

**PURBANCHAL UNIVERSITY**

**HIMALAYAN WHITEHOUSE INTERNATIONAL COLLEGE**

**PUTALISADAK, KATHMANDU**

**A Final Project Report On**

**“Restaurant Billing System”**

By

**Rohan Chaudhary (321040)**

**Saurav Giri (321046)**

SUBMITTED TO THE DEPARTMENT OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF INFORMATION AND TECHNOLOGY

DEPARTMENT OF SCIENCE AND TECHNOLOGY

Kathmandu, Nepal

**February, 2024**

# CERTIFICATE OF APPROVAL

The Project entitled **“Restaurant Billing System”**, submitted by Rohan Chaudhary and Saurav Giri in partial fulfillment of the requirements for the degree of “Bachelor of Information and Technology” has been accepted as a bona fide record of work carried out by them in the department.

Supervisor, Er. Ram Chandra Poudel

Project Guide

Bachelor of Information and Technology

Er. Bimal Sharma

Head of Department

Bachelor of Information and Technology

External

Purbanchal University

# ACKNOWLEDGEMENT

We are very grateful to our supervisor **Er. Ram Chandra Poudel** for his constant encouragement, cooperation and guiding us through this project. We would like to thank our lecturer, friends of Bachelor of Information Technology from our batch as well as the seniors who helped us a lot.

The desired success obtained during wouldn’t have been attained without the facilities and other guideline provided by our Head of Department **Er. Bimal Sharma.** At last, we would like to thank to all those individuals who directly or indirectly helped us in the completion of this project.

# ABSTRACT

The main goal of this project was to develop a console-based billing system for a restaurant using C++ Programming language. This console-based program is designed to administer its users and customers. By using this system, users can easily enter menu items, add items on menu, delete items on menu, edit items, store order history and generate bill effortlessly.

It is an automated program that can handle a lot of information about the restaurant’s menu and generates bill for customer.

The project is entirely implemented in the C++ programming language, utilizing variables, strings, and other relevant concepts for the development. It does not rely on complex graphics features, making it accessible to a wide range of users. With a focus on maintaining data security, the software incorporates a password-based login system. You can also register a new account through a simple process. This ensures that generating bill and placing order to customer will be easier.

TABLE OF CONTENTS

[CERTIFICATE OF APPROVAL i](#_Toc160823891)

[ACKNOWLEDGEMENT ii](#_Toc160823892)

[ABSTRACT iii](#_Toc160823893)

[TABLE OF CONTENTS iv](#_Toc160823894)

[LIST OF FIGURE vi](#_Toc160823895)

[LIST OF CHART vii](#_Toc160823896)

[LIST OF ABBREVIATIONS viii](#_Toc160823897)

[CHAPTER ONE: INTRODUCTION 1](#_Toc160823898)

[1.1 Introduction 1](#_Toc160823899)

[1.2 Statement of problem 1](#_Toc160823900)

[1.3 Motivation 1](#_Toc160823901)

[1.4 Objective 2](#_Toc160823902)

[1.5 Scope 2](#_Toc160823903)

[1.6 Target Audience 2](#_Toc160823904)

[1.7 Development Methodology 2](#_Toc160823905)

[CHAPTER TWO: LITERATURE REVIEW 4](#_Toc160823906)

[2.1 Background Study 4](#_Toc160823907)

[2.2 Literature Review 4](#_Toc160823908)

[CHAPTER THREE: SYSTEM ANALYSIS 6](#_Toc160823909)

[3.1 Requirement Analysis 6](#_Toc160823910)

[3.2 Feasibility Analysis 6](#_Toc160823911)

[CHAPTER FOUR: SYSTEM DESIGN 9](#_Toc160823912)

[4.1 Algorithm Details 9](#_Toc160823913)

[4.2 Flowchart 10](#_Toc160823914)

[CHAPTER FIVE: IMPLEMENTATION AND TESTING 12](#_Toc160823915)

[5.1. Implementation 12](#_Toc160823916)

[5.2 Testing 12](#_Toc160823917)

[5.3 Result Analysis 13](#_Toc160823918)

[CHAPTER SIX: CONCLUSION AND FUTURE RECOMMENDATION 14](#_Toc160823919)

[**6.1** **Conclusion** 14](#_Toc160823920)

[**6.2** **Future Enhancement** 14](#_Toc160823921)

[REFERENCE 15](#_Toc160823922)

[APPENDICES 16](#_Toc160823923)

# LIST OF FIGURE

[Figure 1: flowchart diagram of restaurant billing system 11](#_Toc160084895)

# LIST OF CHART

[Chart 1: Gantt chart of restaurant billing system 8](#_Toc160043119)

# LIST OF ABBREVIATIONS

C++ C Plus Plus

FMS File Management System

RBS Restaurant Billing System

# CHAPTER ONE: INTRODUCTION

## Introduction

A restaurant is a business that prepares and serves food and drinks to customers. Meals are generally served and eaten on the premises, but many restaurants also offer take-out and food delivery services. By facilitating calculating bill manually over program-based billing automatically.

The “Restaurant Billing System” or “RBS” is an application to automate the process of information recording and billing of a restaurant. This console-based application is designed to administer its users and customers. RBS is a billing system, made for the effective utilization of modern technology in the Restaurant. It is an automated program that can handle a lot of information about the restaurant’s menu, prices of items listed on menu, generates bill for customer. It has the capability to process bills and gather information. It is designed for the sole purpose of efficiency, speed and accuracy. Overall, for modern Restaurant to function effectively and efficiently, an RBS is a vital instrument.

## Statement of problem

The billing process is done manually by manpower. It results in delayed time for the consumer and to the organization while the bill is being processed. So, there is a room for improvement here. A certain computer-based billing system could aid the organization to utilize its resources better. Computerized Billing System provide capabilities for entering client, employee and payment information, building a record and managing other related data needs in the organization. Improvements have been made. We anticipate that the issues will be fully resolved soon.

## Motivation

The motivation behind this project is to develop a robust and comprehensive restaurant billing system using the C++ programming language. The C++ programming language is a top choice for creating software applications across many industries, including the banking sector. It is renowned for its effectiveness, portability, and wide library support.

By developing a “RBS” in C++ programming language, we aim to revolutionize the efficiency and effectiveness of billing operations in restaurants. The deployment of the system will lead to increased operational effectiveness, less expenses, improved risk management, and eventually higher profitability for the hotels and restaurants. Process simplification, quicker bill generation and a whole user experience will all benefit customers.

## Objective

The project is with the main aim of meeting the following objectives:

* To ensure precise billing calculation.
* To optimize resource usage.
* To automate order taking from customers.

## Scope

The system should be user-friendly, efficient, and capable of managing various aspects such as menu items, prices, orders, and generating bills. Here's a high-level overview of the scope and components of a restaurant billing system in C++:

* **Menu Display and Selection.**
* **Precise Billing Calculation**.
* **Resource Optimization**.
* Develop a user-friendly interface for restaurant staff.
* Save and retrieve data, such as menu items and their price, using file handling as a database.

## Target Audience

The **target audience** for a restaurant billing system includes **restaurant owners, managers, and staff**. These individuals are directly involved in managing orders, billing, within the restaurant. The system caters to their needs by automating processes and ensuring accurate billing.

## Development Methodology

The methodology for developing RBS in C++ outlines the systematic approach and processes to be followed to ensure the successful completion of the project. This methodology encompasses project planning, development, testing, deployment, and ongoing maintenance. The selected methodology aligns with the project's goals of creating a feature-rich, secure, and scalable restaurant billing system.

Key Development Phases:

* Requirements Gathering: Thoroughly understand the needs of the restaurant, including menu details, billing rules, and customer expectations.
* Design and Architecture: Create a well-thought-out architecture that supports scalability and maintainability.
* Testing: Implement rigorous testing procedures to ensure the system functions correctly and is free of errors.
* Documentation: Maintain documentation for the system's design, user manuals, and support materials.
* User Training: Train restaurant staff on how to use the system effectively.
* Feedback and Continuous Improvement: Continuously gather feedback from users and stakeholders to make improvements and updates to the system.
* Security and Compliance: Pay attention to security practices and ensure compliance with relevant regulations, especially regarding financial transactions and customer data.

# CHAPTER TWO: LITERATURE REVIEW

## 2.1 Background Study

A restaurant is a [business](https://en.wikipedia.org/wiki/Business) which prepares and serves food and drinks to customers in exchange for money, either paid before the meal, after the meal, or with an open account. Meals are generally served and eaten on premises. Restaurants vary greatly in appearance and offerings, including a wide variety of [cuisines](https://en.wikipedia.org/wiki/Cuisine) and [service](https://en.wikipedia.org/wiki/Customer_service) models ranging from inexpensive [fast food](https://en.wikipedia.org/wiki/Fast_food) restaurants to high-priced luxury establishments.

Restaurants are one of the growing businesses in recent times. In Nepal, many restaurants are doing very well by providing quality service to its customers. It has to deal with a lot of information about its client and employees, all of which are kept and recorded manually and physically in a drawer or a room. Due to this situation, finding any reliable data when required and in time is not possible. In today's highly competitive restaurants industry, the need for efficient and effective billing system is more significant.

## 2.2 Literature Review

The review of previous literature is essential for understanding the existing knowledge and research conducted on restaurant billing systems developed using the C++ programming language. This review aims to summarize and analyze relevant literature to identify key trends, challenges, and advancements in this field. It serves as a foundation for the current project, providing valuable insights and informing the development of an efficient and reliable RBS in C++ programming language.

Overview of Existing Literature:

Author Dr. Sheryl E. Kimes: Dr. Kimes This study shows the design and implementation of a restaurant billing system in C++ involves creating a user-friendly interface, choosing appropriate data structures, managing the menu, processing orders, generating bills, managing data and reports, implementing error handling and validation, and conducting thorough testing and debugging. By following these steps, a functional and efficient billing system can be developed to streamline operations and enhance the customer experience in a restaurant setting. [1]

Authors: Gary Macgraw this paper shows Security measures for restaurant billing systems implemented in C++ include implementing input validation to prevent malicious inputs, ensuring secure data storage through encryption techniques, using access control mechanisms to restrict system access, utilizing secure communication protocols, implementing audit trails and logging for monitoring purposes, regularly updating software to address security vulnerabilities, enforcing strong user authentication and password policies, implementing role-based access control for managing user privileges, conducting security testing and code reviews to identify vulnerabilities, and implementing regular backups and disaster recovery planning to protect data. [2]

The use of C++ programming language in Restaurant Billing systems offers advantages such as efficient memory management, low-level control, and extensive library support. Security measures, including encryption, authentication, and access control, are essential. Performance optimization techniques, such as code optimization and efficient data structures, are crucial for achieving high system performance.

The review of previous literature on restaurant billing systems developed in C++ programming language highlights the significance of this approach in designing robust, efficient, and secure systems. The studies emphasize the advantages of C++ programming language. The importance of implementation of security measures, and performance optimization techniques. By building upon the insights from existing research, the current project aims to develop a Restaurant billing system that leverages the capabilities of C++ programming language to provide a reliable, efficient, and secure solution for the restaurant and hotels.

# CHAPTER THREE: SYSTEM ANALYSIS

System analysis involves a thorough examination of a system's processes and interrelationships, focusing on identifying flaws and proposing solutions. It begins when a user or management assesses the current system, collecting data through methods like Data Flow Diagrams. Successful analysis hinges on problem identification, thorough investigation, and proper solution selection. An effective analytical model encompasses problem comprehension and solution framework. This necessitates extensive data gathering. The suggested system is then rigorously evaluated against requirements.

## 3.1 Requirement Analysis

**3.1.1. Functional Requirement**

The system shall provide an interface for restaurant staff to manage menu items, including the ability to insert, edit, and delete menu items. Upon inserting a new menu item, the system shall prompt the user to input the item's name and price. When editing a menu item, the system shall allow the user to modify the item's name and price. Deleting a menu item shall remove it from the menu list. These menu management functionalities ensure that restaurant staff can efficiently update the menu according to changes in offerings and pricing. Additionally, the system shall maintain the integrity of the menu by enforcing unique item names and ensuring accurate pricing inputs.

**3.1.2. Non-Functional Requirements**

Non-functional requirements address the quality attributes of the system:

* Performance: The platform must handle concurrent users and large datasets efficiently.
* Usability: The user interface should be intuitive and accessible on various devices.
* Scalability: The system must scale to accommodate a growing user base.
* Reliability: The platform should be available and reliable with minimal downtime.

## 3.2 Feasibility Analysis

**3.1.2.1. Technical Feasibility**

The only requirement here is adequate knowledge on C++ programming. The project involves only using software and not any form of hardware. Thus, a program that allows C++ programming code to be written is all that is needed.

**3.1.2.2. Operational Feasibility**

The program itself will be very easy to operate. Most of the information will be given on the screen. The user will only be required to choose among the options displayed on the screen. So, the program will be feasible in terms of operation.

**3.1.2.3. Economic feasibility**

* Revenue Model: While this is a beginner's project, consider whether there are opportunities to learn many things but there isn’t any revenue.
* Cost Analysis: This Billing Project is free for all, so anybody can use it as this project is just for the learning experience.

**3.1.2.4. Schedule Feasibility**

This project was finished in under 15 weeks and we have break down the development process into Manageable Phases to make out project efficient and the availability of team members.

Start date:2023/09/01 End date:2023/12/14

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | WEEK | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Research about previous project |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Literal Review |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prepare Proposal & Flowchart |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding will Start |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Will Complete oop Part |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Will Complete File Handling |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding will Completed |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Report will start & complete |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Project will complete |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

***Chart 1: Gantt chart of restaurant billing system***

The tasks are outlined as follows: "Research about the project" spans Week 1, followed by "Literature review" in Week 2. "Prepare proposal and flowchart" is scheduled for Week 3, and "Coding begins" in Week 4. The task "Complete OOP part" extends over Weeks 5 and 6, while "Complete file handling" is anticipated from Weeks 7 to 9. Subsequently, "Complete coding" spans Weeks 10 to 12. Weeks 13 to 14 are allocated for the task "Start and complete the report," and finally, "Complete project" is slated for Week 15.

# CHAPTER FOUR: SYSTEM DESIGN

## 4.1 Algorithm Details

Steps:

1. Start the program

* If the admin is not registered the he/she must enter register page and register themselves as admin entering their username and password.
* The admin needs to login in order to user our system. he/she must be an admin. He/she has to enter the username and password. If the entered data didn’t match, re-entered option will be displayed.

1. Display main menu

* If the person is admin, he will be able to make the necessary changes such as:

1. Insert Items on Menu
2. Edit Items on Menu
3. Delete Items on Menu
4. Display Menu
5. Generate Bills
6. Logout.
7. Insert Items on Menu

* Admin can add Items on menu including item name and price

1. Edit items on menu

* Admin can edit the details of items existing on menu

1. Delete Items on menu

* Admin can delete items existing on menu

1. Display Menu

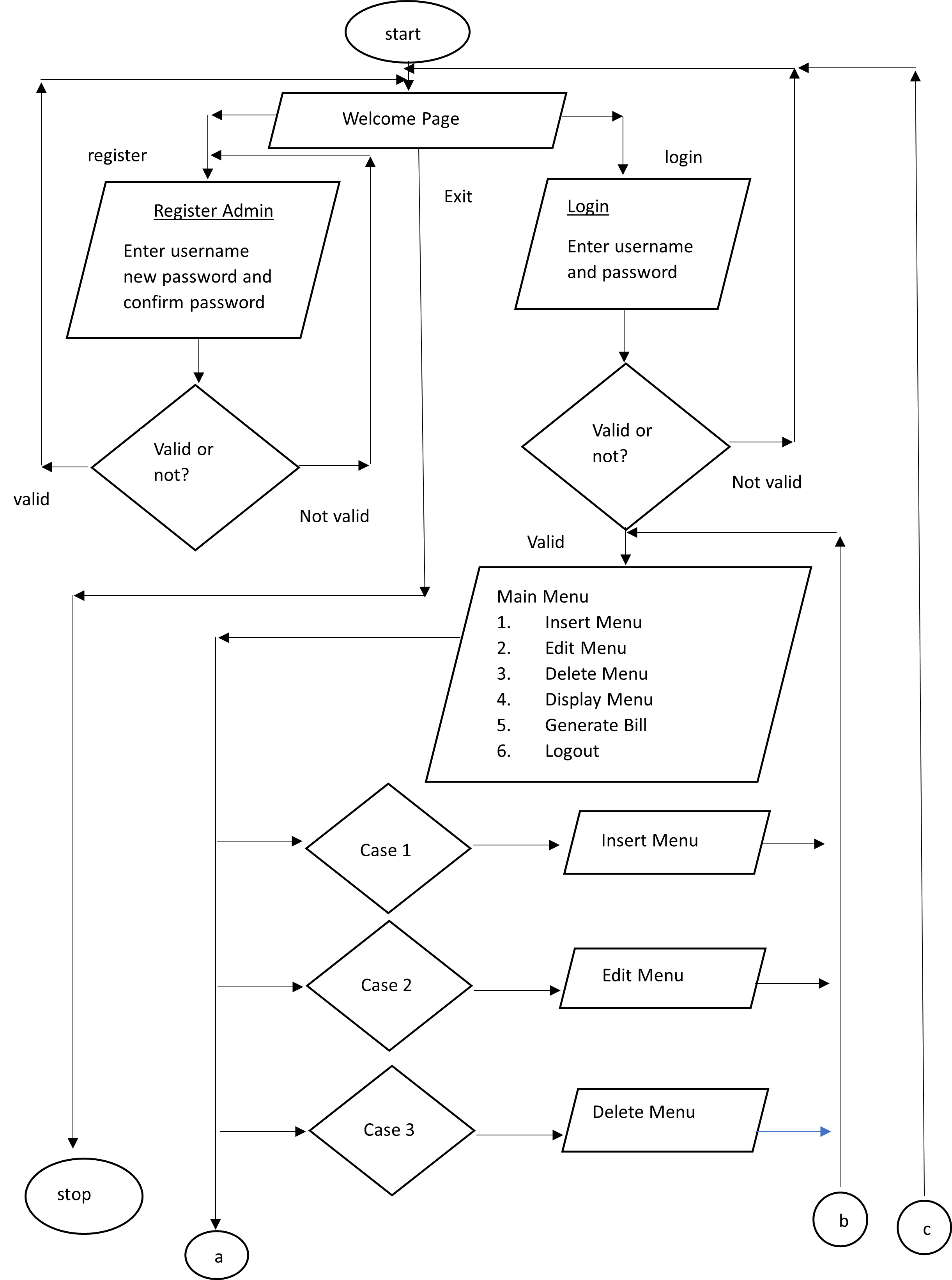
* Admin can view Menu Items

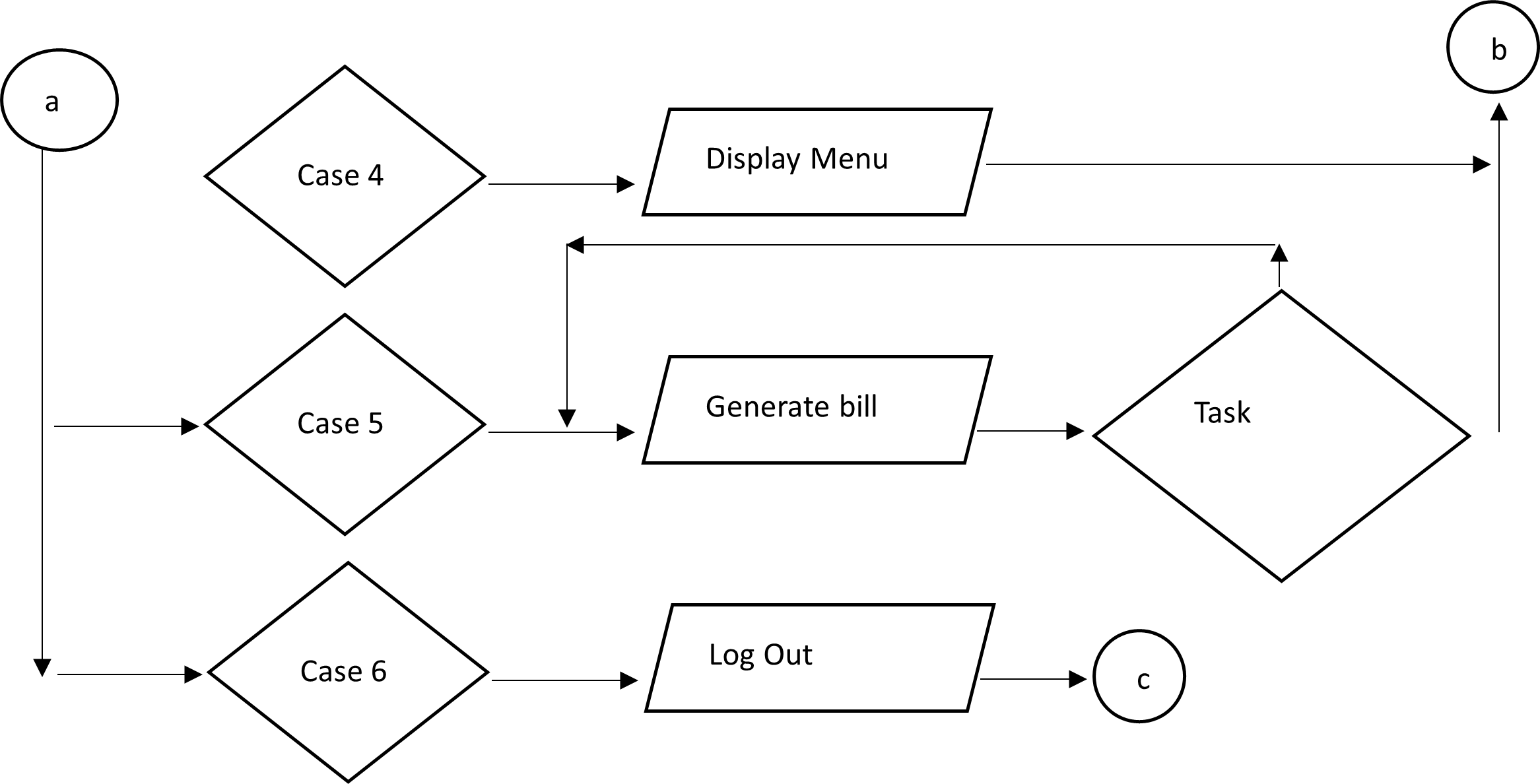
1. Generate Bills

* Admin can generate bills on the basis of what customer has ordered

1. Logout.

## 4.2 Flowchart





Completed?

No

yes

**Figure 1: flowchart diagram of restaurant billing system**

# CHAPTER FIVE: IMPLEMENTATION AND TESTING

## 5.1. Implementation

The implementation of the “Restaurant Billing System” was executed using C++ programming language C++ was chosen for its efficiency, flexibility and suitability for desktop application development. The implementation phase involves the actual development and deployment of the Restaurant Billing System (RBS).

The implemented restaurant billing system efficiently manages menu items, user authentication, and billing processes. It allows users to insert, edit, and delete menu items, ensuring flexibility in menu management. With a robust user authentication mechanism, users can securely register and log in with unique credentials. The billing system accurately calculates subtotals, discounts, VAT, and totals for orders, providing detailed bills with customer names and dates. File handling capabilities ensure the secure storage and retrieval of user credentials and menu items. Through comprehensive testing, the system demonstrates reliability and accuracy in handling various restaurant operations, making it a valuable tool for restaurant management.

## 5.2 Testing

Test cases for Unit Testing:

1. Menu Management
   * Insert several menu items.
   * Edit existing menu items to update their names and prices.
   * Delete menu items.
   * Display the menu to verify changes
2. User Authentication
   * Register new users with unique usernames and passwords.
   * Log in with registered user credentials.
3. Billing System
   * Generate bills for various orders, ensuring correct calculations of subtotals, discounts, VAT, and totals.
   * Test with different combinations of menu items and quantities.
   * Verify that bills are correctly displayed with customer name and date.
4. File Handling
   * Ensure that user credentials are saved and loaded correctly from the login.txt file.
   * Verify that menu items are saved to and loaded from the menu.txt file, preserving their names and prices.

## 5.3 Result Analysis

After thorough testing, the restaurant billing system has demonstrated functionality across all core features including user registration, login, menu management, and bill generation. The user interface is generally intuitive, with password masking enhancing security. While occasional delays were observed during file operations, error handling mechanisms proved robust. Documentation is sufficient for understanding code functionality. Recommendations for improvement include optimizing performance, refining error messages, enhancing the user interface, and exploring additional security measures. Overall, the system meets basic requirements but could benefit from refinement to elevate user experience and ensure optimal performance and security.

# CHAPTER SIX: CONCLUSION AND FUTURE RECOMMENDATION

* 1. **Conclusion**

The restaurant billing system program developed using C++ provides an efficient solution for managing billing operations in a restaurant setting. By using C++'s features such as object-oriented programming, file handling and the program offers functionalities like itemized billing and order management.

This system streamlines the billing process, enhances accuracy, and improves customer service by generating detailed invoices. It also helps in menu management by updating items.

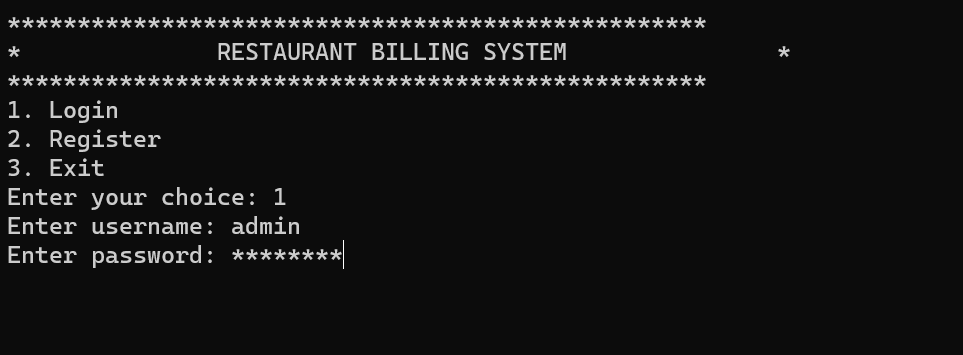
* 1. **Future Enhancement**

Till now we are only able to make CUI based but it can be further enhanced by implementing better GUI implementation. In future we will make it easier to use by improving how it looks and adding buttons and menus. We will add the capabilities in the program like storing data more efficiently and by using a database system like SQLite or MySQL. Allowing customers to order and pay online, making it more convenient for them and creating a mobile app version so staff can use it on their phones, making it easier to manage orders.

# REFERENCE

1. <https://www.academia.edu/35899951/DESIGN_AND_IMPLEMENTATION_OF_COMPUTERIZED_RESTAURANT_MENU_SYSTEM_A_CASE_STUDY_OF_AROMA_RESAURANT_BAUCHI>
2. [https://ieeexplore.ieee.org/abstract/document/9079304?signout=successI](%20https://ieeexplore.ieee.org/abstract/document/9079304?signout=successI)
3. <https://www.studymode.com/essays/Billing_management_system-c-489854.html>
4. Object-Oriented Programming with C++, By E Balagurusamy

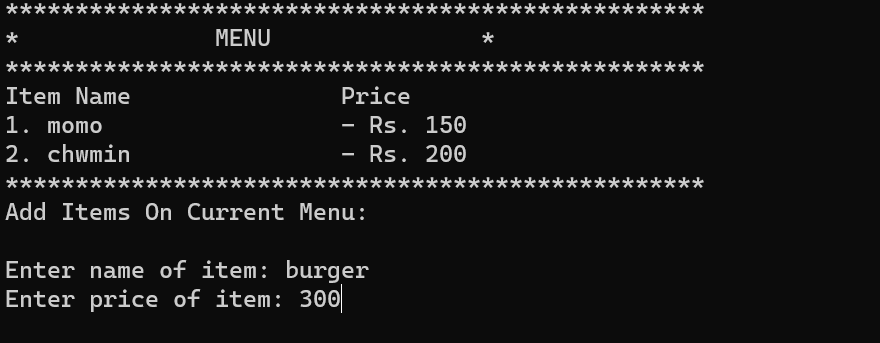
# APPENDICES

****

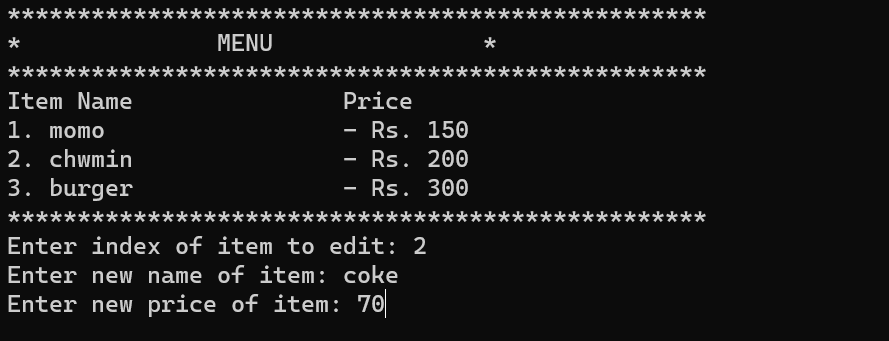
**Login page**



**Main page**

****

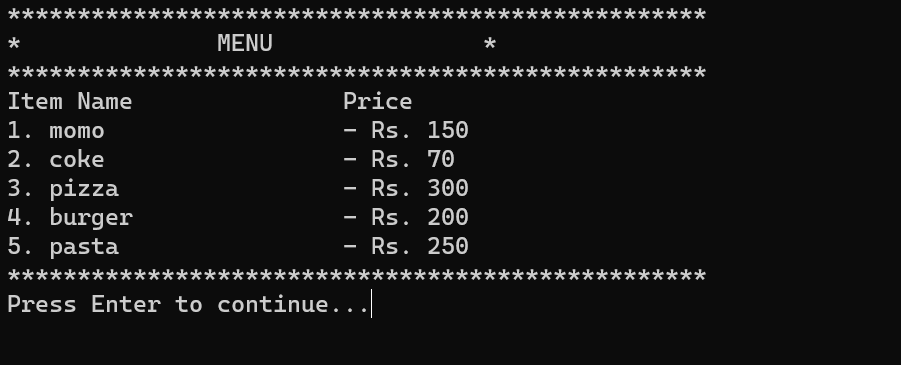
**Insert menu page**

****

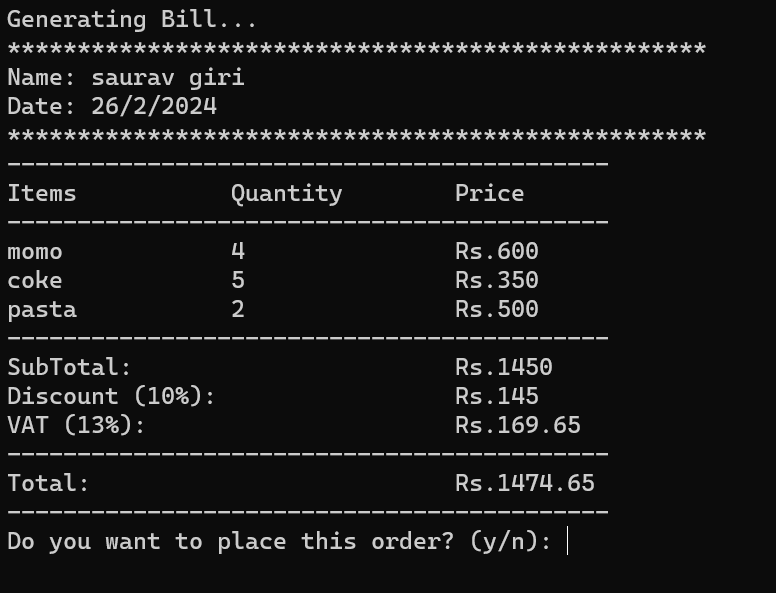
**Edit page**

****

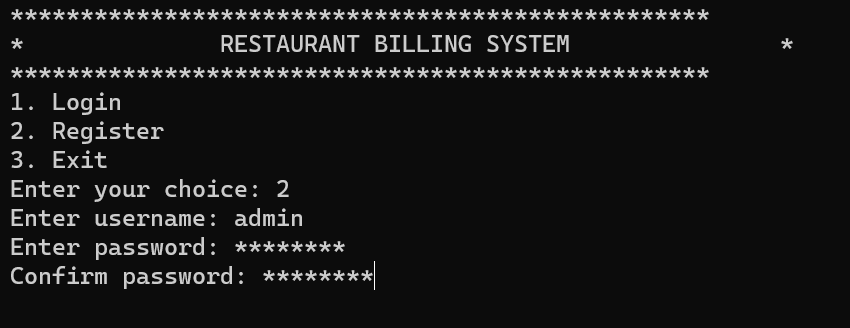
**Delete page**

****

**Display menu page**

****

**Generate bill page**

****

**Register page**