Papa's Pizza

Part 1

1.1 Investigate

Project Development Timeline

1. **Requirements (3 days)**
   - Talk with representatives of Papa Pizza and identify the pizza shop's requirements and document the requirements that need to be met.

2. **Design (1 week)**
   - Brainstorm a user-friendly interface for quick access of certain features via a touch screen, using wireframes.
   - Select appropriate frameworks to meet hardware compatibility.

3. **Implementation and Testing (5 weeks)**
   - Contact third-party services in means of databases, payment, and general management.
   - Develop the back-end programs for receiving and processing orders; tracking cumulative totals; additional home delivery charges; loyalty discounts; addition of Goods and Service Tax (GST).
   - Trial testing of the program.
   - Trial testing within the pizza shop's kiosk systems during the shop's inactive hours.

4. **Deployment (1 week)**
   - Deploy the application to the pizza shop's servers via the use of a cloud hosting platform.
   - Train staff on kiosk usage with application implementation.
   - Actively monitor the application usage on initial days of launch.

5. **Future Support (Ongoing)**
   - Provision of ongoing support and application maintenance via company support or subcontractors.

Project Outline

- To develop a software solution for Papa Pizza, with the goal of using a system to maintain calculation and management tasks when pizza orders take place, to be used by employees of the shop.
- This includes functions like:
  - Receiving and Processing Orders.

  - Calculating costs of individual pizzas.
  - Tracking the cumulative total .
  - Providing a daily summary of the quantities sold
  - Accounting for specific discounts and surcharges.
    - Homedelivery.
    - Loyalty discounts.
    - GST addition.
- Therefore, making it easier for the business to efficiently manage in-store and delivery-based orders.

Project Description

Objective

- This system is for employees to effectively manage and calculate costs for various orders in an efficient manner.

Discounts and Surcharges

- Back-end programming will account for such measures, with a terminal based application for the employee to select options.

Tracking Orders and Daily Sales

- Using back-end programming, information will be stored in long-term storage via a file.

Requirement List

Functional Requirements

- ***Order Placement and Processing***: Employees can select specific options through terminal menu, and don't stop application until quit is selected as an option.
- ***Cost Calculation***: Total cost is calculated based on items selected (with set prices) as well as discounts and surcharges (including 10% GST).
- ***Sales Tracking***: Sales data is recorded into a notepad file in cloud/server of the shop, allowing for later analysis of the data for marketing improvements.
- ***Discount Application***: $8.00 delivery fee; 5% discount to members with a loyalty card or if the orders exceed $100.

Non-Functional Requirements

- ***Usability***: Interface for employee usage with quick and efficient functions.
- ***Performance***: Light-weight application for speed and reliability.
- ***Security***: Encrypted storage procedures and secure payment architecture through trusted third-party services.
- ***Legal***: Compliance with financial laws and privacy laws.

1.2 Design – Pseudocode

****pseudocode.txt is under the main branch on the github page.**

*1.2.2 Design – Testing logic using Trace Tables (**add_pizza** Function)*

add(pizza, quantity)	subtotal	delv()	loy(loyalty)	discount()	gst()	total()	User output
Pepperoni, 1							
	21.00						
		false					
			false				
				21.00			
					2.1		
						23.1	
							23.10
bbq meatlovers, 28							
	714						
		true (+8.00)					
			false				
				685.9			
					68.59		
						754.49	
							754.49
margherita, 2							
	37						
		false					
			true				
				35.15			
					3.515		
						38.665	
							38.66

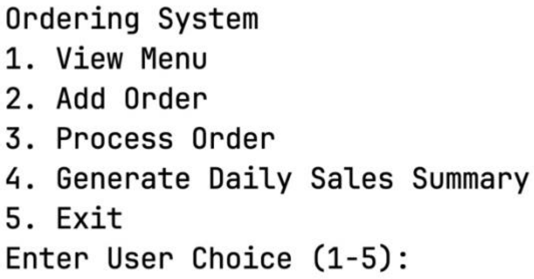
Part 2

****main.py is under the main branch on the github page.**

Part 3

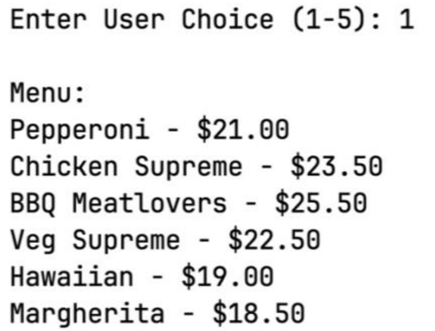
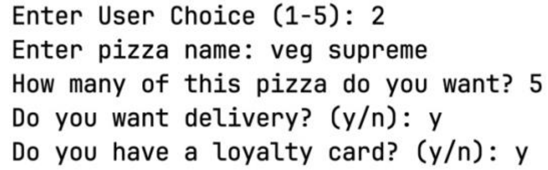
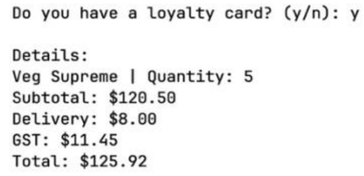
3.1 Testing

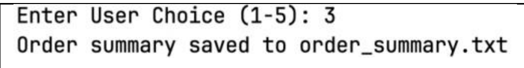
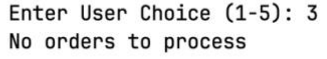
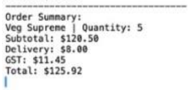
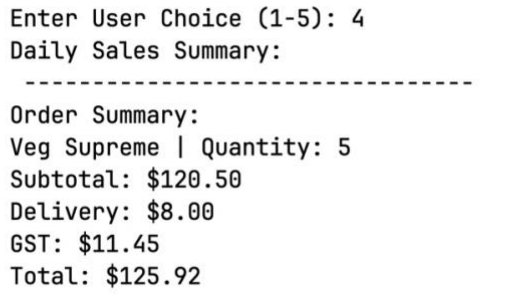
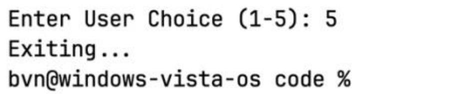
When program is run, the user is greeted with an option menu:



```
Ordering System
1. View Menu
2. Add Order
3. Process Order
4. Generate Daily Sales Summary
5. Exit
Enter User Choice (1-5):
```

Scenario → Veg Supreme – 5 pizzas – delivery (y) – loyalty card (y)

User Input	***Expected Output***	***Actual Output***	***Terminal View of Output***	***Pass/Fail***
Menu Option (1)	*Displays Pizzas with corresponding price.	*Displays Pizzas with corresponding price.	 Enter User Choice (1-5): 1 Menu: Pepperoni - $21.00 Chicken Supreme - $23.50 BBQ Meatlovers - $25.50 Veg Supreme - $22.50 Hawaiian - $19.00 Margherita - $18.50	**Pass**
Menu Option (2)	Prompt user for pizza name, quantity, delivery (y/n) and loyalty (y/n)	Prompt user for pizza name, quantity, delivery (y/n) and loyalty (y/n)	 Enter User Choice (1-5): 2 Enter pizza name: veg supreme How many of this pizza do you want? 5 Do you want delivery? (y/n): y Do you have a loyalty card? (y/n): y	**Pass**
Enter a valid pizza name (veg supreme), enter integer quantity (5), delivery (y), loyalty (y)	Order Details should be the output, including thse subtotal, delivery, gst and the total. The total has to have the surcharges.	Order Details is the output, including the subtotal, delivery, gst and the total. The total has to have the surcharges.	 Do you have a loyalty card? (y/n): y Details: Veg Supreme | Quantity: 5 Subtotal: $120.50 Delivery: $8.00 GST: $11.45 Total: $125.92	**Pass**

Menu Option (3)	If the total > 0 then generate .txt file with specific order format.	Order Summary file was generated.	 Enter User Choice (1-5): 3 Order summary saved to order_summary.txt	**Pass**
	Displays the message “No orders to process” if Total = 0	Displays the message “No orders to process” if Total = 0	 Enter User Choice (1-5): 3 No orders to process  -------------------------------- Order Summary: Veg Supreme | Quantity: 5 Subtotal: $120.50 Delivery: $8.00 GST: $11.45 Total: $125.92	**Pass**
Menu Option (4)	Output data from when the process (3) option was selected. [printing data out from the .txt file onto the terminal]	Output data from when process option was selected.	 Enter User Choice (1-5): 4 Daily Sales Summary: -------------------------------- Order Summary: Veg Supreme | Quantity: 5 Subtotal: $120.50 Delivery: $8.00 GST: $11.45 Total: $125.92	**Pass**
Menu Option(5)	Exits the program [breaking the menu loop]	Exits the program	 Enter User Choice (1-5): 5 Exiting... bvn@windows-vista-os code %	**Pass**

3.2 Evaluation

The program meets the system requirements of the business as the menu loops till exited and stores all data in a .txt file. It can collectively calculate orders and store them in local memory. The program can eventually have a GUI implemented and installed on kiosk systems to be of usage soon. The program works accordingly with the current menu and the worker can take basic orders by selecting the pizza type, quantity of pizza, delivery option and loyalty card option. All calculations including GST subtotal can be performed by the program. The program can be implemented into a cloud-based or local server, to store customer orders. Possibly could be encrypted if business decides to work with EFTPOS payments.

The program could be improved through making the code less redundant, but the program passed all the tests (3.1 – Testing section). According to peer feedback, the program passed most prompts only having an error, once which was fixed with a restart of the program.

Not many changes were made to the final program, although the contents per object had a slight change made to them (i.e.: the orders and main class which contains most of the program's functions, with the main class holding the looping menu – the main function calls on the object class to perform its functions).

- The planning worked well as I was able to efficiently complete it.
- The programming, since it took 6 rewrites to complete the program which constantly ran into indentation and definition errors, which were eventually solved with multiple searches on GitHub and Stack Overflow.
- I would try to find a similar program on GitHub before starting to refer to, then I would replicate the repository and modify the program to my requirements.
- An rare error that occurred sometimes (couldn't find it during the testing period), when stacking orders, the previous order may get repeated onto the .txt file during processing.

Bibliography

[1] Glasshost, "How to save user input to a File in Python," *Medium*, May 04, 2023. https://medium.com/@glasshost/how-to-save-user-input-to-a-file-in-python-20c3b2947266 (accessed Apr. 18, 2024).

[2] "How to delete data from file in Python," *GeeksforGeeks*, Jan. 17, 2020. https://www.geeksforgeeks.org/how-to-delete-data-from-file-in-python/

[3] Greg, "ubuntu - How do I print the content of a .txt file in Python?," *Stack Overflow*, Dec. 07, 2015. https://stackoverflow.com/questions/18256363/how-do-i-print-the-content-of-a-txt-file-in-python

[4] “Print the Content of a Txt File in Python,” *GeeksforGeeks*, Feb. 14, 2024. https://www.geeksforgeeks.org/print-the-content-of-a-txt-file-in-python/ (accessed Apr. 23, 2024).

[5] MrBrownCS, “How Do I Answer Trace Table Questions?,” *www.youtube.com*, Mar. 16, 2021. https://www.youtube.com/watch?v=8kTHGgTUZ2o (accessed Jun. 23, 2022).

[6] OpenAI, “ChatGPT | Using it to debug some parts of the code,” *chat.openai.com*, 2024. https://chat.openai.com/ (accessed Apr. 23, 2024).

[7] “Gemini | Using it to debug some parts of the code,” *gemini.google.com*. https://gemini.google.com (accessed Apr. 23, 2024).

[8] chaitanya kurakula, “Whenever I try to execute the program it says" Pizza does not refer to an value" Can somebody explain why,” *Stack Overflow*, Apr. 02, 2023. https://stackoverflow.com/questions/75914920/whenever-i-try-to-execute-the-program-it-says-pizza-does-not-refer-to-an-value (accessed May 01, 2024).

[9] Todd, “For a school project on Python, I need to write a program that allows users to order pizza. The user must be able to order as many pizzas as they want,” *Stack Overflow*, May 20, 2020. https://stackoverflow.com/questions/61924297/for-a-school-project-on-python-i-need-to-write-a-program-that-allows-users-to-o (accessed May 01, 2024).

[10] O. Polat, “python - first pizza order program,” *Code Review Stack Exchange*, Sep. 26, 2020. https://codereview.stackexchange.com/questions/249903/first-pizza-order-program (accessed May 01, 2024).

[11] elisrae, “Cleaning up a basic Python script that takes and recites a pizza order,” *Code Review Stack Exchange*, Oct. 27, 2021. https://codereview.stackexchange.com/questions/269455/cleaning-up-a-basic-python-script-that-takes-and-recites-a-pizza-order (accessed May 01, 2024).

[12] A. Kaşoğlu, “kasoglu/pizza-deliver-with-python,” *GitHub*, Oct. 28, 2021. https://github.com/kasoglu/pizza-deliver-with-python (accessed May 01, 2024).