اونيۏرسيتي مليسيا قهڠ السلطان عبد الله UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH	**BSD2213 DATA SCIENCE PROGRAMMING I**		**MARKS**: 60 (30%)
	GROUP PROJECT	**DUE DATE: 11/1/2024**	
	GROUP 1 (01G)	**GROUP MEMBER:** 1.CHONG JIN JYE (SD22048) 2.SITI MAISARAH BINTI SUHARDI (SD22006) 3.NURUL SYAFIQAH NATASHA BINTI MOHD RAZI (SD22039) 4.TUAN NURSHAFIEKA WAHIDA BINTI TUAN NADIN (SD22030)	

CLO	Description	PLO Mapping	Percentage	Marks
CLO2	Use appropriate Python programming technique to solve problem.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills C3: Application	5%	10
CLO3	Construct and run program.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	15%	30
CLO4	Work collaboratively to solve assigned task.	PLO4: Functional work skills with focus on Interpersonal skills A3: Valuing	5%	10
CLO5	Demonstrate innovative ideas in developing a graphical user interface.	PLO8: Entrepreneurial skills A3: Valuing	5%	10

OBJECTIVES:

Programming skills are vital in solving practical problems across multiple fields, including data science, engineering, technology, and various industries. The group project in Data Science Programming I requires students to develop a friendly Graphical User Interface (GUI) using Python programming language. This project also serves as an opportunity for students to apply their foundational programming skills, incorporating the use of diverse libraries. Significance, innovation, attractiveness, and user-friendliness are the key standards for the GUI to be developed. It is hoped that this group project will be a motivational and enriching experience for the students.

GENERAL INSTRUCTIONS:

1. A group of three or four members has been set up based on the student's performance. Please refer to the given group
   members listing.
2. Work in a group and disseminate the task among your group members.
3. Before beginning the project, ensure you have received approval for your chosen project title from your lecturer. It needs to be settled **by or before 28th December 2023, 4.00 pm.**
4. Your project report must include the following information:
   i. Introduction.
      Analysis of an existing situation that makes you come out with this project.
   ii. Why this project?

Portray the significance of this project and its innovative values.

iii. How can this project be extended?
State the ideas on how this project can be extended.

iv. Source Code.

v. The screenshot for each activity in the GUI.

5. Submit the **softcopy** file in docx. version through KALAM by or before **11th January 2024, 11.59 pm.**
6. Present your project **on 12th January 2024 or 16th January 2024.** The schedule of the presentation will be given later.
7. Prepare your slide presentation appropriately.

PROJECT SCOPE:

1. Work in a team to develop **your own** graphical user interface (GUI).
2. Your GUI must have the following criteria/properties.
    a. Analysis of an existing situation and identify areas for improvement.
    b. A catchy GUI name.
    c. Portray significance ideas.
    d. Contains high innovative values.
    e. Attractive and pleasant to the eye.
    f. User friendly.
3. Your GUI should contain the following criteria:
    a. Use Tkinter library to develop the Graphical User Interface (GUI).
    b. Use input controls such as buttons, toggles, checkboxes and others.
    c. Use navigation/information components eg. message boxes, pull-down menu and others. d. Use Tkinter geometry manager.
4. You may use an additional library (Numpy/Sympy/Pandas/Matplotlib).

Table of Contents

1.0 Introduction

A notable trend in the food industry is the escalating popularity of food ordering apps. For many years, industry leaders of the fast food industry have been utilizing the convenience and accessibility of online food ordering apps to satisfy their customers' cravings to elevate their business performance. As a result, the implementation of online food ordering either for delivery or pick up has become a desirable feature that every vendor is looking for in the current digital age. The steadfast existence of food ordering systems via electronic gadgets have been an appealing and highly sought-after technology, and therefore became the inspiration for this project.

Food ordering systems must be user-friendly and informative in order to attract customers, given that the core intention of the system is to save time and effort for customers instead of having to arrive at physical stores and queue for ordering. A basic food ordering system must be able to store user data and their selected choices from the menu. Furthermore, the system should be able to clearly convey important information such as the price, description and illustration of the food. By ensuring the interface is straightforward and easy to navigate, the food ordering process will be seamless for the customer. In addition, the constant changes in food prices on the menu is also an issue, considering the probability of frequent promotions or economic changes. Hence, the program must be easily modifiable to reflect these changes without requiring excessive time and effort. The many properties of a competent food ordering system will require thorough understanding of programming techniques such as graphical user interface, thus it will be carefully analyzed in this project.

2.0 Significance of the Project

The significance of the project is to improve upon the current common food ordering systems by introducing innovative ideas that can benefit the customers. Nowadays, food ordering apps are strongly correlated with the fast food industry, where a majority of food orders are made from fast food restaurants. The issue is that fast food has been commonly known to be detrimental to our health, yet the existence of food ordering systems that deliver straight to your residence have made fast food much more accessible and alluring. It is undeniable that this ongoing phenomenon will only contribute to the rise of diseases such as obesity and diabetes in our nation.

By understanding the issue at hand, the project aims to introduce relevant ideas to promote the habit of healthy eating in our society. The project will develop a food ordering system, along with the implementation of graphical displays of nutritional values of the food. Moreover, an accurate description of the food will also be provided. The core idea is to enable customers to be mindful of their nutritional needs and allow them to evaluate their nutrition intake. This empowers the customers by conveying accurate nutritional information, hence increasing the likelihood of people who will consider their health and well-being instead of binge eating fast food. Besides, since self care and healthy living have been a prominent topic recently, the system also aims to attract health-conscious customers and eventually grow a healthy loyal customer base.

3.0 Extending the Project

One of the features of the program is that the displayed data of the menu can be easily modified so changes will be reflected in the whole program, because the data is stored in one dataframe using Pandas library and variables will point to it when required for use. However, the modification of a dataframe will still require programming knowledge and is not user-friendly for business use. Therefore, the project can be extended by storing all the menu data into an Excel spreadsheet to be imported into the program. Consequently, any staff can access the Excel spreadsheet to modify the menu data, and the data will automatically be imported into the program and displayed correctly.

Furthermore, the program can also be extended to collect relevant user data such as email and contact information for marketing purposes. Common customer retention strategies include providing informative newsletters and blog posts to customers via their email. By collecting legal user data, it can be leveraged to build business reputation and understand the customer base.

Another idea to extend the program is the implementation of frame widgets in the Tkinter library. The program will display message boxes and open windows when prompted by the user, but switching frames would offer a better user experience.

4.0 Source Code

```
import tkinter as tk
from tkinter import messagebox
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from PIL import Image, ImageTk


# PART 1
# Creating a data frame to store the menu
# The dataframe that refers to our menu is called MenuDf
FoodBevData = ['Brown Rice With Chicken', 'Brown Rice With Mushrooms', 'Creamy Lemon Shrimp Spaghetti',
               'Smoked Salmon Spaghetti',
               'Chicken Tacos With Avocado', 'Spicy Salmon Bowl', 'Daily Special Salad', 'Apple Pie', 'Orange Juice',
               'Watermelon Juice']
PriceData = ['12.50', '9.50', '12.50', '15.50', '10.00', '10.00', '8.00', '8.00', '4.00', '4.00']
CaloriesData = ['266', '127', '256', '237', '189', '235', '120', '357', '112', '80']
CarbsData = ['24.0', '25.0', '35.0', '31.1', '17.9', '27.0', '6.8', '43.0', '26.0', '21.0']
ProteinData = ['34.0', '4.0', '40.5', '32.1', '32.0', '16.9', '3.5', '4.1', '2.0', '1.0']
FatData = ['12.5', '2.0', '16.0', '22.3', '21.0', '15.6', '2.1', '14.0', '0.0', '0.1']
DescData = ["Brown rice with chicken is a dish combines tender, seasoned chicken with nutty and fiber-rich brown rice.",
            'Brown rice and mushrooms create a flavors of nutty brown rice complement the savory richness of sautéed mushrooms',
            'Creamy lemon shrimp spaghetti is a succulent shrimp, al dente spaghetti, and a velvety, citrus-infused cream sauce',
            'Smoked salmon spaghetti is a combines the delicate smokiness of cured salmon with spaghetti, often accompanied by a creamy dill',
            'Chicken with avocado is a delightful dish featuring tender, seasoned chicken paired with creamy slices of avocado',
            'A spicy salmon bowl is a vibrant and flavorful dish that features succulent, spicy-grilled salmon served over a bed of rice',
            'Salad is a mixed greens, vibrant vegetables, and often complemented by toppings such as nuts, seeds, or cheese, drizzled with a flavorful dressing',
            'Apple pie is a classic dessert, golden-brown crust enveloping sweet and spiced apple slices, often accompanied by a hint of cinnamon',
```

```
        'Orange juice is a refreshing beverage made by extracting the liquid from ripe oranges, delivering a zesty and naturally sweet flavor',
        'Watermelon juice is a hydrating and invigorating beverage crafted by blending or juicing ripe watermelon']
    data = {
        'Name': FoodBevData,
        'Price': PriceData,
        'Calories': CaloriesData,
        'Carbohydrates': CarbsData,
        'Protein': ProteinData,
        'Fat': FatData,
        'Desc': DescData
    }
    MenuDf = pd.DataFrame(data, index=range(1, len(FoodBevData) + 1))
    MenuDf[['Price', 'Calories', 'Carbohydrates', 'Protein', 'Fat']] = MenuDf[
        ['Price', 'Calories', 'Carbohydrates', 'Protein', 'Fat']].astype(float)

    # PART 2

    # Create an introductory window or splash screen
    intro_window = tk.Tk()
    intro_window.title("Welcome to NutriCrave")
    intro_window.geometry('2000x1800')
    # Load the background image
    image_path = "Wallpaper2.jpg"
    original_image = Image.open(image_path)

    # Resize the image without antialiasing
    window_width, window_height = 1800, 1000
    resized_image = original_image.resize((window_width, window_height), Image.NEAREST)

    # Convert the resized image to Tkinter-compatible format
    background_photo = ImageTk.PhotoImage(resized_image)

    # Create a label to display the resized background image
    background_label = tk.Label(intro_window, image=background_photo)
```

```
background_label.place(relwidth=1,relheight=1)

intro_title = tk.Label(intro_window,
                       text="Welcome to NutriCrave",
                       fg='snow',
                       bg='khaki4',
                       font=('Cooper Black', 50))
intro_subtitle = tk.Label(intro_window,
                          text="Your Healthy Meals Are Our Priority.",
                          fg='snow',
                          bg='khaki4',
                          font=('Cooper Black', 30))

intro_title.pack(pady=100)
intro_subtitle.pack(pady=10)

# This function will destroy the intro_window and create the main window

# To store username
def getUserName():
    global username
    username=unentry.get()
    print(username)
    intro_window.destroy()
    create_main_window()

# Add a button to proceed to the main window
unlabel = tk.Label(text="Enter Username:",
                   bg='khaki4',
                   fg='white',
                   font=('Arial',16))
unentry = tk.Entry(fg='black',
                   bg='white',
                   width=50)
```

```
unbutton = tk.Button(text = 'Start Ordering',
                     command=getUserName)
unlabel.place(x=600,y=400)
unentry.place(x=600,y=450)
unbutton.place(x=600,y=500)




# Function to create the main window
def create_main_window():
    # Main window code
    window = tk.Tk()
    window.title("NutriCrave")
    window.geometry('2000x1800')
    # window.configure(bg='khaki4')

    # Load the background image
    background_image = Image.open("Wallpaper 3.png")
    # Convert the image to Tkinter-compatible format
    background_photo = ImageTk.PhotoImage(background_image)

    # Create a label to display the background image
    background_label = tk.Label(window, image=background_photo)
    background_label.place(relwidth=1, relheight=1)

    # Creating basic window interface
    title = tk.Label(text="NutriCrave",
                     fg='snow',
                     bg='burlywood4',
                     font=('Berlin Sans FB Demi', 35,'bold'),
                     )
    subtitle = tk.Label(text="Your Healthy Meals Are Our Priority.",
```

```
                fg='snow',
                bg='burlywood4',
                font=('Berlin Sans FB Demi', 20,'bold'))
        title.place(x=555, y=0)
        subtitle.place(x=455, y=60)

        # Create a menu bar
        menubar = tk.Menu(window)
        window.config(menu=menubar)

        # Create a pull-down menu
        LearnMore_menu = tk.Menu(menubar, tearoff=0)
        menubar.add_cascade(label="Menu", menu=LearnMore_menu)
        LearnMore_menu.add_command(label="About Us", command=lambda: messagebox.showinfo("About Us"))

        def about_us():
            about_us_text = "Meet Our Team:\n\n" \
                "1. CHONG JIN JYE - FOOD SCIENTIST\n" \
                "2. SITI MAISARAH - FOOD SAFETY SPECIALIST\n" \
                "3. SYAFIQAH NATASHA - DIETITIAN\n" \
                "4. SHAFIEKA WAHIDA - SALES REPRESENTATIVE\n\n" \
                "Company Background:\n\n" \
                "NutriCrave is a Malaysian company that provides healthy and delicious meals through its
online.\n" \
                "The mission of making nutritious food accessible to everyone. NutriCrave's system ,\n" \
                "allowing customers to order and purchase food without having to leave their homes.\n"\
                "NutriCrave's program is easily modifiable to reflect changes in food prices on the \n"\
                "menu without requiring excessive time and effort. The company's food ordering system \n"\
                "is built using programming techniques such as graphical user interface to create a competent and
efficient platform."

            # Create a new window to display the About Us information
            about_us_window = tk.Toplevel(window)
            about_us_window.title("About Us")
            about_us_window.geometry('1100x500')
```

```
    about_us_window.configure(bg='burlywood4')

    about_us_label = tk.Label(about_us_window,
                              text=about_us_text,
                              font=('Maiandra GD', 14, 'bold'),
                              justify=tk.LEFT,
                              bg='burlywood4',
                              fg='white')

    about_us_label.pack(pady=20)

    close_button = tk.Button(about_us_window,
                             text="Close",
                             command=about_us_window.destroy)
    close_button.pack(pady=10)

# Creating pull-down menu
menu_bar = tk.Menu(window)
window.config(menu=menu_bar)

LearnMore_menu = tk.Menu(menu_bar, tearoff=0)
menu_bar.add_cascade(label="Learn More", menu=LearnMore_menu)

# Adding "About Us" option to the file menu
LearnMore_menu.add_command(label="About Us", command=about_us)

# Adding a separator after "About Us"
LearnMore_menu.add_separator()

def contact_us():
    contact_us_text = "Contact Us:\n\n" \
                      "Email: info@nutricrave.com\n" \
                      "Phone: +60 3597415\n" \
                      "Address: 123 Nutri Lane, Kuantan, Pahang"
```

```
    # Create a new window to display the Contact Us information
    contact_us_window = tk.Toplevel(window)
    contact_us_window.title("Contact Us")
    contact_us_window.geometry('1000x500')
    contact_us_window.configure(bg='burlywood4')

    img = ImageTk.PhotoImage(Image.open('logo.png').resize((300, 300)))
    lb2 = tk.Label(contact_us_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)

    contact_us_label = tk.Label(contact_us_window,
                                text=contact_us_text,
                                font=('Maiandra GD', 14, 'bold'),
                                justify=tk.LEFT,
                                bg='burlywood4',
                                fg='white')
    # Adjust the x and y coordinates to position the text label
    contact_us_label.place(x=400, y=100)

    # img = ImageTk.PhotoImage(Image.open('logo.png').resize((300, 300)))
    # lb2 = tk.Label(contact_us_window, image=img, bg='dark khaki')
    # lb2.image = img
    # lb2.place(x=50, y=60)

    # contact_us_label.pack(pady=20)

    close_button = tk.Button(contact_us_window,
                             text="Close",
                             command=contact_us_window.destroy)
    close_button.pack(pady=230)

# Adding "Contact Us" option to the NutriCrave menu
```

```
LearnMore_menu.add_command(label="Contact Us", command=contact_us)

# PART 3

# Creating listbox
# Initialize the list and NumPy array as global variables
chosen_values = []
values_array = np.array([])
# List of food items for the listbox
food_options = ['Brown Rice With Chicken', 'Brown Rice With Mushrooms', 'Creamy Lemon Shrimp Spaghetti',
                'Smoked Salmon Spaghetti',
                'Chicken Tacos With Avacado', 'Spicy Salmon Bowl', 'Daily Special Salad', 'Apple Pie',
                'Orange Juice',
                'Watermelon Juice']
# Function to handle listbox selection and store all chosen values in a array to show in cart
global sumprice, y_inc

def store_value():

    selected_index = food_listbox.curselection()

    if selected_index:
        selected_food = MenuDf.iloc[selected_index][0]

        # Append the selected food to the list
        chosen_values.append(selected_food)

        # Convert the list to a NumPy array
        values_array = np.array(chosen_values)
        if len(values_array) <= 15:
            # To display number of item in cart
            display_label = tk.Label(text="Your cart has 0 items.",
                                     bg='khaki3',
                                     font=('Gill Sans MT', 13))
```

```
        v = tk.StringVar()
        r1 = tk.Radiobutton(text='Cash', variable=v, value='Python', tristatevalue=0,bg='burlywood4',font=6)
        r2 = tk.Radiobutton(text='E-Wallet', variable=v, value='Tableau',
tristatevalue=0,bg='burlywood4',font=5)
        r3 = tk.Radiobutton(text='Debit/Credit Card', variable=v, value='C',
tristatevalue=0,bg='burlywood4',font=5)


        r1.place(x=950,y=660)
        r2.place(x=1035, y=660)
        r3.place(x=1150, y=660)




        # Display all chosen values and the NumPy array in the window
        display_label.config(text=f"Your cart has {len(chosen_values)} items.")
        # display_label.config(text=f"Your Cart: {', '.join(chosen_values)}")

        display_label.place(x=1060, y=740)

        y_inc = 130
        sumprice = 0
        rcptext = tk.Label(text='Your Cart',
                    font=('Gill Sans MT', 16),
                    width=20,
                    height=1,
                    bg='burlywood2')

        bg = tk.Label(width=50,
                height=35,
                bg='white')
        bg.place(x=950, y=120)
        rcptext.place(x=1020, y=80)
```

```
            for x in range(0, len(values_array)):
                for y in range(0, len(MenuDf)):
                    if values_array[x] == MenuDf.iloc[y][0]:
                        FinalOrder = tk.Label(text=f"{MenuDf.iloc[y][0]} RM {MenuDf.iloc[y][1]}",
                                              font=('Gill Sans MT', 12),
                                              bg='burlywood2')
                        FinalOrder.place(x=1000, y=y_inc)
                        y_inc += 30
                        sumprice += MenuDf.iloc[y][1]

            def PrintFinal():
                FinalMsg = tk.Label(text=f"Dear {username},\n"
                                         f"Your total pay is RM{sumprice}.\n",
                                    font=('Gill Sans MT', 13),
                                    bg='burlywood2')
                FinalMsg.place(x=1000, y=y_inc)

            OrderButton = tk.Button(text='Finalize Your Order',
                                    font=('Times New Roman', 14),
                                    width=20,
                                    height=1,
                                    command=PrintFinal)
            OrderButton.place(x=1030, y=700)
        else:
            warn = tk.messagebox.showwarning(title='Max limit reached',
                                             message='A maximum of 15 items are allowed in your cart!')



def display_nutrition():
    global chosen_values, values_array  # Declare variables as global

    selected_index = food_listbox.curselection()

    if selected_index:
```

```
selected_food = MenuDf.iloc[selected_index][0]

new_window = tk.Toplevel(window)
new_window.title(f"{selected_food} - Nutrition Values")
new_window.geometry('2000x1500')
new_window.configure(bg='dark khaki')
fig, ax = plt.subplots()
canvas = FigureCanvasTkAgg(fig, master=new_window)
canvas.get_tk_widget().place(x=700, y=20)
x = ['Carbohydrates', 'Protein', 'Fat']
y = [MenuDf.iloc[selected_index][3], MenuDf.iloc[selected_index][4], MenuDf.iloc[selected_index][5]]
total = sum(y)
ax.pie(y, labels=x, autopct=lambda p: '{:.1f}'.format(p * total / 100))
fig.set_facecolor('lightgray')
canvas.draw()

if selected_food == 'Brown Rice With Chicken':
    lb = tk.Label(new_window,
                  text=MenuDf.iloc[0][6],
                  font=('Bahnschrift SemiLight', 16),
                  bg='dark khaki')
    lb.place(x=230, y=550)
    img = ImageTk.PhotoImage(Image.open('brown rice with chicken.png').resize((500, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Brown Rice With Mushrooms':
    lb = tk.Label(new_window,
                  text=MenuDf.iloc[1][6],
                  font=('Bahnschrift SemiLight', 16),
                  bg='dark khaki')
    lb.place(x=150, y=550)
    img = ImageTk.PhotoImage(Image.open('brown rice with mushrooms.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
```

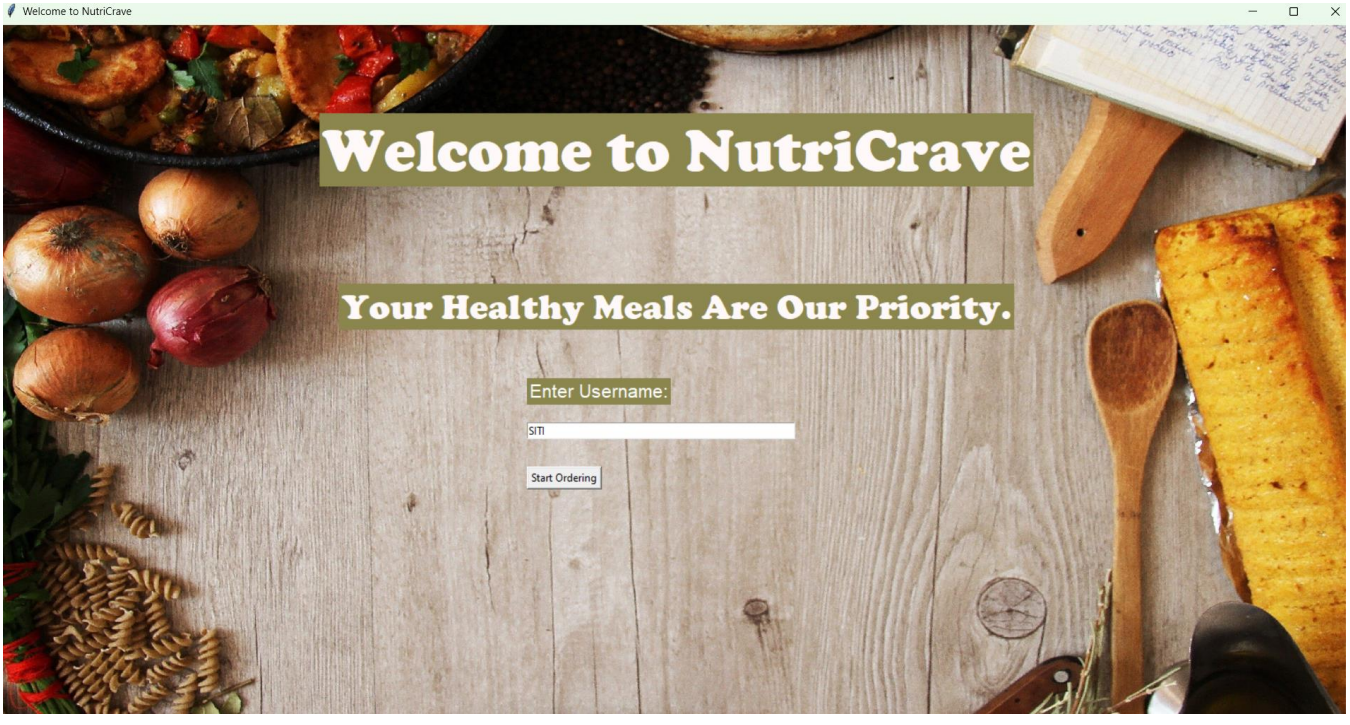
```
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Creamy Lemon Shrimp Spaghetti':
    lb = tk.Label(new_window,
                  text=MenuDf.iloc[2][6],
                  font=('Bahnschrift SemiLight', 16),
                  bg='dark khaki')
    lb.place(x=230, y=550)
    img = ImageTk.PhotoImage(Image.open('creamy lemon shrimp spaghetti.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Smoked Salmon Spaghetti':
    lb = tk.Label(new_window,
                  text=MenuDf.iloc[3][6],
                  font=('Bahnschrift SemiLight', 16),
                  bg='dark khaki')
    lb.place(x=100, y=550)
    img = ImageTk.PhotoImage(Image.open('smoked salmon spaghetti.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Chicken Tacos With Avocado':
    lb = tk.Label(new_window,
                  text=MenuDf.iloc[4][6],
                  font=('Bahnschrift SemiLight', 16),
                  bg='dark khaki')
    lb.place(x=230, y=550)
    img = ImageTk.PhotoImage(Image.open('chicken tacos with avacado.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Spicy Salmon Bowl':
    lb = tk.Label(new_window,
```

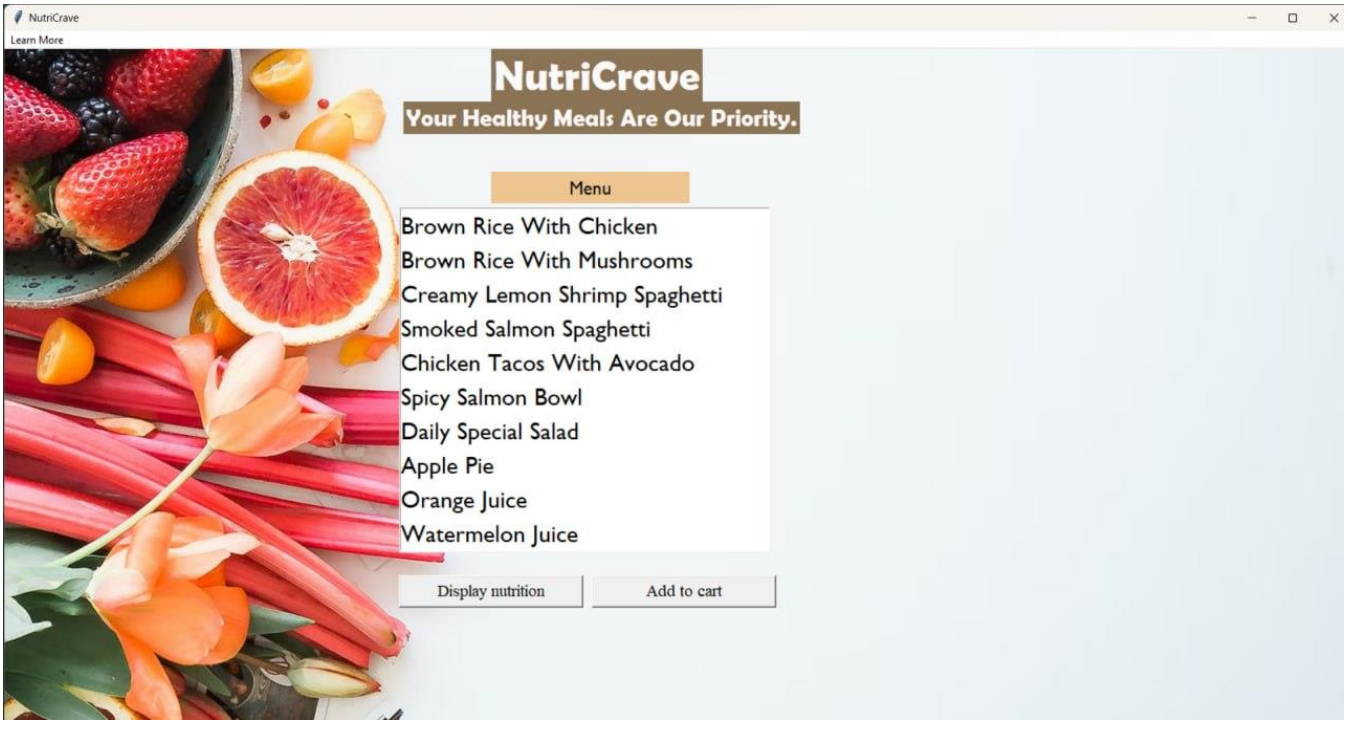
```
                text=MenuDf.iloc[5][6],
                font=('Bahnschrift SemiLight', 16),
                bg='dark khaki')
    lb.place(x=230, y=550)
    img = ImageTk.PhotoImage(Image.open('spicy salmon bowl.png').resize((400, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=150, y=60)
elif selected_food == 'Daily Special Salad':
    lb = tk.Label(new_window,
                text=MenuDf.iloc[6][6],
                font=('Bahnschrift SemiLight', 16),
                bg='dark khaki')
    lb.place(x=100, y=550)
    img = ImageTk.PhotoImage(Image.open('daily special salad.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Apple Pie':
    lb = tk.Label(new_window,
                text=MenuDf.iloc[7][6],
                font=('Bahnschrift SemiLight', 16),
                bg='dark khaki')
    lb.place(x=150, y=550)
    img = ImageTk.PhotoImage(Image.open('apple pie.png').resize((600, 400)))
    lb2 = tk.Label(new_window, image=img, bg='dark khaki')
    lb2.image = img
    lb2.place(x=50, y=60)
elif selected_food == 'Orange Juice':
    lb = tk.Label(new_window,
                text=MenuDf.iloc[8][6],
                font=('Bahnschrift SemiLight', 16),
                bg='dark khaki')
    lb.place(x=100, y=550)
```

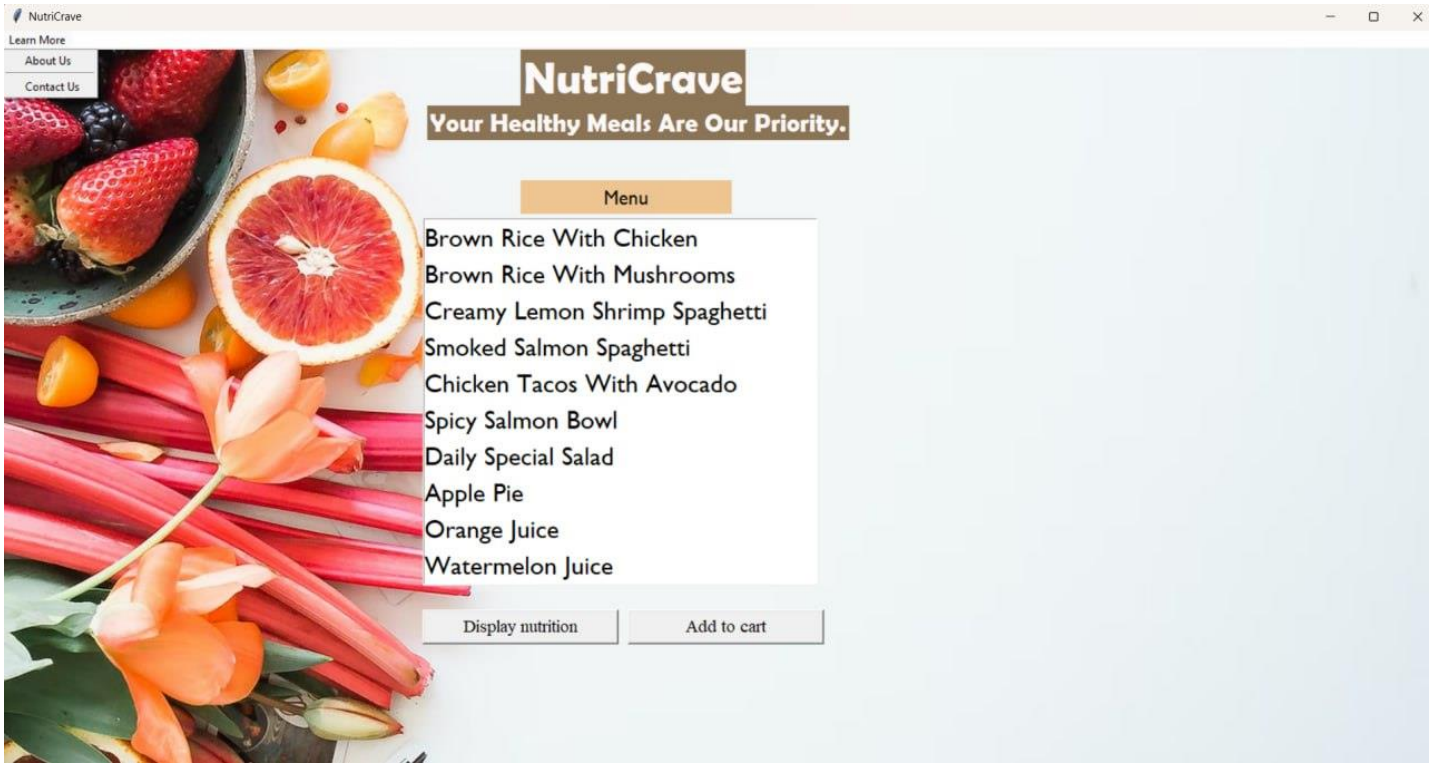
```
            img = ImageTk.PhotoImage(Image.open('orange juice.png').resize((600, 400)))
            lb2 = tk.Label(new_window, image=img, bg='dark khaki')
            lb2.image = img
            lb2.place(x=50, y=60)
        elif selected_food == 'Watermelon Juice':
            lb = tk.Label(new_window,
                          text=MenuDf.iloc[9][6],
                          font=('Bahnschrift SemiLight', 16),
                          bg='dark khaki')
            lb.place(x=230, y=550)
            img = ImageTk.PhotoImage(Image.open('watermelon juice.png').resize((600, 400)))
            lb2 = tk.Label(new_window, image=img, bg='dark khaki')
            lb2.image = img
            lb2.place(x=50, y=60)

        return_button = tk.Button(new_window,
                                  text="Return",
                                  width=20,
                                  height=1,
                                  font=('Times New Roman', 14),
                                  command=new_window.destroy)
        return_button.place(x=580, y=650)

# Creating a listbox for user selection
food_listbox = tk.Listbox(selectmode=tk.SINGLE,
                          exportselection=False,
                          font=('Gill Sans MT', 20),
                          width=30,
                          height=10)
# this enters the list of food from our dataframe into the listbox
for x in FoodBevData:
    food_listbox.insert(tk.END, x)
# Button to add to cart
store_button = tk.Button(text="Add to cart",
```

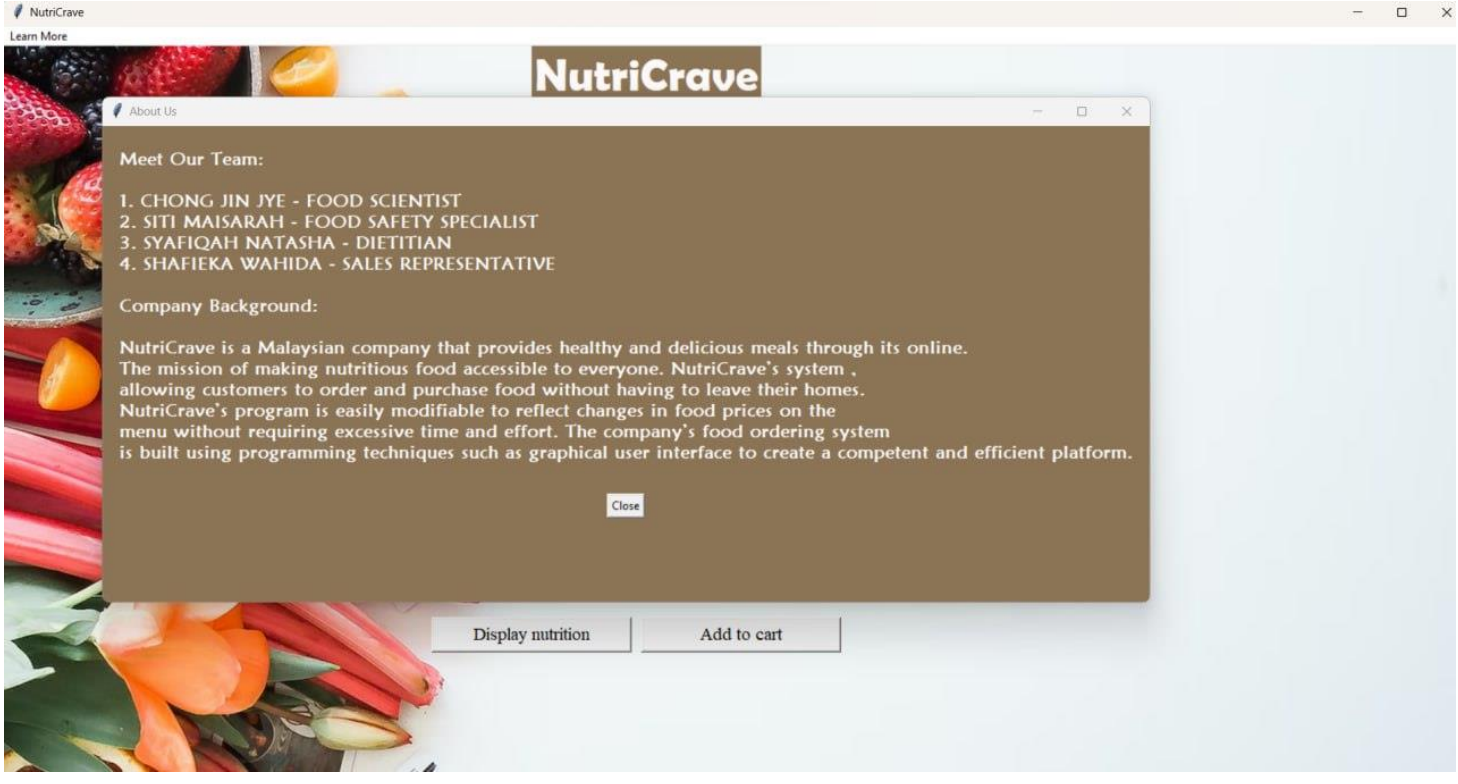
```
                    width=20,
                    height=1,
                    font=('Times New Roman', 14),
                    command=store_value)
    # Label to display number of items

    # Placing the listbox, button, and labels in the window
    food_listbox.place(x=450, y=180)
    store_button.place(x=670, y=600)

    menulb = tk.Label(text='Menu',
                font=('Gill Sans MT', 16),
                width=20,
                height=1,
                bg='burlywood2')
    menulb.place(x=555, y=140)

    PVButton = tk.Button(text='Display nutrition',
                  font=('Times New Roman', 14),
                  width=20,
                  height=1,
                  command=display_nutrition)
    PVButton.place(x=450, y=600)

    window.mainloop()

intro_window.mainloop()
```

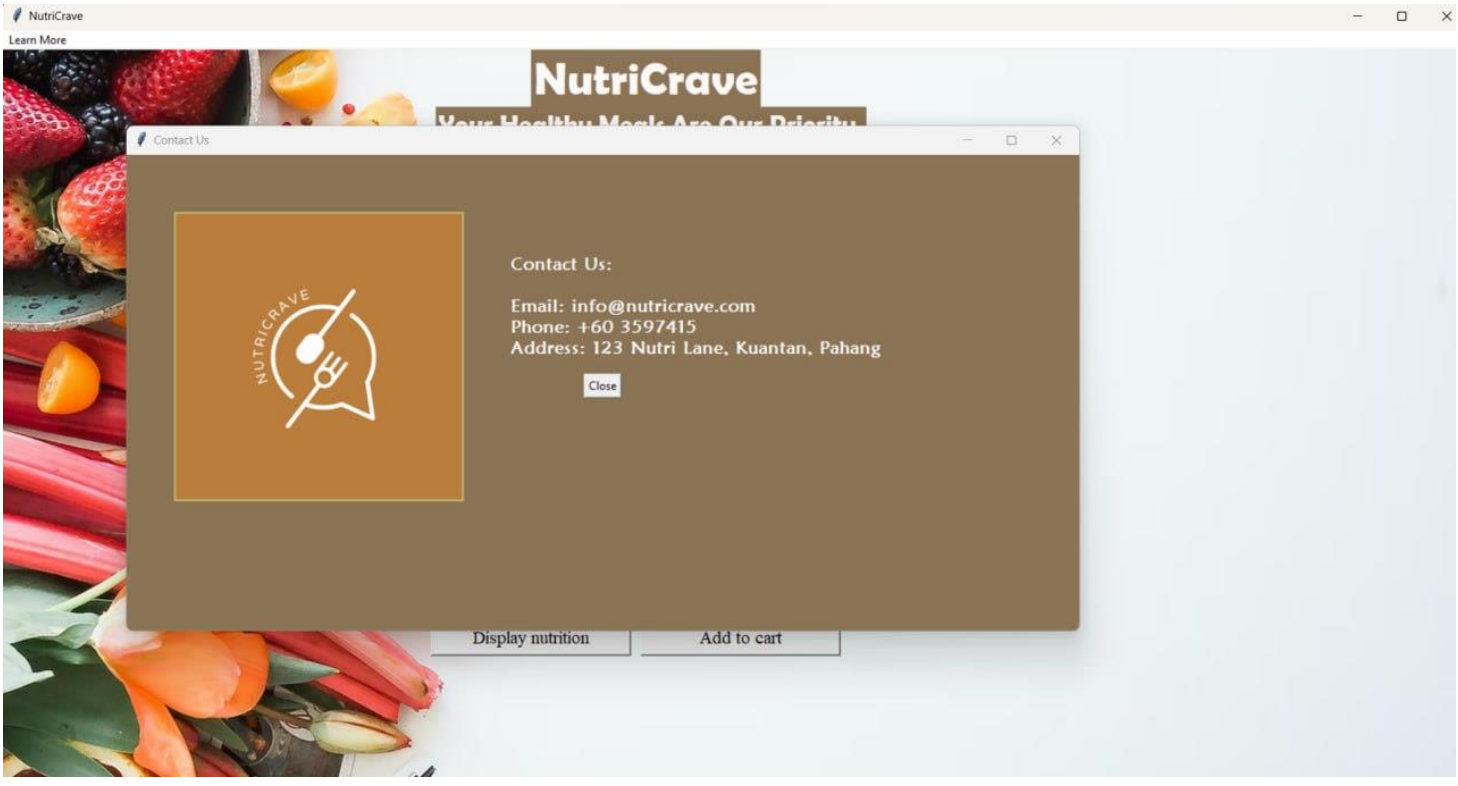
4.0 The screenshot for each activity in the GUI

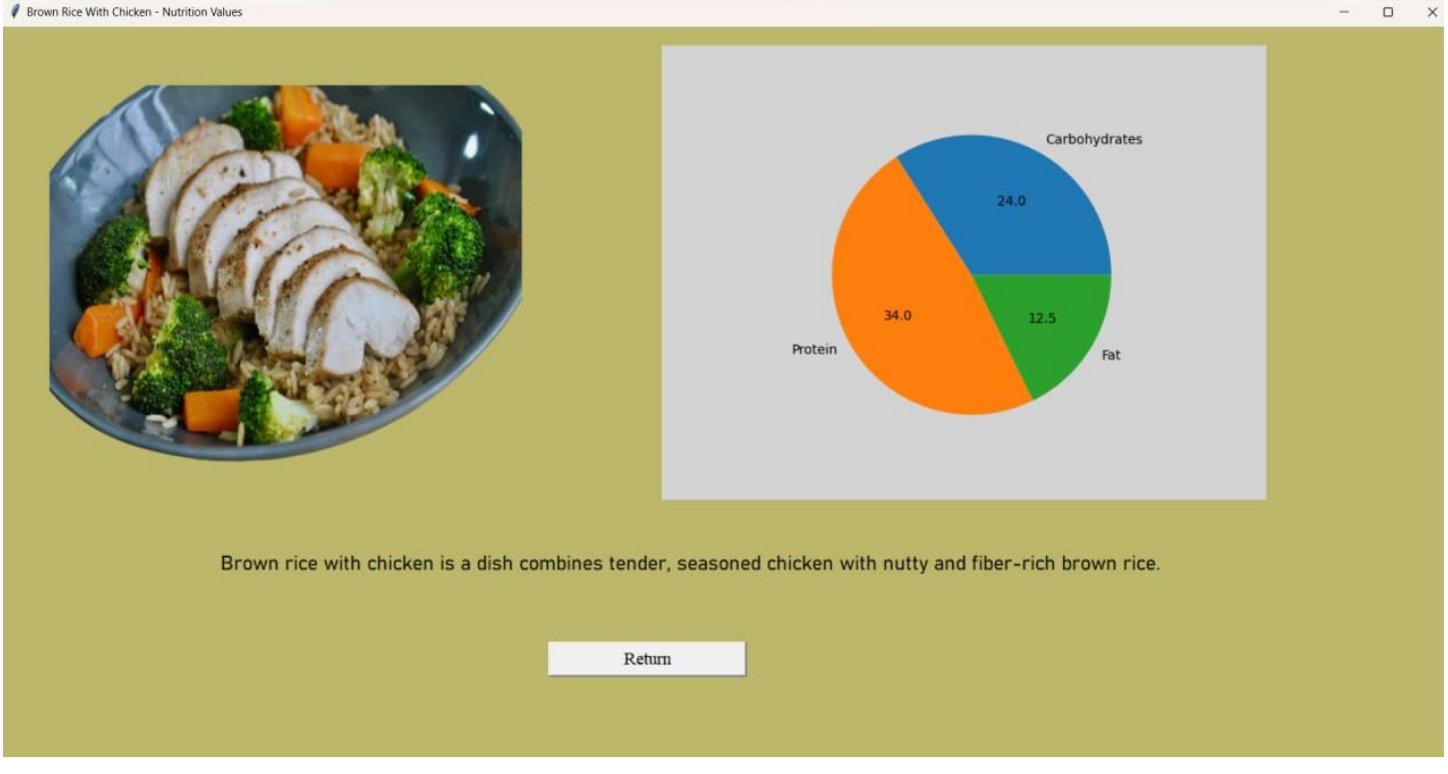




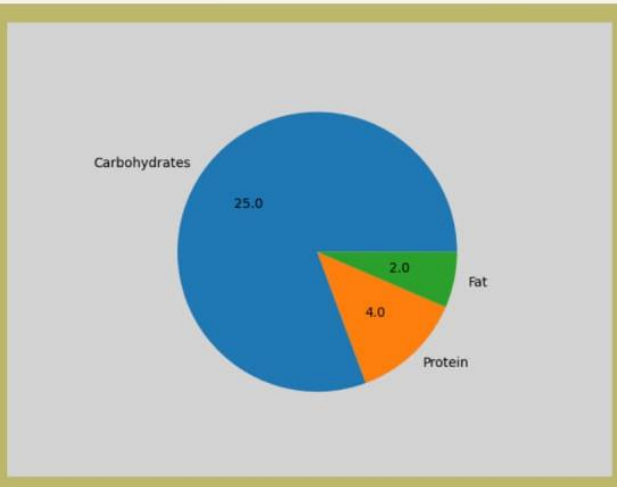

NutriCrave
Learn More
About Us
Contact Us
NutriCrave
Your Healthy Meals Are Our Priority.
Menu
Brown Rice With Chicken
Brown Rice With Mushrooms
Creamy Lemon Shrimp Spaghetti
Smoked Salmon Spaghetti
Chicken Tacos With Avocado
Spicy Salmon Bowl
Daily Special Salad
Apple Pie
Orange Juice
Watermelon Juice
Display nutrition
Add to cart


NutriCrave
Learn More
NutriCrave
About Us
Meet Our Team:
1. CHONG JIN JYE - FOOD SCIENTIST
2. SITI MAISARAH - FOOD SAFETY SPECIALIST
3. SYAFIQAH NATASHA - DIETITIAN
4. SHAFIEKA WAHIDA - SALES REPRESENTATIVE
Company Background:
NutriCrave is a Malaysian company that provides healthy and delicious meals through its online.
The mission of making nutritious food accessible to everyone. NutriCrave's system ,
allowing customers to order and purchase food without having to leave their homes.
NutriCrave's program is easily modifiable to reflect changes in food prices on the
menu without requiring excessive time and effort. The company's food ordering system
is built using programming techniques such as graphical user interface to create a competent and efficient platform.
Close
Display nutrition
Add to cart


NutriCrave
Learn More
NutriCrave
Contact Us
NUTRICRAVE
Contact Us:
Email: info@nutricrave.com
Phone: +60 3597415
Address: 123 Nutri Lane, Kuantan, Pahang
Close
Display nutrition
Add to cart


Brown Rice With Chicken - Nutrition Values
Carbohydrates
24.0
34.0
12.5
Protein
Fat
Brown rice with chicken is a dish combines tender, seasoned chicken with nutty and fiber-rich brown rice.
Return

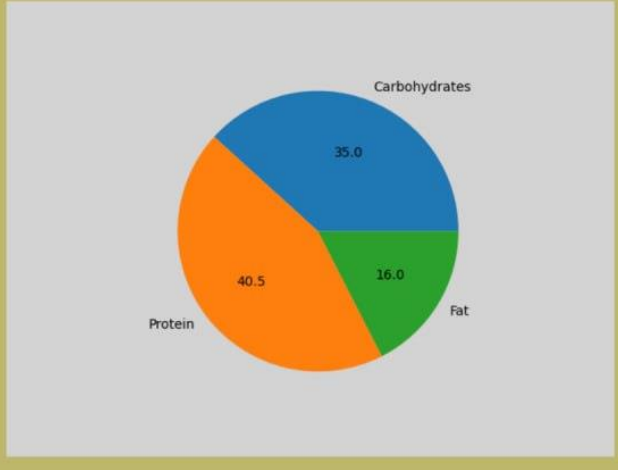
Brown Rice With Mushrooms - Nutrition Values



Brown rice and mushrooms create a flavors of nutty brown rice complement the savory richness of sautéed mushrooms

Return

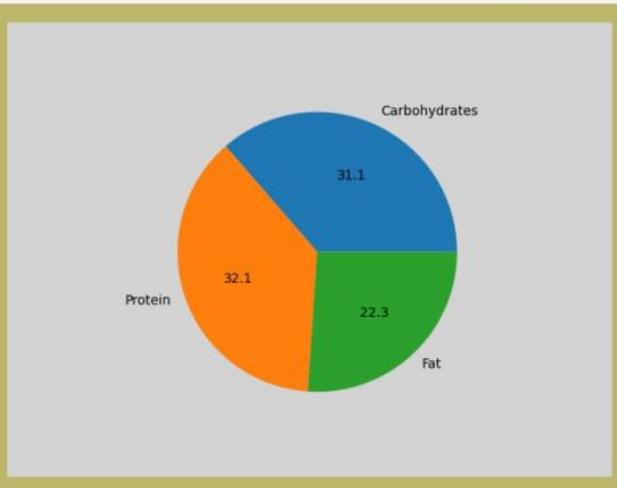
Creamy Lemon Shrimp Spaghetti - Nutrition Values



Creamy lemon shrimp spaghetti is a succulent shrimp, al dente spaghetti, and a velvety, citrus-infused cream sauce

Return

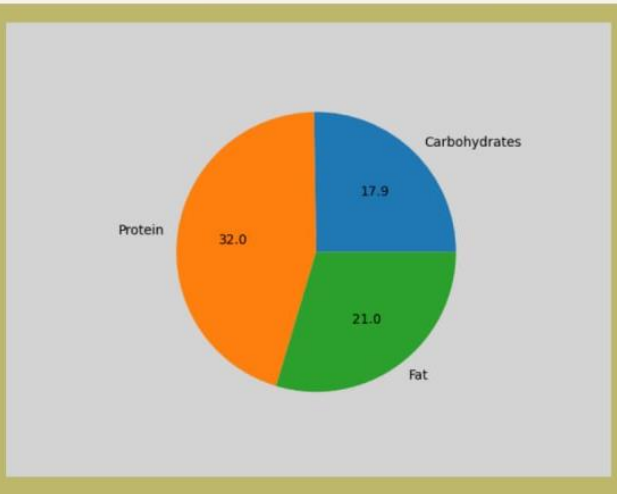
Smoked Salmon Spaghetti - Nutrition Values



Smoked salmon spaghetti is a combines the delicate smokiness of cured salmon with spaghetti, often accompanied by a creamy dill

Return

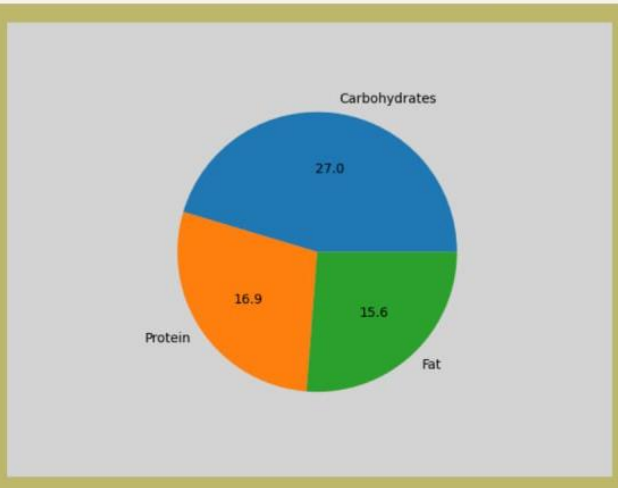
Chicken Tacos With Avocado - Nutrition Values



Chicken with avocado is a delightful dish featuring tender, seasoned chicken paired with creamy slices of avocado

Return

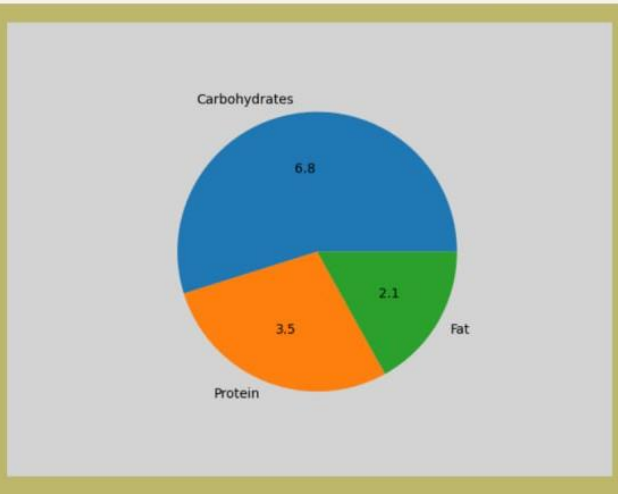
Spicy Salmon Bowl - Nutrition Values



A spicy salmon bowl is a vibrant and flavorful dish that features succulent, spicy-grilled salmon served over a bed of rice

Return

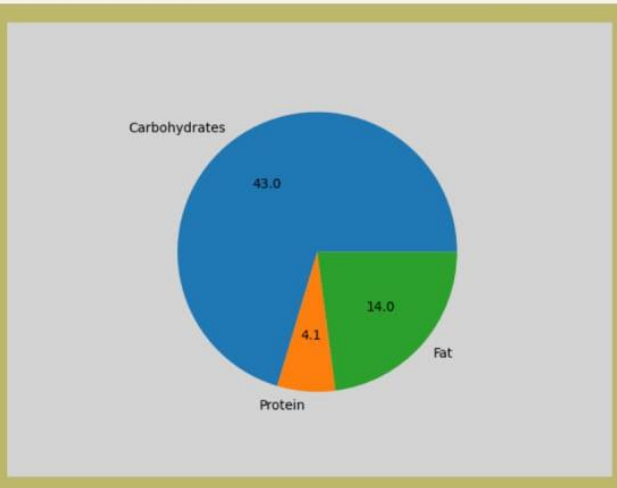
Daily Special Salad - Nutrition Values



Salad is a mixed greens, vibrant vegetables, and often complemented by toppings such as nuts, seeds, or cheese, drizzled with a flavorful dressing

Return

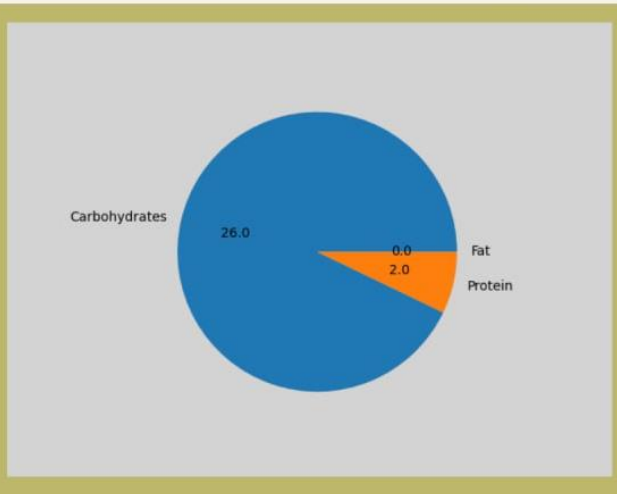
Apple Pie - Nutrition Values



Apple pie is a classic dessert, golden-brown crust enveloping sweet and spiced apple slices, often accompanied by a hint of cinnamon

Return

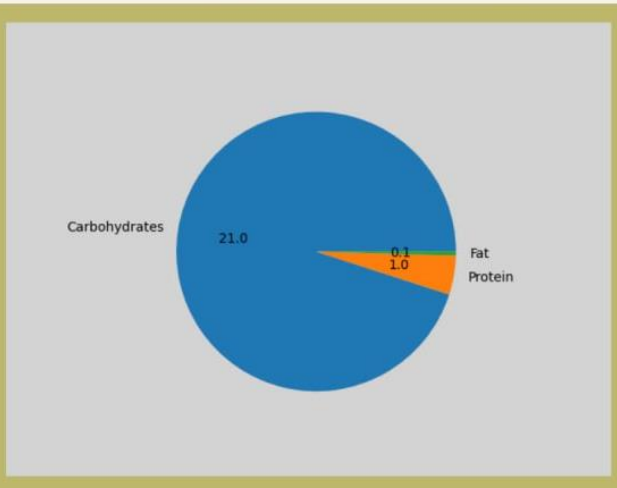
Orange Juice - Nutrition Values



Orange juice is a refreshing beverage made by extracting the liquid from ripe oranges, delivering a zesty and naturally sweet flavor

Return

Watermelon Juice - Nutrition Values



Watermelon juice is a hydrating and invigorating beverage crafted by blending or juicing ripe watermelon

Return

NutriCrave

Learn More

NutriCrave

Your Healthy Meals Are Our Priority.

Menu

- Brown Rice With Chicken
- Brown Rice With Mushrooms
- Creamy Lemon Shrimp Spaghetti
- Smoked Salmon Spaghetti
- Chicken Tacos With Avocado
- Spicy Salmon Bowl
- Daily Special Salad
- Apple Pie
- Orange Juice
- Watermelon Juice

Display nutrition | Add to cart

Your Cart

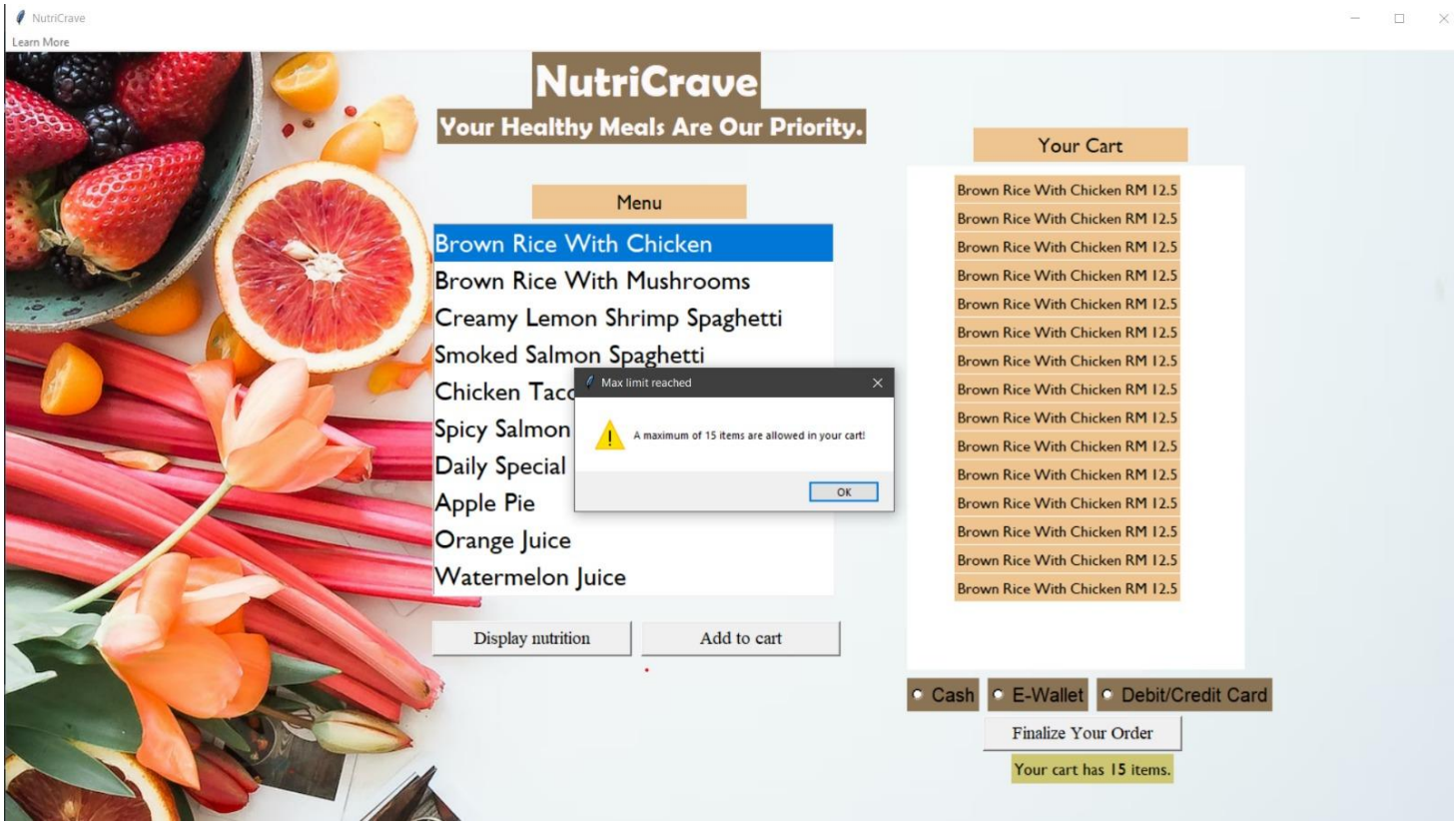
Orange Juice RM 4.0

Dear SITI,
Your total pay is RM4.0.

Cash | E-Wallet | Debit/Credit Card

Finalize Your Order

Your cart has 1 items.


NutriCrave
Learn More
NutriCrave
Your Healthy Meals Are Our Priority.
Menu
Brown Rice With Chicken
Brown Rice With Mushrooms
Creamy Lemon Shrimp Spaghetti
Smoked Salmon Spaghetti
Chicken Taco
Spicy Salmon
Daily Special
Apple Pie
Orange Juice
Watermelon Juice
Max limit reached
A maximum of 15 items are allowed in your cart!
OK
Display nutrition
Add to cart
Your Cart
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Brown Rice With Chicken RM 12.5
Cash
E-Wallet
Debit/Credit Card
Finalize Your Order
Your cart has 15 items.

MARKING SCHEME

CLO	Description	PLO mapping	Percentage	Marks
CLO2	Use appropriate Python programming technique to solve problem.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills C3: Application	5%	10

LEVEL OF ACHIEVEMENT					
0 **None**	**1** **Inadequate**	**2 Emerging**	**3** **Developing**	**4** **Good**	**5** **Excellent**

ELEMENTS	WEIGHTAGE	SCORE
Combination of appropriate controls and layout manager: • Input controls such as buttons, toggles, checkboxes etc. • Navigation controls such as pull-down menu. • Information components eg. message boxes etc. • Tkinter geometry manager (place/pack/grid manager).	**1**	
Task execution by each controls: • Each control is labelled using short and precise words representing the task. • The task for each controls is specified and written neatly. • The task for each control executed correctly and smoothly.	**1**	
TOTAL		

CLO	Description	PLO mapping	Percentage	Marks
CLO3	Construct and run program.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	15%	30

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0 None	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Theory/ Knowledge	No theoretical knowledge is observed.	Very little knowledge provided or information is incorrect.	Some knowledge or information is provided but missing all major points.	Some knowledge or information is provided but still missing some major points.	Good knowledge is observed, missing some minor points.	Excellent knowledge is observed; provides all necessary background principles.	1	
Assembly	Fail to demonstrate the given task.	Partly demonstrate the given task with errors.	Partly demonstrate the given task with wrong output.	Partly demonstrate the given task correctly.	Fully demonstrate the given task with some wrong output.	Demonstrate the given task correctly and perfectly.	2	
Technique used / Effectiveness	Fail to demonstrate the given task.	Demonstrate inappropriate techniques.	Partly correct techniques demonstrated.	Demonstrated technique is correct but not effective or efficient.	Demonstrated technique is partly effective and efficient.	Demonstrated technique is effective and efficient.	2	
GUI	Not submitting GUI.	The GUI presented was taken from the other sources with no modifications. The GUI presented was not effective in debugging the output with a lot of errors and displayed for an inappropriate time.	The GUI presented was modified from the other sources with minimal modifications. Shows less effective debugging on the output with with several errors and displayed for less appropriate time.	The GUI presented was modified well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was modified very well from the other sources. Shows effective debugging on the output with no error and displayed for an appropriate time.	The GUI presented was originally developed. Shows effective debugging on the output with no error and displayed for an appropriate time.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO4	Work collaboratively to solve assigned task.	PLO4: Functional work skills with focus on Interpersonal skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0 None	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Foster Good Relationship	Show no good relationships and unable to work together effectively with other group members towards goal achievement.	No clear evidence of ability to foster good relationships and work together effectively with other group members towards goal achievement.	Able to foster relationship and work together with other group members towards goal achievement but with limited effect and require improvements.	Able to foster relationship and work together with other group members towards goal achievement with some effect(s) and require minor improvements.	Able to foster good relationship and work together with other group members towards goal achievement.	High ability to foster good relationship and work together effectively with other group members towards goal achievement.	1	
Alternate Roles	Show no ability to assume alternate roles as a group leader and group members.	No clear evidence of ability to assume alternate roles as a group leader and group members demonstrated in practice.	Attempt to demonstrate in practice the ability to alternate roles as a group leader and group members but with limited effect and require improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and group members with some effect(s) and require minor improvements.	Able to demonstrate in practice the ability to assume alternate roles as a group leader and a group member to achieve the same goal.	Show clear evidence to assume alternate roles as a group leader and a group member demonstrated in practice.	1	

CLO	Description	PLO mapping	Percentage	Marks
CLO5	Demonstrate innovative ideas in developing a graphical user interface.	PLO8: Entrepreneurial skills A3: Valuing	5%	10

CRITERIA	LEVEL OF ACHIEVEMENT						WEIGHTAGE	SCORE
	0 None	1 Inadequate	2 Emerging	3 Developing	4 Good	5 Excellent		
Analyzing an existing situation and identifying areas for improvement	Not providing any analysis of situation and areas for improvement were not identified.	The analysis of the situation was very limited and areas for improvement were not. identified	The analysis of the situation was limited and areas for improvement were not identified.	The analysis of the situation was appropriate but the identification of areas for improvement was limited.	The situation was appropriately analyzed and the identification of areas for improvement was completed.	The analysis of the situation and the identification of areas for improvement was completed and increases over time.	1	
Creativity/ Innovative ideas	Not presenting any GUI.	GUI presented contains lack of significance ideas, no innovative values, lack of creativity and not user friendly.	GUI presented contains lack of significance ideas, no innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains lack of significance ideas, but still have innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains significance ideas, innovative values, creative enough (catchy apps name & attractive) and user friendly.	GUI presented contains a very significance ideas, high innovative values, creative enough (catchy apps name & attractive) and user friendly.	1	

END OF GROUP PROJECT INSTRUCTIONS