohydrates: 1g");
    }
});

BREAD.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Bread Nutrition Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 15g");
    }
});

CARROT.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Carrot Nutrition Information:\nProtein: 1g\nFat: 0g\nCarbohydrates: 6g");
    }
});

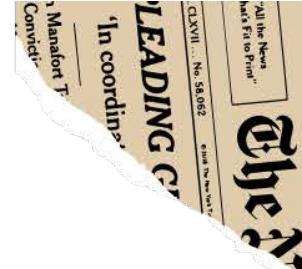
MUSH.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Mushroom Nutrition Information:\nProtein: 2g\nFat: 0g\nCarbohydrates: 2g");
    }
});

AVOCADO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Avocado Nutrition Information:\nProtein: 4g\nFat: 0g\nCarbohydrates: 4g");
    }
});

ALMOND.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Almond Nutrition Information:\nProtein: 5g\nFat: 0g\nCarbohydrates: 3g");
    }
});

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
```



```
String choices = choice.getName();
if (choices.equals("back")) { // Use equals() method for string comparison
    new MP();
    framemp.dispose();
}
}

class DIABETICB implements ActionListener{
    private JFrame framemp;
    private JButton BACK, EGG, CARROT, BREAD, MUSH, AVOCADO, ALMOND;
    private JTextArea minutext;
    private JScrollPane menuscroll;

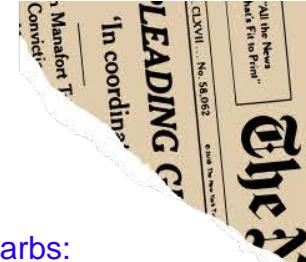
    public DIABETICB(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage2.png"))));
        Image logo =
            Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
        framemp.setResizable(false);
        framemp.setLayout(null);
        framemp.setVisible(true);

        minutext = new JTextArea();
        minutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
        size
        minutext.setEditable(false); // Set as non-editable
        minutext.setOpaque(true); // Make it opaque to show background color
        minutext.setForeground(Color.BLACK); // Set text color
        minutext.setText(" ----DIABETIC FOOD PLAN----"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast: One poached egg and half a small avocado spread on one
slice of Ezekiel bread, one orange. Total carbs: 39."
+ "\n "
+ "\n Lunch: Mexican bowl: Two-thirds of a cup low sodium canned pinto
beans, 1 cup chopped spinach, a quarter cup chopped tomatoes, a quarter cup bell
peppers, 1 ounce (oz) cheese, 1 tablespoon (tbsp) salsa. Total carbs: 30."
    }
}
```





+ "\n"
+ "\n Snack: Twenty 1-gram baby carrots with 2 tbsp hummus. Total carbs: 21."
+ "\n "
+ "\n Dinner: 1 cup cooked lentil penne pasta, 2 oz ground lean turkey, 1.5 cups veggie tomato sauce with garlic, mushrooms, greens, zucchini, and eggplant. Total carbs: 35."
+ "\n "
+ "\n Total carbs for the day: 125."
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast: Three-quarter cup of blueberries, 1 cup cooked oatmeal, 1 oz almonds, 1 teaspoon (tsp) chia seeds. Total carbs: 34."
+ "\n "
+ "\n Lunch: Salad: Half a cup chickpeas, 2 cups fresh spinach, 2 oz grilled chicken breast, half a small avocado, half a cup sliced strawberries, one-quarter cup shredded carrots, 2 tbsp dressing. Total carbs: 52."
+ "\n "
+ "\n Snack: One small peach diced into one-third cup of 2% cottage cheese. Total carbs: 16."
+ "\n "
+ "\n Dinner: Mediterranean couscous: Two-thirds of a cup whole wheat cooked couscous, half a cup sautéed eggplant, four sundried tomatoes, five chopped jumbo olives, half a diced cucumber, 1 tbsp balsamic vinegar, fresh basil. Total carbs: 38."
+ "\n "
+ "\n Total carbs for the day: 140."
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast: Three-quarter cup blueberries, two-egg veggie omelet with spinach, mushrooms, bell pepper, avocado, and a half cup of black beans. Total carbs: 34."
+ "\n "
+ "\n Lunch: Sandwich: Two regular slices of 100% whole wheat bread, 1 tbsp plain, nonfat Greek yogurt, 1 tbsp mustard, 2 oz canned tuna in water mixed with a quarter cup shredded carrots, 1 tbsp dill relish, 1 cup sliced tomato, half a medium apple. Total carbs: 40."
+ "\n "
+ "\n Snack: 1 cup unsweetened kefir. Total carbs: 12."
+ "\n "
+ "\n Dinner: Half a cup succotash, 1 tsp butter, 2 oz pork tenderloin, 1 cup cooked asparagus, half a cup fresh pineapple. Total carbs: 34."
+ "\n "
+ "\n Total carbs for the day: 120."
+ "\n ");

```
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(208, 212, 202)); // Adjust RGB values as
needed rgb(208, 212, 202)
```





```
menuscroll = new JScrollPane(menuText);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling
```

```
EGG = new JButton();
EGG.setIcon(new ImageIcon(getClass().getResource("/EGG.png")));
EGG.setBounds(170,650,100,100);
EGG.setOpaque(false);
EGG.setContentAreaFilled(false);
EGG.setBorderPainted(false);
ImageIcon eggA = new ImageIcon();
EGG.setPressedIcon(eggA);
```

```
BREAD = new JButton();
BREAD.setIcon(new ImageIcon(getClass().getResource("/BREAD.png")));
BREAD.setBounds(310,650,100,100);
BREAD.setOpaque(false);
BREAD.setContentAreaFilled(false);
BREAD.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
BREAD.setPressedIcon(breadA);
```

```
CARROT = new JButton();
CARROT.setIcon(new ImageIcon(getClass().getResource("/CARROT.png")));
CARROT.setBounds(120,650,100,100);
CARROT.setOpaque(false);
CARROT.setContentAreaFilled(false);
CARROT.setBorderPainted(false);
ImageIcon carrotA = new ImageIcon();
CARROT.setPressedIcon(carrotA);
```

```
MUSH = new JButton();
MUSH.setIcon(new ImageIcon(getClass().getResource("/MUSH.png")));
MUSH.setBounds(250,650,100,100);
MUSH.setOpaque(false);
MUSH.setContentAreaFilled(false);
MUSH.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
MUSH.setPressedIcon(mushA);
```

```
AVOCADO = new JButton();
AVOCADO.setIcon(new ImageIcon(getClass().getResource("/AVOCADO.png")));
AVOCADO.setBounds(400,650,100,100);
AVOCADO.setOpaque(false);
AVOCADO.setContentAreaFilled(false);
AVOCADO.setBorderPainted(false);
ImageIcon almondA = new ImageIcon();
AVOCADO.setPressedIcon(almondA);
```

```
ALMOND = new JButton();
ALMOND.setIcon(new ImageIcon(getClass().getResource("/ALMOND.png")));
ALMOND.setBounds(60,650,100,100);
ALMOND.setOpaque(false);
```



```

ALMOND.setContentAreaFilled(false);
ALMOND.setBorderPainted(false);
ImageIcon avocadoA = new ImageIcon();
ALMOND.setPressedIcon(avocadoA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(EGG);
framemp.add(CARROT);
framemp.add(BREAD);
framemp.add(MUSH);
framemp.add(AVOCADO);
framemp.add(ALMOND);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
EGG.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Egg Nutrition Information:\nProtein: 6g\nFat: 5g\nCarbohydrates: 1g");
    }
});

BREAD.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Bread Nutrition Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 15g");
    }
});

CARROT.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Carrot Nutrition Information:\nProtein: 1g\nFat: 0g\nCarbohydrates: 6g");
    }
});

MUSH.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Mushroom Nutrition Information:\nProtein: 2g\nFat: 0g\nCarbohydrates: 2g");
    }
});

AVOCADO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Avocado Nutrition Information:\nProtein: 4g\nFat: 0g\nCarbohydrates: 4g");
    }
});

```

```

        }

    });

ALMOND.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Almond Nutrition
Information:\nProtein: 5g\nFat: 0g\nCarbohydrates: 3g");
    }
});

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MPB();
        framemp.dispose();
    }
}

class HEART implements ActionListener{
    private JFrame framemp;
    private JButton BACK, POTATO, BREAD, PAPAYA, BLUE, TOMATO, CHICK;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public HEART(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage.png"))));
        Image logo =
            Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
        framemp.setResizable(false);
        framemp.setLayout(null);
        framemp.setVisible(true);
    }
}

```

```

menutext = new JTextArea();
menutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("HEART PROBLEM FOOD PLAN"
    + "\n "
    + "\n MONDAY "
    + "\n "
    + "\n Breakfast (271 calories): 1 serving Avocado-Egg Toast"
    + "\n "
    + "\n Lunch (350 calories): 1 serving Loaded Black Bean Nacho Soup"
    + "\n "
    + "\n P.M. Snack (62 calories): 1 medium orange"
    + "\n "
    + "\n Dinner (457 calories): 1 serving Seared Salmon with Green Peppercorn
Sauce1 cup steamed green beans, 1 baked medium red potato, drizzled with 1 tsp.
olive oil, and 1 Tbsp. nonfat plain Greek yogurt and a pinch of pepper"
    + "\n "
    + "\n Daily Totals: 1,224 calories, 60 g protein, 142 g carbohydrates, 28 g
fiber, 52 g fat, 11 g saturated fat, 828 mg sodium"
    + "\n "
    + "\n "
    + "\n TUESDAY"
    + "\n "
    + "\n Breakfast (265 calories): 1 cup bran cereal, 1 cup fat-free milk, and 1/4
cup blueberries."
    + "\n "
    + "\n Lunch (374 calories): 1 serving Spinach & Strawberry Meal-Prep Salad"
    + "\n "
    + "\n P.M. Snack (62 calories): 1 medium orange"
    + "\n "
    + "\n Dinner (429 calories): 1 serving Charred Shrimp, Pesto & Quinoa Bowls"
    + "\n "
    + "\n Daily Totals: 1,225 calories, 75 g protein, 148 g carbohydrates, 39 g
fiber, 50 g fat, 9 g saturated fat, 1,363 mg sodium"
    + "\n "
    + "\n "
    + "\n WEDNESDAY"
    + "\n "
    + "\n Breakfast (297 calories): 1 cup nonfat plain Greek yogurt, 3/4 cup
blueberries, 1 1/2 Tbsp. slivered almonds, and 2 tsp. honey"
    + "\n "
    + "\n Lunch (374 calories): 1 serving Spinach & Strawberry Meal-Prep Salad"
    + "\n "
    + "\n P.M. Snack (62 calories): 1 medium orange"
    + "\n "
    + "\n Dinner (416 calories): 1 serving Curried Sweet Potato & Peanut Soup
and 1 slice whole-wheat bread, toasted"
    + "\n "
    + "\n Daily Totals: 1,212 calories, 70 g protein, 132 g carbohydrates, 30 g
fiber, 51 g fat, 9 g saturated fat, 1,332 mg sodium"
    + "\n ");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);

```

```
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size  
// Set background color using RGB values  
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as  
needed rgb(216, 174, 126)
```

```
menuscroll = new JScrollPane(menutext);  
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_  
ALWAYS); // Enable vertical scrolling
```

```
POTATO = new JButton();  
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));  
POTATO .setBounds(170,650,100,100);  
POTATO .setOpaque(false);  
POTATO .setContentAreaFilled(false);  
POTATO .setBorderPainted(false);  
ImageIcon potatoA = new ImageIcon();  
POTATO .setPressedIcon(potatoA);
```

```
BREAD = new JButton();  
BREAD.setIcon(new ImageIcon(getClass().getResource("/BREAD.png")));  
BREAD.setBounds(310,650,100,100);  
BREAD.setOpaque(false);  
BREAD.setContentAreaFilled(false);  
BREAD.setBorderPainted(false);  
ImageIcon breadA = new ImageIcon();  
BREAD.setPressedIcon(breadA);
```

```
PAPAYA = new JButton();  
PAPAYA.setIcon(new ImageIcon(getClass().getResource("/PAPAYA.png")));  
PAPAYA.setBounds(100,650,100,100);  
PAPAYA.setOpaque(false);  
PAPAYA.setContentAreaFilled(false);  
PAPAYA.setBorderPainted(false);  
ImageIcon papayaA = new ImageIcon();  
PAPAYA.setPressedIcon(papayaA);
```

```
BLUE = new JButton();  
BLUE.setIcon(new ImageIcon(getClass().getResource("/BLUE.png")));  
BLUE.setBounds(250,650,100,100);  
BLUE.setOpaque(false);  
BLUE.setContentAreaFilled(false);  
BLUE.setBorderPainted(false);  
ImageIcon mushA = new ImageIcon();  
BLUE.setPressedIcon(mushA);
```

```
TOMATO = new JButton();  
TOMATO .setIcon(new ImageIcon(getClass().getResource("/TOMATO.png")));  
TOMATO .setBounds(400,650,100,100);  
TOMATO .setOpaque(false);  
TOMATO .setContentAreaFilled(false);  
TOMATO .setBorderPainted(false);  
ImageIcon tomatoA = new ImageIcon();  
TOMATO .setPressedIcon(tomatoA);
```

```

CHICK = new JButton();
CHICK .setIcon(new ImageIcon(getClass().getResource("/CHICK.png")));
CHICK .setBounds(60,650,100,100);
CHICK .setOpaque(false);
CHICK .setContentAreaFilled(false);
CHICK .setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
CHICK .setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(BLUE);
framemp.add(PAPAYA);
framemp.add(BREAD);
framemp.add(POTATO);
framemp.add(TOMATO);
framemp.add(CHICK);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
BLUE.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Blueberries Nutrition
Information:\nProtein: 6g\nFat: 1g\nCarbohydrates: 1g");
    }
});

BREAD.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Bread Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 15g");
    }
});

PAPAYA.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Papaya Nutrition
Information:\nProtein: 3g\nFat: 0g\nCarbohydrates: 6g");
    }
});

POTATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Potato Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }
});

TOMATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

JOptionPane.showMessageDialog(null, "Tomato Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 3g");
}

});

CHICK.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Chick Nutrition
Information:\nProtein: 15g\nFat: 5g\nCarbohydrates: 6g");
    }
});

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MP();
        framemp.dispose();
    }
}
}

class HEARTB implements ActionListener{
    private JFrame framemp;
    private JButton BACK, POTATO, BREAD, PAPAYA, BLUE, TOMATO, CHICK;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public HEARTB(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage2.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
        framemp.setResizable(false);
        framemp.setLayout(null);
    }
}

```



```
framemp.setVisible(true);

menutext = new JTextArea();
menutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("HEART PROBLEM FOOD PLAN"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast (271 calories): 1 serving Avocado-Egg Toast"
+ "\n "
+ "\n Lunch (350 calories): 1 serving Loaded Black Bean Nacho Soup"
+ "\n "
+ "\n P.M. Snack (62 calories): 1 medium orange"
+ "\n "
+ "\n Dinner (457 calories): 1 serving Seared Salmon with Green Peppercorn
Sauce1 cup steamed green beans, 1 baked medium red potato, drizzled with 1 tsp.
olive oil, and 1 Tbsp. nonfat plain Greek yogurt and a pinch of pepper"
+ "\n "
+ "\n Daily Totals: 1,224 calories, 60 g protein, 142 g carbohydrates, 28 g
fiber, 52 g fat, 11 g saturated fat, 828 mg sodium"
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast (265 calories): 1 cup bran cereal, 1 cup fat-free milk, and 1/4
cup blueberries."
+ "\n "
+ "\n Lunch (374 calories): 1 serving Spinach & Strawberry Meal-Prep Salad"
+ "\n "
+ "\n P.M. Snack (62 calories): 1 medium orange"
+ "\n "
+ "\n Dinner (429 calories): 1 serving Charred Shrimp, Pesto & Quinoa Bowls"
+ "\n "
+ "\n Daily Totals: 1,225 calories, 75 g protein, 148 g carbohydrates, 39 g
fiber, 50 g fat, 9 g saturated fat, 1,363 mg sodium"
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast (297 calories): 1 cup nonfat plain Greek yogurt, 3/4 cup
blueberries, 1 1/2 Tbsp. slivered almonds, and 2 tsp. honey"
+ "\n "
+ "\n Lunch (374 calories): 1 serving Spinach & Strawberry Meal-Prep Salad"
+ "\n "
+ "\n P.M. Snack (62 calories): 1 medium orange"
+ "\n "
+ "\n Dinner (416 calories): 1 serving Curried Sweet Potato & Peanut Soup
and 1 slice whole-wheat bread, toasted"
+ "\n "
+ "\n Daily Totals: 1,212 calories, 70 g protein, 132 g carbohydrates, 30 g
fiber, 51 g fat, 9 g saturated fat, 1,332 mg sodium"
+ "\n ");
```





```
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(208, 212, 202)); // Adjust RGB values as
needed rgb(216, 174, 126)

menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling

POTATO = new JButton();
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));
POTATO .setBounds(170,650,100,100);
POTATO .setOpaque(false);
POTATO .setContentAreaFilled(false);
POTATO .setBorderPainted(false);
ImageIcon potatoA = new ImageIcon();
POTATO .setPressedIcon(potatoA);

BREAD = new JButton();
BREAD.setIcon(new ImageIcon(getClass().getResource("/BREAD.png")));
BREAD.setBounds(310,650,100,100);
BREAD.setOpaque(false);
BREAD.setContentAreaFilled(false);
BREAD.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
BREAD.setPressedIcon(breadA);

PAPAYA = new JButton();
PAPAYA.setIcon(new ImageIcon(getClass().getResource("/PAPAYA.png")));
PAPAYA.setBounds(100,650,100,100);
PAPAYA.setOpaque(false);
PAPAYA.setContentAreaFilled(false);
PAPAYA.setBorderPainted(false);
ImageIcon papayaA = new ImageIcon();
PAPAYA.setPressedIcon(papayaA);

BLUE = new JButton();
BLUE.setIcon(new ImageIcon(getClass().getResource("/BLUE.png")));
BLUE.setBounds(250,650,100,100);
BLUE.setOpaque(false);
BLUE.setContentAreaFilled(false);
BLUE.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
BLUE.setPressedIcon(mushA);

TOMATO = new JButton();
TOMATO .setIcon(new ImageIcon(getClass().getResource("/TOMATO.png")));
TOMATO .setBounds(400,650,100,100);
TOMATO .setOpaque(false);
TOMATO .setContentAreaFilled(false);
TOMATO .setBorderPainted(false);
ImageIcon tomatoA = new ImageIcon();
```



```

TOMATO.setPressedIcon(tomatoA);

CHICK = new JButton();
CHICK.setIcon(new ImageIcon(getClass().getResource("/CHICK.png")));
CHICK.setBounds(60,650,100,100);
CHICK.setOpaque(false);
CHICK.setContentAreaFilled(false);
CHICK.setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
CHICK.setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(BLUE);
framemp.add(PAPAYA);
framemp.add(BREAD);
framemp.add(POTATO);
framemp.add(TOMATO);
framemp.add(CHICK);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
BLUE.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Blueberries Nutrition
Information:\nProtein: 6g\nFat: 1g\nCarbohydrates: 1g");
    }
});

BREAD.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Bread Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 15g");
    }
});

PAPAYA.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Papaya Nutrition
Information:\nProtein: 3g\nFat: 0g\nCarbohydrates: 6g");
    }
});

POTATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Potato Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }
});

```

```

TOMATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Tomato Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 3g");
    }
});

CHICK.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Chick Nutrition
Information:\nProtein: 15g\nFat: 5g\nCarbohydrates: 6g");
    }
});

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MPB();
        framemp.dispose();
    }
}

}

class HIGHBLOOD implements ActionListener{
    private JFrame framemp;
    private JButton BACK, SPIN, BEANS, STRAW, BLUE, SALMON, VIN, POTATO,
BREAST;
    private JTextArea menutext;
    private JScrollPane menuscroll;

    public HIGHBLOOD(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
    }
}

```



```
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

menutext = new JTextArea();
menutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText(" --HIGHBLOOD FOOD PLAN--"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast (309 calories): 1 serving Old-Fashioned Oatmeal, 1/4 cup
raspberries, and 3 Tbsp. chopped walnuts"
+ "\n "
+ "\n Lunch (455 calories): 1 serving White Bean & Veggie Salad, and 1
medium apple"
+ "\n "
+ "\n P.M. Snack (62 calories): 1 medium orange"
+ "\n "
+ "\n Dinner (522 calories): 1 serving Garlic Butter-Roasted Salmon with
Potatoes & Asparagus"
+ "\n "
+ "\n Daily Totals: 1,479 calories, 55 g protein, 166 g carbohydrates, 39 g
fiber, 75 g fat, 14 g saturated fat, 3,525 mg potassium, 875 mg sodium"
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast (309 calories): 1 serving Old-Fashioned Oatmeal, 1/4 cup
raspberries, and 3 Tbsp. chopped walnuts"
+ "\n "
+ "\n Lunch (381 calories): 1 serving Vegan Superfood Grain Bowls"
+ "\n "
+ "\n P.M. Snack (186 calories), 10 dry-roasted unsalted almonds, and 2 cups
cantaloupe, cubed"
+ "\n "
+ "\n Dinner (485 calories): 1 serving Chicken Hummus Bowls"
+ "\n "
+ "\n Daily Totals: 1,509 calories, 85 g protein, 144 g carbohydrates, 36 g
fiber, 74 g fat, 10 g saturated fat, 3,569 mg potassium, 1,186 mg sodium"
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast (278 calories): 1 cup nonfat plain Greek yogurt, 1/3 cup
blueberries, and 3 Tbsp. slivered almonds"
+ "\n "
+ "\n Lunch (381 calories): 1 serving Vegan Superfood Grain Bowls"
+ "\n "
+ "\n P.M. Snack (321 calories), 1 medium banana, and 1/3 cup dried walnut
halves"
+ "\n "
+ "\n Dinner (440 calories): 1 serving Baked Eggs in Tomato Sauce with Kale
and 1-oz. slice whole-wheat baguette"
```



```
+ "\n "
+ "\n Daily Totals: 1,492 calories, 76 g protein, 153 g carbohydrates, 31 g
fiber, 73 g fat, 9 g saturated fat, 3,553 mg potassium, 1,175 mg sodium"
+ "\n ");
```

```
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as
needed rgb(216, 174, 126)
```

```
menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position
```

```
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_
ALWAYS); // Enable vertical scrolling
```

```
POTATO = new JButton();
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));
POTATO .setBounds(450,650,100,100);
POTATO .setOpaque(false);
POTATO .setContentAreaFilled(false);
POTATO .setBorderPainted(false);
ImageIcon potatoA = new ImageIcon();
POTATO .setPressedIcon(potatoA);
```

```
BREAST = new JButton();
BREAST.setIcon(new ImageIcon(getClass().getResource("/BREAST.png")));
BREAST.setBounds(50,650,100,100);
BREAST.setOpaque(false);
BREAST.setContentAreaFilled(false);
BREAST.setBorderPainted(false);
ImageIcon breastA = new ImageIcon();
BREAST.setPressedIcon(breastA);
```

```
SPIN = new JButton();
SPIN.setIcon(new ImageIcon(getClass().getResource("/SPIN.png")));
SPIN.setBounds(170,650,100,100);
SPIN.setOpaque(false);
SPIN.setContentAreaFilled(false);
SPIN.setBorderPainted(false);
ImageIcon spinA = new ImageIcon();
SPIN.setPressedIcon(spinA);
```

```
STRAW = new JButton();
STRAW.setIcon(new ImageIcon(getClass().getResource("/STRAW.png")));
STRAW.setBounds(310,650,100,100);
STRAW.setOpaque(false);
STRAW.setContentAreaFilled(false);
STRAW.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
STRAW.setPressedIcon(breadA);
```

```
BEANS = new JButton();
BEANS.setIcon(new ImageIcon(getClass().getResource("/BEANS.png")));
```



```

BEANS.setBounds(130,650,100,100);
BEANS.setOpaque(false);
BEANS.setContentAreaFilled(false);
BEANS.setBorderPainted(false);
ImageIcon papayaA = new ImageIcon();
BEANS.setPressedIcon(papayaA);

BLUE = new JButton();
BLUE.setIcon(new ImageIcon(getClass().getResource("/BLUE.png")));
BLUE.setBounds(230,650,100,100);
BLUE.setOpaque(false);
BLUE.setContentAreaFilled(false);
BLUE.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
BLUE.setPressedIcon(mushA);

SALMON = new JButton();
SALMON.setIcon(new ImageIcon(getClass().getResource("/SALMON.png")));
SALMON.setBounds(400,650,100,100);
SALMON.setOpaque(false);
SALMON.setContentAreaFilled(false);
SALMON.setBorderPainted(false);
ImageIcon tomatoA = new ImageIcon();
SALMON.setPressedIcon(tomatoA);

VIN = new JButton();
VIN.setIcon(new ImageIcon(getClass().getResource("/VIN.png")));
VIN.setBounds(60,640,100,100);
VIN.setOpaque(false);
VIN.setContentAreaFilled(false);
VIN.setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
VIN.setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(BLUE);
framemp.add(STRAW);
framemp.add(BEANS);
framemp.add(BLUE);
framemp.add(SALMON);
framemp.add(VIN);
framemp.add(POTATO);
framemp.add(BREAST);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext setCaretPosition(0);

// Add ActionListener for buttons
BLUE.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

JOptionPane.showMessageDialog(null, "Blueberries Nutrition
Information:\nProtein: 6g\nFat: 1g\nCarbohydrates: 1g");
}

});

STRAW.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Strawberry Nutrition
Information:\nProtein: 4g\nFat: 1g\nCarbohydrates: 3g");
    }
});

BEANS.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Beans Nutrition
Information:\nProtein: 15g\nFat: 0g\nCarbohydrates: 8g");
    }
});

BLUE.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Blueberries Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }
});

});

SALMON.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Salmon Nutrition
Information:\nProtein: 10g\nFat: 5g\nCarbohydrates: 4g");
    }
});

);

VIN.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Balsamic Vinegar Nutrition
Information:\nProtein: 4g\nFat: 3g\nCarbohydrates: 2g");
    }
});

);

POTATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Potato Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }
});

);

BREAST.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Pork Nutrition Information:\nProtein:
16g\nFat: 10g\nCarbohydrates: 9g");
    }
});

);

```

```
}
```

```
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MP();
        framemp.dispose();
    }
}

class HIGHBLOODB implements ActionListener{
    private JFrame framemp;
    private JButton BACK, SPIN, BEANS, STRAW, BLUE, SALMON, VIN, POTATO,
BREAST;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public HIGHBLOODB(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage2.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
        framemp.setResizable(false);
        framemp.setLayout(null);
        framemp.setVisible(true);

        minutext = new JTextArea();
        minutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
        minutext.setEditable(false); // Set as non-editable
        minutext.setOpaque(true); // Make it opaque to show background color
        minutext.setForeground(Color.BLACK); // Set text color
        minutext.setText(" --HIGHBLOOD FOOD PLAN--"
+ "\n "
+ "\n MONDAY "
+ "\n "
```



+ "\n Breakfast (309 calories): 1 serving Old-Fashioned Oatmeal, 1/4 cup raspberries, and 3 Tbsp. chopped walnuts"
+ "\n "
+ "\n Lunch (455 calories): 1 serving White Bean & Veggie Salad, and 1 medium apple"
+ "\n "
+ "\n P.M. Snack (62 calories): 1 medium orange"
+ "\n "
+ "\n Dinner (522 calories): 1 serving Garlic Butter-Roasted Salmon with Potatoes & Asparagus"
+ "\n "
+ "\n Daily Totals: 1,479 calories, 55 g protein, 166 g carbohydrates, 39 g fiber, 75 g fat, 14 g saturated fat, 3,525 mg potassium, 875 mg sodium"
+ "\n "
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast (309 calories): 1 serving Old-Fashioned Oatmeal, 1/4 cup raspberries, and 3 Tbsp. chopped walnuts"
+ "\n "
+ "\n Lunch (381 calories): 1 serving Vegan Superfood Grain Bowls"
+ "\n "
+ "\n P.M. Snack (186 calories), 10 dry-roasted unsalted almonds, and 2 cups cantaloupe, cubed "
+ "\n "
+ "\n Dinner (485 calories): 1 serving Chicken Hummus Bowls"
+ "\n "
+ "\n Daily Totals: 1,509 calories, 85 g protein, 144 g carbohydrates, 36 g fiber, 74 g fat, 10 g saturated fat, 3,569 mg potassium, 1,186 mg sodium"
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast (278 calories): 1 cup nonfat plain Greek yogurt, 1/3 cup blueberries, and 3 Tbsp. slivered almonds"
+ "\n "
+ "\n Lunch (381 calories): 1 serving Vegan Superfood Grain Bowls"
+ "\n "
+ "\n P.M. Snack (321 calories), 1 medium banana, and 1/3 cup dried walnut halves"
+ "\n "
+ "\n Dinner (440 calories): 1 serving Baked Eggs in Tomato Sauce with Kale and 1-oz. slice whole-wheat baguette"
+ "\n "
+ "\n Daily Totals: 1,492 calories, 76 g protein, 153 g carbohydrates, 31 g fiber, 73 g fat, 9 g saturated fat, 3,553 mg potassium, 1,175 mg sodium"
+ "\n ");

```
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(208, 212, 202)); // Adjust RGB values as
needed rgb(216, 174, 126)
```

```
menuscroll = new JScrollPane(menutext);
```



```

menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling

POTATO = new JButton();
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));
POTATO .setBounds(450,650,100,100);
POTATO .setOpaque(false);
POTATO .setContentAreaFilled(false);
POTATO .setBorderPainted(false);
ImageIcon potatoA = new ImageIcon();
POTATO .setPressedIcon(potatoA);

BREAST = new JButton();
BREAST.setIcon(new ImageIcon(getClass().getResource("/BREAST.png")));
BREAST.setBounds(50,650,100,100);
BREAST.setOpaque(false);
BREAST.setContentAreaFilled(false);
BREAST.setBorderPainted(false);
ImageIcon breastA = new ImageIcon();
BREAST.setPressedIcon(breastA);

SPIN = new JButton();
SPIN.setIcon(new ImageIcon(getClass().getResource("/SPIN.png")));
SPIN.setBounds(170,650,100,100);
SPIN.setOpaque(false);
SPIN.setContentAreaFilled(false);
SPIN.setBorderPainted(false);
ImageIcon spinA = new ImageIcon();
SPIN .setPressedIcon(spinA);

STRAW = new JButton();
STRAW.setIcon(new ImageIcon(getClass().getResource("/STRAW.png")));
STRAW.setBounds(310,650,100,100);
STRAW.setOpaque(false);
STRAW.setContentAreaFilled(false);
STRAW.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
STRAW.setPressedIcon(breadA);

BEANS = new JButton();
BEANS.setIcon(new ImageIcon(getClass().getResource("/BEANS.png")));
BEANS.setBounds(130,650,100,100);
BEANS.setOpaque(false);
BEANS.setContentAreaFilled(false);
BEANS.setBorderPainted(false);
ImageIcon papayaA = new ImageIcon();
BEANS.setPressedIcon(papayaA);

BLUE = new JButton();
BLUE.setIcon(new ImageIcon(getClass().getResource("/BLUE.png")));
BLUE.setBounds(230,650,100,100);
BLUE.setOpaque(false);
BLUE.setContentAreaFilled(false);
BLUE.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();

```

```

BLUE.setPressedIcon(mushA);

SALMON = new JButton();
SALMON.setIcon(new ImageIcon(getClass().getResource("/SALMON.png")));
SALMON.setBounds(400,650,100,100);
SALMON.setOpaque(false);
SALMON.setContentAreaFilled(false);
SALMON.setBorderPainted(false);
ImageIcon tomatoA = new ImageIcon();
SALMON.setPressedIcon(tomatoA);

VIN = new JButton();
VIN.setIcon(new ImageIcon(getClass().getResource("/VIN.png")));
VIN.setBounds(60,640,100,100);
VIN.setOpaque(false);
VIN.setContentAreaFilled(false);
VIN.setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
VIN.setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(BLUE);
framemp.add(STRAW);
framemp.add(BEANS);
framemp.add(BLUE);
framemp.add(SALMON);
framemp.add(VIN);
framemp.add(POTATO);
framemp.add(BREAST);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
BLUE.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Blueberries Nutrition
Information:\nProtein: 6g\nFat: 1g\nCarbohydrates: 1g");
    }
});

STRAW.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Strawberry Nutrition
Information:\nProtein: 4g\nFat: 1g\nCarbohydrates: 3g");
    }
});

BEANS.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```



```
JOptionPane.showMessageDialog(null, "Beans Nutrition  
Information:\nProtein: 15g\nFat: 0g\nCarbohydrates: 8g");  
});  
  
BLUE.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JOptionPane.showMessageDialog(null, "Blueberries Nutrition  
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");  
    }  
});  
SALMON.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JOptionPane.showMessageDialog(null, "Salmon Nutrition  
Information:\nProtein: 10g\nFat: 5g\nCarbohydrates: 4g");  
    }  
});  
VIN.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JOptionPane.showMessageDialog(null, "Balsamic Vinegar Nutrition  
Information:\nProtein: 4g\nFat: 3g\nCarbohydrates: 2g");  
    }  
});  
POTATO.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JOptionPane.showMessageDialog(null, "Potato Nutrition  
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");  
    }  
});  
BREAST.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        JOptionPane.showMessageDialog(null, "Pork Nutrition Information:\nProtein:  
16g\nFat: 10g\nCarbohydrates: 9g");  
    }  
});  
}  
  
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton) e.getSource();  
    String choices = choice.getName();  
    if (choices.equals("back")) { // Use equals() method for string comparison  
        new MPB();  
        framemp.dispose();  
    }  
}
```

```

class ULCER implements ActionListener{
    private JFrame framemp;
    private JButton BACK, SPIN, BEANS, STRAW, AVOCADO, SALMON, TOMATO,
POTATO, BREAST;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public ULCER(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        framemp.setBounds(660,0,600,1020);
        framemp.setResizable(false);
        framemp.setLayout(null);
        framemp.setVisible(true);

        minutext = new JTextArea();
        minutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
        minutext.setEditable(false); // Set as non-editable
        minutext.setOpaque(true); // Make it opaque to show background color
        minutext.setForeground(Color.BLACK); // Set text color
        minutext.setText("      -----ULCER FOOD PLAN-----"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast (310 calories): 1 serving 10-Minute Spinach Omelet and 1 cup
cantaloupe, cubed"
+ "\n "
+ "\n Lunch (348 calories), 1 serving White Bean & Avocado Toast, 3/4 cup
low-fat plain Greek yogurt, and 1/4 cup blueberries"
+ "\n "
+ "\n P.M. Snack (206 calories): 1/4 cup unsalted dry-roasted almonds"
+ "\n "
+ "\n Dinner (503 calories): 1 serving Simple Grilled Salmon & Vegetables
and 1 cup cooked quinoa"
+ "\n "
+ "\n Daily Totals: 1,498 calories, 89g protein, 66g fat, 154g carbohydrate,
34g fiber, 1,250mg sodium"

```

```

+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast (384 calories): 1 serving Old-Fashioned Oatmeal, 1/2 cup
blueberries, and 1/4 cup chopped walnuts"
+ "\n "
+ "\n Lunch (376 calories): 1 serving Chicken & Kale Soup and 1 medium
banana"
+ "\n "
+ "\n P.M. Snack (182 calories): 1 cup low-fat plain Greek yogurt and 1/4 cup
raspberries"
+ "\n "
+ "\n Dinner (352 calories): 1 serving One-Pan Chicken & Asparagus Bake"
+ "\n "
+ "\n Daily Totals: 1,494 calories, 104g protein, 52g fat, 134g carbohydrate,
34g fiber, 1,416mg sodium"
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast (384 calories): 1 serving Old-Fashioned Oatmeal, 1/2 cup
blueberries, and 1/4 cup chopped walnuts"
+ "\n "
+ "\n Lunch (376 calories): 1 serving Chicken & Kale Soup and 1 medium
banana"
+ "\n "
+ "\n P.M. Snack (182 calories): 1 cup low-fat plain Greek yogurt and 1/4 cup
raspberries"
+ "\n "
+ "\n Dinner (467 calories): 1 serving Green Goddess Ricotta Pasta, 2 cups
mixed greens, and 1 serving Greek Salad Dressing"
+ "\n "
+ "\n Daily Totals: 1,509 calories, 88g protein, 57g fat, 174g carbohydrate,
33g fiber, 1,444mg sodium"
+ "\n ");

```

```

menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(216, 174, 126)); // Adjust RGB values as
needed rgb(216, 174, 126)

```

```

menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position

```

```

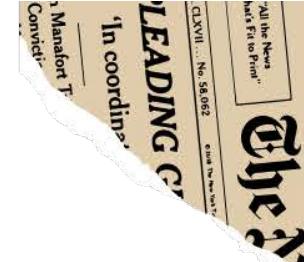
menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_
ALWAYS); // Enable vertical scrolling

```

```

POTATO = new JButton();
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));
POTATO .setBounds(450,650,100,100);
POTATO .setOpaque(false);
POTATO .setContentAreaFilled(false);
POTATO .setBorderPainted(false);

```



```
ImageIcon potatoA = new ImageIcon();
POTATO .setPressedIcon(potatoA);

BREAST = new JButton();
BREAST.setIcon(new ImageIcon(getClass().getResource("/BREAST.png")));
BREAST.setBounds(50,650,100,100);
BREAST.setOpaque(false);
BREAST.setContentAreaFilled(false);
BREAST.setBorderPainted(false);
ImageIcon breastA = new ImageIcon();
BREAST.setPressedIcon(breastA);

SPIN = new JButton();
SPIN.setIcon(new ImageIcon(getClass().getResource("/SPIN.png")));
SPIN.setBounds(170,650,100,100);
SPIN.setOpaque(false);
SPIN.setContentAreaFilled(false);
SPIN.setBorderPainted(false);
ImageIcon spinA = new ImageIcon();
SPIN.setPressedIcon(spinA);

STRAW = new JButton();
STRAW.setIcon(new ImageIcon(getClass().getResource("/STRAW.png")));
STRAW.setBounds(310,650,100,100);
STRAW.setOpaque(false);
STRAW.setContentAreaFilled(false);
STRAW.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
STRAW.setPressedIcon(breadA);

BEANS = new JButton();
BEANS.setIcon(new ImageIcon(getClass().getResource("/BEANS.png")));
BEANS.setBounds(130,650,100,100);
BEANS.setOpaque(false);
BEANS.setContentAreaFilled(false);
BEANS.setBorderPainted(false);
ImageIcon papayaA = new ImageIcon();
BEANS.setPressedIcon(papayaA);

AVOCADO = new JButton();
AVOCADO.setIcon(new ImageIcon(getClass().getResource("/AVOCADO.png")));
AVOCADO.setBounds(230,650,100,100);
AVOCADO.setOpaque(false);
AVOCADO.setContentAreaFilled(false);
AVOCADO.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
AVOCADO.setPressedIcon(mushA);

SALMON = new JButton();
SALMON.setIcon(new ImageIcon(getClass().getResource("/SALMON.png")));
SALMON.setBounds(400,650,100,100);
SALMON.setOpaque(false);
SALMON.setContentAreaFilled(false);
SALMON.setBorderPainted(false);
ImageIcon tomatoA = new ImageIcon();
SALMON.setPressedIcon(tomatoA);
```



```

TOMATO = new JButton();
TOMATO.setIcon(new ImageIcon(getClass().getResource("/TOMATO.png")));
TOMATO.setBounds(200,620,100,100);
TOMATO.setOpaque(false);
TOMATO.setContentAreaFilled(false);
TOMATO.setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
TOMATO.setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(STRAW);
framemp.add(BEANS);
framemp.add(TOMATO);
framemp.add(SALMON);
framemp.add(AVOCADO);
framemp.add(POTATO);
framemp.add(BREAST);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
STRAW.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Strawberry Nutrition
Information:\nProtein: 4g\nFat: 1g\nCarbohydrates: 3g");
    }
});

BEANS.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Beans Nutrition
Information:\nProtein: 15g\nFat: 0g\nCarbohydrates: 8g");
    }
});

AVOCADO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Avocado Nutrition
Information:\nProtein: 4g\nFat: 0g\nCarbohydrates: 4g");
    }
});

SALMON.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Salmon Nutrition
Information:\nProtein: 10g\nFat: 5g\nCarbohydrates: 4g");
    }
});

TOMATO.addActionListener(new ActionListener() {

```

```

public void actionPerformed(ActionEvent e) {
    JOptionPane.showMessageDialog(null, "Tomato Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 3g");
}

});

POTATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Potato Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }

});

BREAST.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Pork Nutrition Information:\nProtein:
16g\nFat: 10g\nCarbohydrates: 9g");
    }

});

}

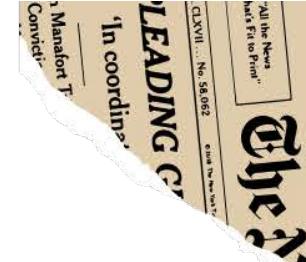
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MP();
        framemp.dispose();
    }
}
}

class ULCERB implements ActionListener{
    private JFrame framemp;
    private JButton BACK, SPIN, BEANS, STRAW, AVOCADO, SALMON, TOMATO,
POTATO, BREAST;
    private JTextArea minutext;
    private JScrollPane menuscroll;

    public ULCERB(){
        framemp = new JFrame();
        framemp.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/MPPage2.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        framemp.setIconImage(logo);
        framemp.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
    }
}

```



```
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

framemp.setBounds(660,0,600,1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

menutext = new JTextArea();
menutext.setFont(new Font("Cooper Black", Font.PLAIN, 30)); // Set larger font
size
menutext.setEditable(false); // Set as non-editable
menutext.setOpaque(true); // Make it opaque to show background color
menutext.setForeground(Color.BLACK); // Set text color
menutext.setText("    ----ULCER FOOD PLAN----"
+ "\n "
+ "\n MONDAY "
+ "\n "
+ "\n Breakfast (310 calories): 1 serving 10-Minute Spinach Omelet and 1 cup
cantaloupe, cubed"
+ "\n "
+ "\n Lunch (348 calories), 1 serving White Bean & Avocado Toast, 3/4 cup
low-fat plain Greek yogurt, and 1/4 cup blueberries"
+ "\n "
+ "\n P.M. Snack (206 calories): 1/4 cup unsalted dry-roasted almonds"
+ "\n "
+ "\n Dinner (503 calories): 1 serving Simple Grilled Salmon & Vegetables
and 1 cup cooked quinoa"
+ "\n "
+ "\n Daily Totals: 1,498 calories, 89g protein, 66g fat, 154g carbohydrate,
34g fiber, 1,250mg sodium"
+ "\n "
+ "\n "
+ "\n TUESDAY"
+ "\n "
+ "\n Breakfast (384 calories): 1 serving Old-Fashioned Oatmeal, 1/2 cup
blueberries, and 1/4 cup chopped walnuts"
+ "\n "
+ "\n Lunch (376 calories): 1 serving Chicken & Kale Soup and 1 medium
banana"
+ "\n "
+ "\n P.M. Snack (182 calories): 1 cup low-fat plain Greek yogurt and 1/4 cup
raspberries"
+ "\n "
+ "\n Dinner (352 calories): 1 serving One-Pan Chicken & Asparagus Bake"
+ "\n "
+ "\n Daily Totals: 1,494 calories, 104g protein, 52g fat, 134g carbohydrate,
34g fiber, 1,416mg sodium"
+ "\n "
+ "\n "
+ "\n WEDNESDAY"
+ "\n "
+ "\n Breakfast (384 calories): 1 serving Old-Fashioned Oatmeal, 1/2 cup
blueberries, and 1/4 cup chopped walnuts"
```





```

+ "\n "
+ "\n Lunch (376 calories): 1 serving Chicken & Kale Soup and 1 medium
banana"
+ "\n "
+ "\n P.M. Snack (182 calories): 1 cup low-fat plain Greek yogurt and 1/4 cup
raspberries"
+ "\n "
+ "\n Dinner (467 calories): 1 serving Green Goddess Ricotta Pasta, 2 cups
mixed greens, and 1 serving Greek Salad Dressing"
+ "\n "
+ "\n Daily Totals: 1,509 calories, 88g protein, 57g fat, 174g carbohydrate,
33g fiber, 1,444mg sodium"
+ "\n ");
menutext.setLineWrap(true);
menutext.setWrapStyleWord(true);
menutext.setPreferredSize(new Dimension(500, 4000)); // Set preferred size
// Set background color using RGB values
menutext.setBackground(new Color(208, 212, 202)); // Adjust RGB values as
needed rgb(216, 174, 126)

```




```

menuscroll = new JScrollPane(menutext);
menuscroll.setBounds(50, 250, 500, 300); // Adjust size and position

menuscroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS); // Enable vertical scrolling

POTATO = new JButton();
POTATO .setIcon(new ImageIcon(getClass().getResource("/POTATO.png")));
POTATO .setBounds(450,650,100,100);
POTATO .setOpaque(false);
POTATO .setContentAreaFilled(false);
POTATO .setBorderPainted(false);
ImageIcon potatoA = new ImageIcon();
POTATO .setPressedIcon(potatoA);

BREAST = new JButton();
BREAST.setIcon(new ImageIcon(getClass().getResource("/BREAST.png")));
BREAST.setBounds(50,650,100,100);
BREAST.setOpaque(false);
BREAST.setContentAreaFilled(false);
BREAST.setBorderPainted(false);
ImageIcon breastA = new ImageIcon();
BREAST.setPressedIcon(breastA);

SPIN = new JButton();
SPIN.setIcon(new ImageIcon(getClass().getResource("/SPIN.png")));
SPIN.setBounds(170,650,100,100);
SPIN.setOpaque(false);
SPIN.setContentAreaFilled(false);
SPIN.setBorderPainted(false);
ImageIcon spinA = new ImageIcon();
SPIN.setPressedIcon(spinA);

STRAW = new JButton();
STRAW.setIcon(new ImageIcon(getClass().getResource("/STRAW.png")));

```



```

STRAW.setBounds(310,650,100,100);
STRAW.setOpaque(false);
STRAW.setContentAreaFilled(false);
STRAW.setBorderPainted(false);
ImageIcon breadA = new ImageIcon();
STRAW.setPressedIcon(breadA);

BEANS = new JButton();
BEANS.setIcon(new ImageIcon(getClass().getResource("/BEANS.png")));
BEANS.setBounds(130,650,100,100);
BEANS.setOpaque(false);
BEANS.setContentAreaFilled(false);
BEANS.setBorderPainted(false);
ImageIcon papayaA = new ImageIcon();
BEANS.setPressedIcon(papayaA);

AVOCADO = new JButton();
AVOCADO.setIcon(new ImageIcon(getClass().getResource("/AVOCADO.png")));
AVOCADO.setBounds(230,650,100,100);
AVOCADO.setOpaque(false);
AVOCADO.setContentAreaFilled(false);
AVOCADO.setBorderPainted(false);
ImageIcon mushA = new ImageIcon();
AVOCADO.setPressedIcon(mushA);

SALMON = new JButton();
SALMON.setIcon(new ImageIcon(getClass().getResource("/SALMON.png")));
SALMON.setBounds(400,650,100,100);
SALMON.setOpaque(false);
SALMON.setContentAreaFilled(false);
SALMON.setBorderPainted(false);
ImageIcon tomatoA = new ImageIcon();
SALMON.setPressedIcon(tomatoA);

TOMATO = new JButton();
TOMATO.setIcon(new ImageIcon(getClass().getResource("/TOMATO.png")));
TOMATO.setBounds(200,620,100,100);
TOMATO.setOpaque(false);
TOMATO.setContentAreaFilled(false);
TOMATO.setBorderPainted(false);
ImageIcon chickA = new ImageIcon();
TOMATO.setPressedIcon(chickA);

framemp.add(menuscroll); // Add scrollable text area to the frame
framemp.add(BACK);
framemp.add(STRAW);
framemp.add(BEANS);
framemp.add(TOMATO);
framemp.add(SALMON);
framemp.add(AVOCADO);
framemp.add(POTATO);
framemp.add(BREAST);

framemp.setBounds(660, 0, 600, 1020);
framemp.setResizable(false);
framemp.setLayout(null);
framemp.setVisible(true);

```



```
// Set the caret position to the beginning
menutext.setCaretPosition(0);

// Add ActionListener for buttons
STRAW.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Strawberry Nutrition
Information:\nProtein: 4g\nFat: 1g\nCarbohydrates: 3g");
    }
});

BEANS.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Beans Nutrition
Information:\nProtein: 15g\nFat: 0g\nCarbohydrates: 8g");
    }
});

AVOCADO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Avocado Nutrition
Information:\nProtein: 4g\nFat: 0g\nCarbohydrates: 4g");
    }
});

SALMON.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Salmon Nutrition
Information:\nProtein: 10g\nFat: 5g\nCarbohydrates: 4g");
    }
});

TOMATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Tomato Nutrition
Information:\nProtein: 3g\nFat: 1g\nCarbohydrates: 3g");
    }
});

POTATO.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Potato Nutrition
Information:\nProtein: 1g\nFat: 2g\nCarbohydrates: 15g");
    }
});

BREAST.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Pork Nutrition Information:\nProtein:
16g\nFat: 10g\nCarbohydrates: 9g");
    }
});
```



```
}
```

```
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) { // Use equals() method for string comparison
        new MPB();
        framemp.dispose();
    }
}
```

```
class INFO implements ActionListener{
    private JFrame frameinfo;
    private JButton BACK, ABOUT, CREDITS, INSTRUCTION;

    public INFO(){
        frameinfo = new JFrame();
        frameinfo.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/INFOPage.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameinfo.setIconImage(logo);
        frameinfo.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        ABOUT = new JButton();
        ABOUT.setIcon(new ImageIcon(getClass().getResource("/ABOUT.png")));
        ABOUT.setOpaque(false);
        ABOUT.setContentAreaFilled(false);
        ABOUT.setBounds(100,100,400,200);
        ABOUT.setBorderPainted(false);
        ImageIcon aboutA = new ImageIcon();
        ABOUT.setPressedIcon(aboutA);
        ABOUT.addActionListener(this);
        ABOUT.setName("about");

        CREDITS = new JButton();
        CREDITS.setIcon(new ImageIcon(getClass().getResource("/CREDITS.png")));
        CREDITS.setOpaque(false);
        CREDITS.setContentAreaFilled(false);
        CREDITS.setBounds(100,400,400,200);
        CREDITS.setBorderPainted(false);
        ImageIcon creditsA = new ImageIcon();
```

```

CREDITS.setPressedIcon(creditsA);
CREDITS.addActionListener(this);
CREDITS.setName("credits");

INSTRUCTION = new JButton();
INSTRUCTION.setIcon(new ImageIcon(getClass().getResource("/INS.png")));
INSTRUCTION.setOpaque(false);
INSTRUCTION.setContentAreaFilled(false);
INSTRUCTION.setBounds(100,700,400,200);
INSTRUCTION.setBorderPainted(false);
ImageIcon insA = new ImageIcon();
INSTRUCTION.setPressedIcon(insA);
INSTRUCTION.addActionListener(this);
INSTRUCTION.setName("ins");

frameinfo.add(BACK);
frameinfo.add(ABOUT);
frameinfo.add(CREDITS);
frameinfo.add(INSTRUCTION);

frameinfo.setBounds(660,0,600,1020);
frameinfo.setResizable(false);
frameinfo.setLayout(null);
frameinfo.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new Homepage();
        frameinfo.dispose();
    }else if (choices == "about"){
        new ABOUT();
        frameinfo.dispose();
    }else if (choices == "credits"){
        new CREDITS();
        frameinfo.dispose();
    }else if (choices == "ins"){
        new INSTRUCTION();
        frameinfo.dispose();
    }
}
}

```

```

class INFOB implements ActionListener{
    private JFrame frameinfo;
    private JButton BACK, ABOUT, CREDITS, INSTRUCTION;

    public INFOB(){
        frameinfo = new JFrame();
        frameinfo.setContentPane(new JLabel(new
        ImageIcon(getClass().getResource("/INFOPage2.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));

```

```
frameinfo.setIconImage(logo);
frameinfo.setTitle("CHEF'S CACHE");

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

ABOUT = new JButton();
ABOUT.setIcon(new ImageIcon(getClass().getResource("/ABOUT.png")));
ABOUT.setOpaque(false);
ABOUT.setContentAreaFilled(false);
ABOUT.setBounds(100,100,400,200);
ABOUT.setBorderPainted(false);
ImageIcon aboutA = new ImageIcon();
ABOUT.setPressedIcon(aboutA);
ABOUT.addActionListener(this);
ABOUT.setName("about");

CREDITS = new JButton();
CREDITS.setIcon(new ImageIcon(getClass().getResource("/CREDITS.png")));
CREDITS.setOpaque(false);
CREDITS.setContentAreaFilled(false);
CREDITS.setBounds(100,400,400,200);
CREDITS.setBorderPainted(false);
ImageIcon creditsA = new ImageIcon();
CREDITS.setPressedIcon(creditsA);
CREDITS.addActionListener(this);
CREDITS.setName("credits");

INSTRUCTION = new JButton();
INSTRUCTION.setIcon(new ImageIcon(getClass().getResource("/INS.png")));
INSTRUCTION.setOpaque(false);
INSTRUCTION.setContentAreaFilled(false);
INSTRUCTION.setBounds(100,700,400,200);
INSTRUCTION.setBorderPainted(false);
ImageIcon insA = new ImageIcon();
INSTRUCTION.setPressedIcon(insA);
INSTRUCTION.addActionListener(this);
INSTRUCTION.setName("ins");

frameinfo.add(BACK);
frameinfo.add(ABOUT);
frameinfo.add(CREDITS);
frameinfo.add(INSTRUCTION);

frameinfo.setBounds(660,0,600,1020);
frameinfo.setResizable(false);
frameinfo.setLayout(null);
frameinfo.setVisible(true);
```

```

        }
    public void actionPerformed(ActionEvent e) {
        JButton choice = (JButton)e.getSource();
        String choices = choice.getName();
        if (choices == "back"){
            new HomepageB();
            frameinfo.dispose();
        }else if (choices == "about"){
            new ABOUT2();
            frameinfo.dispose();
        }else if (choices == "credits"){
            new CREDITS2();
            frameinfo.dispose();
        }else if (choices == "ins"){
            new INSTRUCTION2();
            frameinfo.dispose();
        }
    }
}

```

```

class ABOUT implements ActionListener{
    private JFrame frameabout;
    private JButton BACK;

    public ABOUT(){
        frameabout = new JFrame();
        frameabout.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/ABOUTBG.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameabout.setIconImage(logo);
        frameabout.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        frameabout.add(BACK);

        frameabout.setBounds(660,0,600,1020);
        frameabout.setResizable(false);
        frameabout.setLayout(null);
        frameabout.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {

```



```
 JButton choice = (JButton)e.getSource();
 String choices = choice.getName();
 if (choices == "back"){
     new INFO();
     frameabout.dispose();
 }
}

class ABOUT2 implements ActionListener{
 private JFrame frameabout;
 private JButton BACK;

 public ABOUT2(){
     frameabout = new JFrame();
     frameabout.setContentPane(new JLabel(new
 ImageIcon(getClass().getResource("/ABOUT2.png"))));
     Image logo =
 Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
     frameabout.setIconImage(logo);
     frameabout.setTitle("CHEF'S CACHE");

     BACK = new JButton();
     BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
     BACK.setOpaque(false);
     BACK.setContentAreaFilled(false);
     BACK.setBounds(10,10,70,70);
     BACK.setBorderPainted(false);
     ImageIcon backA = new ImageIcon();
     BACK.setPressedIcon(backA);
     BACK.addActionListener(this);
     BACK.setName("back");

     frameabout.add(BACK);

     frameabout.setBounds(660,0,600,1020);
     frameabout.setResizable(false);
     frameabout.setLayout(null);
     frameabout.setVisible(true);
 }

 public void actionPerformed(ActionEvent e) {
     JButton choice = (JButton)e.getSource();
     String choices = choice.getName();
     if (choices == "back"){
         new INFOB();
         frameabout.dispose();
     }
 }
}
```

```
class CREDITS implements ActionListener{
```



```
private JFrame frameinfo;
private JButton BACK;
```

```
public CREDITS(){
    frameinfo = new JFrame();
    frameinfo.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CREDITSBG.png"))));
    Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
    frameinfo.setIconImage(logo);
    frameinfo.setTitle("CHEF'S CACHE");

    BACK = new JButton();
    BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
    BACK.setOpaque(false);
    BACK.setContentAreaFilled(false);
    BACK.setBounds(10,10,70,70);
    BACK.setBorderPainted(false);
    ImageIcon backA = new ImageIcon();
    BACK.setPressedIcon(backA);
    BACK.addActionListener(this);
    BACK.setName("back");

    frameinfo.add(BACK);

    frameinfo.setBounds(660,0,600,1020);
    frameinfo.setResizable(false);
    frameinfo.setLayout(null);
    frameinfo.setVisible(true);

}
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new INFO();
        frameinfo.dispose();
    }
}
```

```
class CREDITS2 implements ActionListener{
    private JFrame frameinfo;
    private JButton BACK;

    public CREDITS2(){
        frameinfo = new JFrame();
        frameinfo.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/CREDITS2.png"))));
        Image logo =
Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameinfo.setIconImage(logo);
        frameinfo.setTitle("CHEF'S CACHE");
```

```

BACK = new JButton();
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
BACK.setOpaque(false);
BACK.setContentAreaFilled(false);
BACK.setBounds(10,10,70,70);
BACK.setBorderPainted(false);
ImageIcon backA = new ImageIcon();
BACK.setPressedIcon(backA);
BACK.addActionListener(this);
BACK.setName("back");

frameinfo.add(BACK);

frameinfo.setBounds(660,0,600,1020);
frameinfo.setResizable(false);
frameinfo.setLayout(null);
frameinfo.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new INFOB();
        frameinfo.dispose();
    }
}
}

class INSTRUCTION implements ActionListener{
    private JFrame frameinfo;
    private JButton BACK;

    public INSTRUCTION(){
        frameinfo = new JFrame();
        frameinfo.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/INSBG.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameinfo.setIconImage(logo);
        frameinfo.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        frameinfo.add(BACK);
    }
}

```

```

frameinfo.setBounds(660,0,600,1020);
frameinfo.setResizable(false);
frameinfo.setLayout(null);
frameinfo.setVisible(true);

}

public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton)e.getSource();
    String choices = choice.getName();
    if (choices == "back"){
        new INFO();
        frameinfo.dispose();
    }
}

}

class INSTRUCTION2 implements ActionListener{
    private JFrame frameinfo;
    private JButton BACK;

    public INSTRUCTION2(){
        frameinfo = new JFrame();
        frameinfo.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/INS2.png"))));
        Image logo =
        Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameinfo.setIconImage(logo);
        frameinfo.setTitle("CHEF'S CACHE");

        BACK = new JButton();
        BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));
        BACK.setOpaque(false);
        BACK.setContentAreaFilled(false);
        BACK.setBounds(10,10,70,70);
        BACK.setBorderPainted(false);
        ImageIcon backA = new ImageIcon();
        BACK.setPressedIcon(backA);
        BACK.addActionListener(this);
        BACK.setName("back");

        frameinfo.add(BACK);

        frameinfo.setBounds(660,0,600,1020);
        frameinfo.setResizable(false);
        frameinfo.setLayout(null);
        frameinfo.setVisible(true);

    }

    public void actionPerformed(ActionEvent e) {
}

```



```
 JButton choice = (JButton)e.getSource();
 String choices = choice.getName();
 if (choices == "back"){
     new INFOB();
     frameinfo.dispose();
 }
}

public class THEME implements ActionListener {
 private JFrame frameth;
 private static JButton BACK, TH1, TH1A, TH2, TH2A;

 public THEME() {
     frameth = new JFrame();
     frameth.setContentPane(new JLabel(new
     ImageIcon(getClass().getResource("/THPage.png"))));
     Image logo =
     Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
     frameth.setIconImage(logo);
     frameth.setTitle("CHEF'S CACHE");

     TH1 = new JButton(new ImageIcon(getClass().getResource("/th1a.png")));
     TH1.setPressedIcon(new ImageIcon(getClass().getResource("/th1.png")));
     TH1.setBounds(50, 150, 240, 400);
     setButtonProperties(TH1, "th1");

     TH1A = new JButton(new ImageIcon(getClass().getResource("/th1.png")));
     TH1A.setPressedIcon(new ImageIcon(getClass().getResource("/th1a.png")));
     TH1A.setBounds(50, 150, 240, 400);
     setButtonProperties(TH1A, "th1a");

     TH2 = new JButton(new ImageIcon(getClass().getResource("/th2.png")));
     TH2.setPressedIcon(new ImageIcon(getClass().getResource("/th2a.png")));
     TH2.setBounds(310, 150, 240, 400);
     setButtonProperties(TH2, "th2");

     TH2A = new JButton(new ImageIcon(getClass().getResource("/th2a.png")));
     TH2A.setPressedIcon(new ImageIcon(getClass().getResource("/th2.png")));
     TH2A.setBounds(310, 150, 240, 400);
     setButtonProperties(TH2A, "th2a");

     BACK = new JButton(new ImageIcon(getClass().getResource("/back.png")));
     BACK.setPressedIcon(new ImageIcon(getClass().getResource("/back.png")));
     BACK.setBounds(10, 10, 70, 70);
     setButtonProperties(BACK, "back");

     frameth.add(TH1);
     frameth.add(TH1A);
     frameth.add(TH2);
     frameth.add(TH2A);
     frameth.add(BACK);

     frameth.setBounds(660, 0, 600, 1020);
     frameth.setResizable(false);
     frameth.setLayout(null);
     frameth.setVisible(true);
 }
```

```
}
```

```
private void setButtonProperties(JButton button, String name) {
    button.setOpaque(false);
    button.setContentAreaFilled(false);
    button.setBorderPainted(false);
    button.setName(name);
    button.addActionListener(this);
    button.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

@Override
public void actionPerformed(ActionEvent e) {
    JButton choice = (JButton) e.getSource();
    String choices = choice.getName();
    if (choices.equals("back")) {
        new Homepage();
        frameth.dispose();

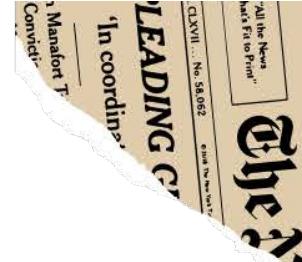
    } else if (choices.equals("th2")) {
        new HomepageB();
        frameth.dispose();
        Homepage.stopMusic();
        JOptionPane.showMessageDialog(null, "Theme 2 applied!");
    }
}

class THEMEB implements ActionListener {
    private JFrame frameth;
    private static JButton BACK, TH1, TH1A, TH2, TH2A;

    public THEMEB() {
        frameth = new JFrame();
        frameth.setContentPane(new JLabel(new
ImageIcon(getClass().getResource("/THPage2.png"))));
        Image logo =
            Toolkit.getDefaultToolkit().getImage(getClass().getResource("/CCLOGO.png"));
        frameth.setIconImage(logo);
        frameth.setTitle("CHEF'S CACHE");

        TH1 = new JButton();
        TH1.setIcon(new ImageIcon(getClass().getResource("/th1.png")));
        TH1.setBounds(50, 150, 240, 400);
        TH1.setOpaque(false);
        TH1.setContentAreaFilled(false);
        TH1.setBorderPainted(false);
        TH1.setPressedIcon(new ImageIcon());
        TH1.addActionListener(this);
        TH1.setName("th1a");

        TH1A = new JButton();
        TH1A.setIcon(new ImageIcon(getClass().getResource("/th1a.png")));
        TH1A.setBounds(50, 150, 240, 400);
        TH1A.setOpaque(false);
        TH1A.setContentAreaFilled(false);
        TH1A.setBorderPainted(false);
    }
}
```



```
TH1A.setPressedIcon(new ImageIcon());  
  
TH2 = new JButton();  
TH2.setIcon(new ImageIcon(getClass().getResource("/th2a.png")));  
TH2.setBounds(310, 150, 240, 400);  
TH2.setOpaque(false);  
TH2.setContentAreaFilled(false);  
TH2.setBorderPainted(false);  
TH2.setPressedIcon(new ImageIcon());  
TH2.addActionListener(this);  
TH2.setName("TH2");  
  
TH2A = new JButton();  
TH2A.setIcon(new ImageIcon(getClass().getResource("/th2.png")));  
TH2A.setBounds(310, 150, 240, 400);  
TH2A.setOpaque(false);  
TH2A.setContentAreaFilled(false);  
TH2A.setBorderPainted(false);  
TH2A.setPressedIcon(new ImageIcon());  
TH2A.addActionListener(this);  
TH2A.setName("th2a");  
  
BACK = new JButton();  
BACK.setIcon(new ImageIcon(getClass().getResource("/back2.png")));  
BACK.setBounds(10, 10, 70, 70);  
BACK.setOpaque(false);  
BACK.setContentAreaFilled(false);  
BACK.setBorderPainted(false);  
BACK.setPressedIcon(new ImageIcon());  
BACK.addActionListener(this);  
BACK.setName("back");  
  
frameth.add(TH1);  
frameth.add(TH1A);  
frameth.add(TH2);  
frameth.add(TH2A);  
frameth.add(BACK);  
  
frameth.setBounds(660, 0, 600, 1020);  
frameth.setResizable(false);  
frameth.setLayout(null);  
frameth.setVisible(true);  
}
```

```
public void actionPerformed(ActionEvent e) {  
    JButton choice = (JButton) e.getSource();  
    String choices = choice.getName();  
    if (choices.equals("back")) {  
        new HomepageB();  
        frameth.dispose();  
    } else if (choices.equals("th1a")) {  
        new Homepage();  
        frameth.dispose();  
        HomepageB.stopMusic();  
        JOptionPane.showMessageDialog(null, "Theme 1 applied!");  
    } else if (choices.equals("th2a")) {
```





new HomepageB();
frameth.dispose();

```
}
```





SCREENSHOT OF ACTUAL USAGE

