**KHAITR DARI**

**--Khaate Raho, Mangate Raho**

**A Project Report**

***Submitted by***

***1.Manvi Lohia(C056)***

***2.Janki Kanakia(C046)***

***3.Yashvi Mehta (C060)***

***Under the Guidance of***

## Prof. KRISHNA SAMDANI

## B. TECH INTEGRATED

## COMPUTER ENGINEERING

## At



**MUKESH PATEL SCHOOL OF TECHNOLOGY, MANAGEMENT AND ENGINEERING**

**DECLARATION**

We, **Manvi Lohia, Janki Kanakia, Yashvi Mehta**,Roll No.**C056, C046, C060** B.Tech Integrated (Computer Engineering), understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graph diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of

the Student:

Name: Manvi Lohia, Janki Kanakia, Yashvi Mehta

Roll No: C056, C046 , C060

Place: Mumbai

Date:

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TOPIC** | **PAGE NO.** |
|  | ABSTRACT | IV |
| 1. | INTRODUCTION | 5 |
| 2. | TOOLS & TECHNIQUES | 6 |
| 3. | SIMULATIONS & RESULTS | 8 |
| 4. | CONCLUSION | 15 |
| 5. | FUTURE WORK | 16 |
| 6. | REFRENCES  APPENDIX | 17 |

**ABSTRACT**

“Khatir Dari ” is an app to help ease your food ordering process. Our app lets you compare prices along with providing other functionalities that are offered by multiple food delivery platforms. You can choose between restaurants and see the prices offered by different platforms for each dish available. You can also view menus of multiple restaurants and then make your choice of ordering. Choice between cash on delivery payment and payment through online wallets has also been provided to the user. Information regarding the person delivering the order is also provided. The user is also asked to rate their experience and provide a feedback regarding the possible improvements. To implement this, CODE::BLOCKS coding platform with GNU GCC compiler has been used. An additional functionality that has been offered is donating leftover food to the underprivileged.

**INTRODUCTION**

“KHAATIR DAARI—Khaate Raho, Mangate Raho”, as the name suggests is another food-based app in this massive sea of unlimited apps for everything but what makes our app standout is its effectiveness/multifunctionality/conciseness/how it makes food ordering easier than ever considered.

This app allows users to compare the prices, delivery times and access offers of multiple food ordering platforms such as Zomato, Swiggy etc. and decide what’s best for you without having to constantly switch between different websites.

We provide the user with discounts from our own side as an additional perk to using out website.

We also provide the user with a chance contribute towards the society by donating any leftover food to the underprivileged. (Food collection and distribution is done by us.)

This ensures that not only the user but also someone else out there rests with a satisfied appetite.

**TOOLS AND TECHNIQUES**

IDE used:

CODE::BLOCKS

Version 17.12

Last Updated on Sunday, 31 December 2017 15:24

1. We provide functionalities like:
   1. login/register:

(Figure 1.10 - 1.12) The first option provided to the user is whether they want to register or login to the application. This choice further alters the offers the user gets and the information the user has to input.

* 1. choose between different restaurants:

(Figure 1.20) The user is now given an option of multiple restaurants which he can order from. Prices offered by different platforms for each dish are also given to help improve quality of comparison. The user can also view menu of different restaurants before moving ahead with placing the order.

* 1. place order:

(Figure 1.30) The user now places the order from the chosen restaurant by inputting the dish codes. The input “6” terminates the order and further calculations follow. On entering the wrong code, an error message is printed and the user is asked to input the correct code.

* 1. choose from multiple delivery sites:

(Figure 1.40- 1.42) The final bill amount and delivery time for the users order is then calculated and printed. Choice is made by user and further offers available are applied. New users get an additional code for further discount along with the discount offered by individual delivery platforms.

* 1. choose between payment modes and options:

(Figure 1.50-1.52) The user is then given choice to choose from multiple payment modes like Cash-on-Delivery and online wallets. Choice between two online wallets is also available. Completion of transaction occurs when user enters phone number and correct otp generated at the start.

* 1. option to donate to the society:

(Figure 1.60) The user is asked if they want to contribute by donating leftover food.

* 1. rate our services and provide feedbacks for improvements

(Figure 1.70) The user is then asked to rate their experience and provide a feedback for any additions they would like in the application.

1. C++ Techniques used:

* inheritance
* file handling
* loops
* nested loops
* switch cases
* nested switch cases
* if else statements
* jump statements
* functions
* classes

**SIMULATIONS AND RESLUTS**

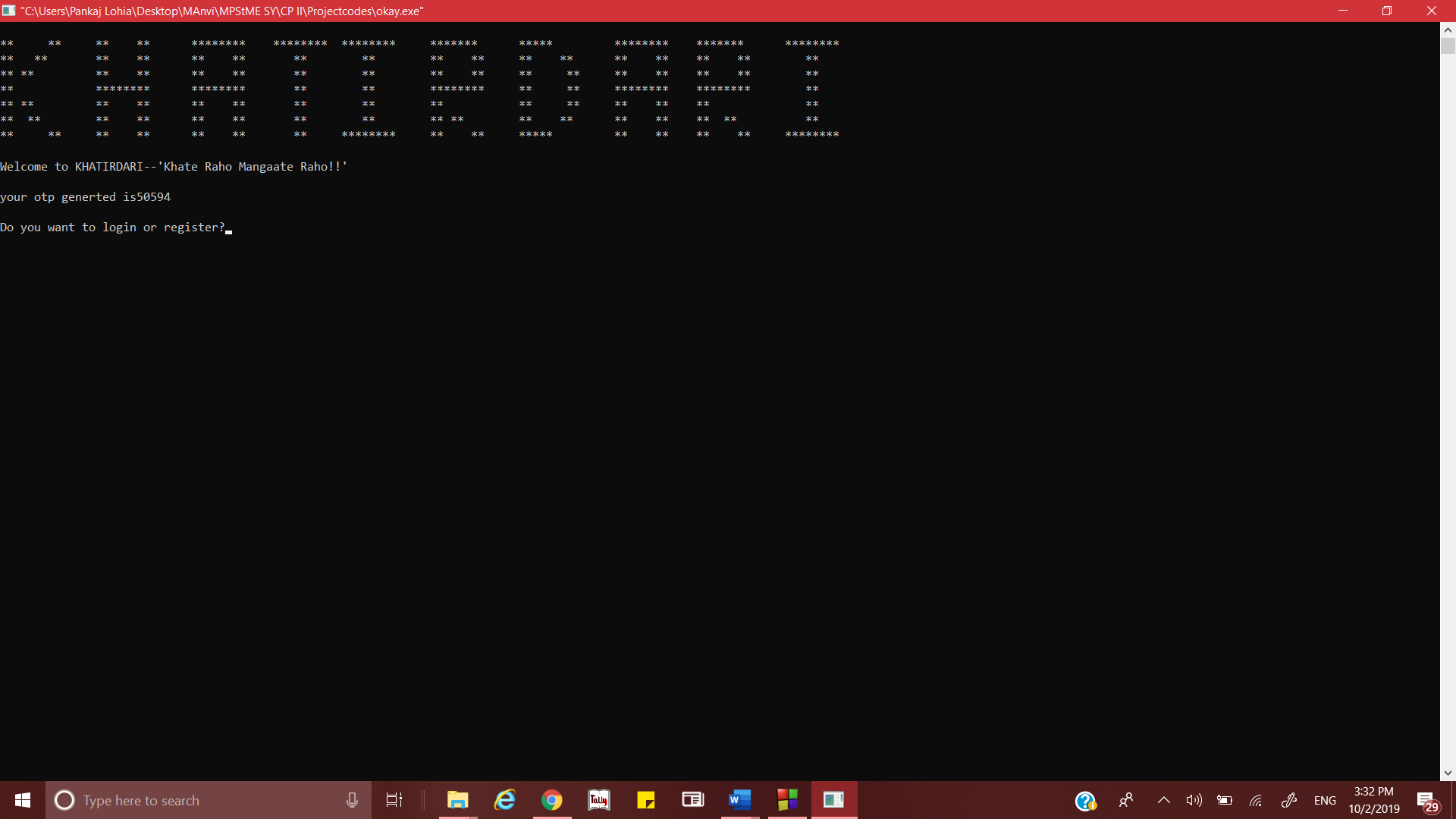
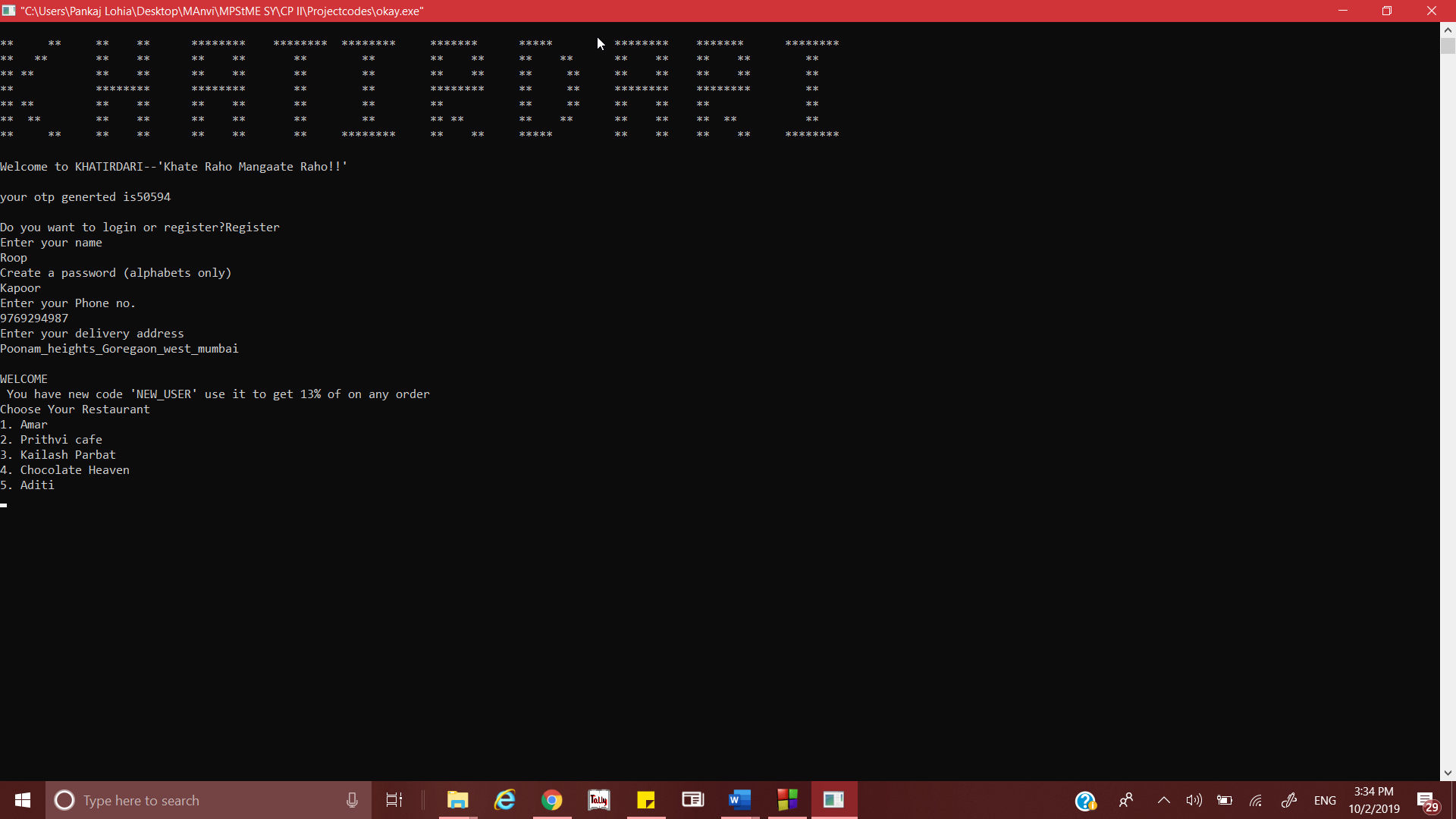


Figure 1.10 – Showing login and register option

Figure 1.11—Showing registration of a user and the details required to register

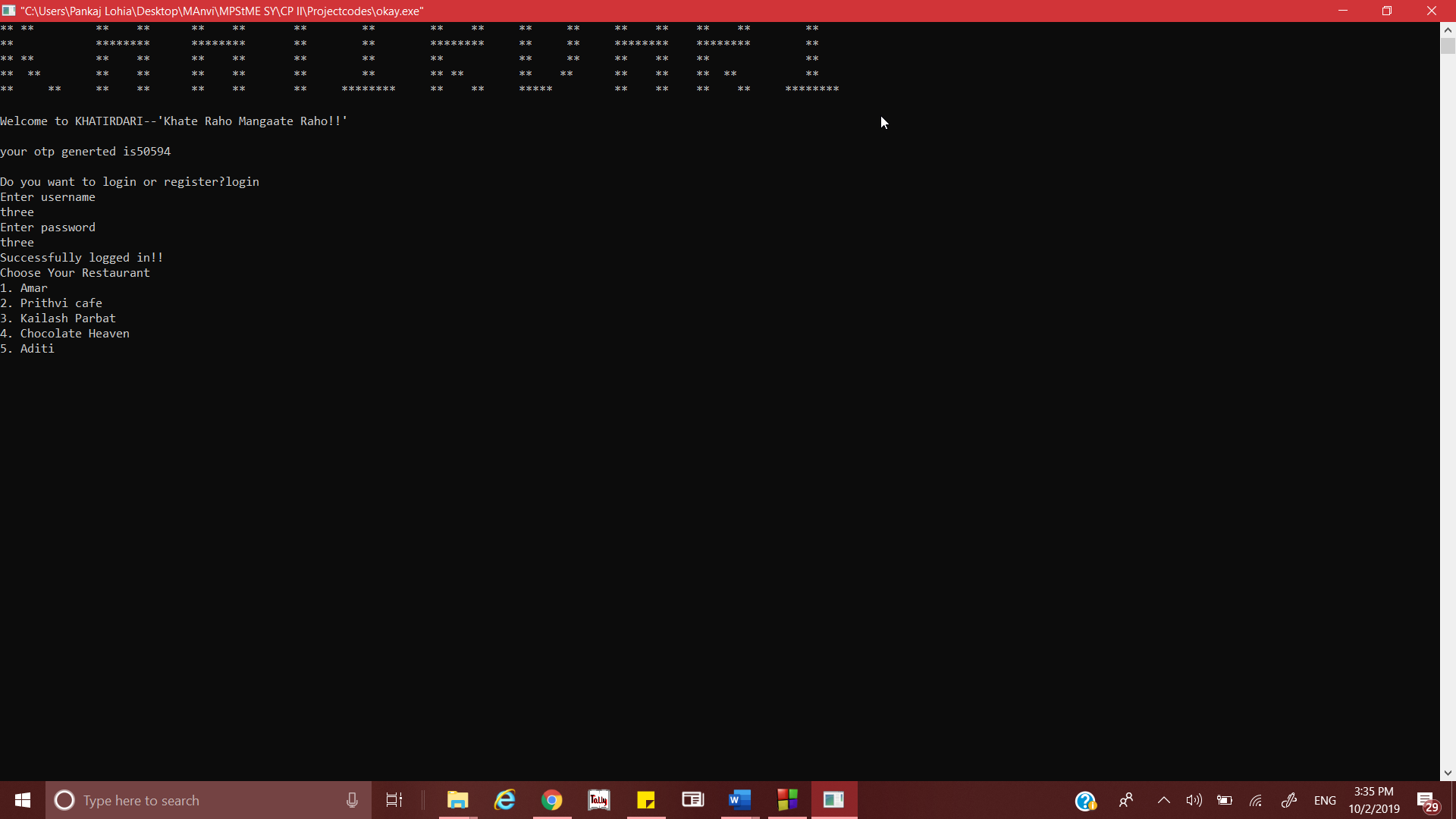


Figure 1.12 – Showing the code flow for login and the choice provided to user for restaurants.

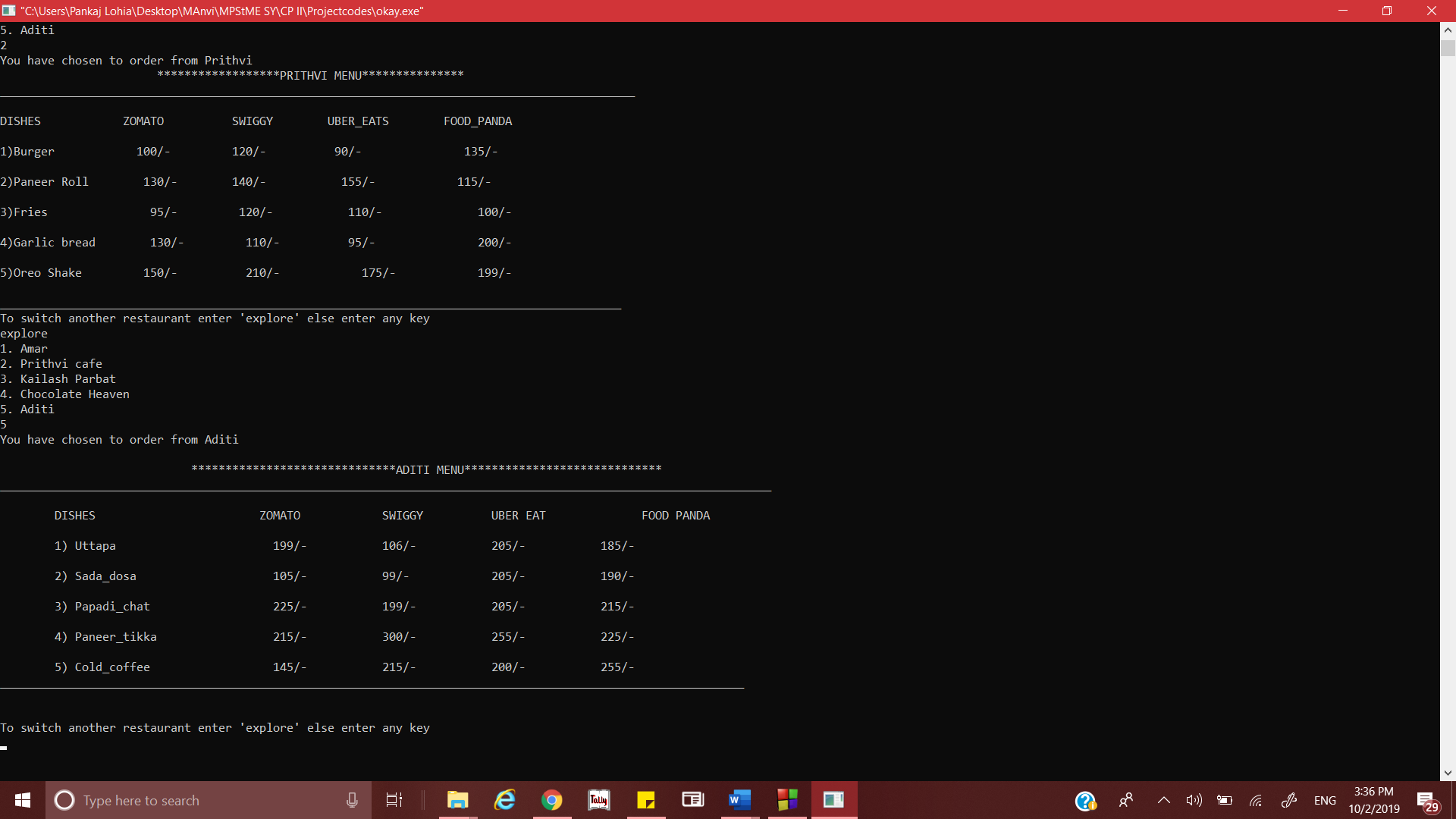


Figure 1.20 —showing the restaurant options and prices offered by all delivery platforms for the dishes.

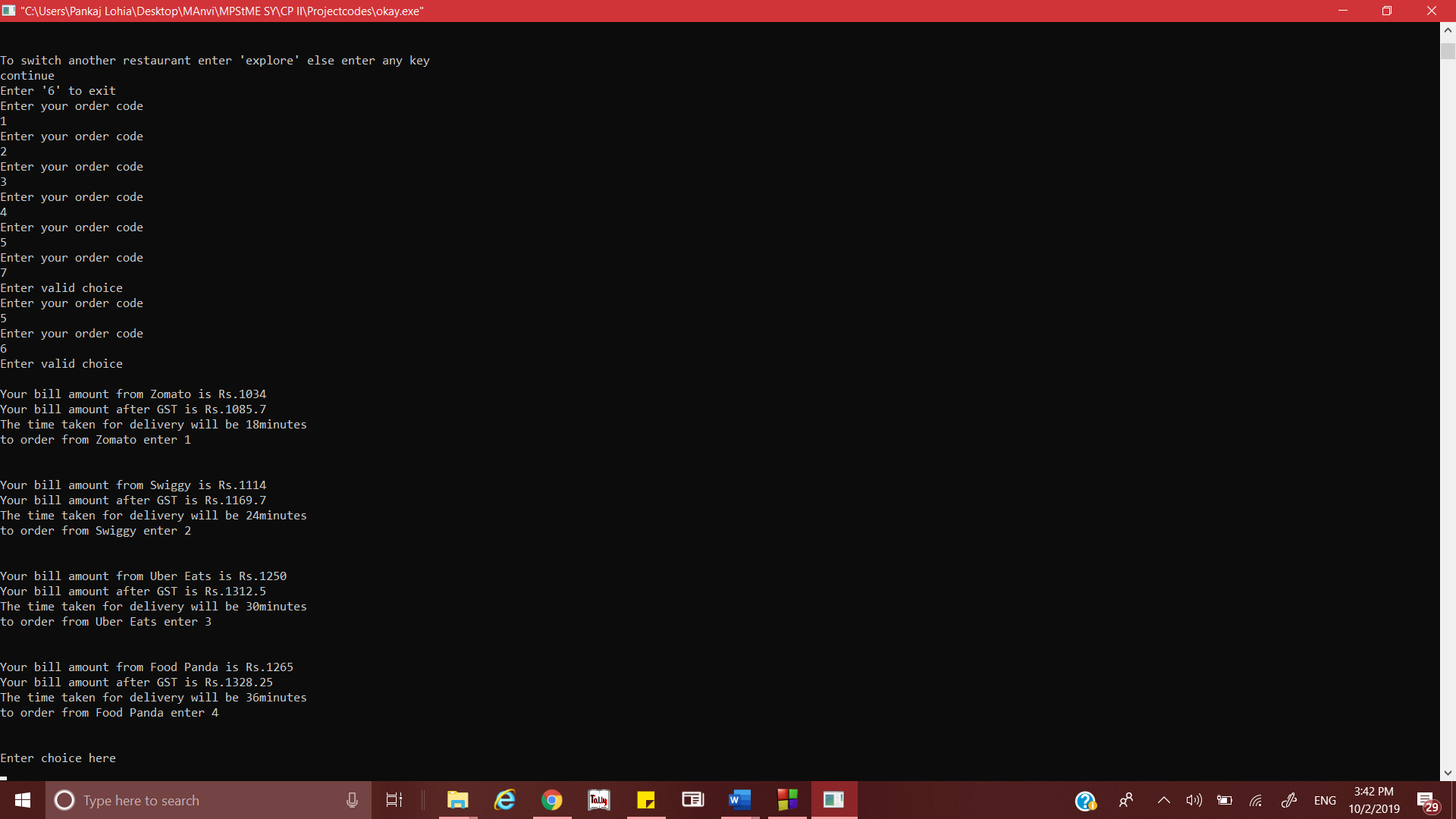


Figure 1.30 —showing the ordering process. If user enters wrong dish code, error message is printed else the system continues to take order till “6” is entered, which terminates the order.

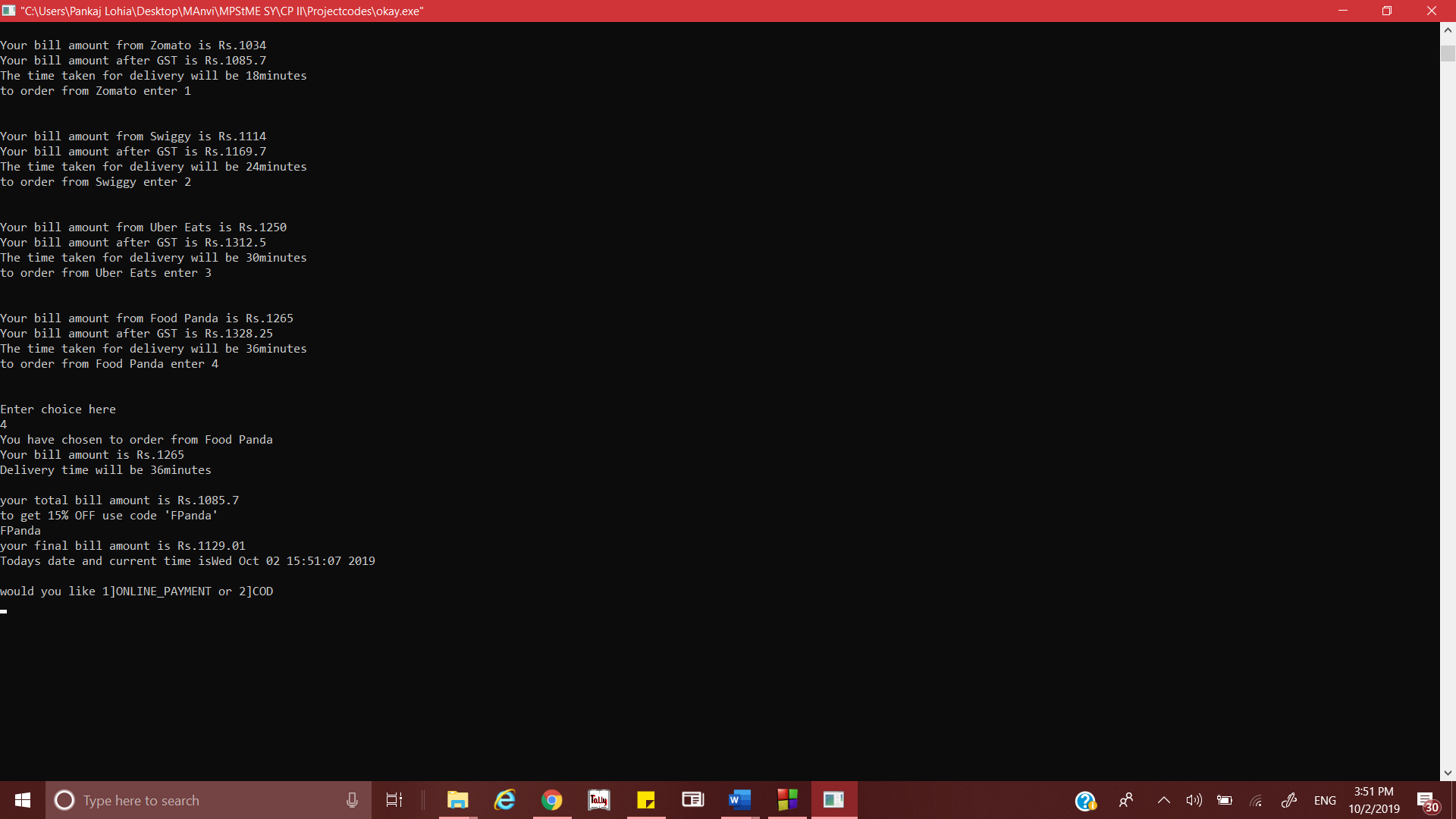


Figure 1.40 —showing printed bill amounts and delivery times for the user to choose from. It also shows the offer available to the users who have logged in.

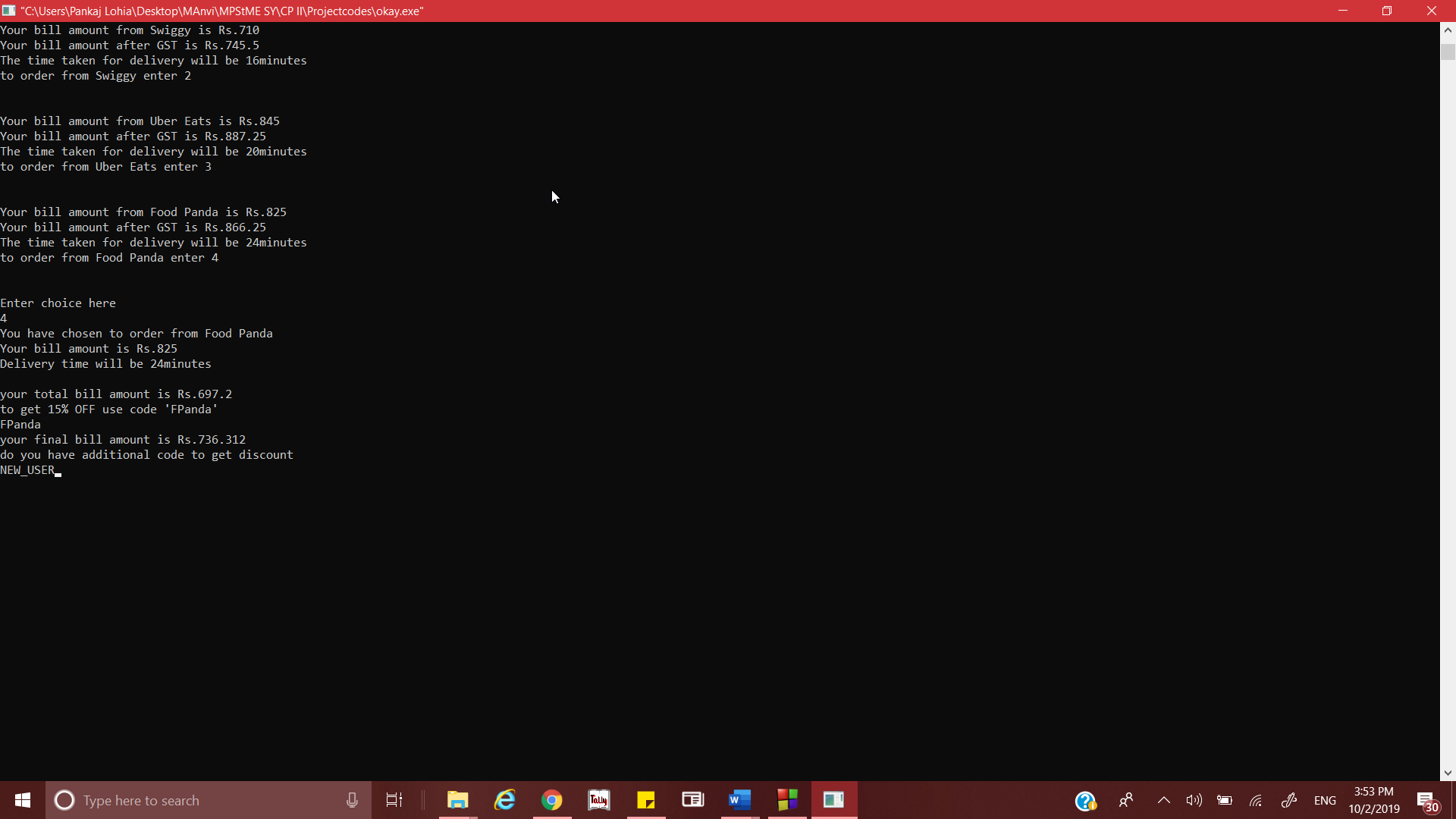


Figure 1.41 – showing additional offers provided to new user (register) on the bill

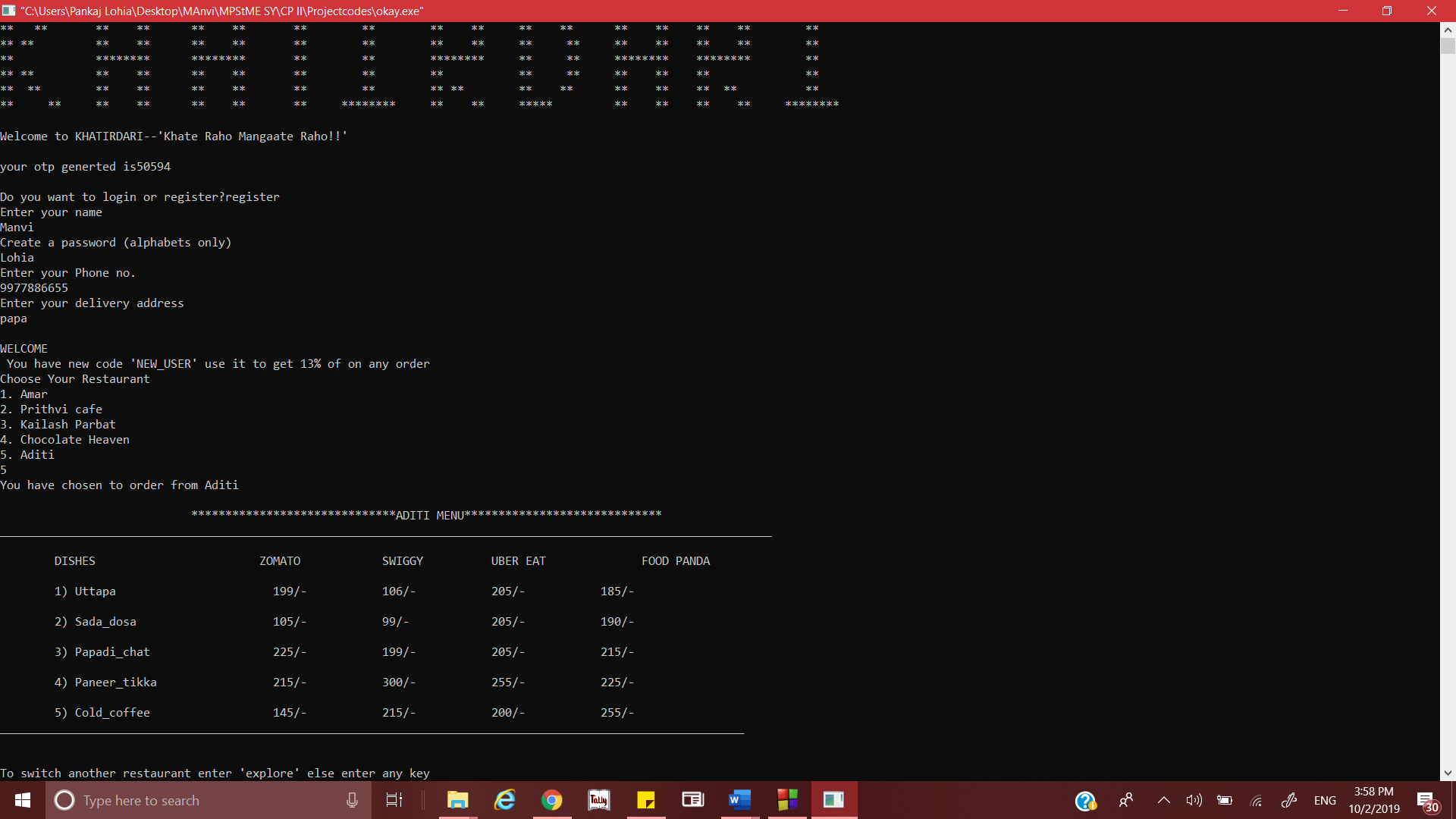


Figure 1.42 – showing additional code provided to new user at start of the application.

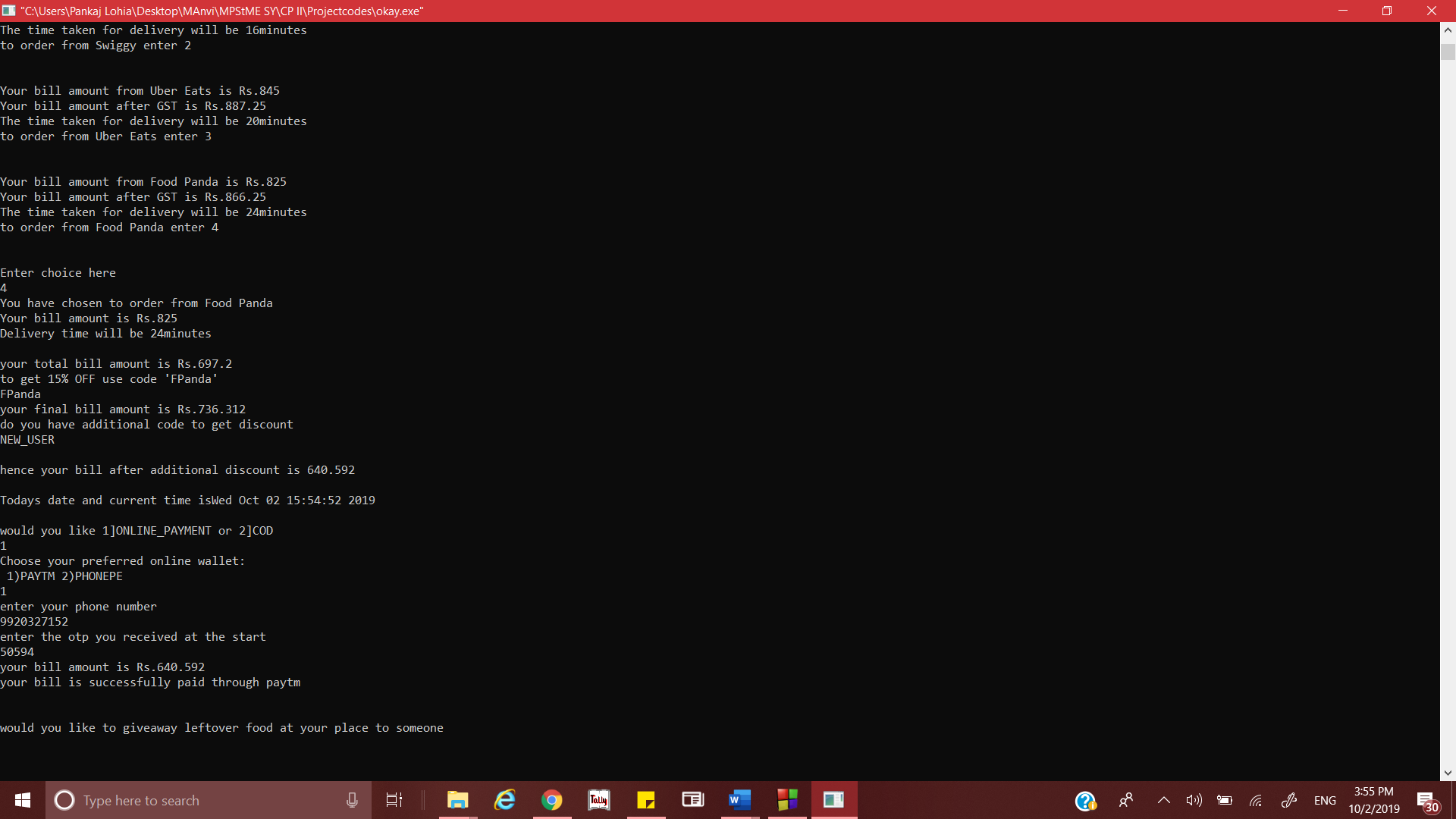


Figure 1.50 – showing the users final bill amount after all discounts and addition of GST. The option to choose from CashOnDelivery or Online Wallets is given which further gives the option of two different online wallets. On choice of online wallets, phone number is asked and otp provided at start(figure 1.51) is asked to verify and conclude the payment.

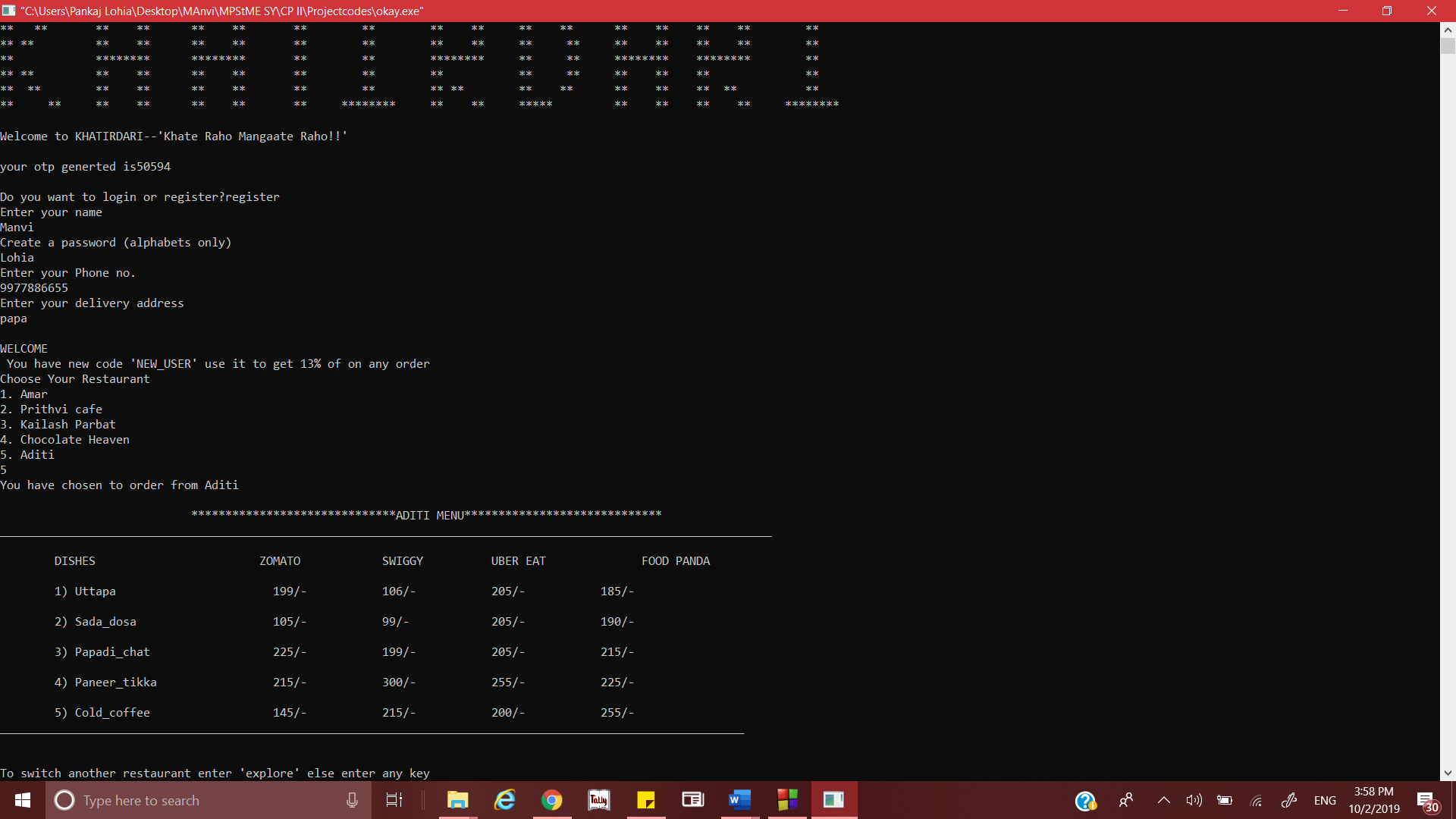


Figure 1.51 – showing otp generated at the start.

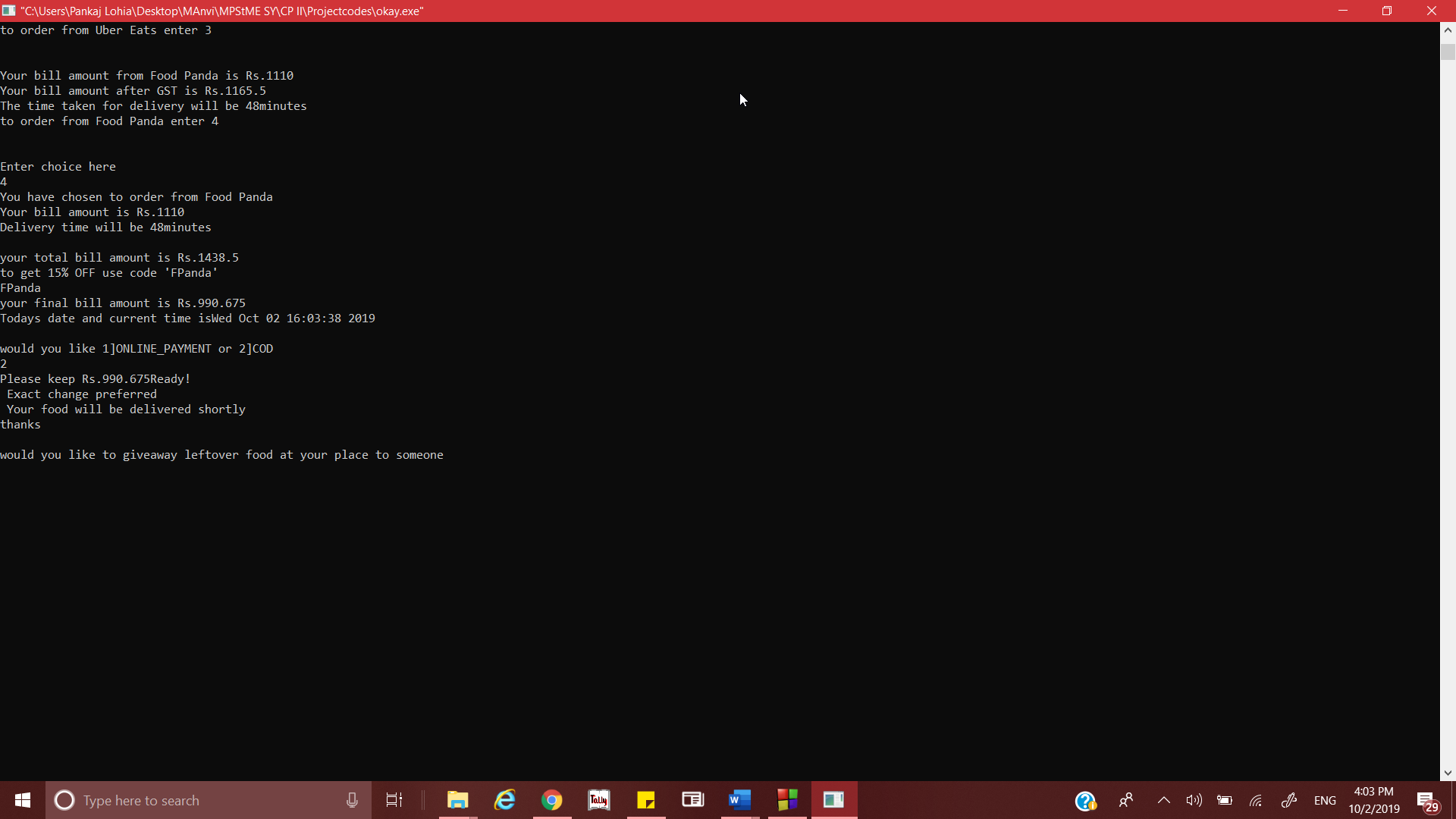


Figure 1.52—showing the flow for Cash on Delivery

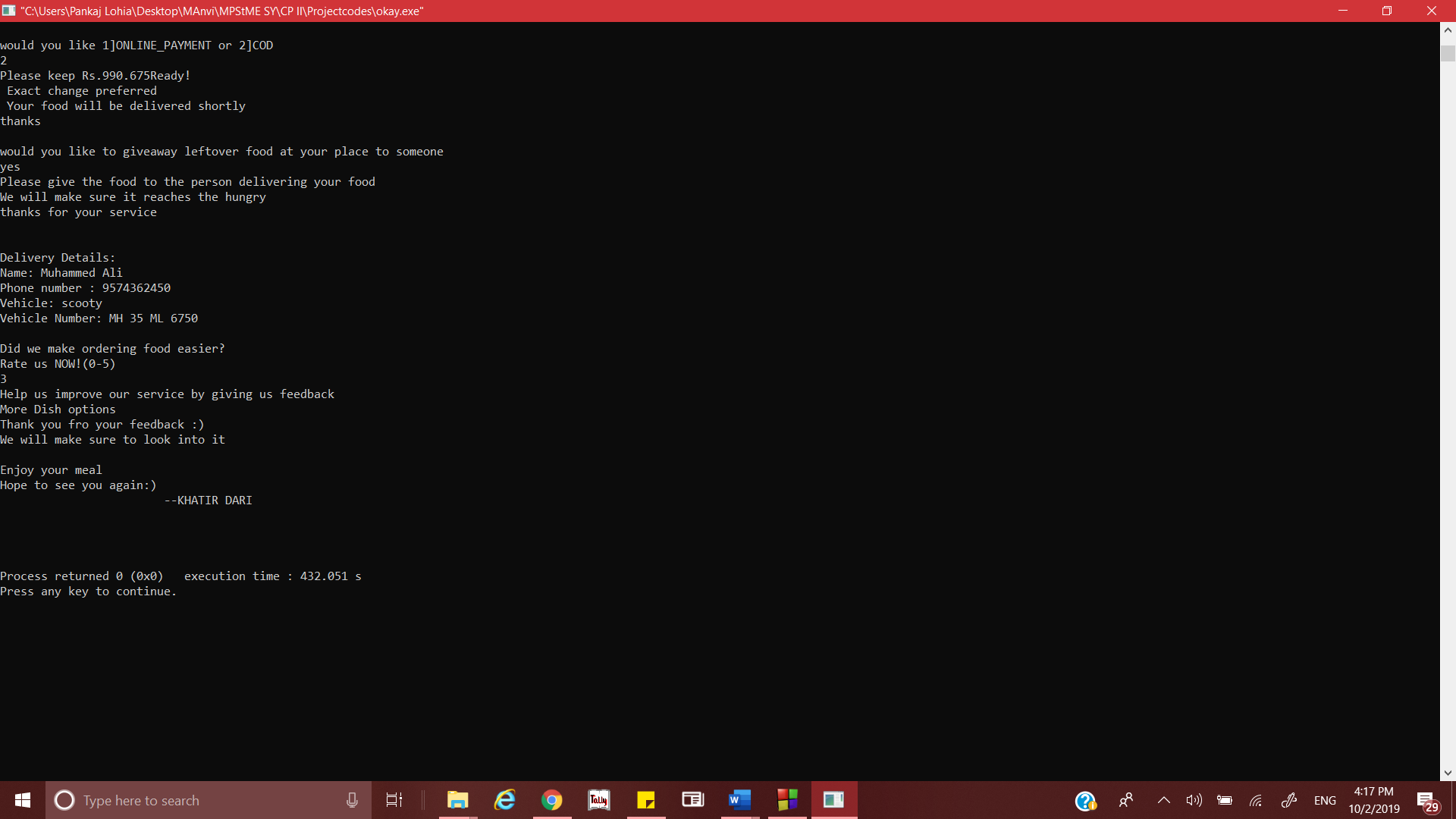


Figure 1.60 – showing option to donate to society and delivery details.

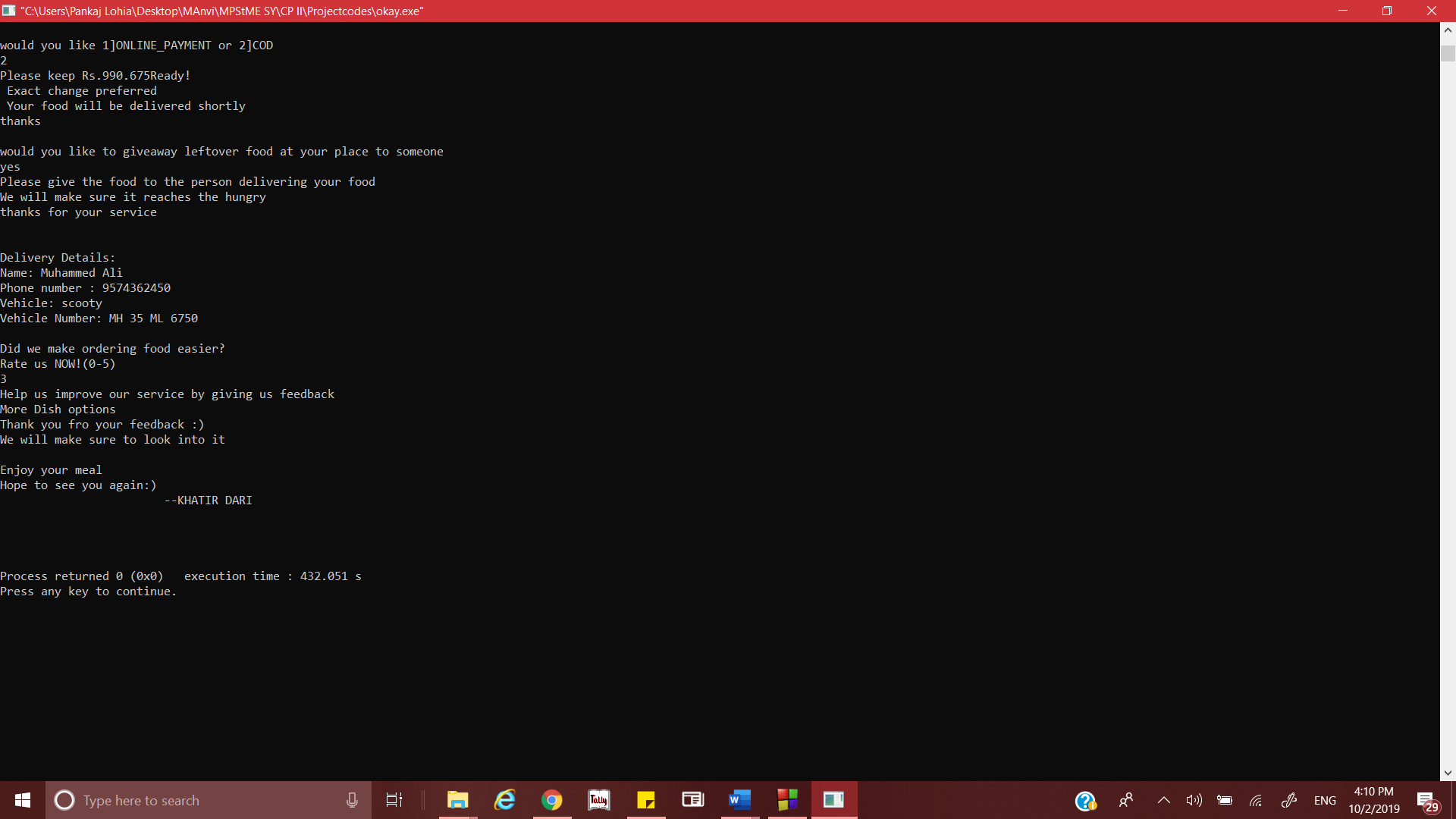


Figure 1.70 – showing the review and feedback system.

**CONCLUSION**

The primary focus of this console application was to brainstorm and bring revolution in the field of food ordering but providing a platform like never seen before. An additional facility of donating leftover food to the society is provided in the console application. We tackled each boulder in our way by working together and getting as many as possible different ways to handle the situation. In doing so we made sure that the most effective, user friendly and a smooth program flow was delivered to the user, free of any errors or limitations.

**FUTURE WORK**

We intend to improve the overall functionality and user-friendliness of the software by:

* collaborating with fast food chains and restaurants directly to help user order directly from the restaurant instead of via another delivery platform.
* Introducing more delivery platforms to improve comparison variety.
* Introducing additional criterias such as delivery reviews, restaurant reviews etc. to improve quality of comparison and thus choice of decision.
* Introducing super membership with perks like restaurant table booking at discounted rates etc.

**REFRENCES**

1. Textbook references: Object Oriented Programming with E. Balagurusamy.
2. Possible functionalities and program flow reference from Trivago. (<https://www.trivago.in/>)
3. Restaurants and prices form

* Zomato: <https://www.zomato.com/mumbai/order-food-online?utm_source=Google&utm_medium=CPC&utm_term=zomato&utm_campaign=Gsearch_P-MWeb_O-NA_C-Brand_A-NewUser_SC-Generic_L-Mumbai&utm_source=google&utm_medium=cpc&utm_campaign=Gsearch_P-MWeb_O-NA_C-Brand_A-NewUser_SC-Generic_L-Mumbai&utm_term=zomato&gclid=CjwKCAjwldHsBRAoEiwAd0JybQBWjHKNObx4YPc7aMARSIwPA2RI3vvXEeniktMKTWdcVbw3m_wdoBoCyAsQAvD_BwE> AND through mobile application.
* Swiggy: <https://www.swiggy.com/restaurants> AND through mobile application.
* Uber Eats: <https://www.ubereats.com/en-IN/?utm_source=google&utm_medium=cpc-brand&utm_campaign=search-google-brand_77_-99_IN-National_e_dsk_acq_cpc_en_CNT_UberEats_Exact&adgroup_name=UberEats%3EExact&utm_term=uber+eats&kw=uber+eats&campaign_id=71700000037880800&cid=71700000037880800&adgroup_id=58700004177301560&adg_id=58700004177301560&kw_id=p34591978700&kwid=p34591978700&ad_id=373986642600&ext_id=&ran=11855163600123151894&lint_id=&lphy_id=9062213&pos=1t1&dev=c&net=g&match=e&placement=&target=&gclid=CjwKCAjwldHsBRAoEiwAd0JybUZQJjYzTJXvaFhP2eKDM7z5CG5530rPFZsM8aZwPV_truINR1BjLRoC2c4QAvD_BwE&gclsrc=aw.ds> AND through mobile application.
* Food Panda: <https://www.foodpanda.in/> (currently not working) AND through mobile application.

1. Other sources were as follows
2. <https://www.geeksforgeeks.org/c-plus-plus/> [28.09.19] [15.08.19]
3. <https://www.tutorialspoint.com/cplusplus/index.htm> [12.08.19] [28.08.19]
4. <https://www.onlinegdb.com/online_c++_compiler> [30.09.19] [01.10.19]

**APPENDIX**

#include<iostream>

#include<stdlib.h>

#include<string>

#include<fstream>

#include<time.h>

using namespace std;

int otp;

void logo()

{

cout<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\*\*\*\*\*\*\* "<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* "<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* "<<endl;

cout<<"\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\* "<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* "<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* "<<endl;

cout<<"\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\* \*\* \*\*\*\*\* \*\* \*\* \*\* \*\* \*\*\*\*\*\*\*\* "<<endl;

}

void random\_otp()

{

otp = rand()\*1234;

cout<<"your otp generted is"<<otp<<endl;

}

void bill\_pay(double osum)

{

int a,online;

double bill\_amt=osum;

long long phone;

int otp1;

cout<<endl;

cout<<"would you like 1]ONLINE\_PAYMENT or 2]COD"<<endl;

cin>>a;

if(a==1)

{

cout<<"Choose your preferred online wallet:\n 1)PAYTM 2)PHONEPE"<<endl;

cin>>online;

switch(online)

{

case 1:

{

cout<<"enter your phone number"<<endl;

cin>>phone;

//check whether entered number and saved number is same

cout<<"enter the otp you received at the start"<<endl;

cin>>otp1;

if(otp1==otp)

{

cout<<"your bill amount is Rs."<<bill\_amt<<endl;

cout<<"your bill is successfully paid through paytm"<<endl;

cout<<endl;

}

else

{

cout<<"otp entered is incorrect"<<endl;

cout<<"redirecting to payment page"<<endl;

bill\_pay(bill\_amt);

}

}break;

case 2:

{

cout<<"enter your phone number"<<endl;

cin>>phone;

cout<<"enter the otp you received at the start"<<endl;

cin>>otp1;

if(otp1==otp)

{

cout<<"your bill is successfully paid through phonepe"<<endl;

cout<<endl;

}

else

{

cout<<"otp entered is incorrect"<<endl;

cout<<"redirecting to payment page"<<endl;

bill\_pay(bill\_amt);

}

}break;

}

}

else//cod

{

cout<<"Please keep Rs."<<bill\_amt<<"Ready!\n Exact change preferred\n Your food will be delivered shortly "<<endl;

cout<<"thanks"<<endl;

}

}

double offer\_r(double osum)

{

string newcode;

double bill\_amt,amt;

cout<<"do you have additional code to get discount"<<endl;

cin>>newcode;

cout<<endl;

if(newcode=="NEW\_USER"|| newcode=="new\_user")

{

amt=0.13\*osum;

bill\_amt=osum-amt;

cout<<"hence your bill after additional discount is "<<bill\_amt<<endl;

cout<<endl;

return bill\_amt;

}

}

void time()

{

time\_t t;

time(&t);

cout<<"Todays date and current time is";

cout<<ctime(&t);

}

class driver

{

public:

void driver1()//zomato

{

cout<<"\n\nDelivery Details:\n";

cout<<"Name: Masnoon "<<endl;

cout<<"Phone number : 9869195959"<<endl;

cout<<"Vehicle: scooty"<<endl;

cout<<"Vehicle Number: MH 04 DR 7865"<<endl;

}

void driver2()//swiggy

{

cout<<"\n\nDelivery Details:\n";

cout<<"Name: Rahul Shah "<<endl;

cout<<"Phone number : 9869145890"<<endl;

cout<<"Vehicle: scooty"<<endl;

cout<<"Vehicle Number: MH 02 EE 1755"<<endl;

}

void driver3()//uber

{

cout<<"\n\nDelivery Details:\n";

cout<<"Name: Preet Gokhale "<<endl;

cout<<"Phone number : 7862486313"<<endl;

cout<<"Vehicle: scooty"<<endl;

cout<<"Vehicle Number: MH 40 JK 2560"<<endl;

}

void driver4()//foodpanda

{

cout<<"\n\nDelivery Details:\n";

cout<<"Name: Muhammed Ali "<<endl;

cout<<"Phone number : 9574362450"<<endl;

cout<<"Vehicle: scooty"<<endl;

cout<<"Vehicle Number: MH 35 ML 6750"<<endl;

}

//extra driver

void driver5()

{

cout<<"\n\nDelivery Details:\n";

cout<<"Name: Prateek Mevad "<<endl;

cout<<"Phone number : 7852146320"<<endl;

cout<<"Vehicle: scooty"<<endl;

cout<<"Vehicle Number: MH 05 YM 1890"<<endl;

}

}driver\_obj;

class Amar

{

protected:

int a,choice;

string ch;

int count\_1=0,count\_2=0,count\_3=0,count\_4=0,count\_5=0;

double cgst,sgst;

//variable list fro Zomato

int count\_tz;

int sum\_1z,sum\_2z,sum\_4z,sum\_5z,sum\_3z,sum\_tz,delTime\_z;

double gst\_z;

// variable list for Swiggy

int count\_ts;

int sum\_1s,sum\_2s,sum\_4s,sum\_5s,sum\_3s,sum\_ts,delTime\_s;

double gst\_s;

// variable list for Uber eats

int count\_tu;

int sum\_1u,sum\_2u,sum\_4u,sum\_5u,sum\_3u,sum\_tu,delTime\_u;

double gst\_u;

// variable list for food panda

int count\_tf;

int sum\_1f,sum\_2f,sum\_4f,sum\_5f,sum\_3f,sum\_tf,delTime\_f;

double gst\_f;

//offer variables

string code;

double osum,off;

public:

void Decide(int a\_status)//status from choice

{

int lore=a\_status;

cout<<endl<<"Enter choice here"<<endl;

cin>>choice;

switch(choice)

{

case 1:

cout<<"You have chosen to order from Zomato"<<endl;

cout<<"Your bill amount is "<<sum\_tz<<endl;

cout<<"Delivery time will be "<<delTime\_z<<"minutes"<<endl;

cout<<endl;

cout<<"your total bill amount after GST is Rs."<<gst\_z<<endl;

cout<<"to get 30% OFF use code 'ZOMATO'"<<endl;

cin>>code;

if(code=="zomato"||code=="ZOMATO")

{

off=(0.3\*gst\_z);

osum = gst\_z-off;

cout<<"your final bill amount is Rs."<<osum<<endl;

}

else

{

cout<<"Invalid code entered"<<endl<<"Discount not applicable"<<endl;

}

if(lore==1)//register

{

offer\_r(osum);

}

time();

bill\_pay(osum);

cout<<"would you like to giveaway leftover food at your place to someone"<<endl;

cin>>ch;

if(ch=="yes" || ch=="y" || ch=="Yes")

{

cout<<"Please give the food to the person delivering your food"<<endl;

cout<<"We will make sure it reaches the hungry"<<endl;

cout<<"thanks for your service"<<endl;

}

driver\_obj.driver1();

break;

case 2:

cout<<"You have chosen to order from Swiggy"<<endl;

cout<<"Your bill amount is Rs."<<sum\_ts<<endl;

cout<<"Delivery time will be "<<delTime\_s<<"minutes"<<endl;

cout<<endl;

cout<<"your total bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"to get 12% OFF use code 'SwIgGy'"<<endl;

cin>>code;

if(code=="SwIgGy")

{

off=(0.12\*gst\_z);

osum = gst\_s-off;

cout<<"your final bill amount is Rs."<<osum<<endl;

}

else

{

cout<<"Invalid code entered"<<endl<<"Discount not applicable"<<endl;

}

if(lore==1)//register

{

osum=offer\_r(osum);

}

time();

bill\_pay(osum);

cout<<"would you like to giveaway leftover food at your place to someone"<<endl;

cin>>ch;

if(ch=="yes" || ch=="y" || ch=="Yes")

{

cout<<"Please give the food to the person delivering your food"<<endl;

cout<<"We will make sure it reaches the hungry"<<endl;

cout<<"thanks for your service"<<endl;

}

driver\_obj.driver2();

break;

case 3:

cout<<"You have chosen to order from Uber Eats"<<endl;

cout<<"Your bill amount is Rs."<<sum\_tu<<endl;

cout<<"Delivery time will be "<<delTime\_u<<"minutes"<<endl;

cout<<endl;

cout<<"your total bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"to get 20% OFF use code 'UEats'"<<endl;

cin>>code;

if(code=="UEats")

{

off=(0.2\*gst\_u);

osum = gst\_u-off;

cout<<"your final bill amount is Rs."<<osum<<endl;

}

else

{

cout<<"Invalid code entered"<<endl<<"Discount not applicable"<<endl;

}

if(lore==1)//register

{

osum=offer\_r(osum);

}

time();

bill\_pay(osum);

cout<<"would you like to giveaway leftover food at your place to someone"<<endl;

cin>>ch;

if(ch=="yes" || ch=="y" || ch=="Yes")

{

cout<<"Please give the food to the person delivering your food"<<endl;

cout<<"We will make sure it reaches the hungry"<<endl;

cout<<"thanks for your service"<<endl;

}

driver\_obj.driver3();

break;

case 4:

cout<<"You have chosen to order from Food Panda"<<endl;

cout<<"Your bill amount is Rs."<<sum\_tf<<endl;

cout<<"Delivery time will be "<<delTime\_f<<"minutes"<<endl;

cout<<endl;

cout<<"your total bill amount is Rs."<<gst\_z<<endl;

cout<<"to get 15% OFF use code 'FPanda'"<<endl;

cin>>code;

if(code=="FPanda")

{

off=(0.15\*gst\_f);

osum = gst\_f-off;

cout<<"your final bill amount is Rs."<<osum<<endl;

}

else

{

cout<<"Invalid code entered"<<endl<<"Discount not applicable"<<endl;

}

if(lore==1)//register

{

osum=offer\_r(osum);

}

time();

bill\_pay(osum);

cout<<endl;

cout<<"would you like to giveaway leftover food at your place to someone"<<endl;

cin>>ch;

if(ch=="yes" || ch=="y" || ch=="Yes")

{

cout<<"Please give the food to the person delivering your food"<<endl;

cout<<"We will make sure it reaches the hungry"<<endl;

cout<<"thanks for your service"<<endl;

}

driver\_obj.driver4();

break;

default:

cout<<"Enter valid choice"<<endl;

}

}

void order()

{

cout<<"Enter '6' to exit"<<endl;

do

{

cout<<"Enter your order code"<<endl;

cin>>a;

switch(a)

{

case 1:

count\_1++;

break;

case 2:

count\_2++;

break;

case 3:

count\_3++;

break;

case 4:

count\_4++;

break;

case 5:

count\_5++;

break;

default:

cout<<"Enter valid choice"<<endl;

}

//cout<<"Do you want to add something to your existing order??"<<endl;

//cin>>ch;

}while(a!=6);

}

void Zomato()

{

sum\_1z=count\_1\*99;

sum\_2z=count\_2\*105;

sum\_3z=count\_3\*300;

sum\_4z=count\_4\*150;

sum\_5z=count\_5\*205;

sum\_tz=sum\_1z+sum\_2z+sum\_3z+sum\_4z+sum\_5z;

count\_tz=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_z=20+(count\_tz\*3);

cgst=(2.5/100)\*sum\_tz;

sgst=(2.5/100)\*sum\_tz;

gst\_z=sum\_tz+(cgst+sgst);

cout<<endl<<"Your bill amount from Zomato is Rs."<<sum\_tz<<endl;

cout<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_z<<"\n";

cout<<"The time taken for delivery will be "<<delTime\_z<<"minutes"<<endl;

cout<<"to order from 'ZOMATO' enter 1"<<endl;

cout<<endl<<endl;

}

void Swiggy()

{

sum\_1s=count\_1\*106;

sum\_2s=count\_2\*199;

sum\_3s=count\_3\*250;

sum\_4s=count\_4\*100;

sum\_5s=count\_5\*180;

sum\_ts=sum\_1s+sum\_2s+sum\_3s+sum\_4s+sum\_5s;

count\_ts=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_s=20+(count\_ts\*4);

cgst=(2.5/100)\*sum\_ts;

sgst=(2.5/100)\*sum\_ts;

gst\_s=sum\_ts+(cgst+sgst);

cout<<"Your bill amount from Swiggy is Rs."<<sum\_ts<<endl;

cout<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"The time taken for delivery will be "<<delTime\_s<<"minutes"<<endl;

cout<<"to order from 'SWIGGY' enter 2"<<endl;

cout<<endl<<endl;

}

void Uber\_eats()

{

sum\_1u=count\_1\*115;

sum\_2u=count\_2\*120;

sum\_3u=count\_3\*250;

sum\_4u=count\_4\*135;

sum\_5u=count\_5\*215;

sum\_tu=sum\_1u+sum\_2u+sum\_3u+sum\_4u+sum\_5u;

count\_tu=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_u=20+(count\_tu\*5);

cgst=(2.5/100)\*sum\_tu;

sgst=(2.5/100)\*sum\_tu;

gst\_u=sum\_tu+(cgst+sgst);

cout<<"Your bill amount from Uber Eats is Rs."<<sum\_tu<<endl;

cout<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"The time taken for delivery will be "<<delTime\_u<<"minutes"<<endl;

cout<<"to order from 'UBER EATS' enter 3"<<endl;

cout<<endl<<endl;

}

void Food\_panda()

{

sum\_1f=count\_1\*135;

sum\_2f=count\_2\*90;

sum\_3f=count\_3\*200;

sum\_4f=count\_4\*200;

sum\_5f=count\_5\*199;

sum\_tf=sum\_1f+sum\_2f+sum\_3f+sum\_4f+sum\_5f;

count\_tf=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_f=20+(count\_tf\*6);

cgst=(2.5/100)\*sum\_tf;

sgst=(2.5/100)\*sum\_tf;

gst\_f=sum\_tf+(cgst+sgst);

cout<<"Your bill amount from Food Panda is Rs."<<sum\_tf<<endl;

cout<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_f<<endl;

cout<<"The time taken for delivery will be "<<delTime\_f<<"minutes"<<endl;

cout<<"to order from 'FOOD PANDA' enter 4"<<endl;

cout<<endl<<endl;

}

/\* void bill()

{

cout<<"GENERATING BILL";

for(long k=0;k<=1000000000;k++);

cout<<".";

for(long k=0;k<=1000000000;k++);

cout<<".";

for(long k=0;k<=1000000000;k++);

cout<<".";

for(long k=0;k<=1000000000;k++);

cout<<".";

for(long k=0;k<=1000000000;k++);

cout<<".";

cout<<" ";

cout<<"\f";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" KHATIRDARI "<<endl;

cout<<" KHAATE RAHO MANGATE RAHO "<<endl;

cout<<" Customer care number: 8085213684 "<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

// cout<<"NAME"<<"\t" "\t" "\t"<<user\_name;

}\*/

}obj1;

class Prithvi:private Amar

{

public:

void order()

{

Amar::order();

}

void Zomato()

{

sum\_1z=count\_1\*100;

sum\_2z=count\_2\*130;

sum\_3z=count\_3\*95;

sum\_4z=count\_4\*130;

sum\_5z=count\_5\*150;

sum\_tz=sum\_1z+sum\_2z+sum\_3z+sum\_4z+sum\_5z;

count\_tz=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_z=count\_tz\*3;

cgst=(2.5/100)\*sum\_tz;

sgst=(2.5/100)\*sum\_tz;

gst\_z=sum\_tz+(cgst+sgst);

cout<<endl<<"Your bill amount from Zomato is Rs."<<sum\_tz<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_z<<endl;;

cout<<"The time taken for delivery will be "<<delTime\_z<<"minutes"<<endl;

cout<<"to order from Zomato enter 1"<<endl;

cout<<endl<<endl;

}

void Swiggy()

{

sum\_1s=count\_1\*120;

sum\_2s=count\_2\*140;

sum\_3s=count\_3\*120;

sum\_4s=count\_4\*105;

sum\_5s=count\_5\*160;

sum\_ts=sum\_1s+sum\_2s+sum\_3s+sum\_4s+sum\_5s;

count\_ts=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_s=count\_ts\*4;

cgst=(2.5/100)\*sum\_ts;

sgst=(2.5/100)\*sum\_ts;

gst\_s=sum\_ts+(cgst+sgst);

cout<<"Your bill amount from Swiggy is Rs."<<sum\_ts<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"The time taken for delivery will be "<<delTime\_s<<"minutes"<<endl;

cout<<"to order from Swiggy enter 2"<<endl;

cout<<endl<<endl;

}

void Uber\_eats()

{

sum\_1u=count\_1\*90;

sum\_2u=count\_2\*155;

sum\_3u=count\_3\*110;

sum\_4u=count\_4\*110;

sum\_5u=count\_5\*210;

sum\_tu=sum\_1u+sum\_2u+sum\_3u+sum\_4u+sum\_5u;

count\_tu=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_u=count\_tu\*5;

cgst=(2.5/100)\*sum\_tu;

sgst=(2.5/100)\*sum\_tu;

gst\_u=sum\_tu+(cgst+sgst);

cout<<"Your bill amount from Uber Eats is Rs."<<sum\_tu<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"The time taken for delivery will be "<<delTime\_u<<"minutes"<<endl;

cout<<"to order from Uber Eats enter 3"<<endl;

cout<<endl<<endl;

}

void Food\_panda()

{

sum\_1f=count\_1\*135;

sum\_2f=count\_2\*115;

sum\_3f=count\_3\*100;

sum\_4f=count\_4\*95;

sum\_5f=count\_5\*175;

sum\_tf=sum\_1f+sum\_2f+sum\_3f+sum\_4f+sum\_5f;

count\_tf=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_f=count\_tf\*6;

cgst=(2.5/100)\*sum\_tf;

sgst=(2.5/100)\*sum\_tf;

gst\_f=sum\_tf+(cgst+sgst);

cout<<"Your bill amount from Food Panda is Rs."<<sum\_tf<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_f<<endl;

cout<<"The time taken for delivery will be "<<delTime\_f<<"minutes"<<endl;

cout<<"to order from Food Panda enter 4"<<endl;

cout<<endl;

}

void Decide(int a\_status)

{

int status=a\_status;

Amar::Decide(status);

}

}obj2;

class Kailash:private Amar

{

public:

void order()

{

Amar::order();

}

void Zomato()

{

sum\_1z=count\_1\*199;

sum\_2z=count\_2\*105;

sum\_3z=count\_3\*225;

sum\_4z=count\_4\*215;

sum\_5z=count\_5\*145;

sum\_tz=sum\_1z+sum\_2z+sum\_3z+sum\_4z+sum\_5z;

count\_tz=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_z=count\_tz\*3;

cgst=(2.5/100)\*sum\_tz;

sgst=(2.5/100)\*sum\_tz;

gst\_z=sum\_tz+(cgst+sgst);

cout<<endl<<"Your bill amount from Zomato is Rs."<<sum\_tz<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_z<<endl;

cout<<"The time taken for delivery will be "<<delTime\_z<<"minutes"<<endl;

cout<<"to order from Zomato enter 1"<<endl;

cout<<endl<<endl;

}

void Swiggy()

{

sum\_1s=count\_1\*106;

sum\_2s=count\_2\*205;

sum\_3s=count\_3\*199;

sum\_4s=count\_4\*300;

sum\_5s=count\_5\*215;

sum\_ts=sum\_1s+sum\_2s+sum\_3s+sum\_4s+sum\_5s;

count\_ts=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_s=count\_ts\*4;

cgst=(2.5/100)\*sum\_ts;

sgst=(2.5/100)\*sum\_ts;

gst\_s=sum\_ts+(cgst+sgst);

cout<<"Your bill amount from Swiggy is Rs."<<sum\_ts<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"The time taken for delivery will be "<<delTime\_s<<"minutes"<<endl;

cout<<"to order from Swiggy enter 2"<<endl;

cout<<endl<<endl;

}

void Uber\_eats()

{

sum\_1u=count\_1\*205;

sum\_2u=count\_2\*115;

sum\_3u=count\_3\*205;

sum\_4u=count\_4\*255;

sum\_5u=count\_5\*200;

sum\_tu=sum\_1u+sum\_2u+sum\_3u+sum\_4u+sum\_5u;

count\_tu=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_u=count\_tu\*5;

cgst=(2.5/100)\*sum\_tu;

sgst=(2.5/100)\*sum\_tu;

gst\_u=sum\_tu+(cgst+sgst);

cout<<"Your bill amount from Uber Eats is Rs."<<sum\_tu<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"The time taken for delivery will be "<<delTime\_u<<"minutes"<<endl;

cout<<"to order from Uber Eats enter 3"<<endl;

cout<<endl<<endl;

}

void Food\_panda()

{

sum\_1f=count\_1\*185;

sum\_2f=count\_2\*190;

sum\_3f=count\_3\*215;

sum\_4f=count\_4\*225;

sum\_5f=count\_5\*255;

sum\_tf=sum\_1f+sum\_2f+sum\_3f+sum\_4f+sum\_5f;

count\_tf=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_f=count\_tf\*6;

cgst=(2.5/100)\*sum\_tf;

sgst=(2.5/100)\*sum\_tf;

gst\_f=sum\_tf+(cgst+sgst);

cout<<"Your bill amount from Food Panda is Rs."<<sum\_tf<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_f<<endl;

cout<<"The time taken for delivery will be "<<delTime\_f<<"minutes"<<endl;

cout<<"to order from Food Panda enter 4"<<endl;

cout<<endl;

}

void Decide(int a\_status)

{

int status=a\_status;

Amar::Decide(status);

}

}obj3;

class ChoHeaven:private Amar

{

public:

void order()

{

Amar::order();

}

void Zomato()

{

sum\_1z=count\_1\*100;

sum\_2z=count\_2\*230;

sum\_3z=count\_3\*115;

sum\_4z=count\_4\*190;

sum\_5z=count\_5\*250;

sum\_tz=sum\_1z+sum\_2z+sum\_3z+sum\_4z+sum\_5z;

count\_tz=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_z=count\_tz\*3;

cgst=(2.5/100)\*sum\_tz;

sgst=(2.5/100)\*sum\_tz;

gst\_z=sum\_tz+(cgst+sgst);

cout<<endl<<"Your bill amount from Zomato is Rs."<<sum\_tz<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_z<<endl;

cout<<"The time taken for delivery will be "<<delTime\_z<<"minutes"<<endl;

cout<<"to order from Zomato enter 1"<<endl;

cout<<endl<<endl;

}

void Swiggy()

{

sum\_1s=count\_1\*150;

sum\_2s=count\_2\*210;

sum\_3s=count\_3\*112;

sum\_4s=count\_4\*205;

sum\_5s=count\_5\*199;

sum\_ts=sum\_1s+sum\_2s+sum\_3s+sum\_4s+sum\_5s;

count\_ts=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_s=count\_ts\*4;

cgst=(2.5/100)\*sum\_ts;

sgst=(2.5/100)\*sum\_ts;

gst\_s=sum\_ts+(cgst+sgst);

cout<<"Your bill amount from Swiggy is Rs."<<sum\_ts<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"The time taken for delivery will be "<<delTime\_s<<"minutes"<<endl;

cout<<"to order from Swiggy enter 2"<<endl;

cout<<endl<<endl;

}

void Uber\_eats()

{

sum\_1u=count\_1\*190;

sum\_2u=count\_2\*199;

sum\_3u=count\_3\*99;

sum\_4u=count\_4\*170;

sum\_5u=count\_5\*210;

sum\_tu=sum\_1u+sum\_2u+sum\_3u+sum\_4u+sum\_5u;

count\_tu=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_u=count\_tu\*5;

cgst=(2.5/100)\*sum\_tu;

sgst=(2.5/100)\*sum\_tu;

gst\_u=sum\_tu+(cgst+sgst);

cout<<"Your bill amount from Uber Eats is Rs."<<sum\_tu<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"The time taken for delivery will be "<<delTime\_u<<"minutes"<<endl;

cout<<"to order from Uber Eats enter 3"<<endl;

cout<<endl<<endl;

}

void Food\_panda()

{

sum\_1f=count\_1\*105;

sum\_2f=count\_2\*215;

sum\_3f=count\_3\*100;

sum\_4f=count\_4\*115;

sum\_5f=count\_5\*225;

sum\_tf=sum\_1f+sum\_2f+sum\_3f+sum\_4f+sum\_5f;

count\_tf=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_f=count\_tf\*6;

cgst=(2.5/100)\*sum\_tf;

sgst=(2.5/100)\*sum\_tf;

gst\_f=sum\_tf+(cgst+sgst);

cout<<"Your bill amount from Food Panda is Rs."<<sum\_tf<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_f<<endl;

cout<<"The time taken for delivery will be "<<delTime\_f<<"minutes"<<endl;

cout<<"to order from Food Panda enter 4"<<endl;

cout<<endl;

}

void Decide(int a\_status)

{

int status=a\_status;

Amar::Decide(status);

}

}obj4;

class Aditi:private Amar

{

public:

void order()

{

Amar::order();

}

void Zomato()

{

sum\_1z=count\_1\*199;

sum\_2z=count\_2\*105;

sum\_3z=count\_3\*225;

sum\_4z=count\_4\*215;

sum\_5z=count\_5\*145;

sum\_tz=sum\_1z+sum\_2z+sum\_3z+sum\_4z+sum\_5z;

count\_tz=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_z=count\_tz\*3;

cgst=(2.5/100)\*sum\_tz;

sgst=(2.5/100)\*sum\_tz;

gst\_z=sum\_tz+(cgst+sgst);

cout<<endl<<"Your bill amount from Zomato is Rs."<<sum\_tz<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_z<<endl;

cout<<"The time taken for delivery will be "<<delTime\_z<<"minutes"<<endl;

cout<<"to order from Zomato enter 1"<<endl;

cout<<endl<<endl;

}

void Swiggy()

{

sum\_1s=count\_1\*106;

sum\_2s=count\_2\*99;

sum\_3s=count\_3\*199;

sum\_4s=count\_4\*300;

sum\_5s=count\_5\*205;

sum\_ts=sum\_1s+sum\_2s+sum\_3s+sum\_4s+sum\_5s;

count\_ts=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_s=count\_ts\*4;

cgst=(2.5/100)\*sum\_ts;

sgst=(2.5/100)\*sum\_ts;

gst\_s=sum\_ts+(cgst+sgst);

cout<<"Your bill amount from Swiggy is Rs."<<sum\_ts<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_s<<endl;

cout<<"The time taken for delivery will be "<<delTime\_s<<"minutes"<<endl;

cout<<"to order from Swiggy enter 2"<<endl;

cout<<endl<<endl;

}

void Uber\_eats()

{

sum\_1u=count\_1\*205;

sum\_2u=count\_2\*215;

sum\_3u=count\_3\*205;

sum\_4u=count\_4\*225;

sum\_5u=count\_5\*200;

sum\_tu=sum\_1u+sum\_2u+sum\_3u+sum\_4u+sum\_5u;

count\_tu=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_u=count\_tu\*5;

cgst=(2.5/100)\*sum\_tu;

sgst=(2.5/100)\*sum\_tu;

gst\_u=sum\_tu+(cgst+sgst);

cout<<"Your bill amount from Uber Eats is Rs."<<sum\_tu<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_u<<endl;

cout<<"The time taken for delivery will be "<<delTime\_u<<"minutes"<<endl;

cout<<"to order from Uber Eats enter 3"<<endl;

cout<<endl<<endl;

}

void Food\_panda()

{

sum\_1f=count\_1\*185;

sum\_2f=count\_2\*190;

sum\_3f=count\_3\*215;

sum\_4f=count\_4\*225;

sum\_5f=count\_5\*225;

sum\_tf=sum\_1f+sum\_2f+sum\_3f+sum\_4f+sum\_5f;

count\_tf=count\_1+count\_2+count\_3+count\_4+count\_5;

delTime\_f=count\_tf\*6;

cgst=(2.5/100)\*sum\_tf;

sgst=(2.5/100)\*sum\_tf;

gst\_f=sum\_tf+(cgst+sgst);

cout<<"Your bill amount from Food Panda is Rs."<<sum\_tf<<endl;

cout<<"Your bill amount after GST is Rs."<<gst\_f<<endl;

cout<<"The time taken for delivery will be "<<delTime\_f<<"minutes"<<endl;

cout<<"to order from Food Panda enter 4"<<endl;

cout<<endl;

}

void Decide(int a\_status)

{

int status=a\_status;

Amar::Decide(status);

}

}obj5;

void choice(int a)

{

int status =a;

int rest\_choice;

cout<<"1. Amar"<<endl;

cout<<"2. Prithvi cafe"<<endl;

cout<<"3. Kailash Parbat"<<endl;

cout<<"4. Chocolate Heaven"<<endl;

cout<<"5. Aditi"<<endl;

cin>>rest\_choice;

string explore;

switch(rest\_choice)

{

case 1:

{

cout<<"You have chosen to order from Amar"<<endl;

string lines\_a;

ifstream menu\_amar;

menu\_amar.open("menu1fin.txt");

while(!menu\_amar.eof())

{

getline(menu\_amar, lines\_a);

//cout<<lines<<endl;

std::cout<<lines\_a<<endl;

}

cout<<"To switch another restaurant enter 'explore' else enter any key"<<endl;

cin>>explore;

if(explore=="Explore" || explore=="explore")

{

choice(status);

}

else

{

obj1.order();

obj1.Zomato();

obj1.Swiggy();

obj1.Uber\_eats();

obj1.Food\_panda();

obj1.Decide(status);

}

}

break;

case 2:

{

cout<<"You have chosen to order from Prithvi"<<endl;

string lines\_p;

ifstream menu\_prithvi;

menu\_prithvi.open("prithvi.txt");

while(!menu\_prithvi.eof())

{

getline(menu\_prithvi, lines\_p);

//cout<<lines<<endl;

std::cout<<lines\_p<<endl;

}

cout<<"To switch another restaurant enter 'explore' else enter any key"<<endl;

cin>>explore;

if(explore=="Explore" || explore=="explore")

{

choice(status);

}

else

{

obj2.order();

obj2.Zomato();

obj2.Swiggy();

obj2.Uber\_eats();

obj2.Food\_panda();

obj2.Decide(status);

}

}

break;

case 3:

{

cout<<"You have chosen to order from Kailash Parbat"<<endl;

string lines\_kp;

ifstream menu\_kp;

menu\_kp.open("kp.txt");

while(!menu\_kp.eof())

{

getline(menu\_kp, lines\_kp);

//cout<<lines<<endl;

std::cout<<lines\_kp<<endl;

}

cout<<"To switch another restaurant enter 'explore' else enter any key"<<endl;

cin>>explore;

if(explore=="Explore" || explore=="explore")

{

choice(status);

}

else

{

obj3.order();

obj3.Zomato();

obj3.Swiggy();

obj3.Uber\_eats();

obj3.Food\_panda();

obj3.Decide(status);

}

}

break;

case 4:

{

cout<<"You have chosen to order from Chocolate Heaven"<<endl;

string lines\_ch;

ifstream menu\_ch;

menu\_ch.open("ch.txt");

while(!menu\_ch.eof())

{

getline(menu\_ch, lines\_ch);

//cout<<lines<<endl;

std::cout<<lines\_ch<<endl;

}

cout<<"To switch another restaurant enter 'explore' else enter any key"<<endl;

cin>>explore;

if(explore=="Explore" || explore=="explore")

{

choice(status);

}

else

{

obj4.order();

obj4.Zomato();

obj4.Swiggy();

obj4.Uber\_eats();

obj4.Food\_panda();

obj4.Decide(status);

}

}

break;

case 5:

{

cout<<"You have chosen to order from Aditi"<<endl;

string lines\_ad;

ifstream menu\_aditi;

menu\_aditi.open("aditi.txt");

while(!menu\_aditi.eof())

{

getline(menu\_aditi, lines\_ad);

//cout<<lines<<endl;

std::cout<<lines\_ad<<endl;

}

cout<<"To switch another restaurant enter 'explore' else enter any key"<<endl;

cin>>explore;

if(explore=="Explore" || explore=="explore")

{

choice(status);

}

else

{

obj5.order();

obj5.Zomato();

obj5.Swiggy();

obj5.Uber\_eats();

obj5.Food\_panda();

obj5.Decide(status);

}

}

break;

default:

cout<<"Wrong choice"<<endl;

choice(status);

}

}

class LoRe

{

string user\_name,user\_password,address;

long long user\_phone;

string names,passwords;

int counter\_n=0,counter\_p=0;

int i=0;

public:

int u\_register()

{

cout<<"Enter your name"<<endl;

cin>>user\_name;

ofstream save;

save.open("name.txt",ios::app);

save<<endl<<user\_name;

save.close();

cout<<"Create a password (alphabets only)"<<endl;

cin>>user\_password;

save.open("password.txt",ios::app);

save<<endl<<user\_password;

save.close();

cout<<"Enter your Phone no."<<endl;

cin>>user\_phone;

ofstream phone;

phone.open("phone.txt",ios::app);

phone<<endl<<user\_phone;

phone.close();

cout<<"Enter your delivery address"<<endl;

cin>>address;

ofstream address1;

address1.open("address.txt",ios::app);

address1<<endl<<address;

address1.close();

cout<<endl;

cout<<"WELCOME"<<endl<<" You have new code 'NEW\_USER' use it to get 13% of on any order"<<endl;

return 1;

}

int login()

{

ifstream name;

int y;

cout<<"Enter username"<<endl;

cin>>user\_name;

cout<<"Enter password"<<endl;

cin>>user\_password;

name.open("name.txt");

while(!name.eof())

{

getline(name,names);

if(names==user\_name)

{break;}

else{

counter\_n++;

}

}

name.close();

ifstream pass;

pass.open("password.txt");

while(pass.good())

{

getline(pass,passwords);

if(passwords==user\_password)

{

break;}

else{

counter\_p++;

}

}

if(counter\_n==counter\_p)

{

cout<<"Successfully logged in!! "<<endl;

}

else

{

cout<<"Login error"<<endl;

cout<<"Redirecting to registration page"<<endl;

u\_register();

}

}

}obj;

void ratings()

{

int rating;

cout<<endl;

char feedback[100];

cout<<"Did we make ordering food easier?"<<endl;

cout<<"Rate us NOW!(0-5)"<<endl;

cin>>rating;

if(rating>5)

{

cout<<"Please rate out of five"<<endl;

ratings();

}

else

{

if(rating<=3)

{

cout<<"Help us improve our service by giving us feedback"<<endl;

cin>>feedback;

cout<<"Thank you fro your feedback :)"<<endl;

cout<<"We will make sure to look into it"<<endl;

}

cout<<endl;

cout<<"Enjoy your meal"<<endl;

cout<<"Hope to see you again:)"<<endl;

cout<<" --KHATIR DARI"<<endl<<endl<<endl<<endl;

}

}

int main()

{

int status;

logo();

cout<<endl;

cout<<"Welcome to KHATIRDARI--'Khate Raho Mangaate Raho!!'"<<endl;

cout<<endl;

random\_otp();

string lore;

cout<<endl;

cout<<"Do you want to login or register?";

cin>>lore;

if(lore == "Login" || lore== "login" )

{

status=obj.login();

}

else if(lore=="register" || lore=="Register")

{

status=obj.u\_register();

}

cout<<"Choose Your Restaurant"<<endl;

choice(status);

ratings();

return 0;

}