A Project Report on

VENDING MACHINE

Submitted in partial fulfillment of requirements for the award of the course of

**EGA1202 – PYTHON PROGRAMMING**

Under the guidance of

**Mr. M.YUVARAJ M.E.,**

**Assistant Professor / EEE**

Submitted By

**SUJITHKUMAR S (927623BEE107)**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

# M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

**KARUR – 639 113**

DECEMBER 2024

**M. KUMARASAMY COLLEGE OF ENGINEERING**

**(Autonomous Institution affiliated to Anna University, Chennai)**

**KARUR – 639 113**

**BONAFIDE CERTIFICATE**

Certified that this project report on **“VENDING MACHINE”** is the bonafide work of **SUJITHKUMAR S (927623BEE107)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

|  |  |
| --- | --- |
| Signature | Signature |
| **Mr.M.YUVARAJ M.E.,** | **Dr. J.UMA M.E.,Ph.D.,** |
| **SUPERVISOR,** | **HEAD OF THE DEPARTMENT,** |
| Department of Electrical & Eletronics and Engineering, | Department of Electrical and Electronics Engineering, |
| M.Kumarasamy College of Engineering, | M.Kumarasamy College of Engineering, |
| Thalavapalayam, Karur -639 113. | Thalavapalayam, Karur -639 113. |



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**VISION OF THE INSTITUTION**

To emerge as a leader among the top institutions in the field of technical education

**MISSION OF THE INSTITUTION**

* Produce smart technocrats with empirical knowledge who can surmount the global challenges
* Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
* Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

**VISION OF THE DEPARTMENT**

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility

**MISSION OF THE DEPARTMENT**

* Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects
* Inculcate the students in problem solving and lifelong learning ability
* Provide entrepreneurial skills and leadership qualities
* Render the technical knowledge and skills of faculty members Produce hi-tech professionals in the field of Electrical and Electronics Engineering by inculcating core knowledge

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO 1: Core Competence:** Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering

**PEO 2: Professionalism:** Graduates will provide feasible solutions for the challenging problemsthroughcomprehensiveresearchandinnovationinthealliedareasofElectronicsandCommuni cationEngineering.

**PEO 3: Lifelong Learning:** Graduates will contribute to the social needs through life long learning, practicing professional ethics and leadership quality.

**PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions

**11.Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12.Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

**PSO1:** Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.

**PSO2:** Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfil the industrial expectations.

v

# ABSTRACT

The Python program is a simulation of a vending machine that allows users to purchase items conveniently. It features a predefined menu of items, each associated with a unique number, name, and price. Users can interact by selecting an item and inserting money. The program displays the available options, validates the user’s selection, and calculates whether the inserted amount is sufficient to cover the item's cost. If the payment is adequate, the selected item is dispensed, and any change is returned to the user. The program handles scenarios like insufficient funds, invalid inputs, and incorrect item selections by prompting the user appropriately to retry. Users can exit the vending machine at any time by selecting the quit option. This program highlights fundamental programming concepts, including dictionaries for storing data, input validation, control structures like loops and conditionals, and basic arithmetic operations for calculating change. Its structured design ensures both functionality and ease of use, making it a practical application of Python programming.

**ABSTRACT WITH POs AND PSOs MAPPING**

|  |  |  |
| --- | --- | --- |
| **ABSTRACT** | **POs**  **MAPPED** | **PSOs**  **MAPPED** |
| The Python program simulates a simple vending |  |  |
| machine system. It includes a menu of items, each with | **PO1(2)** |
| a unique number, name, and price. Users can interact | **PO2(2)** |
| with the vending machine by selecting an item and | **PO3(2)** |
| inserting money. The program displays available items, | **PO4(2)** |
| processes user choices, and calculates the total amount | **PO5(3)** |
| inserted. If sufficient funds are provided, the program | **PO6(1)** |
| dispenses the selected item, calculates any change due, | **PO7(3)** |
| and provides a confirmation message. In cases of | **PO8(2)** |
| insufficient funds or invalid inputs, it prompts the user | **PO9(3)** |
| to reattempt the process. The program allows users to | **PO10(3)** |
| quit at any time, ensuring a user-friendly experience. It | **PO11(2)** |
| demonstrates core programming concepts such as loops, | **PO12(2)** |
| conditionals, dictionary handling, and input validation. |  |

Note: 1- Low, 2-Medium, 3- High

**SUPERVISOR HEAD OF THE DEPARTMENT**

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **No.** | **TITLE** | **PAGE**  **No.** |
|  | **ABSTRACT** | vi |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 Objective | 1 |
|  | 1.2 Overview | 1 |
|  | 1.3 Python Programming concepts | 2 |
| **2** | **PROJECT METHODOLOGY** | **3** |
|  | 2.1Proposed Work | 3 |
|  | 2.2 Block Diagram | 5 |
| **3** | **MODULE DESCRIPTION** | **6** |
|  | 3.1 Display Items | 6 |
|  | 3.2 User Input Module | 6 |
|  | 3.3 Selection of Item | 6 |
|  | 3.4 Inserting Money | 7 |
|  | 3.5 Output of Item | 7 |
| **4** | **RESULTS AND DISCUSSION** | **8** |
| **5** | **CONCLUSION** | **12** |
|  | **REFERENCES** | **13** |
|  | **APPENDIX** | **14** |

**CHAPTER 1** **INTRODUCTION**

* 1. **Objective**

The objective of the vending machine program is to create a simple and interactive system that mimics the functionality of a real vending machine. It allows users to select items from a menu, insert money, and receive their chosen item along with any change if applicable. The program ensures accurate calculations, validates user inputs, and handles errors gracefully, providing a smooth and user-friendly experience. It serves as a practical application of Python programming concepts, including loops, conditionals, and data handling.

* 1. **Overview**

The vending machine program allows users to select from a variety of items such as Soda, Chips, Candy, and Water. Each item is associated with a price, and users can insert money to make a purchase. The program displays available items and prompts users to select one by entering a number. If the inserted money is sufficient, the item is dispensed, and any change is returned. If the funds are insufficient, the user is prompted to insert more money. The user can also exit the program by entering 'q'. The program ensures proper handling of inputs, providing feedback for both successful and unsuccessful transactions.

* 1. **Python Programming Concepts**

The vending machine program in Python incorporates several fundamental programming concepts that are crucial for building functional applications. It starts by using dictionaries to store the details of the items available in the machine, such as their names and prices, making it easy to manage and access this data. Variables are used to represent each item's attributes and store user inputs, such as the amount of money inserted. The for loop is utilized in the display\_items() function to iterate through the items and display them dynamically based on the data stored in the dictionary.

The program uses control flow statements like if, else, and elif to handle user interactions and guide the program's logic. For instance, it checks if the user’s selection is valid and whether they have enough money for their desired purchase. Functions are used to organize the code into reusable blocks. The process\_purchase() function handles the transaction logic, including calculating and returning any change, while display\_items() focuses on showing the menu.

Exception handling with try and except ensures that if a user enters a non-numeric value when inserting money, the program will not crash but instead prompt them with an error message. The program also demonstrates input validation, ensuring that only valid choices are processed and guiding users through the purchase process. Additionally, the while loop enables continuous operation of the vending machine until the user chooses to exit, making the program interactive and user-friendly. Through these concepts, the vending machine program effectively handles user input, data processing, and error management.

* 1. **Proposed Work**

# CHAPTER 2 PROJECT METHODOLOGY

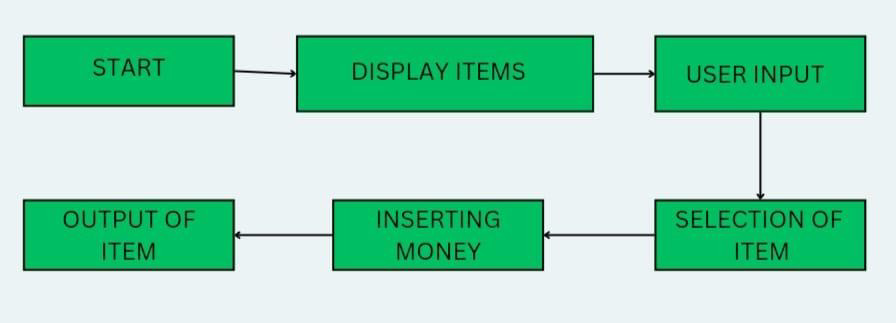
The proposed vending machine program utilizes several key concepts to create a functional, interactive application. The program begins by defining a dictionary to store the available items in the vending machine, including their names and prices. This structure allows easy access to item details and is highly flexible for future modifications, such as adding new products or updating prices. The display\_items() function leverages a for loop to iterate over the dictionary and display each item to the user dynamically, making the program scalable and user-friendly.

The control flow is managed with if statements to guide the user’s experience. For example, if the user inputs an invalid selection, the program will prompt them to try again. If the user has insufficient funds, the program will inform them and request more money. The program is also designed to handle user input effectively, with exception handling to ensure the input is valid (e.g., when entering money). If the user enters a non-numeric value, the program will catch the error and ask them to input a valid amount.

The while loop is crucial for keeping the vending machine running until the user decides to exit by entering 'q'. This allows the system to continuously prompt the user for a choice and handle transactions until they are finished. The functions process\_purchase() and display\_items() separate the logic into manageable blocks, improving the code's readability and maintainability. The program uses string formatting to display prices and change in a clean, user-friendly manner.

Additionally, the program could be extended to include a payment validation system, allowing for different payment methods (coins, cards) and more advanced features like tracking inventory . The dictionaries and lists, provide a solid foundation for managing the items and transactions, making it easy to expand and customize the vending machine as needed. Overall, the program showcases key Python programming concepts such as control flow, functions, exception handling, and input validation in a real-world application.

* 1. **Block Diagram**



**Figure.2.2.1-Block Diagram**

* 1. **Display Items**

# CHAPTER 3 MODULE DESCRIPTION

The display item module displays the available products in the vending machine along with their prices. It uses a dictionary to store the items, where the keys are the item numbers (as strings) and the values are dictionaries that contain the item name and price. The display\_items() function iterates over each item and prints its details in a formatted way. The function makes it easy to update the product list or add new items in the future.

* 1. **User Input Module**

The program prompts the user to make a choice from the displayed items. The user can select an item by entering its number or quit the program by typing 'q'. The input is validated to ensure that the user either selects a valid item or chooses to exit. If the user makes an invalid choice (e.g., entering a number not associated with any item), the program will prompt them again.

* 1. **Selection of Item**

Once the user selects a valid item, the program retrieves the item details (name and price) from the dictionary and prepares for the payment process. If the user selects an invalid option or enters an invalid key, the program will ask them to try again. This step helps in ensuring that only valid products are processed for purchase.

* 1. **Inserting Money**

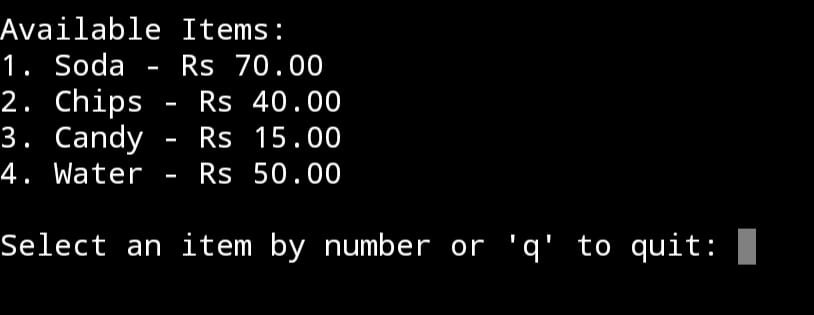
In this module, the program prompts the user to insert money. It ensures that the amount entered is numeric by using a try and except block to catch invalid inputs (such as text or symbols). If the user enters a valid amount, it is then compared with the price of the selected item. If the amount is sufficient, the program proceeds; otherwise, it prompts the user to insert more money.

* 1. **Output of Item**

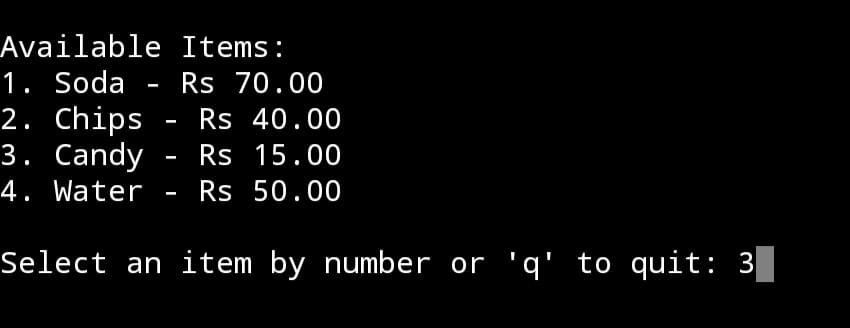
After the user inserts the money, this module processes the purchase. It checks if the amount entered is sufficient to cover the price of the selected item. If the user has inserted more money than required, the program calculates the change and returns it. If the inserted amount is less, the program will ask the user to insert more money. The transaction is completed with a confirmation message, and the program prepares for the next selection or allows the user to exit.

# CHAPTER 4 RESULTS AND DISCUSSION

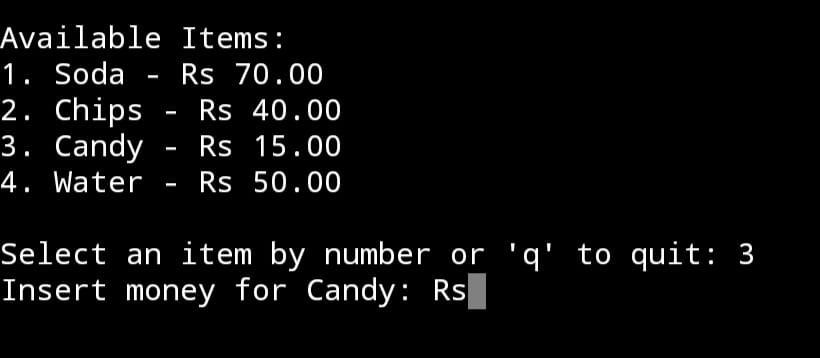
* 1. **Results**
     1. **Display Items**



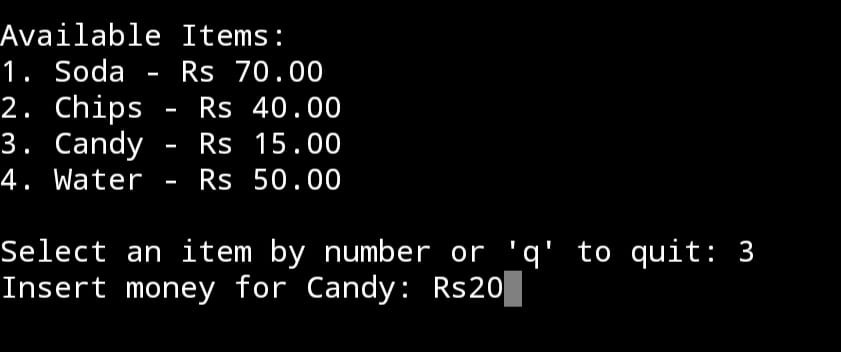
* + 1. **User Input Module**



* + 1. **Selection of Item**



**4.1.3 Inserting Money**



**4.1.5 Output of Item**



# Discussion

The vending machine program is an interactive Python script designed to simulate the functionality of a basic vending machine. It uses a dictionary to store the available items, their prices, and corresponding selection numbers. The program first displays a list of items with their prices and allows the user to select an item by entering its number.

Once a valid selection is made, the program prompts the user to input the amount of money they wish to insert. It verifies if the inserted amount is sufficient to purchase the selected item. If the payment covers the price, the program calculates and returns any change and displays a confirmation message for the purchase. If the amount is insufficient, it prompts the user to add more money. The program also handles invalid inputs, such as entering non-numeric values for the amount or selecting an

unavailable item. A key feature of the program is its continuous loop, allowing the user to make multiple purchases until they choose to exit by entering 'q'. This modular structure, with functions like display\_items and process\_purchase, ensures clarity and maintainability. The program emphasizes user interaction, error handling, and a user- friendly experience, making it a practical demonstration of programming logic and flow control.

# CHAPTER 5 CONCLUSION

The vending machine program is an interactive Python script that emulates the functionality of a real-world vending machine. It provides users with a menu of items, each listed with its price, and allows them to select and purchase an item by inputting its corresponding number. The program ensures a user-friendly experience by validating inputs and handling errors, such as insufficient funds or invalid selections. Upon a successful purchase, it calculates and returns the change (if applicable) and thanks the user. The inclusion of a quit option allows users to exit the program gracefully. Overall, the program demonstrates the effective use of dictionaries, loops, and functions to create a practical, real-world application that emphasizes input validation and user interaction.

**REFERENCES:**

1. “Vending Machine Program in Python”- Geeks for Geeks
2. “Python Vending Machine” - GitHub
3. “Simple Python Vending Machine” - W3schools

# APPENDIX

# Vending Machine Program

# Items in the vending machine items = {

"1": {"name": "Soda", "price": 70},

"2": {"name": "Chips", "price": 40},

"3": {"name": "Candy", "price": 15},

"4": {"name": "Water", "price": 50},

}

def display\_items(): print("\nAvailable Items:") for key, item in items.items():

print(f"{key}. {item['name']} - Rs {item['price']:.2f}") def process\_purchase(choice, amount):

item = items[choice]

if amount >= item['price']: change = amount - item['price']

print(f"\nYou have purchased {item['name']} for Rs{item['price']:.2f}.") if change > 0:

print(f"Your change is: Rs{change:.2f}") print("Thank you for your purchase!")

else:

print("\nInsufficient funds. Please insert more money.") def vending\_machine():

while True: display\_items()

choice = input("\nSelect an item by number or 'q' to quit: ")

if choice == 'q':

print("Exiting the vending machine. Goodbye!") break

if choice in items: try:

amount = float(input(f"Insert money for {items[choice]['name']}: Rs")) process\_purchase(choice, amount)

except ValueError:

print("Invalid input. Please enter a numeric value for money.")

else:

print("Invalid selection. Please try again.") # Run the vending machine program vending\_machine()