VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", Belagavi-590018

A Mini Project Report on

"FOOD ORDERING MANAGEMENT SYSTEM"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING IN
INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

PRADEEP GOWDA H (1JB20IS044)
MANU RATHAN SETTY S (1JB20IS032)

Under the Guidance of

Mr. VISHRUTH B G
Assistant Professor,
Dept. of ISE, SJBIT
Bengaluru-60

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY, KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2022-2023

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Information Science & Engineering

CERTIFICATE

Certified that the Mini-project work entitled "**FOOD ORDERING MANAGEMENT SYSTEM",** is bonafide work carried out by **PRADEEP GOWDA H (1JB20IS044) and MANU RATHAN SETTY S (1JB20IS032),** bonafide students of **SJB Institute of Technology,** in partial fulfilment for 6th semester in **INFORMATION SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** during the academic year **2022-23.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini-project prescribed for the said degree.

Mr. Vishruth B G **Assistant Professor** **Dept. of ISE, SJBIT**	**Dr. Shashidhara H R** **Professor & Head** **Dept. of ISE, SJBIT**

EXTERNAL VIVA

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

I would like to express my profound thanks to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing me an opportunity to pursue my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

I express my gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing excellent facilities and academic ambience, which have helped me in satisfactory completion of mini project work.

I extend my sincere thanks to **Dr. Shashidhara H.R**, Head of the Department, Information Science and Engineering, for providing an invaluable support throughout the period of mini project work.

I wish to express heartfelt gratitude to my guide**, Mr. Vishruth B G,** Assistant Professor, for his valuable guidance, suggestions and cheerful encouragement during the entire period of this work.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, Teaching & Non-teaching staffs of the department, the library staff and all my friends, for their continuous support and encouragement.

PRADEEP GOWDA H
(1JB20IS044)
MANU RATHAN SETTY S
(1JB20IS032)

ABSTRACT

The problem we face in today's restaurant world is not having complete access to the menu. The menu of a restaurant must provide complete details of the food items available in that restaurant. A customer who wants to order food online from any restaurant relies on an online menu, which the application we have built provides. And our aim is to allow the users to order their food online with complete access to the menu of a restaurant. Provide the customers with an opportunity to choose from a variety of dishes and get the food delivered in the quickest possible time. It allows the user to also access his/her previous orders and allows the existing customer to enter into the application using the login gateway and new customers by registering at the sign-in gateway. The basic idea behind this project is to solve the issues faced by customers who order food online. We provide them a hassle-free application to place their order and access their previous orders.

Table of Contents

List of Tables (In separate page)

List of Figures (In separate page)

Chapter 1

INTRODUCTION

1.1 File Structure

A file structure is a combination of representations for data in files and of operations for accessing the data. A file structure allows applications to read, write, and modify data. It might also support finding the data that matches some search criteria or reading through the data in some order. An improvement in file structure design may make an application hundreds of times faster. The details of the representation of the data and the implementation of the operations determine the efficiency of the file structure for applications.

1.2 History

Early work with files presumed that files were on tape since most files were. Access was sequential, and the cost of access grew in direct proportion to the size of the file. As files grew intolerably large for unaided sequential access and as storage devices such as hard disks became available, indexes were added to files. The indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly. With key and pointer, the user had direct access to the large, primary file. But simple indexes had some of the same sequential flawsas the data file, and as the indexes grew, they too became difficult to manage, especially for dynamic files in which the set of keys changes.

In the early 1960's, the idea of applying tree structures emerged. But trees can grow very unevenly as records are added and deleted, resulting in long searches requiring many diskaccessestofindarecord.
In 1963, researchers developed an elegant, self-adjusting binary tree structure, called AVL tree, for data in memory. The problem was that, even with a balanced binary tree, dozens of accesses were required to find a record in even moderate-sized files. A method was needed to keep a tree balanced when each node of the tree was not a single record, as in a binary tree, but a file block containing dozens, perhaps even hundreds, of records. Hashing is a good way to get what we want with a single request, with files that do not change size greatly over time.

Hashed indexes were used to provide fast access to files. But until recently, hashing did not work well with volatile, dynamic files. Extendible dynamic hashing can retrieve information with 1 or at most 2 disk accesses, no matter how big the file became.

1.3 About the File

When we talk about a file on disk or tape, we refer to a particular collection of bytes stored there.A file, when the word is used in this sense, physically exists. A disk drive may contain hundreds, even thousands of these physical files. From the standpoint of an application program, a file is somewhat like a telephone line connection to a telephone network. The program can receive bytes through this phone line or send bytes down it, but it knows nothing about where these bytescome from or where they go. The program knows only about its end of the line. Even though there may be thousands of physical files on a disk, a single program is usually limited to the use of only about 20 files.

The application program relies on the OS to take care of the details of the telephone switching system. It could be that bytes coming down the line into the program originate from a physical file they come from the keyboard or some other input device. Similarly, bytes the program sends down the line might end up in a file, or they could appear on the terminal screen or some other output device. Although the program doesn't know where the bytes are coming from or where they are going, it does know which line it is using. This line is usually referred to as the logical file, to distinguish it from the physical files on the disk or tape.

1.4 Application of File Structure

Relative to other parts of a computer, disks are slow. 1 can pack thousands of megabytes on a disk that fits into a notebook computer.

The time it takes to get information from even relatively slow electronic random-access memory (RAM) is about 120 nanoseconds. Getting the same information from a typical disk takes 30 milliseconds. So, the disk access is a quarter of a million times longer than a memory access. Hence, disks are *very* slow compared to memory. On the other hand, disks provide enormous capacity at much less cost than memory. They also keep the information stored on them when they are turned off.

Tension between a disk's relatively slow access time and its enormous, non-volatile capacity,

is the driving force behind file structure design. Good file structure design will give us access to allthe capacity without making our applications spend a lot of time waiting for the disk.

1.5 Introduction to the Project

An online food ordering management system is a software application or platform that facilitates the process of ordering food online from various restaurants or food establishments. It streamlines the entire workflow, from order placement to delivery, and provides a convenient and efficient solution for customers, restaurants, and delivery personnel.

Key components and features of an online food ordering management system typically include:

1. User Interface: A user-friendly interface allows customers to browse menus, select items, customize orders, and place orders online. It may include search filters, categories, and images to help users make informed choices.
2. Menu and Item Management: Restaurants can create and manage their menus, including adding, editing, and removing dishes, along with descriptions, prices, and availability. This information is made accessible to customers through the platform.
3. Online Ordering and Payments: Customers can place orders online by selecting items, specifying preferences (e.g., special instructions, dietary restrictions), and making payments securely through various payment methods like credit cards, digital wallets, or cash on delivery.
4. Order Tracking: Once an order is placed, customers can track its status, from preparation to dispatch and delivery. Real-time updates regarding the estimated delivery time and the location of the delivery personnel may also be provided.
5. Restaurant Management: Restaurants can manage incoming orders efficiently through a dedicated dashboard. They can receive order notifications, view order details, and update order status (e.g., confirmed, preparing, dispatched). Additionally, they can manage their inventory, update menu items, and monitor customer reviews and ratings.
6. Delivery Management: The system may include features to manage delivery personnel and their routes. Dispatchers can assign orders to available delivery staff, track their location, and

optimize delivery routes for efficiency. Integration with GPS systems or mobile apps can enhance real-time tracking.

7. Customer Reviews and Ratings: Customers can provide feedback on their ordering experience, food quality, and delivery service through ratings and reviews. This helps other customers make informed decisions and allows restaurants to maintain quality standards.

8. Analytics and Reporting: The system may offer analytical tools and reports to provide valuable insights into sales trends, popular dishes, customer preferences, and other key metrics. This information can be utilized by restaurants to optimize their operations and marketing strategies.

9. Integration and Partnerships: Integration with third-party services such as online payment gateways, food delivery aggregators, and customer relationship management (CRM) systems can enhance the functionality of the platform and provide a seamless experience.

Overall, an online food ordering management system simplifies the process of ordering food online, improves operational efficiency for restaurants, and enhances the convenience for customers by offering a streamlined and user-friendly interface.

1.6 Objective

The main objectives of the project are as follows:

- Display the menu of a restaurant.
- Take an order.
- Display the billing details.
- Store the order details for future references.

Chapter 2

SYSTEM REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users' details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a menu driven console-based application whose requirements are mentioned in this section.

The specific requirements of Bus Billing System are stated as follows:

2.1 Software Requirements

1. OS: Windows XP,7,8,10,11
2. Editor: Microsoft Visual Studio / Microsoft Visual Basic C++ 6.0

2.2 Hardware Requirements

Hardware Components used:

1. Pentium IV Processor, i3, i4, i5, i7
2. 1GB RAM
3. 40GB HDD
4. 1024 * 768 Resolution Colour Monitor

- The hardware requirements specified are the hardware components/capacity of the system inwhich the application is developed and deployed.
- The above software requirements are the necessary software required to develop the applicationand run the application.
- Microsoft Visual Basic C++ 6.0 is used to develop the application on windows platform.
- The project is developed in C++ language with concepts of File handling and arrays.

2.3 Technology Used:

C++ is a general-purpose programming language created as an extension of the c programming language or "C with classes ". The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented asa compiled language, and many vendors provide C++ compilers, including the free software foundation, Microsoft, Intel, Oracle, and IBM so it is available on many platforms.

C++ was designed with an orientation toward systems programming and embedded, resource-constrainedsoftware and large systems, with performance, efficiency, and flexibility of use as its design highlights.

C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, video games, servers (e.g., e- commerce, web search, or databases), and performance-critical applications.

C++ is standardized by the ISO, with the latest standard version ratified and published by ISO in December 2020 as *ISO/IEC 14882:2020* (informally known as C++20). The C++ programming language was initiallystandardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the C++03, C++11, C++14, and C++17 standards. The current C++20 standard supersedes these with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Stroustrup at Bell Labs since 1979 as an extension of the C language; he wanted an efficient and flexible language similar to C that also provided high-level features for program organization. Since 2012, C++ has been on a three-year release schedule with C++23 as the next planned standard.

Features of C++Programming Language:

- Object oriented.
- Simple
- Platform dependent
- Mid-level programming language
- Structured programming language
- Rich library
- Memory management
- Powerful and fast
- Pointers
- Compiler based.
- Syntax based language.

Chapter 3

System Design

The purpose of the design phase is to develop a clear understanding of what the developer wants people to gain from his/her project. As the developer works on the project, the test forevery design decision should be.

"Does this feature fulfil the ultimate purpose of the project?"

A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself. The Design Document will verify that the current design meets all the explicit requirements contained in the system model as well asthe implicit requirements desired by the customer.

3.1 Operations Performed on a File

- Insertion:

  The system is used to add details related to a customer when he/she is entering the application for the first time through sign-up gateway.
- Display:

  The system is used to display the details of an order made by the user by printing the bill amount and the ETA of food delivery.
- Search:

  The system is used to search for the orders previously placed by a customer and also while searching for the existing user when he/she tries to sign up with the same credentials.

3.2 Data Flow Diagrams:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig.

progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.

Data flow diagrams were popularized in the late 1970s, arising from the book Structured Design, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the "data flow graph" computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

Also contributing were two related concepts:

- Object Oriented Analysis and Design (OOAD), put forth by Yourdon and Peter Coad to analyse and design an application or system.

- Structured Systems Analysis and Design Method (SSADM), a waterfall method to analyse and design information systems. This rigorous documentation approach contrasts with modern agile approaches such as Scrum and Dynamic Systems Development Method (DSDM.) Three other experts contributing to this rise in DFD methodology were Tom DeMarco, Chris Gane and Trish Sarson. They teamed up in different combinations to be the main definers of the symbols and notations used for a data flow diagram. A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

The below figure shows zero level data flow diagram of restaurant management system. Zero level data flow diagrams concentrate mainly on overview of the whole system or process being analysed or modelled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

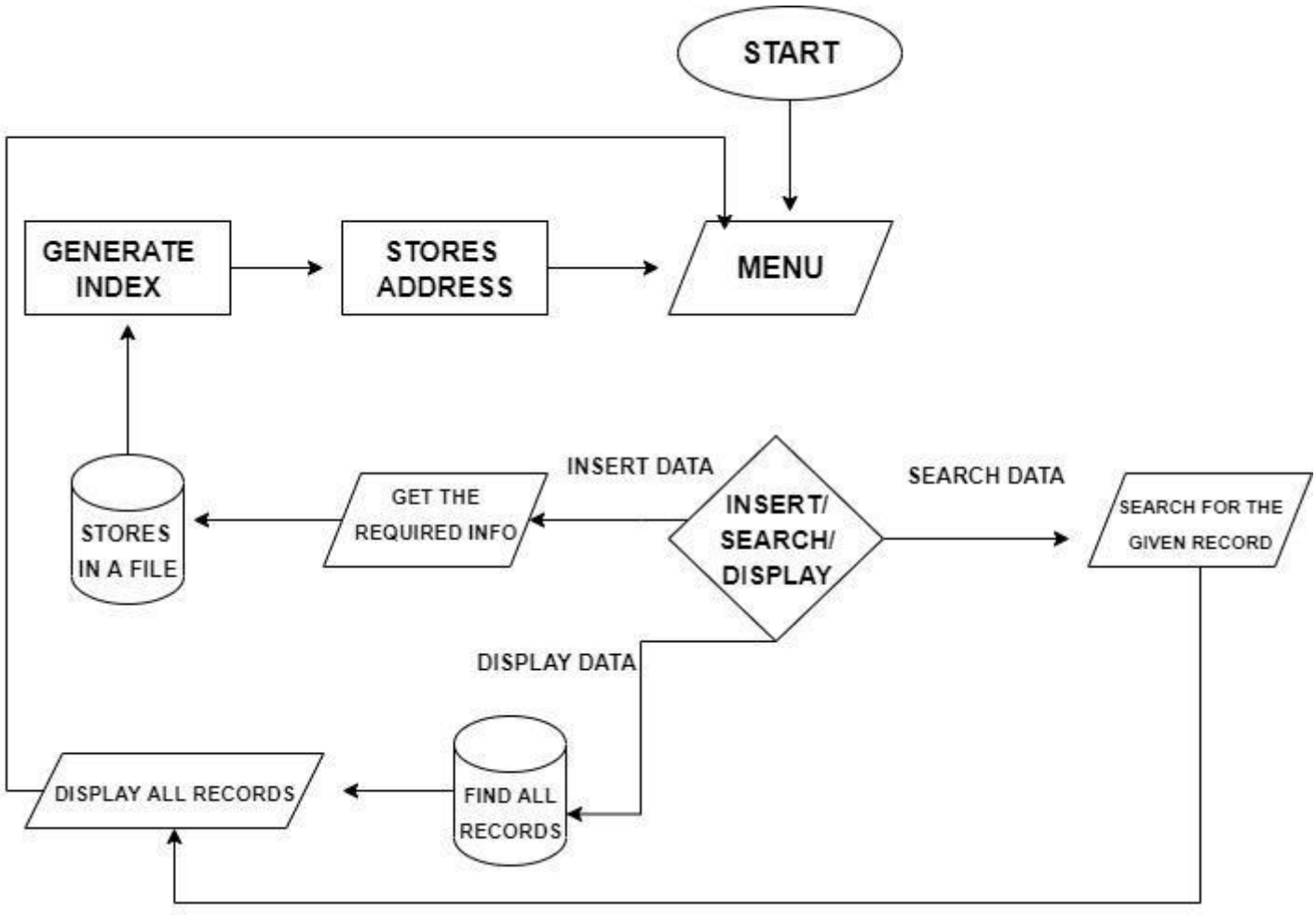


Fig. 3.1 Data Flow Diagram for Restaurant Management System

Chapter 4

Implementation

Implementation is the stage in the project where the theoretical design is turned into a working system and gives confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover, and an evaluation of change over methods.

Implementation is the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. Any system developed should be secured and protected against possible hazards.

Component testing is a method where testing of each component in an application is done separately. Component testing is also known as module, unit, or program testing. It finds the defects in the module and verifies the functioning of software.

```
#include<fstream>
#include<iostream>
#include<cstring>
#include<conio.h>
#include<process.h>
#include<stdlib.h>

using namespace std;

fstream fp,fp1;
char username[15], password[10], phone[10], address[50];
char pizza1[]="Chicken-Fazita" ,pizza2[]="Chicken-Bar-BQ" ,pizza3[]="Peri-Peri" ,pizza4[]="Creamy-Max";
char roll1[]="Chicken-Chatni-Roll", roll2[]="Chicken-Mayo-Roll", roll3[]="Veg-Roll-With-Fries";
char bur1[]="Zinger-Burger",bur2[]="Chicken-Burger",bur3[]="Paneer-Burger";
char sand1[]="Club-Sandwich", sand2[]="Chicken-Crispy-Sandwich", sand3[]="Extream-Veg-Sandwich";
char bir1[]="Chicken-Biryani", bir2[]="Prawn-Biryani", bir3[]="Paneer-Biryani", gotostart;
int choice=0,quantity,pchoice,pchoice1;
class Order {
    private:
        char name[30],phone[10],city[20],order[50],price[5];
    public:
        void search();
        void orderItem(char [],char []);
        bool validateLogin(char [],char []);
        void saveSignup(char[],char[],char[],char[]);
        void bill(int,int,char[],char[],char[]);
        void previousOrder(char [],char []);
};
```

```
void Order::previousOrder(char username[],char password[]){
            fp1.open("Orders.txt",ios::in);
            char buffer1[300],buffer2[300],temp[300],temp2[300];
            char *t;
            char *p;
            strcpy(buffer1,username);
            strcat(buffer1,password);
            while(!fp1.fail())
            {
            fp1>>buffer2;
            strcpy(temp,buffer2);
            strcpy(temp2,buffer2);
            t=strtok(temp,"|");
            if((strcmp(buffer1,t)==0))
            {
                p=strtok(temp2,"|");
                cout<<"Your previous order :\n";
                p=strtok(NULL,"|");
                cout<<"Quantity: "<<p<<endl;
                p=strtok(NULL,"|");
                cout<<"Total price: "<<p<<endl;
                p=strtok(NULL,"|");
                cout<<"Item name: "<<p<<endl;
            }
            }
}
```

```
void Order::bill(int quantity, int choice, char fooditem[], char username[], char password[])
{
            cout<<"\t\t\t--------Your Order---------\n";
            cout<<""<<quantity<<" "<<fooditem;
            cout<<"\nYour Total Bill is "<<choice<<".00\nYour Order Will be delivered in 40 Minutes";
            cout<<"\n\nThank you For Ordering From Carl's Jr. Fast Food\n";
            char buffer2[200];
            fp1.open("Orders.txt",ios::app);
            strcpy(buffer2,username);
            strcat(buffer2,password);
            strcat(buffer2,"|");
            char str1[10]={0};
            itoa(quantity,str1,10);
            char cstr1[10]={0};
            itoa(choice,cstr1,10);
            strcat(buffer2,str1);
            strcat(buffer2,"|");
            strcat(buffer2,cstr1);
            strcat(buffer2,"|");
            strcat(buffer2,fooditem);
            strcat(buffer2,"|");
            fp1<<buffer2;
            fp1<<"\n";
            fp1.close();
}

```

```
void Order::orderItem(char username[],char  password[])
{
    cout<<"\t\t\t----------Peter's Jr. Fast Food----------\n\n";
    cout<<"Please Enter Your Name: ";
    cin>>name;
    beginning:
    cout<<"Hello "<<name<<", What would you like to order?\n\n";
    cout<<"\t\t\t\t--------Menu--------\n\n";
    cout<<"1) Pizzas\n";
    cout<<"2) Burgers\n";
    cout<<"3) Sandwich\n";
    cout<<"4) Rolls\n";
    cout<<"5) Biryani\n\n";
    cout<<"\nPlease enter your Choice: ";
    cin>>choice;

    if(choice==1)
     {
        cout<<"\n1) "<<pizza1<<"\n";
        cout<<"2) "<<pizza2<<"\n";
        cout<<"3) "<<pizza3<<"\n";
        cout<<"4) "<<pizza4<<"\n";
        cout<<"\nPlease Enter which Flavour would you like to have?:";
        cin>>pchoice;
     if(pchoice>=1 && pchoice<=4)
        {
            cout<<"\n1) Small Rs.250\n"<<"2) Regular Rs.500\n"<<"3) Large Rs.900\n";
            cout<<"\nChoose Size Please:";
            cin>>pchoice1;
            if(pchoice1>=1 && pchoice1<=3)
            {
```

```
            {
             cout<<"\nPlease Enter Quantity: ";
             cin>>quantity;
             switch(pchoice1)
             {
             case 1: choice = 250*quantity;
             break;
             case 2: choice = 500*quantity;
             break;
             case 3: choice = 900*quantity;
             break;
             }
            }
            else
            cout<<"Invalid Choice\n\n";
        switch (pchoice1)
        {
             case 1:
             bill(quantity,choice,(char *)"Chicken-Fazita",username,password);
             break;
             case 2:
             bill(quantity,choice,(char *)"Chicken-Bar-BQ",username,password);
             break;
             case 3:
             bill(quantity,choice,(char *)"Peri-Peri",username,password);
             break;
             case 4:
             bill(quantity,choice,(char *)"Creamy-Max",username,password);
             break;
             default : cout<<"Invalid Choice\n\n";
        }
```

```
            cout<<"Would you like to order anything else? Y / N:";
            cin>>gotostart;
            if(gotostart=='Y' || gotostart=='y')
            {
              goto beginning;
            }
            else
              return;
    }
    else
    {
        cout<<"Invalid Choice\n\n";
        goto beginning;
    }
}
    else if(choice==2)
     {
        cout<<"\n1 "<<bur1<<" Rs.180"<<"\n";
        cout<<"2 "<<bur2<<" Rs.150"<<"\n";
        cout<<"3 "<<bur3<<" Rs.160"<<"\n";
        cout<<"\nPlease Enter which Burger you would like to have?: ";
        cin>>pchoice1;
        if(pchoice1>=1 && pchoice1<=3)
        {
            cout<<"\nPlease Enter Quantity: ";
            cin>>quantity;
            switch(pchoice1)
            {
            case 1: choice = 180*quantity;
            break;
            case 2: choice = 150*quantity;
            break;
```

```
            case 3: choice = 160*quantity;
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            switch (pchoice1)
            {
             case 1:
             bill(quantity,choice,(char *)"Zinger-Burger",username,password);
             break;
             case 2:
             bill(quantity,choice,(char *)"Chicken-Burger",username,password);
             break;
             case 3:
             cout<<"\t\t--------Your Order---------\n";
             bill(quantity,choice,(char *)"Paneer-Burger",username,password);
             break;
             default : cout<<"Invalid Choice\n\n";
            }
            cout<<"\nWould you like to order anything else? Y / N:";
            cin>>gotostart;
            if(gotostart=='Y' || gotostart=='y')
            {
              goto beginning;
            }
            else
              return;
            }
            else
```

```
            else
    {
        cout<<"Invalid Choice\n\n";
        goto beginning;
    }
    }
    else if(choice==3)
     {
        cout<<"\n1  "<<sand1<<" Rs.240"<<"\n";
        cout<<"2  "<<sand2<<" Rs.160"<<"\n";
        cout<<"3  "<<sand3<<" Rs.100"<<"\n";
        cout<<"\nPlease Enter which Sandwich you would like to have?:";
        cin>>pchoice1;
        if(pchoice1>=1 && pchoice1<=3)
        {
            cout<<"\nPlease Enter Quantity: ";
            cin>>quantity;
            switch(pchoice1)
            {
            case 1: choice = 240*quantity;
            break;
            case 2: choice = 160*quantity;
            break;
            case 3: choice = 100*quantity;
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            switch (pchoice1)
            {
             case 1:
             bill(quantity,choice,(char *)"Club-Sandwich",username,password);
             break;
```


```
            else
    {
        cout<<"Invalid Choice\n\n";
        goto beginning;
    }
    }
    else if(choice==3)
     {
        cout<<"\n1  "<<sand1<<" Rs.240"<<"\n";
        cout<<"2  "<<sand2<<" Rs.160"<<"\n";
        cout<<"3  "<<sand3<<" Rs.100"<<"\n";
        cout<<"\nPlease Enter which Sandwich you would like to have?:";
        cin>>pchoice1;
        if(pchoice1>=1 && pchoice1<=3)
        {
            cout<<"\nPlease Enter Quantity: ";
            cin>>quantity;
            switch(pchoice1)
            {
            case 1: choice = 240*quantity;
            break;
            case 2: choice = 160*quantity;
            break;
            case 3: choice = 100*quantity;
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            switch (pchoice1)
            {
             case 1:
             bill(quantity,choice,(char *)"Club-Sandwich",username,password);
             break;
```

```
            break;
            case 2:
            bill(quantity,choice,(char *)"Chicken-Crispy-Sandwich",username,password);
            break;
            case 3:
            bill(quantity,choice,(char *)"Extream-Veg-Sandwich",username,password);
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            cout<<"Would you like to order anything else? Y / N:";
            cin>>gotostart;
            if(gotostart=='Y' || gotostart=='y')
            {
              goto beginning;
            }
            else
              return;
            }
            else
    {
        cout<<"Invalid Choice\n\n";
        goto beginning;
    }
            }
        else if(choice==4)
     {
        cout<<"\n1 "<<roll1<<" Rs.150"<<"\n";
        cout<<"2 "<<roll2<<" Rs.100"<<"\n";
        cout<<"3 "<<roll3<<" Rs.120"<<"\n";
        cout<<"\nPlease Enter which you would like to have?: ";
        cin>>pchoice1;
        if(pchoice1>=1 && pchoice1<=3)
        {
```

```
            cout<<"\nHow Much Rolls Do you want: ";
            cin>>quantity;
            switch(pchoice1)
            {
            case 1: choice = 150*quantity;
            break;
            case 2: choice = 100*quantity;
            break;
            case 3: choice = 120*quantity;
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            switch (pchoice1)
            {
             case 1:
             bill(quantity,choice,(char *)"Chicken-Chatni-Roll",username,password);
             break;
             case 2:
             bill(quantity,choice,(char *)"Chicken-Mayo-Roll",username,password);
             break;
             case 3:
             bill(quantity,choice,(char *)"Veg-Roll-With-Fries",username,password);
             break;
             default : cout<<"Invalid Choice\n\n";
            }
            cout<<"Would you like to order anything else? Y / N:";
            cin>>gotostart;
            if(gotostart=='Y' || gotostart=='y')
            {
              goto beginning;
```

```
            {
              goto beginning;
            }
            else
              return;
            }
            else
    {
        cout<<"Invalid Choice\n\n";
        goto beginning;
    }
            }
    else if(choice==5)
     {
        cout<<"\n1 "<<bir1<<" Rs.160"<<"\n";
        cout<<"2 "<<bir2<<" Rs.220"<<"\n";
        cout<<"3 "<<bir3<<" Rs.140"<<"\n";
        cout<<"\nPlease Enter which Biryani you would like to have?:";
        cin>>pchoice1;
        if(pchoice1>=1 && pchoice1<=3)
        {
            cout<<"\nPlease Enter Quantity: ";
            cin>>quantity;
            switch(pchoice1)
            {
            case 1: choice = 160*quantity;
            break;
            case 2: choice = 220*quantity;
            break;
```

```
            break;
            case 3: choice = 140*quantity;
            break;
            default : cout<<"Invalid Choice\n\n";
            }
            switch (pchoice1)
            {
             case 1:
             bill(quantity,choice,(char *)"Chicken-Biryani",username,password);
             break;
             case 2:
             bill(quantity,choice,(char *)"Prawn-Biryani",username,password);
             break;
             case 3:
             bill(quantity,choice,(char *)"Paneer-Biryani",username,password);
             break;
             default : cout<<"Invalid Choice\n\n";
            }
            cout<<"Would you like to order anything else? Y / N:";
            cin>>gotostart;
            if(gotostart=='Y' || gotostart=='y')
            {
              goto beginning;
            }
            else
              return;
            }
        }
```

```
            else
            {
                cout<<"Please Select Right Option: \n";
                cout<<"Would You like to Start the program again? Y / N: " ;
                cin>>gotostart;

                if(gotostart=='Y' || gotostart=='y')
                {
                  goto beginning;
                }
                else
                  return;
            }
    getch();
}

bool Order::validateLogin(char username[],char password[]) {
    fp.open("Customers.txt",ios::in);
    char buffer[143],temp[143];
    char *usr;
    char *pass;
    while(!fp.fail())
    {
        fp>>buffer;
        strcpy(temp,buffer);
        usr=strtok(temp,"|");
        pass=strtok(NULL,"|");
    if ( (strcmp(username,usr)==0) && (strcmp(password,pass)==0) ) {
        return true;
```

```
        return true;
    }
  }
    return false;
}

void Order::saveSignup(char username[], char password[], char address[], char phone[]) {
    char buffer[143];
    cout << "Signup successful! Username: " << username << endl;
    fp.open("Customers.txt",ios::app);
    strcpy(buffer,username);
    strcat(buffer,"|");
    strcat(buffer,password);
    strcat(buffer,"|");
    strcat(buffer,phone);
    strcat(buffer,"|");
    strcat(buffer,address);
    strcat(buffer,"|");
    fp << buffer;
    fp << "\n";
    fp.close();
}

int main()
{
    Order O1;
    char loginpage;
    loginpage:
    cout<<"\n\t\t\t----------Welcome to Peter's Jr. Fast Food----------\n\n";
    cout<<"Press 1> To Login\n";
    cout<<"Press 2> To Sign-up\n";
    cout<<"Press 3> To Exit\n";
    cout<<"Please enter your choice....  ";
    int ch;
```

```
    int ch;
    cin>>ch;
    if(ch==1)
    {
        cout << "Login System\n";
        cout<<"Username: ";
        cin>>username;
        cout<<"Password: ";
        cin>>password;
        if(O1.validateLogin(username,password))
        {
            int choice;
            cout << "Login successful!\n";
            cout << "Press 1 to view your previous order or Press 2 to place a new order\n";
            cin>>choice;
            if(choice==1)
            {
                O1.previousOrder(username,password);
            }
            else if(choice==2)
            {
                O1.orderItem(username,password);
            }
            else
            cout<<"Invalid input\n";
        }
        else
        {
        cout<<"Invalid username or password. Login failed.\n";
        cout<<"Please try again or Register as a new customer";
        goto loginpage;
```

```
        goto loginpage;
        }
    }
    else if(ch==2)
    {
        cout<<"Signup System\n";
        cout<<"Username: ";
        cin>>username;
        cout<<"Password: ";
        cin>>password;
        cout<<"Phone: ";
        cin>>phone;
        cout<<"Delivery address: ";
        cin>>address;
        O1.saveSignup(username, password, address, phone);
        O1.orderItem(username,password);
    }
    else
    {
        cout<<"Exiting.....\n";
        return 0;
    }
}
```

Chapter 5

Testing

Software testing is the stage of implementation, which is aimed at ensuring that the software works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and a complete verification to determine whether the objectives are met, and the user requirements are satisfied. The aim is quality assurance. Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies, a test plan is carried out on each module. The various tests performed are unit testing, integration testing and user acceptance testing.

5.1 UNIT TESTING

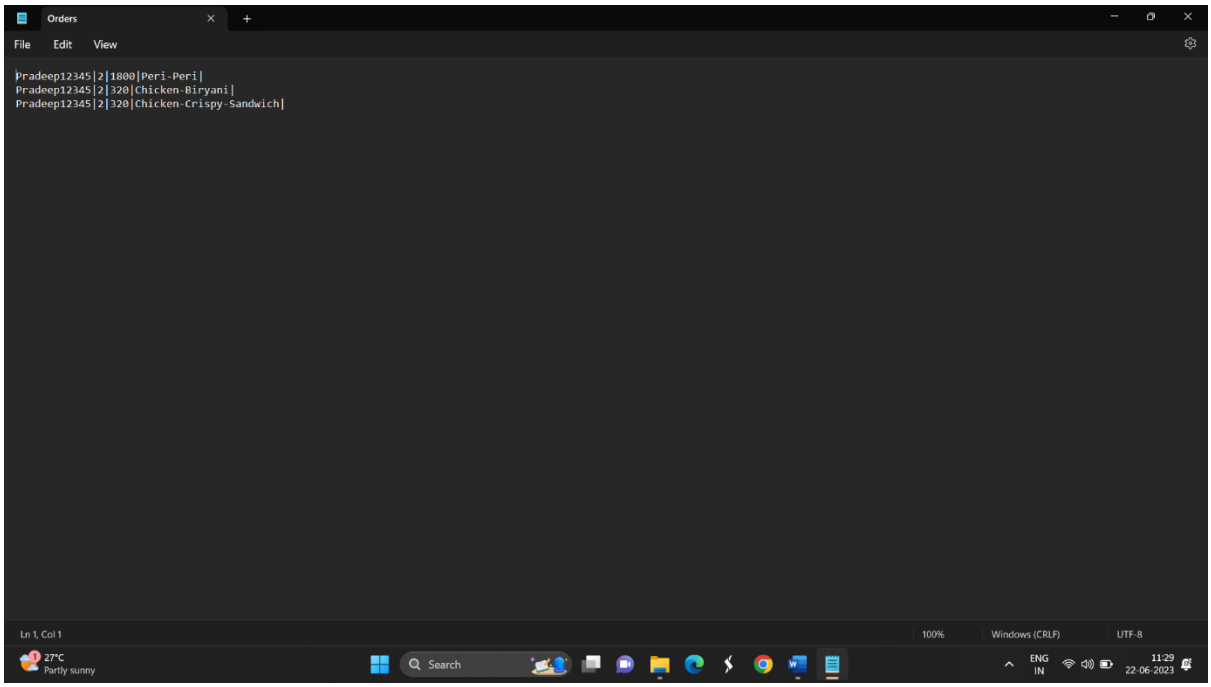
Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing. The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables us to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

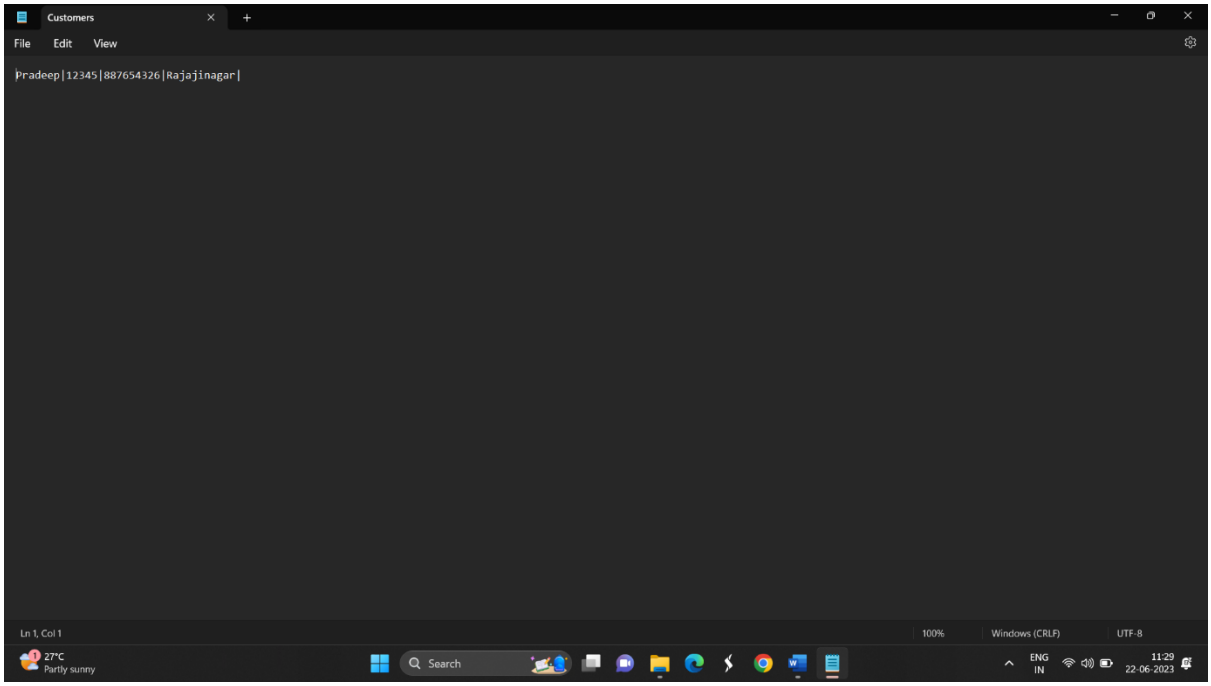
Sl.No	Description(Input)	Output	Actual Output	Status
1	As the sign-up option is selected, new user gets entered in the customer's file.	Insertion of user successful	Insertion of user successful	Pass
2	As the login option is selected, the user details are taken in and verified.	Verification of the user credentials is successful	Existing user's verification was done successfully	Pass
3	As the customer places the order, the order gets stored in the Orders file.	Storing of the orders is done successfully	Storing of the orders in the Order's file was done successfully	Pass
4	Searching for the previous orders of the current user.	Retrieval of all the orders placed by the current user was done successfully	Retrieval of the orders was done successfully	Pass

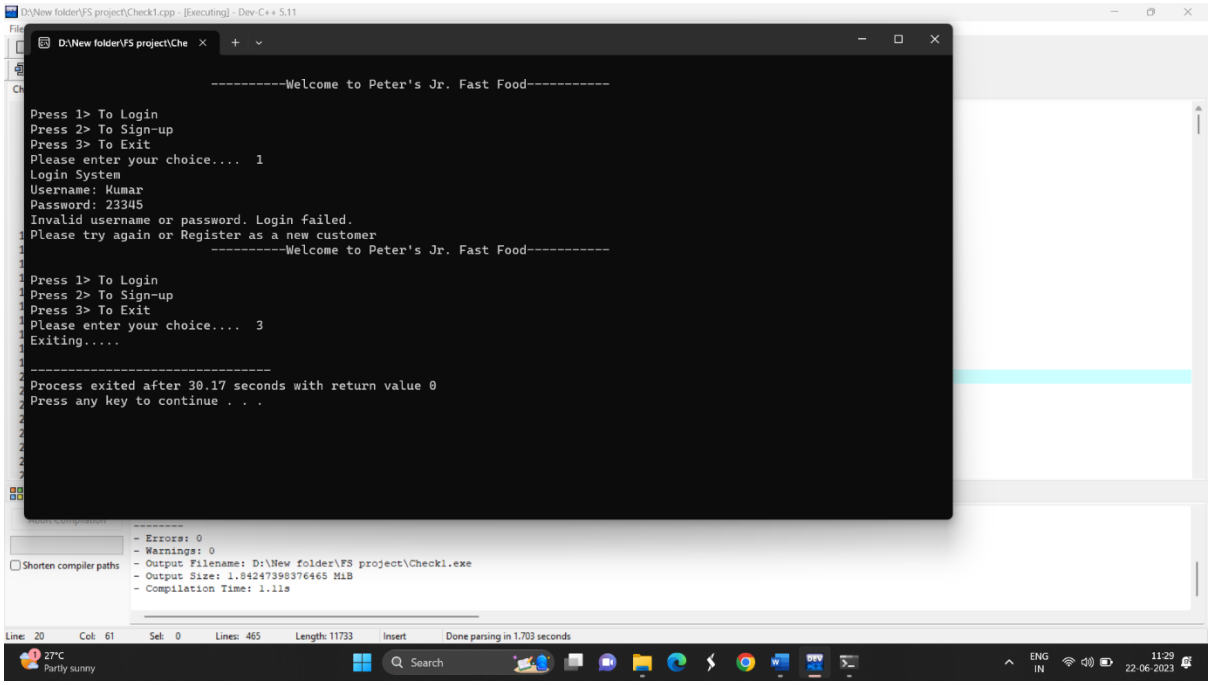
Table 5.1 Test cases

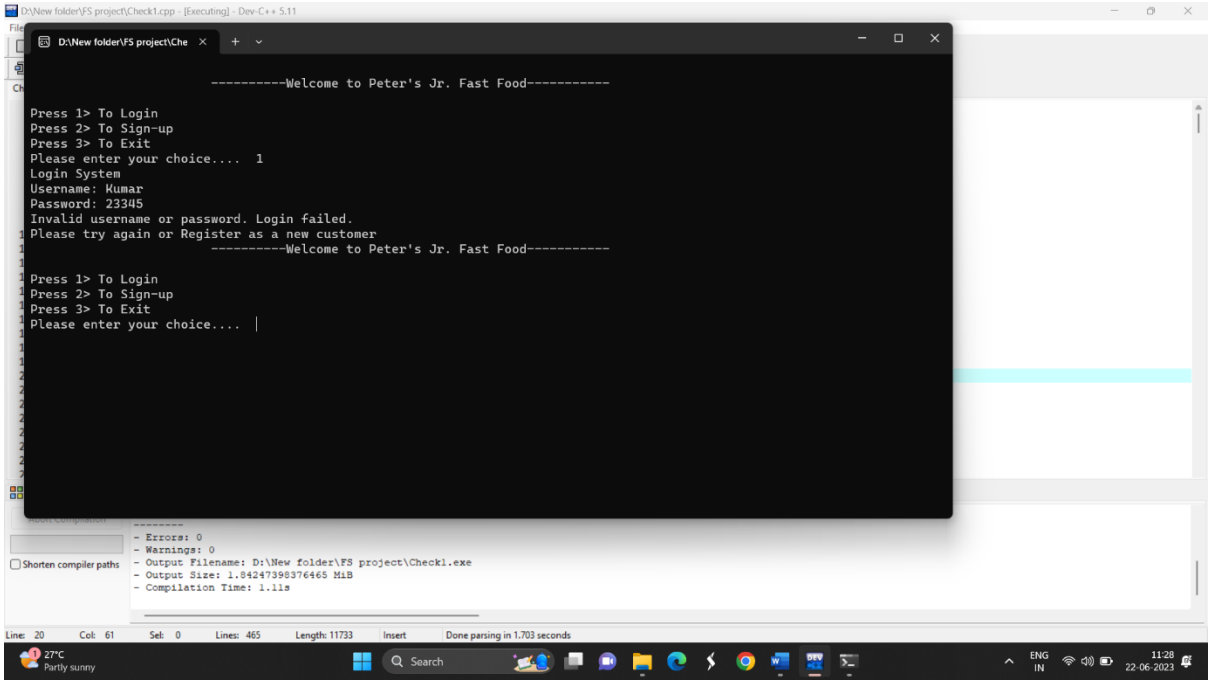
Chapter 6

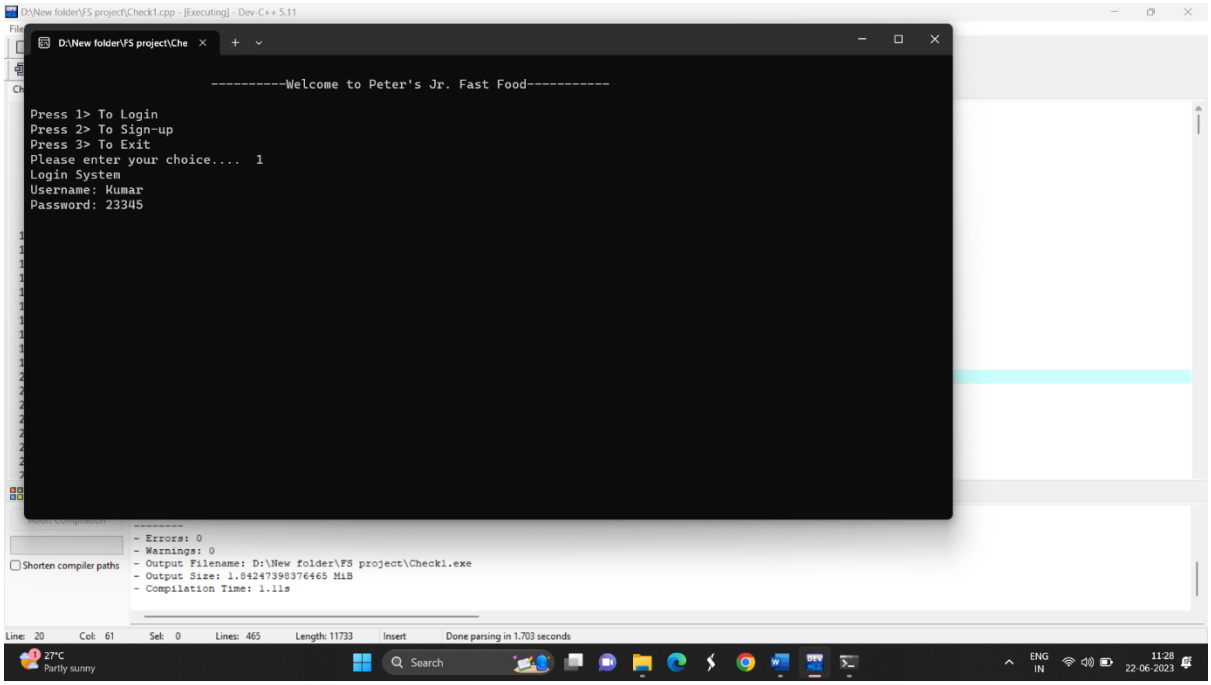
Result

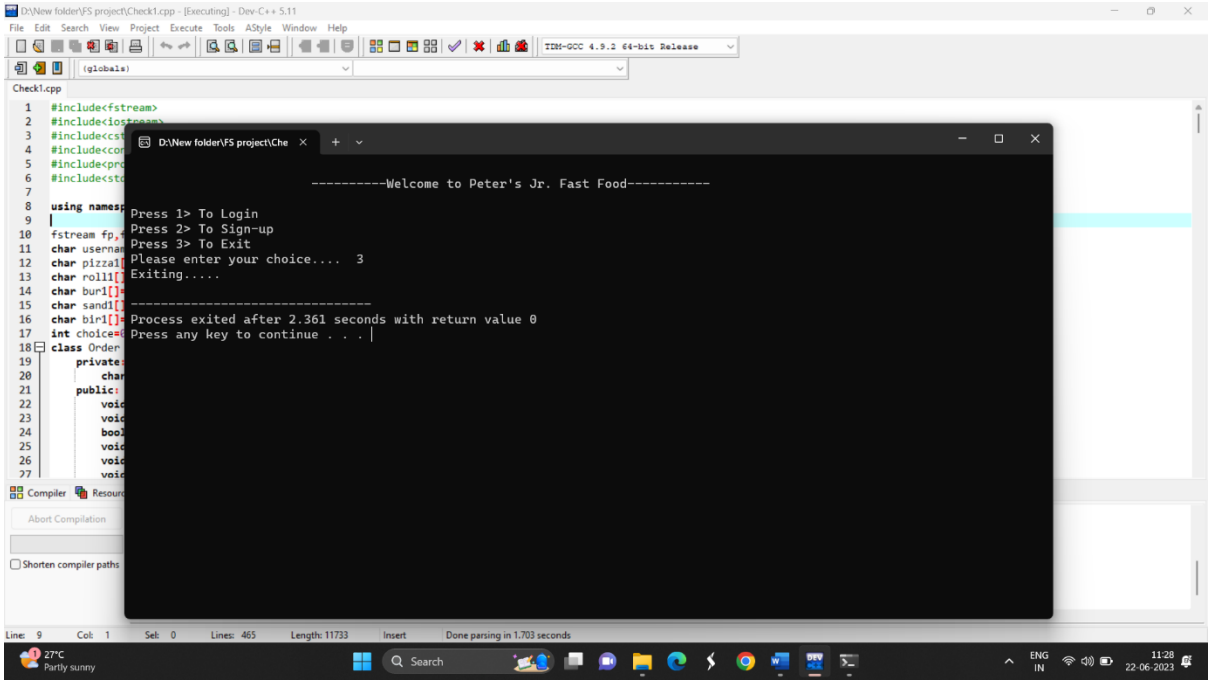


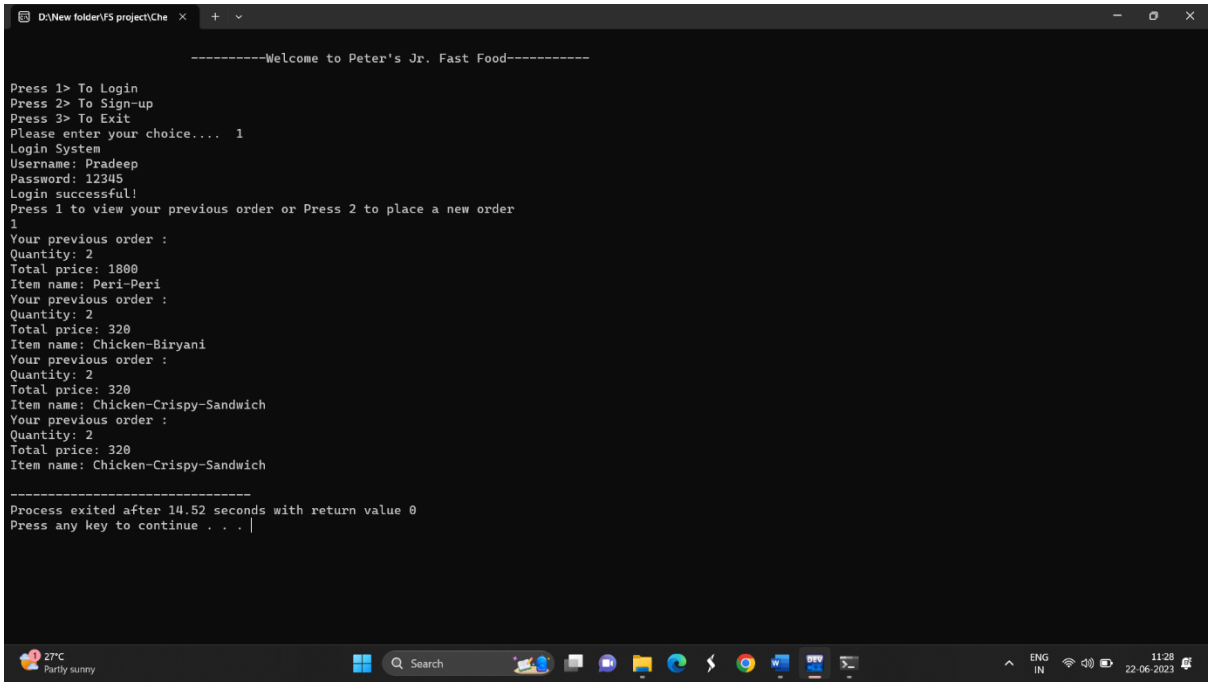


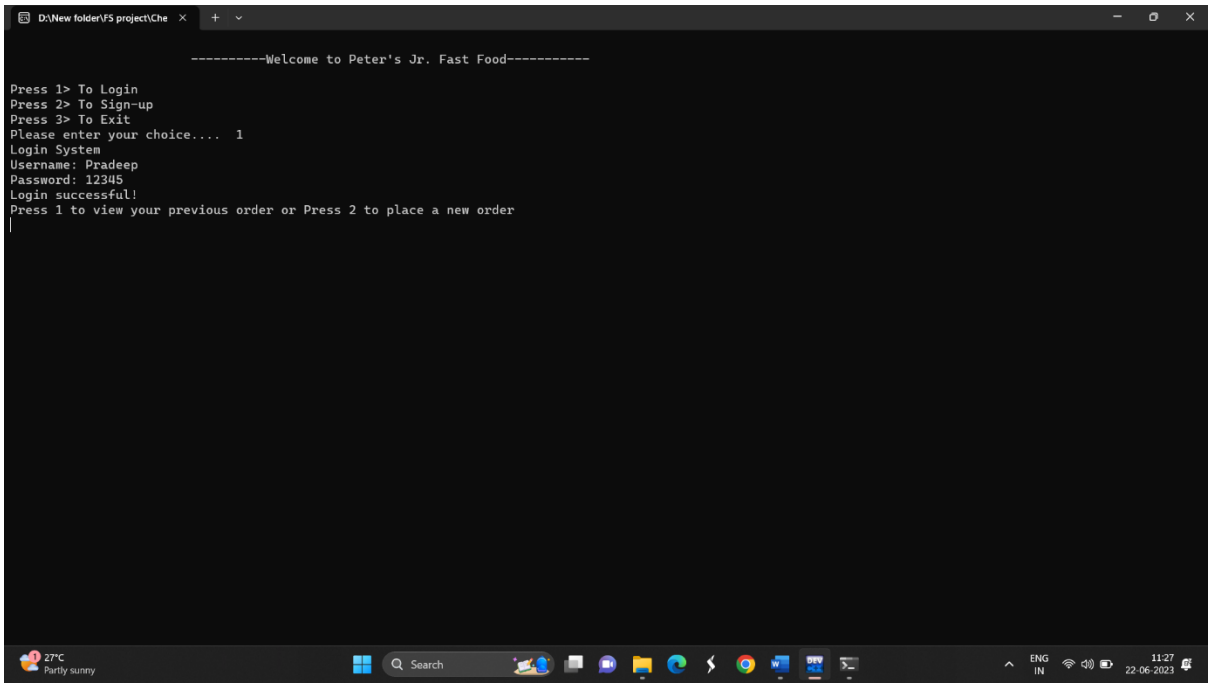

----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Kumar
Password: 23345
Invalid username or password. Login failed.
Please try again or Register as a new customer
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 3
Exiting.....
Process exited after 30.17 seconds with return value 0
Press any key to continue . . .
- Errors: 0
- Warnings: 0
- Output Filename: D:\New folder\FS project\Check1.exe
- Output Size: 1.842473983764 65 MiB
- Compilation Time: 1.11s

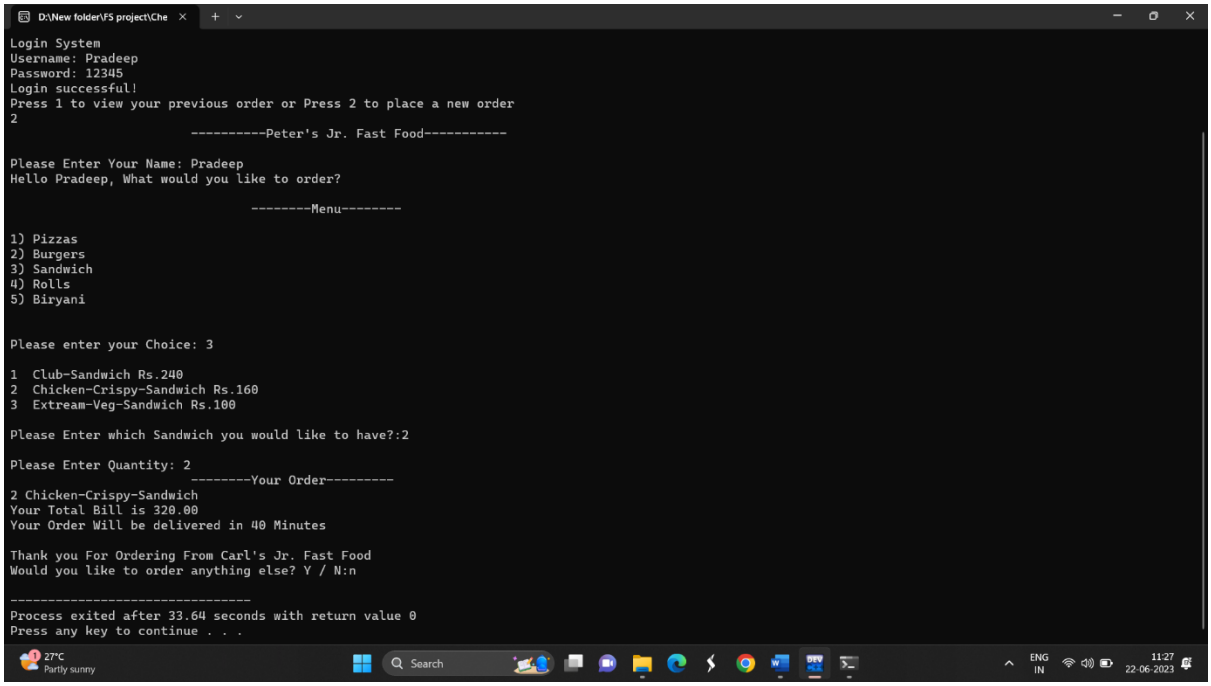

----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Kumar
Password: 23345
Invalid username or password. Login failed.
Please try again or Register as a new customer
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice....
- Errors: 0
- Warnings: 0
- Output Filename: D:\New folder\FS project\Check1.exe
- Output Size: 1.842473983764 65 MiB
- Compilation Time: 1.11s

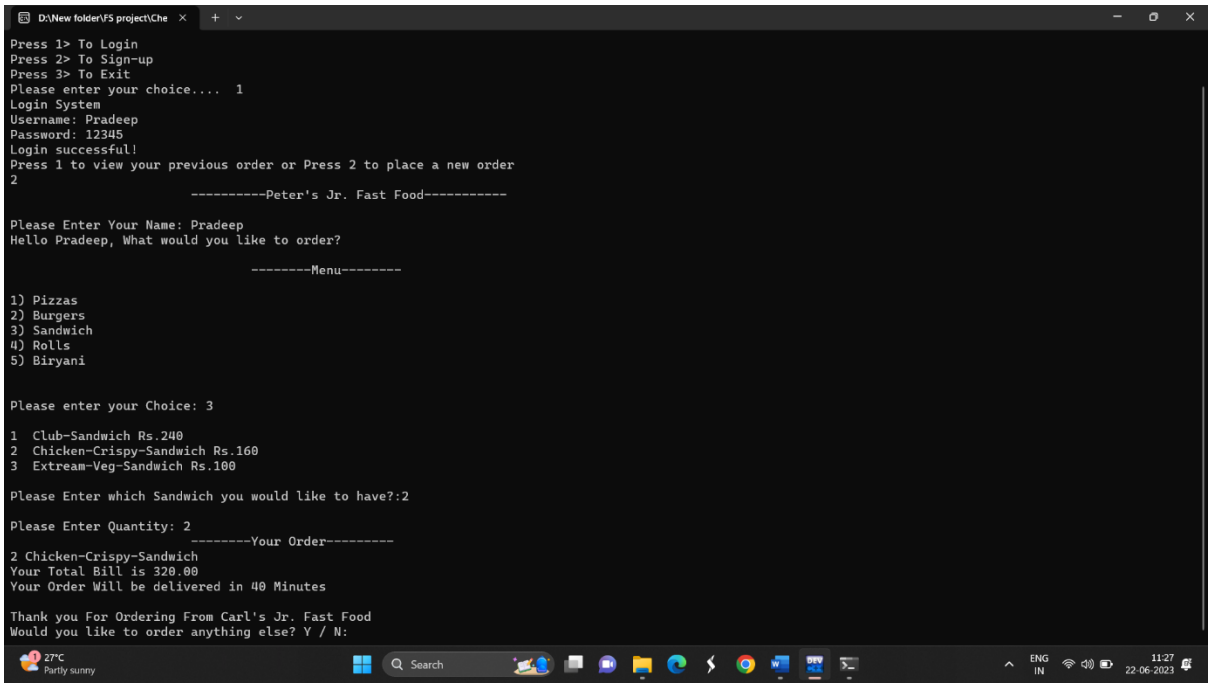

D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Kumar
Password: 23345
- Errors: 0
- Warnings: 0
- Output Filename: D:\New folder\FS project\Check1.exe
- Output Size: 1.84247398376465 MiB
- Compilation Time: 1.11s

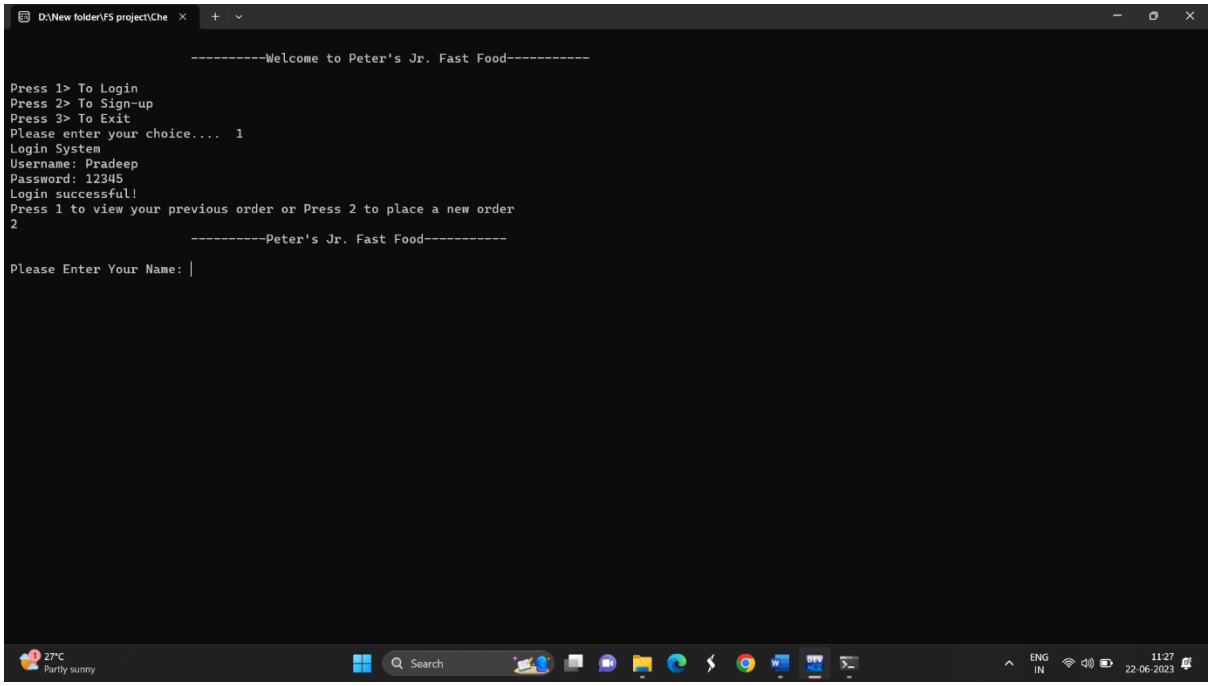

Check1.cpp
D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 3
Exiting.....
Process exited after 2.361 seconds with return value 0
Press any key to continue . . .

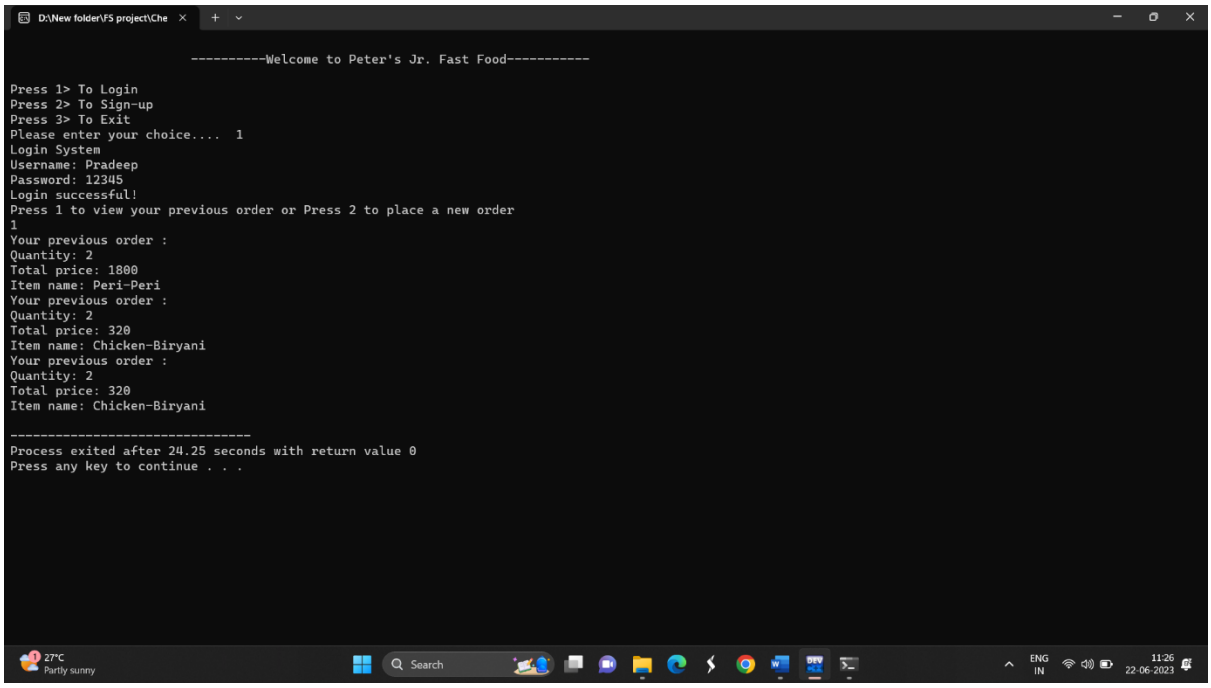

D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
1
Your previous order :
Quantity: 2
Total price: 1800
Item name: Peri-Peri
Your previous order :
Quantity: 2
Total price: 320
Item name: Chicken-Biryani
Your previous order :
Quantity: 2
Total price: 320
Item name: Chicken-Crispy-Sandwich
Your previous order :
Quantity: 2
Total price: 320
Item name: Chicken-Crispy-Sandwich
Process exited after 14.52 seconds with return value 0
Press any key to continue . . .
27°C
Partly sunny
Search
ENG
IN
11:28
22-06-2023


D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
27°C
Partly sunny
Search
ENG
IN
11:27
22-06-2023

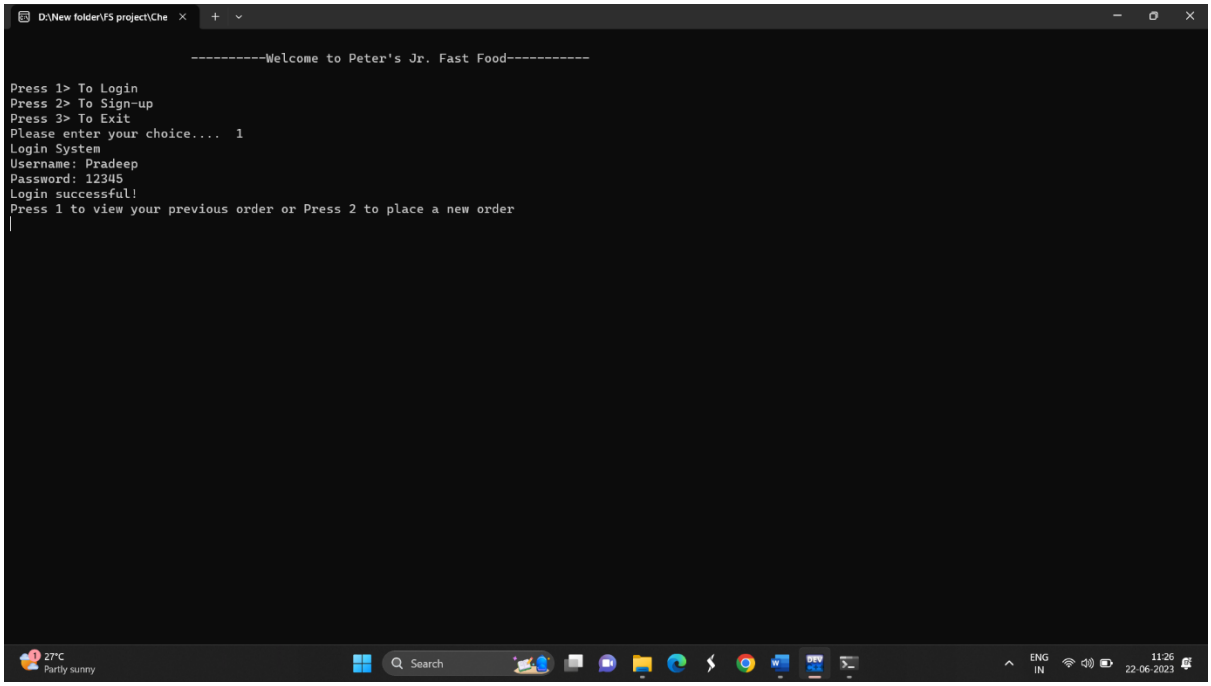

D:\New folder\FS project\Che
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
2
----------Peter's Jr. Fast Food----------
Please Enter Your Name: Pradeep
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice: 3
1 Club-Sandwich Rs.240
2 Chicken-Crispy-Sandwich Rs.160
3 Extream-Veg-Sandwich Rs.100
Please Enter which Sandwich you would like to have?:2
Please Enter Quantity: 2
--------Your Order---------
2 Chicken-Crispy-Sandwich
Your Total Bill is 320.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:n
Process exited after 33.64 seconds with return value 0
Press any key to continue . . .
27°C
Partly sunny
Search
ENG
IN
11:27
22-06-2023

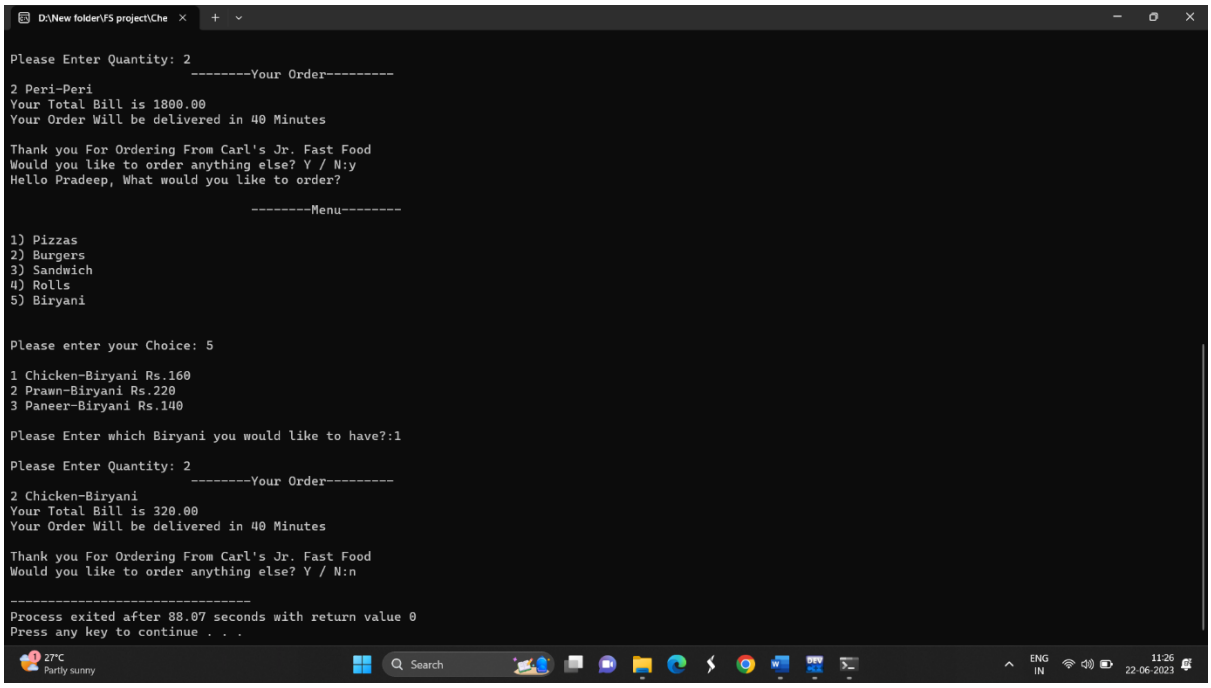

D:\New folder\FS project\Che
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
2
----------Peter's Jr. Fast Food----------
Please Enter Your Name: Pradeep
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice: 3
1 Club-Sandwich Rs.240
2 Chicken-Crispy-Sandwich Rs.160
3 Extream-Veg-Sandwich Rs.100
Please Enter which Sandwich you would like to have?:2
Please Enter Quantity: 2
--------Your Order---------
2 Chicken-Crispy-Sandwich
Your Total Bill is 320.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:
27°C
Partly sunny
Search
ENG
IN
11:27
22-06-2023

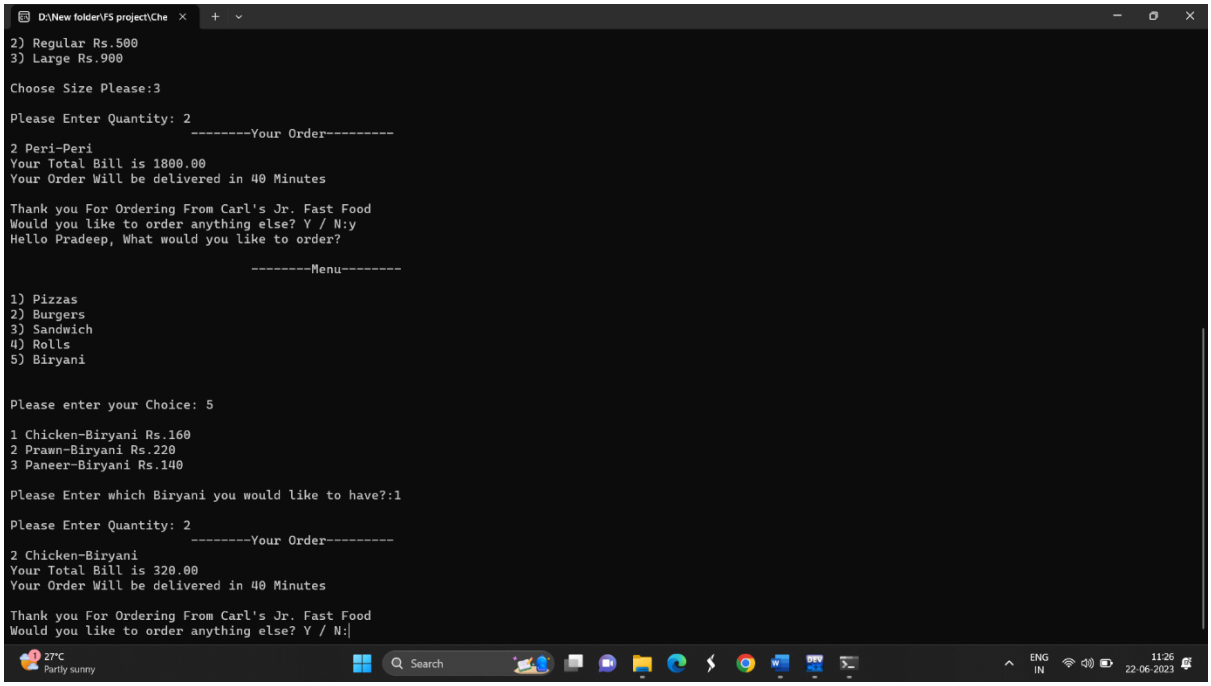

D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
2
----------Peter's Jr. Fast Food----------
Please Enter Your Name:
27°C
Partly sunny
Search
ENG
IN
11:27
22-06-2023

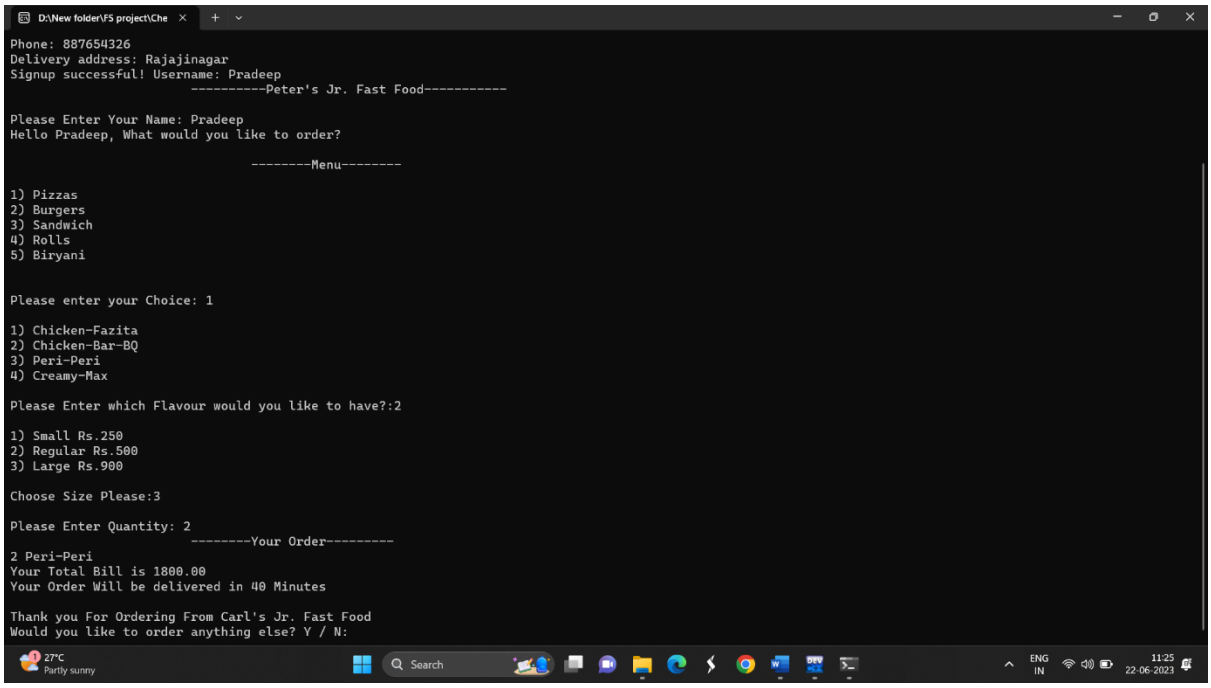

D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
1
Your previous order :
Quantity: 2
Total price: 1800
Item name: Peri-Peri
Your previous order :
Quantity: 2
Total price: 320
Item name: Chicken-Biryani
Your previous order :
Quantity: 2
Total price: 320
Item name: Chicken-Biryani

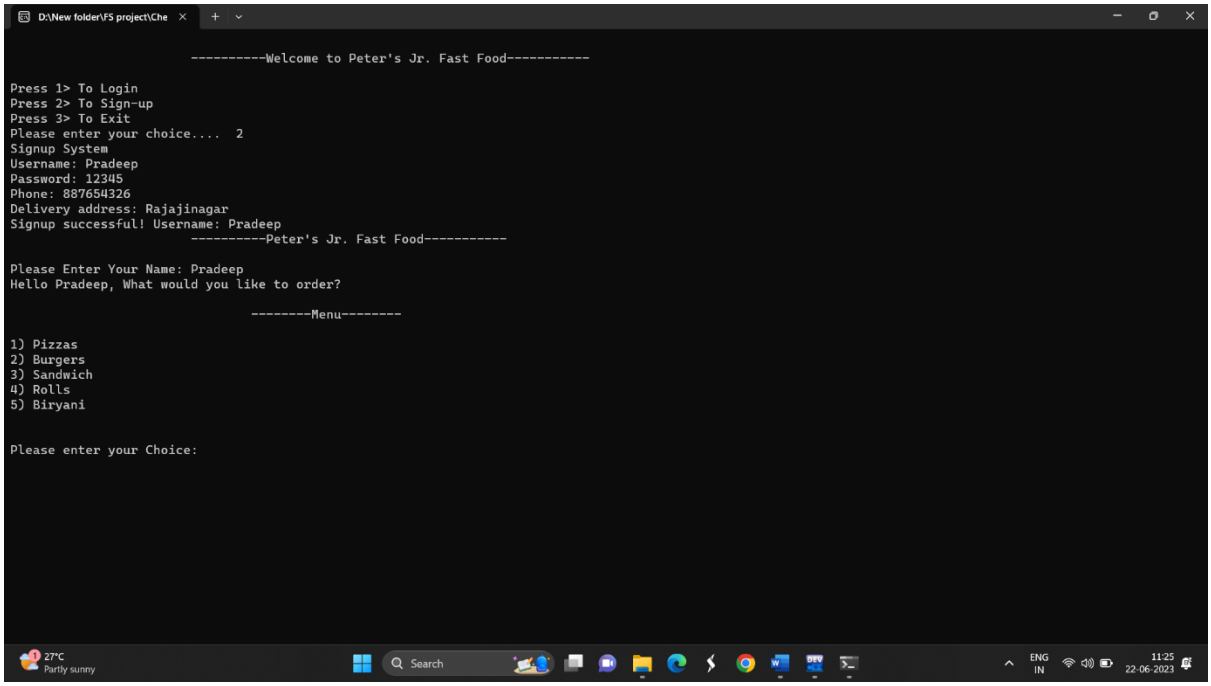
Process exited after 24.25 seconds with return value 0
Press any key to continue . . .
27°C
Partly sunny
Search
ENG
IN
11:26
22-06-2023

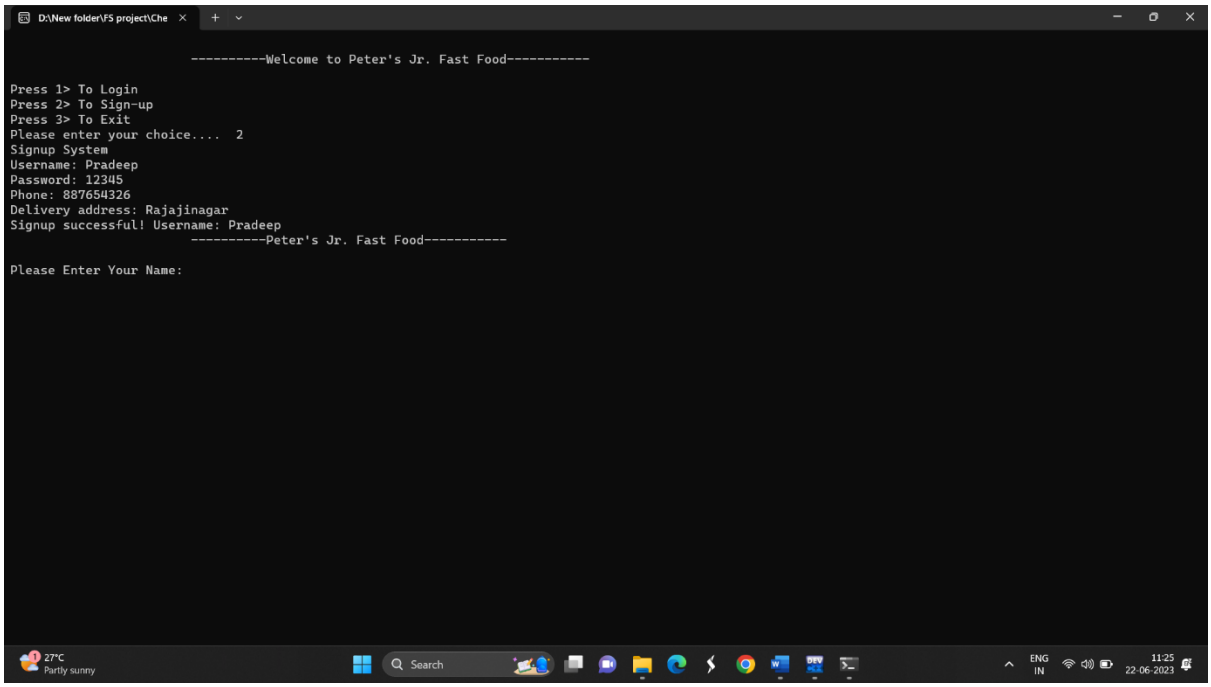

D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 1
Login System
Username: Pradeep
Password: 12345
Login successful!
Press 1 to view your previous order or Press 2 to place a new order
27°C
Partly sunny
Search
ENG
IN
11:26
22-06-2023

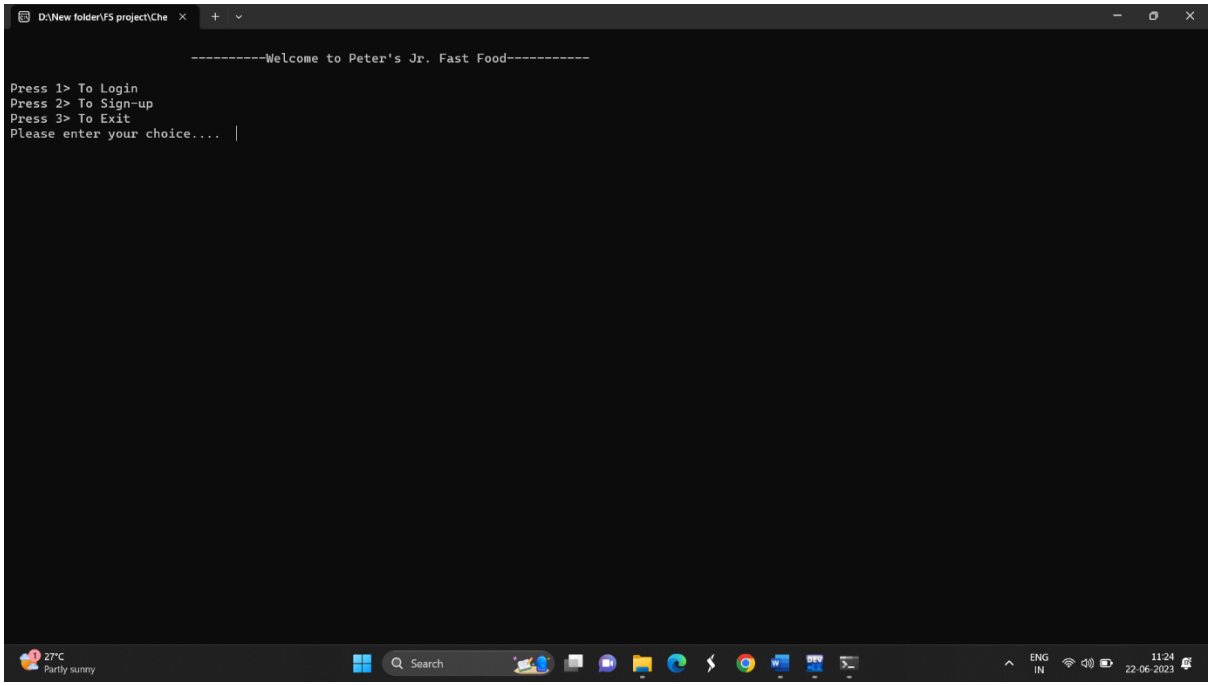

D:\New folder\FS project\Che
Please Enter Quantity: 2
--------Your Order---------
2 Peri-Peri
Your Total Bill is 1800.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:y
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice: 5
1 Chicken-Biryani Rs.160
2 Prawn-Biryani Rs.220
3 Paneer-Biryani Rs.140
Please Enter which Biryani you would like to have?:1
Please Enter Quantity: 2
--------Your Order---------
2 Chicken-Biryani
Your Total Bill is 320.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:n
Process exited after 88.07 seconds with return value 0
Press any key to continue . . .
27°C
Partly sunny
Search
ENG
IN
11:26
22-06-2023


D:\New folder\FS project\Che
2) Regular Rs.500
3) Large Rs.900
Choose Size Please:3
Please Enter Quantity: 2
--------Your Order---------
2 Peri-Peri
Your Total Bill is 1800.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:y
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice: 5
1 Chicken-Biryani Rs.160
2 Prawn-Biryani Rs.220
3 Paneer-Biryani Rs.140
Please Enter which Biryani you would like to have?:1
Please Enter Quantity: 2
--------Your Order---------
2 Chicken-Biryani
Your Total Bill is 320.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:
27°C
Partly sunny
Search
ENG
IN
11:26
22-06-2023


D:\New folder\FS project\Che
Phone: 887654326
Delivery address: Rajajinagar
Signup successful! Username: Pradeep
----------Peter's Jr. Fast Food----------
Please Enter Your Name: Pradeep
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice: 1
1) Chicken-Fazita
2) Chicken-Bar-BQ
3) Peri-Peri
4) Creamy-Max
Please Enter which Flavour would you like to have?:2
1) Small Rs.250
2) Regular Rs.500
3) Large Rs.900
Choose Size Please:3
Please Enter Quantity: 2
--------Your Order---------
2 Peri-Peri
Your Total Bill is 1800.00
Your Order Will be delivered in 40 Minutes
Thank you For Ordering From Carl's Jr. Fast Food
Would you like to order anything else? Y / N:
27°C
Partly sunny
Search
ENG
IN
11:25
22-06-2023


D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 2
Signup System
Username: Pradeep
Password: 12345
Phone: 887654326
Delivery address: Rajajinagar
Signup successful! Username: Pradeep
----------Peter's Jr. Fast Food----------
Please Enter Your Name: Pradeep
Hello Pradeep, What would you like to order?
--------Menu--------
1) Pizzas
2) Burgers
3) Sandwich
4) Rolls
5) Biryani
Please enter your Choice:
27°C
Partly sunny
Search
ENG
IN
11:25
22-06-2023


D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice.... 2
Signup System
Username: Pradeep
Password: 12345
Phone: 887654326
Delivery address: Rajajinagar
Signup successful! Username: Pradeep
----------Peter's Jr. Fast Food----------
Please Enter Your Name:
27°C
Partly sunny
Search
ENG
IN
11:25
22-06-2023


D:\New folder\FS project\Che
----------Welcome to Peter's Jr. Fast Food----------
Press 1> To Login
Press 2> To Sign-up
Press 3> To Exit
Please enter your choice....
27°C
Partly sunny
Search
ENG
IN
11:24
22-06-2023

Conclusion and Future Enhancements

Conclusion

Food Ordering Management System has been successfully developed using Dev C++ under Windows platform.

The basic idea of this project was to allow users to order their food in a hassle-free environment. It provides customers with a bill and details of the order and an access to their previous orders, which makes this application a convenient platform to place the orders.

Future Enhancements

The further development could be made in the following way,

- Add more restaurants and dishes to choose from.
- Providing a real-time view on the progress of delivery.
- Providing a way to cancel the order.
- Provide a payment mechanism other than COD.
- Allow users to add in some comments to their order which could help the chefs and the delivery person in understanding the needs of the customer.

REFERENCES

BOOKS

• File Structures: An Object-Oriented Approach with C++ 3rd Edition by Michael J. Folk (Author), BillZoellick (Author), Greg Riccardi (Author).

• C Projects by Yashavant Kanetkar.

• Programming Projects in C for Students of Engineering, Science, and Mathematics by RoubenRostamian.

ONLINE WEBSITES

• For Data Flow Diagram and Flowchart, Lucid chart is a web-based commercial service to createflowcharts, organizational charts, website wireframes, and other things.

• [https://www.lucidchart.com/documents/edit/6830d573-57de-4424- 83b2-660f3108bd9b]- for DataFlow Diagram

• www.stackoverflow.com/files/

• C Programming Tutorial by Mike Dane at freeCodeCamp – [https://www.youtube.com/watch?v=KJgsSFOSQv0]