Program Name goes here

Link to github Repository: (N/A)

Links to trello board / project management tools:https://trello.com/invite/b/5QYamlel/ATTIc6979de20083e48b037a0b80196811e3B885A198/91896-assesment

You MUST provide evidence showing how the problem has been decomposed, how the components have been developed and trialled, and of how they have been assembled and tested to create a final, working outcome.

Relevant Implications

The functionality of the burger ordering code refers to how well it performs its intended tasks, providing a smooth and guided process for users to make selections and complete their order. The code addresses functionality by ensuring each step such as choosing burger sizes, toppings, and payment methods is clear and easy to follow. Input validation ensures that users make correct choices, preventing mistakes like invalid sizes or improperly formatted phone numbers. Additionally, the code effectively calculates and communicates costs, offering a seamless and error-free ordering experience from start to finish.

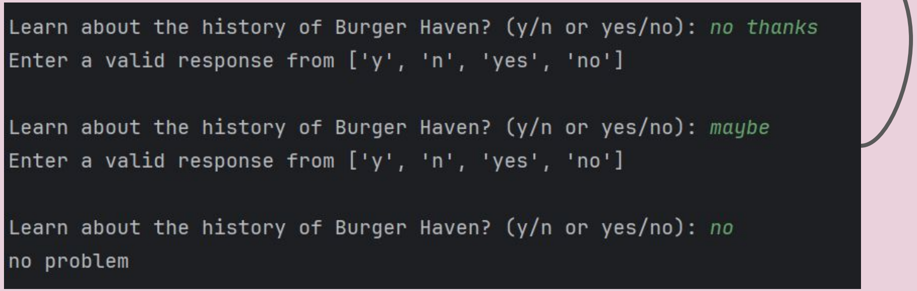
Functionality is how something works and the specific tasks it can perform, such as features in a program, tool, or system that make it useful for users. It's essentially what something is capable of doing based on its design.

A ally,

Relevant Implications

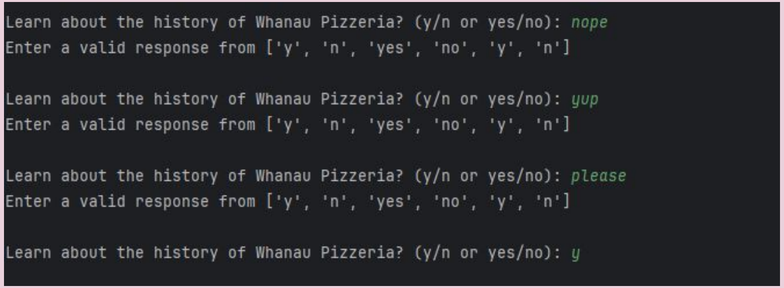
The code’s aesthetics are highlighted by its clean, organized look with well-formatted menus and engaging images. This design creates a visually appealing and user-friendly burger ordering interface. Rather than using typical loading bars, the creative images add a fun and unique touch, making the interface more enjoyable and attractive to use. This is important to my code as it can appeal to people of all ages and all over the world, as I have not put any images that will offend anybody.

The code’s usability is shown through its clear messages, effective error recovery, and easy navigation. It lets users customize their burger orders, select payment options, and review their choices. The straightforward design ensures a smooth, hassle-free experience for everyone. This is important as without error recovery the website would be unusable and users would not be able to effectively order a burger.

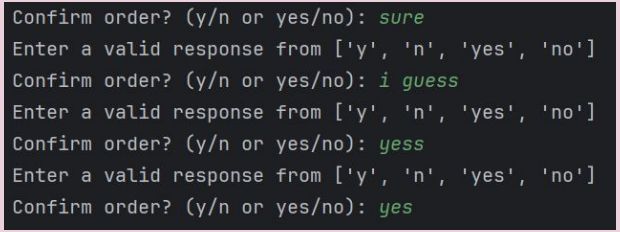


```
Learn about the history of Burger Haven? (y/n or yes/no): no thanks
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): maybe
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): no
no problem
```

Relevant implications evidence

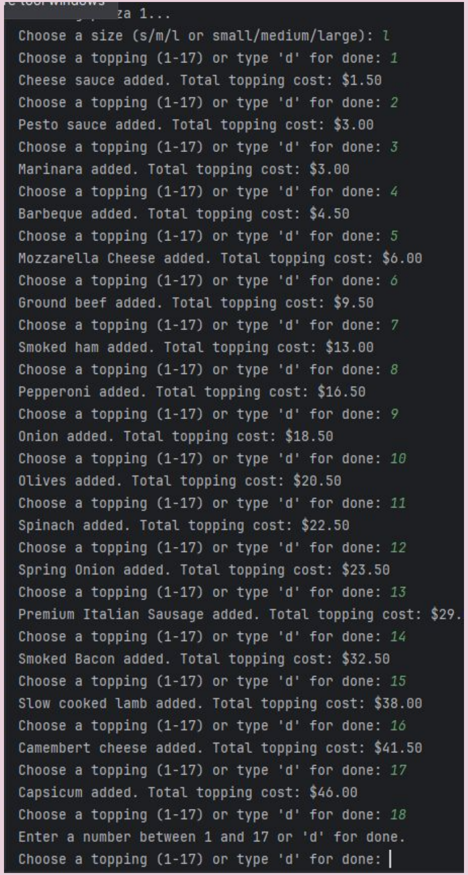


```
Learn about the history of Whanau Pizzeria? (y/n or yes/no): nope
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): yup
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): please
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): y
```

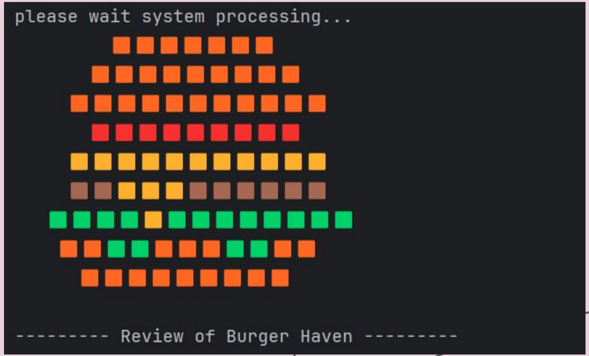


```
Confirm order? (y/n or yes/no): sure
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): i guess
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): yess
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): yes
```

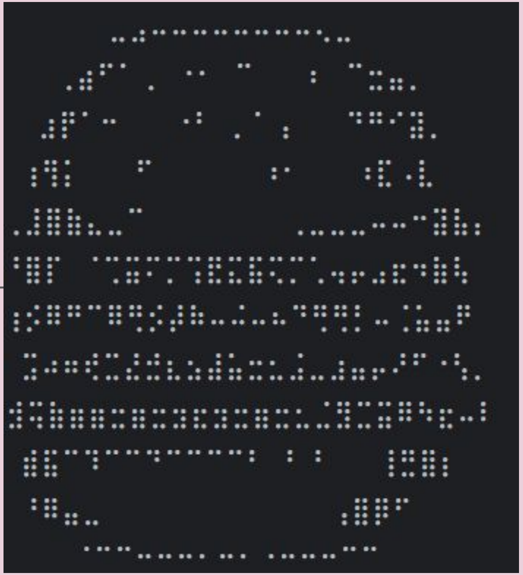
Topping choice



```
za 1...
Choose a size (s/m/l or small/medium/large): l
Choose a topping (1-17) or type 'd' for done: 1
Cheese sauce added. Total topping cost: $1.50
Choose a topping (1-17) or type 'd' for done: 2
Pesto sauce added. Total topping cost: $3.00
Choose a topping (1-17) or type 'd' for done: 3
Marinara added. Total topping cost: $3.00
Choose a topping (1-17) or type 'd' for done: 4
Barbeque added. Total topping cost: $4.50
Choose a topping (1-17) or type 'd' for done: 5
Mozzarella Cheese added. Total topping cost: $6.00
Choose a topping (1-17) or type 'd' for done: 6
Ground beef added. Total topping cost: $9.50
Choose a topping (1-17) or type 'd' for done: 7
Smoked ham added. Total topping cost: $13.00
Choose a topping (1-17) or type 'd' for done: 8
Pepperoni added. Total topping cost: $16.50
Choose a topping (1-17) or type 'd' for done: 9
Onion added. Total topping cost: $18.50
Choose a topping (1-17) or type 'd' for done: 10
Olives added. Total topping cost: $20.50
Choose a topping (1-17) or type 'd' for done: 11
Spinach added. Total topping cost: $22.50
Choose a topping (1-17) or type 'd' for done: 12
Spring Onion added. Total topping cost: $23.50
Choose a topping (1-17) or type 'd' for done: 13
Premium Italian Sausage added. Total topping cost: $29.
Choose a topping (1-17) or type 'd' for done: 14
Smoked Bacon added. Total topping cost: $32.50
Choose a topping (1-17) or type 'd' for done: 15
Slow cooked lamb added. Total topping cost: $38.00
Choose a topping (1-17) or type 'd' for done: 16
Camembert cheese added. Total topping cost: $41.50
Choose a topping (1-17) or type 'd' for done: 17
Capsicum added. Total topping cost: $46.00
Choose a topping (1-17) or type 'd' for done: 18
Enter a number between 1 and 17 or 'd' for done.
Choose a topping (1-17) or type 'd' for done: 
```



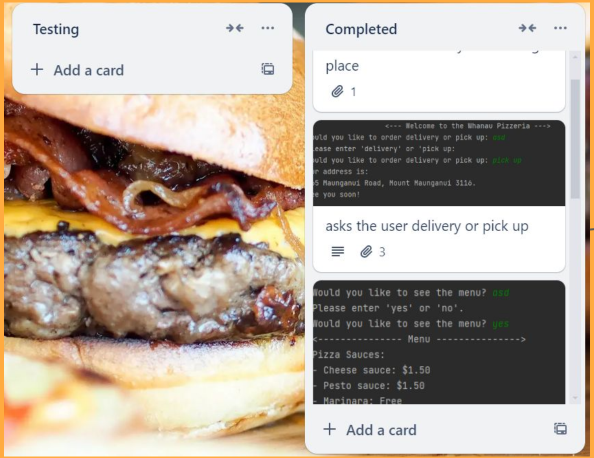
```
please wait system processing...

--------- Review of Burger Haven ---------
```



Aesthetics

Component 1 (Trello screenshot)

"Added delivery or pick-up options and fixed errors."



place

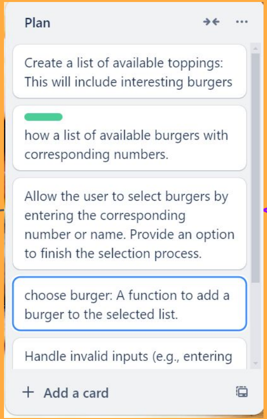
1

```
        <--- Welcome to the Whanau Pizzeria --->
uld you like to order delivery or pick up: asd
ease enter 'delivery' or 'pick up:
uld you like to order delivery or pick up: pick up
r address is:
5 Maunganui Road, Mount Maunganui 3116.
e you soon!
```

asks the user delivery or pick up

3

```
Would you like to see the menu? asd
Please enter 'yes' or 'no'.
Would you like to see the menu? yes
<--------------- Menu --------------->
Pizza Sauces:
- Cheese sauce: $1.50
- Pesto sauce: $1.50
- Marinara: Free
```



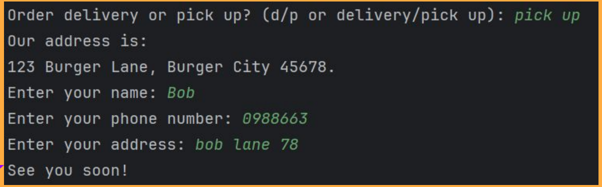
Create a list of available toppings: This will include interesting burgers

how a list of available burgers with corresponding numbers.

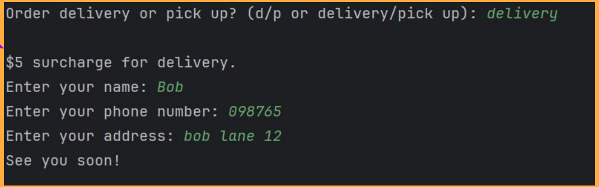
Allow the user to select burgers by entering the corresponding number or name. Provide an option to finish the selection process.

choose burger: A function to add a burger to the selected list.

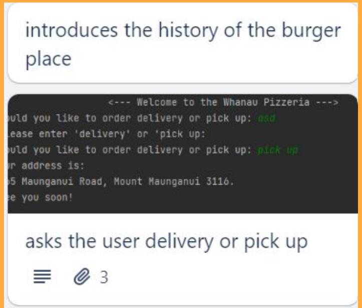
Handle invalid inputs (e.g., entering



```
Order delivery or pick up? (d/p or delivery/pick up): pick up
Our address is:
123 Burger Lane, Burger City 45678.
Enter your name: Bob
Enter your phone number: 0988663
Enter your address: bob lane 78
See you soon!
```



```
Order delivery or pick up? (d/p or delivery/pick up): delivery

$5 surcharge for delivery.
Enter your name: Bob
Enter your phone number: 098765
Enter your address: bob lane 12
See you soon!
```

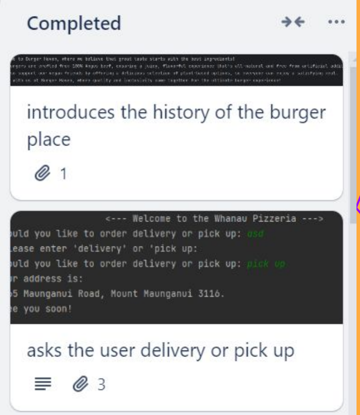


introduces the history of the burger place

```
        <--- Welcome to the Whanau Pizzeria --->
uld you like to order delivery or pick up: asd
ease enter 'delivery' or 'pick up:
uld you like to order delivery or pick up: pick up
r address is:
5 Maunganui Road, Mount Maunganui 3116.
e you soon!
```

asks the user delivery or pick up

3



introduces the history of the burger place

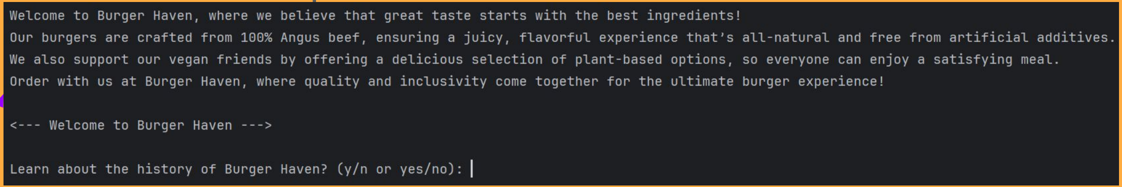
1

```
        <--- Welcome to the Whanau Pizzeria --->
uld you like to order delivery or pick up: asd
ease enter 'delivery' or 'pick up:
uld you like to order delivery or pick up: pick up
r address is:
5 Maunganui Road, Mount Maunganui 3116.
e you soon!
```

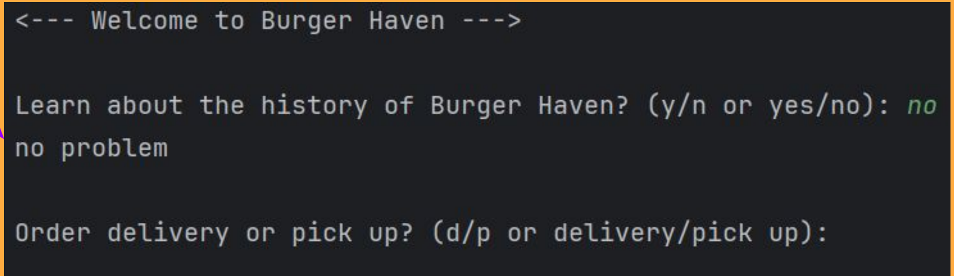
asks the user delivery or pick up

3

Introduction completed

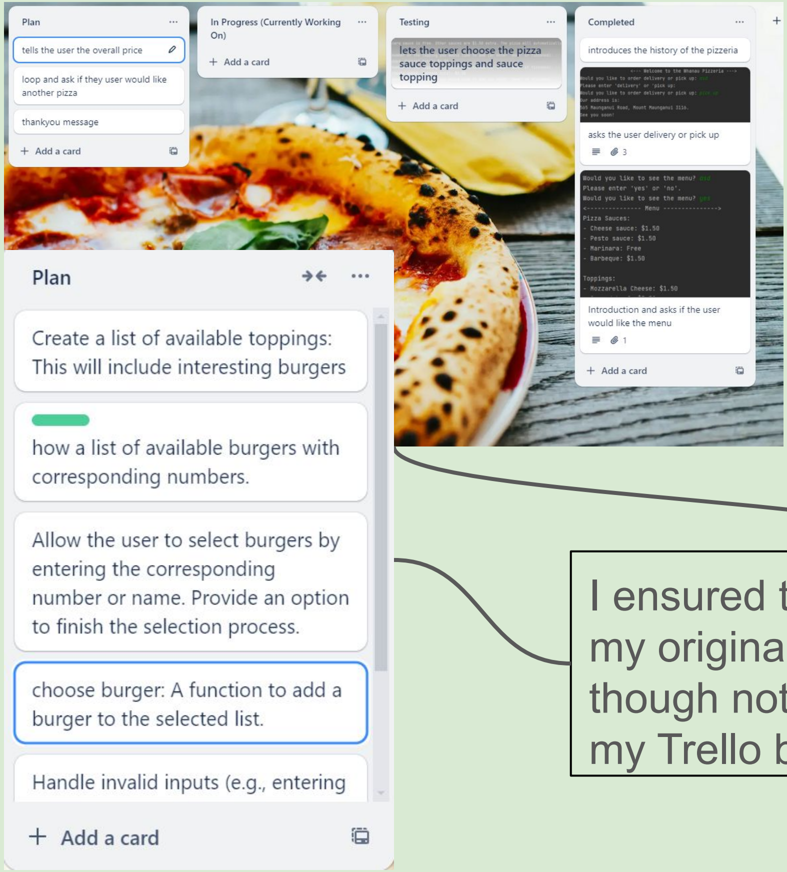


```
Welcome to Burger Haven, where we believe that great taste starts with the best ingredients!
Our burgers are crafted from 100% Angus beef, ensuring a juicy, flavorful experience that's all-natural and free from artificial additives.
We also support our vegan friends by offering a delicious selection of plant-based options, so everyone can enjoy a satisfying meal.
Order with us at Burger Haven, where quality and inclusivity come together for the ultimate burger experience!

<--- Welcome to Burger Haven --->

Learn about the history of Burger Haven? (y/n or yes/no): |
```



```
<--- Welcome to Burger Haven --->

Learn about the history of Burger Haven? (y/n or yes/no): no
no problem

Order delivery or pick up? (d/p or delivery/pick up):
```

Component final (Trello screenshot)



Plan

- tells the user the overall price
- loop and ask if they user would like another pizza
- thankyou message

In Progress (Currently Working On)

Testing

- lets the user choose the pizza sauce toppings and sauce topping

Completed

- introduces the history of the pizzeria

```
                    <--- Welcome to the Whanau Pizzeria --->
Would you like to order delivery or pick up: asd
Please enter 'delivery' or 'pick up:
Would you like to order delivery or pick up: pick up
Our address is:
565 Maunganui Road, Mount Maunganui 3116.
See you soon!
```

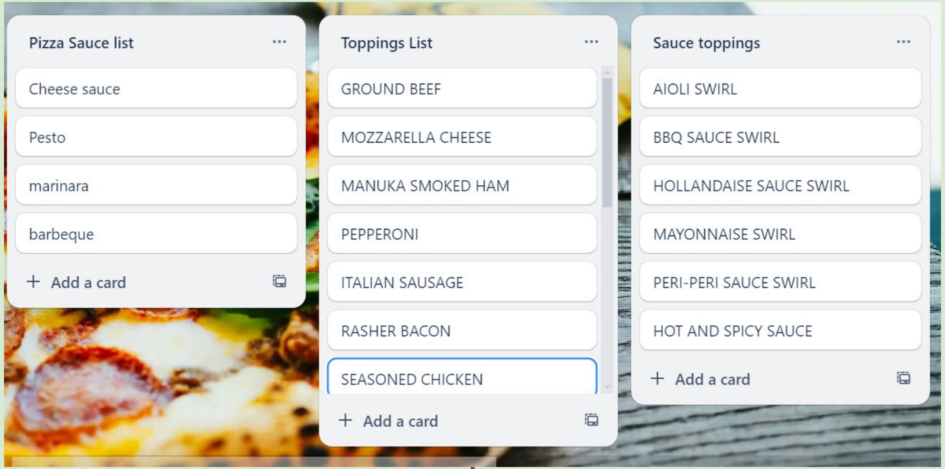
- asks the user delivery or pick up

```
Would you like to see the menu? asd
Please enter 'yes' or 'no'.
Would you like to see the menu? yes
<--------------- Menu --------------->
Pizza Sauces:
- Cheese sauce: $1.50
- Pesto sauce: $1.50
- Marinara: Free
- Barbeque: $1.50

Toppings:
- Mozzarella Cheese: $1.50
```

- Introduction and asks if the user would like the menu

Plan

- Create a list of available toppings: This will include interesting burgers
- how a list of available burgers with corresponding numbers.
- Allow the user to select burgers by entering the corresponding number or name. Provide an option to finish the selection process.
- choose burger: A function to add a burger to the selected list.
- Handle invalid inputs (e.g., entering



Pizza Sauce list

- Cheese sauce
- Pesto
- marinara
- barbeque

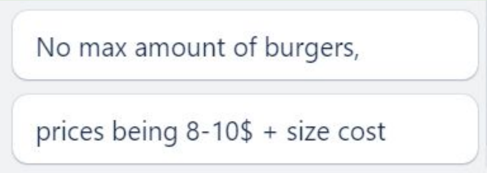
Toppings List

- GROUND BEEF
- MOZZARELLA CHEESE
- MANUKA SMOKED HAM
- PEPPERONI
- ITALIAN SAUSAGE
- RASHER BACON
- SEASONED CHICKEN

Sauce toppings

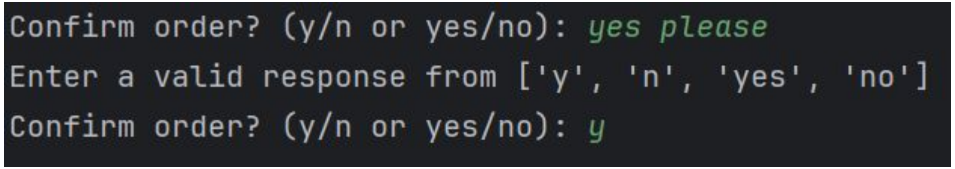
- AIOLI SWIRL
- BBQ SAUCE SWIRL
- HOLLANDAISE SAUCE SWIRL
- MAYONNAISE SWIRL
- PERI-PERI SAUCE SWIRL
- HOT AND SPICY SAUCE

I ensured that my program adhered to my original plan, incorporating several, though not all, of the toppings listed on my Trello board.

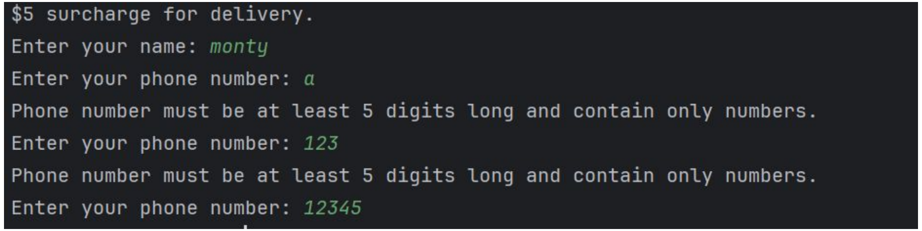


- No max amount of burgers,
- prices being 8-10$ + size cost

Error recovery system

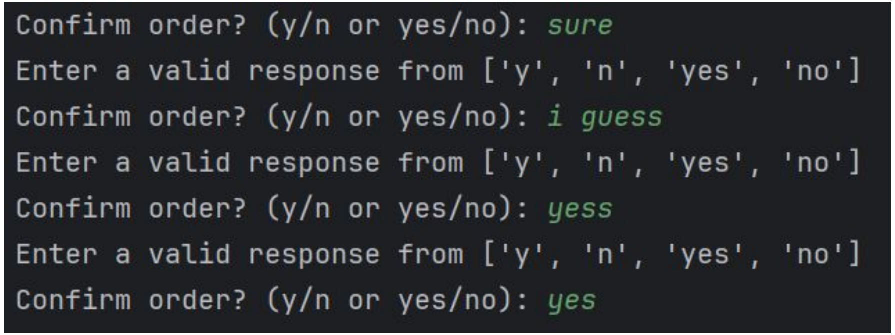


```
Confirm order? (y/n or yes/no): yes please
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): y
```

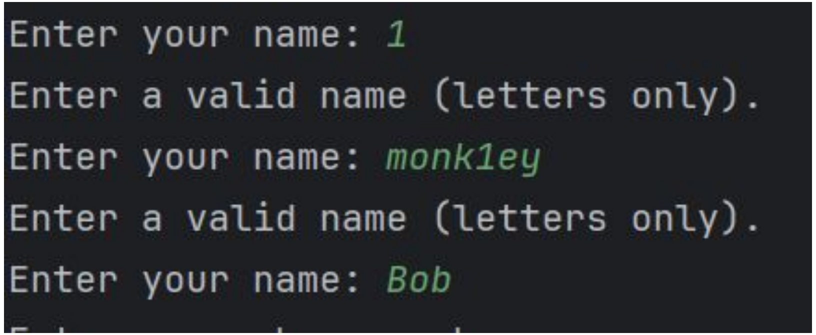


```
$5 surcharge for delivery.
Enter your name: monty
Enter your phone number: a
Phone number must be at least 5 digits long and contain only numbers.
Enter your phone number: 123
Phone number must be at least 5 digits long and contain only numbers.
Enter your phone number: 12345
```

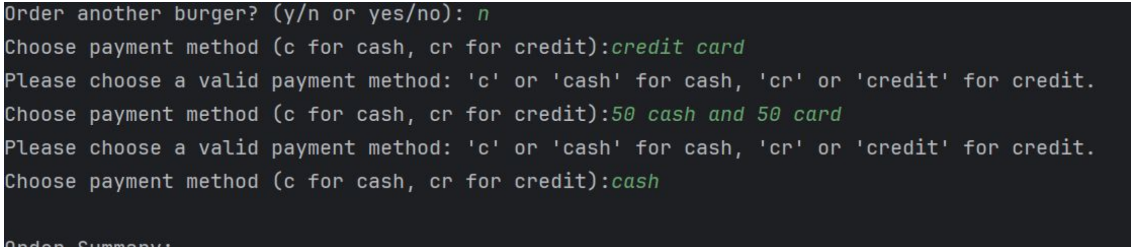
Fixed human errors such as typos and accidental inputs to provide users with a more seamless and smooth experience in my burger selection program



```
Confirm order? (y/n or yes/no): sure
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): i guess
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): yess
Enter a valid response from ['y', 'n', 'yes', 'no']
Confirm order? (y/n or yes/no): yes
```

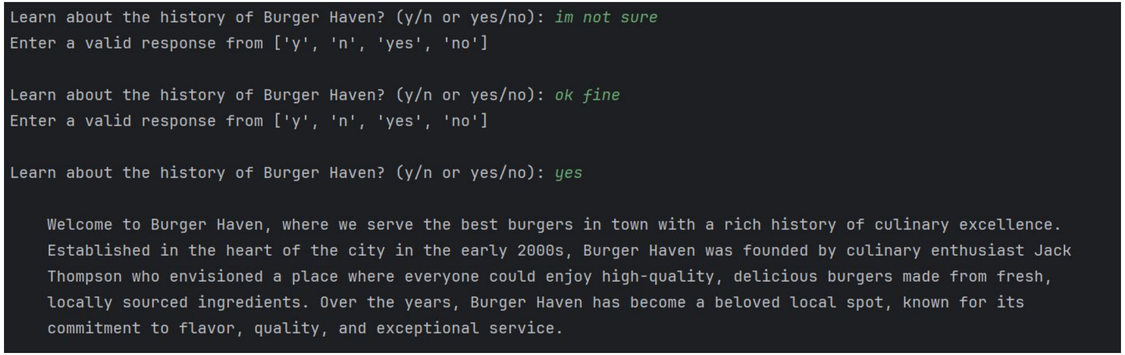


```
Enter your name: 1
Enter a valid name (letters only).
Enter your name: monk1ey
Enter a valid name (letters only).
Enter your name: Bob
```

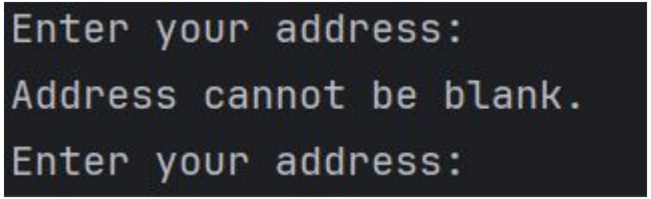


```
Order another burger? (y/n or yes/no): n
Choose payment method (c for cash, cr for credit):credit card
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):50 cash and 50 card
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):cash

Order Summary:
```

Error recovery system 🔨

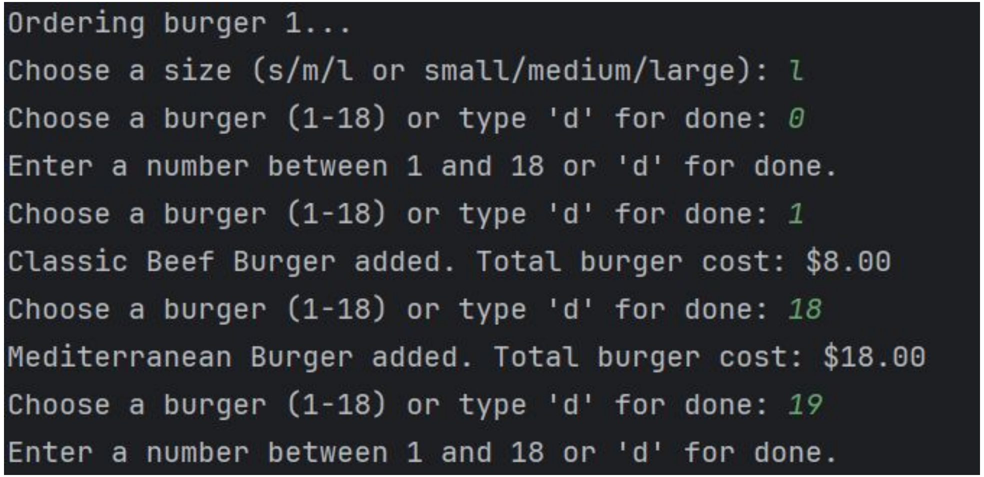


```
Learn about the history of Burger Haven? (y/n or yes/no): im not sure
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): ok fine
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): yes

    Welcome to Burger Haven, where we serve the best burgers in town with a rich history of culinary excellence.
    Established in the heart of the city in the early 2000s, Burger Haven was founded by culinary enthusiast Jack
    Thompson who envisioned a place where everyone could enjoy high-quality, delicious burgers made from fresh,
    locally sourced ingredients. Over the years, Burger Haven has become a beloved local spot, known for its
    commitment to flavor, quality, and exceptional service.
```



```
Enter your address:
Address cannot be blank.
Enter your address:
```

Not_blank function working

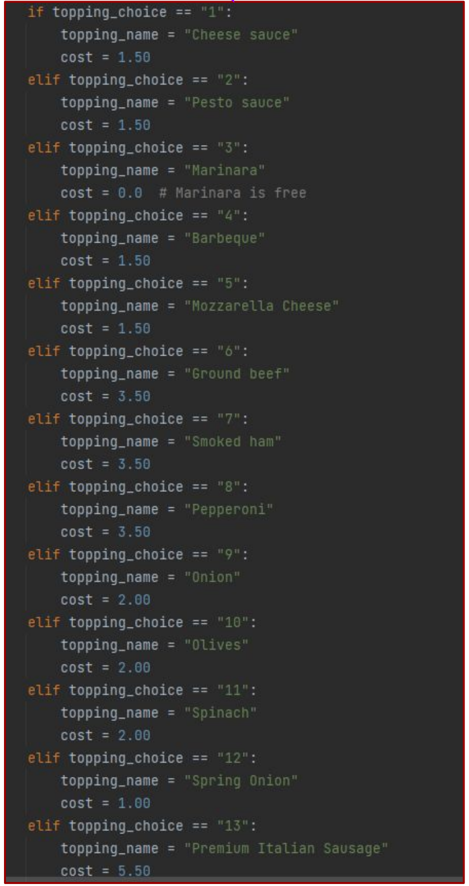


```
Ordering burger 1...
Choose a size (s/m/l or small/medium/large): l
Choose a burger (1-18) or type 'd' for done: 0
Enter a number between 1 and 18 or 'd' for done.
Choose a burger (1-18) or type 'd' for done: 1
Classic Beef Burger added. Total burger cost: $8.00
Choose a burger (1-18) or type 'd' for done: 18
Mediterranean Burger added. Total burger cost: $18.00
Choose a burger (1-18) or type 'd' for done: 19
Enter a number between 1 and 18 or 'd' for done.
```

Testing boundaries (0,1) and (18,19)

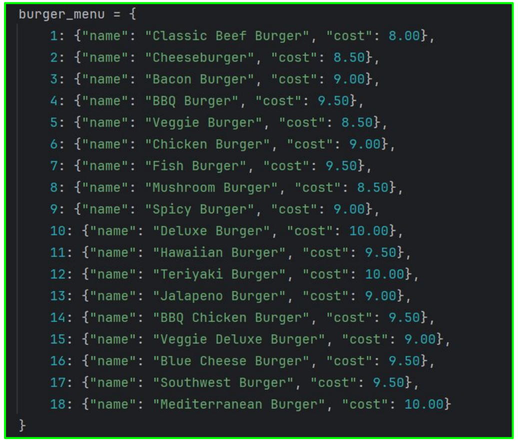
Making things more efficient (1) 🔍

Version 1 pizza version

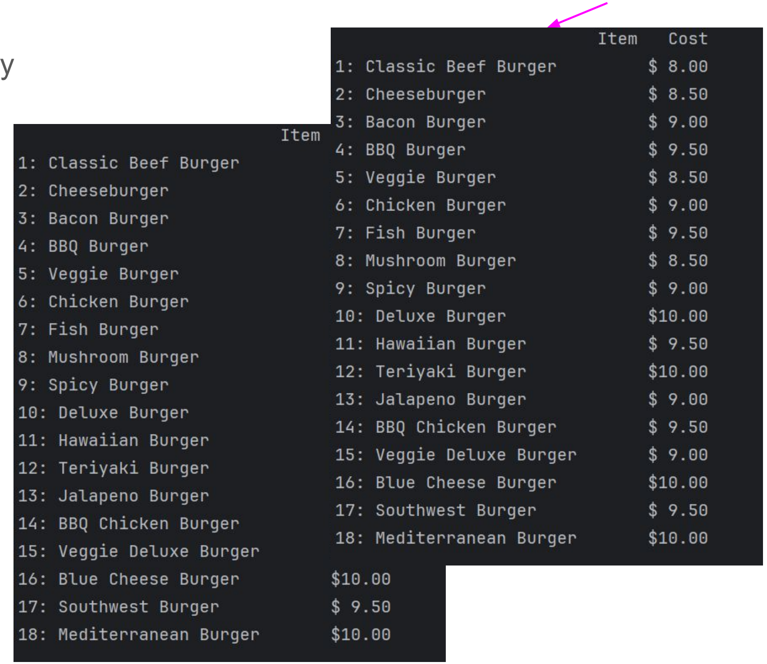


```
if topping_choice == "1":
    topping_name = "Cheese sauce"
    cost = 1.50
elif topping_choice == "2":
    topping_name = "Pesto sauce"
    cost = 1.50
elif topping_choice == "3":
    topping_name = "Marinara"
    cost = 0.0  # Marinara is free
elif topping_choice == "4":
    topping_name = "Barbeque"
    cost = 1.50
elif topping_choice == "5":
    topping_name = "Mozzarella Cheese"
    cost = 1.50
elif topping_choice == "6":
    topping_name = "Ground beef"
    cost = 3.50
elif topping_choice == "7":
    topping_name = "Smoked ham"
    cost = 3.50
elif topping_choice == "8":
    topping_name = "Pepperoni"
    cost = 3.50
elif topping_choice == "9":
    topping_name = "Onion"
    cost = 2.00
elif topping_choice == "10":
    topping_name = "Olives"
    cost = 2.00
elif topping_choice == "11":
    topping_name = "Spinach"
    cost = 2.00
elif topping_choice == "12":
    topping_name = "Spring Onion"
    cost = 1.00
elif topping_choice == "13":
    topping_name = "Premium Italian Sausage"
    cost = 5.50
```

Version 2



```
burger_menu = {
    1: {"name": "Classic Beef Burger", "cost": 8.00},
    2: {"name": "Cheeseburger", "cost": 8.50},
    3: {"name": "Bacon Burger", "cost": 9.00},
    4: {"name": "BBQ Burger", "cost": 9.50},
    5: {"name": "Veggie Burger", "cost": 8.50},
    6: {"name": "Chicken Burger", "cost": 9.00},
    7: {"name": "Fish Burger", "cost": 9.50},
    8: {"name": "Mushroom Burger", "cost": 8.50},
    9: {"name": "Spicy Burger", "cost": 9.00},
    10: {"name": "Deluxe Burger", "cost": 10.00},
    11: {"name": "Hawaiian Burger", "cost": 9.50},
    12: {"name": "Teriyaki Burger", "cost": 10.00},
    13: {"name": "Jalapeno Burger", "cost": 9.00},
    14: {"name": "BBQ Chicken Burger", "cost": 9.50},
    15: {"name": "Veggie Deluxe Burger", "cost": 9.00},
    16: {"name": "Blue Cheese Burger", "cost": 9.50},
    17: {"name": "Southwest Burger", "cost": 9.50},
    18: {"name": "Mediterranean Burger", "cost": 10.00}
}
```



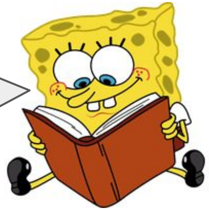
With dictionary

```
                                 Item
1: Classic Beef Burger
2: Cheeseburger
3: Bacon Burger
4: BBQ Burger
5: Veggie Burger
6: Chicken Burger
7: Fish Burger
8: Mushroom Burger
9: Spicy Burger
10: Deluxe Burger
11: Hawaiian Burger
12: Teriyaki Burger
13: Jalapeno Burger
14: BBQ Chicken Burger
15: Veggie Deluxe Burger
16: Blue Cheese Burger         $10.00
17: Southwest Burger           $ 9.50
18: Mediterranean Burger       $10.00
```

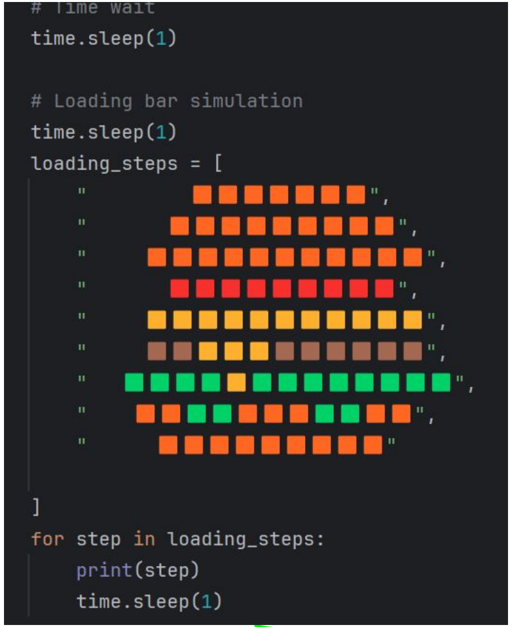
Without dictionary

```
                          Item    Cost
1: Classic Beef Burger           $ 8.00
2: Cheeseburger                  $ 8.50
3: Bacon Burger                  $ 9.00
4: BBQ Burger                    $ 9.50
5: Veggie Burger                 $ 8.50
6: Chicken Burger                $ 9.00
7: Fish Burger                   $ 9.50
8: Mushroom Burger               $ 8.50
9: Spicy Burger                  $ 9.00
10: Deluxe Burger                $10.00
11: Hawaiian Burger              $ 9.50
12: Teriyaki Burger              $10.00
13: Jalapeno Burger              $ 9.00
14: BBQ Chicken Burger           $ 9.50
15: Veggie Deluxe Burger         $ 9.00
16: Blue Cheese Burger           $10.00
17: Southwest Burger             $ 9.50
18: Mediterranean Burger         $10.00
```

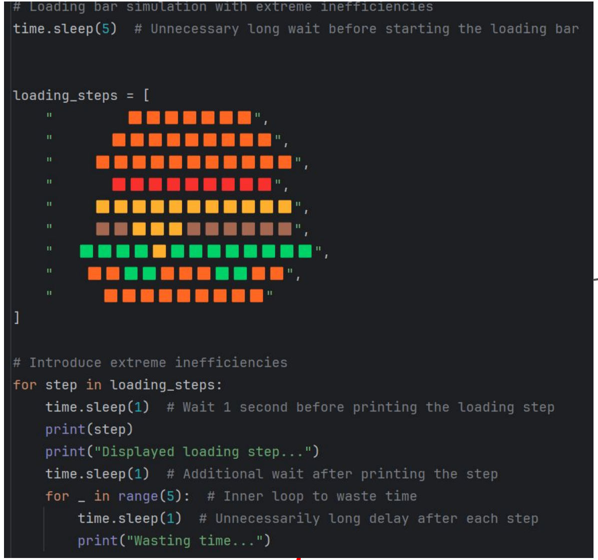
By converting my burger selection tool into a dictionary, I've significantly reduced the amount of code needed. This not only makes the code more efficient but also improves the program's readability and ease of navigation.



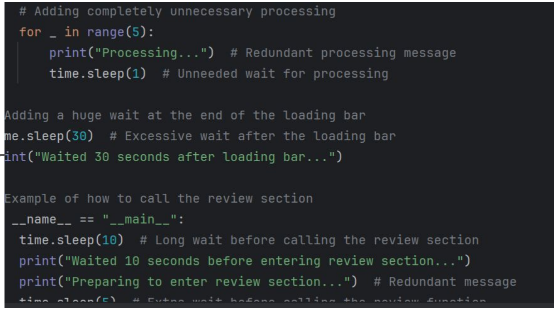
Making things more efficient (2)



```
# Time wait
time.sleep(1)

# Loading bar simulation
time.sleep(1)
loading_steps = [
    "       🟧🟧🟧🟧🟧🟧🟧",
    "     🟧🟧🟧🟧🟧🟧🟧🟧🟧",
    "   🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧",
    "     🟥🟥🟥🟥🟥🟥🟥🟥🟥",
    "   🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨",
    "   🟫🟫🟨🟨🟨🟫🟫🟫🟫🟫🟫",
    " 🟩🟩🟩🟩🟨🟩🟩🟩🟩🟩🟩🟩🟩",
    "  🟧🟧🟩🟩🟧🟧🟧🟩🟩🟧🟧",
    "    🟧🟧🟧🟧🟧🟧🟧🟧🟧"

]
for step in loading_steps:
    print(step)
    time.sleep(1)
```

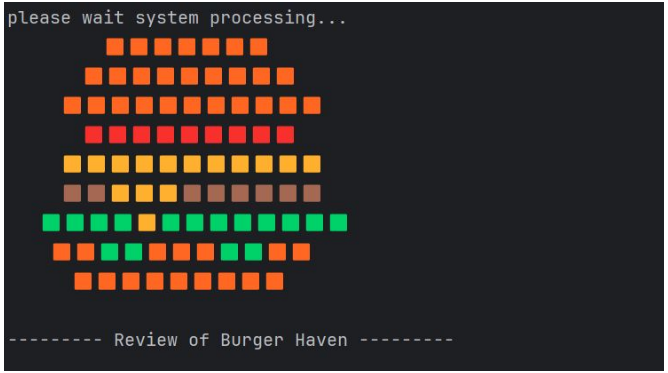


```
# Loading bar simulation with extreme inefficiencies
time.sleep(5)  # Unnecessary long wait before starting the loading bar


loading_steps = [
    "       🟧🟧🟧🟧🟧🟧🟧",
    "     🟧🟧🟧🟧🟧🟧🟧🟧🟧",
    "   🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧",
    "     🟥🟥🟥🟥🟥🟥🟥🟥🟥",
    "   🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨",
    "   🟫🟫🟨🟨🟨🟫🟫🟫🟫🟫🟫",
    " 🟩🟩🟩🟩🟨🟩🟩🟩🟩🟩🟩🟩🟩",
    "  🟧🟧🟩🟩🟧🟧🟧🟩🟩🟧🟧",
    "    🟧🟧🟧🟧🟧🟧🟧🟧🟧"
]

# Introduce extreme inefficiencies
for step in loading_steps:
    time.sleep(1)  # Wait 1 second before printing the loading step
    print(step)
    print("Displayed loading step...")
    time.sleep(1)  # Additional wait after printing the step
    for _ in range(5):  # Inner loop to waste time
        time.sleep(1)  # Unnecessarily long delay after each step
        print("Wasting time...")
```



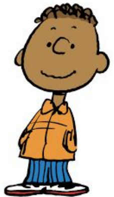
```
  # Adding completely unnecessary processing
  for _ in range(5):
      print("Processing...")  # Redundant processing message
      time.sleep(1)  # Unneeded wait for processing

Adding a huge wait at the end of the loading bar
me.sleep(30)  # Excessive wait after the loading bar
int("Waited 30 seconds after loading bar...")

Example of how to call the review section
 __name__ == "__main__":
  time.sleep(10)  # Long wait before calling the review section
  print("Waited 10 seconds before entering review section...")
  print("Preparing to enter review section...")  # Redundant message
```

Dose the same thing
just more efficient!

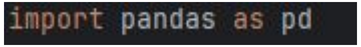


```
please wait system processing...
       🟧🟧🟧🟧🟧🟧🟧
     🟧🟧🟧🟧🟧🟧🟧🟧🟧
   🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧🟧
     🟥🟥🟥🟥🟥🟥🟥🟥🟥
   🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨
   🟫🟫🟨🟨🟨🟫🟫🟫🟫🟫🟫
 🟩🟩🟩🟩🟨🟩🟩🟩🟩🟩🟩🟩🟩
  🟧🟧🟩🟩🟧🟧🟧🟩🟩🟧🟧
    🟧🟧🟧🟧🟧🟧🟧🟧🟧


--------- Review of Burger Haven ---------
```

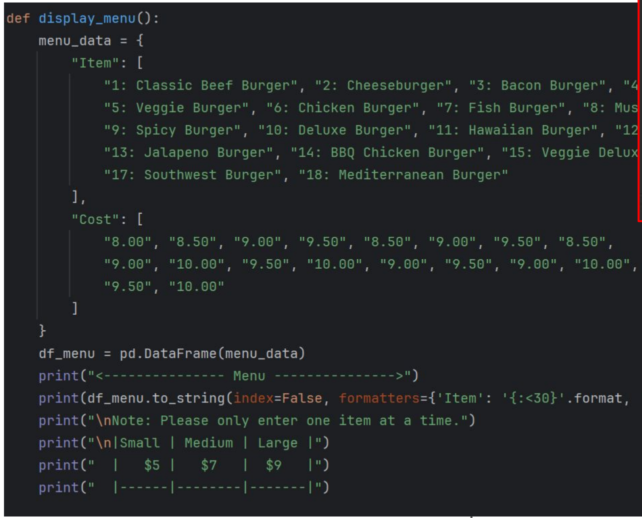
print(“Improved the burger logo end loading screen efficiency by removing unnecessary code”)



Pandas!

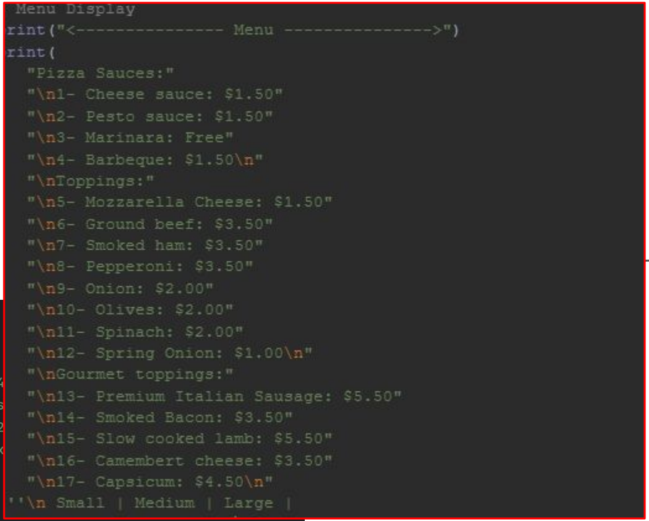


```
import pandas as pd
```



```
def display_menu():
    menu_data = {
        "Item": [
            "1: Classic Beef Burger", "2: Cheeseburger", "3: Bacon Burger", "4
            "5: Veggie Burger", "6: Chicken Burger", "7: Fish Burger", "8: Mus
            "9: Spicy Burger", "10: Deluxe Burger", "11: Hawaiian Burger", "12
            "13: Jalapeno Burger", "14: BBQ Chicken Burger", "15: Veggie Delux
            "17: Southwest Burger", "18: Mediterranean Burger"
        ],
        "Cost": [
            "8.00", "8.50", "9.00", "9.50", "8.50", "9.00", "9.50", "8.50",
            "9.00", "10.00", "9.50", "10.00", "9.00", "9.50", "9.00", "10.00",
            "9.50", "10.00"
        ]
    }
    df_menu = pd.DataFrame(menu_data)
    print("<-------------- Menu -------------->")
    print(df_menu.to_string(index=False, formatters={'Item': '{:<30}'.format, 'Cost': '${:>5}'.format}))
    print("\nNote: Please only enter one item at a time.")
    print("\n|Small | Medium | Large |")
    print("  |   $5 |   $7   |  $9   |")
    print("  |------|--------|-------|")
```

With pandas



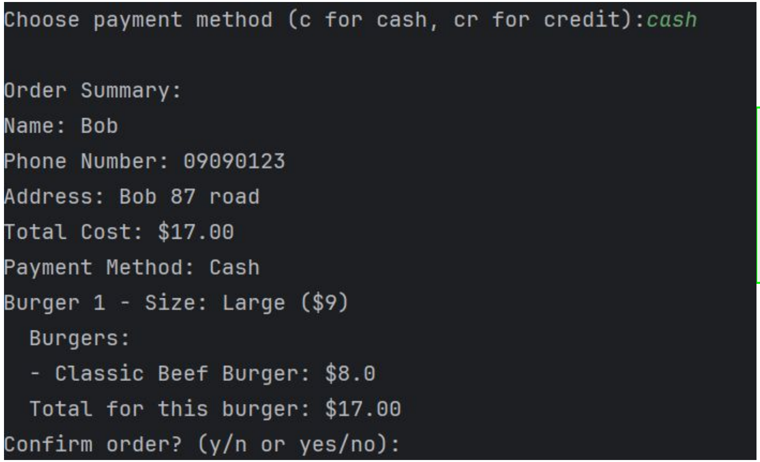
```
 Menu Display
rint("<--------------- Menu --------------->")
rint(
  "Pizza Sauces:"
  "\n1- Cheese sauce: $1.50"
  "\n2- Pesto sauce: $1.50"
  "\n3- Marinara: Free"
  "\n4- Barbeque: $1.50\n"
  "\nToppings:"
  "\n5- Mozzarella Cheese: $1.50"
  "\n6- Ground beef: $3.50"
  "\n7- Smoked ham: $3.50"
  "\n8- Pepperoni: $3.50"
  "\n9- Onion: $2.00"
  "\n10- Olives: $2.00"
  "\n11- Spinach: $2.00"
  "\n12- Spring Onion: $1.00\n"
  "\nGourmet toppings:"
  "\n13- Premium Italian Sausage: $5.50"
  "\n14- Smoked Bacon: $3.50"
  "\n15- Slow cooked lamb: $5.50"
  "\n16- Camembert cheese: $3.50"
  "\n17- Capsicum: $4.50\n"
''\n Small | Medium | Large |
```

Before (pizza version)

Using `pandas` will make it easier to add new features to my pizza program, like sales. It would be able to handle new data smoothly and helps analyze and manage it efficiently. Compared to the old version

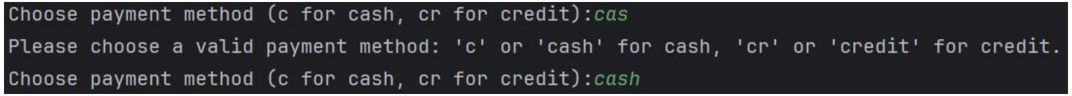
print("Using pandas in Python makes working with data faster and easier. It provides a special table-like structure called a DataFrame that handles large amounts of data more efficiently. With pandas, my program can quickly sort, and analyze the data, as well as create charts like my menu. It simplifies many tasks that would be more complicated and time-consuming with just basic Python code such as print.")

Testing and additions (1)



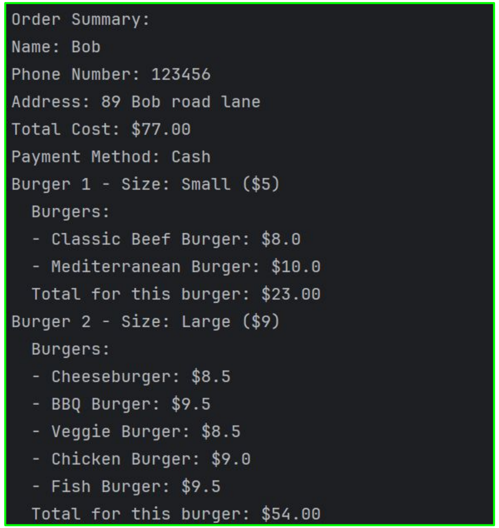
```
Choose payment method (c for cash, cr for credit):cash

Order Summary:
Name: Bob
Phone Number: 09090123
Address: Bob 87 road
Total Cost: $17.00
Payment Method: Cash
Burger 1 - Size: Large ($9)
  Burgers:
  - Classic Beef Burger: $8.0
  Total for this burger: $17.00
Confirm order? (y/n or yes/no):
```

Resolved issues in the cash and credit payment system.



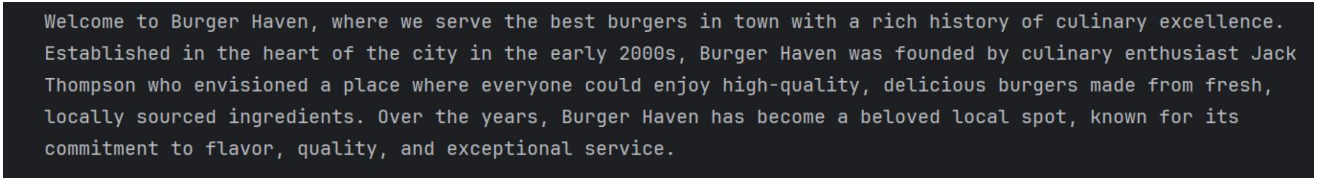
```
Choose payment method (c for cash, cr for credit):cas
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):cash
```

Added a price total receipt feature at the end of the order process with name phone number and address



```
Order Summary:
Name: Bob
Phone Number: 123456
Address: 89 Bob road lane
Total Cost: $77.00
Payment Method: Cash
Burger 1 - Size: Small ($5)
  Burgers:
  - Classic Beef Burger: $8.0
  - Mediterranean Burger: $10.0
  Total for this burger: $23.00
Burger 2 - Size: Large ($9)
  Burgers:
  - Cheeseburger: $8.5
  - BBQ Burger: $9.5
  - Veggie Burger: $8.5
  - Chicken Burger: $9.0
  - Fish Burger: $9.5
  Total for this burger: $54.00
```

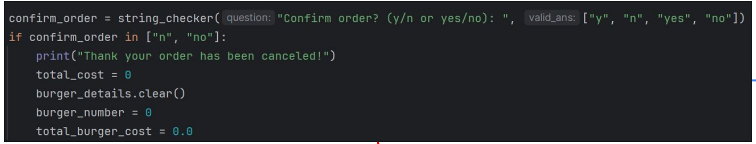
Added a burger history option to the start of the program



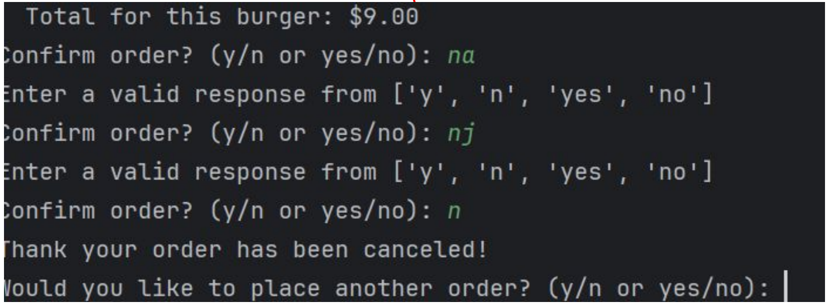
```
Welcome to Burger Haven, where we serve the best burgers in town with a rich history of culinary excellence.
Established in the heart of the city in the early 2000s, Burger Haven was founded by culinary enthusiast Jack
Thompson who envisioned a place where everyone could enjoy high-quality, delicious burgers made from fresh,
locally sourced ingredients. Over the years, Burger Haven has become a beloved local spot, known for its
commitment to flavor, quality, and exceptional service.
```

Testing and additions

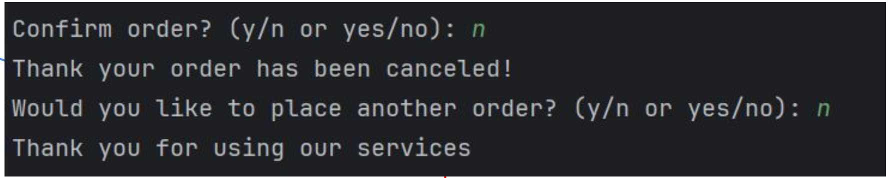
Error recovery



```python
confirm_order = string_checker( question: "Confirm order? (y/n or yes/no): ",  valid_ans: ["y", "n", "yes", "no"])
if confirm_order in ["n", "no"]:
    print("Thank your order has been canceled!")
    total_cost = 0
    burger_details.clear()
    burger_number = 0
    total_burger_cost = 0.0
```

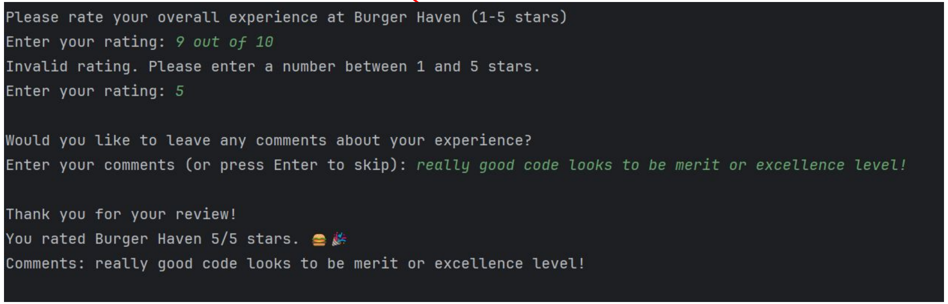


```
 Total for this burger: $9.00
onfirm order? (y/n or yes/no): na
nter a valid response from ['y', 'n', 'yes', 'no']
onfirm order? (y/n or yes/no): nj
nter a valid response from ['y', 'n', 'yes', 'no']
onfirm order? (y/n or yes/no): n
hank your order has been canceled!
ould you like to place another order? (y/n or yes/no): |
```



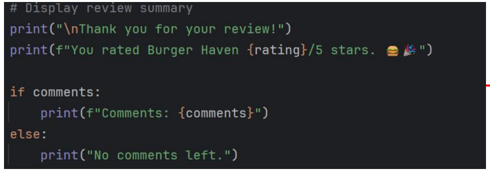
```
Confirm order? (y/n or yes/no): n
Thank your order has been canceled!
Would you like to place another order? (y/n or yes/no): n
Thank you for using our services
```

Leaving a review

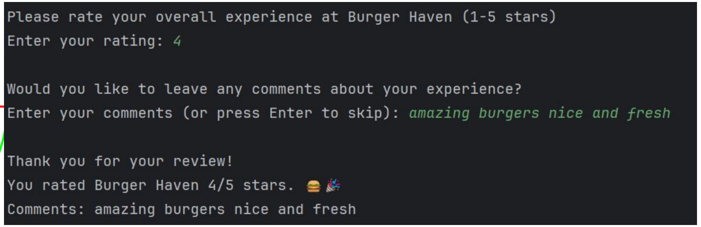


```
Please rate your overall experience at Burger Haven (1-5 stars)
Enter your rating: 9 out of 10
Invalid rating. Please enter a number between 1 and 5 stars.
Enter your rating: 5

Would you like to leave any comments about your experience?
Enter your comments (or press Enter to skip): really good code looks to be merit or excellence level!

Thank you for your review!
You rated Burger Haven 5/5 stars. 🍔🎉
Comments: really good code looks to be merit or excellence level!
```

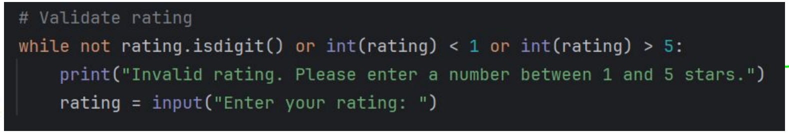
Revamped the final order review menu and implemented an error recovery system.



```python
# Display review summary
print("\nThank you for your review!")
print(f"You rated Burger Haven {rating}/5 stars. 🍔🎉")

if comments:
    print(f"Comments: {comments}")
else:
    print("No comments left.")
```



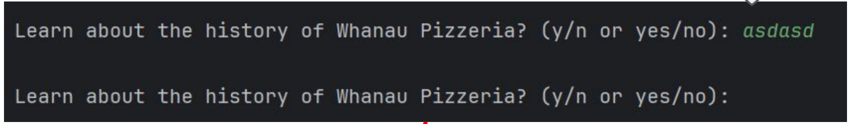
```
Please rate your overall experience at Burger Haven (1-5 stars)
Enter your rating: 4

Would you like to leave any comments about your experience?
Enter your comments (or press Enter to skip): amazing burgers nice and fresh

Thank you for your review!
You rated Burger Haven 4/5 stars. 🍔🎉
Comments: amazing burgers nice and fresh
```



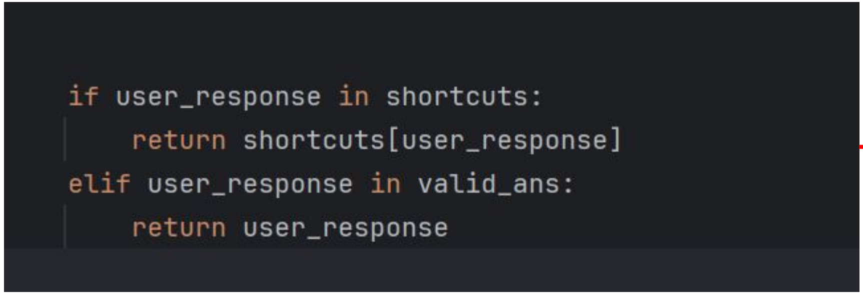
```python
# Validate rating
while not rating.isdigit() or int(rating) < 1 or int(rating) > 5:
    print("Invalid rating. Please enter a number between 1 and 5 stars.")
    rating = input("Enter your rating: ")
```

More testing and additions

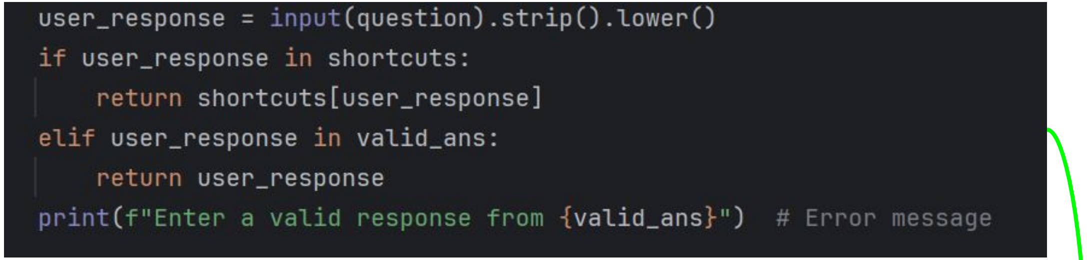
Before



```
Learn about the history of Whanau Pizzeria? (y/n or yes/no): asdasd

Learn about the history of Whanau Pizzeria? (y/n or yes/no):
```



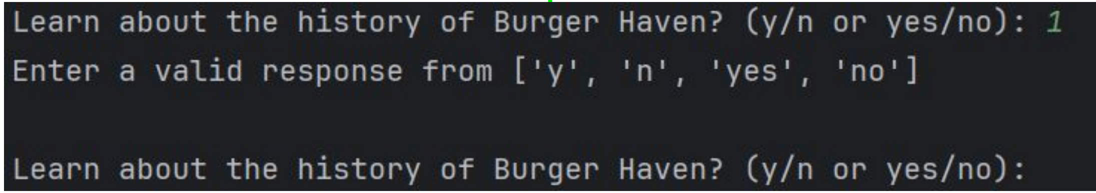
```
if user_response in shortcuts:
    return shortcuts[user_response]
elif user_response in valid_ans:
    return user_response
```



```
user_response = input(question).strip().lower()
if user_response in shortcuts:
    return shortcuts[user_response]
elif user_response in valid_ans:
    return user_response
print(f"Enter a valid response from {valid_ans}")  # Error message
```

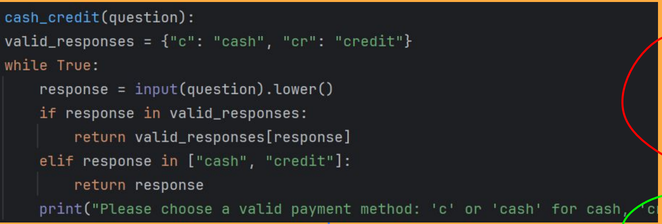
Adding this line of code enhances user guidance by clearly listing all possible input options, making it easier for users to understand what they can enter.

after

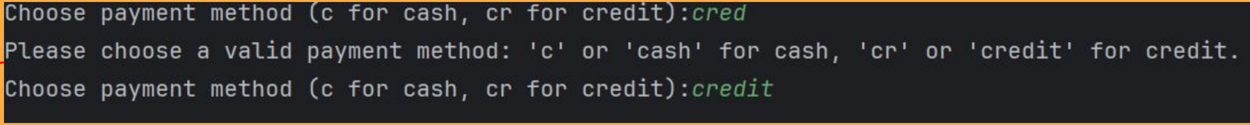


```
Learn about the history of Burger Haven? (y/n or yes/no): 1
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no):
```

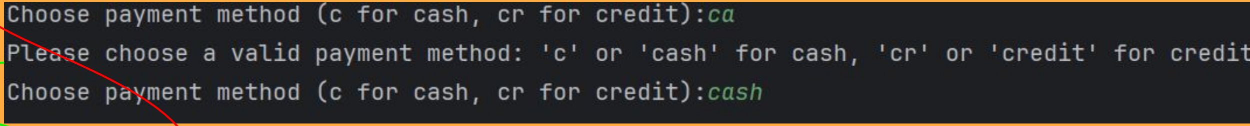
Additions 🔧



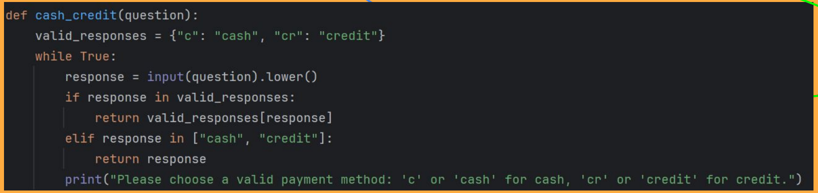
```python
cash_credit(question):
valid_responses = {"c": "cash", "cr": "credit"}
while True:
    response = input(question).lower()
    if response in valid_responses:
        return valid_responses[response]
    elif response in ["cash", "credit"]:
        return response
    print("Please choose a valid payment method: 'c' or 'cash' for cash, 'c
```



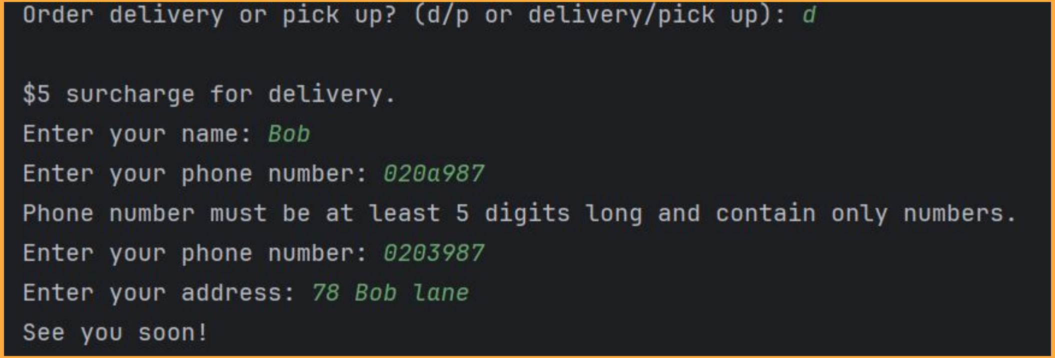
```
Choose payment method (c for cash, cr for credit):cred
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):credit
```



```
Choose payment method (c for cash, cr for credit):ca
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit
Choose payment method (c for cash, cr for credit):cash
```

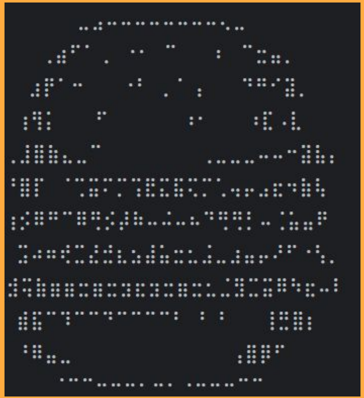


```python
def cash_credit(question):
    valid_responses = {"c": "cash", "cr": "credit"}
    while True:
        response = input(question).lower()
        if response in valid_responses:
            return valid_responses[response]
        elif response in ["cash", "credit"]:
            return response
        print("Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.")
```



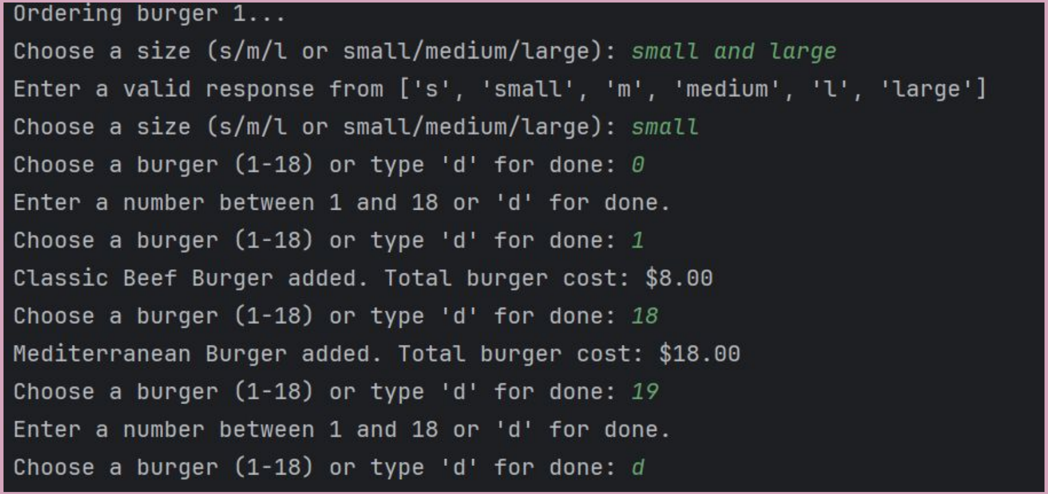
```
Order delivery or pick up? (d/p or delivery/pick up): d


$5 surcharge for delivery.
Enter your name: Bob
Enter your phone number: 020a987
Phone number must be at least 5 digits long and contain only numbers.
Enter your phone number: 0203987
Enter your address: 78 Bob lane
See you soon!
```

Enhanced the code by incorporating additional detail options and included a burger image at the beginning

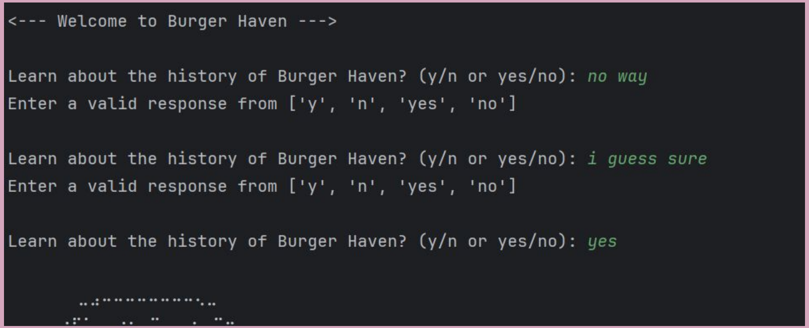


Testing

Testing that all toppings work



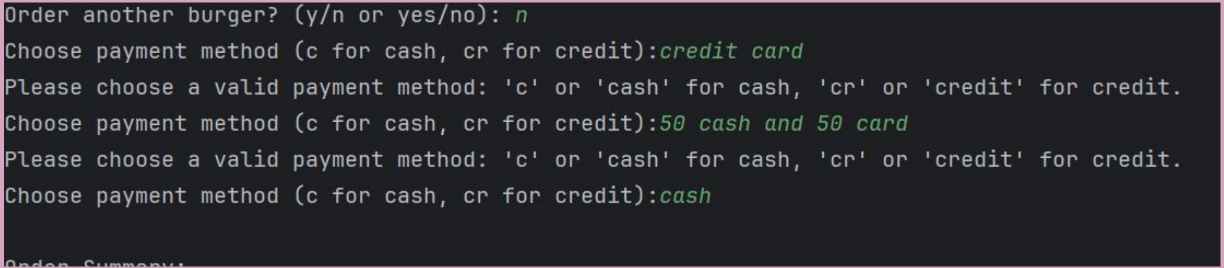
```
Ordering burger 1...
Choose a size (s/m/l or small/medium/large): small and large
Enter a valid response from ['s', 'small', 'm', 'medium', 'l', 'large']
Choose a size (s/m/l or small/medium/large): small
Choose a burger (1-18) or type 'd' for done: 0
Enter a number between 1 and 18 or 'd' for done.
Choose a burger (1-18) or type 'd' for done: 1
Classic Beef Burger added. Total burger cost: $8.00
Choose a burger (1-18) or type 'd' for done: 18
Mediterranean Burger added. Total burger cost: $18.00
Choose a burger (1-18) or type 'd' for done: 19
Enter a number between 1 and 18 or 'd' for done.
Choose a burger (1-18) or type 'd' for done: d
```



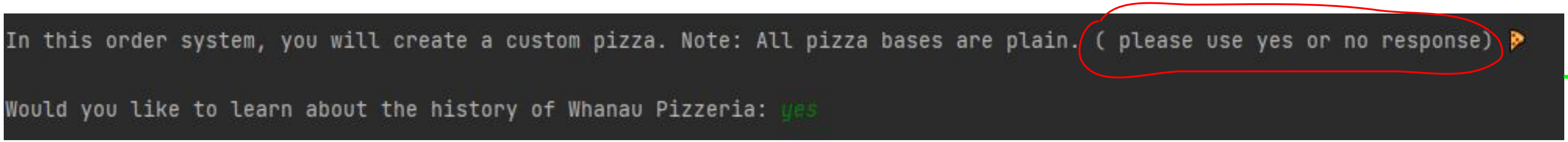
```
<--- Welcome to Burger Haven --->

Learn about the history of Burger Haven? (y/n or yes/no): no way
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): i guess sure
Enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Burger Haven? (y/n or yes/no): yes
```

#Youtube link for full code run test#

https://youtu.be/1ZH-gCTfhvM

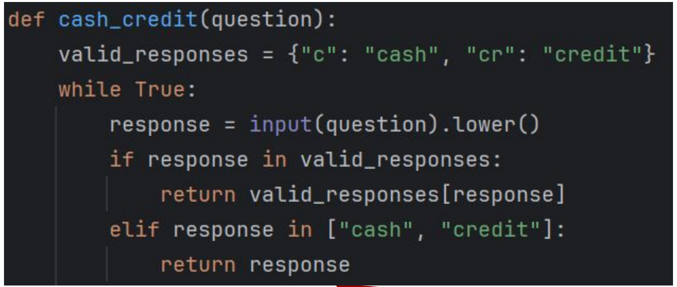


```
Order another burger? (y/n or yes/no): n
Choose payment method (c for cash, cr for credit):credit card
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):50 cash and 50 card
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.
Choose payment method (c for cash, cr for credit):cash

Order Summary:
```



```
In this order system, you will create a custom pizza. Note: All pizza bases are plain. ( please use yes or no response) 🍕

Would you like to learn about the history of Whanau Pizzeria: yes
```

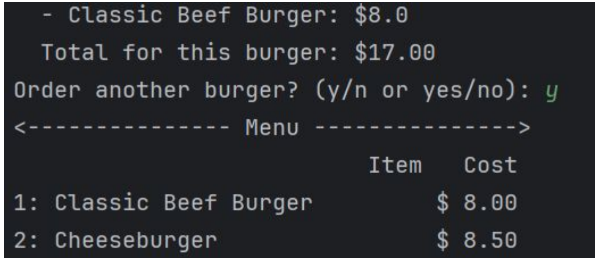
Added to ensure users know that 'yes' or 'no' are valid responses.



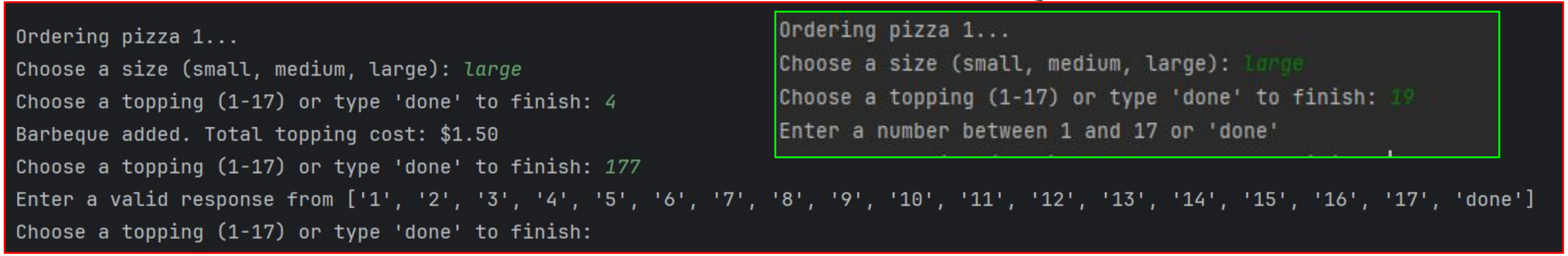
```python
def cash_credit(question):
    valid_responses = {"c": "cash", "cr": "credit"}
    while True:
        response = input(question).lower()
        if response in valid_responses:
            return valid_responses[response]
        elif response in ["cash", "credit"]:
            return response
```

more options have been added to further reduce the likelihood of mistakes

Changed to shorter answer more efficient



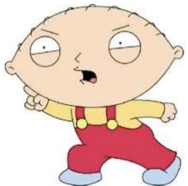
```
  - Classic Beef Burger: $8.0
  Total for this burger: $17.00
Order another burger? (y/n or yes/no): y
<--------------- Menu --------------->
                          Item    Cost
1: Classic Beef Burger          $ 8.00
2: Cheeseburger                 $ 8.50
```

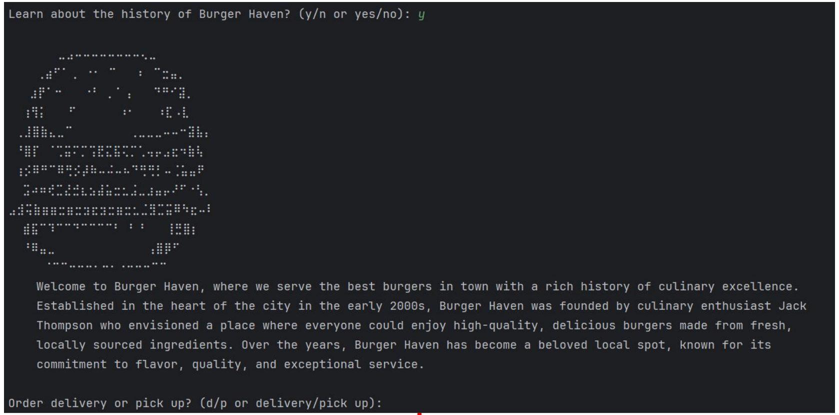


```
Ordering pizza 1...
Choose a size (small, medium, large): large
Choose a topping (1-17) or type 'done' to finish: 4
Barbeque added. Total topping cost: $1.50
Choose a topping (1-17) or type 'done' to finish: 177
Enter a valid response from ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', 'done']
Choose a topping (1-17) or type 'done' to finish:
```

```
Ordering pizza 1...
Choose a size (small, medium, large): large
Choose a topping (1-17) or type 'done' to finish: 19
Enter a number between 1 and 17 or 'done'
```

User guidance

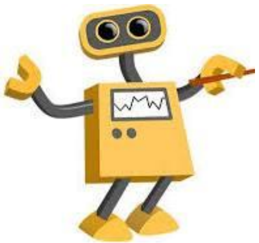




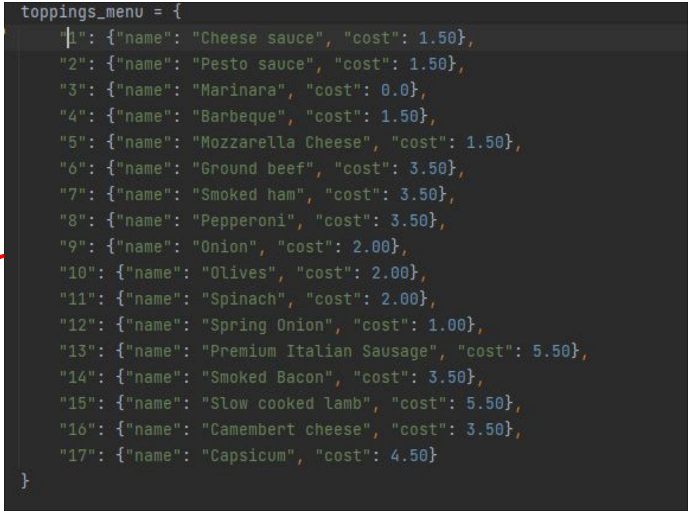
```
Learn about the history of Burger Haven? (y/n or yes/no): y

    Welcome to Burger Haven, where we serve the best burgers in town with a rich history of culinary excellence.
    Established in the heart of the city in the early 2000s, Burger Haven was founded by culinary enthusiast Jack
    Thompson who envisioned a place where everyone could enjoy high-quality, delicious burgers made from fresh,
    locally sourced ingredients. Over the years, Burger Haven has become a beloved local spot, known for its
    commitment to flavor, quality, and exceptional service.

Order delivery or pick up? (d/p or delivery/pick up):
```

Single-letter responses are allowed as it is more efficient

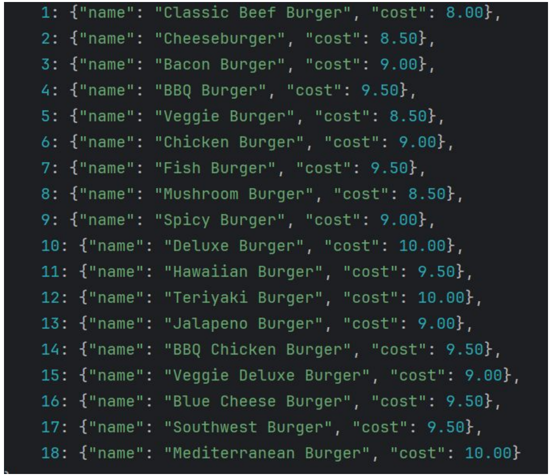
Other comments



This allows the code to interpret the input as numerical values rather than symbols, enabling it to correctly recognize numbers ranging from 1 to 17.

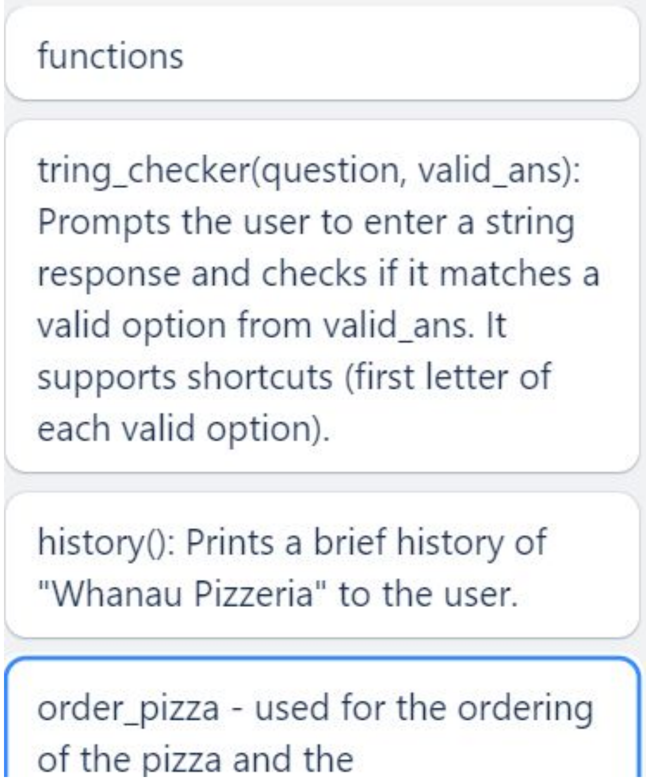


```
toppings_menu = {
    "1": {"name": "Cheese sauce", "cost": 1.50},
    "2": {"name": "Pesto sauce", "cost": 1.50},
    "3": {"name": "Marinara", "cost": 0.0},
    "4": {"name": "Barbeque", "cost": 1.50},
    "5": {"name": "Mozzarella Cheese", "cost": 1.50},
    "6": {"name": "Ground beef", "cost": 3.50},
    "7": {"name": "Smoked ham", "cost": 3.50},
    "8": {"name": "Pepperoni", "cost": 3.50},
    "9": {"name": "Onion", "cost": 2.00},
    "10": {"name": "Olives", "cost": 2.00},
    "11": {"name": "Spinach", "cost": 2.00},
    "12": {"name": "Spring Onion", "cost": 1.00},
    "13": {"name": "Premium Italian Sausage", "cost": 5.50},
    "14": {"name": "Smoked Bacon", "cost": 3.50},
    "15": {"name": "Slow cooked lamb", "cost": 5.50},
    "16": {"name": "Camembert cheese", "cost": 3.50},
    "17": {"name": "Capsicum", "cost": 4.50}
}
```



```
    1: {"name": "Classic Beef Burger", "cost": 8.00},
    2: {"name": "Cheeseburger", "cost": 8.50},
    3: {"name": "Bacon Burger", "cost": 9.00},
    4: {"name": "BBQ Burger", "cost": 9.50},
    5: {"name": "Veggie Burger", "cost": 8.50},
    6: {"name": "Chicken Burger", "cost": 9.00},
    7: {"name": "Fish Burger", "cost": 9.50},
    8: {"name": "Mushroom Burger", "cost": 8.50},
    9: {"name": "Spicy Burger", "cost": 9.00},
    10: {"name": "Deluxe Burger", "cost": 10.00},
    11: {"name": "Hawaiian Burger", "cost": 9.50},
    12: {"name": "Teriyaki Burger", "cost": 10.00},
    13: {"name": "Jalapeno Burger", "cost": 9.00},
    14: {"name": "BBQ Chicken Burger", "cost": 9.50},
    15: {"name": "Veggie Deluxe Burger", "cost": 9.00},
    16: {"name": "Blue Cheese Burger", "cost": 9.50},
    17: {"name": "Southwest Burger", "cost": 9.50},
    18: {"name": "Mediterranean Burger", "cost": 10.00}
```

Trello functions

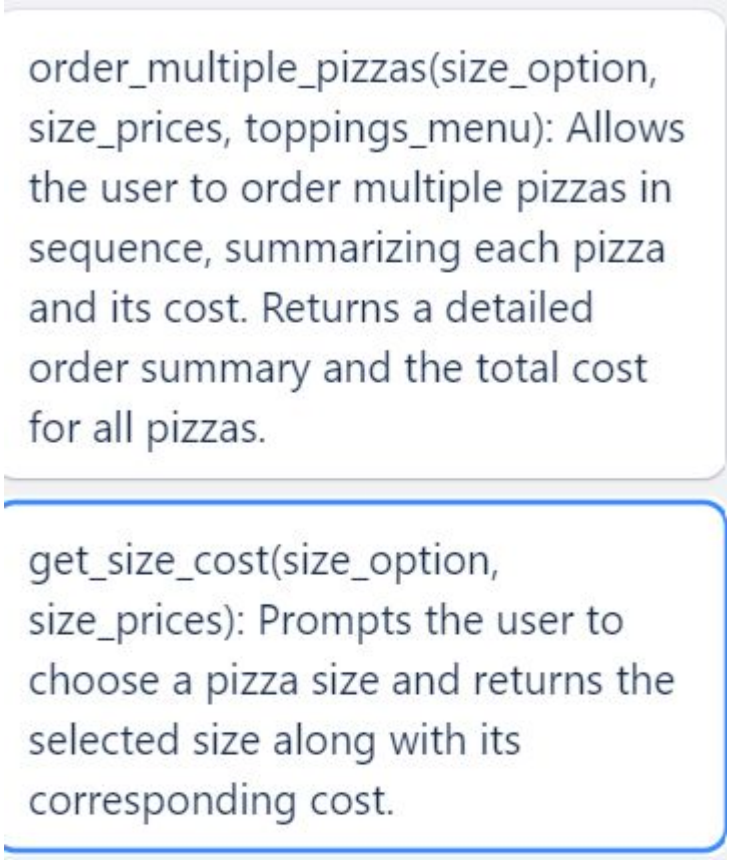


functions

tring_checker(question, valid_ans): Prompts the user to enter a string response and checks if it matches a valid option from valid_ans. It supports shortcuts (first letter of each valid option).

history(): Prints a brief history of "Whanau Pizzeria" to the user.

order_pizza - used for the ordering of the pizza and the

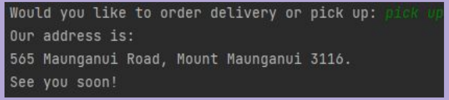


order_multiple_pizzas(size_option, size_prices, toppings_menu): Allows the user to order multiple pizzas in sequence, summarizing each pizza and its cost. Returns a detailed order summary and the total cost for all pizzas.

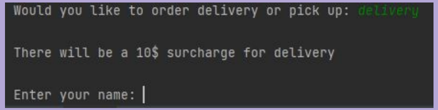
get_size_cost(size_option, size_prices): Prompts the user to choose a pizza size and returns the selected size along with its corresponding cost.

Component 1 - Test Plan (?and screenshot)

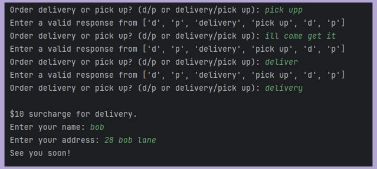
Test Case	Expected Values
Pick up or p - string checker	Our address is: 565 Maunganui Road, Mount Maunganui 3116. See you soon!
Delivery or d	There will be a 10$ surcharge for delivery Enter your name:
reajsd	Enter a valid response from ['delivery', 'pick up'] Would you like to order delivery or pick up:



```
Would you like to order delivery or pick up: pick up
Our address is:
565 Maunganui Road, Mount Maunganui 3116.
See you soon!
```



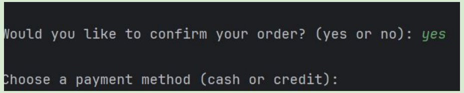
```
Would you like to order delivery or pick up: delivery

There will be a 10$ surcharge for delivery

Enter your name: 
```



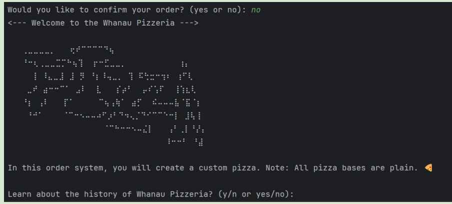
```
Order delivery or pick up? (d/p or delivery/pick up): pick upp
Enter a valid response from ['d', 'p', 'delivery', 'pick up', 'd', 'p']
Order delivery or pick up? (d/p or delivery/pick up): ill come get it
Enter a valid response from ['d', 'p', 'delivery', 'pick up', 'd', 'p']
Order delivery or pick up? (d/p or delivery/pick up): deliver
Enter a valid response from ['d', 'p', 'delivery', 'pick up', 'd', 'p']
Order delivery or pick up? (d/p or delivery/pick up): delivery

$10 surcharge for delivery.
Enter your name: bob
Enter your address: 28 bob lane
See you soon!
```

Component 2 - Test Plan (?and screenshot)

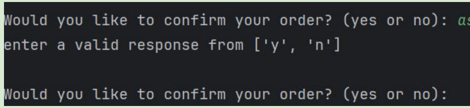
Test Case	Expected Values
Confirm order: yes or y	Would you like to confirm your order? (yes or no): yes Choose a payment method (cash or credit):
Confirm order: no or n	Would you like to confirm your order? (yes or no): no Code restarts
Confirm order: asfda	enter a valid response from ['y', 'n']



```
Would you like to confirm your order? (yes or no): yes

Choose a payment method (cash or credit):
```



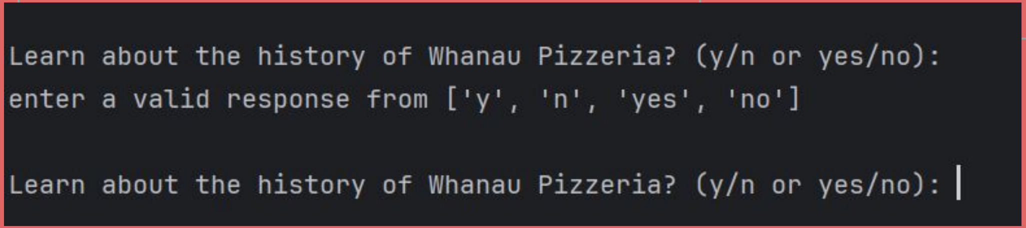
```
Would you like to confirm your order? (yes or no): no
<--- Welcome to the Whanau Pizzeria --->

In this order system, you will create a custom pizza. Note: All pizza bases are plain. 🍕

Learn about the history of Whanau Pizzeria? (y/n or yes/no):
```



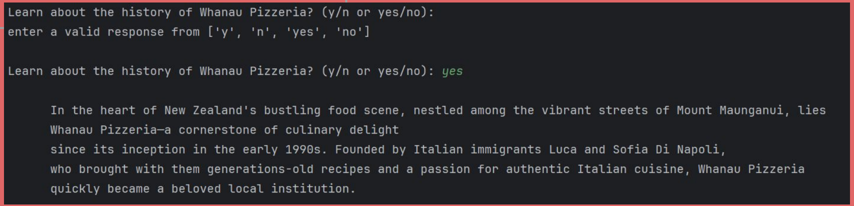
```
Would you like to confirm your order? (yes or no): a
enter a valid response from ['y', 'n']

Would you like to confirm your order? (yes or no):
```

Component 3 - Test Plan (?and screenshot)

Test Case	Expected Values
`not_blank(question):`	Ensures that the user input is not blank. Prompts repeatedly until a non-blank response is provided.
Blank	Enter a valid response from (valid_answer)
y,n,yes,no	Prints next part of code



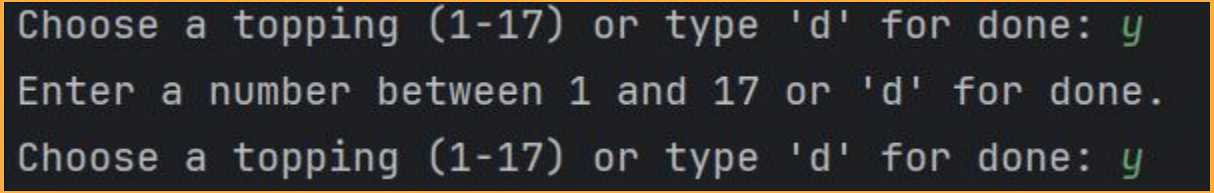
```
Learn about the history of Whanau Pizzeria? (y/n or yes/no):
enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): 
```



```
Learn about the history of Whanau Pizzeria? (y/n or yes/no):
enter a valid response from ['y', 'n', 'yes', 'no']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): yes

    In the heart of New Zealand's bustling food scene, nestled among the vibrant streets of Mount Maunganui, lies
    Whanau Pizzeria—a cornerstone of culinary delight
    since its inception in the early 1990s. Founded by Italian immigrants Luca and Sofia Di Napoli,
    who brought with them generations-old recipes and a passion for authentic Italian cuisine, Whanau Pizzeria
    quickly became a beloved local institution.
```

Component 4 - Test Plan (?and screenshot)

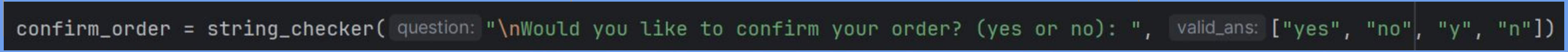
Test Case	Expected Values
Num_checker	Checks if the input is a number instead of a symbol
Number in non number question	Enter a valid response from (valid_answer)
P,d,pick up, delivery	Prints next part of code



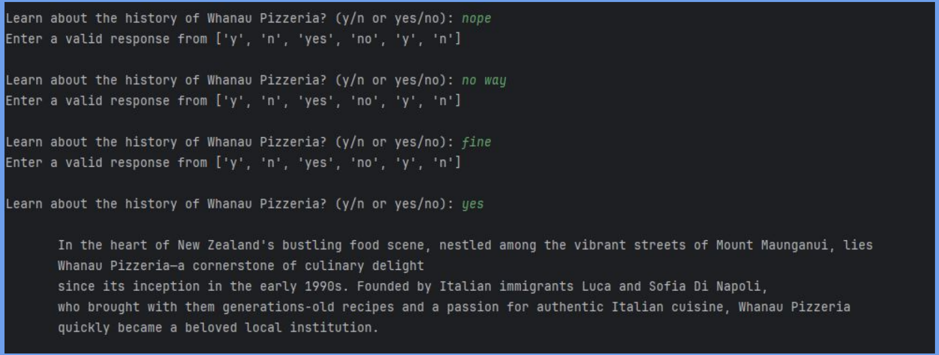
```
Choose a topping (1-17) or type 'd' for done: y
Enter a number between 1 and 17 or 'd' for done.
Choose a topping (1-17) or type 'd' for done: y
```

Component 5 - Test Plan (?and screenshot)

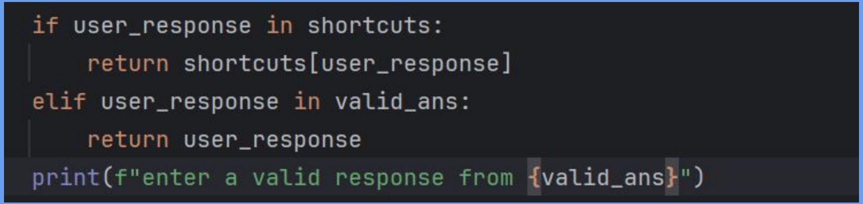
Test Case	Expected Values
String checker = Yes no checker	Checks for yes or no or y or n
Yes or no	Moves to next part of code
Something other then yes or no	Re asks question and gets new answer



```
confirm_order = string_checker( question: "\nWould you like to confirm your order? (yes or no): ", valid_ans: ["yes", "no", "y", "n"])
```



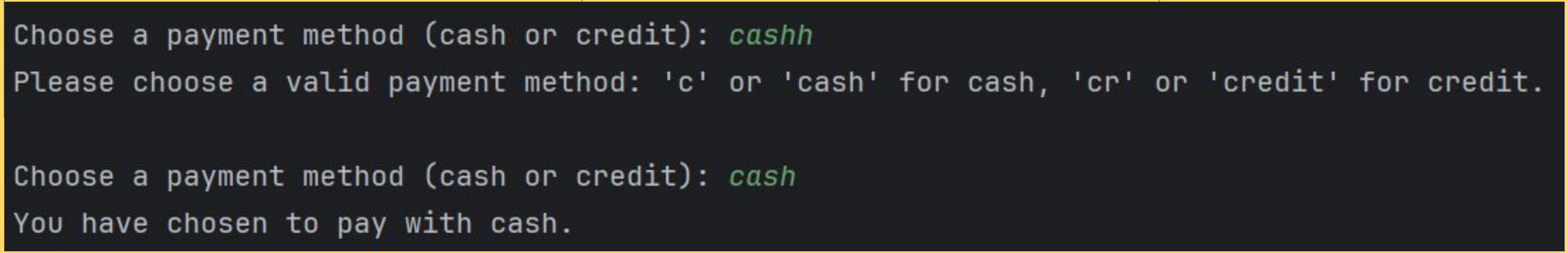
```
Learn about the history of Whanau Pizzeria? (y/n or yes/no): nope
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): no way
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): fine
Enter a valid response from ['y', 'n', 'yes', 'no', 'y', 'n']

Learn about the history of Whanau Pizzeria? (y/n or yes/no): yes

    In the heart of New Zealand's bustling food scene, nestled among the vibrant streets of Mount Maunganui, lies
    Whanau Pizzeria—a cornerstone of culinary delight
    since its inception in the early 1990s. Founded by Italian immigrants Luca and Sofia Di Napoli,
    who brought with them generations-old recipes and a passion for authentic Italian cuisine, Whanau Pizzeria
    quickly became a beloved local institution.
```



```
if user_response in shortcuts:
    return shortcuts[user_response]
elif user_response in valid_ans:
    return user_response
print(f"enter a valid response from {valid_ans}")
```

Component 6 - Test Plan (?and screenshot)

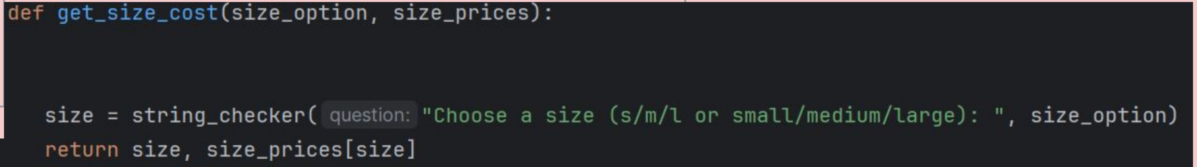
Test cash credit	Expected Values
`cash_credit(question):`	Prompts the user to choose between "cash" or "credit" as a payment method. Allows for abbreviations (e.g., 'c' for cash).
Cash or credit	Will register if user has chosen to pay with cash or credit
Something other then cash or credit	Will ask them to put cash or credit



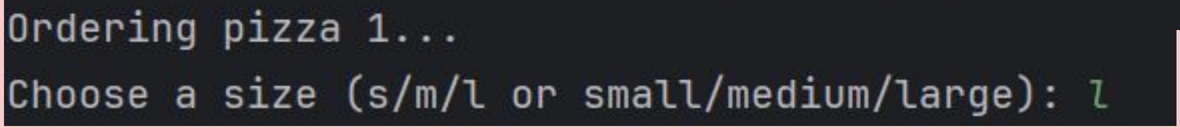
```
Choose a payment method (cash or credit): cashh
Please choose a valid payment method: 'c' or 'cash' for cash, 'cr' or 'credit' for credit.

Choose a payment method (cash or credit): cash
You have chosen to pay with cash.
```

Component 7 - Test Plan (?and screenshot)

Test get size	**Expected Values**
`get_size_cost(size_option, size_prices)`:	Prompts the user to choose a pizza size and returns the selected size along with its corresponding cost.
Inputs l,m,s	Adds the size of the pizza to the total
Input other the l,m,s	Asks you to put a valid answer



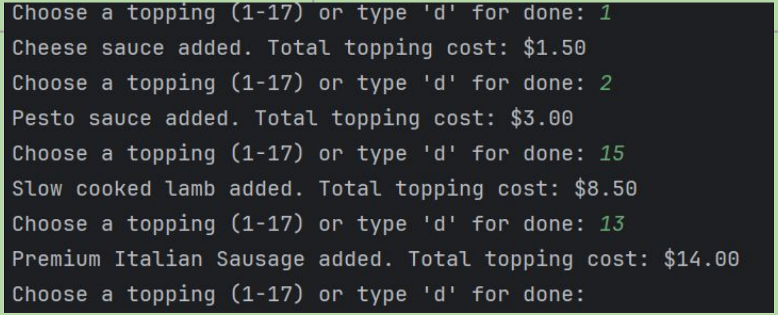
```
def get_size_cost(size_option, size_prices):


    size = string_checker( question: "Choose a size (s/m/l or small/medium/large): ", size_option)
    return size, size_prices[size]
```



```
Ordering pizza 1...
Choose a size (s/m/l or small/medium/large): l
```

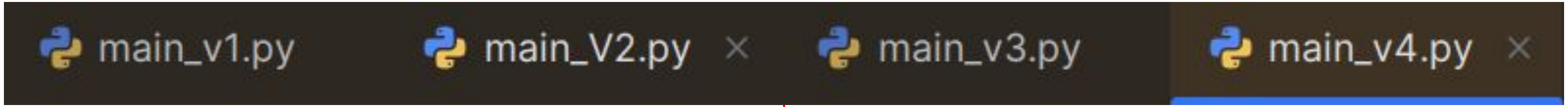
Component 8 - Test Plan (?and screenshot)

Test	Expected Values
int_checker(question, low, high)	Prompts the user to enter an integer within a specified range (`low` to `high`). Returns the integer if valid or `'d'` if the user indicates they are done.
Invalid input - e.g - letter, or numbers outside of 1-17	Will ask you to put a valid answer
Between 1-17 (valid input 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)	Will move onto the next part of the code



```
Choose a topping (1-17) or type 'd' for done: 1
Cheese sauce added. Total topping cost: $1.50
Choose a topping (1-17) or type 'd' for done: 2
Pesto sauce added. Total topping cost: $3.00
Choose a topping (1-17) or type 'd' for done: 15
Slow cooked lamb added. Total topping cost: $8.50
Choose a topping (1-17) or type 'd' for done: 13
Premium Italian Sausage added. Total topping cost: $14.00
Choose a topping (1-17) or type 'd' for done:
```

Version Control Evidence



main_v1.py

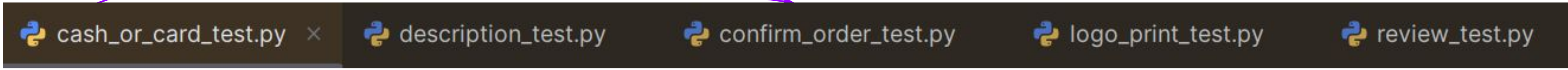
main_V2.py ×

main_v3.py

main_v4.py ×

```
print(“I kept things organized by creating separate
main files for each version to manage major changes and
differences effectively. This approach allowed me to
keep the project structured, making it easier to test
and fix issues. Additionally, it ensured that I could
maintain and reference older versions without impacting
the development of newer ones, leading to a more
efficient workflow.”)
```

Testing file for testing different things such as cash credit options menu and topping calculator



cash_or_card_test.py ×

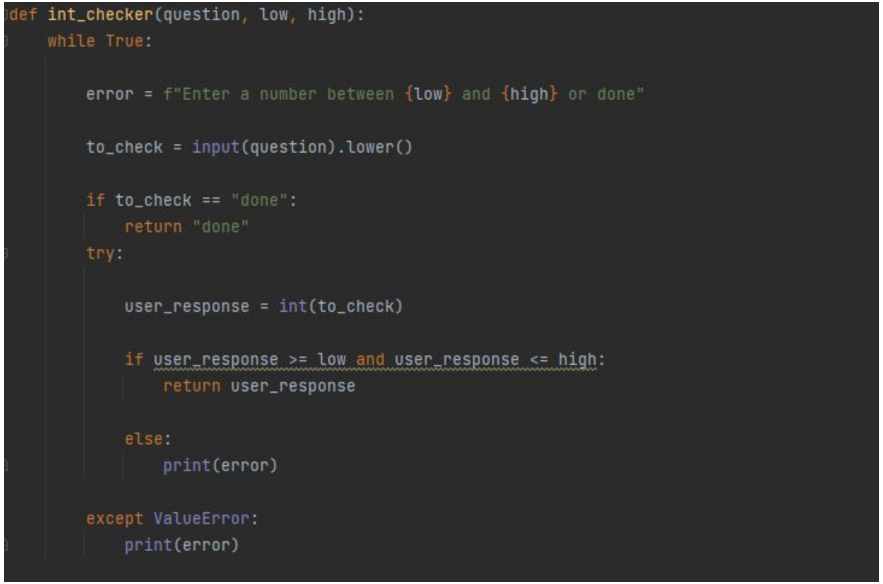
description_test.py

confirm_order_test.py

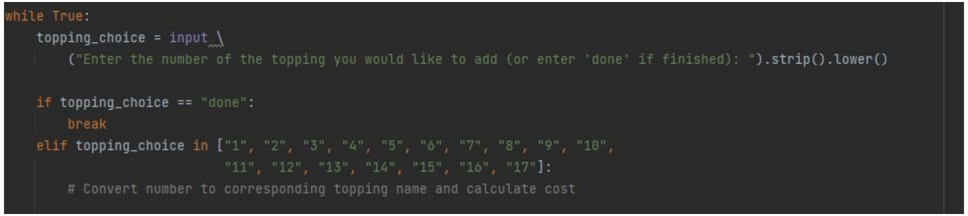
logo_print_test.py

review_test.py

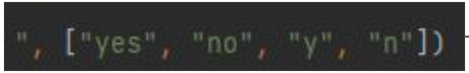
Other random additions



```
def int_checker(question, low, high):
    while True:

        error = f"Enter a number between {low} and {high} or done"

        to_check = input(question).lower()

        if to_check == "done":
            return "done"
        try:

            user_response = int(to_check)

            if user_response >= low and user_response <= high:
                return user_response

            else:
                print(error)

        except ValueError:
            print(error)
```



```
while True:
    topping_choice = input \
        ("Enter the number of the topping you would like to add (or enter 'done' if finished): ").strip().lower()

    if topping_choice == "done":
        break
    elif topping_choice in ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
                            "11", "12", "13", "14", "15", "16", "17"]:
        # Convert number to corresponding topping name and calculate cost
```



```
", ["yes", "no", "y", "n"])
```

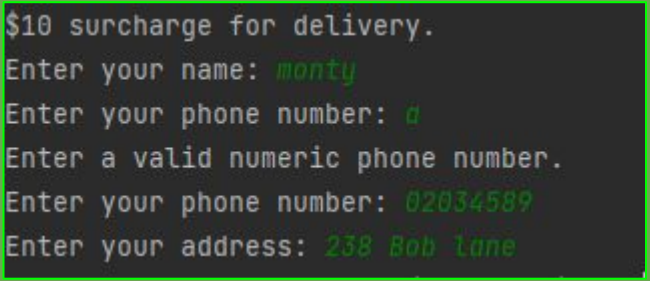
Added more options

Style Errors

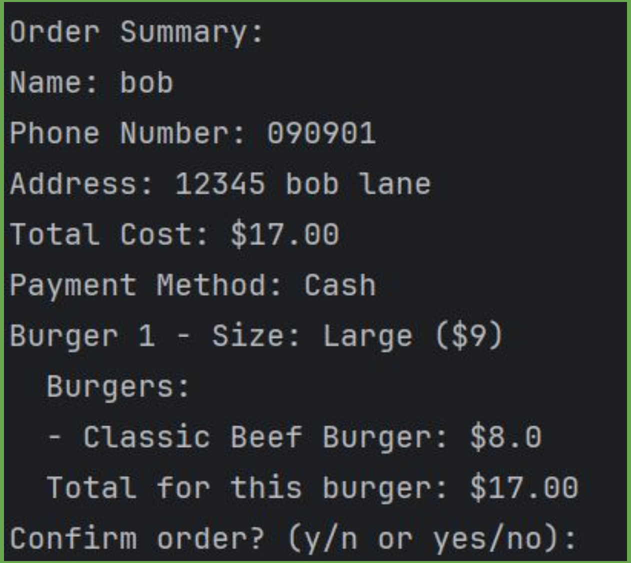
Error document link

Fixed Error document link

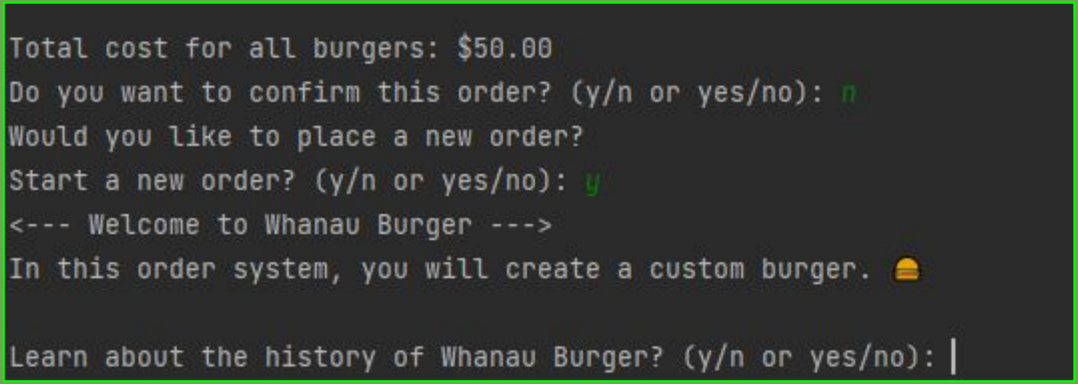
Burger additions



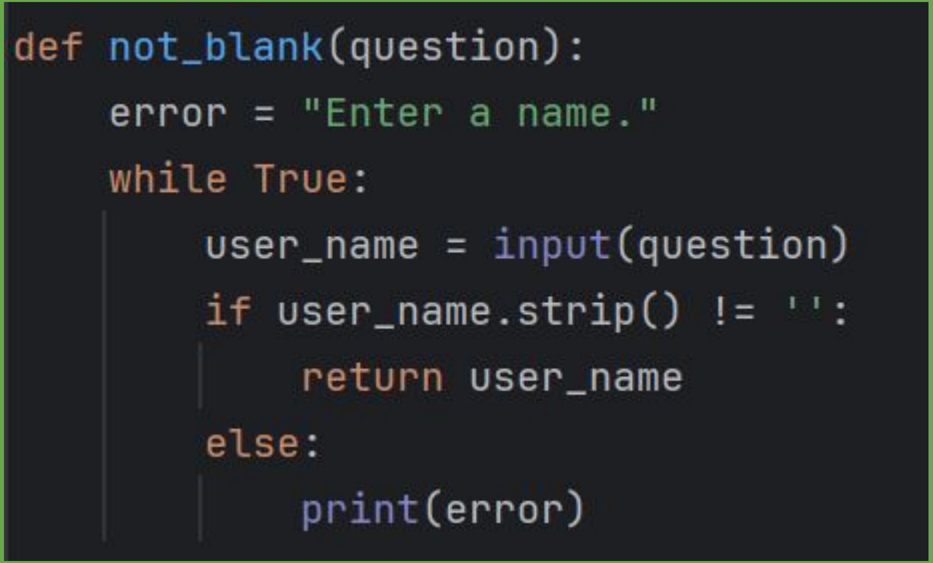
```
$10 surcharge for delivery.
Enter your name: monty
Enter your phone number: a
Enter a valid numeric phone number.
Enter your phone number: 02034589
Enter your address: 238 Bob lane
```



```
Order Summary:
Name: bob
Phone Number: 090901
Address: 12345 bob lane
Total Cost: $17.00
Payment Method: Cash
Burger 1 - Size: Large ($9)
  Burgers:
  - Classic Beef Burger: $8.0
  Total for this burger: $17.00
Confirm order? (y/n or yes/no):
```



```
Total cost for all burgers: $50.00
Do you want to confirm this order? (y/n or yes/no): n
Would you like to place a new order?
Start a new order? (y/n or yes/no): y
<--- Welcome to Whanau Burger --->
In this order system, you will create a custom burger. 🍔

Learn about the history of Whanau Burger? (y/n or yes/no):
```



```python
def not_blank(question):
    error = "Enter a name."
    while True:
        user_name = input(question)
        if user_name.strip() != '':
            return user_name
        else:
            print(error)
```

Discusses how the information from planning, testing and trialling of components assisted in the development of a high-quality outcome: The

The information gathered during planning, testing, and trialling was crucial in developing a high-quality pizza selection tool.

Planning Trello provided a structured approach to the project, allowing for a clear understanding of the requirements. The organization facilitated by Trello made the project more manageable, enabling effective time management and contributing to the overall quality of the final program. By breaking down tasks into manageable components helped me stay focused and on track, ultimately leading to a more refined and comprehensive tool.

Testing played a vital role in identifying potential issues and areas for improvement. Through rigorous testing, I was able to pinpoint bugs, usability problems, and other challenges that could hinder the user experience. Each identified issue was addressed, leading to iterative improvements in the program's functionality and stability. Testing also revealed which features were most effective and which needed refinement, ensuring that the final product met the users' needs and expectations.

Discusses how the information from planning, testing and trialling of components assisted in the development of a high-quality outcome: The

Trialling allowed for real-world application and small adjustments that fine-tuned the program's performance. By trialling the tool in different scenarios, I was able to make minor tweaks that enhanced its usability and efficiency. This approach ensured that the final product was not only functional but also smooth for the user.

Another key aspect of the development process was

gathering feedback from others who tested the program. This input provided valuable insights that I might have overlooked. By incorporating feedback from users with different perspectives, I was able to make further improvements, resulting in a program that was both user-friendly and robust. The process of assessing, refining, and adjusting based on testing and trialling ultimately led to a well-developed, effective program that had significantly evolved since the first version.