# ALIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY ALIGARH



(AFFILIATED BY A.I.C.T.E & AFFILIATED TO A.K.T.U, LUCKNOW)

2018-2022

PROJECT REPORT ON

“CAFÉ BILLING SYSTEM”

SUBMITTIED IN PARTIAL FULFILLMENT OF THE REQUIRMENT FOR THE

AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

DIVYANSHU VARSHNEY

(1810910025)

UNDER THE GUIDANCE

Mr. Ritesham Shastri

Dr. ANAND SHARMA

HOD (CSE)

**DECLARATION**

I do hereby declare that this project work entilled “CAFÉ BILLING SYSTEM” submitted by me for the partial fulfilment of the requirement for the award of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING (B. TECH) is a record of my own work and that, to the best of our knowledge and belief.

The report embodies the finding based on my study and observation.

Date: Name: Divyanshu Varshney

Roll no: 1810910025

# **CERTIFICATE**

# This is certify that the project report entitled “\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

# Submitted in partial fulfilment of the degree of Bachelor of Technology from **Dr. A.P.J Abdul Kalam Technical University,** **Lucknow** is an authentic and original work carried out by Mr. / Mrs. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with Roll no. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ under my guidance.

# The matter embodied in this project is genuine work done by the student and has not been submitted whether to this university or to any other university/institute for the fulfilment of the requirements of any course of study

# **ACKNOWLEDGEMENT**

# The project report on “**CAFÉ BILLING SYSTEM**” needed co-operation and guidance of a number of people. We therefore consider it our prime duty to thank all those who helped us through this venture.

With extreme happiness, we would like to thank all the helping hand which helped us to complete this project. We express our immense gratitude and appreciation to our supervisor, **Asst. Prof. Mr. Sonu Sharma** for his personal interest, support and encouragement for completion of this project. we extend our immense gratitude to Asst. **Prof. Anand Sharma** for his motivational and encouragement for our project.

We express our sincere thanks to the **Dr. Anand Sharma, Head** **of Department**, Computer science & engineering for their kind co-operation.

I would like to express my special gratitude and thanks to **Mr**. **Ritesham Shastri**, Managing director, Rapidcode Technology [p] Ltd. For giving me such attention & time.

We are also thankful to all the helping hands that were there to help and guide us in achieving the right path leading to the completion of this project.

Date: Divyanshu varshney

ACET, Aligarh

**ABSTRACT**

The main objective of this project is to generate a bill for café to maintain information of the customer. This project is aimed to developing an online café billing system.

The entire project has been developed keeping in view of owner-customer relationship, in mind.

This project used to create the bill for a selected item. Through this system admin can maintain the customer details and see the café item data to generate a bill.

# 

# **TABLE OF CONTENT**

# Introduction

# Project category Types of Report

# Technologies and tools

# Hardware

# Future scope Analysis of Present system Feasibility Analysis

# Data flow diagram

# Entity Relation diagram

Input code

Output Screenshot

Bibliography

**INTRODUCTION**

The project “Billing system” is a software to semi-automate the process of ordering and billing of a “dark-eat cafe”. This software is designed considering the dark eat cafe. Billing system may be manual, electronic, computerized or point of sale (pos). point of sale or point of service is a location where a transaction occurs. Point of sale system are used in café, supermarket, restaurant, hotels, as well as almost any type of retail establishment.

# **PROJECT CATEGORY**

**PROGRAMMING LANGUAGE - C:**

The project is based on the concepts of c, c is a procedural programming language. It was initially developed by Dennis Ritchie between 1969 and 1973. It was mainly developed as a system programming language to write an operating system. The main features of C language include low-level access to memory, a simple set of keywords, and clean style, these features make C language suitable for system programming’s like an operating system or compiler development.

Many later languages have borrowed syntax/features directly or indirectly from C language. Like syntax of Java, PHP, JavaScript and many other languages are mainly based on C language. C++ is nearly a superset of C language (There are few programs that may compile in C, but not in C++).

# **TYPES OF REPORT**

# 

# The following are the type of reports that can be generated using the designed system.

# 1.Daily Sales Report

# 2.Monthly Customer Report

# 3.Due date Report (report of a particular day)

# 4.Billing Report

# 

# **TECHNOLOGIES AND TOOLS**

Software Used:

Web browser: Google Chrome

Languages Used: C

Red hat enterprise LINUX (for compiling and executing the programs)

Hardware Used:

CPU configuration

Pentium IV processor

RAM 512 MB

Monitor:

Any monitor

Operating System:

UNIX/LINUX

# 

# **GDB-DEBUGGER**

# The GNU Debugger, usually called just GDB and named gdb as an executable file, is the standard debugger for the GNU software system. It is a portable debugger that works for many programming languages, including Ada, C, C++, Objective-C, and Pascal. It allows you to see what is going on `inside' another program while it executes -- or what another program was doing at the moment it crashed. GNU Debugger helps you in finding out followings:

# 1) If a core dump happened then what statement or expression did the program crash on?

# 2) If an error occurs while executing a function, what line of the program contains the call to that function, and what are the parameters?

# 3) What are the values of program variables at a particular point during execution of the program?

# 4) What is the result of a particular expression in a program?

# 

# 

# **GIT**

# Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows**.**

Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.[12] Its current maintainer since 2005 is Junio Hamano. As with most other distributed version-control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version-tracking abilities, independent of network access or a central server.[13] Git is free and open-source software distributed under the terms of the GNU General Public License version 2.

# **ANALYSIS OF PRESENT SYSTEM**

# Before we begin a new system, it is important to study the system that will be improved or replaced (if there is one). We need to analyse how this system uses hardware, software, network and the people resources to convert data resources, such as transaction data, into information products, such as reports and displays. Thus, we should document how the information system activities of input, processing, output, storage and control are accomplished.

# 

# 

# **FEASIBILITY ANALYSIS**

Title:

Feasibility report for the computerization of the various activities of the company.

Background:

The company facing the problem of inconsistent and out of time information in its activities. Very much time is consuming for report generation, which is not very helpful for decision making. So, we want a system, which provide immediate information.

Method of study:

The analysis procedure comprised of field trips in the various departments of the company. The following documents and sources were looked up:

· The purchase order that contain items to be purchased.

· The accounts register.

· Purchase order issues to vendors.

· Bills receive from vendors. · Bills give to the customers.

· Purchase return forms (if any) give to vendors.

Need for Feasibility Study:

The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its work ability, its impact on the organization ability to meet user needs and effective use of resources. It is usually carried out by a small number of people who are familiar with the information system techniques, understand the part of the business or organization that will be involved or effected by the project and are skilled in the sThe key consideration involve in the feasibility study are:

1. Technical
2. Behavioural

1. Technical Feasibility

Technical feasibility centres on the existing computer system (hardware, Software etc) and to what extent it can support the proposed system

Addition. For example, if the current system is operating at 70% capacity (an Arbitrary value), then another application could overload the system or require additional hardware. If the budget is serious constrain then the project is judged not feasible.

The technologies ant the environment which are used in this project are

SOFTWARE

Front End

1. Language used: C

Operating System:

Platform: UNIX/LINUX

Hardware:

Intel based processor-run computer system, which have keyboard as input devices. This has been decided for its case of availability and up-gradation.

The various registers maintained at the different department have

enough information recording, which will help in digitizing the available data.

2. Behavioural Feasibility:

An evaluation of the behaviour of the end users, which may effect the Envelopment of the system. People are inherently resistant to change and Computers have to know to facilitate changes and computers have to know to facilitate changes. An estimate should be made of how strong a reaction the user staffs is likely to have towards the development of a computerized System. It is a common knowledge that a computer installation has something to do with turnover, transfer, retraining and changes in employee job status, therefore the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of conducting business. The personal of the user organization will be affected by the proposed system. As the aim of the system is only to satisfy the information needs, no employees will lose their position by the proposed system. In fact, the proposed system will help the organization in reducing the voluminous work involved. Also, the involvement of users in every stage of the project is going to increase the success factor.

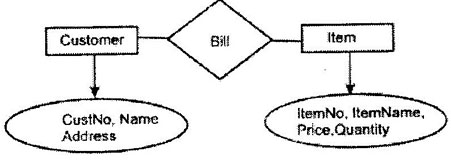
The staff in not well educated for running a computerized system. They are adamant in perceiving a mechanical process of working as they have long been used to the manual entry system. This aspect needs considerable amount of attention.

Our system is also feasible for organization because it supports of the organization and its strategic plan.

# **DATA FLOW DIAGRAM**

A data flow diagram is graphical representation that depicts the information flow and the transforms that r applied as date moves from input to output. It can be used to represent a software at any level of abstraction. In fact, DFDs may be partitioned in to levels. That represents increasing information flow and functional details.

DFDs are defined in levels with every level decreasing the level of abstraction as well as defining a greater detail of the functional organs of the system. A zero level DFD also known as context or fundamental system model represents the entire software elements as a single bubble with input and output data entities which are indicated as incoming and outgoing arrows. Data Flow Diagram help understanding the basic flow of data from one process to another process.



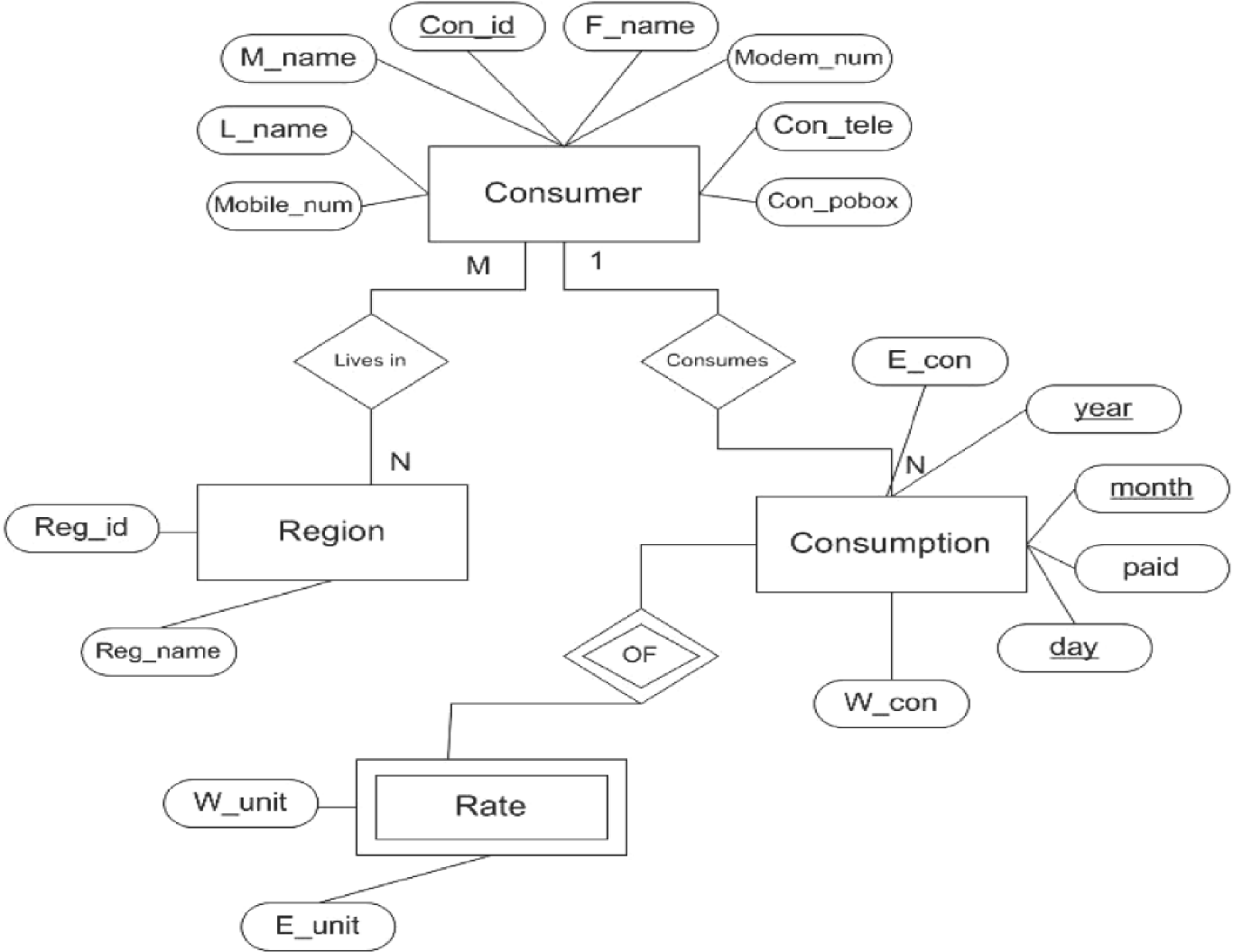
# **ENTITY RELATION DIAGRAM**

Entity Relation Diagram represents the object relationship pairs in graphical forms thus we can say that the primary goal of ER diagrams is represent data objects along with their relationships.

ER model for data uses three features to describe data:

· Entities which satisfy distinct real-world items in an application · Relationships connecting different entities and representing meaningful dependencies between them

· Attributes which specify various properties of entities and relations involved in a system.



# **PROGRAMMING CODE**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DARKEAT - CAFE BILLING SYSTEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//author- Divyanshu Varshney

//date- 20/07/2019

//copyright- All right reserved

//mentor - Ritesham Shastri & Abhishek Nair

#include<stdio.h>

#include<ctype.h>

void detail (double\*, char\*);

int amt;

// login section of cafe

void login ()

{

int a=0;

char uname [10];

char pword [10], code [10];

char user [10] ="darkeat";

char pass [10] ="darkeat";

{

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t\t-----------------------------------------------------------------------------------");

printf("\n\t\t\t \*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-DARK EAT CAFETERIA-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*");

printf ("\n \t\t\t ------------------------ ALIGARH --------------------------");

printf ("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ONCE EAT NEVER FORGET\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf ("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ONLY ONE IN WORLD\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf ("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COLLEGE DROP-OUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ENTERPRENEUR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t\t-----------------------------------------------------------------------------------");

printf("\n\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* login section\*\*\*\*\*\*\*\*\*\*\t\t\t\t\n");

printf ("enter the user name: ");

scanf ("%s”, uname);

printf ("\n enter the password: ");

scanf ("%s”, pword);

if (strcmp (uname, “darkeat”) ==0 && strcmp (pword, “darkeat”) ==0)

{

printf ("\n---------------- login successfully-----------!!! \n\n\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO DARK EAT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

else

{

printf("\n-----------sorry, login unsuccessfully----------");

exit (200);

}

}

}

void bill ()

{

int temp, ch, che, quantity;

FILE \*fptr, \*order;

fptr=fopen("menu.txt","r+");

order=fopen("bill.txt","w+");

if (! fptr)

{

printf("\cannot! open file");

// exit (1);

}

while((ch=fgetc(fptr))! =EOF)

printf("%c”, ch);

fprintf (order," ITEMS QUANTITY PRICE TOTAL");

while (1)

{

printf("enter the food item code\n");

scanf("%d",&che);

if(che>=61)

{

printf ("final bill");

break;

}

printf("enter the quantity");

scanf ("%d”, &quantity);

switch(che)

{

case 1:

printf ("\n\tSMOKE COOKIES\n");

temp=120\*quantity;

amt=amt+120\*quantity;

fprintf (order,"\n\t\t\smoke COOKOES: %d \* 120 = %d”, quantity, temp);

break;

case 2:

printf ("\smoke CREAME BISCUIT");

temp=120\*quantity;

amt=amt+120\*quantity;

fprintf (order,"\n\t\t\smoke Cream biscuit: %d \* 120 = %d”, quantity, temp);

break;

case 3:

printf ("\smoke VANILLA FILLED BISCUIT");

temp=130\*quantity;

amt=amt+130\*quantity;

fprintf(order,"\n\t\t\smoke vanilla filled biscuit: %d \* 130 = %d", quantity ,temp);

break;

case 4:

printf ("\nSMOKE BISCUIT");

temp=110\*quantity;

amt=amt+110\*quantity;

fprintf(order,"\n\t\t\tSMOKE biscuit: %d \* 110 = %d", quantity ,temp);

break;

case 5:

printf("\nSMOKE POPCORN");

temp=110\*quantity;

amt=amt+110\*quantity;

fprintf(order,"\n\t\t\tSMOKE popcorn: %d \* 110 = %d", quantity ,temp);

break;

case 6:

printf("SMOKE STRAWBERRY BISCUIT");

temp=110\*quantity;

amt=amt+110\*quantity;

fprintf(order,"\n\t\t\tSMOKE strawberry biscuit: %d \* 110 = %d", quantity ,temp);

break;

case 7:

printf("MIXED FRUIT ROLL ICECREAM");

temp=120\*quantity;

amt=amt+160\*quantity;

fprintf(order,"\n\t\t\tSMOKE COOKOES: %d \* 120 = %d", quantity ,temp);

break;

case 8:

printf("CADBURY DAIRY MILK ROLL ICECREAM");

temp=190\*quantity;

amt=amt+190\*quantity;

fprintf(order,"\n\t\t\tcadbury dairy milk roll ice cream: %d \* 190 = %d", quantity ,temp);

break;

case 9:

printf("MANGO NUTS OVERLOADED ROLL ICECREAME");

temp=160\*quantity;

amt=amt+160\*quantity;

fprintf(order,"\n\t\t\tmango nuts overloaded roll ice creame: %d \* 160 = %d", quantity ,temp);

break;

case 10:

printf("VANILLA CHOCO ROLL ICECREAM");

temp=180\*quantity;

amt=amt+180\*quantity;

fprintf(order,"\n\t\t\tvanilla choco roll icecream: %d \* 180 = %d", quantity ,temp);

break;

case 11:

printf("BANANA ROLL ICE CREAM");

temp=145\*quantity;

amt=amt+145\*quantity;

fprintf(order,"\n\t\t\tbanana roll ice-cream: %d \* 145 = %d", quantity ,temp);

break;

case 12:

printf("BUTTER SCOTCH");

temp=180\*quantity;

amt=amt+180\*quantity;

fprintf(order,"\n\t\t\tbutter scotch: %d \* 180 = %d", quantity ,temp);

break;

case 13:

printf("OREO NUTELLA");

temp=180\*quantity;

amt=amt+180\*quantity;

fprintf(order,"\n\t\t\toreo nutella: %d \* 180 = %d", quantity ,temp);

break;

case 14:

printf("KITKAT COOKIE");

temp=180\*quantity;

amt=amt+180\*quantity;

fprintf(order,"\n\t\t\tkitkat cookie: %d \* 180 = %d", quantity ,temp);

break;

case 15:

printf("FRESH STRAWBERRY");

temp=170\*quantity;

amt=amt+170\*quantity;

fprintf(order,"\n\t\t\tfresh strawberry: %d \* 170 = %d", quantity ,temp);

break;

case 16:

printf("CHOCOLATE HOT");

temp=160\*quantity;

amt=amt+160\*quantity;

fprintf(order,"\n\t\t\tchocolate hot: %d \* 160 = %d", quantity ,temp);

break;

case 17:

printf("COLDDRINK ICE-CREAM");

temp=140\*quantity;

amt=amt+140\*quantity;

fprintf(order,"\n\t\t\tcolddrink icecreame: %d \* 140 = %d", quantity ,temp);

break;

case 18:

printf("MAGAI PAAN");

temp=130\*quantity;

amt=amt+130\*quantity;

fprintf(order,"\n\t\t\tmagai pann: %d \* 130 = %d", quantity ,temp);

break;

case 19:

printf("OREO CRUNCH");

temp=180\*quantity;

amt=amt+180\*quantity;

fprintf(order,"\n\t\t\toreo crunch: %d \* 180 = %d", quantity ,temp);

break;

case 20:

printf("VANILLA CARAMAL");

temp=160\*quantity;

amt=amt+160\*quantity;

fprintf(order,"\n\t\t\tvanilla caramal: %d \* 160 = %d", quantity ,temp);

break;

case 21:

printf("BLUE BERRY");

temp=150\*quantity;

amt=amt+150\*quantity;

fprintf(order,"\n\t\t\tblue berry: %d \* 150 = %d", quantity ,temp);

break;

case 22:

printf("STRAWBERRY+ BLUE BERRY");

temp=220\*quantity;

amt=amt+220\*quantity;

fprintf(order,"\n\t\t\t strawberry+blue berry: %d \* 220 = %d", quantity ,temp);

break;

case 23:

printf("DELICIOUS MANGO");

temp=160\*quantity;

amt=amt+160\*quantity;

fprintf(order,"\n\t\t\tdelicious mango: %d \* 160 = %d", quantity ,temp);

break;

case 24:

printf("PLAIN PARATHA");

temp=40\*quantity;

amt=amt+40\*quantity;

fprintf(order,"\n\t\t\tplain paratha: %d \* 40 = %d", quantity ,temp);

break;

case 25:

printf("ALOO PARATHA");

temp=79\*quantity;

amt=amt+79\*quantity;

fprintf(order,"\n\t\t\taloo paratha: %d \* 79 = %d", quantity ,temp);

break;

case 26:

printf("PANEER PARATHA");

temp=89\*quantity;

amt=amt+89\*quantity;

fprintf(order,"\n\t\tpaneer paratha:

break;

case 27:

printf("ONION PARATHA");

temp=79\*quantity;

amt=amt+79\*quantity;

fprintf(order,"\n\t\t\tonion paratha: %d \* 79= %d", quantity ,temp);

break;

case 28:

printf("EGG PARATHA");

temp=89\*quantity;

amt=amt+89\*quantity;

fprintf(order,"\n\t\t\tegg paratha: %d \* 89 = %d", quantity ,temp);

break;

case 29:

printf("MIX VEG. PARATHA");

temp=89\*quantity;

amt=amt+89\*quantity;

fprintf(order,"\n\t\t\tmix veg. paratha: %d \* 89 = %d", quantity ,temp);

break;

case 30:

printf("PANJABI PARATHA");

temp=99\*quantity;

amt=amt+99\*quantity;

fprintf(order,"\n\t\t\tpanjabi paratha: %d \* 99 = %d", quantity ,temp);

break;

case 31:

printf("CHOCOLATE PARATHA");

temp=129\*quantity;

amt=amt+129\*quantity;

fprintf(order,"\n\t\t\tchocolate paratha: %d \* 129 = %d", quantity ,temp);

break;

case 32:

printf("CHEESE PARATHA");

temp=119\*quantity;

amt=amt+119\*quantity;

fprintf(order,"\n\t\t\tcheese paratha: %d \* 119 = %d", quantity ,temp);

break;

case 33:

printf("CHICKEN PARATHA");

temp=159\*quantity;

amt=amt+159\*quantity;

fprintf(order,"\n\t\t\tchicken paratha: %d \* 159 = %d", quantity ,temp);

break;

case 34:

printf("COCONUT LASSI");

temp=55\*quantity;

amt=amt+55\*quantity;

fprintf(order,"\n\t\t\tcoconut lassi: %d \* 55 = %d", quantity ,temp);

break;

case 35:

printf("MANGO LASSI");

temp=55\*quantity;

amt=amt+55\*quantity;

fprintf(order,"\n\t\t\tmango lassi: %d \* 55 = %d", quantity ,temp);

break;

case 36:

printf("KEASR LASSI");

temp=70\*quantity;

amt=amt+70\*quantity;

fprintf(order,"\n\t\t\tkesar lassi: %d \* 70 = %d", quantity ,temp);

break;

case 37:

printf("NEPALI SALTED MILK LASSI");

temp=55\*quantity;

amt=amt+55\*quantity;

fprintf(order,"\n\t\t\tnepali salted milk lassi: %d \* 55 = %d", quantity ,temp);

break;

case 38:

printf("SPC. DARK EAT LASSI");

temp=80\*quantity;

amt=amt+80\*quantity;

fprintf(order,"\n\t\t\tSpc. dark eat lassi: %d \* 80 = %d", quantity ,temp);

break;

case 39:

printf("CHOCO LASSI");

temp=70\*quantity;

amt=amt+70\*quantity;

fprintf(order,"\n\t\t\tchoco lassi: %d \* 70 = %d", quantity ,temp);

break;

case 40:

printf("ROSE LASSI");

temp=70\*quantity;

amt=amt+70\*quantity;

fprintf(order,"\n\t\t\trose lassi: %d \* 70 = %d", quantity ,temp);

break;

case 41:

printf("DRY FRUIT LASSI");

temp=120\*quantity;

amt=amt+120\*quantity;

fprintf(order,"\n\t\t\tdry fruit lassi: %d \* 120 = %d", quantity ,temp);

break;

case 42:

printf("PUNJABI LASSI");

temp=70\*quantity;

amt=amt+70\*quantity;

fprintf(order,"\n\t\t\tpunjabi lassi: %d \* 70= %d", quantity ,temp);

break;

case 43:

printf("VEG PAKORA");

temp=79\*quantity;

amt=amt+79\*quantity;

fprintf(order,"\n\t\t\tveg pakora: %d \* 79 = %d", quantity ,temp);

break;

case 44:

printf("PANEER PAKORA");

temp=89\*quantity;

amt=amt+89\*quantity;

fprintf(order,"\n\t\t\tpaneer pakora: %d \* 89 = %d", quantity ,temp);

break;

case 45:

printf("ONION PAKORA");

temp=59\*quantity;

amt=amt+59\*quantity;

fprintf(order,"\n\t\t\tonion pakora: %d \* 59 = %d", quantity ,temp);

break;

case 46:

printf("PANEER ONION PAKORA");

temp=99\*quantity;

amt=amt+99\*quantity;

fprintf(order,"\n\t\t\tpaneer onion pakora: %d \* 120 = %d", quantity ,temp);

break;

case 47:

printf("MASALA PAPAD DRY");

temp=39\*quantity;

amt=amt+39\*quantity;

fprintf(order,"\n\t\t\tmasala papad dry: %d \* 39 = %d", quantity ,temp);

break;

case 48:

printf("MASALA PAPAD FRIES");

temp=49\*quantity;

amt=amt+49\*quantity;

fprintf(order,"\n\t\t\tmasala papad fries: %d \* 49= %d", quantity ,temp);

break;

case 49:

printf("CHOLE BHATURE");

temp=119\*quantity;

amt=amt+119\*quantity;

fprintf(order,"\n\t\t\t chole bhature: %d \* 119 = %d", quantity ,temp);

break;

case 50:

printf("CHOLE KULCHE");

temp=79\*quantity;

amt=amt+79\*quantity;

fprintf(order,"\n\t\t\tcholekulche: %d \* 79 = %d", quantity ,temp);

break;

default:

printf("finall bill\n\n");

goto x;

break;

}

}

//printf("finall bill");

x:

fprintf(order,"\ntotal: %d",amt);

printf("YOUR TOTAL BILL =%d\n",amt);

printf("THANKS FOR YOUR COMING!!!!!");

fclose(fptr);

fclose(order);

}

int main()

{

int ch;

double number;

char name [20];

FILE \*f;

login ();

detail (&number, name);

printf ("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOMER DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf ("\t\tNAME: %s\n",name);

printf ("\t\phone NUMBER: %ld”, number);

bill ();

f=fopen("bill.txt”,”r");

while((ch=fgetc(f))! =EOF)

printf ("%c”, ch);

fclose(f);

return 0;

}

void detail (double\* number, char\* name)

{

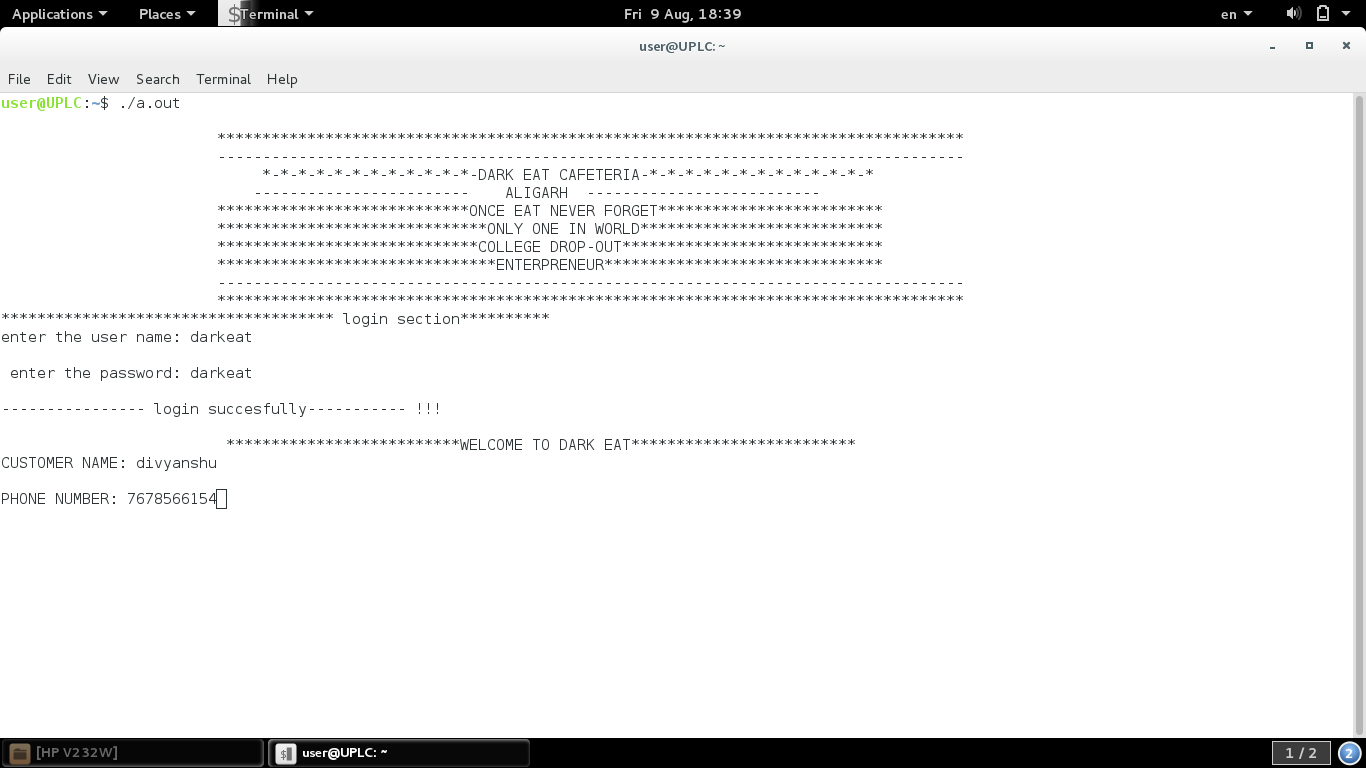
printf ("CUSTOMER NAME: ");

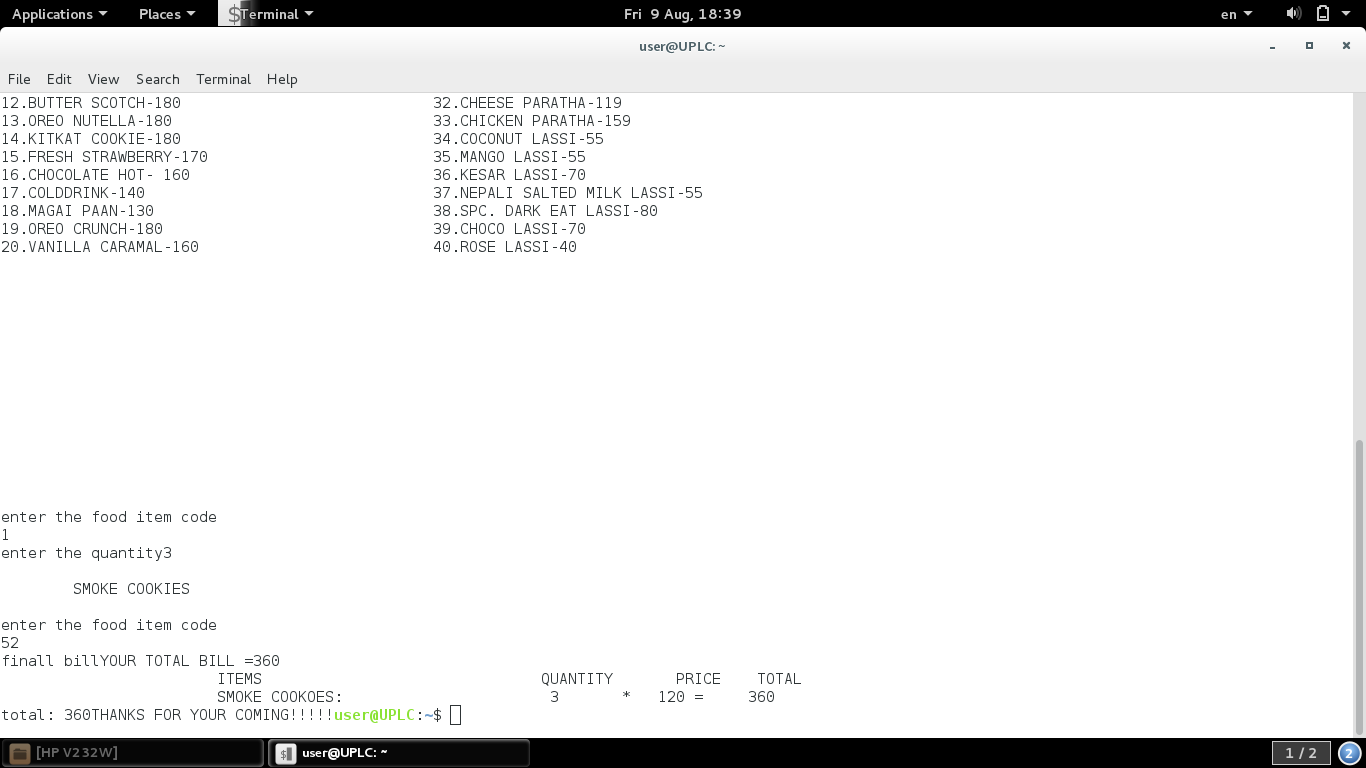
scanf ("%s”, name);

printf ("PHONE NUMBER: ");

scanf ("%ld”, &number);}

OUTPUT SCREENSHOT





BIBLIOGRAPHY

BOOKS REFERRED

* EXPLORING C, THE C PROGRAMMING LANGUAGE

WEBSITES REFERRED

1. [www.google.com](http://www.google.com)

2-[www.youtube.com](http://www.youtube.com)

3-[www.slideshare.net](http://www.slideshare.net)