# CARR Online Food Ordering and Delivery Software

Abhigya Shrestha, Crystal Raj Khadka, Rajil Shrestha, Rushmit Karki

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST4008CEM: Computing Activity Led Learning Project 1

Module-Leader: Giriraj Rawat

July 28, 2022

# Table of Contents

CARR Online Food Ordering and Delivery Software 6

Introduction 6

Aim 8

Objectives 8

Problem Statement 9

Features 10

Functional Requirements 11

Non-functional Requirements 12

Scope 13

Development Methodology 14

Methodology 14

Tools and Technology 16

Conceptual Diagram 17

System Architecture 18

Project Plan 19

Prototype 21

Developed System 23

Login page 23

Reset password 25

Sign up page 27

Homepage 29

Menu Page 30

Cart page 32

Billing System 34

Contact 36

Inventory page 38

Database Table 40

Account page 43

System Testing 44

Version Control 49

Conclusion 50

References 51

# Table of Figures

[Figure 1 7](#_Toc109935913)

[Figure 2 9](#_Toc109935914)

[Figure 3 10](#_Toc109935915)

[Figure 4 13](#_Toc109935916)

[Figure 5 14](#_Toc109935917)

[Figure 6 16](#_Toc109935918)

[Figure 7 17](#_Toc109935919)

[Figure 8 19](#_Toc109935920)

[Figure 9 19](#_Toc109935921)

[Figure 10 21](#_Toc109935922)

[Figure 11 21](#_Toc109935923)

[Figure 12 22](#_Toc109935924)

[Figure 13 22](#_Toc109935925)

[Figure 14 23](#_Toc109935926)

[Figure 15 24](#_Toc109935927)

[Figure 16 25](#_Toc109935928)

[Figure 17 26](#_Toc109935929)

[Figure 18 27](#_Toc109935930)

[Figure 19 28](#_Toc109935931)

[Figure 20 29](#_Toc109935932)

[Figure 21 30](#_Toc109935933)

[Figure 22 31](#_Toc109935934)

[Figure 23 32](#_Toc109935935)

[Figure 24 33](#_Toc109935936)

[Figure 25 34](#_Toc109935937)

[Figure 26 35](#_Toc109935938)

[Figure 27 36](#_Toc109935939)

[Figure 28 37](#_Toc109935940)

[Figure 29 38](#_Toc109935941)

[Figure 30 39](#_Toc109935942)

[Figure 31 39](#_Toc109935943)

[Figure 32 40](#_Toc109935944)

[Figure 33 41](#_Toc109935945)

[Figure 34 42](#_Toc109935946)

[Figure 35 42](#_Toc109935947)

[Figure 36 43](#_Toc109935948)

[Figure 37 43](#_Toc109935949)

[Figure 38 44](#_Toc109935950)

# 

# CARR Online Food Ordering and Delivery Software

# Introduction

‘CARR Online Food Ordering and Delivery Software’ is a service in which customers can order food from a restaurant, bakery, or an independent food delivery company online without visiting them personally. Customers can order from a website, mobile app, or desktop app.

This project provides a platform to order foods from several restaurants of the customer’s choice. The member restaurants can register themselves on this software and update their products for the customer to choose from. The customers can browse the food products in the restaurants of their choice. They can also register themselves if they wish to be a regular customer or continue as a guest. There are options to choose different types of payment modes and deliveries. The orders are then notified to the respective restaurants for the process. The customers can track their order process by signing up to a new account or logging in. This application has other features like purchase-history, rating, categories, Search-bar etc. Figure 1 provides an example of online food delivery.

### Figure 1

Food delivery system [(PosterMyWall, n.d.)](#c).



# Aim

Making overall food delivery experience much easier for the customer to order a food or get food in short period of time without personal visit.

# Objectives

* Easy to use and simple to understand
* Provides personal customer dashboard
* Choice of payment online or after delivery
* Choice of delivery
* Display the food for the customer to choose
* Ensure the security of data provided by the customers
* Can be used in daily basis to order a food as they like
* Simplify the food ordering method
* Provides offers like discount, coupons
* Aware the people about the technology
* Reduce the use of cash to save the papers

# Problem Statement

In the past, customers needed to pay a personal visit to restaurants to order food. The order was normally taken by pen and paper, which was a slow process and hence the delivery of food was also slower. The recording of the bills was also major problem because they needed to be performed manually by staffs. It was difficult to keep track of customer’s order due to work overload. This could cause the scenarios of missing orders to be delivered.

These problems of traditional ordering system can be easily solved through online food delivery system in today’s world of technology where internet is easily available. People can just order their food using a single smartphone and don’t need to visit the restaurant personally. They can order from different restaurants with a single app. Digital order makes the process fast, so restaurants can prepare the food faster and deliver them sooner as well. These keep up with the customer’s expectation. Use of computer also solves the problem of billing since everything is done digitally, hence minimizing the errors in manual calculation. These problem statements are shown in figure 2.

### Figure 2

Problem solving ([Fixlastmile, 2021)](#a)



# Features

* Customers can create account so that they order food on short period of time.
* Many customers can access the food delivery apps with their own accounts concurrently.
* Customer details are secured and cannot be misused.
* IT is easy to use and simple to understand
* In the home menu customer can click the menu button and view the details of the food.
* Customers can select food from the menu and add to cart.
* Customer can choose their method of delivery.
* Customers can pay the bill by the online system or after delivery with cash.
* Customers can change the password and other credentials as they want.
* Customers can log-out their account as they want.
* Offers like discount and coupons ae regularly provided.

### Figure 3

Features of Food ordering and Delivery apps [(Gupta, 2021)](#b).



# Functional Requirements

* **Check the menu:** Food menu can be checked by the customer whenever they want.
* **Edit and update stored information:** Customer can update their account information.
* **Registration:** Customer can register their own account. Restaurants can also register themselves.
* **Login and logout:** Customer can login when correct credentials are entered and customer can logout anytime they want.
* **Reset password:** Customers can reset their password as they want when they forget their password.
* **Add to cart**: Customer can add items to their cart as they want.
* **Payment**: Customer also can pay the bill through online or after delivery.

# Non-functional Requirements

* **Simplicity**:

The UI of these project is easy to learn and user friendly. The elements of the GUI are self-descriptive.

* **Cross platform application:**

This software can run in all platforms and all kinds of devices connected to the internet.

* **Efficiency:**

The application runs quickly with maximum lagging time of 15 seconds.

* **Security:**

The application secures the details of the user by logging off automatically if the application is idle for 2 minutes.

# Scope

Figure 4 explains the main scope of online food ordering system.

### Figure 4

Scope of online food ordering system



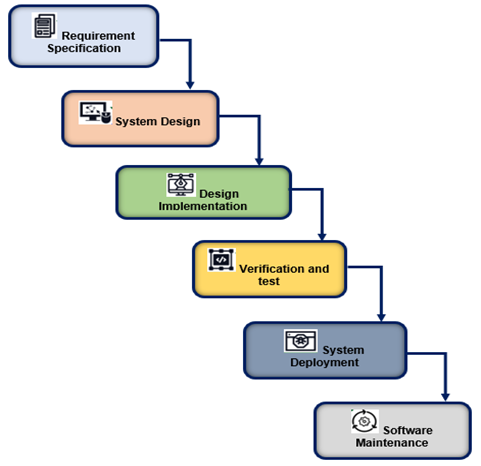
# Development Methodology

There are lot of development methodology in software engineering. For this project, we have used modern waterfall model as a development method as this method is easy to follow, understand and implement. In this method, new phase starts only after the completion of the previous phase. So, phases do not overlap with each other which creates less complexity in a project.

## Methodology

### Figure 5

Steps of modern waterfall model



Modern waterfall model is a process with a linear approach. In this method, one phase needs to be completed so another phase can start. Figure 5 presents the various steps in this method. Initially, a feasibility analysis is conducted followed by documentation of the requirement specification also known as SRS (Specific Requirement Specification). After documentation, the designing of the prototype of the project is done. Design can be done with traditional as well as modern approaches. Coding begins after approval of the design in which unit-testing is also done. In unit testing, each line of code is executed for the test. This phase is also known as the implementation phase. After coding, a series of testing is conducted by tester. Maintenance is done for fixing errors and bugs discovered after the release of the final product [(Smartsheet, n.d.)](#e).

## Tools and Technology

* **Google**: Google is used for feasibility study and requirement analysis.
* **Figma**: Figma was used in design phase to create prototypes.
* **Photoshop**: It was used to create photos.
* **YouTube**: YouTube was used for the purpose for research.
* **Visual studio code**: It was used as IDE for the purpose of coding.
* **Python**: We have used python as a programming language and interpreter.
* **Tkinter**: Tkinter was used as a GUI framework of python.
* **SQLite3**: It was used for database in this project.
* **Git and GitHub**: Both Git and GitHub were used for version control.
* **Microsoft Excel**: It was used to make Gantt chart.

### Figure 6

Tools and technology

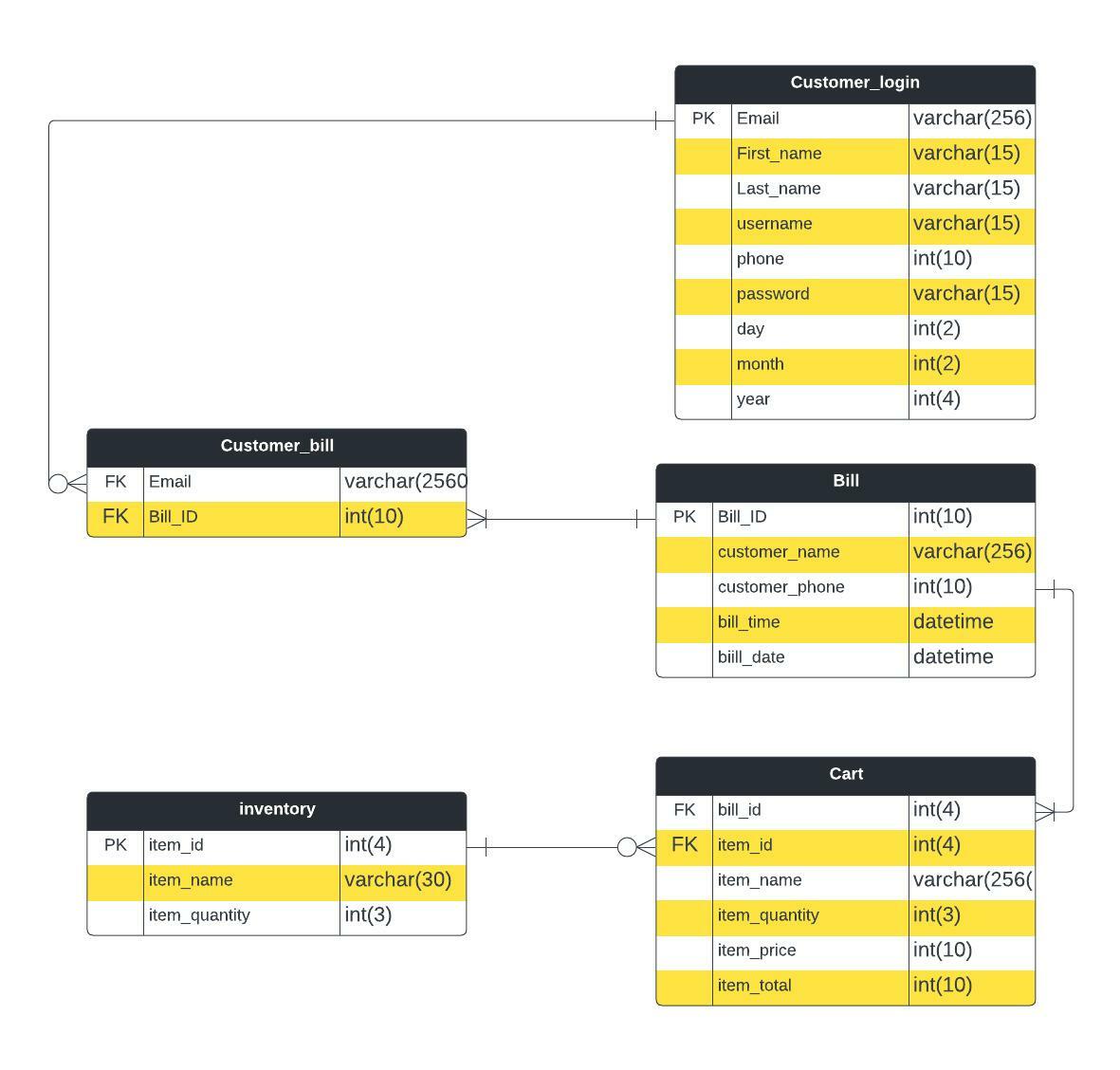


Figure 6 shows the tools and technologies used in this project.

## Conceptual Diagram

### Figure 7

*ER Diagram of the project*

**

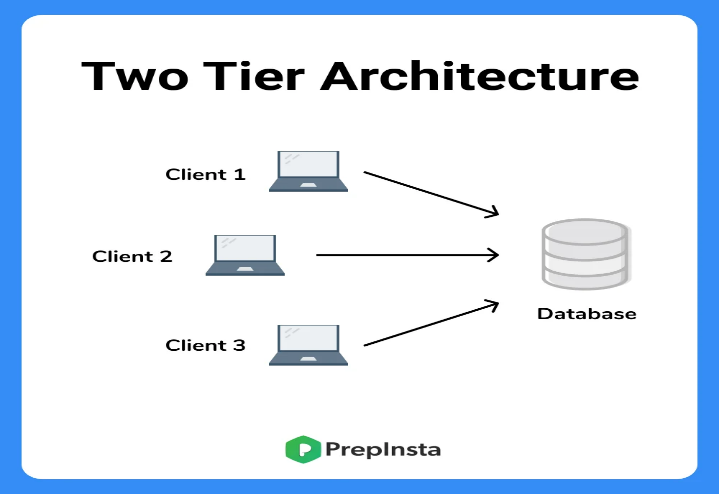
# System Architecture

The developed application is based on two tier architecture in which there are two layers. In this type of architecture, a presentation layer or interface runs on a client, and a data layer gets stored separately. Two tier architecture provides added security to the database, and it also provide faster and direct communication.

In our project, user data like restaurant details, customer details, order details, menu details and sales details are stored in server system. However, these data can be retrieved from client system as per need. Hence users can access it through different devices. As it is simple and easily understandable it provides direct and faster communication. The data can also be modified with proper authentication. This type of two layered database enables quickness in computing and security of data. Figure 8 presents a two-tier architecture.

### Figure 8

Two tier Architecture ([PosterMyWall, n.d.).](#d)



# Project Plan

### Figure 9

Gantt Chart

# Prototype

### Figure 10

*Login Page*

*Graphical user interface, application

Description automatically generated*

### Figure 11

Register page

Graphical user interface, application

Description automatically generated

### Figure 12

*Menu Page*

A picture containing diagram

Description automatically generated

### Figure 13

*Service Page*

*Graphical user interface

Description automatically generated*

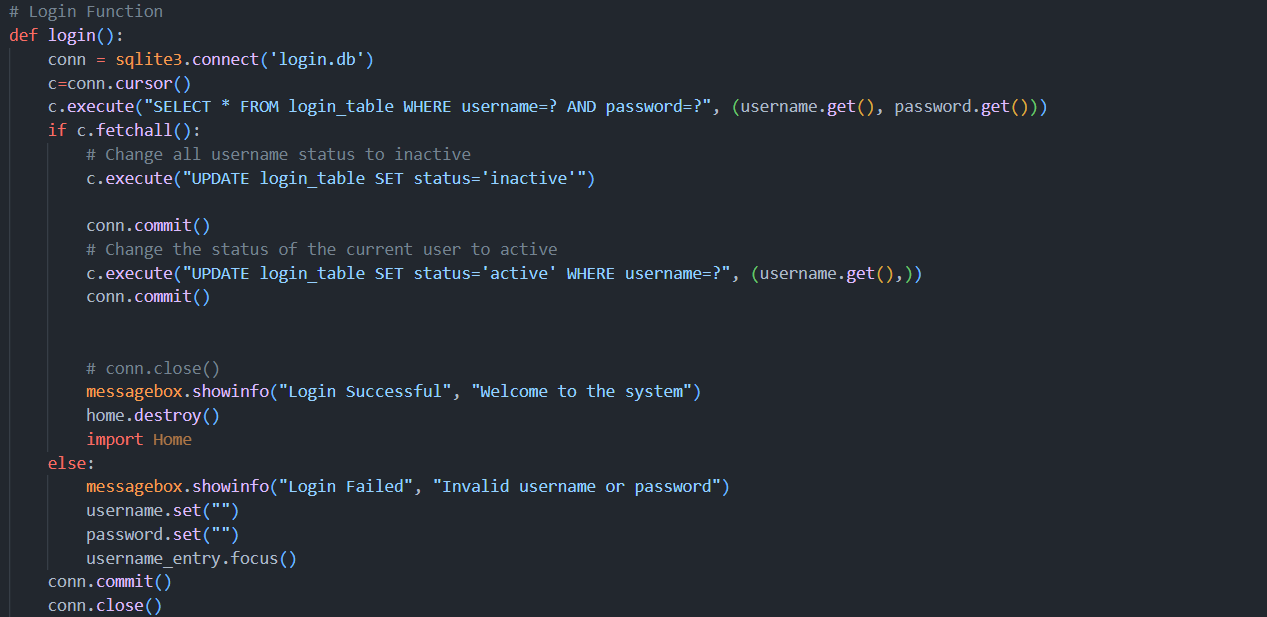
# Developed System

## Login page

In Login page, User can login into the system after inputting correct username and password in the entry box. After clicking forgot password button, reset password display is opened. There are conditions applied in the login function. If the entry boxes are empty, an error message box is displayed. If the username is not registered an error is displayed.

### Figure 14

Login codes and developed system



### Figure 15

GUI of Login



## Reset password

User can also reset their password by clicking on the “Forget password” feature which needs to enter the email and security question answer while registering for the system. After this user can enter a new password to reset the old password.

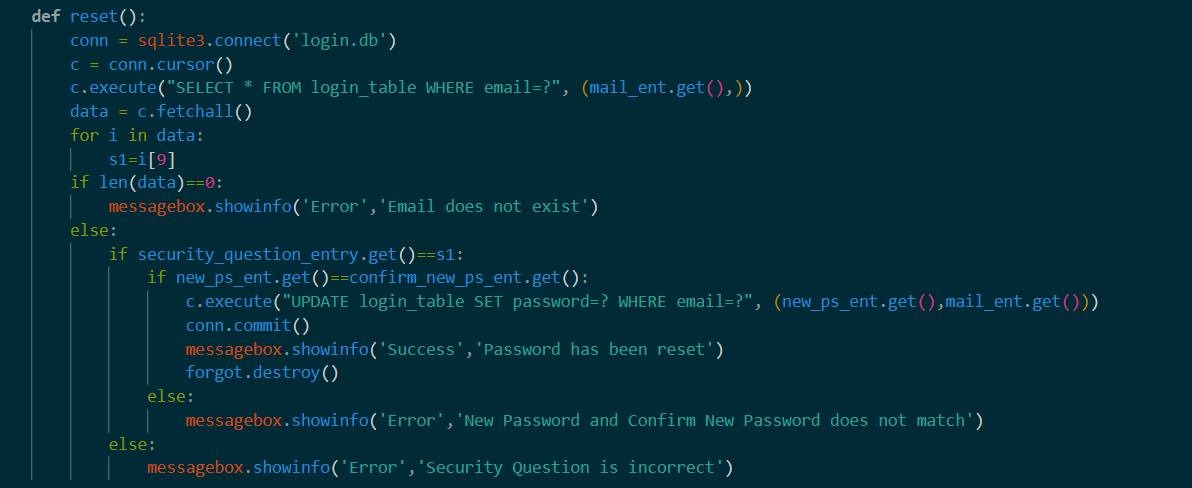
### Figure 16

GUI of Reset password



### Figure 17

Image of reset password code

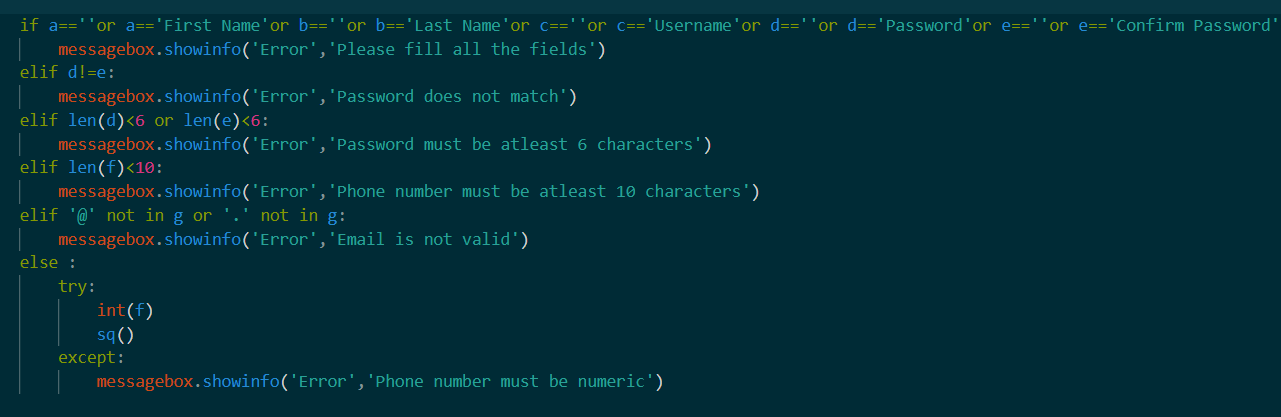


## Sign up page

In this page user have to input their details as they want and add verification. It also consists of exception handling using try and except blocks. In this page, conditions are used to reduce bugs and errors. If there is no input in entry boxes, account is not registered. Valid phone number is also checked using int(phone) and length of phone number. If already registered, user can press ‘ALREADY HAVE AN ACCOUNT’, ‘LOGIN’ button to directly login. User are required fill all the fields to sign up or register. Email is valid or not is also checked. Sign Up button functions as a register button.

### Figure 18

Image of Sign-up codes



### Figure 19

GUI of sign-up page

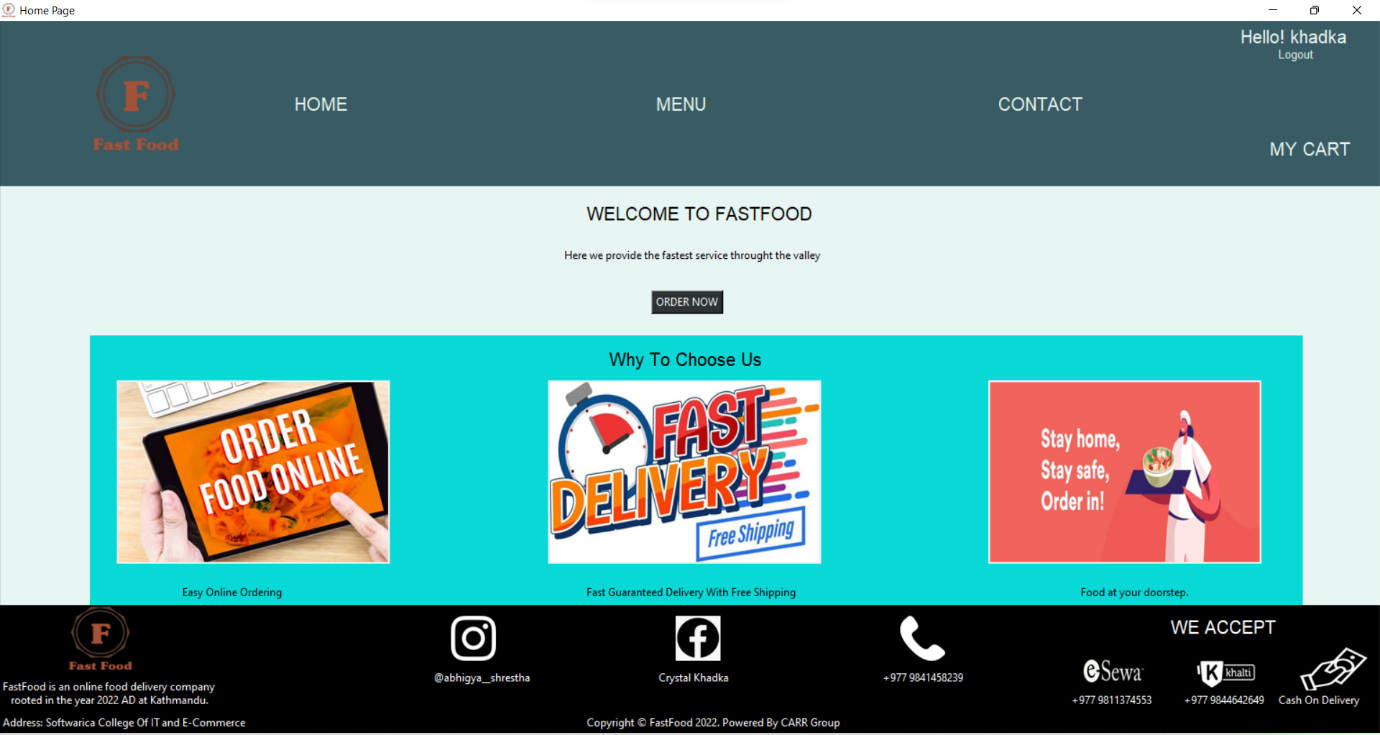


## Homepage

After login, a home page is displayed. In which there is a navigation bar to help the user navigate the app. There is a dashboard in navigation bar. In the body of homepage, there is about the app and an order button. After clicking order button, a menu page is shown. There is also about the services we provide in the homepage.

### Figure 20

GUI of Homepage

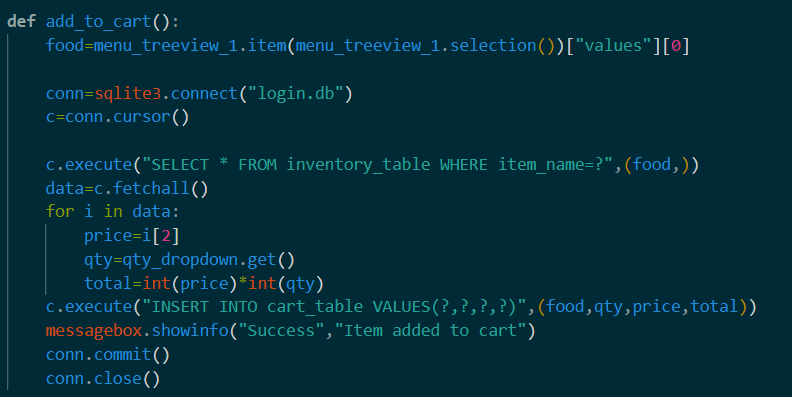


## Menu Page

Menu page can be accessed by the customer after clicking order button in homepage or after clicking menu button in the navbar. In menu customer can see menu of foods that they can add to cart. There is a quantity list to select quantity of food. Along with add to cart button and show cart button. After clicking add to cart, item gets added to cart.

### Figure 21

Codes of menu page



### Figure 22

GUI of Menu page

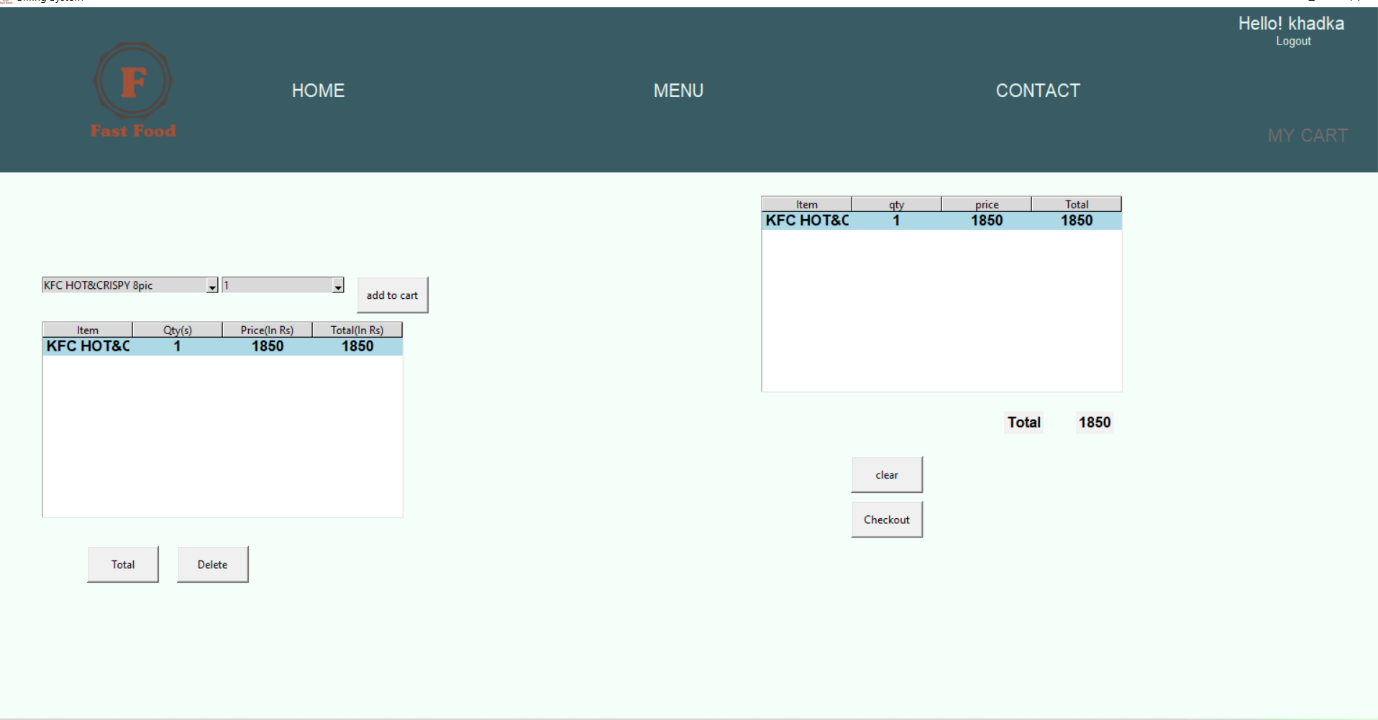


## Cart page

After clicking show cart in menu page or clicking cart button in navbar, customers get navigated to the cart page. In which customer can check their cart, delete unwanted food/item or add food. Customer also clears their whole cart after clicking the clear cart button. Bill can be generated by clicking checkout button which also clears the cart.

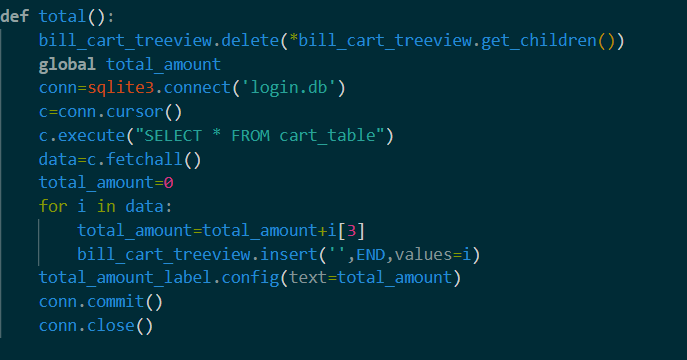
### Figure 23

GUI of cart page



### Figure 24

Codes of cart page

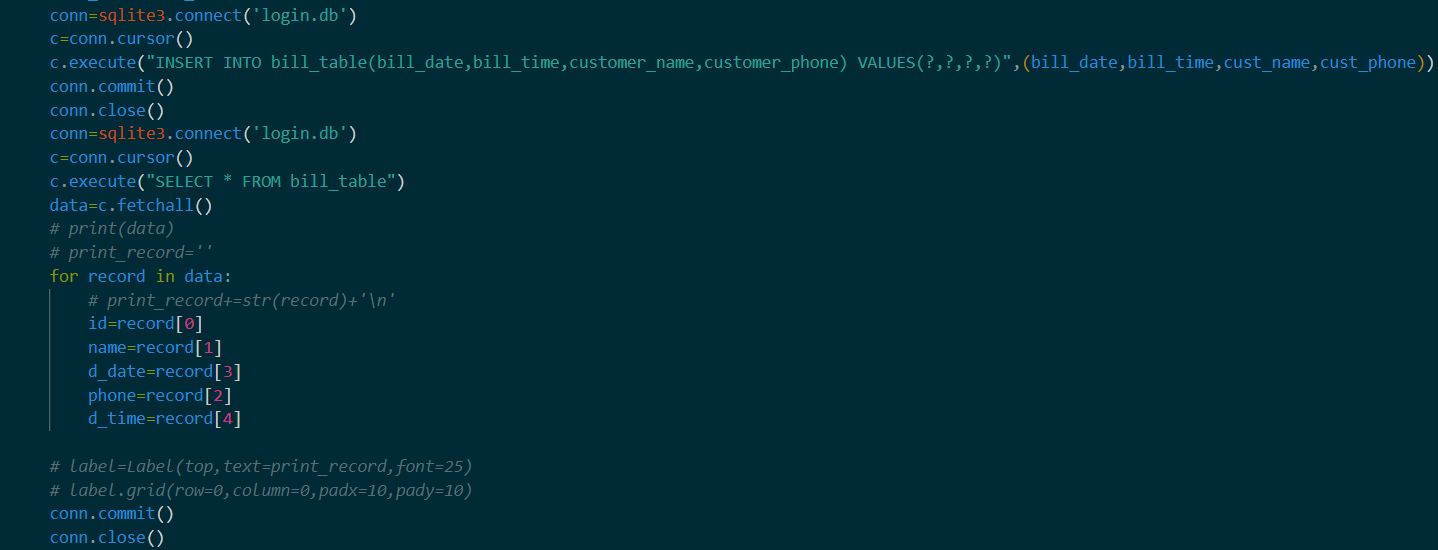


## Billing System

Bill can be generated by the user after clicking checkout in the cart page. Every bill has a unique bill id which is auto generated by the system. User can pay the bill through clicking pay button. After paying, cart is cleared and bill is saved in database.

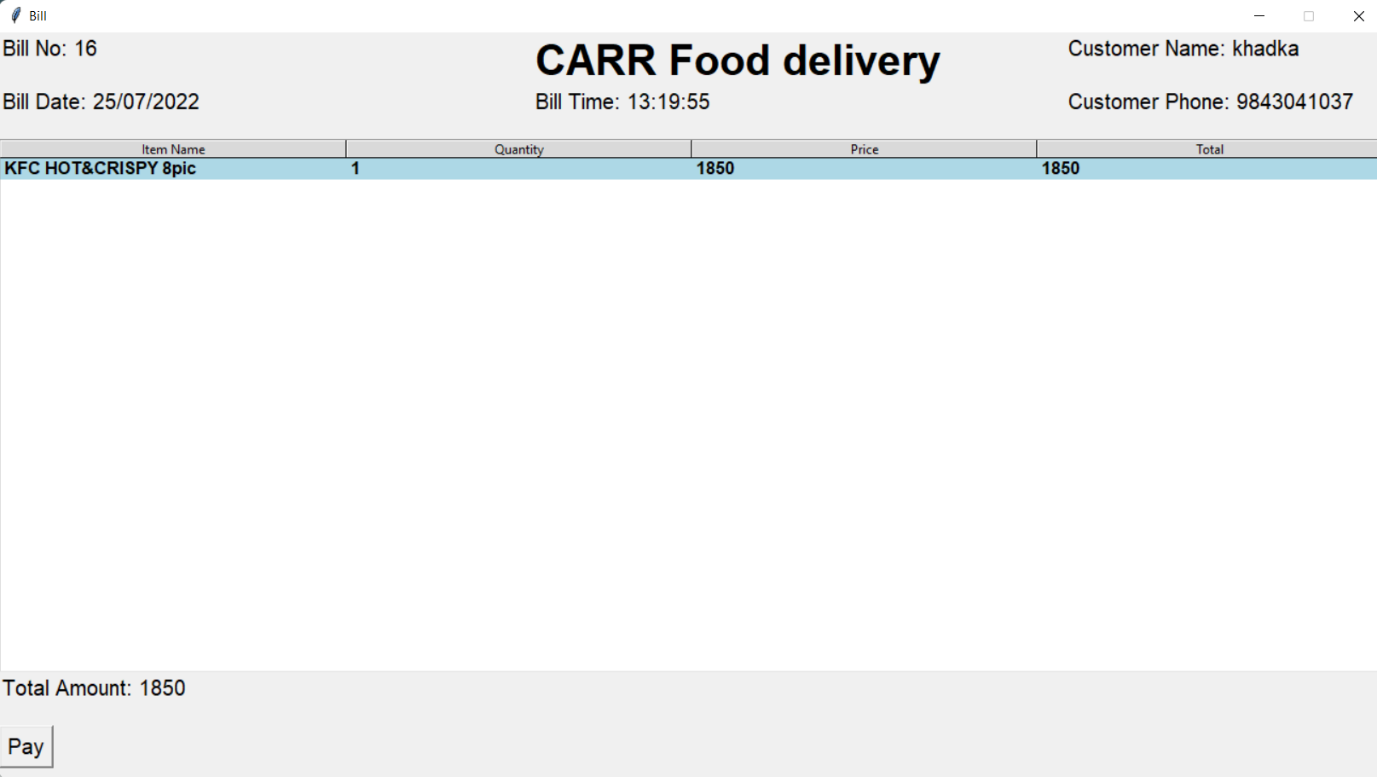
### Figure 25

Codes of Billing System



### Figure 26

GUI of Billing system

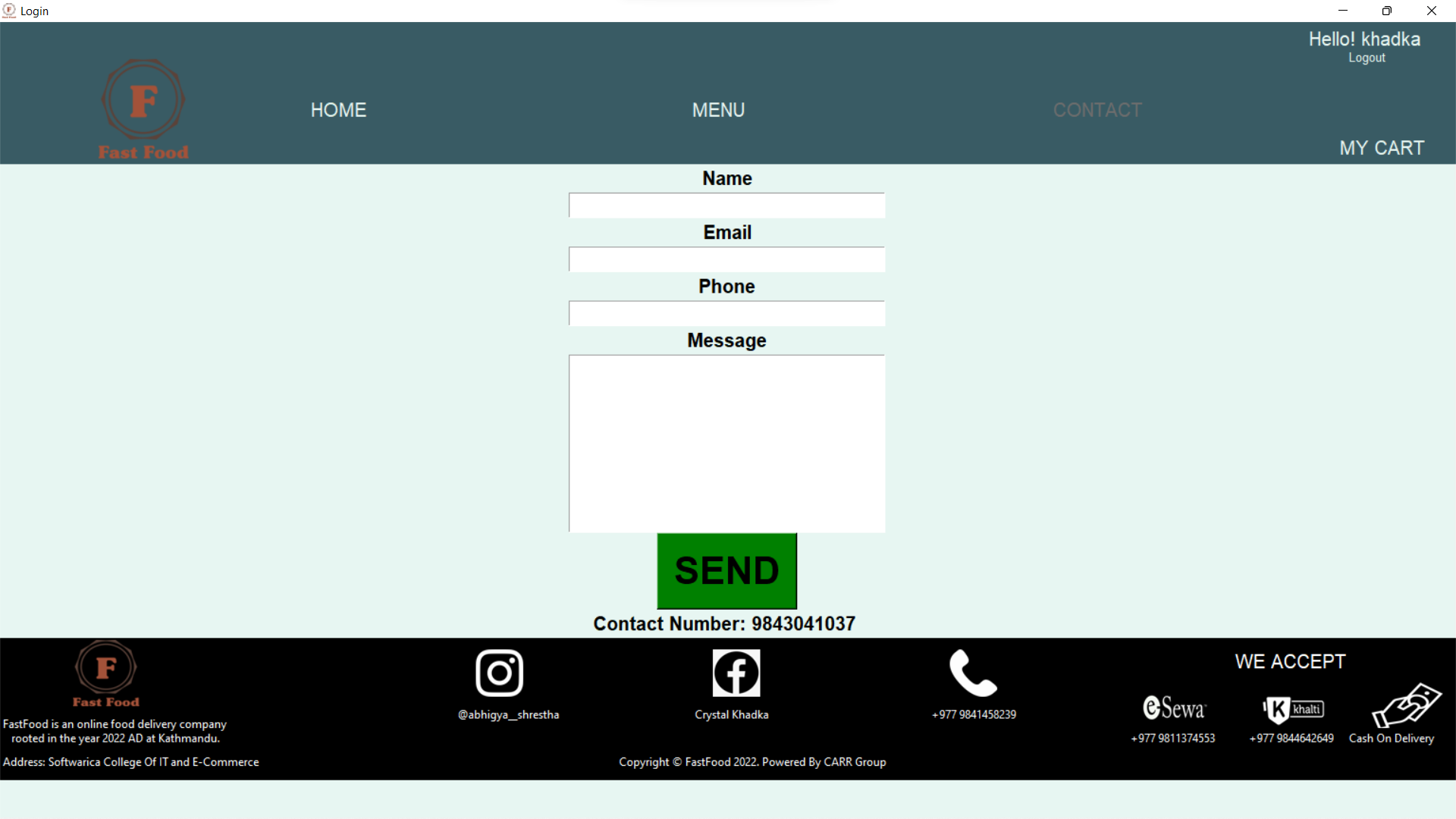


## Contact

Customer can message their name and email with their corresponding message to the developers. The contact page is opened after clicking the contact button in the navbar. Customers are required to fill all the entry boxes for messages to be sent.

### Figure 27

Image of contact developed system



### Figure 28

Codes of contact page

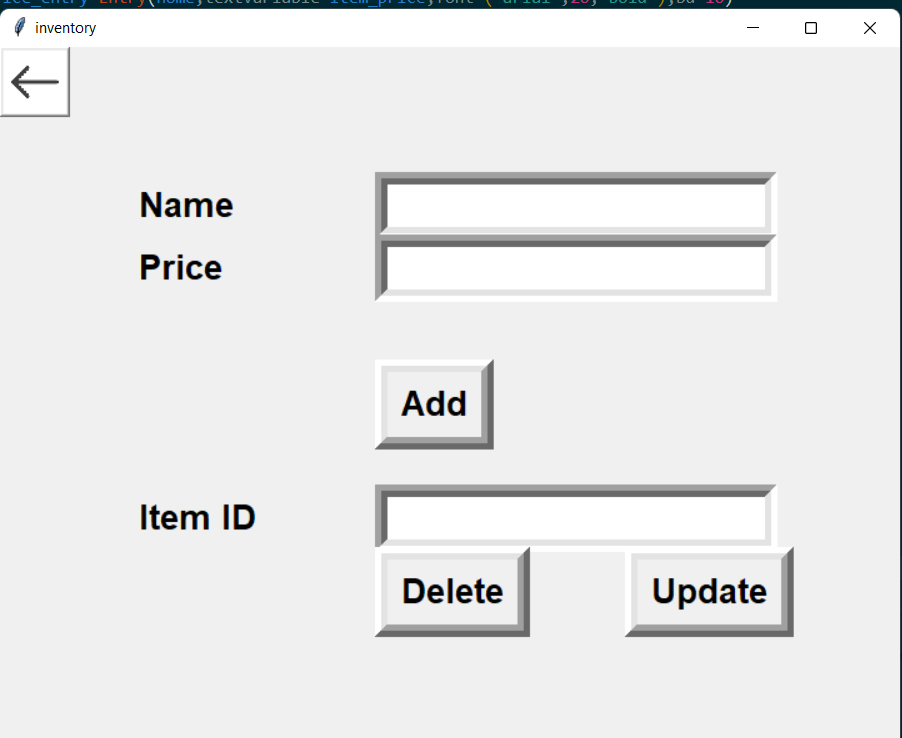


## Inventory page

Inventory page can be accessed through menu page if current admin password is clicked. Item can be added, deleted, and updated in the inventory page.

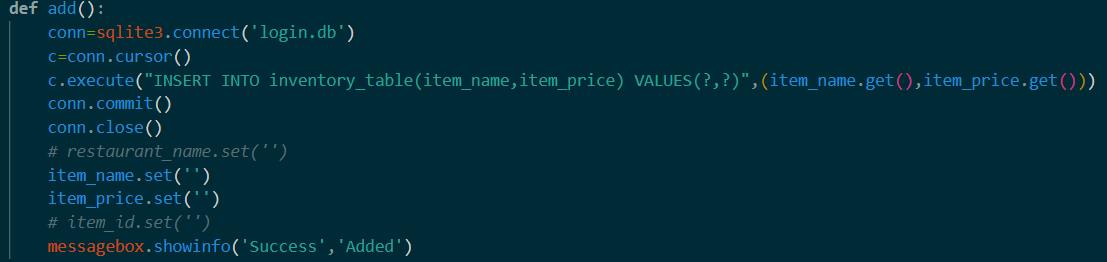
### Figure 29

GUI of inventory page



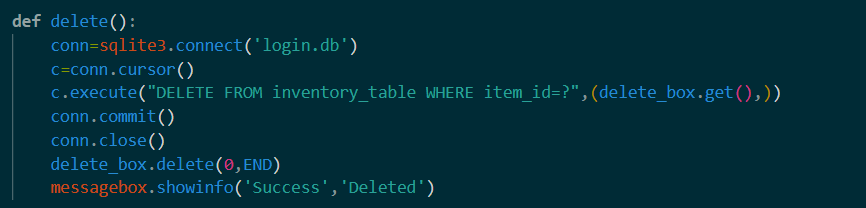
### Figure 30

Inventory add code

****

### Figure 31

Inventory delete code

****

## Database Table

There is use of database in the system. There is total four tables used. There is login table, cart table, inventory table and bill table. Database can be used to store data.

### Figure 32

Code for bill table



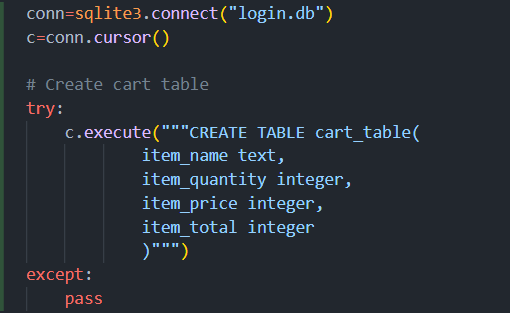
### Figure 33

Code for login table



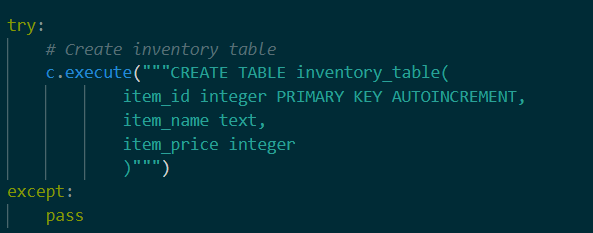
### Figure 34

Code for cart table



### Figure 35

Code for inventory table

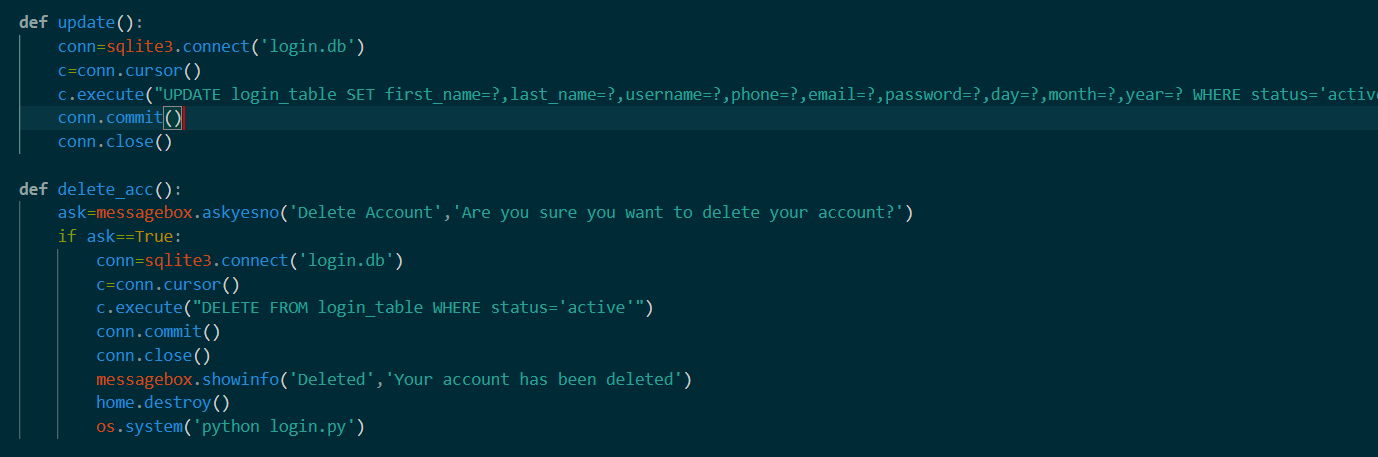


## Account page

Account page can be accessed by clicking ‘Hello! Username’ button in the navbar. In account page we can check the details of the account which has status active. The details of account can be changed by the user. If user doesn’t want the account, user can delete their account. User can also logout from account page.

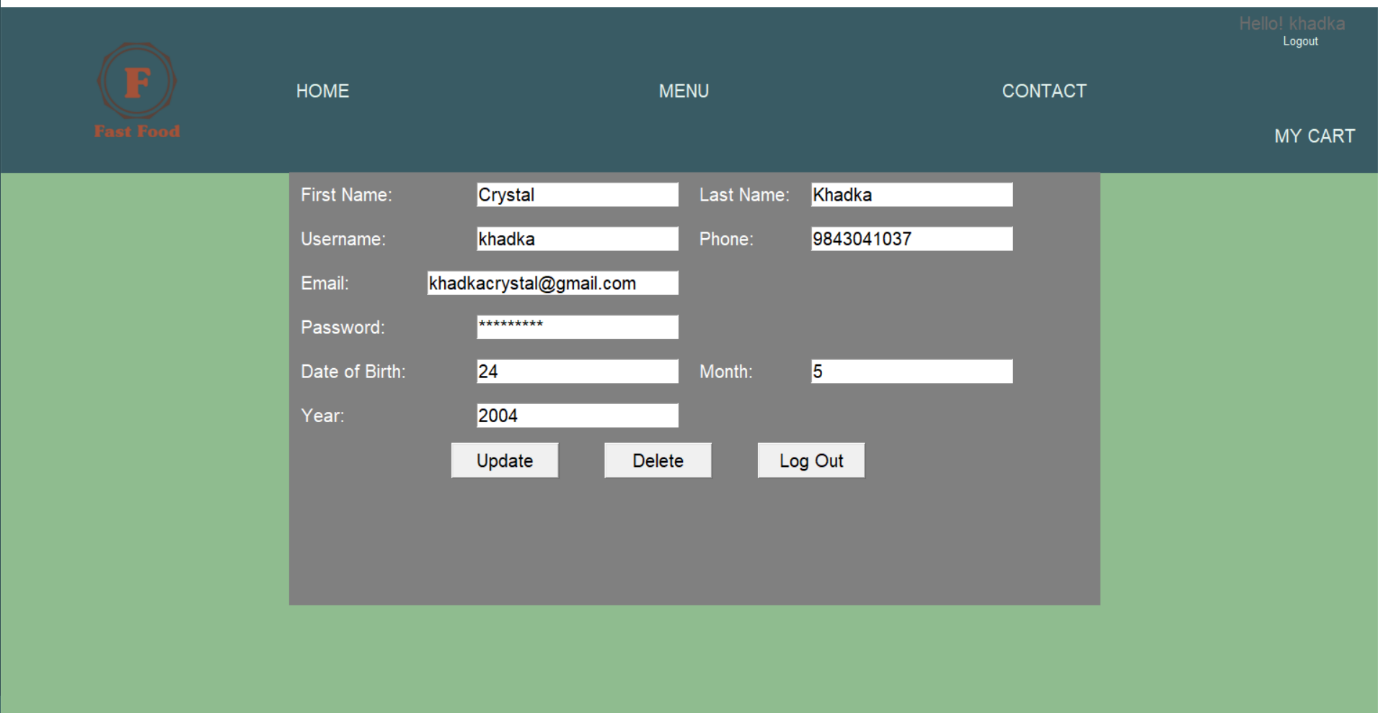
### Figure 36

Image of codes for account page

**

### Figure 37

GUI of account page

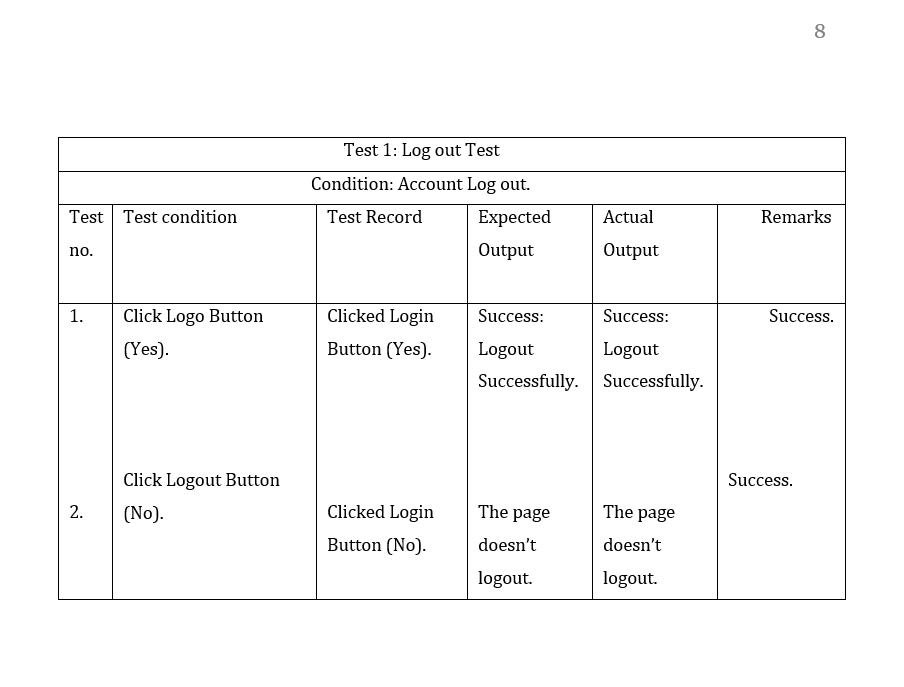


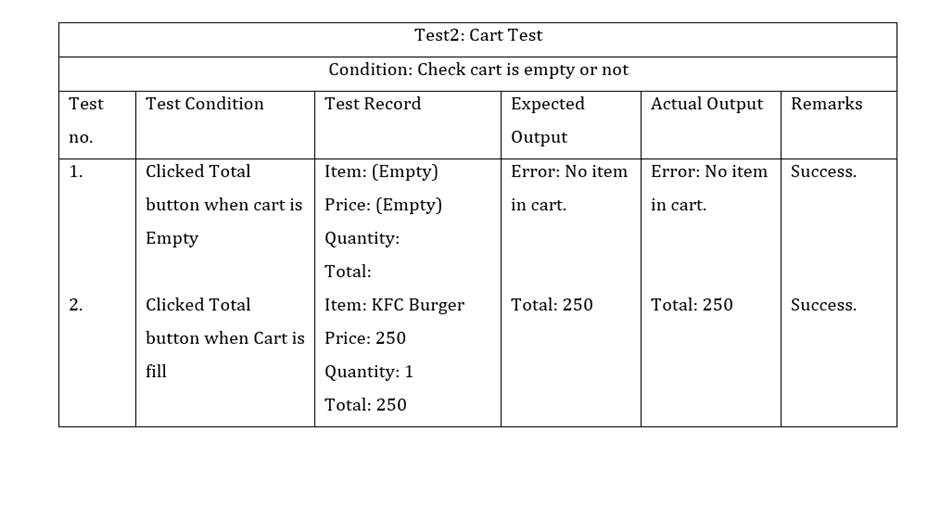
# System Testing

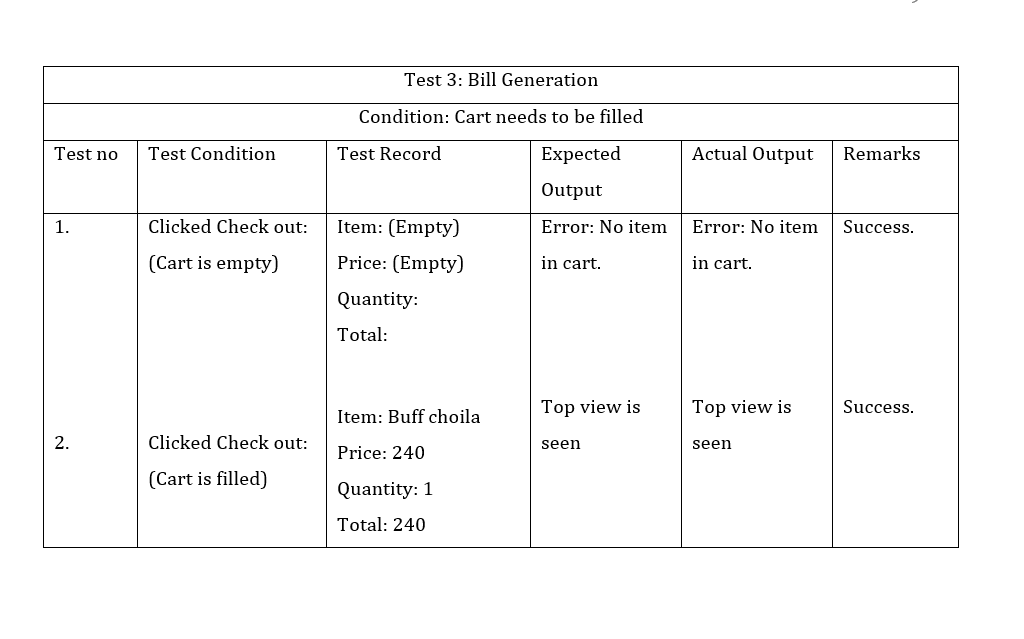
The software was tested using black box test. The tests were all successful as shown in figure 37:

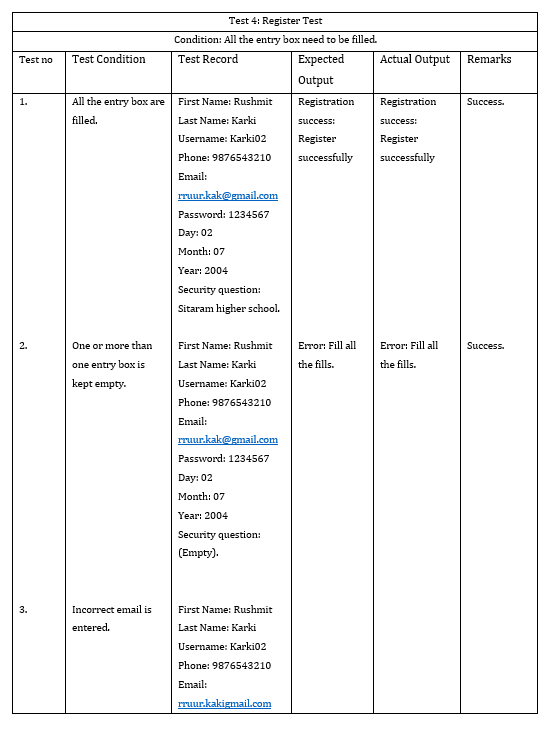
### Figure 38

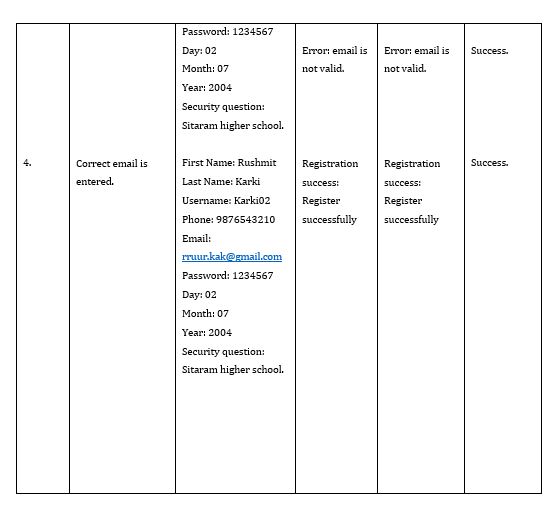
Black Box testing

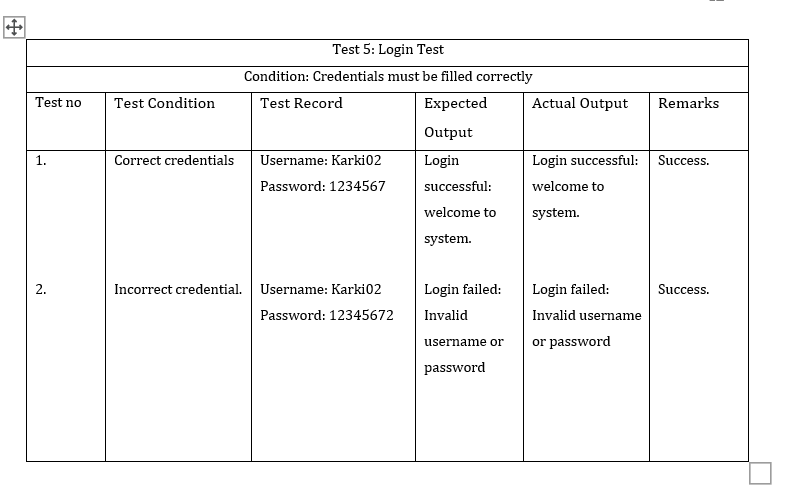


**









# Version Control

**Git hub links:** [**https://github.com/softwarica-github/st4008cem-cw-32a-carr.git**](https://github.com/softwarica-github/st4008cem-cw-32a-carr.git)

**YouTube Video: xxx**

# Conclusion

A working software is built by following the modern waterfall model by the team members. All the knowledge obtained in the class were applied in the development of this project from methodology, documentation, designs, coding, and testing. The final result is a working software where customers can order food easily, restaurants can post their menu with just a few clicks and the order is billed automatically by the system with less to no mistakes. Thus, we have satisfied our aim to build a software that orders food online in a simple and fast method.

The main problem of this project was connection to the database because of many customers and restaurants entry. Two tier architecture was used in this project which is later found not to be suitable for multiple requests concurrently and it may cause issue in data integrity. In the next project, the team will perform broad research on this matter and choose an appropriate database to curb problems seen in this project. The team learned the importance of groupwork, interactions and network to share and broaden their experience which was regularly stated by the instructor in the class.

# References

Fixlastmile, P. (2021, July 5). *What are the challenges in starting a food-delivery solution and how can you solve it?* Fixlastmile. Retrieved June 6, 2022, from <https://www.fixlastmile.com/challenges-of-last-mile-delivery-for-food-delivery-business/>.

Gupta, A. (2021, May 4). *How Online Food Delivery/Ordering App Development Can Help Your Business Grow!* SAG IPL - A Technology Blog. Retrieved June 10, 2022, from <https://blog.sagipl.com/online-food-ordering-app-development/>

PosterMyWall. (n.d.). *Food delivery*. Retrieved June 15, 2022, from <https://www.postermywall.com/index.php/art/template/eaf8ef08673fe3f7a195ad2722294d2f/food-delivery-design-template#.Yt1mf3ZBzIV>

Prepinsta Technologies Private Limited. (2022, June 8). *2 Tier Architecture in DBMS | Database Management System | Prepinsta*. PREP INSTA. Retrieved June 9, 2022, from <https://prepinsta.com/dbms/2-tier-architecture/>

Smartsheet. (n.d.). *Waterfall*. Retrieved June 20, 2022, from <https://www.smartsheet.com/content-center/best-practices/project-management/project-management-guide/waterfall-methodology>