A
Project Report On

<u>Cafe Management</u>

Submitted By:

Bhanderi Kavan &
Mathiya Kano

BCA Semester–5

Project Guide:

Prof. Harsh Joshi

Submitted To:

Geetanjali College Of
Computer Science And Commerce (B.B.A).
Rajkot.

Academic Year: 2024-2025

<u>Acknowledgement</u>

We Are Happy To Submit Our Idea Of " Cafe Management " System To Saurashtra University, Rajkot For BCA Course In Computer Branch.

We Are Also Grateful To Prof. Brijesh Shah, The Head of The Department And All The Faculty Members Of The Department Of Computer Science For Their Kind Support Through Out This Journey.

We Take The Privilege To Acknowledge The Elite Authors Of Numerous Books And Papers And Blogs Which We Have Referred During Progress Of The Project.

The Feeling Of Gratefulness To Any One's Help Directly Arises From The Bottom Of Our Heart. A Small But An Important And Timely Help Can Prove To Be A Milestone In One's Life.

Index

Project Profile

- Project Title   :   Cafe Management

- Development Software   :   Visual Studio 2022

- Front End   :   C#

- Backend   :   SQL

- Academic Year   :   2024-2025

- Developed By   :   Bhanderi Kavan &
                                                                     Mathiya Kano

- Submitted To   :   Geetanjali College

- Documentation Tool   :   Microsoft Word

- Operating System   :   Windows

- Language   :   C#

System Development Life Cycles

For The Development Of This Project We Have Followed The Simple Waterfall Model Of SDLC.

The Waterfall Model Was The First Process Model To Be Introduced. It Is Also Referred To As A Linear-Sequential Life Cycle Model. It Is Very Simple To Understand And Use. In A Waterfall Model, Each Phase Must Be Completed Before The Next Phase Can Begin And There Is No Overlapping In The Phases.

The Waterfall Model Is The Earliest SDLC Approach That Was Used For Software Development.

The Waterfall Model Illustrates The Software Development Process In A Linear Sequential Flow. This Means That Any Phase In The Development Process Begins Only If The Previous Phase Is Complete. In This Waterfall Model, The Phases Do Not Overlap.

Steps   :   1   -   Requirement gathering
                          2   -   Project Planning
                          3   -   Design
                          4   -   Coding & Implementation
                          5   -   Testing
                          6   -   Implementation
                          7   -   Maintenance

1 - Requirement Gathering

Any Software Development Process Must Include The Requirement Gathering Stage. After Choosing The Project Topic, One Must Research Every Criteria Needed To Construct That Specific Project.

We began gathering requirements for the Café Management Software in the C# Windows application by analyzing various restaurant and café management tools available on multiple platforms. To gain insights into the essential features for running a café smoothly, we reviewed different software applications, paying close attention to user interface design, customer handling, and operational workflows. This approach helped us ensure that our system would meet the specific needs and requirements of café management in a Windows environment.

Features that are needed in the Café Management Software for the admin are as follows:

Add Item Page:

Admin can add new menu items to the system, including details like item name, price, and category. This feature allows café managers to update the menu quickly and efficiently when new items are introduced.

Update Item Page:

Admin can modify existing menu items by updating details such as price, availability, or description. This feature is essential for making real-time changes to the menu, ensuring that the system always reflects current offerings.

Remove Item Page:

Admin can remove items from the menu that are no longer available or needed. This helps in maintaining a clean and accurate menu list for the café, ensuring customers are only viewing available options.

Place Order Page:

Admin can place orders by selecting items from the menu, specifying quantity, and adding any special instructions. This page allows efficient order processing for both customers and staff, ensuring seamless service in the café environment.

2   -   Project Planning

After the requirement collection phase has been completed, the next step is to create a project plan. This involves determining the necessary project modules and selecting the most suitable technologies for development.

For the Café Management Software, selecting the right technologies means choosing both frontend and backend frameworks that work best for an offline, Windows-based system.

In this case, **Windows Forms (C#)** is used to develop the user interface of the application. It is a stable and widely-used technology for building desktop applications, and our team's experience with C# makes it an ideal choice for this project.

For the backend, **SQL Server** is utilized to store all the necessary data, including menu items, orders, and customer information. Since it is a lightweight, serverless database, it perfectly suits an offline application, ensuring data can be managed locally without the need for an internet connection. This combination of C# and SQL Server provides a reliable and efficient solution for the café's operational needs.

3   -   Design

The design process for the Café Management Software can be divided into two key phases: **Preliminary Design** and **Final Design**.

During the **Preliminary Design** phase, the basic layout of the software is created either on paper or using a design tool. This includes drafting the main screens such as the menu management, order placement, and customer management pages. The development team then reviews this initial design to evaluate its feasibility in terms of functionality and user experience.

The **Final Design** phase follows, where the project's final or nearly accurate design is developed. This phase incorporates feedback from the preliminary design, addressing any identified issues or impractical elements. The final design reflects a more detailed and refined version, ready for actual development using **Windows Forms (C#)** for the user interface and **SQL Server** for the backend, ensuring the software meets the café's operational requirements effectively.

4   -   Coding & Implementation

After gathering all requirements, obtaining customer approval of the design, and assessing the project's feasibility, the actual application for the Café Management Software is coded in the selected programming language.

This phase is considered the longest in the Software Development Life Cycle (SDLC). To complete this stage, various tools such as integrated development environments (IDEs), browsers, and database tools are required. For development, we chose **Visual Studio** as our primary IDE for coding the application in **C#**.

To Avoid Confusion Throughout The Coding Phase, We Also Adhere To The Following Coding Standards:

1) Variable Naming:

Descriptive and Contextual Names: Variable names should clearly describe their purpose and context. For example, variables associated with UI controls (like text boxes or labels) should include the control type in the name, such as txtCustomerName, lblTotalPrice, or btnSubmitOrder.

Consistency: Follow camelCase for local variables and method parameters, and PascalCase for class names, properties, and method names.

2) Control Naming Conventions:

All UI controls (buttons, labels, textboxes, etc.) should use meaningful names with prefixes based on the control type:

- txt for TextBox
- lbl for Label
- btn for Button
- cmb for ComboBox
- chk for CheckBox
- dgv for DataGridView

3) Method Naming:

Methods should be named using PascalCase and describe the action they perform.

Methods that retrieve data should start with the prefix `Get`, while action-oriented methods should use verbs like `Add`, `Update`, `Delete`, or `Calculate`.

4) Event Handler Naming:

Event handler methods should follow the format ControlName_EventName. This clearly identifies which control the event handler is associated with and what event it responds to.

5) Array and List Naming:

Arrays or collections should include the suffix Array or List to clearly denote their structure.

6) File and Folder Structure:

Each class should reside in its own file, and file names should match the class name (e.g., Order.cs for the Order class).

Organize files into folders based on their functionality (e.g., "Forms" for Windows Forms, "Models" for data models, and "Services" for business logic).

7) DRY Principle (Don't Repeat Yourself):

Avoid code duplication by creating reusable methods for common tasks. This makes the code cleaner and easier to maintain.

8) Form Initialization and Loading:

Ensure proper initialization of components in the Form_Load event, including setting default values and loading any necessary data (e.g., from the database) when the form opens.

The Application Development Divided To Two Phases, Which Are:

1. Database Making :

The database is designed based on the application's requirements and the café's operational needs. For this offline system, we use **SQL Server**, which is a lightweight and serverless database suitable for local storage in desktop applications. The database stores information about menu items, orders, customers, and other essential data needed for the smooth operation of the café. SQL Server allows the application to run without requiring an internet connection, making it ideal for offline use in a Windows environment.

2. Application Development :

In this phase, the design is translated into actual development using **C# Windows Forms**. The user interface is built using Windows Forms to ensure a smooth and intuitive experience for the café staff. Core functionalities such as menu management, order processing, and customer management are implemented based on the design specifications. Integration with the SQL Server database allows the application to retrieve, manipulate, and store data locally, ensuring efficient management of café operations.

5 - Testing

Once the coding and implementation phases of the Café Management Software are complete, the application is ready to be tested for usability and to ensure it operates as planned.

For testing purposes, the software is shared with café staff, managers, and technical advisors to verify its functionality, efficiency, and user- friendliness. They provide feedback on the system's ability to manage menu items, process orders, and handle customer interactions effectively.

If any issues are identified, or if the application does not function as intended, the Software Development Life Cycle (SDLC) process is repeated, starting from the first step (requirements analysis) to the last, until the system meets all operational and user expectations.

6 - **Implementation :**

This phase is initiated after the system has been tested and accepted by theuser. In this phase, the system is installed to support the intended businessfunctions. System performance is compared to performance objectives established during the planning phase.

7 - **Maintenance :**

The maintenance phase of the SDLC is where the software is monitored to ensure it continues to function as it was designed to, and repairs or upgrades are performed needed.

System Requirement Specifications

To Develop This Project, The Following System Hardware And Network Are Required:

Minimum Hardware Requirement: For

Windows Application:

Operating System	Windows 10 or Windows 11
CPU/Processor	A minimum of 1.8 GHz or faster processor.
Ram	Minimum of 4 GB of RAM

About The Tools & Technologies

C# :

        C# (C-Sharp) is a modern, object-oriented programming language developed by **Microsoft** as part of its **.NET** platform. Introduced in the early 2000s, C# has evolved into one of the most widely used languages for developing a variety of applications, including web, desktop, mobile, and cloud-based systems.

        C# can be used to build applications for a wide range of devices, including mobile devices, desktop and laptop computers, servers, and the cloud. C# is especially popular for building web-based applications because it can create interactive user environments.

        C# is designed to be simple, modern, and support software engineering principles like strong type checking, automatic garbage collection, and array bounds checking. It also incorporates features from other paradigms, such as functional programming.

        C# is a powerful, versatile, and easy-to-learn programming language, suitable for a wide range of application types. Its rich feature set, strong ecosystem, and support for modern development practices make it an excellent choice for developers of all experience levels. Whether you are building desktop applications, web services, or even games, C# provides the tools and framework to succeed.

Visual Studio :

        Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop GUI (Grapical User Interface), console , web apps , web applications , mobile apps cloud , and webservices , etc.

        With the help of IDE , You can create managed code as well as native code.

        It uses various platforms of microsoft software development softwarelike Windows store , microsoft silverlight , and windows API , etc.

        It is not language specific IDE as you can use this to write code in C# , C++ , VB(Visual Basic) , Python , JavaScript , and many more languages.

SQL Server Database :

        SQL Server is a relational database management system (RDBMS) developed by Microsoft.

        It is designed to store, retrieve, and manage data efficiently and is widely used in

enterprise-level applications, web applications, and data warehousing solutions.

SQL Server supports a wide range of applications and is known for its robust performance, scalability, and security features.
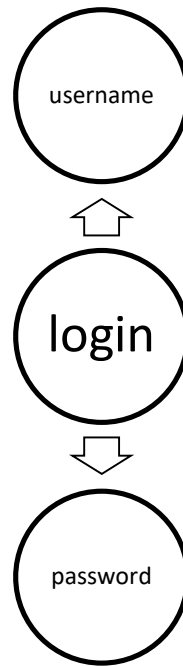
Microsoft Word :

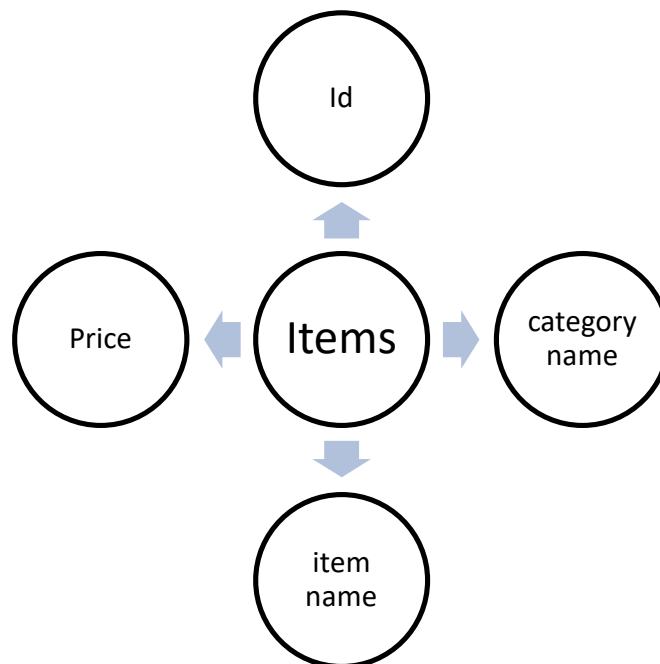To make the documentation we have used Microsoft word. It has

features like formatting data , pictures , tables , charts , mailings , etc.
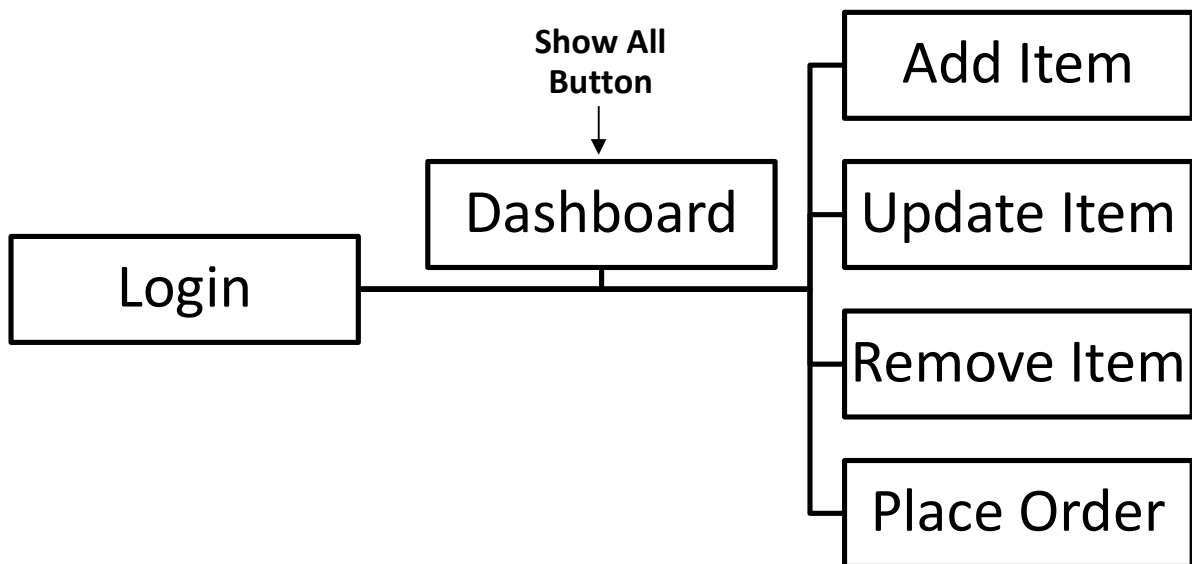
Entity Relationship Diagram

1) Login :

```
        ┌──────────┐
        │ username │
        └──────────┘
             ⇧
        ┌──────────┐
        │  login   │
        └──────────┘
             ⇩
        ┌──────────┐
        │ password │
        └──────────┘
```

2) Items :

```
                    ┌──────┐
                    │  Id  │
                    └──────┘
                       ⬆
  ┌───────┐        ┌───────┐        ┌──────────┐
  │ Price │ ◀────  │ Items │  ────▶ │ category │
  └───────┘        └───────┘        │   name   │
                       ⬇            └──────────┘
                   ┌────────┐
                   │  item  │
                   │  name  │
                   └────────┘
```

11

Data Flow Diagram

```
                                    Show All
                                    Button
                                       |
                                       v
                  +--------------+
                  |  Dashboard   |------+------+--------------+
   +-----------+  +--------------+      |      |  Add Item    |
   |   Login   |---------|               |      +--------------+
   +-----------+                         |      +--------------+
                                         +------| Update Item  |
                                         |      +--------------+
                                         |      +--------------+
                                         +------| Remove Item  |
                                         |      +--------------+
                                         |      +--------------+
                                         +------| Place Order  |
                                                +--------------+
```

<u>Data Dictionary</u>

Login Table:

Sr no	Field	Data Type	Size	Constraint
1	username	String	-	-
2	password	String	-	-

Items Table:

Sr no	Field	Data Type	Size	Constraint
1	Id	String	-	Primary Key
2	Category	String	-	-
3	Item Name	String	-	-
4	Price	int	-	-

Screen Shots

1. **Login Page**
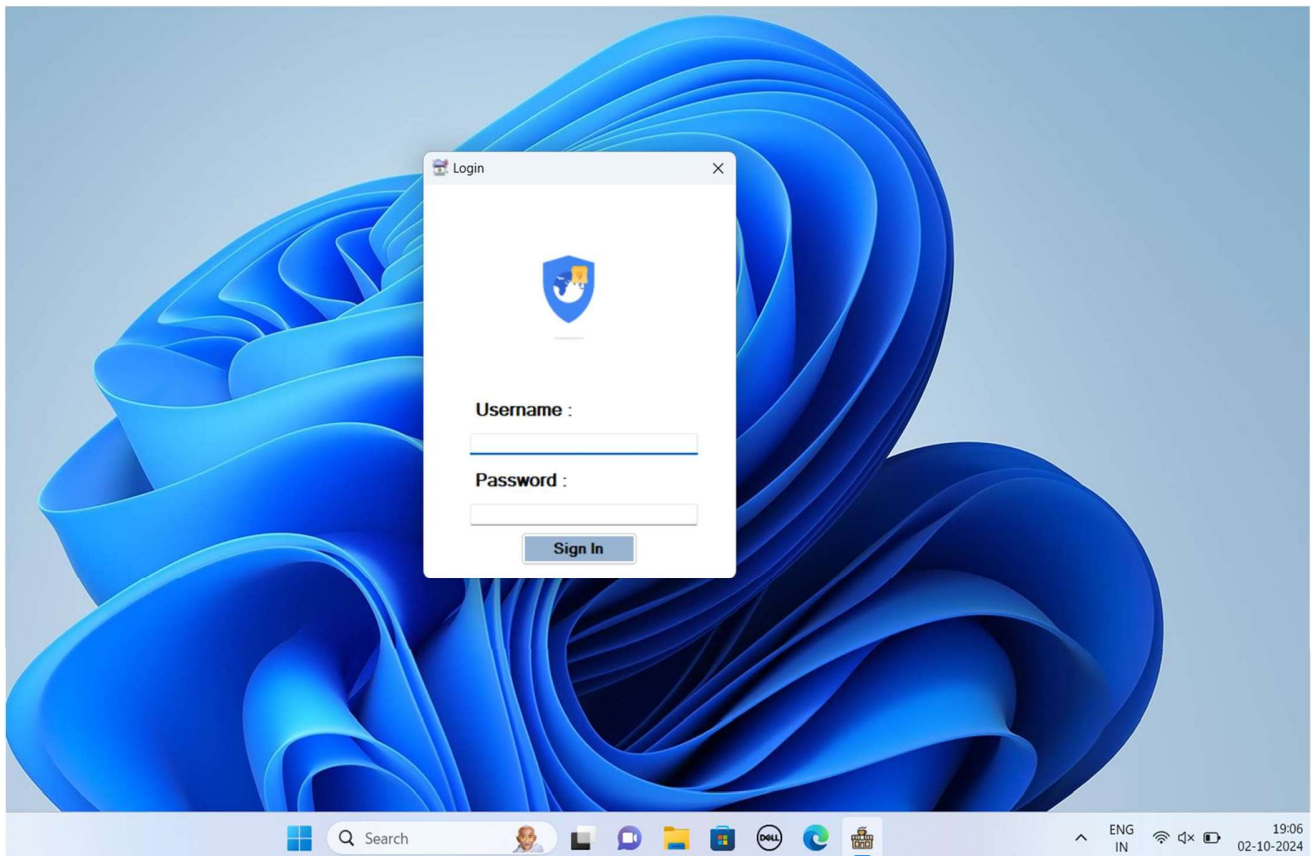
- **Login Page Description:**

The login page is the first screen of the application.

After successful login, the user is directed to the dashboard page.

If the login is unsuccessful, an error message is displayed, prompting the user to re-enter their credentials.

2. Dashboard Page
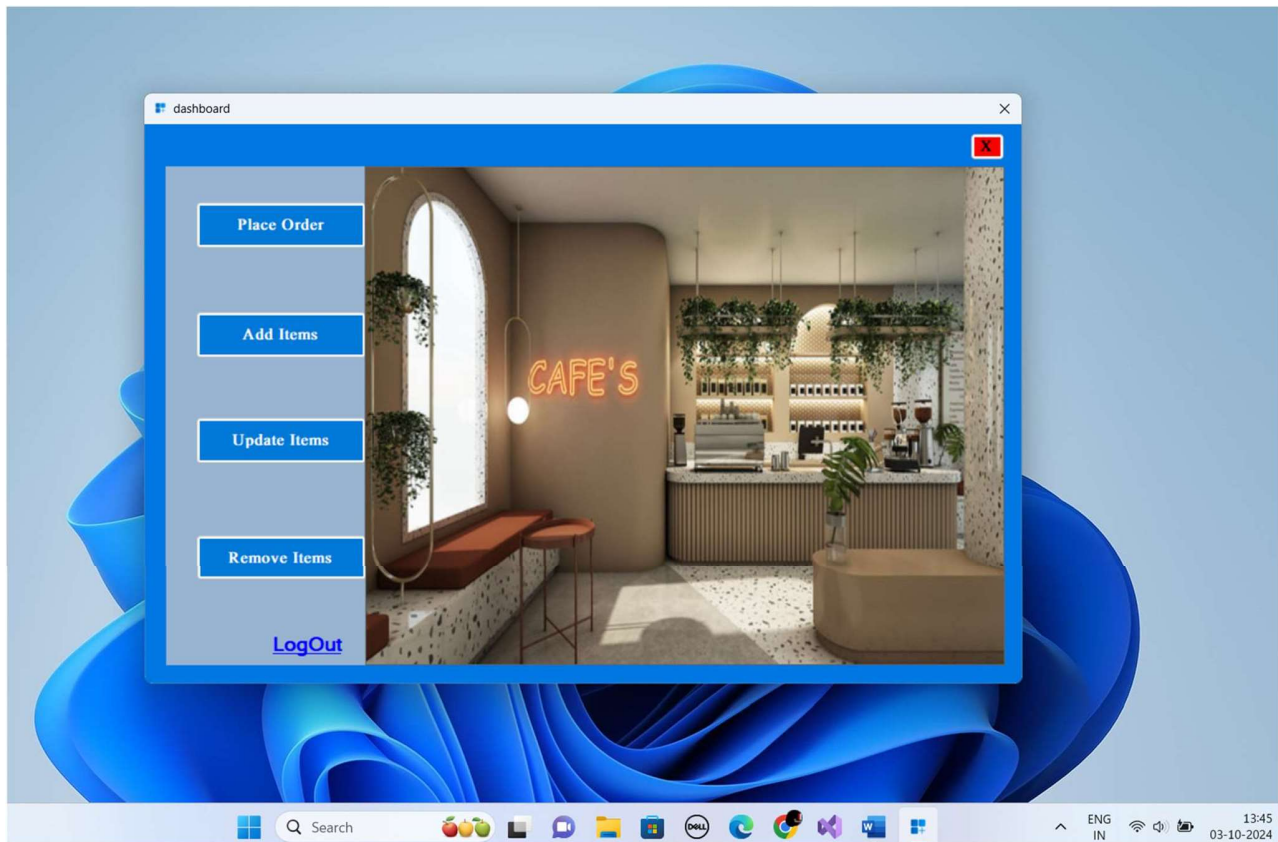
- **Dashboard Page Description:**

Add Item : users to add new items to the menu by entering the item details such as name, price, and category.

Update Item : Provides functionality to modify existing food items, including changes to the item's price, description, or availability.

Remove Item : Enables users to remove food items from the menu, ensuring the list is always up to date.

Place Order : The staff takes the order from the customer and then the item list is given to the admin operator and then the bill is generated.

3. Add Item Page
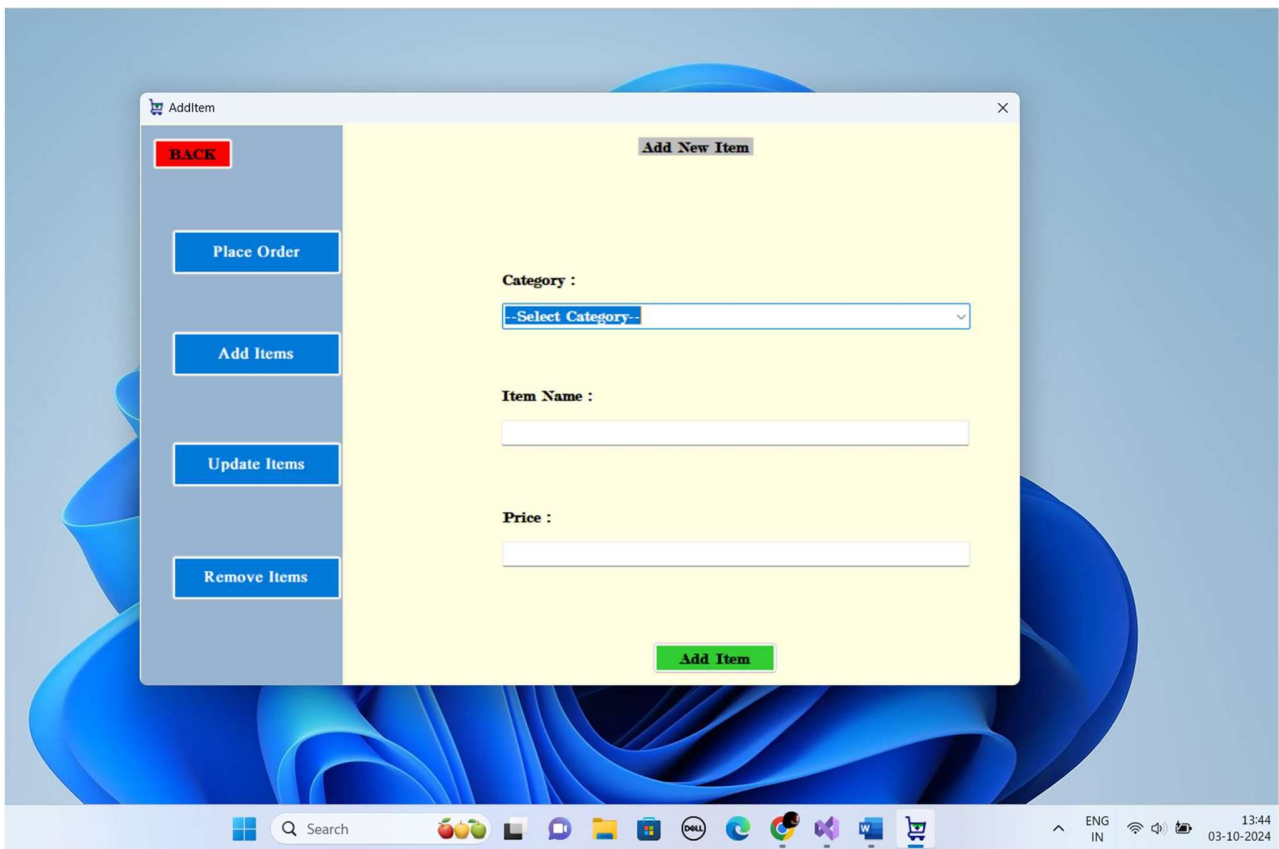
- **Add Item Page Description:**

  The Add Item page is where users can create and add new menu items to the food zone.

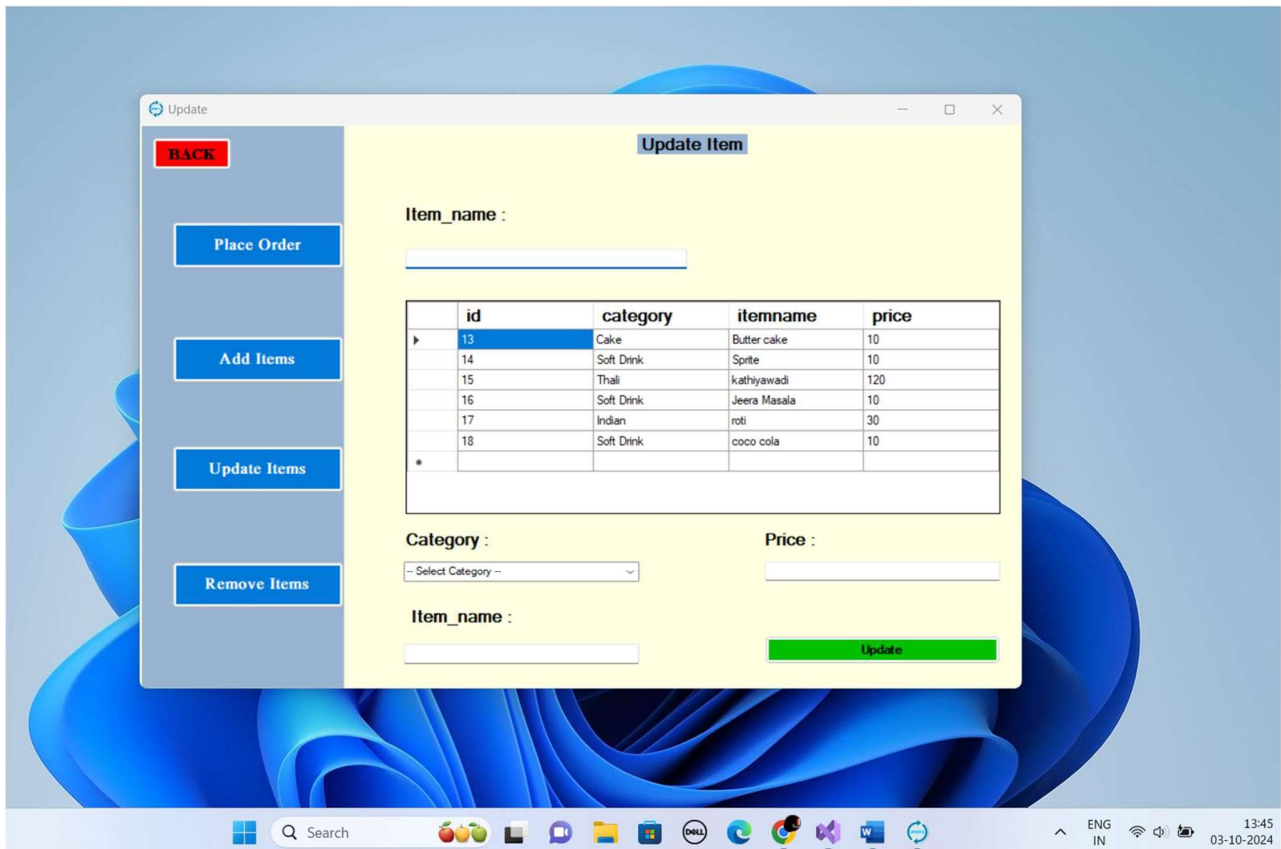 It features input fields to enter important details for each new item:

  Item Name: The name of the food item.

  Category: Select the category (e.g., appetizers, main course, desserts) the item belongs to.

  Price: Set the price for the item.

  A **Add Item** button ensures the item is successfully added to the list, and users can return to the continue adding more items.

4. Update Item Page

- **Update Item Page Description:**

  The **Update Item** page allows users to modify the details of existing menu items in the food zone.

 Users can search for or select an item from the menu to update its details.

   The page includes editable fields for :

   Item Name: Modify the name of the food item if needed. **Category:** Update the category to which the item belongs (e.g., appetizers, main course, desserts).
   Price: Change the price of the item.

  After making changes, users can click the **Update** button to apply updates, which will reflect in the system immediately (or sync when online if offline mode is enabled).

5. Remove Item Page
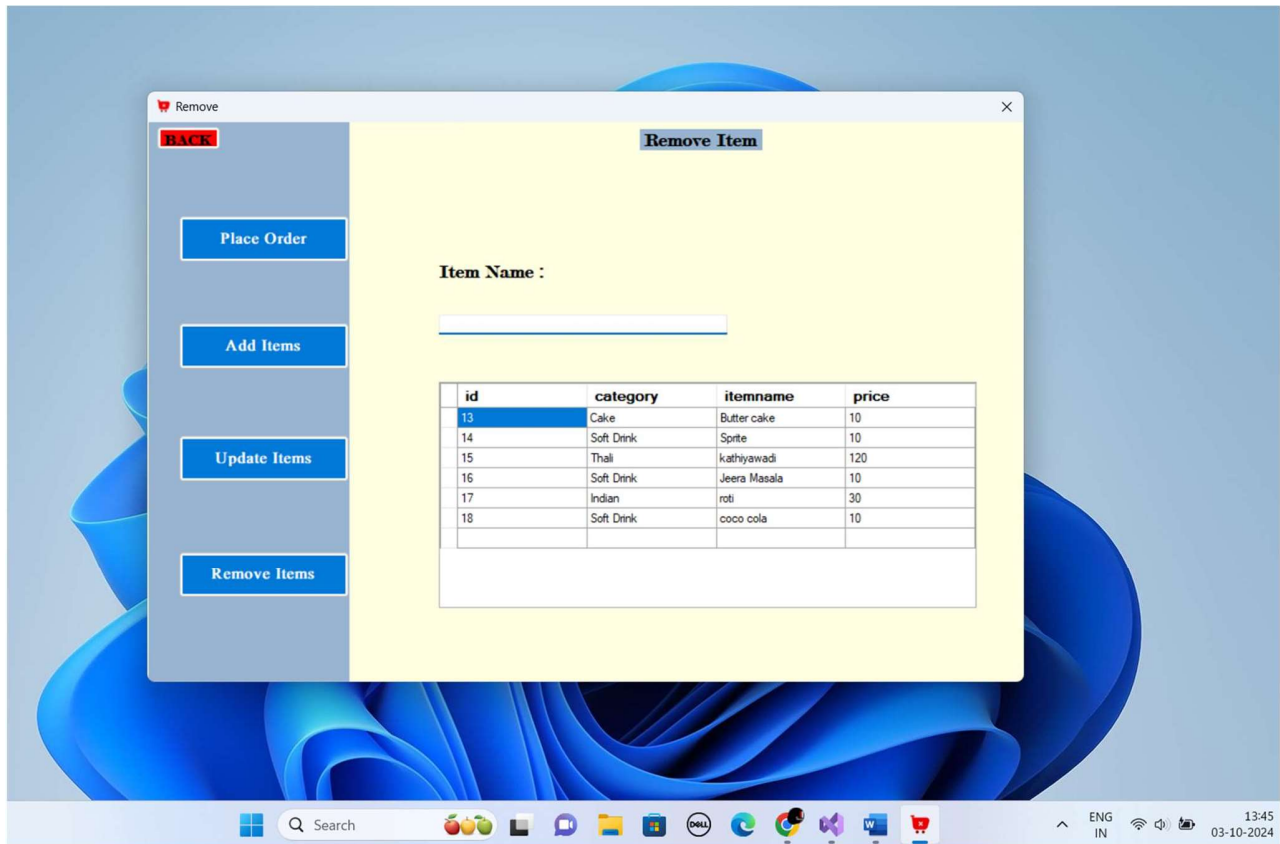
- **Remove Item Feature Description:**

  The Remove Item feature allows users to delete a selected menu item from the food zone.

  To remove an item, users simply select the desired row/item from the list.

  A **Remove** button is provided, which, when clicked, prompts the user to confirm the action to prevent accidental deletions.

  Upon confirmation, the selected item is permanently removed from the menu and will no longer be available for order.

6. Place Order Page

- **Place Order Feature Description:**

  The Place Order feature facilitates the process of taking and processing customer orders within the food zone.

 Staff members take orders directly from customers, noting their selections and any special requests.

  Once the order is taken, the staff compiles a list of items selected by the customer.

  This item list is then submitted to the admin operator for processing.

  The admin operator reviews the order and generates a bill based on the selected items and their prices.

  A detailed bill is prepared, outlining each item, quantity, and total cost, which is then presented to the customer for payment.

7. Print Bill Page
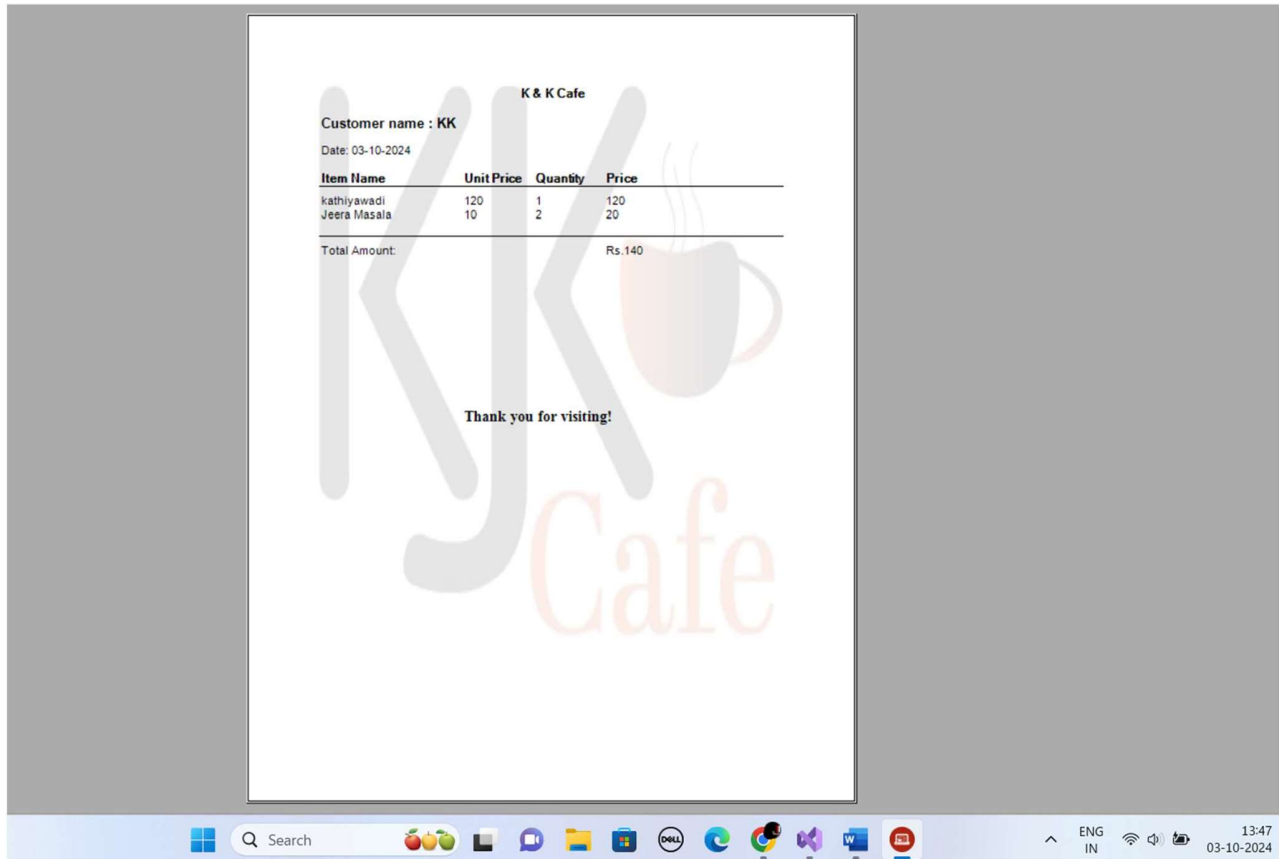
- **Print Bill Page Description:**

The Print Bill page is designed to display the finalized bill for customer orders, providing a clear and organized summary of all charges.

The page includes the following key components:

- Header: Displays the restaurant name and date of the transaction.

- Customer Details: Contains fields for customer information, including name.

- Order Summary: Lists all items ordered by the customer, including:

- Item Name: The name of each food item.
- Quantity: The number of each item ordered.
- Price: The price of each item.
- Subtotal: The total cost for each item based on quantity.

- Total Amount: Displays the total bill amount, including any applicable taxes and discounts.

- Thank You Note: A friendly message thanking the customer for their order and encouraging them to return.

Users can choose to print the bill directly from the page using a **Print** button, which sends the bill to the connected printer.

The page may also include options to email the bill to the customer or save it as a PDF for record-keeping.

Test Cases

Test Case	Description	Actual Output	Test Result
Login Screen	successful login, the user is directed to the dashboard page.	successful login, the user is directed to the dashboard page.	**Pass**
Dashboard Screen	Display All Button. Picture	Display All Button. Picture	**Pass**
Add Item Screen	Display All Food Items Of Each Category Of Available In Food Zone.	Display All Food Items Of Each Category Of Available In Food Zone.	**Pass**
Update Item Screen	Modify the details of existing menu items in the food zone.	Modify the details of existing menu items in the food zone.	**Pass**
Remove Item Screen	Users simply select the desired row/item from the list.	Users simply select the desired row/item from the list.	**Pass**
Place Order	Generates a bill based on the selected items and their prices.	Generates a bill based on the selected items and their prices.	**Pass**
Print Page	print the bill directly from the page using a Print button, which sends the bill to the connected printer.	print the bill directly from the page using a Print button, which sends the bill to the connected printer.	**Pass**

<u>Limitation</u>

The design may not be fully responsive on all devices, potentially affecting usability on smaller screens.

The layout might require optimization for accessibility, as users with disabilities may encounter challenges.

The system may not allow for bulk updates or removals of items, requiring manual entry for each action.

Adding new features or functionalities may require significant redevelopment efforts.

<u>Future Enhancement</u>

Add functionality for bulk updates and removals of items, allowing admins to manage large inventories more efficiently.

Implement a loyalty program that rewards repeat customers with discounts, special offers, or points redeemable for future purchases.

Allow customers to track their loyalty points and rewards through the point.

Implement real-time data synchronization to ensure that all changes (like order updates or item modifications) are reflected instantly, reducing the chances of data inconsistencies.

<u>Webliography</u>

- ✓ https://stackoverflow.com/
- ✓ https://www.w3schools.com/
- ✓ https://icon-icons.com/
- ✓ https://www.geeksforgeeks.org/